

CZECH TECHNICAL UNIVERSITY IN PRAGUE  
FACULTY OF ELECTRICAL ENGINEERING  
DEPARTMENT OF CYBERNETICS  
MULTI-ROBOT SYSTEMS

SELF-STABILIZED GROUPS  
OF MICRO AERIAL  
VEHICLES

Habilitation Thesis

Martin Saska

Prague, Feb, 2019



---

## ABSTRACT - EN

---

The goal of this habilitation thesis is to aggregate our research on stabilization and control of cooperating robots in real environments without the need to build a dedicated infrastructure. The thesis presents methodologies and enabling techniques for controlling teams of autonomous robots acting in a shared working space, using onboard localization sensors only.

The main contribution of this work lies in the design of methods for motion planning and stabilization of formations of ground and aerial robots, and for bio-inspired behavioral control patterns for swarms of multi-rotor micro aerial vehicles (MAVs). All presented approaches are designed to satisfy the constraints imposed by a system of onboard relative localization of teammates, which was designed as the main sensor to enable the deployment of groups of cooperating robots in environments without external localization. The attached publications present theoretical studies specifying the requirements for the relative localization sensor to satisfy the stability of MAV swarms and convergence of motion of compact formations.

In addition to the theoretical contribution, a large part of this document is dedicated to applications of the methodologies, since robotics is an application-oriented and application-motivated scientific field. Most of the methods presented here have been verified experimentally in environments reflecting the target applications. Continuous experimental verification of the methods is an added value of the results that have been achieved. Direct interconnection between the methods designed for MAV group control and stabilization and the properties of the MAV sensors described in the real-world experiments is a crucial attribute of the work presented here. It has enabled the scientific results to be transferred into real deployment and into industrial applications, as has been shown in numerous examples in the thesis. Intensive validation of system performance in demanding environments was also a key factor in the MBZIRC 2017 competition, where our solution for a multi-robot challenge outperformed the results of all other competitors, most of whom had relied on laboratory testing only. Cohesive linkage between theoretical principles, on the one hand, and real experience and industrial solutions, on the other, has also provided a decisive competitive advantage for students of our team in their further career.



---

## ABSTRACT - CZ

---

Cílem této habilitační práce je shrnout výzkum směřující k možnosti stabilizovat a řídit skupiny spolupracujících robotů v reálném prostředí bez nutnosti budovat speciální infrastrukturu. Práce prezentuje metodologii a jednotlivé nutné technologie umožňující řídit týmy autonomních robotů ve sdíleném pracovním prostoru pouze s využitím palubních senzorů pro jejich lokalizaci.

Hlavním přínosem práce je návrh metod plánování pohybu a stabilizace formací pozemních a vzdušných robotů a přírodou inspirovaných vzorců chování rojů malých bezpilotních helikoptér (v práci značených zkratkou MAV - z ang. Micro Aerial Vehicle). Všechny tyto přístupy jsou navrhovány s cílem respektovat omezení daná systémem palubní relativní lokalizace sousedních helikoptér skupiny. Systém relativní lokalizace byl navržen a je postupně inovován s cílem umožnit nasazení skupin spolupracujících robotů v prostředí bez externí lokalizace. V publikacích přiložených k této práci jsou uvedeny teoretické studie specifikující požadavky na tento senzor, které zajistí stabilitu MAV roje a konvergenci pohybu kompaktní formace do požadovaného cíle.

Kromě teoretického přínosu práce se její velká část zabývá konkrétními aplikacemi navržené metodologie, protože robotika je aplikačně orientovaná a motivovaná vědecká disciplína. Naprostá většina prezentovaných metod byla kromě teoretických analýz též experimentálně verifikována v prostředích maximálně odpovídajících cílové aplikaci. Kontinuální experimentální verifikace vyvíjených metod odlišuje dosažené výsledky od většiny publikovaných přístupů skupinové robotiky, které často nerespektují omezení reálných systémů. Právě přímá provázanost metod řízení a stabilizace MAV týmů s vlastnostmi jejich senzorů je klíčový prvek umožňující transfer výsledků prezentovaného základního výzkumu do reálného nasazení a průmyslových aplikací, což je v práci demonstrováno na několika konkrétních příkladech. Intenzivní průběžné ověřování funkcionalit vyvíjených přístupů v reálných prostředích se také ukázalo jako rozdílový faktor v soutěži MBZIRC 2017, kde naše řešení vícerobotických úloh převýšilo výsledky ostatních týmů testujících své systémy převážně v laboratořích. Kombinace teoretických přístupů s reálnou praxí a průmyslovými řešeními se také ukázala jako velká konkurenční výhoda našich studentů na trhu práce.



---

## CANDIDATE'S CONTRIBUTIONS IN PARTICULAR PAPERS

---

Papers presented in Appendices A, B, C, D, E and I summarize the candidate's theoretical work in formations and swarms control. The candidate is the main author in these publications, and the work was carried out with the help of his colleagues and students mainly related to experimental verification. Papers presented in Appendixes F, H, J, and K summarize the most important contributions of the candidate's students. The candidate's main contribution was in providing the initial motivation for the entire work, the theoretical design of the methodologies, ongoing advisory work on implementation and experimental verification, and support and direct contributions in the publication phase. The paper presented in Appendix G is an interesting example of joint work carried out by three groups, led by Tomas Krajník, Jan Faigl, and the candidate, which influenced further scientific work of all co-authors. Candidate's contribution was mainly in the initial design, and in motivating the entire work. He also significantly helped with experimental verification of the system and with its deployment in numerous real-world scenarios.

The exact specification of the contributions to each paper is expressed as a percentage in the attached list of publications, based on an agreement with the co-authors.



---

## ACKNOWLEDGEMENTS

---

I would like to express my gratitude to all of my group members: Vojtěch Vonásek, Petr Štěpán, Tomáš Báča, Vojtěch Spurný, Robert Pěnička, Viktor Walter, Matěj Petrlík, Matouš Vrba, and Daniel Heřt, who have become my friends. Without their enormous help I would not have been able to achieve the presented results. I would also like to thank other team members: Tomáš Krajník, Giuseppe Loianno, Justin Thomas, Dinesh D. Thakur, Vijay Kumar, Tomáš Svoboda, and Karel Zimmermann, who helped us to succeed in the MBZIRC competition, which was the breakpoint that accelerated my research team in the worldwide context. My thanks go to Jan Faigl, Antonio Franchi, Klaus Schilling and Dušan M. Stipanovič for helping me in various stages of my research career. My thanks go to Libor Přeučil, who motivated me to establish my own robotic groups, and to Jan Kybic and Pavel Ripka, who provided great support for me to really do it. Above all, I am grateful to my wife Věra and to my sons.

During my research, I have been supported by Czech Technical University grants SGS10/195/OHK3/2T/13, SGS14/073/OHK3/1T/13, SGS15/157/OHK3/2T/13, SGS17/187/OHK3/3T/13. The Ministry of Education of the Czech Republic has funded my work through grants LH11053 and 7AMB16FR017. The Czech Science Foundation supported this work through project GPP103/12/P756, 17-07690S, and 17-16900Y, the CESNET association provided support through two research projects, and the Ministry of Culture through project NAKI II no. DG18P02OVV069. The European Union has funded this work through Replicator 216240 and Symbion 216342 projects. Khalifa University has supported this research through MBZIRC 2017 and MBZIRC 2020 funding.



---

## COPYRIGHT

---

This work is a compilation of papers published throughout the course of my research efforts. The works included in this habilitation thesis are presented and reprinted in accordance with the copyright agreements with the respective publishers. Further copying or reprinting can be done exclusively with the permission of the respective publishers.



---

## CONTENTS

---

1	INTRODUCTION	15
2	COMPACT MUTUALLY STABILIZED FORMATIONS	17
2.1	Heterogeneous formations of cooperating aerial and ground robots . . . . .	17
2.2	Fault-tolerant formation flying . . . . .	19
2.3	Compact formations of aerial unmanned vehicles . . . . .	20
2.4	Complex manoeuvres of compact MAV formations . . . . .	22
3	SWARMS OF CLOSELY STABILIZED MICRO AERIAL VEHICLES	25
3.1	Bio-inspired boids swarming model . . . . .	25
3.2	Coherent swarming with minimum sensory requirements . . . . .	28
4	MUTUAL ONBOARD LOCALIZATION OF MULTIPLE COOPERATING ROBOTS	31
4.1	Localization patterns detectable in the visible spectrum . . . . .	31
4.2	Ultraviolet-based relative localization . . . . .	33
4.3	Localization without patterns purposely placed on vehicles that are being localized . . . . .	35
5	APPLICATIONS OF THE PROPOSED CONCEPTS	39
5.1	Documentation of dark areas of historical buildings with the use of MAV formations . . . . .	40
5.2	Autonomous landing on a moving platform . . . . .	44
5.3	Cooperative autonomous search, grasping and delivering of objects . . . . .	46
6	CONCLUSION	51
A	KEY ARTICLE [1] - INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH 2014	65
B	KEY ARTICLE [2] - JOURNAL OF INTELLIGENT AND ROBOTIC SYSTEMS 2014	87
C	KEY ARTICLE [3] - AUTONOMOUS ROBOTS 2017	109
D	KEY ARTICLE [4] - ICARCV 2016	137
E	KEY ARTICLE [5] - ICRA 2014	145
F	KEY ARTICLE [6] - ECMR 2017	153
G	KEY ARTICLE [7] - JOURNAL OF INTELLIGENT AND ROBOTIC SYSTEMS 2014	161
H	KEY ARTICLE [8] - IEEE ROBOTICS AND AUTOMATION LETTERS (RA-L)	187
I	KEY ARTICLE [9] - IEEE ETFA 2017	197
J	KEY ARTICLE [10] - JOURNAL OF FIELD ROBOTICS 2018	207
K	KEY ARTICLE [11] - JOURNAL OF FIELD ROBOTICS 2018	235



---

## INTRODUCTION

---

Autonomous control of Micro Aerial Vehicles (MAVs) and stabilization of groups of MAVs are currently fast-growing research fields in mobile robotics. This area may be considered as an intersection and a continuation of the massive research endeavour given to multi-robot systems of cooperating Unmanned Ground Vehicles (UGV), and the progress in technologies of Unmanned Aerial Vehicles (UAVs) together with their miniaturization towards small-size MAVs. In this habilitation thesis, contributions to the coordination and stabilization of groups of ground and aerial autonomous robots and precise control and motion planning of micro aerial robots with limited computational and sensory capabilities will be summarized and situated into the context of state-of-the-art robotic research (see Fig. 1).



Figure 1: Examples of the deployment of the presented systems in real-world environments. Upper left: A formation documenting the Church of St. Moritz in Olomouc using a smart lighting technique (see section 5.1 for details). Upper right: Tests of enabling technologies for swarming in forest-like work-spaces (section 3.1). Bottom: A compact formation of self-stabilized MAVs (section 2).

The deployment of groups of MAVs has been motivated by the increasing robustness of robotic systems, and it has motivated research of methods for using UGV teams since the beginning of this century. Researchers can take advantage of the possible redundancy of small and cheap vehicles and fusion information from distributed simple sensors, instead of using a single well equipped aerial vehicle. In addition, small flying robots enable better manoeuvrability and better reachability of desired locations in a cluttered workspace, for

which they are especially appealing. Moreover, there are numerous applications that cannot be solved by a single vehicle, and multirobot deployment is required (see section 5 for examples).

Current research on multi-MAV systems is often limited to approaches applicable only in laboratory conditions using a precise motion capture system (e.g. Vicon - <http://www.vicon.com/>). This enables MAVs to be localised with precision within fractions of millimetres, but requires the installation of fast cameras in the environment prior to the mission [44, 45, 46, 47, 48]. In rare cases, simple methods of MAV cooperation are designed for use with Global Navigation Satellite Systems (GNSS) [49, 50, 51, 52]. Reliance on GNSS, such as GPS, limits the deployment of MAV teams in workspaces with obstacles (urban environment, forests, canyons) and indoors (underground mines, caves, factory halls, warehouses). Nevertheless, MAVs are much better suited exactly for these environments than larger UAVs designed for open sky missions, where a GNSS signal is required. Another significant disadvantage of using GNSS as the main localization system is its limited precision and reliability, which makes it impossible to form compact formations and swarms flying close to obstacles.

The overall objective of the research achievements presented here, and also the main research agenda of the candidate's Multi-robot systems group at CTU in Prague (<http://mrs.felk.cvut.cz/>) is to focus on aspects of the deployment of self-stabilized MAV groups in GNSS-denied environments. In particular, we are focused on designing and analysing systems that rely only on onboard sensors and computational power, using direct visual relative localization among the cooperating robots. Methods for direct visual localization [12, 13], which we are also focused on, are tools being used by the applicant and his group for stabilizing MAV formations in fixed relative positions [14, 15], for stabilizing dynamic MAV swarms [16, 17, 18, 19], and stabilizing heterogeneous teams of MAVs and UGVs [1, 2], [20, 21] in an arbitrary (indoor and outdoor) GPS-denied environment and with no external localization. Aspects of all these various multi-MAV systems will be discussed in this thesis, together with a theory that specifies the requirements for their deployment in real-world conditions.

In the presented work, two main streams of multi-robot stabilization and control are followed: centralized formations and decentralized swarms (see Fig. 1 for examples). The required mutual positions for all robots within the team are given and the overall shape of the group is specified by high-level planning in the first control scheme, for which the term formation is used here. This approach enables us to estimate the future behaviour of the group as well as of particular robots, and to analyse properties of the system, including its robustness, its convergence into the required equilibrium, and its stability. However, these methods often require communication between the robots, centralized coordination, and they are sensitive to the failure of a single robot, mainly in the frequently used leader-follower scheme [53, 54, 55, 56, 57, 58, 59].

The second stream that has been investigated, and that stands as the main motivation for the research agenda of the candidate's team, is focused on fully-decentralized MAV stabilization with no direct communication between the robots in the team. This approach to MAV group control, which is called swarming in this work, brings high scalability and adaptability [49, 60, 61, 62]. We have even shown that increasing the number of robots in the team increases the group stability, while keeping the same complexity of swarming rules and computational requirements. The possibility of group stabilization and coordinated motion without a need for communication is a crucial factor in robustness to external environmental conditions. In hundreds of multi-robot experiments that have been realised, we have observed that communication within large multi-MAV teams is the main bottleneck in current robotic technology. In real-world conditions, a problematic task is to establish a robust communication mesh even for groups of just 4-5 flying robots, and keeping connectivity in a larger group is impossible with currently available devices.

---

 COMPACT MUTUALLY STABILIZED FORMATIONS
 

---

The first example of stabilizing multiple closely-cooperating robots is a centralized approach to formation movement, where the robots keep required mutual positions within the group. In this approach, all robots assigned as followers follow the path of a real or virtual leader at a given distance behind the leader and within a given deviation from the path. This allows us to reliably compose a formation of arbitrary shape if there are safe distances between the robots satisfied (see Fig. 2 for an example of such defined formation and Fig. 3 for examples of an experimental deployment of the system).

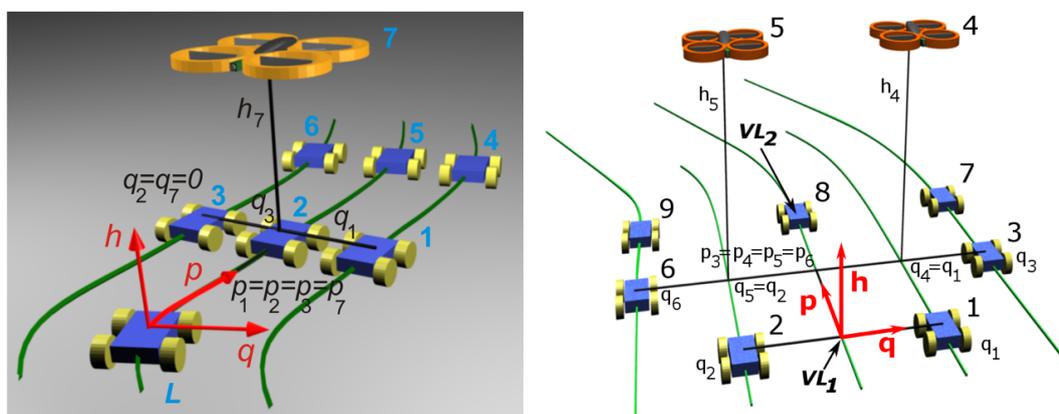


Figure 2: Positions of robots in heterogeneous formations defined in a curvilinear coordinate system with axes  $p, q, h$ . Left: A formation with a real leader. Right: A formation with two virtual leaders,  $VL_1$  for forward motion and  $VL_2$  for backward motion. For more details and examples of the deployment of such formations, see [1]. Source: [1].

### 2.1 HETEROGENEOUS FORMATIONS OF COOPERATING AERIAL AND GROUND ROBOTS

The simplest case of achieving a compact formation-flying ability with onboard sensors and limited computational resources is multi-robot cooperation within heterogeneous teams of ground and aerial vehicles. In such a system, unmanned ground vehicles (UGVs) help to stabilize the overall formation relatively to the surrounding environment and to repress undesired motion oscillations. The MAVs in the formation do not need information on their velocity relatively to the environment, which is difficult to achieve. Knowledge of the current velocity is crucial for stabilization of a single MAV as well as for stabilization of groups composed of MAVs only. The basic ideas of our approach to the planning and stabilization of heterogeneous MAV-UGV formations are presented in [21] and extended with experiments and convergence analyses in [1]. In Appendix A (paper [1]), a leader-follower formation driving approach adapted for onboard visual relative localization of heterogeneous teams of unmanned helicopters (quadrotors) and ground robots is described. The main motivation for the work in [1] is the deployment of large teams of



Figure 3: Examples of an MAV formation flying in an outdoor workspace.

unmanned vehicles in real-world search and rescue (SAR) missions, where a multi-robot system has to explore large areas cooperatively in a short time, and where precise relative localization among the robots is required. In these tasks, robots cannot rely on a pre-installed global localization infrastructure, which is usually not available, or is damaged, in sites under SAR exploration. The deployment of multi-robot systems in environments where external localization is not available is an objective that runs through this entire document.

In [1], the localization system required for group stabilization and its movement in a coordinated manner is based on simple light-weight bottom cameras mounted on the MAVs and identification patterns used for the mutual localization sensor placed on both the UGVs and the MAVs. With this top-view approach, the problem of loss of direct visibility, which is a common source of localization uncertainty in SAR missions, can be better tackled. This type of problem occurs if visual relative localization systems are employed for ground robots operating in a workspace with scattered obstacles, as is common in SAR scenarios. The ability to localize robots from the top view increases the robustness and the precision of determining the relative position. The live video stream from the flying cameras can be an additional useful source of information for human operators supervising the SAR mission.

The proposed MAV-UGV standalone system provides a light-weight, low-cost and efficient solution, and could be an enabling technology for extensive utilization of simple micro-scale robots in demanding scenarios. Article [1] focuses on theoretical and implementation aspects of the formation driving mechanism suited for the real-world deployment of autonomous robots relying on top-view relative localization (referred to as the “hawk-eye concept” in [1]). The formation driving approach relies on Model Predictive Control (MPC), which allows the involvement of constraints imposed by vehicles (mobility constraints), obstacles (environment constraints), and inter-vehicle relations into the formation driving. The inter-vehicle relations are specified by the shape of the formation feasible for the *hawk-eye* like relative localization approach.

The MPC technique is used for stabilizing the followers in their desired positions behind the leader, and for planning the leader trajectory into the desired goal area. A new MPC concept combining trajectory planning into the desired goal region and immediate control of the formation in a single optimization process was proposed in [1]. This research extends the candidate’s previous work on stabilization of formations of ground robots (for the technical and implementation details of the UGV formation driving system, see [22, 23, 24, 25, 26, 27, 28], while the theoretical aspects and analyses of convergence are presented

in [29, 30, 31]). The method proposed in [1] can respond continuously to changes in the vicinity of the formation while keeping the cohesion of the actual control inputs with the directions of movement of the MAV-UGV formation in the future. Furthermore, a new obstacle avoidance function for multi-vehicle trajectory planning, which includes a model of the group that respects the restrictions of the *hawk-eye* concept, was integrated into the MPC schema. The obstacle avoidance function ensures direct visibility between MAVs and UGVs, which is a crucial requirement for the top-view relative localization approach. In addition to the novel MPC schema, sound theoretical analyses of the convergence of the entire system were provided in [1] to be able to specify requirements for practical utilization of the method, taking into account functional properties of onboard sensors and actuators.

## 2.2 FAULT-TOLERANT FORMATION FLYING

As stated above, and discussed more in greater detail in the concluding remarks of this thesis summary, the main disadvantage of the leader-follower approach is its limited robustness to MAV failures in comparison with the fully decentralized swarming systems described in section 3. Possibilities of increasing the robustness of such formation flying approaches are presented in Appendix B (paper [2]), where a fault-tolerant method for stabilization and navigation of heterogeneous formations is proposed. Article [2] is a journal extension of paper [20], which received the main best paper award at the largest conference exclusively dedicated to aerial vehicles. The fault detection and recovery mechanism introduced in [2] relies on identifying sensor and actuator faults, or even a failure of a team member. This information is used to adapt the group behaviour to maintain the system performance with limited capabilities. A scenario of multi-robot surveillance was used to present this ability in realistic missions, where a formation of autonomous vehicles has to drive repeatedly through a workspace in a phalanx to cover a large operating space. The use of heterogeneous teams of MAVs and UGVs allows us to consider their deployment in surveillance missions that cannot be carried out solely by teams of MAVs or UGVs, or in which these teams would not be efficient without their direct cooperation. MAVs can reach locations inaccessible by UGVs and may provide a top view survey of the scene as discussed in the previous section. On the contrary, UGVs may operate in workspaces constrained by obstacles (e.g., in abundant vegetation) and can carry a much heavier payload, which means that they can employ more powerful surveillance sensors. UGVs have a larger operational range, and they may even provide an additional power source for MAVs through a mobile heliport (see our work in [32]) and prolong their flight time.

The work presented in [2] and [20] extends the paper [1], discussed in section 2.1, in the following aspects. The primary extension consists in the fault diagnosis and recovery mechanism. Malfunctions of any follower are automatically identified by analysing the cost function values applied in the MPC control of the followers, and by comparing the progress of the cost function with the predicted cost values. If a deviation of a follower behaviour from the expected operational mode is detected, this follower is considered as a dynamic obstacle in the formation stabilization process. Another contribution of the work presented in [2] is a new obstacle avoidance function, which enables dynamic obstacle avoidance of external moving objects as well as inter-vehicle avoidance of faulty formation members. Finally, the formation stabilization approach was adapted to the use of a vision-based technique, which enables repeated formation navigation along a given path required by the surveillance task.

The formation driving method in [2] relies on a navigation approach called GeNav [63], which uses features detected in images that are gathered by a monocular camera carried by a UGV. This very simple method enables the group to be robustly navigated along a

pre-learned path consisting of a sequence of straight segments (for a proof of stability, see [63]). This combination of top-view relative localization and visual navigation provides a lightweight, low-cost, easy-to-deploy and efficient solution for simple MAVs with limited sensory equipment. The formation control approach adapted for stabilization along a required path follows our research on UGV formations driving along a spline reference path (see [31]). In comparison with the system discussed in [1], the global trajectory planning mechanism does not need to be integrated into the formation control in the surveillance task defined in this way, and a shorter planning horizon coupled with the desired path is preferred. The shorter horizon reduces the dimension of the optimization process and speeds up the control loop, which is essential for a fast response to dynamic obstacles and failures of team-mates, but it still allows us to prove convergence of the system into equilibrium and to specify requirements on system stability. For proof of convergence of the planning approach based on the Lyapunov theorem of stability, see our work in [31].

### 2.3 COMPACT FORMATIONS OF AERIAL UNMANNED VEHICLES

The first work addressing the problems of stabilization of compact teams of aerial vehicles with limited onboard sensing and processing capabilities is introduced in Appendix C (paper [3]). The deployment of the system, which includes MAV control, motion planning, stabilization, and trajectory planning, is discussed for three different realistic robotic scenarios. The primary objective of the paper is to determine the advantages and disadvantages of using onboard relative localization systems in various techniques of MAV-group control.

The first scenario is an extension of the MAV-UGV formation stabilization system introduced in the previous section. In this scenario, a formation of multiple MAVs follows a virtual leader into the desired target region in a complex 3D environment with obstacles (the movement of the MAV-UGV formation in [1] was constrained into a plane). Similarly, as in [1], stabilization of the MAV followers is integrated with an obstacle avoidance ability into a global trajectory planning mechanism that leads the entire formation into the desired goal area. This inclusion of global planning directly into the formation control mechanism is essential for ensuring the feasibility of obtained solutions of the formation motion planning task with respect to onboard relative visual localization. In the case of an MAV formation flying freely in a 3D space, the required direct visibility among team members may be ensured by analysing collisions between obstacles and a 2D convex hull of the positions of the followers, as proposed in [3]. The 2D convex hull, which represents the 3D formation, is obtained as a projection of the positions of the followers into a plane that is orthogonal to the trajectory of a virtual leader in its current position (see Fig. 4).

The second scenario in [3] is motivated by tasks connected with searching for extremes (locations in a 3D environment) of a measured physical value, e.g., locations with a minimum GSM signal in difficult-to-reach areas. Swarm intelligence is applied to speed up the process of searching for extremes in the measured intensity. In particular, a modification of the bio-inspired Fish Search School (FSS) technique [64] is used to define the motion of an MAV group based on the actual state of the particles (individual MAVs). Each particle in the FSS swarm defines its future movement based on its current state and the states of neighbours, obtained by onboard systems only. This approach is preferable to most of the other evolutionary approaches, such as Particle Swarm Optimization (PSO), where new desired positions of MAVs are determined based on the best-achieved position of a particle in the swarm so far (the global best) and the best-achieved positions of each particle (the personal best). This requires these locations in the environment to be remembered or denoted and to be localized relative to the actual positions of the team members. Using the FSS control rules, the proposed relative localization approach can be directly used for

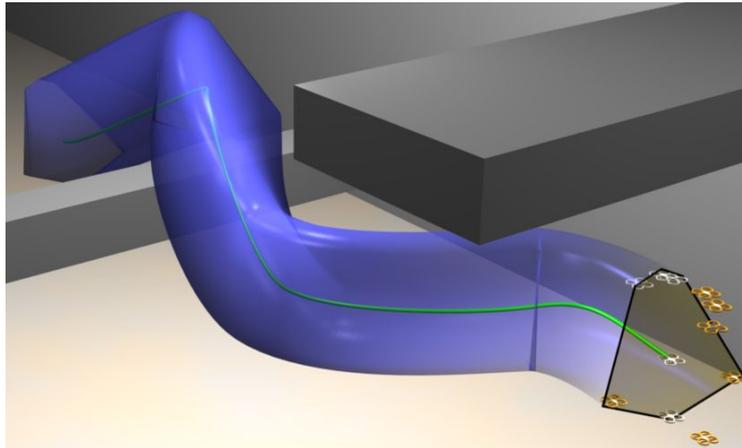


Figure 4: 3D visualization of the convex hull of a heterogeneous formation projected along the leader's trajectory planned through an environment with obstacles. Source: [3].

steering a compact aerial swarm if each MAV is considered as an FSS particle. In such tangible FSS method, information about the global position of MAVs in the environment is not necessary, as the group is steered by the distribution of the measured signal intensity only.

Finally, the third scenario demonstrates the deployment of the proposed system in a cooperative surveillance task (see [18] for implementation details of this method). This application is used as an example of robotic problems, where the MAV trajectories have to be purposely computed prior to the mission to enable verification of their safeness and feasibility by a human supervisor. In the scenario, a set of goals (areas) is assigned to a limited number of MAVs with the aim to find a static swarm configuration that can guard the areas, and to maximize the information that is acquired there. Similarly, as in all of the previous cases, the MAV swarm has to respect the motion, localization and sensing constraints that have to be satisfied in the final static swarm distribution as well as during the swarm deployment to these locations.

To sum up, the proposed methodology is designed for finding a feasible static shape of the swarm and a feasible plan of motion from the initial configuration to this target shape, i.e., locations of particular swarm entities optimized to solve the surveillance task and trajectories for all MAVs to enable the swarm to be deployed to the obtained locations. To ensure collision-free motion and to satisfy the localization constraints, a motion planning technique needs to be integrated directly into the core of a multi-objective optimization engine. In [3], this problem is solved by employing particle swarm optimization (PSO), in which each PSO particle represents the entire MAV swarm. This means that all MAV positions are encoded into a single optimization vector. Since the motion, localization and obstacle-avoidance constraints can be checked in each PSO step, the history of the evolution of the best particle (the obtained solution of the optimization problem) can be directly used as a feasible motion plan for all MAVs to solve the surveillance mission.

Let us compare these three different approaches regarding the performance of the onboard localization system. The FSS-based approach is very robust to an inaccuracy in the measurement of relative distances, but it is sensitive to a dropout of the localization system, which is used directly in the FSS swarming rules. Conversely, the formation flying and swarm deployment approaches are robust to a dropout of the localization method, and are sensitive to an inaccuracy in the measurements of the relative positions. In these scenarios, the onboard visual relative localization method is used to unify the local refer-

ence frames of the MAVs. In case of a temporary dropout of the relative localization, the MAVs can safely continue in their mission based on visual odometry. However, if the precision of the relative localization is low, the performance of formation flying and swarm deployment may be even worse than when only the odometry outputs of particular MAVs are used.

As mentioned in the introduction, the mutual localization information provided by the onboard sensory system significantly reduces the communication traffic necessary for coordinating MAVs. From this point of view, the most efficient approach is the FSS-based method, where almost no communication is required (the advantages of bio-inspired swarming approaches are discussed in greater detail in section 3.1). In the surveillance scenario, a communication channel is required at the beginning of the mission, where plans for all MAVs are distributed within the team, and then group coordination relies on the mutual localization method only. On the contrary, in the leader-follower approach, the required trajectories in a short-term control horizon have to be sent frequently to all followers, which requires a constant flow of data between the robots in the formation.

#### 2.4 COMPLEX MANOEVRES OF COMPACT MAV FORMATIONS

The work in Appendix D (paper [4]) is aimed at exploiting one of the main advantages of MAVs, which is possible deployment in a cluttered workspace, flying close to obstacles, and achieving inter-vehicle coordination of MAVs in small relative distances. In [4], a novel concept of motion planning and stabilization of MAV formations in cluttered GPS-denied environments based on a dynamic virtual leader was designed and experimentally evaluated.



Figure 5: An MAV formation following a migrating virtual leader through a corridor.

The proposed method enables the autonomous design of complex manoeuvres of a compact MAV team in a virtual-leader-follower scheme by migrating a virtual leader along a hull surrounding the formation. This enables a sudden change in formation motion in all directions, independent of the actual orientation of the formation. The

required high manoeuvrability is again achieved by including model predictive control (MPC) and planning the trajectories that are feasible for MAV formations into a single optimization process. This extension of the classical leader-follower concept enables us to rely on the well-conducted theory of the leader-follower control technique and to guarantee the stability of the group. For experimental verification of this approach, see video <https://youtu.be/slz1Htve3kY> and snapshots in Fig. 5.



---

## SWARMS OF CLOSELY STABILIZED MICRO AERIAL VEHICLES

---

The motivation for the research on swarms of MAVs presented in this section is to find a methodology for stabilization of large groups of very simple and small-size aerial robots in real-world conditions, and to investigate the minimal sensory, communication and computational requirements for safe flocking. A practical inspiration for conducting this basic research on swarming principles is the need for systems that enable robust cooperative flying in the event of a complete dropout of communication among the vehicles within the group and external global positioning systems (e.g. GNSS). This system enables us to realize a safety mechanism for more complex systems to overcome temporarily sensory dropouts, while still retaining cohesion of the group, and to continue in an autonomous multi-robot mission with limited performance. Swarming behaviour independent of external localization systems and communication availability is also desirable for cooperative flying in demanding environments with obstacles, and in scenarios where a transition between a GNSS environment and a GNSS-denied environment is required. In these situations, robots cannot rely on GNSS and, in addition, communication malfunction is expected to occur mainly in the case of large groups. Moreover, a communication channel and a global localization system cannot be applied in numerous industrial, security, and defence applications.

### 3.1 BIO-INSPIRED BOIDS SWARMING MODEL

An algorithm for stabilization of swarms of Micro Aerial Vehicles (MAVs) in a compact shape and for their navigation through a complex environment with obstacles is presented in Appendix E (paper [5]). The proposed method relies strictly on onboard sensors, with no need for global localization and communication, which enables the multi-MAV system to be used independent of preinstalled infrastructures. The major source of information applied for group stabilization provides visual localization modules carried by all MAVs to estimate the relative positions of neighbours in the swarm (see section 4 for details on relative localization systems). The guess about the positions of neighbouring MAVs and the information on the relative positions of the obstacles are integrated into group stabilization rules via the bio-inspired Reynold's boids model of swarm behaviour [65]. This approach requires only limited information on the local proximity of each swarm particle for designing a control law in a distributed way. An important advantage of this method is the possibility to stabilize a swarm without sharing global knowledge between the particles. Only the start command needs to be distributed within the group, and this can be realized indirectly using, for example, a start sign detected by the vision modules if no communication infrastructure is available.

The relative localization system uses monocular cameras carried by all MAVs and (possibly) only simple localization patterns attached to all MAVs for a more precise and robust estimate of their mutual positions. See section 4 for various approaches to mutual localization with various levels of precision, reliability, and requirements on localization patterns and onboard HW. The new usage of onboard relative localization allows us to gain in-

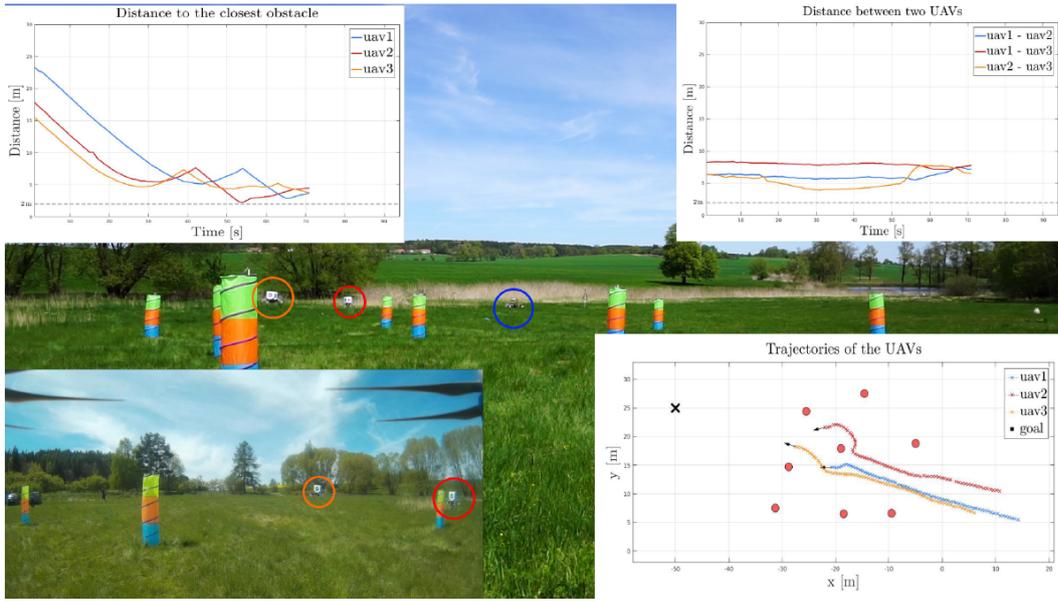


Figure 6: An MAV swarm stabilized via the boids model in an “artificial” forest. Control and motion planning are realized onboard the MAVs in a decentralized way.

formation on the proximity of each MAV, similarly as it is done in swarms of animals in nature. The relative localization sensors have characteristics similar to those of the sense organs of birds and fish. In both cases, animals may observe only neighbours in swarms under a limited viewing angle, the information acquired on a relative position of these neighbours is quite precise, and only a rough guess about their motion prediction is available. Nature-inspired control principles could be an ideal starting point for swarming research and they have also provided promising results, despite their simplicity, in our real-world experimental tests (see Fig. 6 and video <https://youtu.be/roX7bqRozNM> for swarming in an environment with obstacles).

Our research endeavor presented in papers [5] and [16] aims to satisfy the requirements for swarms, as listed in [66]: scalability for large groups, high redundancy and fault tolerance, usability in tasks not solvable by a single robot, and locally limited sensing and communication abilities. This definition specifies some research constraints, and enables a rigid analysis of swarming properties. It also perfectly meets the requirements for real applications, where the swarm technology could be efficiently applied. Although several research teams investigated the swarming abilities for ground robots in a planar environment [52, 67, 68, 69, 70, 71, 72], and some teams have even used the Reynold’s boids model for swarm control [73, 74, 75], most of these systems were designed for use in laboratory conditions, or even for use of simulated robots often considered as dimensionless points only. These approaches therefore frequently ignore constraints encountered in the real deployment of swarms, and they cannot be applied in real-world applications outside the laboratory.

Our experience has demonstrated that motion, stabilization, communication and sensing constraints have to be included directly into the swarming model in order to achieve reliable behaviour. Although our aim is to design a simple light-weight solution, a mechanism fusing data from multiple available sensors is a key technique that needs to be involved to provide state estimation of single entities as well as the entire group. Mainly because simple, light and cheap sensors are used, perfectly reliable behaviour for all critical control inputs cannot be expected and redundancy needs to be available. In the case of swarming, redundancy can be considered not only on an individual level,

but we can also take advantage of group behaviour, and we can fuse information from multiple entities together. Merging information from multiple decentralized units is a challenging task if a direct communication channel cannot be used, and if an indirect information exchange via observation of behaviour of neighbours is required (see video <https://youtu.be/RY0Muk05Yt0> and Fig. 7, for an experimental deployment of the swarming approach without direct communication). To be more specific, we have for example theoretically analysed and experimentally verified that a short-term drop-out of measurements of speed relative to the ground based on optical flow, which is a crucial control input for single vehicle stabilization, can be compensated through observation of neighbours if their speed measurement sensors work properly. For details on using optical flow information in low-level control, see [33, 34].



Figure 7: An MAV swarm stabilized via the boids model using a UVDAR (Ultraviolet Direction and Ranging) localization system (described in section 4.2) without any explicit communication between the robots (see <https://youtu.be/RY0Muk05Yt0> for a video of the experiment).

Another interesting way to use the results of observations of neighbouring entities in the swarm is estimating the state of their behaviour, which is important for dynamic swarming if no communication channels are available. Our theoretical studies and numerical simulation have shown that various behavioural patterns may be propagated through the group without direct communication. This enables us to implement a low-level escape behaviour functionality [76]. Similarly as in nature, once an individual entity identifies a threat (a predator in the case of fish/birds, and a moving object/person in the case of MAVs) that evokes an avoidance manoeuvre, other teammates can identify this avoidance pattern. In these individuals, the avoidance manoeuvre can be activated without seeing the potential danger. See examples of the propagation of such an escape behaviour state through a group of MAVs using onboard sensing in a video <https://youtu.be/Md5eV823I4o>.

Similarly, start and end mission commands can be propagated through the group by influencing only a limited number of entities, which are then followed by other group members using mutual localization information only. This approach enables large swarms to be commanded without a direct communication channel from an operation station to MAVs and between the MAVs.

### 3.2 COHERENT SWARMING WITH MINIMUM SENSORY REQUIREMENTS

In this research stream, we go even further in our objective to minimize the sensory information required for group stabilization. The algorithm proposed in Appendix F (paper [6]) enables a compact group to be stabilized and navigated using even less data than is required for the boids model. Methods based on the boids model involve complete information (bearing and distance) about all neighbours in a 360-degree view, which needs at least three cameras of the mutual localization system to be used onboard each vehicle (see section 4 for details on the vision-based localization system). Using the method proposed in [6], swarm coherence can be achieved with a single camera and using only binary information on distances to neighbours. Our theoretical analyses and practical tests verified that a single camera with a 120-degree field of view is sufficient for group stabilization. Moreover, no exact information on mutual distances to the detected neighbours is required by the system in [6], and only a decision whenever the localized object (another teammate or an obstacle) is closer to or further than a given threshold is used. To provide this sensory information, we purposely designed a very robust smart sensor using a single onboard camera and UV LED lights. This could be an enabling technology for stabilization of large swarms (see section 4.2 for details).

This information is applied for controlling the transitions between three behavioural states, *forward* (the default setting for all MAVs), *coherence*, and *avoidance*, which are the core elements of the swarming model proposed for MAV-group stabilization in [6]. MAVs in the *forward state* continue flying towards mission objectives as long as they are forced to move to another state, using the results of the onboard mutual localization system. Each MAV stays in the *forward state* as long as at least  $\alpha$  neighbours are kept within a given localization distance. A proper value of parameter  $\alpha$ , which is the required number of localized neighbouring agents, depends on the required density of the swarm and sensory model of the localization device. If less than  $\alpha$  neighbours remain within the required distance, the MAV enters into the *coherence state*, in which it turns back to restore the lost connection. Once the required number of neighbours appears back in the localization range, the particular MAV transfers back into the *forward state* and may continue in the mission. A proper setting of  $\alpha$  is important mainly for large swarms, which we are focused on, where it is not possible to keep the localization constraints between all pairs of MAVs. Accurately defined  $\alpha$  enables a broader and better-structured swarm, where (ideally) each robot has precisely  $\alpha$  neighbours and the swarm forms a stable regular net, which is not precomputed and arises autonomously.

If an MAV gets closer to another MAV/other MAVs or an obstacle than a given threshold, an evasive manoeuvre has to be performed, and the MAV enters into the *avoidance state*. In the *avoidance state*, a simple strategy for finding a new safe motion direction is applied using the immediate knowledge of all objects that are closer than the avoidance distance. This simple reactive approach, if performed by all teammates simultaneously in a strictly distributed way, provides safe and coherent flocking even in a complex environment with obstacles (see the results of complex simulations in [6] and in a video [https://youtu.be/MINM-dB\\_Ku4](https://youtu.be/MINM-dB_Ku4)). Based on theoretical and experimental analyses, we have found that the group coherence even increases with the number of swarm members, and the requirements on the mutual localization system are decreased. This observation

is a fundamental and promising result towards designing a large compact group of cooperating light and simple MAVs.

The main contribution of this approach is the possibility to stabilize an MAV swarm in environments with obstacles, without the need for global positioning systems and explicit wireless communication, and with significantly limited computational and sensory requirements. Using this method in a comprehensive analysis of the achieved swarming behaviour, we have been able to confirm/disprove a set of hypotheses assumed in the swarm robotics literature or being compiled on the basis of our experience with the boids model used for MAV swarm control (for the set of investigated hypotheses, see [6]). We have shown that using even such limited sensory data the swarm intelligence is beneficial for keeping coherence, decreasing the number of collisions and increasing the reliability of the system. Again, the crucial part of this approach was the integration of motion and localization constraints directly into the swarming rules. Statistical tests in various robotic simulators and environments were carried out to analyse the algorithm performance in its different configurations and to verify its reliability, taking into account the limitations of the designed system of mutual localization of swarm members.



---

## MUTUAL ONBOARD LOCALIZATION OF MULTIPLE COOPERATING ROBOTS

---

Systems of multiple closely cooperating aerial robots require numerous challenging techniques to be solved, including precise stabilization control, motion planning, coordination, sensor fusion, obstacle avoidance, in some cases communication, and always precise mutual localization. Although, all these challenges had to be tackled and solved during our long-term research towards fully autonomous MAV swarms, mainly the mutual onboard localization of neighbouring entities is the most relevant tool for stabilization of compact MAV groups in real-world scenarios, where GNSS information is not sufficiently precise and motion capture systems are not available. A single chapter is therefore dedicated to this research stream in this manuscript.

### 4.1 LOCALIZATION PATTERNS DETECTABLE IN THE VISIBLE SPECTRUM

Our preliminary attempts to design a system that can estimate the states of neighbouring robots (aerial and ground) relatively to a robot carrying the localization sensors (relative distance, bearing, orientation) were focused on using color detection patterns and cameras. This approach can solve the relative localization problem using the limited computational power of onboard microprocessors in real time, i.e., with minimal latency and more than 30 times per second, which is a common requirement for MAV controllers. Although we have been able to stabilize compact MAV swarms and heterogeneous MAV-UGV formations (see Fig. 8 and [15, 19, 20, 21] for examples), our experience has shown that such a vision system is highly dependent on the light conditions.

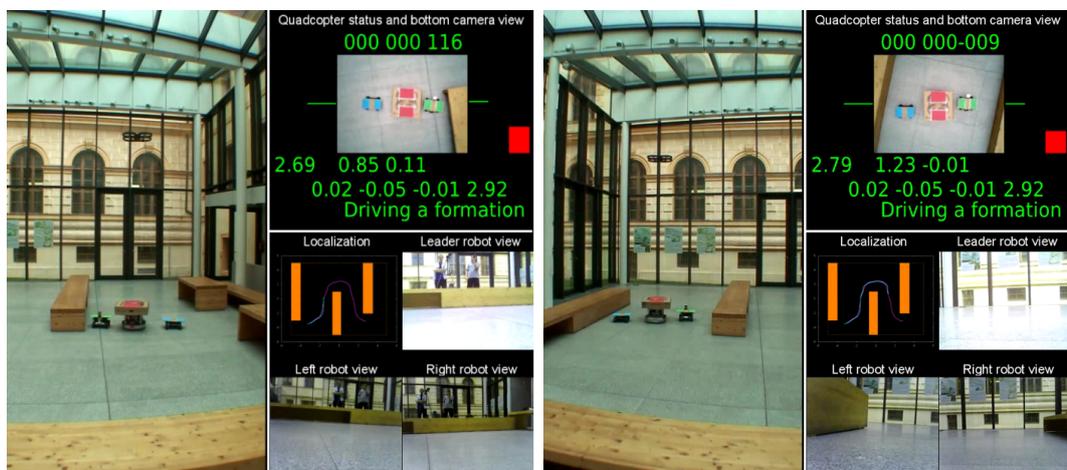


Figure 8: Snapshots from a formation driving experiment with 3 UGVs and 1 MAV using color detection patterns. Source: [1].



Figure 9: Examples of the deployment of the circle pattern detector published in [12] and [7]. This picture was composed with the help of Tomas Krajnik, the main author of the circle detector [7], and other co-authors.

The color map used for segmenting the image pixels and for identifying the color blobs adherent to parts of the localization pattern had to be re-calibrated after any change in intensity or in the source of illumination. Often, several different color maps had to be used in different areas of one room, which limited the usability of the system to simple laboratory experiments. This did not satisfy our primary intention to take MAV swarms out of the laboratory and into real application scenarios.

Significant progress towards this long-term objective was achieved by employing black and white patterns of known shape and dimensions. In particular, circular planar ring patterns (roundels) were used. This approach, presented in Appendix G (paper [7]), is robust to variable lighting conditions, achieves sub-pixel precision, and its computational complexity is independent of the size of the processed image. Such a setup achieves precision in the order of millimetres for target distances in the order of meters, and can track hundreds of targets at camera frame-rate onboard computationally restricted platforms. The fast update rate allows the localization system to be employed directly in the feedback loop of MAVs in swarms and formations. An important outcome of our research in [7] is a model of the localization sensor arising from theoretical analyses of the vision system and an experimental evaluation of the system performance in real scenarios. As was mentioned in previous sections, this model needs to be integrated into the motion constraints of MAVs acting together in a group to achieve the required system stability.

Algorithm [7] outperforms other pattern detectors [77, 78, 79, 80, 81] in terms of speed by an order of magnitude, while achieving similar precision and robustness. These properties are achieved by searching images for circular black and white objects using a combination of flood-fill techniques, on-demand thresholding, and on-the-fly statistics calculation.

The statistical information gathered on-the-fly is used to test whether the continuous areas of pixels are likely to represent the searched pattern, and quickly reject false candidates. The main advantage of the method is that it can be initiated from any position in the image without a performance penalty, which facilitates the implementation of pattern tracking. In a typical situation, the algorithm processes only the area that is occupied by the pattern itself, which results in a significant performance boost.

This approach goes beyond usual computer vision techniques, which require high computational time and rarely enable a frame rate higher than one frame per second with onboard processors, although they present incredible performance and reliability in object detection even without a need for purposely placed localization patterns. The work in [7] was therefore considered as one of the first techniques enabling onboard mutual localization of MAVs in the robotic community, and its authors have counted more than 20 examples of its real deployment (see Fig. 9 for examples) by leading robotic groups, including the University of Pennsylvania, the University of Birmingham, Technical University Vienna, the University of Lincoln, the University of Buenos Aires, etc.

#### 4.2 ULTRAVIOLET-BASED RELATIVE LOCALIZATION

Another important step towards reliable mutual onboard localization completely independent of light conditions and the type of work-space is the research of using onboard ultraviolet-based markers and cameras. This approach exploits the evidence that the spectrum of sunlight, which is the most significant outdoor illumination source, shows a marked decrease in intensity in near ultraviolet wavelengths in comparison with visible light. The decrease in intensity is faster near the visible range than on the infrared side. The environment at daylight is therefore dark in the ultraviolet range, as the appearance of UV light is limited to reflected and refracted sunlight only. Artificial radiation sources with strong UV emissions are rare, making this wavelength range very attractive for mutual localization with active markers.

Localization markers composed from artificial UV light sources (UV LEDs) can be observed through optical band-pass filters, where particular LEDs are simply detected as isolated bright spots on a dark background. Using the filter significantly reduces the complexity of markers detection by removing additional information in images. In the algorithm presented in Appendix H (paper [8]), positions of bright spots that represent the markers are found by image thresholding and consequent computationally efficient FAST-like surroundings comparison (see video <https://youtu.be/rY7bPYMAWko> for neighbouring MAV localization from an onboard camera in demanding outdoor conditions).

Two possible methods for relative position estimation are proposed in [35], where the system we call UVDAR (Ultraviolet Direction and Ranging) is described (in [8], UVDAR was used for formation flying). In the first setup, multiple UV LED lights are used to create a localization marker ( see Fig. 10). To increase the precision of the distance estimation by maximizing the pixel distance of the bright spots in the image, the UV LEDs are positioned on the ends of each arm of a multicopter-type MAV. For analyses of various configurations chosen according to the specific application requirements, see [13]. The mutual position between two MAVs (the bearing towards the detected MAV and the relative distance) is calculated using the known geometrical layout of the markers in this approach. In the second approach, only a single MAV is used for the mutual localization. In this case, the process of estimating the relative distance to the neighbouring robot exploits the fact that the light from UV LEDs affects multiple pixels in the image if the detected MAV is close to the MAV that is carrying a camera. With increasing distance, the area of pixels with non-zero measured intensity decreases, and it becomes a single pixel if a particular distance between the camera and the light source is reached. The correlation between the size of the blob and the distance is not reliable, and depends significantly on multiple

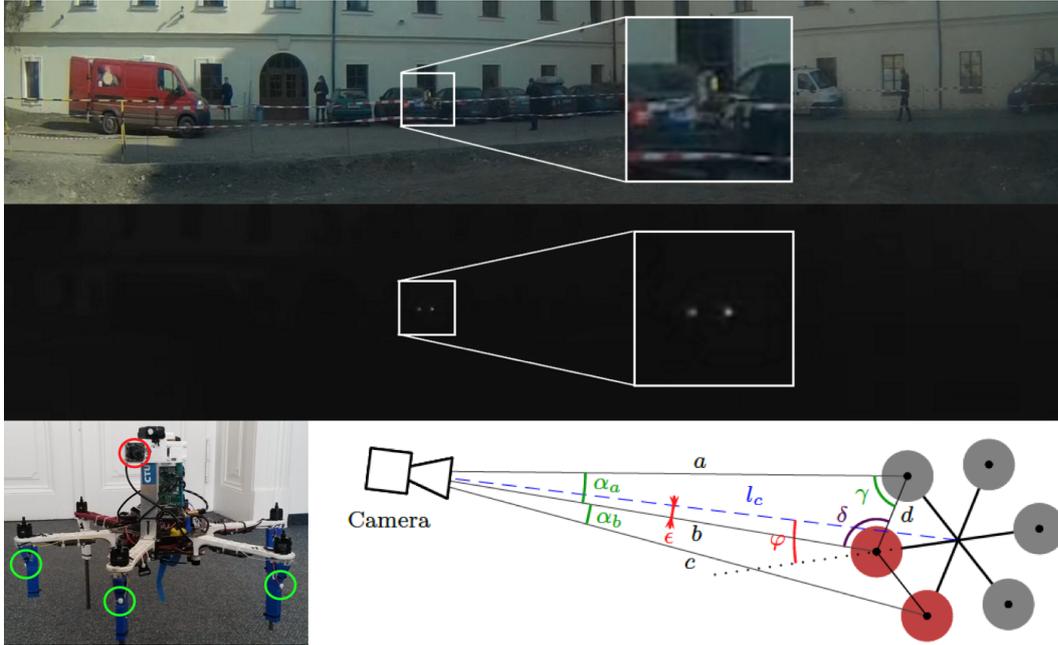


Figure 10: A demonstration of the UV-based localization system. Above: A drone flying in front of cars is very difficult to recognize in the visible spectrum, while it can be easily recognized in UV. Below left: An MAV equipped with an onboard camera with a filter made by a red circle and UV LEDs in green. Below right: The principle of distance measurements from multiple detected UV lights. For more details, see [8], [35] and [13].

factors, such as the light conditions, the relative orientation of the LED and the camera, the setting of the camera and the lens, etc. Nevertheless, the transition between sensing a blob composed of multiple pixels and detecting a single pixel is usually sharp, and indicates precisely that the neighbouring MAV is closer than the threshold. As described in section 3.2, such binary information can be sufficient for safe and reliable swarming, and it can be obtained with the use of minimum computational resources and equipment.

As has been mentioned, the only source of UV light in nature is the sun, which can be simply excluded by using its known size in the image and also its position relative to the MAV, if a compass measurement is available. Nevertheless, specular reflections of direct sunlight can occur on rare occasions, and can be identified as false positive detections. We have experienced disturbing reflections from car windows, for example. Therefore, additional information has been encoded in the markers by modulating the LEDs brightness by blinking to exclude false positive detections, and an algorithm for retrieving such information from a set of consecutive images was developed. The algorithm (described in [35]) is based on a special implementation of the 3D Time-Position Hough transform for tracking periodically appearing objects in time.

The frequency of the blinking signal retrieved from the markers may be used as an identifier for the given marker to be able to distinguish among multiple MAVs, if required. If multiple blinking frequencies are used for LEDs in a single pattern, unambiguity of relative orientation can be achieved. Another advantage of the UVDAR approach is that the images need to be obtained with a short exposure time to maintain a high contrast between the markers and the ambient illumination. This enables us to achieve a high frame-rate, which is required for stabilizing dynamic systems, such as the MAV formations [14, 15] and swarms [16, 18, 19]). See videos <https://youtu.be/RyOMuk05Yt0> and

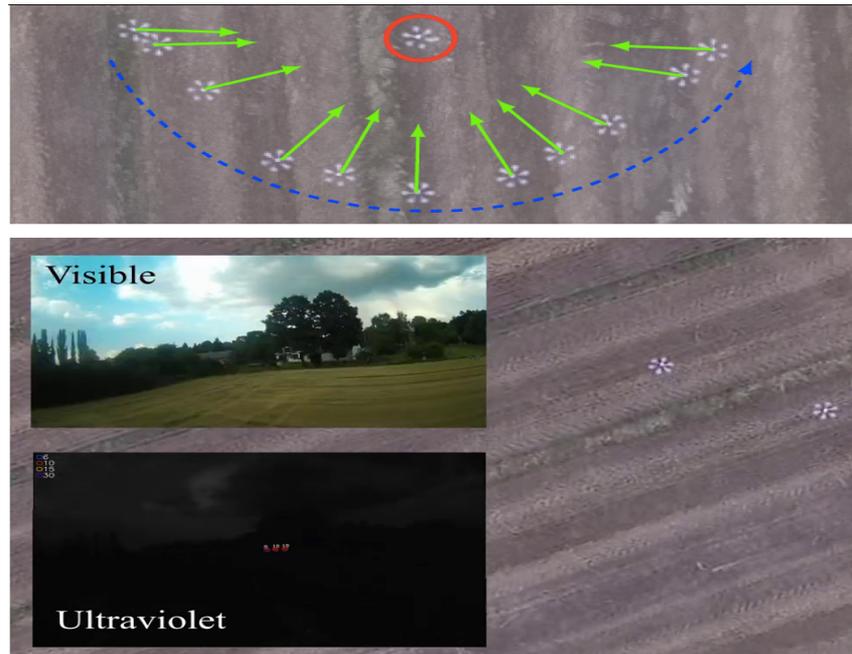


Figure 11: A simple leader-follower experiment showing the performance of the UVDAR relative localization system used in MAV control feedback in demanding outdoor conditions without any explicit communication between the robots. The neighbouring MAV is robustly localized in various light conditions and against a changing background (see [https://youtu.be/ixNe0ye\\_02I](https://youtu.be/ixNe0ye_02I) for a video of the experiment). Source: [8].

[https://youtu.be/ixNe0ye\\_02I](https://youtu.be/ixNe0ye_02I) (with snapshots in Fig. 7 and Fig. 11, respectively) for the latest experimental results.

#### 4.3 LOCALIZATION WITHOUT PATTERNS PURPOSELY PLACED ON VEHICLES THAT ARE BEING LOCALIZED

Finally, let us mention our ongoing research on fast onboard relative localization without any purposely placed markers. Although the UV LED lights used in the system [8] are lightweight and have low power consumption, the MAV body needs to be modified, e.g. by installing wiring, by attaching LED mounts, and by adding a board in order to control the required blinking frequency. However, the more dominant motivation for the research presented in this section is the requirement for relative localization of objects/flying vehicles on which a compatible marker cannot be installed. An application for air protection against unauthorized drones can be mentioned as an example. For our preliminary results in this scenario, and for results related to the first challenge of the MBZIRC 2020 competition <sup>1</sup>, which was also motivated by this task, and in which our team participates, see Fig. 12 and videos <https://youtu.be/Z4lSam-1pN8>, <https://youtu.be/RTzac8PLpkY>, <https://youtu.be/D1lygdYu0m0>.

In the first approach that we are investigating, a Convolutional Neural Network (CNN) is employed for relative localization of objects from images gathered by an onboard camera. Using CNN YOLO object detection [82, 83], our system can keep to the 30Hz limit for image processing with an onboard MAV computer. Although this approach brings promising results and can be applied for most swarming and formation flying scenarios,

<sup>1</sup> <https://www.mbzirc.com/challenge/2020>

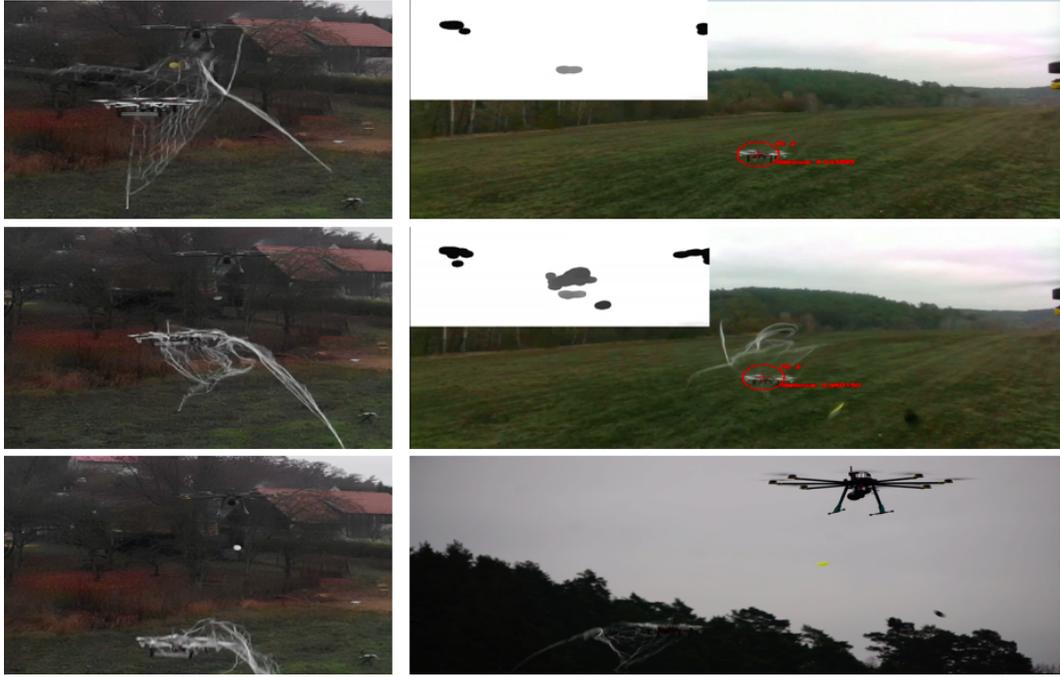


Figure 12: Demonstration of hunting for an unauthorized drone with the use of a net. The target drone is detected in real-time using an onboard 3D camera. Examples of pictures captured by the 3D camera are shown in the upper left corners of images captured from an onboard RGB camera.

its reliability may depend on the working space. The background of images strongly influences the frequency of false positive detections (see examples of MAV detection with various image backgrounds in Fig. 13). Moreover, the localization range is limited, as the CNN detector requires a sufficiently large image of detected objects in the pictures. Although the bearing information is sufficiently precise with this method, the distance estimation of the identified objects (a bounding box of the detected object is used) requires a priori knowledge of the size of the localized object. For the MAV detection task, a general image of the drone can be learned using a set of pictures of different drones in various situations, but various drones with similar shapes can have different dimensions.

A more robust and precise solution provides processing data from onboard 3D cameras (see examples of an MAV detected in an image from an Intel Realsense Depth Camera in Fig. 13). This approach to MAV localization is very reliable, mainly in an open space, which is a situation often encountered in scenarios of protection against unauthorized drones. Then, the flying object can easily be recognized from a distant background. Thermal cameras can provide another interesting source of sensory information, which enables to detect motors and onboard computers as a significant pattern in the case of multi-rotor MAVs (see Fig. 14). Thermal cameras are valuable mainly when there is limited visibility. Finally, we have tested radar-based methods for drone detection, which theoretically promise a long range and high reliability. However, have not yet achieved successful results with these methods.

To conclude this section, let us compare the available methods from the application point of view. The method with the black and white circular pattern is efficient for simple proof of concepts, laboratory experiments, and deployment in a stable environment, such as an office space, where the active markers, such as the UV lights, could distract working people. One of the main advantages of using circular patterns is straightforward deploya-



Figure 13: Examples of pictures obtained by onboard cameras for visual drone detection. Left: images taken by an RGB camera. Right: images taken by Intel RealSense Depth Camera. The red blobs represent the detected drone, while the blue blobs are caused by propellers and are not considered in the object detection algorithm.



Figure 14: Examples of pictures obtained by various onboard cameras for visual drone detection. Left: RGB camera. Middle: 3D camera. Right: Thermal camera. Based on different background and distance to the detected object, different cameras provide reliable results, and therefore a fusion of multiple sensors is required.

bility, as it is only necessary to print the pattern, to use any camera attached to a PC, and to download and run the open-source software that is provided.

The approach on the basis of UV lights is an efficient tool for multi-robot applications irrespective of environmental and lighting conditions. It provides the best performance and reliability in nature, urban, indoor, and outdoor workspaces, at night as well as in direct sunlight. The main limitation is the requirement for specialized hardware. A camera with a filter has to be integrated onboard the controlled MAV platform, and UV LEDs need to be installed onboard the MAV that is to be localized.

The vision-based CNN approach may be an exciting technique enabling to employ tiny, lightweight and cheap MAV swarms. Although it is not mature enough to provide high reliability, initial experiments indicated a possibility of short-term stabilization of large groups, since some swarming techniques can rely on unreliable sensory information taking advantage of shared group knowledge (see section 3.2 for details).

To achieve robust perception in the application of protection against unauthorized drones, a fusion of multiple sources of information seems to be necessary. A combination of the depth camera, the thermal camera and CNN-based computer vision provides high reliability in various light conditions, together with high precision. Cameras in the visible spectrum usually have higher resolution than depth cameras and thermal cameras, and therefore provide more precise information on a vector towards the identified object. However, the relative distance to the detected object is obtained most precisely using a depth camera, which does not require any a priori information on the object dimensions.

---

 APPLICATIONS OF THE PROPOSED CONCEPTS
 

---

Although motion planning, control, stabilization, and mutual relative localization of compact MAV teams acting in a cluttered GPS-denied environment are demanding scientific tasks with numerous basic research challenges, robotics is an application-motivated science, and the requirements for real deployment should be considered during the entire research pipeline. In our scientific endeavour, we are focused on designing principles and key technologies that enable us to tackle tasks that are difficult or even impossible to solve with a single-robot system (see Fig. 15 for an example). Only this kind of motivation can bring a sufficient added value to justify the usage of multi-robot systems with their high-complexity.

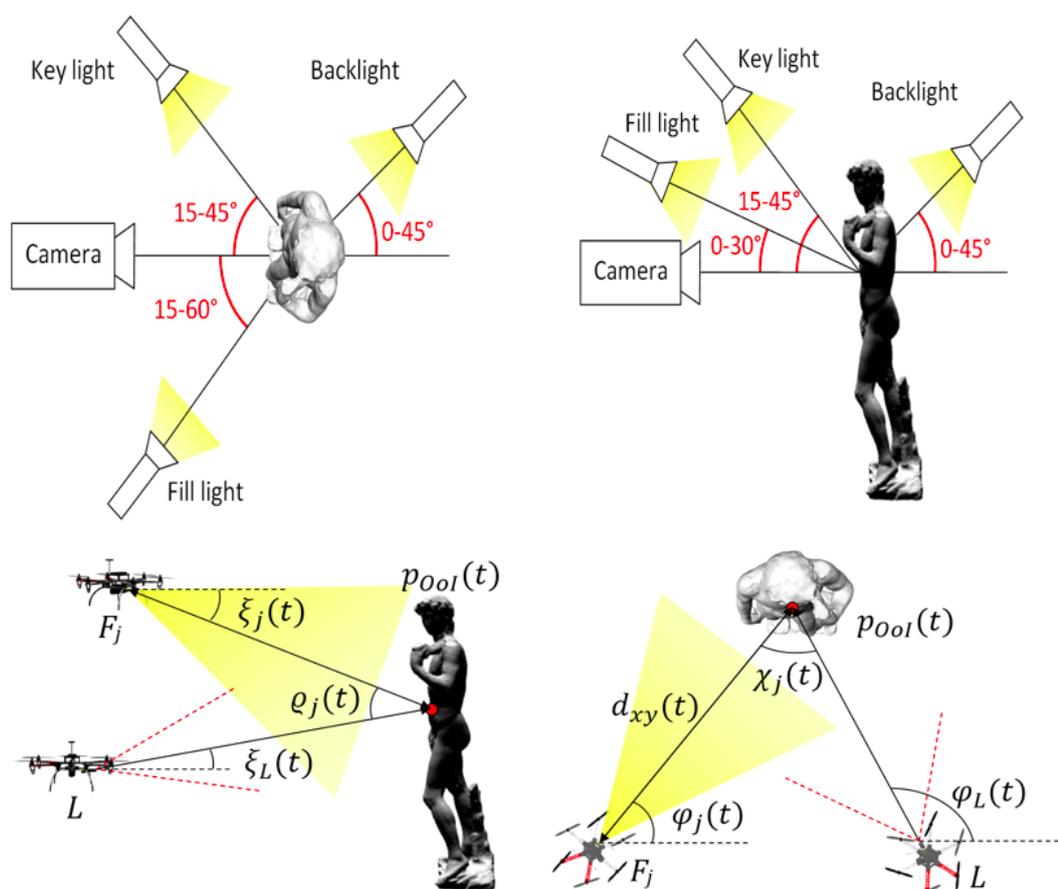


Figure 15: Illustration of the Three Points Lighting technique and its application by an MAV formation. Source: [9].

### 5.1 DOCUMENTATION OF DARK AREAS OF HISTORICAL BUILDINGS WITH THE USE OF MAV FORMATIONS

One of the most significant examples, in which considerable added value can be provided by applying MAV teams in a cluttered GPS-denied environment is in documenting hardly accessible areas of large historical buildings, where an MAV group is able to deal with limited light conditions (see Fig. 16 for motivation). In Appendix I (paper [9]), a method for filming and visual inspection in dark conditions by a self-stabilized formation of multi-rotor helicopters is employed. The setup, in which one of the MAVs carries a camera and neighbouring MAVs carry a source of light, aims to autonomously realize two techniques often used by historians and restorers for manual inspection of the interiors of historical monuments nowadays. The first technique, the so-called *Three Point Lighting* approach [84, 85], is a filming technique in which 1-3 light sources are used in different locations relative to the camera optical axis (see Fig. 15). The method enables to create an illusion of a three-dimensional object in a two-dimensional image and to illuminate the object being shot (such as sculptures in historical buildings), while controlling the shadows produced by the lighting. This property is essential for presenting the interiors of historical monuments to the broad public. This lighting technique removes the boring flatness from the images and videos, and it adds value to the consequent analysis of sensory data gathered with MAVs by historians.



Figure 16: The motivation for using MAVs in the inspection task in the interiors of historical objects. Left: Scaffolding in the church in Šternberk required for manual inspection. The church was closed for four months for services and tourists. Right: The same work in the same church conducted by our MAVs eight years later. The church was closed for four hours only.

The second technique frequently used by restorers employs a strong side-light for illuminating flat objects, such as walls with parget and mosaics. In this method, a strong light needs to be placed as close as possible to the scanned plane, which makes visible shadows in the image when the surface is rough. Restorers and conservationists can detect from such illuminated pictures if a tile in the mosaic is not fixed correctly, or if a painting is affected by a humidity, which is indicated by buckling of the wall surface.

None of these techniques can be realized using a single MAV only. The light source (or even sources) needs to be placed at a relatively long distance from the position of the camera. Illumination from the ground is not sufficient, as has been found out in numerous preliminary experiments in real historical buildings with the support of historians and restorers (see the outputs from these campaigns at <http://mrs.felk.cvut.cz/research/historical-monuments-documentation>). We implemented these two methods using formations of autonomous unmanned helicopters. MAV formations enable the use of these methods in locations in the interiors of large historical objects (e.g., churches)



Figure 17: The first (most probably worldwide) deployment of an autonomous formation of MAVs inside a historical building (church in Štenberk).

that are not accessible without installation some costly staging (see Fig. 16). In cooperation with restorers and filmmakers, we defined the *Three Point Lighting* and *Strong Side Lighting* techniques as multi-objective optimization problems in the MPC framework. The MPC approach is used for controlling all formation members, taking into account task objectives as well as the following constraints. The formation motion constraints ensure obstacle avoidance and mutual collision avoidance abilities. Low-level MAV stabilization constraints include motion and actuators constraints (see [34] for details on the control scheme and the constraints). In addition, constraints due to the limited camera field of view and the requirement for all MAVs to be kept out of the images are taken into account. Finally, the illumination constraints ensure that the intensity of the lighting is within the required range and that the allowed range of angle between the light and the camera axis is kept.

The proposed solution for this application arises from our long-term application work in historical buildings. More than 10 historical buildings with different structures and environmental properties were involved in this research, and 20 more objects are scheduled for the next four years. This application-oriented research involved the complete process from preliminary proofs of concept of manually-controlled MAV formations through semi-autonomous deployment into fully autonomous stabilization of the MAV team. In the semi-autonomous mode, the leading MAV with a camera is teleoperated, and the followers with lights are autonomously stabilized. For the first trials of autonomous formation stabilization in a church, see video [https://youtu.be/-sTUwzFf\\_Mk](https://youtu.be/-sTUwzFf_Mk) and pictures in Fig. 17. The video in <https://youtu.be/g1NuPnLCFTg> and the pictures in Fig. 18 demonstrate one of the highlight the deployment of the system in the Church of Saint Nicolas in the Old Town Square in Prague. Interesting practical results of the cooperative lighting approach were achieved in an abandoned church in Stará voda. Hidden graffitos of Soviet soldiers waiting for execution were studied on the basis of pictures obtained by the MAV team there (see video <https://youtu.be/yNc1WfebIag> and pictures in Fig. 19). During these experimental works, a new approach to simultaneous localization and mapping using laser scans from a simple onboard rotational laser scanner (LIDAR) was also successfully tested (see video <https://youtu.be/f0FAZYjWujI>).

The biggest and most important historical object to have been documented by our systems so far is the Church of Saint Maurice in Olomouc (see video [https://youtu.be/\\_cXa2yBLAeY](https://youtu.be/_cXa2yBLAeY)). Indoor formation flying, exterior mapping and 3D modelling were all real-

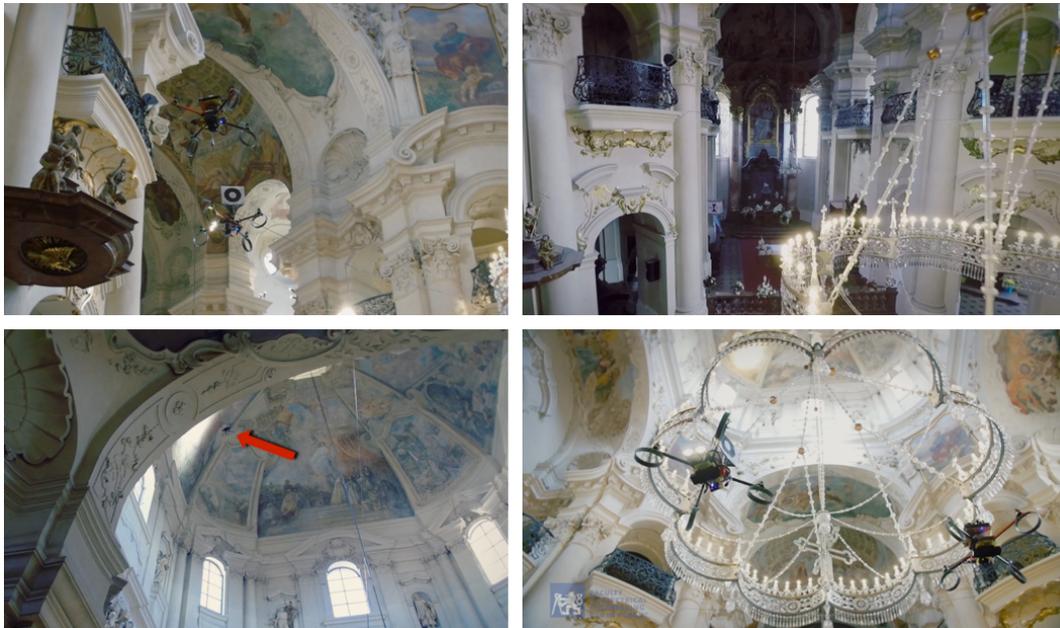


Figure 18: An example of the deployment of our system in the Church of Saint Nicolas in the Old Town Square in Prague.



Figure 19: Documentation and exploration works in a church in Stará voda village, which was a part of the Libavá military region and training area. A formation of cooperating MAVs provided a set of pictures of difficult-to-access parts of the church and statues damaged by Czechoslovakian soldiers (before 1968) and by Soviet soldiers (after 1968). The upper right picture of the graffiti was captured by a single MAV carrying both the camera and the light, while the bottom picture was taken by a pair of MAVs, one with a camera and the other with an external light.



Figure 20: 3D modelling of a facade of Plumlov Castle within a project Dronument <https://dronument.cz>. Above: RGB pictures. Bottom: 3D models with a texture gained from captured images.

ized there. A 3D model of a facade was also obtained in Plumlov Castle, using photogrammetry combining pictures from a large UAV georeferenced via the precise RTK GPS, while the task of a smaller MAV flying close to obstacles in a GPS shadow was to provide details (see video <https://youtu.be/TgFknWQyJnY> and pictures in Fig. 20). The last historical object that we would like to highlight in this list is a grotto in a park of Gorzanów Castle in Poland. An MAV was able to reach a not accessible chamber above the main cupola while carrying a rotating LIDAR and a camera, and helped to discover a unique artwork on a floor mosaic (see video <https://youtu.be/6mRYxciDLCM> and pictures in Fig. 21).

In addition to the smart lighting approach, numerous other techniques that would benefit from the use of cooperating flying robots are being used by restorers in historical buildings nowadays. Cooperation between two mobile units is required for roentgen analyses of the inner structure of statues located in difficult-to-access places. A mobile X-ray emitting device carried by one of the teammates needs to be synchronized with the movement of an X-ray sensitive layer, which is placed beyond the scanned object by another MAV.

Detection of infrared light getting through a painting is used for visualizing hidden layers on large free-hanging pictures that are difficult to remove from the churches. In this task, an IR camera (carried by one MAV in our case) has to be synchronized with a source of IR light placed beyond the picture (carried by another MAV). In both of these techniques, an interruption of direct visibility between the cooperating MAVs, and therefore a temporary malfunction of mutual localization, has to be taken into account in the formation stabilization system.

An even more challenging task, which requires direct cooperation of multiple MAVs, is the use of UV light for analyses of paintings. In this technique, an external UV light is used to excite photons from a UV sensitive painting material. The photons are detected by a distant camera. This powerful technique for non-invasive analyses of historical paintings can be realized only in completely dark conditions, which will bring us back into the ground research in this project. Flying in complete darkness is a challenging research



Figure 21: Documentation and exploration of a grotto in a park at Gorzanów Castle in Poland. Left and bottom right images show mosaics on a floor photographed by a down-looking camera, where pictures of animals were discovered. The upper right picture shows an MAV flying into a not accessible chamber above the main cupola, carrying a camera and a rotating LIDAR (see a scan from the LIDAR in the picture).

task, since most of the state-of-the-art techniques employed for MAV stabilization rely on vision in the visible spectrum.

## 5.2 AUTONOMOUS LANDING ON A MOVING PLATFORM

Another example of a direct application of the results of formation flying and multi-robot coordination research is the task of autonomous landing on a moving vehicle, which was motivated by the Mohamed Bin Zayed International Robotics Challenge<sup>1</sup> (MBZIRC) 2017, which is a competition organized by the Khalifa University of Science in Abu Dhabi. In this robotic scenario, the system for onboard mutual localization of moving objects in the proximity of a controlled robot can be employed with minor modifications only. The detected object state estimation method and the MAV state estimation method, which were required for stable MAV formation control, can also be applied together with the MPC technique in this task. Beyond the deployment of the research results presented in sections 2 and 4, some new scientific challenges appeared in this task due to the highly dynamic manoeuvres required mainly in the final phase of the landing. The control architecture proposed in Appendix J (paper [10]) is tailored specifically for the task of following fast dynamic objects with an MAV by leveraging the non-linear state controller in conjunction with a model predictive control tracker. Future predictions of the car movement are reshaped by the MPC tracker into a feasible state reference, which is reflected on the states of the real MAV by the non-linear state controller. The MPC tracker allows arbitrarily long reference trajectories to be tracked without solving an MPC task for the whole reference. This enables real-time and robust execution of the demanding task in real-world experimental conditions. The system is robust to very challenging outdoor conditions with a

<sup>1</sup> <https://www.mbzirc.com/>

wind speed of up to 10m/s, varying light conditions, and sand blowing in the air (the initial tests of the system were conducted in the desert in Abu Dhabi emirate). High repeatability in the landing task is a crucial element for most fully autonomous missions, e.g. periodic surveillance, reconnaissance, object carrying, and monitoring.

The system described in [10] presented the best reliability among all 142 registered teams in the MBZIRC 2017, and it achieved the fastest performance in the entire competition. Only our system was able to land three times in the competition in autonomous mode (see Fig. 22 for one of the successful attempts of our team). The fastest time of landing was achieved by the proposed system during the grand challenge, in which all MBZIRC challenges were solved simultaneously. The key component of the system that provided this high reliability and performance in comparison with the other teams and state-of-the-art works solving the landing task (see e.g. [86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96]) is the MAV state estimation approach integrated with a predictive trajectory tracking technique. This scheme enables to track and predict the estimated position of a moving target with the necessary precision and manoeuvrability to be able to follow the car even in turns of its path. Detected positions of the car are filtered using an Unscented Kalman Filter (UKF)-based technique with an assumed car-like model of the vehicle, while predictions of future car positions takes into account the known profile of the track.



Figure 22: A sequence of images from the second trial during the MBZIRC competition. The videos and additional material, including onboard footage, can be found at <http://mrs.felk.cvut.cz/jfr2018landing>. For more pictures of autonomous landing in various environmental conditions, see Appendix J (paper [10]).

Another important factor in the MBZIRC competition, which brings the designed system closer to real deployment, is the necessity to achieve mission objectives after only a few minutes of preparation, without an option to postpone the beginning of the time slot allocated for each team. According to standard practice in most laboratory experiments, no repeated tests were allowed and, moreover, the system robustness was exhibited in the current environmental conditions (light and windy), since the teams could not influence

the start of their trial. This even increases the demands on system robustness and immediate deployment without any preparation, which is important for further applicability of the designed solution.

Although the methodology was designed for the specific conditions of the first challenge of MBZIRC 2017, it can be used in various MAV systems to allow vertical take-off and landing on boats, trains or cars. This ability allows short-term flights from a mobile helipad located close to the area of the MAV deployment, which efficiently exploits the performance and the usability of MAVs. Combining them with a moving platform extends their operational range, which further increases the application potential of MAVs.

### 5.3 COOPERATIVE AUTONOMOUS SEARCH, GRASPING AND DELIVERING OF OBJECTS

The basic research achievements presented above have also been exploited in a "Treasure Hunt" scenario of cooperative autonomous search, grasping and delivering of objects, which was required in the third challenge in the MBZIRC 2017 competition. The cooperative autonomous search, grasping and delivering scenario that was introduced in this challenge, requires cooperation among multiple MAVs as well as interaction with the real environment. This is a challenging task significantly exceeding state-of-the-art robotic solutions. The task consists of several sub-tasks that were tackled in the swarm and formation flying research introduced above, such as cooperative motion planning, object detection, state estimation, MAV self-localization, precise motion control, trajectory tracking, and decentralized team coordination. New scientific challenges were discovered mainly in relation to aerial grasping and object dropping. Reliability and fast deployability is again a critical requirement, with even stronger impact due to the deployment of multiple robots. This motivated our intensive research in this area as well.

Multi-rotor MAVs are suitable for tasks where a physical interaction with objects and manipulation of objects are required, due to the ability of these MAVs to hover on the spot. Their usage in this field has already been investigated in several research works, mainly for a single MAV, focusing on control and object detection sub-tasks. For examples of control techniques, see [97], where a planner for high-speed aerial grasping is introduced, [98] using a monocular IR camera for autonomous grasping, [99, 100] for simultaneous control of the MAV and the manipulator joints, and the works [101, 102, 103] for a vision guidance approach using an image-based visual servo for an aerial manipulator.

A crucial scientific problem in most of these works is how to achieve flying stability when the mass-inertia parameters of the system are changed during the object manipulation. An interesting study on determining the stability bounds of a PID flight controller in object manipulation tasks can be found in [104]. Theoretical analyses and experimental tests indicated that the MPC mechanism that we introduced in [34] could be applied with just a few modifications for grasping metal objects to achieve the performance required for the treasure hunt scenario. The MPC-based approach allows integration of the MAV state estimate (including the external forces produced by the wind, the ground effect, and changing mass-inertia) and target state estimate (a position and velocity estimate of the currently observed object). The system enables our robots to reach the target with a maximum position error of 8cm, which is the threshold determined by the diameter of the object and the size of the gripper. For the initial tests on compensation of external forces within the MPC framework, see video <https://youtu.be/hv0ueLvYGSc>, which shows the response to changing a center of mass, and video <https://youtu.be/JvA62F71UXQ>, which shows compensation of wind disturbance.

Object detection and state estimation are investigated for example in [105], where online detection of a known object using features from images is described, and in [106], where an object extraction mechanism based on stereo vision for autonomous grasping of objects is presented. To detect the coloured objects in the MBZIRC challenge, we modified a

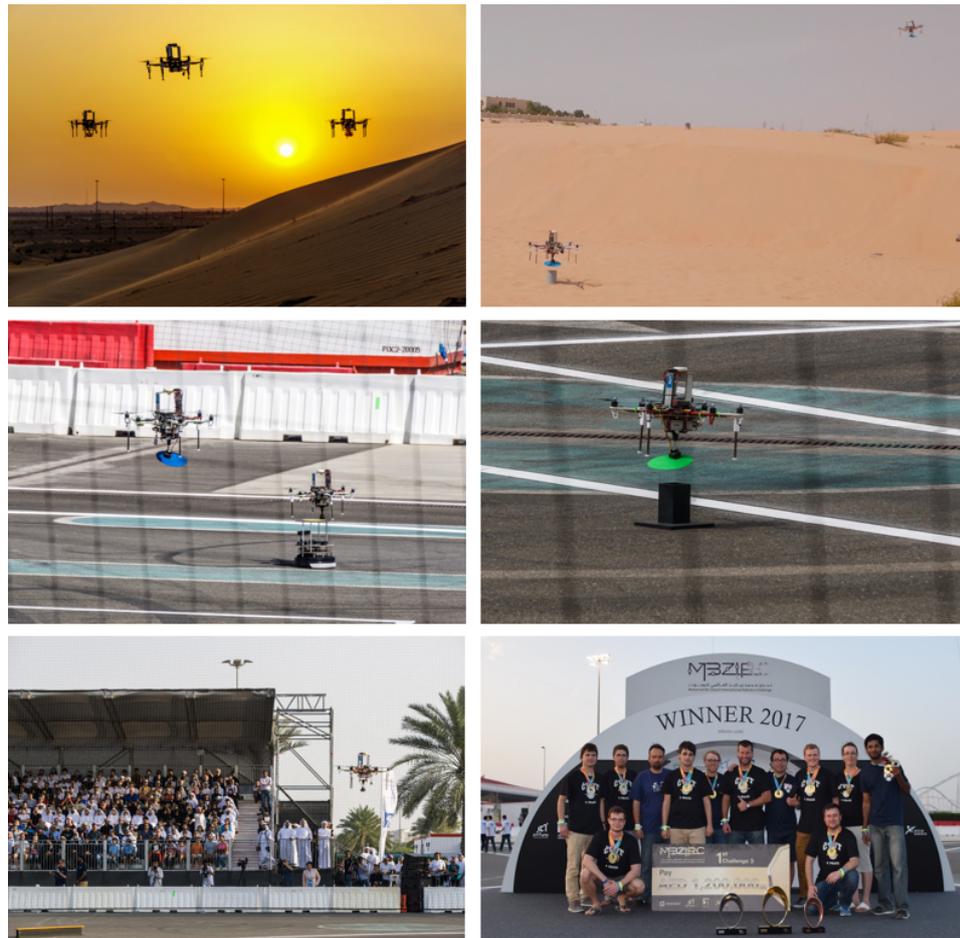


Figure 23: Pictures from the experiments on desert dunes in Abu Dhabi emirate and the third challenge in the MBZIRC competition. Videos and additional material from the tests in various conditions and the competition itself, including onboard footage, can be found at <http://mrs.felk.cvut.cz/jfr2018treasurehunt>. For more pictures, see Appendix K (paper [11]).

computationally efficient method that we had introduced in [7] for the mutual localization of swarm members, and which had already proved its reliability and accuracy in real-world conditions.

Preliminary tests of the system designed for the competition (see video reports [https://youtu.be/JYXeUrkn\\_cU](https://youtu.be/JYXeUrkn_cU) and <https://youtu.be/e1JEMmaZmDU>, which qualified our team for the MBZIRC 2017 finals) indicated that the most important element in this very complex task is the need to achieve high robustness and reliability in demanding outdoor deployment using real sensors and actuators with their uncertainties, possible failures, and sensory dropouts. Although the rules of the MBZIRC competition allowed the use of WiFi communication, GNSS (Global Navigation Satellite System), and the even more precise DGPS (Differential Global Positioning System), the availability of these systems was not guaranteed. For example, GNSS information was available only intermittently, due to interference with other transmitters located at the competition site and due to occlusion of the satellites by the surrounding buildings. In addition, the Wi-Fi infrastructure that was provided was very unreliable.

The low reliability of GNSS and communication motivated us to design and theoretically and experimentally evaluate a Failure recovery and Synchronization jobs Manager

(FSM), which leverages the combination of different modes of the system based on the availability of Wi-Fi, GNSS, and DGPS (see Appendix K - paper [11] - for details). The FSM was used to integrate all the mission sub-tasks, to decrease the vulnerability to individual sub-task failures in real-world conditions, and for coordinating all MAVs sharing the same workspace. In addition to the FSM, a sensor fusion mechanism was proposed for combining information from various onboard sensors (onboard IMU, GPS, DGPS, rangefinder, camera), which must be considered as potentially unreliable at any time.

Malfunctions of MAV subsystems, such as camera dropouts, incorrect rangefinder measurements, gripper failure or gripper feedback failure, and also imprecise object gripping, can be expected in demanding outdoor conditions. All these eventualities need to be considered by the system to enable undisturbed operation of the remaining robots in the event of an MAV failure, limited operation of an MAV with a faulty subsystem, or an unsuccessful or interrupted grasping task. From this point of view, the proposed FSM concept can be considered as a hierarchical state machine with included synchronization and failure recovery abilities, which can be effectively re-used in any complex multi-MAV task involving environment interaction. It is vital that the MAV may continue with the task despite lacking some sensory data (e.g., precisely measured height above the ground) because the competition rules did not allow any human intervention or debugging during the trials. Reliable deployment without the possibility of mission interruption due to a malfunction of a subsystem is also required in most real-world applications of fully autonomous aerial systems. In this way, the MBZIRC competition contributed to robotic state-of-the-art, as had been intended by its organizers. The theoretical results and the FSM methodology introduced in [11] turned out to be one of our biggest contributions in relation to the MBZIRC competition. The FSM technique has been transferred into several other multi-robot projects, such as the cooperative scanning of historical buildings, where safeness and system reliability are even more critical factors.

In the MBZIRC competition, the system that we have presented in [11] enabled to solve the delivery task composed of acquisition, transport, and drop-off in its full complexity, including searching for objects with unknown positions, grasping moving objects, and cooperation among multiple MAVs working in concert. The system exhibited the best performance among all participants in MBZIRC 2017, and our team won the third challenge of the competition (see video <https://youtu.be/-ix6nzxw2wE> and Fig. 23). Our solution achieved the best reliability, and we obtained the highest score in all four trials within the competition.

These results are especially valuable since there were strong wind gusts in the location between the coast and the desert. The variable external force caused by the changing wind significantly influences the precision and the stability of the MAV controllers, particularly in the final phase of object grasping, where it is combined with the ground effect. In addition, the light conditions (mainly the strong and variable sunshine) make the vision task significantly more complicated than in a laboratory environment. For a summary of our achievements in the MBZIRC competition see a video <https://youtu.be/ogmQSjkqqp0>, including some fascinating experiments carried out by the MAV team in dunes near Abu Dhabi, the fastest landing in the entire competition, and examples of objects manipulation in the competition.

In addition, the object grasping and delivery tasks, the overall system designed for the third MBZIRC 2017 competition has been successfully deployed in numerous multi-MAV applications, including detection of sources of radiation and EMF fields [36] (see videos [https://youtu.be/ovp\\_AByx4z4](https://youtu.be/ovp_AByx4z4), <https://youtu.be/ju3YbCtXpEw> and Fig. 24,25), inspection and documentation of historical sites [9] (see section 5.1), cooperative transport of large objects (see videos <https://youtu.be/FQH769AnYbQ>, <https://youtu.be/Pdg3j791I9c> and Fig. 26).

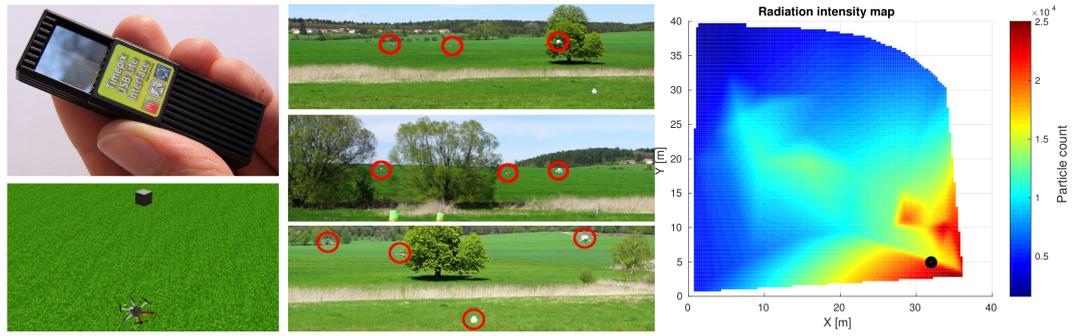


Figure 24: An example of an autonomous source of radiation detection and radiation mapping by a composed radiation sensor consisting of multiple simulated Timepix sensors carried by MAVs. The pictures on the left side show the Timepix sensor and a snapshot from Gazebo simulator, where we have implemented a model of the Timepix sensor and sources of radiation to be able to verify the proposed principles with large teams of robots. Source: [36].



Figure 25: Snapshots from experiments with a formation of MAVs carrying directive (left) and rotating omnidirectional (middle) onboard antennas. The picture on the right side demonstrates the influence of the localization error on the shape and position of the formation. Source: [36].

The system has been utilized in reconnaissance and surveillance missions [37, 38, 39, 40] (see videos <https://youtu.be/5MPSAReNzJU>, <https://youtu.be/fwG2cvAhf3c>, <https://youtu.be/QWzcXShvYIs>, <https://youtu.be/3PhRBCx7fQg>, and <https://youtu.be/gagYFLpGVC4>), to verify fast obstacle avoidance of multiple MAVs [41] (see video <https://youtu.be/cG6V5Dw0kBQ>), etc.

The scientific results obtained during our preparations for the MBZIRC competition have been published in 5 articles in well-recognized journals with impact factor [11], [10], [42], [43], and [37]. These publications provide evidence that innovations beyond the robotic state-of-the-art were achieved within the framework of the competition, in accordance with organisers' intentions.



Figure 26: Snapshots from an outdoor experiment, in which a large object is transported by a pair of MAVs through a narrow passage (left) and through a field with cylindrical obstacles (right).

---

## CONCLUSION

---

This document consists of a compilation of the research achievements of the applicant and his group in the areas of motion planning, localization, control, coordination, and stabilization of compact MAV teams in GNSS-denied environments. In all of the methods presented here, onboard mutual localization of flying robots was used to achieve reliable and safe group stabilization in robot workspaces that are not equipped with an external positioning system. Mutual localization can be considered as a glue connecting all the formation flying and swarming algorithms that are presented here. However, the research on the localization system itself is also a non-negligible contribution to the robotic community. The scientific achievements include theoretical contributions on designing a general methodology for stabilization of compact aerial formations and swarms, accompanied by studies of convergence into a required equilibrium, which have enabled us to specify the requirements for onboard sensors and actuators. Most of the theoretical foundations have been experimentally verified in demanding real-world conditions, leading to realistic assumptions and requirements for the systems. An important aspect of this work is the transfer from basic research achievements into applications, such as cooperative documentation of historical buildings, surveillance, reconnaissance, odour source localization, search and rescue, inspection, object transport, precise agriculture, and information gathering. Our endeavour has been also focused on the deployment of MAVs in areas with high obstacle density, e.g. research of swarming in a forest, which is one of our long-term motivations. In these workspaces, the main capabilities of small MAVs, i.e. better manoeuvrability than F-UAVs in a cluttered workspace, the possibility to operate in low speed, and the ability to hover, are fully exploited.



---

THE CORE PUBLICATIONS SELECTED TO DESCRIBE THE MAIN  
STREAM OF OUR RESEARCH TOWARDS SELF-LOCALIZED  
AUTONOMOUS MULTI-ROBOT SYSTEMS.

---

- [1] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil, "Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.
- [2] M. Saska, T. Krajnik, V. Vonasek, Z. Kasl, V. Spurny, and L. Preucil, "Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1-4, pp. 603–622, 2014.
- [3] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots.*, vol. 41, no. 4, pp. 919–944, 2017.
- [4] M. Saska, T. Baca, and D. Hert, "Formations of unmanned micro aerial vehicles led by migrating virtual leader," in *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.
- [5] M. Saska, J. Vakula, and L. Preucil, "Swarms of Micro Aerial Vehicles Stabilized Under a Visual Relative Localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [6] Daniel Brandtner and Martin Saska, "Coherent swarming of unmanned micro aerial vehicles with minimum computational and communication requirements," in *ECMR*, 2017.
- [7] T. Krajnik, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.
- [8] V. Walter, N. Staub, A. Franchi, and M. Saska, "Uvdar system for visual relative localization with application to leader-follower formations of multirotor uavs," *Accepted in IEEE Robotics and Automation Letters*, 2019.
- [9] Martin Saska, Vit Kratky, Vojtech Spurny, and Tomas Baca, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *IEEE ETFA*, 2017.
- [10] T Baca, P Stepan, V Spurny, D Hert, R Penicka, M Saska, J Thomas, G Loianno, and V Kumar, "Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle," *Journal of Field Robotics*, *online first*, 2019.
- [11] V. Spurny, T. Baca, M. Saska, R. Penicka, T. Krajnik, J. Thomas, D. Thakur, G. Loianno, and V. Kumar, "Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs," *Journal of Field Robotics*, vol. 36, no. 1, pp. 125–148, 2019.



---

RELATED PUBLICATIONS EXTENDING THE CONTRIBUTION OF  
THE CORE PAPERS.

---

- [12] J. Faigl, T. Krajník, J. Chudoba, L. Preucil, and M. Saska, "Low-Cost Embedded System for Relative Localization in Robotic Swarms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [13] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of uavs using ultraviolet led markers," in *International Conference of Unmanned Aircraft System (ICUAS)*, 2018.
- [14] M. Saska, Z. Kasl, and L. Preucil, "Motion Planning and Control of Formations of Micro Aerial Vehicles," in *The 19th World Congress of the International Federation of Automatic Control (IFAC)*, 2014.
- [15] V. Spurný, T. Baca, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016.
- [16] M. Saska, "MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
- [17] M. Saska, J. Langr, and L. Preucil, "Plume Tracking by a Self-stabilized Group of Micro Aerial Vehicles," in *Modelling and Simulation for Autonomous Systems*, 2014.
- [18] M. Saska, J. Chudoba, L. Preucil, J. Thomas, G. Loianno, A. Tresnak, V. Vonasek, and V. Kumar, "Autonomous Deployment of Swarms of Micro-Aerial Vehicles in Cooperative Surveillance," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.
- [19] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles," *Journal of Intelligent & Robotic Systems.*, vol. 84, no. 1-4, pp. 469–492, 2016.
- [20] M. Saska, T. Krajník, V. Vonasek, P. Vanek, and L. Preucil, "Navigation, Localization and Stabilization of Formations of Unmanned Aerial and Ground Vehicles," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013.
- [21] M. Saska, V. Vonasek, T. Krajník, and L. Preucil, "Coordination and Navigation of Heterogeneous UAVs-UGVs Teams Localized by a Hawk-Eye Approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [22] M. Saska, V. Spurný, and L. Preucil, "Trajectory Planning and Stabilization for Formations Acting in Dynamic Environments," in *Progress in Artificial Intelligence*, 2013.
- [23] M. Saska, V. Vonasek, and L. Preucil, "Control of ad-hoc formations for autonomous airport snow shoveling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [24] M. Saska, V. Vonasek, and T. Krajník, "Airport snow shoveling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

- [25] M. Saska, M. Hess, and K. Schilling, "Efficient airport snow shoveling by applying autonomous multi-vehicle formations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [26] M. Saska, J. S. Mejia, D. M. Stipanovic, and K. Schilling, "Control and navigation of formations of car-like robots on a receding horizon," in *3rd IEEE Multi-conference on Systems and Control*, 2009.
- [27] M. Saska, V. Vonasek, and L. Preucil, "Roads Sweeping by Unmanned Multi-vehicle Formations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [28] M. Saska, M. Hess, and K. Schilling, "Route scheduling approach for airport snow shoveling using formations of autonomous ploughs," in *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2008.
- [29] M. Saska, J.S. Mejia, D.M. Stipanovic, V. Vonasek, K. Schilling, and L. Preucil, "Control and Navigation in Manoeuvres of Formations of Unmanned Mobile Vehicles," *European Journal of Control*, vol. 19, no. 2, pp. 157–171, 2013.
- [30] M. Saska, V. Vonasek, and L. Preucil, "Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs," *Journal of Intelligent and Robotic Systems*, vol. 72, no. 2, pp. 239–261, 2013.
- [31] M. Saska, V. Spurny, and V. Vonasek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, no. Part B, pp. 379–397, 2016.
- [32] M. Saska, T. Krajnik, and L. Preucil, "Cooperative uav-ugv autonomous indoor surveillance," in *International Multi-Conference on Systems, Signals Devices*, 2012.
- [33] V. Walter, T. Novák, and M. Saska, "Self-localization of unmanned aerial vehicles using onboard sensors, with focus on optical flow from camera image," in *Modelling and Simulation for Autonomous Systems (MESAS)*, 2017.
- [34] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016.
- [35] V. Walter, M. Saska, N. Staub, and A. Franchi, "Mutual localization of uavs based on blinking ultraviolet markers and 3d time-position hough transform," in *IEEE CASE*, 2018.
- [36] Martin Saska, "Large sensors with adaptive shape realised by self-stabilised compact groups of micro aerial vehicles," in *ISRR*, 2017.
- [37] J. Faigl, P. Vana, R. Penicka, and M. Saska, "Unsupervised Learning based Flexible Framework for Surveillance Planning with Aerial Vehicles," *Journal of Field Robotics*, vol. 36, no. 1, pp. 270–301, 2019.
- [38] Robert Pěnička, Jan Faigl, Petr Váňa, and Martin Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, 2017.
- [39] Robert Pěnička, Martin Saska, Christophe Reymann, and Simon Lacroix, "Reactive dubins traveling salesman problem for replanning of information gathering by uavs," in *European Conference of Mobile Robotics (ECMR)*, 2017.
- [40] Robert Pěnička, Jan Faigl, Petr Váňa, and Martin Saska, "Dubins orienteering problem with neighborhoods," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

- [41] Tomas Baca, Daniel Hert, Giuseppe Loianno, Martin Saska, and Vijay Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [42] G. Loianno, V. Spurny, T. Baca, J. Thomas, D. Thakur, T. Krajník, A. Zhou, A. Cho, M. Saska, and V. Kumar, "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert like environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1576–1583, 2018.
- [43] P. Štěpán, T. Krajník, M. Petrlik, and M. Saska, "Vision techniques for on-board detection, following and mapping of moving targets," *Journal of Field Robotics*, vol. 36, no. 1, pp. 252–269, 2019.



---

## OTHER REFERENCES

---

- [44] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.
- [45] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [46] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.
- [47] A. Purohit, Pei Zhang, B.M. Sadler, and S. Carpin, "Deployment of swarms of micro-aerial vehicles: From theory to practice," in *IEEE ICRA*, 2014.
- [48] Alex Kushleyev, Daniel Mellinger, and Vijay Kumar, "Towards a swarm of agile micro quadrotors," *Robotics: Science and Systems*, 2012.
- [49] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Tamás Nepusz, Agoston E. Eiben, and Tamás Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, pp. 466 – 479, 2018.
- [50] Amedeo Rodi Vetrella, Giancarmine Fasano, and Domenico Accardo, "Attitude estimation for cooperating uavs based on tight integration of gnss and vision measurements," *Aerospace Science and Technology*, vol. 84, pp. 966 – 979, 2019.
- [51] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *IEEE/RSJ IROS*, 2014.
- [52] L. Weng, Q. Liu, M. Xia, and Y. Song, "Immune network-based swarm intelligence and its application to unmanned aerial vehicle (uav) swarm coordination," *Neurocomputing*, vol. 125, no. 1, pp. 134–141, 2014.
- [53] Z. Hou and I. Fantoni, "Interactive leader–follower consensus of multiple quadrotors based on composite nonlinear feedback control," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1732–1743, 2018.
- [54] Wilson O. Quesada, Jonathan I. Rodriguez, Juan C. Murillo, Gustavo A. Cardona, David Yanguas-Rojas, Luis G. Jaimes, and Juan M. Calderón, "Leader-follower formation for uav robot swarm based on fuzzy logic theory," in *International Conference on Artificial Intelligence and Soft Computing*, 2018.
- [55] X. Liu, S. S. Ge, and C. Goh, "Vision-based leader-follower formation control of multiagents with visibility constraints," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2018.
- [56] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C.J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2003.

- [57] J. Fredslund and M.J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 837–846, 2002.
- [58] Gregor Klančar, Drago Matko, and Sašo Blažič, "A control strategy for platoons of differential drive wheeled mobile robot," *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 57 – 64, 2011.
- [59] Hyeun J. Min and Nikolaos Papanikolopoulos, "Robot formations using a single camera and entropy-based segmentation," *Journal of Intelligent and Robotic Systems*, vol. 68, pp. 21–41, 2012.
- [60] Xiaohua Wang, Yimin Deng, and Haibin Duan, "Edge-based target detection for unmanned aerial vehicles using competitive bird swarm algorithm," *Aerospace Science and Technology*, vol. 78, pp. 708 – 720, 2018.
- [61] Qiang Liu, Ming He, Daqin Xu, Ning Ding, and Yong Wang, "A mechanism for recognizing and suppressing the emergent behavior of uav swarm," *Mathematical Problems in Engineering*, pp. 1–14, 2018.
- [62] Liguó Weng, Qingshan Liu, Min Xia, and Y.D. Song, "Immune network-based swarm intelligence and its application to unmanned aerial vehicle (uav) swarm coordination," *Neurocomputing*, vol. 125, pp. 134 – 141, 2014.
- [63] Tomáš Krajník, Jan Faigl, Vojtech Vonásek, Karel Kosnar, Miroslav Kulich, and Libor Preucil, "Simple yet stable bearing-only navigation," *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.
- [64] Carmelo J. A. Bastos Filho, Fernando B. Lima Neto, Anthony J. C. C. Lins, Antônio I. S. Nascimento, and Marília P. Lima, "Fish school search," *Nature-Inspired Algorithms for Optimisation, Studies in Computational Intelligence*, vol. 193, pp. 261–277, 2009.
- [65] Craig Reynolds, "Flocks, herds and schools: A distributed behavioral model.," in *14th annual conference on Computer graphics and interactive techniques*, 1987.
- [66] V. Trianni, *Evolutionary Swarm Robotics*, Springer, 2008.
- [67] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406 – 2411, 2009.
- [68] A. Purohit, Pei Zhang, B.M. Sadler, and S. Carpin, "Deployment of swarms of micro-aerial vehicles: From theory to practice," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [69] S. Berman, A. Halasz, M.A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927 –937, 2009.
- [70] W. Liu, A. Winfield, J. Sa, J. Chen, and L. Dou, "Strategies for energy optimisation in a swarm of foraging robots," *Swarm Robotics*, vol. 4433, pp. 14–26, 2007.
- [71] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 320 –330, 2007.
- [72] A. Winfield, W. Liu, J. Nembrini, and A. Martinoli, "Modelling a wireless connected swarm of mobile robots," *Swarm Intelligence*, vol. 2, pp. 241–266, 2008.

- [73] Dongbing Gu and Huosheng Hu, "Using fuzzy logic to design separation function in flocking algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 826–838, 2008.
- [74] Hongkyu Min and Zhidong Wang, "Design and analysis of group escape behavior for distributed autonomous mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [75] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [76] J. Bulusek, "Active and passive escape behavior designed for groups of autonomous helicopters," in *CTU thesis*, 2014.
- [77] S. Yang, S.a. Scherer, and A. Zell, "An onboard monocular vision system for autonomous take off, hovering and landing of a micro aerial vehicle," *The International Journal of Robotics Research*, vol. 69, no. 1-4, pp. 499–515, 2012.
- [78] M. Bosnak, D. Matko, and S. Blazic, "Quadrocopter hovering using position-estimation information from inertial sensors and a high-delay video system," *The International Journal of Robotics Research*, vol. 67, no. 1, pp. 43–60, 2012.
- [79] E. Stump, V. Kumar, B. Grocholsky, and P.M. Shiroma, "Control for localization of targets using range-only sensors," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 743–757, 2009.
- [80] I. Rekleitis, D. Meger, and G. Dudek, "Simultaneous planning, localization, and mapping in a camera sensor network," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 921–932, 2006.
- [81] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *12th Computer Vision Winter Workshop (CVWW'07)*, 2007.
- [82] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision and Pattern Recognition*. *arXiv:1506.02640*, 2015.
- [83] J. Redmon and A. Farhadi, "Yolog000: Better, faster, stronger," in *Computer Vision and Pattern Recognition*. *arXiv:1612.08242*, 2016.
- [84] Yubo Zhang and Kwan-Liu Ma, "Lighting design for globally illuminated volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2946–2955, 2013.
- [85] J. Birn, *Digital Lighting and Rendering, chapter: Three-Point Lighting for 3D Renderings*, Springer, 2013.
- [86] Alessandro Benini, Matthew J Rutherford, and Kimon P Valavanis, "Real-time, gpu-based pose estimation of a uav for autonomous takeoff and landing," in *International Conference on Robotics and Automation (ICRA)*, 2016.
- [87] Shanggang Lin, Matthew A Garratt, and Andrew J Lambert, "Monocular vision-based real-time target recognition and tracking for autonomously landing an uav in a cluttered shipboard environment," *Autonomous Robots*, vol. 41, no. 4, pp. 881–901, 2017.
- [88] Wooyoung Jung, Youngjoo Kim, and Hyochoong Bang, "Target state estimation for vision-based landing on a moving ground target," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.

- [89] Chun Kiat Tan, Jianliang Wang, Yew Chai Paw, and Fang Liao, "Autonomous ship deck landing of a quadrotor using invariant ellipsoid method," *Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 891–903, 2016.
- [90] Oualid Araar, Nabil Aouf, and Ivan Vitanov, "Vision based autonomous landing of multirotor uav on moving platform," *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 369–384, 2017.
- [91] Khaled A Ghamry, Yiqun Dong, Mohamed A Kamel, and Youmin Zhang, "Real-time autonomous take-off, tracking and landing of uav on a moving ugv platform," in *24th Mediterranean Conference on Control and Automation (MED)*, 2016.
- [92] Hanseob Lee, Seokwoo Jung, and David Hyunchul Shim, "Vision-based uav landing on the moving vehicle," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.
- [93] Lingyun Xu and Haibo Luo, "Towards autonomous tracking and landing on moving target," in *International Conference on Real-time Computing and Robotics (RCAR)*, 2016.
- [94] Alexandre Borowczyk, Duc-Tien Nguyen, André Phu-Van Nguyen, Dang Quang Nguyen, David Saussié, and Jerome Le Ny, "Autonomous landing of a quadcopter on a high-speed ground vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 2378–2385, 2017.
- [95] Tru Hoang, Enkhmurun Bayasgalan, Ziyin Wang, Gavriil Tsechpenakis, and Dimitra Panagou, "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," in *American Control Conference (ACC)*, 2017.
- [96] Shaogang Jin, Jiyang Zhang, Lincheng Shen, and Tengxiang Li, "On-board vision autonomous landing techniques for quadrotor: A survey," in *35th Chinese Control Conference (CCC)*, 2016.
- [97] Justin Thomas, Giuseppe Loianno, Joseph Polin, Koushil Sreenath, and Vijay Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration & Biomimetics*, vol. 9, no. 2, pp. 250–275, 2014.
- [98] Vaibhav Ghadiok, Jeremy Goldin, and Wei Ren, "On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor," *Autonomous Robots*, vol. 33, no. 1, pp. 41–68, 2012.
- [99] M. Kamel, S. Comari, and R. Siegwart, "Full-body multi-objective controller for aerial manipulation," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016.
- [100] S. Kannan, S. Quintanar-Guzman, J. Dentler, M. A. Olivares-Mendez, and H. Voos, "Control of aerial manipulation vehicle in operational space," in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016.
- [101] A. Santamaria-Navarro, P. Grosch, V. Lippiello, J. Sola, and J. Andrade-Cetto, "Uncalibrated visual servo for unmanned aerial manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1610–1621, 2017.
- [102] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1912–1923, 2016.

- [103] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. Á. Trujillo, Y. R. Esteves, and A. Viguria, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2016.
- [104] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE International Conference on Robotics and Automation*, 2011.
- [105] Pablo Ramon Soria, Begoña C Arrue, and Anibal Ollero, "Detection, location and grasping objects using a stereo sensor on uav in outdoor environments," *Sensors*, vol. 17, no. 1, pp. 103–126, 2017.
- [106] Pablo Ramon Soria, Robert Bevec, Begoña C Arrue, Aleš Ude, and Anibal Ollero, "Extracting objects for aerial manipulation on uavs using low cost stereo sensors," *Sensors*, vol. 16, no. 5, pp. 700–725, 2016.





---

KEY ARTICLE [1] - INTERNATIONAL JOURNAL OF ROBOTICS  
RESEARCH 2014

---

©[2014] SAGE Publications. This article was published in International Journal of Robotics Research: **M. Saska**, V. Vonasek, T. Krajnik, and L. Preucil: Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach, 2014.



# Coordination and navigation of heterogeneous MAV–UGV formations localized by a ‘hawk-eye’-like approach under a model predictive control scheme

Martin Saska<sup>1</sup>, Vojtěch Vonásek<sup>1</sup>, Tomáš Krajník<sup>2</sup>  
and Libor Přeučil<sup>1</sup>

## Abstract

*An approach for coordination and control of 3D heterogeneous formations of unmanned aerial and ground vehicles under hawk-eye-like relative localization is presented in this paper. The core of the method lies in the use of visual top-view feedback from flying robots for the stabilization of the entire group in a leader–follower formation. We formulate a novel model predictive control-based methodology for guiding the formation. The method is employed to solve the trajectory planning and control of a virtual leader into a desired target region. In addition, the method is used for keeping the following vehicles in the desired shape of the group. The approach is designed to ensure direct visibility between aerial and ground vehicles, which is crucial for the formation stabilization using the hawk-eye-like approach. The presented system is verified in numerous experiments inspired by search-and-rescue applications, where the formation acts as a searching phalanx. In addition, stability and convergence analyses are provided to explicitly determine the limitations of the method in real-world applications.*

## Keywords

Formation control, model predictive control, unmanned aerial vehicles, unmanned ground vehicles, trajectory planning, obstacle avoidance, receding horizon control

## 1. Introduction

Precise relative localization within large teams of unmanned vehicles is required in real-world search-and-rescue (SAR) missions, where a multi-robot system has to cooperatively explore large areas in a short time. In these tasks, robots may not rely on a pre-installed global localization infrastructure, which is usually not available, or is damaged, in sites under SAR exploration. Systems available worldwide (like GPS) lack the required precision for compact formations of small robots, and lose reliability in urban and indoor environments. A common solution is to use systems of relative localization carried onboard autonomous vehicles. This approach brings additional movement constraints to the robotic team, which have to be integrated into the formation control and stabilization.

In this paper, we present a formation-driving approach adapted for onboard visual relative localization of heterogeneous teams of unmanned helicopters (quadrotors) and ground robots. The localization is based on simple lightweight bottom cameras mounted on unmanned micro aerial vehicles (MAVs). Identification patterns for the localization are placed on both the unmanned ground vehicles (UGVs) and the MAVs. With this top-view approach, the

problem of loss of direct visibility can be tackled better. Such a problem occurs if systems of visual relative localization are employed for ground robots operating in a workspace with scattered objects/obstacles, as is common in SAR scenarios. The possibility of localizing robots from the top view increases robustness and precision in determining the relative position. Additionally, the top view brings another perspective for human operators supervising the mission. MAVs may also complement the team of UGVs with their ability to visit/search places inaccessible to ground vehicles, as we demonstrated in Saska et al. (2012b). For more opportunities and advantages to MAVs, see the survey in Kumar and Michael (2012).

The proposed MAV–UGV stand-alone system provides a lightweight, low-cost and efficient solution, which may act

<sup>1</sup>Department of Cybernetics, Czech Technical University in Prague, Czech Republic

<sup>2</sup>Lincoln Centre for Autonomous Systems, School of Computer Science, University of Lincoln, UK

## Corresponding author:

Martin Saska, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague 6, Czech Republic.

Email: saskam1@fel.cvut.cz

as an enabling technique for extensive utilization of simple micro-scale robots in demanding scenarios. This article focuses on theoretical and implementation aspects of the formation-driving mechanism suited for the real-world deployment of autonomous robots relying on top-view relative localization (in this paper referred to as the *hawk-eye* concept). Technical details on the visual relative localization of team members from ‘flying cameras’ are omitted, but they are available in Saska et al. (2012a). The main contribution of the work presented here lies in the proposed incorporation of the hawk-eye concept into formation control and its shape stabilization. The idea of using the hawk-eye relative localization of team members requires new formation-driving and robot control approaches that are presented in the rest of the paper.

The aim of our research effort is to enable the deployment of closely cooperating groups of MAVs outside laboratories equipped with precise motion-capture systems (e.g. the Vicon system) and, on the basis only of relative localization, to achieve the same results obtained using these systems nowadays (Turpin et al., 2011; Kushleyev et al., 2012; Mellinger et al., 2012).

### 1.1. State-of-the-art methods and progress beyond the current formation-driving approaches

Formation-driving algorithms can be divided into three main approaches: virtual structures (Beard et al., 2001; Ren, 2008; Michael and Kumar, 2009; Ghommam et al., 2010; Liu and Jia, 2012), behavioral techniques (Langer et al., 1994; Lawton et al., 2003; Olfati-Saber, 2006), and leader–follower methods (Desai et al., 2001; Fredslund and Mataric, 2002; Das et al., 2003; Mastellone et al., 2008; Sira-Ramiandrez and Castro-Linares, 2010; Yang et al., 2010; Klančar et al., 2011; Min and Papanikolopoulos, 2012). For further references on distributed robotic control see Bullo et al. (2009). In our work, we consider a modification of the leader–follower method, in which all robots (MAVs and UGVs) of the formation follow a virtual leader. Formation stabilization is achieved by sharing knowledge of the virtual leader’s position within the formation.

Recently, research endeavor in the formation-driving community has been aimed mainly at tasks of formation stabilization (Hengster-Movrić et al., 2010; Dong, 2011; Liu and Jia, 2012) and formation following a predefined path (Xiao et al., 2009; Ghommam et al., 2010; Sira-Ramiandrez and Castro-Linares, 2010; Do and Lau, 2011). For example, in Dong (2011), the task of formation stabilization and convergence to a desired pattern is tackled for formations with communication delays. In Hengster-Movrić et al. (2010), a multi-agent control system using an artificial potential based on bell-shaped functions is proposed. In Liu and Jia (2012), a distributed iterative learning scheme is employed for solving formation control with a switching strategy in the virtual structure and virtual leader–follower schemes.

The path-following problem is tackled by designing a nonlinear formation control law in Ghommam et al. (2010). The method based on the virtual structure approach uses propagation of a virtual target along the path. In Do and Lau (2011), path-following is investigated for groups of robots with a limited sensing range. In Sira-Ramiandrez and Castro-Linares (2010), according to the leader–follower concept, the leader robot is forced to follow a given path, while the followers track the leader’s path with a fixed time delay. In Xiao et al. (2009), in addition to trajectory tracking, the autonomous design of a desired geometric formation pattern is discussed.

In addition to methods of formation driving for UGVs, we should mention some approaches designed for unmanned aerial vehicles (UAVs) (Saffarian and Fahimi, 2009; Burdakov et al., 2010; Abdessameud and Tayebi, 2011; Liu et al., 2011; No et al., 2011). In No et al. (2011), the formation stabilization and desired-shape-keeping are treated as a dynamic 3D tracking problem. The relative geometry of multiple UAVs is kept via a cascade-type guidance law under the leader–follower concept. A leader–follower approach for stabilizing helicopter formations using a nonlinear model predictive control (MPC) is proposed in Saffarian and Fahimi (2009). This method is optimized for an online embedded solution enabling a response to the fast dynamic of UAVs in Liu et al. (2011). In Burdakov et al. (2010), UAVs in a static formation form relay chains for communication in surveillance applications. The formation stabilization of the vertical take-off and landing of UAVs in the presence of communication delays is addressed in Abdessameud and Tayebi (2011). Finally, let us mention Tanner and Christodoulakis (2007), who consider a heterogeneous team of UAVs and UGVs. The aim of the approach is to stabilize a formation of UAVs above UGVs in circular orbits using interconnections of UAV and UGV groups via ground-to-air-only communication.

In most of the approaches cited above, it is supposed that the desired trajectory followed by the formation is designed by a human operator or by a standard path-planning method modified for the formation requirements. The method presented in this paper goes beyond these works. It does not rely on following a given trajectory, as in most of the state-of-the-art methods. The global trajectory planning is directly integrated into the formation control mechanism. This is necessary for finding a feasible solution for the hawk-eye concept, where the constraints of direct visibility have to be satisfied. Direct incorporation of trajectory planning and formation stabilization enables effective operation of the group in an environment with obstacles, while the hawk-eye relative localization is ensured.

In the literature, direct inclusion of trajectory planning in formation driving is rarely found. To the best of our knowledge, we can mention only the leader–follower approach based on potential fields presented in Garrido et al. (2011) as an appropriate example. This algorithm enables both formation stabilization and navigation of the formation into

a desired goal. Although the method provides interesting results and seems to be computationally inexpensive, it has been developed for ground holonomic robots, and it suffers from the usual problems of algorithms inspired by potential fields. The authors state that their method behaved correctly in spite of the suddenly changed direction of movement around the sequence of points given by the planning method. However, it would be difficult to explicitly involve follower mobility constraints for a formation of nonholonomic robots. Further research would also be necessary to incorporate the requirements of 3D formations, and in particular the constraints given by the relative visual localization as proposed in the method presented here. Therefore, this method cannot be used for direct comparison with results achieved in this paper.

Finally, we should mention the paper by Dorigo et al. (2013), which is similar to our work in terms of deployment of eye-bots on the ceiling with the aim of providing a bird's-eye view. In Dorigo et al. (2013), the eye-bots are not moving together with ground robots if the top-view localization is in operation; they are fixed with the ceiling. Therefore, the motion coordination and formation driving of heterogeneous teams do not need to be solved there.

### *1.2. State-of-the-art methods and progress beyond the current MPC approaches for formation control*

In our method, we rely on MPC. This allows us to involve constraints imposed by vehicles (mobility constraints), by obstacles (environment constraints), and by inter-vehicle relations into the formation driving. The inter-vehicle relations are specified mainly by the shape of the formation feasible for the hawk-eye-like relative localization.

The MPC approach is often used for stabilization of nonlinear systems with control constraints. In Saffarian and Fahimi (2009) and Liu et al. (2011), it was shown that the computational power of microprocessors available onboard unmanned helicopters enables the employment of MPC techniques also for the formation control of these highly dynamic systems, as is proposed here.

For descriptions and for a general survey of MPC methods, see Barambones and Etxebarria (2000), Mayne et al. (2000) and Alamir (2006) and the references therein. Early works applying MPC to formation control are presented in Dunbar and Murray (2006) and Franco et al. (2008). These papers utilized MPC for formation forming in a workspace without obstacles. Recently, researchers have taken advantage of MPC to respond to changes in a dynamic environment, again, mainly in tasks including path tracking and formation stabilization (Saffarian and Fahimi, 2009; Shin and Kim, 2009; Chen et al., 2010; Defoort, 2010; Zhang et al., 2010; Liu et al., 2011; Chao et al., 2012). In Chao et al. (2012), the authors introduce a new cost penalty

into MPC optimization to guarantee obstacle avoidance. A priority strategy is employed to ensure inter-vehicle collision avoidance. In Defoort (2010), a decentralized receding horizon motion planner is developed to coordinate robots using neighbor-independent planning. This is followed by adjusting the plans with inter-team collisions using locally exchanged information. The trajectory tracking mechanism developed in Zhang et al. (2010) is based on integrating a differential evolution algorithm into the MPC concept. In Shin and Kim (2009), a heuristic approach is developed to reduce the required computational time of MPC iterations and to enable path tracking with an obstacle-avoidance function. Formation stabilization on a pre-computed path based on the MPC leader-follower concept is presented in Chen et al. (2010).

In our approach, we go beyond these works in several aspects. We apply the MPC technique to the stabilization of followers in the desired positions behind the leader, as well as to the trajectory planning into a desired goal area. We propose a new MPC concept combining the trajectory planning into the desired goal region and the immediate control of the formation in a single optimization process. The method can continuously respond to changes in the vicinity, while keeping the cohesion of the immediate control inputs with the directions of movement of the MAV-UGV formation in the future. Furthermore, we propose a novel obstacle-avoidance function for multi-vehicle trajectory planning. The avoidance function respects the restrictions of the proposed hawk-eye concept.

The paper is structured as follows. The problem statement is summarized in Section 2. In Section 3, necessary preliminaries are given. The novel methodology is described in Section 4, focusing on the utilization of a heterogeneous formation of UGVs and MAVs under the hawk-eye concept. A proof of the convergence of the formation into the desired target region, together with a discussion on the assumptions necessary to ensure formation stability under the hawk-eye relative localization, is shown in Section 5. Numerical and hardware experiments are presented in Section 6, which is followed by our conclusions in Section 7. A discussion on a controller for the Ar.Drone, which was designed to enable integration of the drone into the proposed MPC formation stabilization scheme, can be found in the Appendix. For clarification purposes, lists of variables used in this paper are summarized in Tables 1 and 2.

The basic ideas of the planning for heterogeneous MAV-UGV formations under hawk-eye relative localization were presented in a conference paper (Saska et al., 2012c), which is extended here. In comparison with Saska et al. (2012c), we provide here a more comprehensive description of the method, accompanied by the proof of convergence. An additional extension is the real-world experiment and numerical analysis verifying the robustness of the proposed methodology.

**Table 1.** List of variables and notation used in the preliminary part of the paper.

$n_r \in \mathbb{N}$	Number of followers
$n_0 \in \mathbb{N}$	Number of static and dynamic obstacles
$L$	Variables related to the virtual leader
$i$	Variables related to the $i$ th follower
$j$	Variables related to the $j$ th entity, a follower or the virtual leader
$\mathcal{C}$	Configuration space of robots
$\mathcal{C}_{obs}$	Subspace of configurations of robots colliding with an obstacle
$\mathcal{C}_{free}$	Subspace of feasible configurations
$o_l$	The $l$ th obstacle
$S_F$	Desired target region
$\varphi_j(t) \in \mathbb{R}$	Heading of the $j$ th entity at time $t$
$\psi_j(t) \in \mathbb{R}^4$	Configuration (position and heading) of the $j$ th entity at time $t$
$\bar{p}_j(t) \in \mathbb{R}^3$	Position of the $j$ th entity at time $t$
$(x_j(t), y_j(t), z_j(t))$	Position in Cartesian coordinates at time $t$
$(p_i, q_i, h_i)$	Position of the $i$ th follower within the formation in curvilinear coordinates
$v_j(t) \in \mathbb{R}$	Forward velocity of the $j$ th entity at time $t$
$K_j(t) \in \mathbb{R}$	Curvature of the $j$ th entity at time $t$
$w_j(t) \in \mathbb{R}$	Ascent velocity of the $j$ th entity at time $t$
$\bar{u}_j(t) \in \mathbb{R}^3$	Control inputs (velocity and heading)
$\Delta t(k) \in \mathbb{R}$	Time difference between the $k$ th and $(k + 1)$ th transition points

**Table 2.** List of variables used for describing the method.

$(\cdot)_{max,j}$	Index denoting the upper bound of the control inputs of the $j$ th entity
$(\cdot)_{min,j}$	Index denoting the lower bound of the control inputs of the $j$ th entity
$r_s \in \mathbb{R}$	Radius of a spherical detection boundary
$r_a \in \mathbb{R}$	Radius of a spherical avoidance boundary
$T_N$	First part of the control horizon with a constant time difference between transition points: it provides the local control
$T_M$	Second part of the control horizon with a variable time difference between transition points: it provides the global planning
$N \in \mathbb{N}$	Number of transition points on $T_N$
$M \in \mathbb{N}$	Number of transition points on $T_M$
$n \in \mathbb{N}$	Number of transition points (on $T_N$ ), which are applied in each receding step
$\Delta t(\cdot) \in \mathbb{R}$	Variable time difference between transition points on the time interval $T_M$
$\Delta t$	Constant sampling time between transition points on the time interval $T_N$
$\mathcal{T}_{L,M}^\Delta \in \mathbb{R}^M$	Set of varying values of time difference between neighboring transition points on the interval $T_M$
$\Psi_{L,N} \in \mathbb{R}^{4N}$	Set of states (transition points) on $T_N$
$\Psi_{L,M} \in \mathbb{R}^{4M}$	Set of states (transition points) on $T_M$
$\mathcal{U}_{L,N} \in \mathbb{R}^{3N}$	Set of control inputs applied between the transition points on interval $T_N$
$\mathcal{U}_{L,M} \in \mathbb{R}^{3M}$	Set of control inputs applied between the transition points on interval $T_M$
$\Omega_L \in \mathbb{R}^{7N+8M}$	Optimization vector used for trajectory planning of the virtual leader
$(\cdot)^\circ$	Denotes results of the optimization process
$\Psi_{d,i} \in \mathbb{R}^{3N}$	Set of desired states derived from $\Omega_L^\circ$ for the $i$ th follower
$\Psi_i \in \mathbb{R}^{3N}$	Set of states (transition points) of the $i$ th follower
$\mathcal{U}_i \in \mathbb{R}^{2N}$	Set of control inputs of the $i$ th follower applied between the transition points
$\Omega_i \in \mathbb{R}^{5N}$	Optimization vector used for trajectory tracking of the $i$ th follower
$\{x^L, y^L\}$	Coordinate system in the plane orthogonal to the trajectory of the virtual leader in its current position
CH	Convex hull of points, in which the followers intersect the plane orthogonal to the trajectory of the virtual leader in its actual position
DCH	The convex hull CH dilated by the detection boundary radius $r_s$
PDCH	Projection of the DCH PDCH along the leader's trajectory
$R_{DCH}$	Half of the maximal width of the DCH measured in the $x^L$ coordinate
$D(\cdot)$	The perturbations given by the imprecise model and actuators between two MPC planning steps

## 2. Problem statement

In this paper, we consider the formation-driving problem in scenarios where a team of robots has to reach a desired target region or a sequence of target regions given by a scenarios motivated by SAR applications. We are interested

supervising expert. During the movement between these given areas, the robots have to keep a fixed-shape formation satisfying the mission requirements. The robots can form a searching phalanx (a line formation) to be able to search for victims in large areas or they can form a compact fleet of vehicles (a formation of a general shape) for transportation purposes.

We assume a group of simple ground nonholonomic robots without any onboard sensors for their localization. In addition, we assume a group of unmanned MAVs (quadrotors) equipped with a bottom camera and an image-processing system (Saska et al., 2012a). The image-processing system provides information on the relative position between the camera and the center of an identification pattern. The patterns are carried by all UGVs and MAVs except the one flying at the highest altitude. We assume that one of the robots (UGV or MAV) is equipped with a global localization (e.g. the vision-based navigation in Krajník et al., 2010). The precision and reliability of the system in Krajník et al. (2010) enables a rough estimation of the position of the formation in the map, but it is not sufficient for the coordination of the robots in a compact formation.

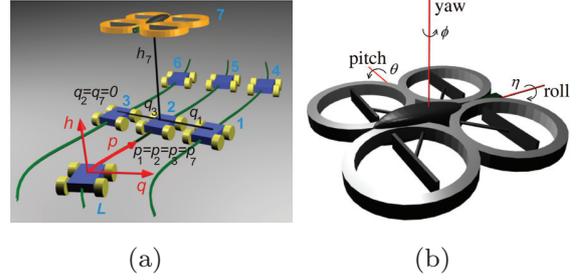
We assume that the required relative distances between the robots are significantly bigger than the precision of the visual relative localization. The precision of the employed system (described in Saska et al., 2012a) is  $\sim 1$  cm. Therefore the minimal allowed distance between the robots is 10 cm in the experiments. Between two UAVs, the spacing usually has to be enlarged, due to airflow effects that depend on the utilized platform. Moreover, we assume that the shape of the formation is designed in such a way that all robots, except the MAV follower flying at the highest altitude, are in the field of view of at least one of the bottom cameras mounted on the MAVs.

In the assumed scenario, the map of the environment is partly known by all robots. The group is capable of detecting unknown and dynamic obstacles using their onboard sensors. These updates of the map are shared by the robots via WiFi communication. The position of the target region or a sequence of target regions and the desired shape of the formation are also known.

In this paper, we solve the task in which the 3D formations of MAVs and UGVs have to reach a target region or a sequence of target regions, while the requirements given by hawk-eye relative localization are satisfied. This means that direct visibility between the vehicles has to be maintained during deployment of the formation.

### 3. Preliminaries

Let  $\psi_j(t) = \{x_j(t), y_j(t), z_j(t), \varphi_j(t)\} \in \mathcal{C}$ , with  $j \in \{L, 1, \dots, n_r\}$ , denote configurations of a virtual leader  $L$  and  $n_r$  followers at time  $t$ . The virtual leader is positioned in front of the formation and on the axis of the formation, which is important for the symmetric obstacle-avoidance



**Fig. 1.** (a) The desired shape of the formation described in curvilinear coordinates. (b) Coordination system of the quadcopter.

function.  $\mathcal{C}$  is the configuration space of the robots. The Cartesian coordinates  $x_j(t)$ ,  $y_j(t)$ , and  $z_j(t)$  define the positions  $\bar{p}_j(t)$  of the robots, and  $\varphi_j(t)$  denotes their heading. All MAVs and UGVs are denoted as followers in the approach presented here. For the MAVs, the heading  $\varphi_j(t)$  directly becomes the yaw (see Figure 1 for the coordinates system of the MAVs). The roll and the pitch do not need to be included directly in the kinematic model employed in the MPC. They depend on the velocity and the turning curvature, as shown for a quadrotor helicopter in the Appendix.

Let us assume that the environment of the robots contains a finite number  $n_0$  of compact obstacles  $o_l$ ,  $l \in \{1, \dots, n_0\}$ . The configuration space  $\mathcal{C}$  can then be divided into two segments:  $\mathcal{C}_{obs}$ , representing the configurations of the robots colliding with an obstacle, and  $\mathcal{C}_{free}$ , representing the subspace of the feasible configurations as  $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$ .

**Definition 3.1. (Target region.)** Let us define a target region  $S_F$  as a convex compact region such that, for any robot with position  $\bar{p}_j(\cdot) \in S_F$ , the relation  $\psi_j(\cdot) \in \mathcal{C}_{free}$  is satisfied.

The kinematics for any robot  $j$  in 3D is described by the following simple nonholonomic kinematic model:

$$\begin{aligned} \dot{x}_j(t) &= v_j(t) \cos \varphi_j(t) \\ \dot{y}_j(t) &= v_j(t) \sin \varphi_j(t) \\ \dot{z}_j(t) &= w_j(t) \\ \dot{\varphi}_j(t) &= K_j(t) v_j(t) \end{aligned} \quad (1)$$

Forward velocity  $v_j(t)$ , curvature  $K_j(t)$ , and ascent velocity  $w_j(t)$  represent control inputs denoted as  $\bar{u}_j(t) = \{v_j(t), K_j(t), w_j(t)\}$ . For UGVs (in the presented results for car-like robots), these control inputs can be directly employed for steering them. (We assume that UGVs operate on a flat surface and that  $z_j(\cdot) = 0$  and  $w_j(\cdot) = 0$  for each of the UGVs.) In case of MAVs,  $v_j(\cdot)$ ,  $K_j(\cdot)$ , and  $w_j(\cdot)$  values are inputs for the controller shown in the Appendix.

Let us now define a time interval  $[t_0, t_{end}]$  containing a finite sequence of elements of increasing time  $\{t_0, t_1, \dots, t_{end-1}, t_{end}\}$ , such that  $t_0 < t_1 < \dots < t_{end-1} < t_{end}$ .

The control inputs are held constant in each time interval  $[t_k, t_{k+1})$ , where  $k \in \{0, \dots, \text{end} - 1\}$ . From this point we may refer to  $t_k$  by using its index  $k$ . By integrating the kinematic model over the interval  $[t_0, t_{\text{end}}]$ , we can derive the following model for transition points at which the control inputs change:

$$x_j(k+1) = \begin{cases} x_j(k) + \frac{1}{K_j(k+1)} [\sin(\varphi_j(k) + \\ K_j(k+1)v_j(k+1)\Delta t(k+1)) - \\ \sin(\varphi_j(k))] , \text{ if } K_j(k+1) \neq 0; \\ x_j(k) + v_j(k+1)\cos(\varphi_j(k))\Delta t(k+1) \\ \text{if } K_j(k+1) = 0 \end{cases}$$

$$y_j(k+1) = \begin{cases} y_j(k) - \frac{1}{K_j(k+1)} [\cos(\varphi_j(k) + \\ K_j(k+1)v_j(k+1)\Delta t(k+1)) - \\ \cos(\varphi_j(k))] , \text{ if } K_j(k+1) \neq 0; \\ y_j(k) + v_j(k+1)\sin(\varphi_j(k))\Delta t(k+1) , \\ \text{if } K_j(k+1) = 0 \end{cases}$$

$$z_j(k+1) = z_j(k) + w_j(k+1)\Delta t(k+1)$$

$$\varphi_j(k+1) = \varphi_j(k) + K_j(k+1)v_j(k+1)\Delta t(k+1) \quad (2)$$

where  $x_j(k)$ ,  $y_j(k)$ , and  $z_j(k)$  are Cartesian coordinates and  $\varphi_j(k)$  is the heading angle at the transition point with index  $k$  for any robot  $j \in \{L, 1, \dots, n_r\}$ . The sampling time  $\Delta t(k+1)$  may not be uniform in the whole interval  $[t_0, t_{\text{end}}]$ , as shown below. The control inputs  $v_j(k+1)$ ,  $K_j(k+1)$ , and  $w_j(k+1)$  are constant between the transition points with indexes  $k$  and  $k+1$ . For each follower  $i \in \{1, \dots, n_r\}$ , the control inputs are limited by vehicle kinematic constraints (i.e. implied by the steering limitations and the drive system) as  $v_{\min,i} \leq v_i(k) \leq v_{\max,i}$ ,  $|K_i(k)| \leq K_{\max,i}$  and for the MAVs also  $w_{\min,i} \leq w_i(k) \leq w_{\max,i}$ . These values may differ for each of the followers.

Finally, we need to define a spherical detection boundary with radius  $r_s$  and a spherical avoidance boundary with radius  $r_a$ , where  $r_s > r_a$ . Single robots should not respond to obstacles detected outside the region with radius  $r_s$ . On the contrary, a distance between the robots and obstacles of less than  $r_a$  is considered inadmissible.

### 3.1. Formation-driving concept

The shape of the entire formation is maintained by a leader-follower technique based on the method presented in Barfoot and Clark (2004). The approach in Barfoot and Clark (2004) was designed for formations of UGVs working in a planar environment. Later, it was employed in an airport snow-shoveling project by formations of autonomous ploughs in Hess et al. (2009) and Saska et al. (2011). Here, we extend the notation from Barfoot and Clark (2004) to 3D.

In the proposed method, both types of followers, MAVs and UGVs, follow the same trajectory of the virtual leader in distances defined in the  $p, q, h$  curvilinear coordinate system, as visualized in Figure 1(a). The position of each follower  $i$  is uniquely determined: 1) by states  $\psi_L(t_{p_i})$  in the traveled distance  $p_i$  from the actual position of the virtual

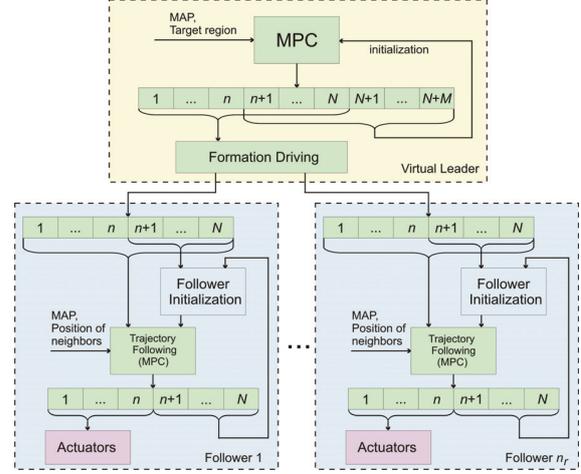


Fig. 2. Scheme of the complete planning and control system.

leader along the leader's trajectory; 2) by the *offset distance*  $q_i$  from the trajectory in the perpendicular direction; and 3) by the elevation  $h_i$  above the trajectory. Note that  $t_{p_i}$  denotes the time when the virtual leader was at the traveled distance  $p_i$  behind the actual position.

To convert the state of the followers in curvilinear coordinates to a state in Cartesian coordinates, the following equations can be applied:

$$\begin{aligned} x_i(t) &= x_L(t_{p_i}) - q_i \sin(\varphi_L(t_{p_i})) \\ y_i(t) &= y_L(t_{p_i}) + q_i \cos(\varphi_L(t_{p_i})) \\ z_i(t) &= z_L(t_{p_i}) + h_i \\ \varphi_i(t) &= \varphi_L(t_{p_i}) \end{aligned} \quad (3)$$

where  $\psi_L(t_{p_i}) = \{x_L(t_{p_i}), y_L(t_{p_i}), z_L(t_{p_i}), \varphi_L(t_{p_i})\}$  is the state of the virtual leader at time  $t_{p_i}$ .

The virtual leader has no constraints given by its mechanical capabilities. It is a virtual point, but it must respect the constraints of the guided formation. For the virtual leader, the admissible control set can be determined by applying the leader-follower approach as

$$\begin{aligned} K_{\max,L} &= \min_{i=1,\dots,n_r} \left( \frac{K_{\max,i}}{1 + q_i K_{\max,i}} \right) \\ K_{\min,L} &= \max_{i=1,\dots,n_r} \left( \frac{-K_{\max,i}}{1 - q_i K_{\max,i}} \right) \\ v_{\max,L}(t) &= \min_{i=1,\dots,n_r} \left( \frac{v_{\max,i}}{1 + q_i K_L(t)} \right) \\ v_{\min,L}(t) &= \max_{i=1,\dots,n_r} \left( \frac{v_{\min,i}}{1 + q_i K_L(t)} \right) \\ w_{\max,L} &= \min_{i=1,\dots,n_r} (w_{\max,i}) \\ w_{\min,L} &= \max_{i=1,\dots,n_r} (w_{\min,i}) \end{aligned} \quad (4)$$

These restrictions must be applied to satisfy different values for the curvature and the speed of the robots in different

positions within the formation. Intuitively, the robot following the inner track during turning goes more slowly but with a bigger curvature than the robot further from the center of the turning. The equations arise from the fact that the followers turn around the same instantaneous center of curvature (ICC) and at the same angular speed. These restrictions ensure that the formation remains compact while turning.

The common ICC implies that robots with different positions within the formation have to turn with different curvatures. Therefore, the limits on the curvature of the leader's trajectory must ensure that all of the robots are capable of following a curvature that depends on their position within the formation. The constant angular speed of robots turning with a different curvature forces the followers to move at different velocities to be able to pass the curve at the same time. Again, the limits on the leader's velocity must ensure that all of the robots are capable of going at the velocity that is determined by their position within the formation.

## 4. Integrated trajectory planning and formation stabilization under the hawk-eye concept

### 4.1. Method overview

The proposed formation-driving system is divided into two blocks: see the scheme depicted in Figure 2. In the *Virtual Leader* part, the *Trajectory Planning* block provides the complete trajectory into the target region for the virtual leader. The result is feasible for the entire formation and respects the requirements of the hawk-eye localization via the model of the formation. For this trajectory planning and control task, we have developed a novel method based on the MPC. The standard MPC solves a finite horizon optimization control problem for the system represented by the kinematic model. The MPC plan starts from the current states over the time interval  $\langle t_0, t_0 + N\Delta t \rangle$ . This interval is known as the *control horizon*. The sampling time  $\Delta t$  between the  $N$  transition points is constant in this interval. We denote this horizon as  $T_N$ . We have extended this standard scheme with an additional time interval  $\langle t_0 + N\Delta t, t_0 + (N + M)\Delta t \rangle$ . This *planning horizon* is used for planning the trajectory of the leader into the desired target region. The time difference between the  $M$  transition points is variable in this time interval, which is denoted by  $T_M$ . This planning algorithm respects the constraints given by the desired shape of the formation, by the hawk-eye localization, and by the kinematics of the followers. In our approach, the entire horizon is divided into two segments: 1) the control horizon with a constant sampling rate used to obtain a refined immediate control, and 2) the planning horizon, where the time differences between the transition points are also variables that take part in the planning problem. Details on the construction of the horizons, with emphasis on the incorporation of the 3D formation, are presented in Section 4.3.

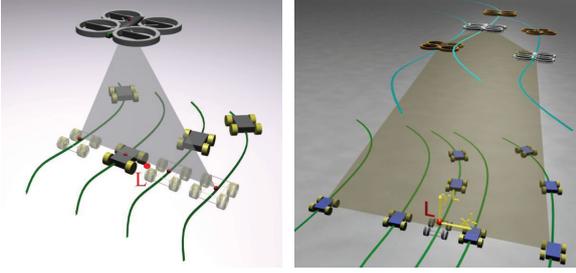
The resulting trajectory obtained in the *Trajectory Planning* block is described by a sequence of configurations of the virtual leader  $\psi_L(k)$ ,  $k \in \{1, \dots, N + M\}$ , and by constant control inputs applied in the intervals between the transition points. According to the MPC concept, only a portion of the computed control actions is applied. This utilized interval,  $\langle t_0, t_0 + n\Delta t \rangle$ , is known as the receding step. In the next planning step, this process is repeated on the interval  $\langle t_0 + n\Delta t, t_0 + n\Delta t + N\Delta t \rangle$  as the finite horizon moves by the time steps  $n\Delta t$ , yielding a state feedback control scheme strategy. The output trajectory is used as an input for the *Formation Driving* module in the proposed system. In this module, the plan is transformed to the desired configurations of the followers (using equation (3)). Additionally, the plan is adapted for re-initialization of the optimization in the next planning step.

The core of the second main block is the *Trajectory Following* module. This part enables the design of appropriate collision-free control inputs for each of the MAV and UGV followers. It is responsible for avoiding impending collisions with obstacles or with other members of the team, and it corrects deviations from the desired trajectory provided by the virtual leader. Again this is ensured by the MPC concept using the actual data (states of neighbors and the map) which is shared within the team. Details on the control of followers can be found in Section 4.4.

### 4.2. Convex hull representing the formation in the trajectory planning

An important issue, which arises with the trajectory planning for heterogeneous 3D formations using the hawk-eye relative localization, is the need to design a valuable representation of the entire group. The aim of the representation is to incorporate the requirement of direct visibility between the robots into the concept of the trajectory planning with obstacle avoidance and formation stabilization functionalities.

We propose modeling the entire shape of the 3D formation defined in curvilinear coordinates with a convex hull of points representing the positions of the followers. The points are obtained by projecting the followers' positions into the plane, which is orthogonal to the trajectory of the virtual leader in its actual position (see Figure 3). To describe how to acquire these points, let us define a coordinate system  $\{x^L; y^L\}$  in this plane, as sketched in Figure 4. The projection of the  $i$ th follower's position can then be obtained as  $x_i^L := q_i$  and  $y_i^L := h_i$ . The convex hull of the set of points  $\{x_i^L; y_i^L\}$ ,  $i \in \{1, \dots, n_r\}$ , is an appropriate representation of the 3D formation in the proposed leader-follower constellation for two reasons: 1) Each follower  $i$  intersects the plane orthogonal to the trajectory of the virtual leader at the point  $\{x_i^L; y_i^L\}$  in the future. 2) The convex hull of this set of points denotes the borders of the area that should remain obstacle-free. It ensures the direct visibility



**Fig. 3.** Two examples of convex hulls of asymmetric formations. The formation on the right side is utilized in the experiment in Figure 6. The shaded contours represent projections of MAVs and UGVs into the plane of the virtual leader.

between MAVs and UGVs that is crucial for the presented formation-driving using the hawk-eye localization.

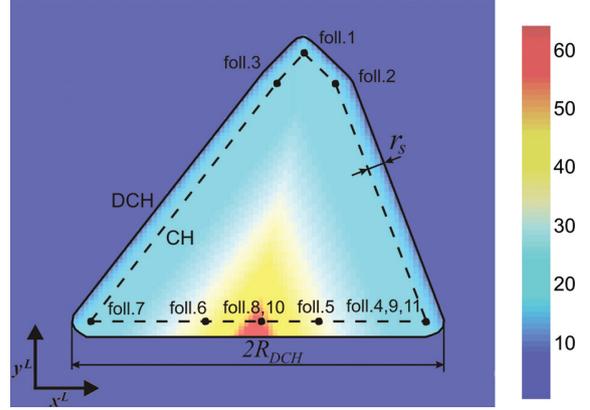
Moreover, for the obstacle-avoidance function presented in Section 4.3, the convex hull needs to be dilated by the detection boundary radius  $r_s$ . This ensures that obstacles are kept at a sufficient distance from the followers. An example of the dilated convex hull (DCH) of a formation is depicted in Figure 4.

#### 4.3. Trajectory planning and control for the virtual leader

As mentioned above, we propose solving, in a single optimization step, two problems that are usually separated: long-term trajectory planning feasible for the formation, and computation of the immediate control sequences. To define the trajectory planning problem over the two time intervals (the control horizon and the planning horizon) in a compact form, we need to gather states  $\psi_L(k)$ , where  $k \in \{1, \dots, N\}$ , and  $\psi_L(k)$ , where  $k \in \{N+1, \dots, N+M\}$ , into vectors  $\Psi_{L,N} \in \mathbb{R}^{4N}$  and  $\Psi_{L,M} \in \mathbb{R}^{4M}$ . Similarly, the control inputs  $\tilde{u}_L(k)$ , where  $k \in \{1, \dots, N\}$ , and  $\tilde{u}_L(k)$ , where  $k \in \{N+1, \dots, N+M\}$ , can be gathered into vectors  $\mathcal{U}_{L,N} \in \mathbb{R}^{3N}$  and  $\mathcal{U}_{L,M} \in \mathbb{R}^{3M}$ , one for each of the horizons. Finally, the values  $\Delta t(k)$ , where  $k \in \{N+1, \dots, N+M\}$ , which become variables in the planning horizon, can be gathered into a vector  $\mathcal{T}_{L,M}^\Delta \in \mathbb{R}$ . All variables that describe the complete trajectory from the actual position of the virtual leader until the target region can be collected into the optimization vector  $\Omega_L = [\Psi_{L,N}, \mathcal{U}_{L,N}, \Psi_{L,M}, \mathcal{U}_{L,M}, \mathcal{T}_{L,M}^\Delta] \in \mathbb{R}^{7N+8M}$ .

The trajectory planning and the dynamic obstacle avoidance problem can then be transformed to minimization of the cost function  $J_L(\cdot)$ . The function is subject to sets of equality constraints  $h(\cdot)$  and inequality constraints  $g_{T_N}(\cdot)$ ,  $g_{T_M}(\cdot)$ ,  $g_{S_F}(\cdot)$ :

$$\begin{aligned} \min J_L(\Omega_L), \text{ s.t. } & h(k) = 0, \forall k \in \{0, \dots, N+M-1\} \\ & g_{T_N}(k) \leq 0, \forall k \in \{1, \dots, N\} \\ & g_{T_M}(k) \leq 0, \forall k \in \{N+1, \dots, N+M\} \\ & g_{S_F}(\psi_L(N+M)) \leq 0 \end{aligned} \quad (5)$$



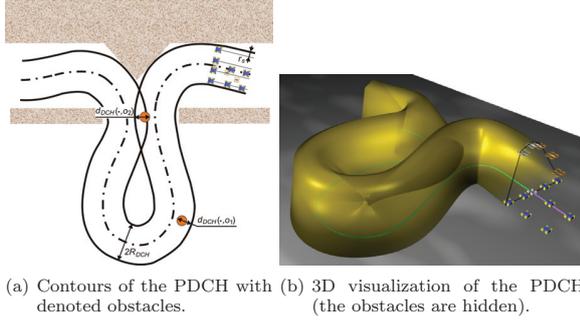
**Fig. 4.** A color map of the function that ensures collision-free trajectories for a formation operating under the hawk-eye concept. The color map was composed for the second formation introduced in Figure 3.

The cost function  $J_L(\Omega_L)$  is given by

$$\begin{aligned} J_L(\Omega_L) &= J_{L,time}(\Omega_L) + \alpha J_{L,obstacles}(\Omega_L) \\ &= \left( N\Delta t + \sum_{k=N+1}^{N+M} \Delta t(k) \right) \\ &\quad + \alpha \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{d_{DCH}(\Omega_L, o_l)}{d_{DCH}(\Omega_L, o_l) - R_{DCH}} \right\} \right)^2 \end{aligned} \quad (6)$$

The first part,  $J_{L,time}(\Omega_L)$ , minimizes the total time to the target region. The second term  $J_{L,obstacles}(\Omega_L)$  is an avoidance function motivated by Stipanović et al. (2007), where a similar approach was used for cooperative collision avoidance in multi-agent systems. In our case, the term  $J_{L,obstacles}(\Omega_L)$  contributes to the final cost when an obstacle is inside the DCH representing the formation. Its value (the penalization) increases as the obstacle approaches the center of the convex hull. The aim of this term is to penalize solutions of the virtual leader trajectory planning in which an obstacle is inside the DCH projected along the trajectory that corresponds to the solution. This is to prevent collisions or breakages of direct visibility between robots by the obstacle. A breakage of direct visibility could interrupt the relative localization necessary for steering the followers. Let us denote the projection of the DCH along the leader's trajectory  $\Omega_L$  as PDCH. An example of PDCH is depicted in Figure 5. The decrease of the penalization value with distance from the center of PDCH and its zero value at the borders of PDCH are required properties necessary for optimization convergence into a feasible solution.

These properties are obtained as follows. The constant  $R_{DCH}$  is equal to half of the maximal width of DCH measured in the  $x^L$  coordinate. The meaning of  $R_{DCH}$  is also denoted in Figures 4 and 5. The function  $d_{DCH}(\Omega_L, o_l)$  provides the shortest distance in the direction of the  $x^L$  coordinate from the furthest part of obstacle  $o_l$  to the borders of PDCH. See the obstacle  $o_1$  and the denoted value



**Fig. 5.** An example of the DCH projected along a trajectory. This trajectory would be infeasible for the formation under the approach presented here, since two obstacles appear inside the PDCH. This is penalized by the  $d_{DCH}(\cdot, o_1)$  and  $d_{DCH}(\cdot, o_2)$  values in the cost function. The hull overlaps due to the sharp curve of the trajectory and therefore obstacle  $o_2$  occurs in the hull twice. As marked, only the occurrence of the obstacle that is furthestmost from the border of the hull contributes to the penalization function  $d_{DCH}(\cdot, o_2)$ .

$d_{DCH}(\cdot, o_1)$  for illustration in Figure 5. The function value is positive if the obstacle is in PDCH and negative if the obstacle is completely outside the projected hull. If an obstacle occurs in several projections of DCH along  $\Omega_L$  (e.g. in a sharp curve of the trajectory), it is counted only once in equation (6). Always, only the largest value of the set of the shortest distances from the obstacle to the border of the relevant projection of DCH is used (see obstacle  $o_2$  in Figure 5). The direction of the gradient of an avoidance function so defined (see the values of this function depending on the position of an obstacle in Figure 4) is to the side of DCH in the  $x^L$  coordinate. This feature is important for the convergence of the optimization process into an obstacle-free solution. If the resulting trajectory changes its shape during the optimization with the aim of having the obstacle outside DCH, the value of  $J_{L,obstacles}(\Omega_L)$  decreases smoothly. This is important for the convergence of the optimization into a feasible solution.

The influence of both parts of the cost function is adjusted by the constant  $\alpha$ . The value of  $\alpha$  needs to be set empirically depending on the particular application. A value within the range 100–1000 is recommended if safety of the system is preferred. Values in the range of 0.01–1 should be used in applications requiring short and fast solutions. A compromise value  $\alpha = 1$  is used for the experimental results presented in this paper.

The equality constraints  $h(k)$ ,  $\forall k \in \{0, \dots, N + M - 1\}$ , represent the kinematic model in equation (2) with the initial conditions given by the actual state of the leader. This ensures that the obtained trajectory stays feasible with respect to the kinematics of the utilized robots. The sets of inequality constraints  $g_{T_N}(k)$ ,  $\forall k \in \{1, \dots, N\}$ , for the control horizon, and  $g_{T_M}(k)$ ,  $\forall k \in \{N + 1, \dots, N + M\}$ , for the planning horizon, characterize the limits on the control inputs (equation (4)) of the virtual leader. Furthermore, the

constraints  $g_{T_M}(k)$  ensure that inequalities  $\Delta t(k) \geq 0$  are satisfied for  $\forall k \in \{N + 1, \dots, N + M\}$ .

Finally,  $g_{S_F}(\psi_L(N + M))$  is a convergence constraint guaranteeing that the found trajectory enters the target region  $S_F$ . For simplification, it is supposed that the target region is a sphere with radius  $r_{S_F}$  and center  $C_{S_F}$ . Then, the convergence constraint is given by

$$g_{S_F}(\psi_L(N + M)) := \|\bar{p}_L(N + M) - C_{S_F}\| - r_{S_F} \quad (7)$$

Let us denote the solution of the optimization problem in equation (5) by the symbol  $(\cdot)^\circ$ . As mentioned above, the vector  $\Omega_L^\circ$  represents a continuous trajectory with the beginning at time  $t_1$  and the end at time  $t_2$ . The trajectory reaches the desired target region and it is feasible for the formation using the hawk-eye relative localization. Let us denote such a trajectory as  $\Omega_L(t_1; t_2)^\circ$ , if necessary for further analysis.

Once we have obtained a feasible trajectory as a result of the optimization process, we can write the following remark.

**Remark 4.1.** Each trajectory  $\Omega_L(t_1; t_3)^\circ$  can be split into two parts  $\Omega_L(t_1; t_2)^\circ$  and  $\Omega_L(t_2; t_3)^\circ$ , where  $t_1 < t_2 < t_3$ .

This is possible due to the fact that the trajectory consists of a sequence of transition points and a sequence of constant control inputs applied between the points. The splitting can be realized simply by placing a new transition point at time  $t_2$  on the trajectory  $\Omega_L(t_1; t_3)^\circ$ . This transition point is part of both the trajectories arising, which satisfy the constraints in equation (5), except the  $g_{S_F}(\cdot)$  convergence constraint, which is not satisfied for the  $\Omega_L(t_1; t_2)^\circ$  part.

Remark 4.1 is important for the convergence analysis of the formation movement into the desired target region presented in Section 5. During the movement, always only a part of the trajectory is followed by the formation in the MPC concept. It is important to show that this splitting is feasible and that the group of robots will approach the target in a sequence of replanning steps of the optimization problem considered in equation (5).

#### 4.4. Trajectory tracking for followers

In accordance with the leader–follower concept, the trajectory of the virtual leader, which is computed as the result of the previous section, is used as an input for trajectory tracking for the followers. First of all, the solution needs to be transformed for each of the following vehicles using the transformation in equation (3). This transformation takes place in the *Formation Driving* block in Figure 6. The obtained sequences  $\psi_{d,i}(k) = (\bar{p}_{d,i}(k), \varphi_{d,i}(k))$ , where  $k \in \{1, \dots, N\}$ , are then utilized as the desired states for the trajectory tracking algorithm with the obstacle-avoidance function for each of the followers (MAVs and UGVs). This approach (realized in the *Path Following* block in Figure 2) enables responses to events that occur in the environment behind the actual position of the leader, and to incorrect movement of a neighbor in the formation.

Similarly to the leader planning in Section 4.3, the states  $\psi_i(k)$  and the control vectors  $\bar{u}_i(k)$ , where  $k \in \{1, \dots, N\}$ , describing the trajectory of the  $i$ th follower, can be gathered as vectors  $\Psi_i \in \mathbb{R}^{4N}$  and  $\mathcal{U}_i \in \mathbb{R}^{3N}$ . The optimization vector  $\Omega_i = [\Psi_i, \mathcal{U}_i] \in \mathbb{R}^{7N}$  is then used to capture the dynamic behavior of the discrete trajectory tracking with a collision avoidance ability as a static optimization process under the receding horizon scheme.

The discrete-time trajectory tracking for each follower is then transformed to an optimization problem with the cost function  $J_i(\cdot)$ . The function is subject to a number of equality constraints  $h_i(\cdot)$  and inequality constraints  $g_i(\cdot)$ :

$$\begin{aligned} \min J_i(\Omega_i), i \in \{1, \dots, n_r\}, \\ \text{s.t. } h_i(k) = 0, \forall k \in \{0, \dots, N-1\} \\ g_i(k) \leq 0, \forall k \in \{1, \dots, N\} \end{aligned} \quad (8)$$

The proposed cost function  $J_i(\cdot)$  consists of three components with their influence adjusted by constants  $\alpha_i$  and  $\beta_i$  (set as  $\alpha_i = 1$  and  $\beta_i = 1$  in the experimental part of this paper):

$$\begin{aligned} J_i(\Omega_i) = & \sum_{k=1}^N \left\| (\bar{p}_{d,i}(k) - \bar{p}_i(k)) \right\|^2 \\ & + \alpha_i \left( \min \left\{ 0, \frac{\text{dist}(\Omega_i) - r_s}{\text{dist}(\Omega_i) - r_a} \right\} \right)^2 \\ & + \beta_i \sum_{j \in \bar{n}_n} \left( \min \left\{ 0, \frac{d_{i,j}(\Omega_i, \Omega_j^\circ) - r_s}{d_{i,j}(\Omega_i, \Omega_j^\circ) - r_a} \right\} \right)^2 \end{aligned} \quad (9)$$

The first component penalizes deviations of the positions  $\bar{p}_i(k)$  from the desired positions  $\bar{p}_{d,i}(k)$ ,  $\forall k \in \{1, \dots, N\}$ . As mentioned above, the desired positions are derived from the result of the virtual leader planning using the formation-driving approach presented in Section 3.1. The second term in  $J_i(\cdot)$  ensures that dynamic or lately detected obstacles are avoided. The function  $\text{dist}(\Omega_i)$  provides the Euclidean distance between the closest obstacle and the follower's trajectory. The third component of  $J_i(\cdot)$  is the sum of the avoidance functions in which the other members of the team are also considered to be dynamic obstacles. This part protects the robots in the case of unexpected behavior of defective neighbors. Function  $d_{i,j}(\Omega_i, \Omega_j^\circ)$  returns the minimal distance between the planned trajectory of follower  $i$  and the actually used plan of other followers  $j \in \bar{n}_n$ , where  $\bar{n}_n = \{1, \dots, i-1, i+1, \dots, n_r\}$ . The equality constraints  $h_i(\cdot)$  are identical to the equality constraints  $h(\cdot)$  in Section 4.3. The inequality constraints  $g_i(\cdot)$  are identical to the constraints  $g_{T_N}(\cdot)$ .

The shape of  $J_i(\Omega_i)$  allows the repositioning of followers (UGVs and MAVs) with the aim of obstacle avoidance, compensation for actuators' and sensors' uncertainty, or collision avoidance between neighbors. Each UGV follower  $i$  can change its position and heading by optimizing its curvature  $K_i$  and velocity  $v_i$ . The MAV followers

may also change their altitude  $z_i$  by optimizing their ascent velocity  $w_i$ .

Finally, we should highlight that only the first  $n$  control inputs of the obtained solutions  $\Omega_i^\circ$  are used for steering of robots in the MPC concept. The rest of these solutions can be recycled via the *Follower Initialization* module (depicted in Figure 2) in the next iteration. This approach significantly decreases the computational time required for optimization, since the unused remainder of the solution only needs to be changed due to movement of dynamic obstacles or due to diminishing of disturbances. The influence of the initialization is even more perceivable in the leader trajectory planning. Not only the part of control inputs on the control horizon, but also the complete solution on the planning horizon, can be re-utilized there.

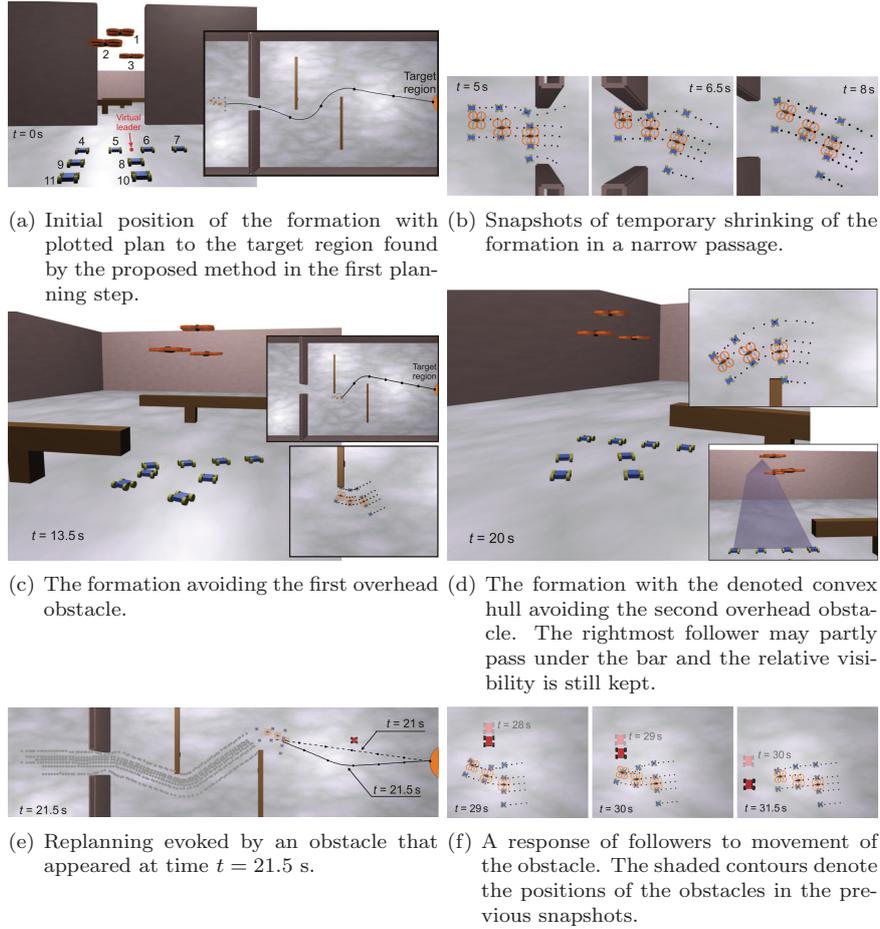
## 5. Analysis of formation convergence from a feasible initial solution

This section aims to verify that the formation-driving method is capable of navigating the formation into the target region if feasible solutions  $\Omega_L(\cdot; \cdot)^\circ$  and  $\Omega_i(\cdot; \cdot)^\circ$ , with  $i \in \{1, \dots, n_r\}$ , are known at initial time  $t_0$ . The initial feasible solutions, which satisfy the constraints given by equations (5) and (8), can be found by the optimization method proposed herein, or provided by a high-level planning system. In particular, this section suggests a proof that the formation will reach the target region with solution  $\Omega_L(\cdot; \cdot)^\circ$  that is always replanned after every  $n$  control steps. In addition, it shows that the followers will be stabilized in their positions within the formation by using plans  $\Omega_i(\cdot; \cdot)^\circ$ , which are updated with a period of  $n$  control steps. This periodic replanning is important for compensation for sensor and actuator uncertainties and for dynamic obstacle avoidance.

We should emphasize that the aim of this section is not to prove convergence to feasible solutions for the optimization problems introduced by equations (5) and (8). The local convergence of these optimization processes is guaranteed by properties of the cost functions, which decrease smoothly and contain local extremes that correspond to sub-optimal trajectories. However, a global optimization method that is always able to find the globally optimal solutions of problems in (5) and (8) in a reasonable amount of time is not available. Therefore, it is not possible to guarantee that feasible initial solutions of problems (5) and (8) will be found even if such feasible solutions do exist.

The aim of this analysis is the specification of the conditions necessary for the reaching of the desired equilibrium by the formation. It enables guaranteeing that the obtained initial plan is feasible for the group. To be able to show the convergence of the entire formation into the desired target region under the approach presented here, let us first specify an assumption on the desired reachability.

**Assumption 1.** (*Desired reachability.*) *At the initial time  $t_0$ , there exists a feasible solution of the optimization problem introduced in equation (5). The solution represents the*



**Fig. 6.** Simulation with eight UGVs and three MAVs verifying the performance of the proposed formation-driving algorithm.

trajectory for the formation to reach the target region. It guarantees that the trajectory is situated at a sufficient distance from obstacles and that direct visibility between the robots is ensured, which is a crucial aspect of relative localization under the hawk-eye concept. In addition, the utilized optimization method is capable of finding such a solution, not necessarily globally optimal, from the initial configuration  $\psi_L(t_0) \in C_{free}$  to any configuration  $\psi_L(t_f)$ , with  $t_f > t_0$ , which is inside the target region.

Further, we need to show that the following lemmas hold for the cost function introduced in equation (6).

**Lemma 5.1.** *Splitting any trajectory  $\Omega_L(t_1; t_3)$  with the beginning at time  $t_1$  and the end at time  $t_3$ , which satisfies the constraints given in equation (5), into two parts  $\Omega_L(t_1; t_2)$  and  $\Omega_L(t_2; t_3)$ , where  $t_1 < t_2 < t_3$ , the following inequality holds:  $J_L(\Omega_L(t_1; t_3)) \leq J_L(\Omega_L(t_1; t_2)) + J_L(\Omega_L(t_2; t_3))$ .*

*Proof.* Let us suppose that the new transition point added at time  $t_2$  (as described in Remark 4.1) lies on the trajectory  $\Omega_L(t_1; t_3)$  between the  $K$ th and  $(K + 1)$ th transition points. Thus,  $t_K < t_2 < t_{K+1}$ , where  $t_K$  and  $t_{K+1}$  are the times of the  $K$ th and  $(K + 1)$ th transition points, respectively.

As introduced in equation (6), the value of the first part of the cost function of  $\Omega_L(t_1; t_3)$  is obtained as

$$J_{L,time}(\Omega_L(t_1; t_3)) = N\Delta t + \sum_{k=N+1}^{N+M} \Delta t(k) \quad (10)$$

If  $K < N$ , which means that the new transition point is placed within the interval  $T_N$ , the first part of the cost function of the split trajectories can be expressed as

$$J_{L,time}(\Omega_L(t_1; t_2)) = K\Delta t + (t_2 - t_K) \quad (11)$$

and

$$J_{L,time}(\Omega_L(t_2; t_3)) = (N - K - 1)\Delta t + (t_{K+1} - t_2) + \sum_{k=N+1}^{N+M} \Delta t(k) \quad (12)$$

Combining equations (10), (11), and (12) together with equation  $\Delta t = t_{K+1} - t_K$ , we can write

$$\begin{aligned} J_{L,time}(\Omega_L(t_1; t_2)) + J_{L,time}(\Omega_L(t_2; t_3)) &= K\Delta t + \Delta t + N\Delta t \\ &\quad - K\Delta t - \Delta t + \sum_{k=N+1}^{N+M} \Delta t(k) \\ &= J_{L,time}(\Omega_L(t_1; t_3)) \end{aligned} \quad (13)$$

If  $K \geq N$ , which means that the new transition point is placed within the interval  $T_M$ , the first part of the cost function of the split trajectories can be expressed as

$$J_{L,time}(\Omega_L(t_1; t_2)) = N\Delta t + (t_2 - t_K) + \sum_{k=N+1}^{K-1} \Delta t(k) \quad (14)$$

and

$$J_{L,time}(\Omega_L(t_2; t_3)) = (t_{K+1} - t_2) + \sum_{k=K+1}^{N+M} \Delta t(k) \quad (15)$$

Considering equations (10), (14), and (15) together with equation  $\Delta t(K) = t_{K+1} - t_K$ , we can again write

$$\begin{aligned} J_{L,time}(\Omega_L(t_1; t_2)) + J_{L,time}(\Omega_L(t_2; t_3)) &= N\Delta t + \Delta t(K) \\ &+ \sum_{k=N+1}^{K-1} \Delta t(k) + \sum_{k=K+1}^{N+M} \Delta t(k) \\ &= J_{L,time}(\Omega_L(t_1; t_3)) \end{aligned} \quad (16)$$

The second part of the cost function of  $\Omega_L(t_1; t_3)$  is expressed as

$$\begin{aligned} J_{L,obstacles}(\Omega_L(t_1; t_3)) \\ = \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{d_{DCH}(\Omega_L(t_1; t_3), o_l)}{d_{DCH}(\Omega_L(t_1; t_3), o_l) - R_{DCH}} \right\} \right)^2 \end{aligned} \quad (17)$$

and similarly

$$\begin{aligned} J_{L,obstacles}(\Omega_L(t_1; t_2)) + J_{L,obstacles}(\Omega_L(t_2; t_3)) &= \\ = \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{d_{DCH}(\Omega_L(t_1; t_2), o_l)}{d_{DCH}(\Omega_L(t_1; t_2), o_l) - R_{DCH}} \right\} \right)^2 &+ \\ + \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{d_{DCH}(\Omega_L(t_2; t_3), o_l)}{d_{DCH}(\Omega_L(t_2; t_3), o_l) - R_{DCH}} \right\} \right)^2 \end{aligned} \quad (18)$$

As already mentioned, obstacles contribute to the cost function if they are placed inside the DCH projected along the trajectory. It is clear that obstacles appearing in the projection of DCH along  $\Omega_L(t_1; t_3)$  must also contribute with the same value in one of the split parts. Therefore, the value of the sum  $J_{L,obstacles}(\Omega_L(t_1; t_2)) + J_{L,obstacles}(\Omega_L(t_2; t_3))$  cannot be smaller than  $J_{L,obstacles}(\Omega_L(t_1; t_3))$ . Nevertheless, it may happen that an obstacle appears in the projection of DCH along both parts, since these can overlap (e.g. in a sharp turn of the trajectory, as shown in Figure 5). In this case, the obstacle contributes to the cost functions twice and the value of the sum is increased. This multiple appearance is eliminated in  $J_{L,obstacles}(\Omega_L(t_1; t_3))$  as follows. To obtain the value of the function  $d_{DCH}(\Omega_L(t_1; t_3), o_l)$ , the shortest distance from the border of DCH projected along  $\Omega_L(t_1; t_3)$  to obstacle  $o_l$  in the direction of the  $x^L$  coordinate has to be computed. If obstacle  $o_l$  occurs in the projection of DCH multiple times (e.g. like obstacle  $o_2$  in Figure 5), all

the shortest distances have to be obtained. The value of the function  $d_{DCH}(\Omega_L(t_1; t_3), o_l)$  is then the largest value from the set of these shortest distances.

Considering these observations together with equation (13) and equation (16), we can conclude that

$$J_L(\Omega_L(t_1; t_3)) \leq J_L(\Omega_L(t_1; t_2)) + J_L(\Omega_L(t_2; t_3)) \quad (19)$$

□

**Lemma 5.2.** *Splitting any trajectory  $\Omega_L(t_1; t_3)$ , with  $t_1 < t_3$  and which satisfies the constraints given in equation (5), at the time  $t_T$  of entering into the target region, the following inequality holds:  $J_L(\Omega_L(t_1; t_3)) \geq J_L(\Omega_L(t_1; t_T))$ .*

*Proof.* Since the trajectory  $\Omega_L(t_1; t_3)$  satisfies the constraints in equation (5), the following inequality holds:  $t_1 < t_T \leq t_3$ .

If  $t_T = t_3$ , one can directly write that  $J_L(\Omega_L(t_1; t_3)) = J_L(\Omega_L(t_1; t_T))$ .

If  $t_T < t_3$ , considering equation (10) for  $t_2 := t_T$  and the evident fact that  $J_{L,time}(\Omega_L(t_T; t_3)) > 0$  (see for example equation (12)), one can write that  $J_{L,time}(\Omega_L(t_1; t_3)) > J_{L,time}(\Omega_L(t_1; t_T))$ . Taking into account the fact that only the obstacles contributing to  $J_{L,obstacles}(\Omega_L(t_1; t_3))$  may also contribute to  $J_{L,obstacles}(\Omega_L(t_1; t_T))$ , one can write that  $J_{L,obstacles}(\Omega_L(t_1; t_3)) \geq J_{L,obstacles}(\Omega_L(t_1; t_T))$ . Combining the inequalities for  $J_{L,time}$  and  $J_{L,obstacles}$ , we obtain the inequality  $J_L(\Omega_L(t_1; t_3)) > J_L(\Omega_L(t_1; t_T))$  for situations with  $t_T < t_3$ . □

**Remark 5.3.** *Considering Remark 4.1 and Lemma 5.2, we can conclude that by splitting any trajectory, which satisfies the constraints in equation (5), at the time of crossing the border of the target region a feasible solution of the formation to the target region problem, evaluated by a lower (or the same) value of the cost function, is obtained.*

Now, we are prepared to show the convergence of the formation into the target region.

**Theorem 5.4.** *Under Assumption 1, having a feasible solution of the problem in equation (5) at time  $t_0$ , the formation is guided by the MPC scheme towards the target region if the inequality  $D(k) < J_L(\Omega_L(\tau; n\Delta t + \tau)^\circ)$ , where  $\tau = kn\Delta t + t_0$ ,  $k \in \mathbb{Z}^+$ , and  $k < (t_T - t_0)/n\Delta t$ , is satisfied.  $D(k)$  denotes the perturbations on  $J_L(\Omega_L(\cdot)^\circ)$  and  $t_T$  is the time at which the formation approaches the target region.*

*Proof.* Being inspired by the theory of nonlinear systems in Chapter 4 of Khalil (2001), we can prove the convergence of the formation into the target region if we show the decrease in the value of the cost function introduced in equation (6) over time. This means that we have to show the conditions in which the following inequality holds:

$$J_L(\Omega_L(n\Delta t + \tau; t_2)^\circ) - J_L(\Omega_L(\tau; t_1)^\circ) < 0 \quad (20)$$

In this equation, the term  $J_L(\Omega_L(\cdot; \cdot)^\circ)$  is the cost of the solution found by the optimization method. The vector  $\Omega_L(\tau; t_1)^\circ$  represents the computed trajectory of the virtual leader with the beginning at time  $\tau$  and the end at time  $t_1$ . The term  $J_L(\Omega_L(n\Delta t + \tau; t_2)^\circ)$  represents the cost of the optimization vector found in the next control step. This solution is used after applying the first  $n$  elements of the trajectory with the beginning at time  $\tau$ . Using an ideal optimization method, which is always capable of finding the global optimal solution, both solutions would end directly on the border of the target region. Any trajectory containing a part inside the target region may not be optimal, which is obvious from Remark 5.3 and Lemma 5.2. With real optimization algorithms working in a finite time, it is impossible to find the global optimal solution. The obtained solutions that satisfy the constraints from equation (5) always terminate inside the target region (not exactly on the border). The length of the part of the trajectory inside the desired target region can differ in each planning step of the MPC algorithm. Therefore, we have to omit this part inside the region to be able to show the convergence of the formation to the target region by analyzing the contraction of the trajectory between consequent MPC planning steps.

This means that we have to split the trajectories  $\Omega_L(\tau; t_1)^\circ$  and  $\Omega_L(n\Delta t + \tau; t_2)^\circ$  at the time when they enter the target region. In Remarks 4.1 and 5.3 and Lemma 5.2 it is shown that such shortened trajectories satisfy the constraints from equation (5), and they represent solutions of the optimization problem with lower values for the cost function. Equation (20) may then be rewritten as

$$J_L(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) - J_L(\Omega_L(\tau; t_{T1})^\circ) < 0 \quad (21)$$

where  $t_{T1}$  and  $t_{T2}$  are the time instants at which the trajectories  $\Omega_L(\tau; t_1)^\circ$  and  $\Omega_L(n\Delta t + \tau; t_2)^\circ$  enter the target region. Further, let us split the trajectory  $\Omega_L(\tau; t_{T1})^\circ$  at time  $n\Delta t + \tau$ , as shown in Remark 4.1. Using Lemma 5.1, we can write that

$$J_L(\Omega_L(\tau; t_{T1})^\circ) \leq J_L(\Omega_L(\tau; n\Delta t + \tau)^\circ) + J_L(\Omega_L(n\Delta t + \tau; t_{T1})^\circ) \quad (22)$$

Combining equations (21) and (22) we obtain

$$-J_L(\Omega_L(\tau; n\Delta t + \tau)^\circ) - J_L(\Omega_L(n\Delta t + \tau; t_{T1})^\circ) + J_L(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) < 0 \quad (23)$$

Let us substitute

$$-J_L(\Omega_L(n\Delta t + \tau; t_{T1})^\circ) + J_L(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) := D(k) \quad (24)$$

into equation (23), so that we obtain the inequality  $D(k) < J_L(\Omega_L(\tau; n\Delta t + \tau)^\circ)$ , which represents limits on perturbations. The inequality is satisfied if the formation is outside the target region, which agrees with the limitation  $\tau = kn\Delta t + t_0$ , where  $k \in \mathbb{Z}^+$  and  $k < (\bar{t} - t_0)/n\Delta t$ , as stated in Theorem 5.4.  $\square$

### 5.1. Analysis of results of the convergence proof

The aim of this subsection is to show the meaning of the perturbations  $D(k)$  and a practical utilization of the results of the convergence analysis presented in the previous section.

Analyzing equation (24), one can see that perturbations  $D(k)$  represent changes (usually an increase) in the values of the cost function that evaluates the trajectories found in two consecutive MPC planning steps. The trajectories, both beginning at time  $n\Delta t + \tau$  and ending on the border of the target region, are found by the optimization method, one at time  $\tau$  and one at time  $n\Delta t + \tau$ .

- 1) Let us first consider a situation without dynamic or unknown obstacles. In this case, the increase of the value of the second term of the cost function (6) may be neglected:  $J_{L,obstacles}(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) - J_{L,obstacles}(\Omega_L(n\Delta t + \tau; t_{T1})^\circ) \doteq 0$ , and  $D(k) \doteq J_{L,time}(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) - J_{L,time}(\Omega_L(n\Delta t + \tau; t_{T1})^\circ)$ . The perturbations are therefore caused mainly by imprecise actuators and by the simplification of the kinematic model. This results in deviations in the position of the formation after each MPC step. These deviations need to be compensated for, and they prolong the total time to the goal by the time difference  $T_2 - T_1$ . In situations without dynamic or unknown obstacles, this time difference is approximately equal to the value  $D(k)$ . Considering Theorem 5.4 and the obvious fact that  $J_{L,obstacles}(\Omega_L(\tau; n\Delta t + \tau)^\circ) \geq 0$ , it has to be ensured that

$$D(k) < n\Delta t \quad (25)$$

It can be seen that the value of the cost function is decreasing and the plant converges to the desired target region. This inequality is important for practical utilization of the method.

- 2) In the presence of dynamic obstacles or suddenly detected obstacles, the difference  $J_{L,obstacles}(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) - J_{L,obstacles}(\Omega_L(n\Delta t + \tau; t_{T1})^\circ)$  may be the dominant part of the perturbations. Equation (25) may then be violated even if the uncertainty of the actuators is sufficiently small. In this case, the convergence to the target region is temporarily broken, which can be detected by the increase in the cost function value. In real-world applications, it is sometimes necessary to allow a temporary increase in the value of the cost function. For example, newly detected obstacles can be avoided by the replanning included in the MPC concept, and the convergence is restored. A problem occurs in the presence of dynamic obstacles that push, by their influence via the cost function, the overall formation from the target region. If such a situation is detected by long-term growth of the value of the cost function, the formation has to stop and the planning process needs to be restarted.

### 5.2. Analysis of the stability of the followers in their desired positions within the formation

The stability of the formation in the desired shape is solved through the distribution of transformed states of the virtual leader using the formation-driving concept in Barfoot and Clark (2004), extended here to the 3D case (see Section 3.1). Using this approach, the formation stabilization is transformed into the independent trajectory tracking processes running onboard the following robots. The classical MPC approach is then employed for trajectory tracking with the obstacle- and failing-neighbor-avoidance functionalities. It is not the aim of this paper to analyze the performance of the standard trajectory tracking mechanism, but we would like to point out the overall behavior of the formation. Again, let us highlight that it is not the aim of this analysis to show the convergence of the optimization problem described in equation (8). The aim is to specify the conditions for which the shape of the formation remains stable (followers follow their desired position behind the virtual leader) using periodically replanned results of the problem in equation (8).

In an ideal state without perturbations due to dynamic obstacles, actuator uncertainties, and sensor uncertainties, the equality  $J_i(\Omega_i(\cdot; \cdot)^\circ) = 0$  holds. As shown for example in Figure 7, the value of the cost function is increased if a robot deviates from its desired position (the first term of equation (9) contributes) or if an obstacle or a neighbor is in close proximity to the robot (the second term of equation (9) contributes). Based on the values of the cost function, one can decide that a follower or a group of followers has broken away from the formation, and has to be considered as an independent object/sub-formation. An approach similar to the concept utilized for the formation navigation presented in Section 4.3 may be used for such a sub-formation to rejoin the group. This concept may be employed if it is enabled by the relative localization (the robot is still within the field of view of MAVs acting as hawk eyes) and by the communication range. In this case, the area around the desired position within the former formation has to be considered the target region. The sub-formation is then controlled using the approach presented in Section 4, where only the members of the unstuck group are considered in the convex hull representing the formation. If only a single robot is unstuck, the convex hull is reduced to a circle with radius equal to the detection radius  $r_s$ .

Theorem 5.4 and the related proof can be utilized for the convergence analysis, similarly to what was done for the static target region. Only the practical meaning of the perturbations  $D(k)$  is changed. The uncertainty of the actuators and the imprecise kinematic model are still included in the perturbations according to equation (24). Nevertheless, the prolonged total time to the goal given by the time difference  $T_2 - T_1$  is caused mainly by the movement of the former formation, which is followed by the new formation. To better understand the problem, let us split the

perturbations into two parts:  $D(k)$ , which represents the perturbations given by the imprecise actuators and model, and  $D_{drift}(k)$ , which includes the influence of the drift of the dynamic target region (the required place of the unstuck followers in the former formation). The equation  $D_{total}(k) = D(k) + D_{drift}(k)$  holds for the situation with the dynamic target region, whereas, in the analysis in the previous subsection, the equations  $D_{drift}(k) = 0$  and  $D_{total}(k) = D(k)$  hold.

During the MPC step with duration  $n\Delta t$ , the dynamic target region moves over the distance  $v_{target}n\Delta t$  in the worst case. The symbol  $v_{target}$  denotes the maximum speed of movement of the former formation, and therefore also the maximum speed of the dynamic target region. This displacement prolongs the expected time to the goal by the time difference  $v_{target}n\Delta t/v_{unstuck}$ , where  $v_{unstuck}$  is the maximal feasible speed of the unstuck formation/robot. For the dynamic target region, the requirement on the perturbations given by the imprecise actuators and model (for the static target region presented in equation (25)) can be expressed as

$$D(k) < n\Delta t - n\Delta t \frac{v_{target}}{v_{unstuck}} \quad (26)$$

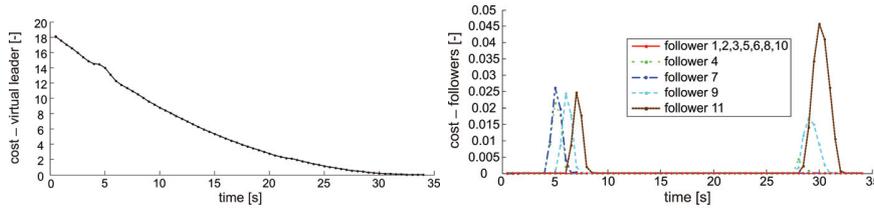
## 6. Experimental results

In this section, we demonstrate the performance of the method presented here, and we experimentally verify the theoretical results introduced in Section 5. The response of the planning and control mechanisms to the detected static and dynamic obstacles, and also to failures of neighbors in the formation, is shown in the addressed robotic scenarios. The experiment with real robots and the computational time analysis reflect the applicability of the system. Movies of simulations and experiments are attached to this paper and can be found at Saska (2013).

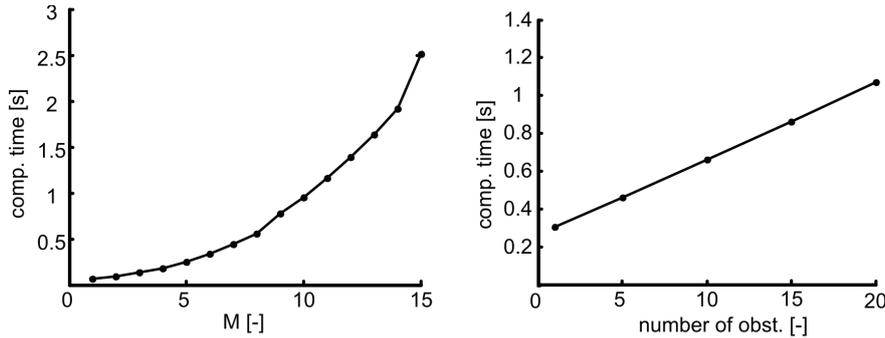
The presented results were obtained using the proposed algorithm with the following parameters:  $n = 2$ ,  $N = 4$ ,  $M = 6$ ,  $\alpha = 1$ ,  $\alpha_i = 1$ ,  $\beta_i = 1$ , and  $\Delta t = 0.25$  s. We employed sequential quadratic programming (SQP) (Nocedal and Wright, 2006) to solve the optimization problems used in the virtual leader trajectory planning and for the stabilization of followers. This solver provided the best performance among the available evaluated algorithms. However, any optimization method which is able to solve the optimization problems defined in this paper can be used.

The map of the environment, the position of the target region, and the desired shape of the formation are always known at the beginning of the missions in the experiments. The positions of dynamic obstacles are unknown.

In the first experiment presented in this section, a formation of 11 followers (eight UGVs and three MAVs) has to move into a target region through an environment with two overhead obstacles and one dynamic obstacle. See Table 3 for the parameters of the formation and Figure 6 for snapshots of the experiment. Figure 7 illustrates the progress of



**Fig. 7.** Progress of the values of the cost function used for the virtual leader’s trajectory planning, equation (6), and the cost function employed for stabilization of the followers in their desired positions behind the virtual leader, equation (9). The decrease in the values of equation (6) shows the convergence of the formation to the desired target. The deviations from the zero value of equation (9) are caused by the narrow passage (see Figure 6(b)) and by the dynamic obstacle (see Figure 6(f)).



**Fig. 8.** Computational demands of the method: influence of the number of obstacles and the number of transition points at the planning interval  $T_M$ . The mean computational time of one MPC planning step was obtained from 1000 runs of the algorithm.

**Table 3.** Curvilinear coordinates of followers within the formation used in the experiment presented in Figure 6.

$i$	1	2	3	4	5	6	7	8	9	10	11
$p_i$	1.5	3	0	0	0	0	0	2	2	4	4
$q_i$	0.5	1	0	2	0.7	-0.7	-2	0	2	0	2
$h_i$	5	4	4	0	0	0	0	0	0	0	0

the values of the cost function used for the virtual leader’s trajectory planning (equation (6)) and the values of the cost function employed for the stabilization of the followers (equation (9)).

The initial position of the formation and the trajectory obtained in the first planning loop of the presented MPC algorithm are shown in Figure 6(a). The snapshots in Figure 6(b) demonstrate the ability of the formation stabilization algorithm to autonomously modify the desired shape of the group if this is necessary due to restrictions given by the robots’ workspace. The pictures show the response of the followers’ planning algorithm to the narrow entrance and the consequent temporary shrinking of the formation.

In Figure 6(c) and 6(d), the formation passes by obstacles that verify the ability of the approach to keep direct visibility between team members. Although the UGVs could pass under the obstacles and the MAVs could fly over them (which would decrease the time to the goal), the planned trajectory leads around the obstacles to keep them outside the convex hull representing the formation. The UGVs may partly pass under the obstacles, with the aim to follow

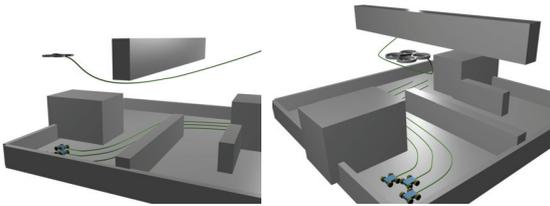
the trajectory that is as short as possible according to the optimization problem specified in equation (5).

A response of the trajectory planning algorithm to obstacles that suddenly appear is shown in Figure 6(e), where a new obstacle was detected. Finally, dynamic obstacle avoidance behavior of the method is shown in Figure 6(f). In this situation, the virtual leader’s planning algorithm could not respond to the movement of the obstacle behind the position of the virtual leader. Therefore, the followers are forced via the avoidance function in equation (9) to deviate temporarily from their desired positions within the formation.

The results of a statistical test of the performance of the algorithm are presented in Table 4. The aim is to show the reliability and the practical utilization of the method with different values of parameter  $M$ , whose proper setting is crucial for the deployment of the system. For the test, a set of 1000 positions for the target region was randomly generated in the free space on the right side of the workspace introduced in Figure 6(a). The algorithm was tested with different values of  $M$  for each of these configurations of the target, using Intel Core Duo CPU, 3.2 GHz, 4 GB RAM.

**Table 4.** The mean time to reach the target region and the success rate obtained from 1000 runs with random positions of the target region. Each set of experiments was performed with a different setting of the number of transition points at the planning interval  $T_M$ .

$M$	3	4	5	6	8	10
Time to goal [s]	51.2	45.8	39.6	34.5	32.8	32.6
Success rate [%]	42.1	86.9	98.2	99.6	100	100



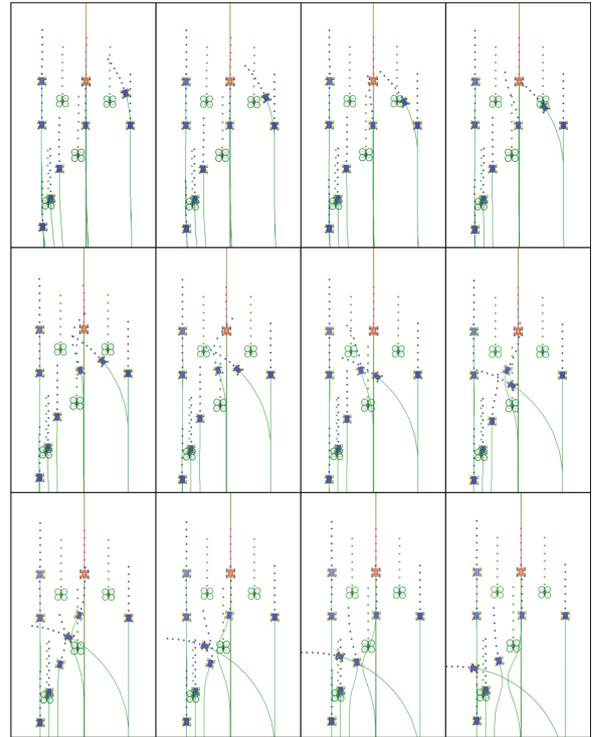
**Fig. 9.** An example of the trajectories followed by a formation of four followers (three UGVs and one MAV). The MAV follows the virtual leader, while it descends to avoid the top obstacle.

The simulations are counted as successful if the formation reached the desired target region without collisions and with relative visibility kept during the entire movement. The total time to reach the goal indicates the quality of the solutions.

The most time-consuming part of the proposed MPC approach is evaluation of the objective function in equation (6). This function is recalled in each iteration of the optimization process. The second term of the function, which implements the obstacle avoidance, represents a major contribution to the computational demands. In particular, two variables influence the computational complexity of the formation-driving system that is proposed here: the number of obstacles considered for the planning, and the dimension of the optimization vector. Figure 8 shows that the mean computational time linearly depends on the number of obstacles. This confirms the expectations, since the distance from the convex hull is computed separately for each of the obstacles only in the second term of equation (6). The distance-to-obstacle calculation is the most computationally intensive part of the algorithm.

The length of the optimization vector predominantly affects the number of iterations of the optimization process. The mean computational time of the planning process exponentially depends on the length of the optimization vector (see experimental results in Figure 8 and analysis of quadratic programming algorithms in Nocedal and Wright, 2006).

The simulation presented in Figure 9 shows the performance of the algorithm in the situation where an obstacle (the traverse beam under the ceiling) blocks the MAV follower from reaching the desired target region at the desired

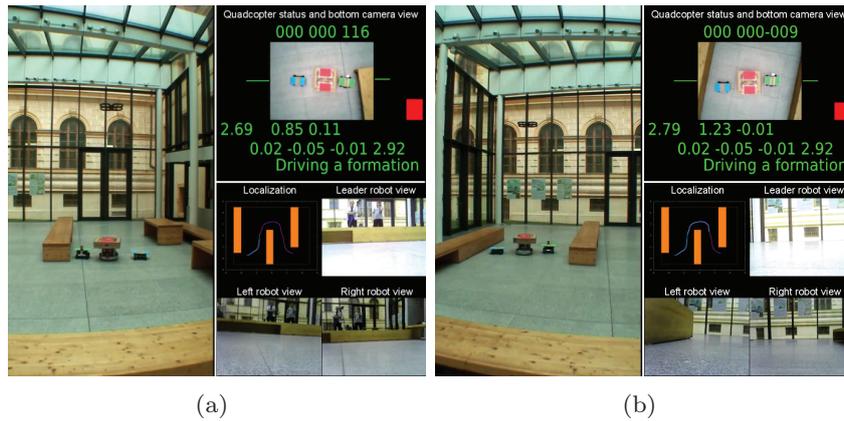


**Fig. 10.** A sequence of snapshots presenting the failure tolerance of the system by simulating a follower failure.

altitude ( $h_i$  coordinate). Similarly to the previous experiment, where the formation passed a narrow corridor, the formation is forced to change its desired shape temporarily.

The experiment presented in Figure 10 demonstrates the ability of the method to avoid collisions between the robots in the formation. To show this functionality, the failure of a follower (its steering was blocked) was simulated to show the failure tolerance and the robustness of the system. The snapshots show successful avoidance maneuvers of neighboring followers as a response to predictions of possible collisions (see the last part of equation (9) for details on the applied avoidance function).

In the real experiment presented in Figure 11, a formation of three ground robots and one helicopter has to move from its initial locations into the desired target region. The trajectory planning method presented in this paper was employed to practically verify the usefulness of the visual relative localization and consequently the stabilization of followers from flying robots. Two different UGVs, one G2Bot platform (Chudoba et al., 2006) (the bigger robot in Figure 11) and two MMP5 platforms are used in this experiment. To verify the concept of the proposed 3D formations, an Ar.Drone quadcopter is used as a flying follower. The MAV is equipped with a vision system to be able to follow the proposed hawk-eye approach. It carries a bottom monocular camera supplemented by a vision algorithm (Saska et al., 2012a) that is able to identify the location and the size of the color markers of the UGVs in the image.



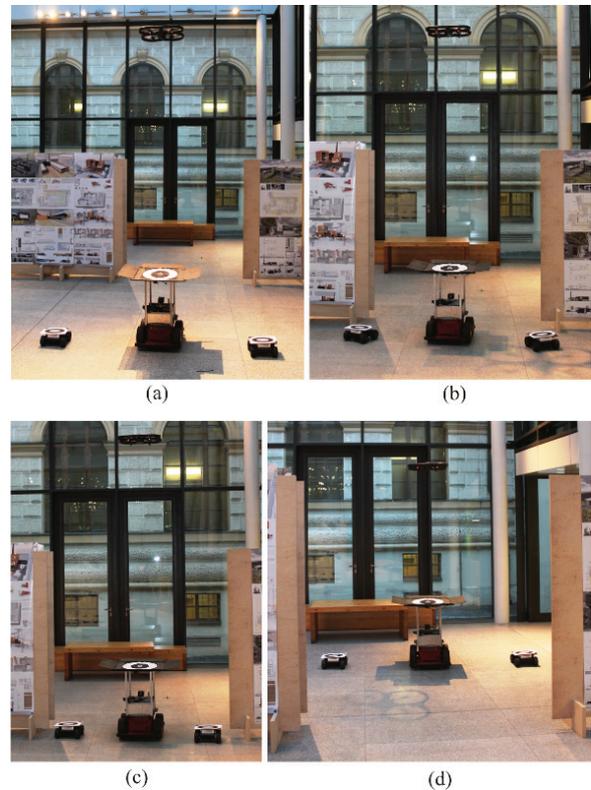
**Fig. 11.** Snapshots from the formation-driving experiment with three UGVs and one MAV following a virtual leader.

This information is used for the relative localization of all members of the formation. The estimated relative positions from such hawk eyes are sent to the UGV followers over a WiFi link as feedback to maintain the predefined formation shape.

Beside the pictures of the experiment, a GUI monitoring the formation deployment in the reconnaissance applications is shown in Figure 11, on the right side. The GUI shows pictures from the cameras carried by all followers and a schematic map of the environment. The MAV camera is primarily designated as the hawk eye. Additionally, it provides a general overview of the scene for the supervisor of the mission. The UGVs' cameras are employed for reconnaissance purposes. The actual plan of the virtual leader found by our approach and the history of the leader's movement are also depicted in the map. The position of the virtual leader is estimated from the odometry of the G2Bot follower, which is intentionally placed in the same position as the virtual leader.

The experiment in Figure 12 demonstrates the ability of the obstacle avoidance by temporarily shrinking the formation. In the experiment, the Pioneer 3-AT robotic platform with a mobile heliport, two MMP5 platforms and the Ar.Drone MAV act as followers. The positions of the outer UGVs within the formation are autonomously changed as a result of the multi-criteria cost function (the first and the second term of  $J_i(\cdot)$  contribute in an antagonistic way) to pass safely through the narrow corridor.

The onboard visual localization, which was used in the experiments, provides relative positions of robots at a frame rate of 10–30 Hz (Faigl et al., 2013). This is significantly less than the update rates provided by motion-capture systems (e.g. Vicon), which are often used for stabilization of MAV groups. Therefore, a MAV low-level stabilization and limits on the maximal speed of robots have to be employed with the onboard localization. Internal stabilization of the Ar.Drone platform based on optical flow obtained from the bottom camera and the inertial measurement unit were used in the experiments. For details, software implementation, and interfacing with Ar.Drone, see Krajník (2013).



**Fig. 12.** Experiment verifying formation-shrinking while it is driven through a narrow corridor.

The system presented in this paper enables stabilizing the heterogeneous formation up to a speed of 0.7 m/s. For the experimental evaluation presented here, the speed of the leader was limited to 0.3 m/s for safety reasons.

## 7. Conclusion

A control methodology developed for formation driving of 3D heterogeneous MAV–UGV formations stabilized via hawk-eye-like visual localization is presented in this paper.

A novel MPC scheme is introduced with an integrated obstacle-avoidance function that ensures direct visibility between MAVs and UGVs. Visibility between MAVs and UGVs is crucial for the top-view relative localization of the team members. This may act as an enabling technique for real-world deployment of formations of micro-scale robots. Our experiments show the performance of the method and verify its robustness in an environment with dynamic obstacles. In addition, the requirements for practical utilization of the method are specified by sound theoretical analyses. As a result of these analyses, we propose a simple mechanism to detect and tackle eventual violations of the convergence of the 3D formation's movement into the target region.

The main contributions of this paper from the perspectives of control, formation driving, and robotics in general are the following:

1. Top-view visual relative localization, which enables the deployment of teams of MAVs and simple ground robots in environments without any pre-installed global localization infrastructure.
2. The novel MPC approach with an additional planning horizon, which is crucial for incorporating global trajectory planning and local control. In addition, this approach enables the inclusion of constraints given by the top-view relative localization of heterogeneous formations.
3. An extended leader-follower concept with a novel representation of 3D formations that satisfies the requirements of direct visibility between the team members.

### Acknowledgements

Our special thanks go to Tom Duckett, Nicola Bellotto, and Sam Malone from the University of Lincoln for their careful proofreading and valuable comments and to Dušan M Stipanovic and Juan S Mejía from the University of Illinois at Urbana-Champaign for very useful advice concerning the theoretical issues of the paper. We would also like to thank the anonymous reviewers and the editors for their valuable comments.

### Funding

The work was supported by the Grant Agency of the Czech Republic (postdoc grant number P103-12/P756) and by MŠMT under project Kontakt II (number LH11053). The European Union supported this work within its Seventh Framework Programme project ICT-600623 'STRANDS'.

### References

- Abdessameud A and Tayebi A (2011) Formation control of VTOL unmanned aerial vehicles with communication delays. *Automatica* 47(11): 2383–2394.
- Alamir M (2006) *Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes (Lecture Notes in Control and Information Sciences, vol. 339)*. Berlin/Heidelberg: Springer.
- Barambones O and Etxebarria V (2000) Robust adaptive control for robot manipulators with unmodelled dynamics. *Cybernetics and Systems* 31(1): 67–86.
- Barfoot TD and Clark CM (2004) Motion planning for formations of mobile robots. *Robotics and Autonomous Systems* 46: 65–78.
- Beard R, Lawton J and Hadaegh F (2001) A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology* 9(6): 777–790.
- Bullo F, Cortés J and Martínez S (2009) *Distributed Control of Robotic Networks*. Princeton: Princeton University Press.
- Burdakov O, Doherty P, Holmberg K, et al. (2010) Relay positioning for unmanned aerial vehicle surveillance. *The International Journal of Robotics Research* 29(8): 1069–1087.
- Chao Z, Zhou SL, Ming L, et al. (2012) UAV formation flight based on nonlinear model predictive control. *Mathematical Problems in Engineering* 2012(1): 1–16.
- Chen J, Sun D, Yang J, et al. (2010) Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. *The International Journal of Robotics Research* 29: 727–747.
- Chudoba J, Mazl R and Preucil L (2006) A control system for multi-robotic communities. In: *ETFA 2006*.
- Das A, Fierro R, Kumar V, et al. (2003) A vision-based formation control framework. *IEEE Transactions on Robotics and Automation* 18(5): 813–825.
- Defoort M (2010) Distributed receding horizon planning for multi-robot systems. In: *IEEE International conference on control applications (CCA)*, pp. 1263–1268.
- Desai J, Ostrowski J and Kumar V (2001) Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation* 17(6): 905–908.
- Do KD and Lau MW (2011) Practical formation control of multiple unicycle-type mobile robots with limited sensing ranges. *Journal of Intelligent and Robotic Systems* 64(2): 245–275.
- Dong W (2011) Robust formation control of multiple wheeled mobile robots. *Journal of Intelligent and Robotic Systems* 62(3–4): 547–565.
- Dorigo M, Floreano D, Gambardella LM, et al. (2013) Swarmoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*. 20(4): 60–71.
- Dunbar W and Murray R (2006) Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica* 42(4): 549–558.
- Faigl J, Krajník T, Chudoba J, et al. (2013) Low-cost embedded system for relative localization in robotic swarms. In: *Proceedings of the IEEE International conference on robotics and automation*.
- Franco E, Magni L, Parisini T, et al. (2008) Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. *IEEE Transactions on Automatic Control* 53(1): 324–338.
- Fredslund J and Mataric M (2002) A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation* 18(5): 837–846.
- Garrido S, Moreno L and Lima PU (2011) Robot formation motion planning using fast marching. *Robotics and Autonomous Systems* 59(9): 675–683.
- Ghommam J, Mehrjerdi H, Saad M, et al. (2010) Formation path following control of unicycle-type mobile robots. *Robotics and Autonomous Systems* 58(5): 727–736.

- Hengster-Movrić K, Bogdan S and Draganjac I (2010) Multi-agent formation control based on bell-shaped potential functions. *Journal of Intelligent and Robotic Systems* 58(2): 165–189.
- Hess M, Saska M and Schilling K (2009) Application of coordinated multi vehicle formations for snow shoveling on airports. *Intelligent Service Robotics* 2(4): 205–217.
- Khalil H (2001) *Nonlinear Systems*. 3rd edn. Englewood Cliffs, NJ: Prentice Hall.
- Klančar G, Matko D and Blažič S (2011) A control strategy for platoons of differential drive wheeled mobile robot. *Robotics and Autonomous Systems* 59(2): 57–64.
- Krajník T (2013) ARDrone quadcopter in robotics research. Available at: <http://labe.felk.cvut.cz/~tkrajnik/ardrone/> (accessed 19 September 2013).
- Krajník T, Faigl J, Vonásek V, et al. (2010) Simple yet stable bearing-only navigation. *Journal of Field Robotics* 27(5): 511–533.
- Krajník T, Vonásek V, Fišer D, et al. (2011) AR-Drone as a platform for robotic research and education. In: Obdržálek D and Gottscheber A (eds) *Research and Education in Robotics: EUROBOT 2011*. Berlin/Heidelberg: Springer.
- Kumar V and Michael N (2012) Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research* 31(11): 1279–1291.
- Kushleyev A, Mellinger D and Kumar V (2012) Towards a swarm of agile micro quadrotors. In: *Proceedings of robotics: Science and systems*.
- Langer D, Rosenblatt J and Hebert M (1994) A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation* 10(6): 776–783.
- Lawton J, Beard R and Young B (2003) A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation* 19(6): 933–941.
- Liu C, Chen WH and Andrews J (2011) Piecewise constant model predictive control for autonomous helicopters. *Robotics and Autonomous Systems* 59(7–8): 571–579.
- Liu Y and Jia Y (2012) An iterative learning approach to formation control of multi-agent systems. *Systems & Control Letters* 61(1): 148–154.
- Mastellone S, Stipanovic DM, Graunke CR, et al. (2008) Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research* 27(1): 107–126.
- Mayne DQ, Rawlings JB, Rao CV, et al. (2000) Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789–814.
- Mellinger D, Michael N and Kumar V (2012) Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research* 31(5): 664–674.
- Michael N and Kumar V (2009) Planning and control of ensembles of robots with non-holonomic constraints. *The International Journal of Robotics Research* 28(8): 962–975.
- Min HJ and Papanikolopoulos N (2012) Robot formations using a single camera and entropy-based segmentation. *Journal of Intelligent and Robotic Systems* (1): 1–21.
- Nocedal J and Wright SJ (2006) Numerical optimization. In: Glynn P and Robinson SM (eds) *Springer Series in Operations Research*. New York, NY: Springer.
- No TS, Kim Y, Tahk MJ, et al. (2011) Cascade-type guidance law design for multiple-UAV formation keeping. *Aerospace Science and Technology* 15(6): 431–439.
- Olfati-Saber R (2006) Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control* 51: 401–420.
- Ren W (2008) Decentralization of virtual structures in formation control of multiple vehicle systems via consensus strategies. *European Journal of Control* 14: 93–103.
- Saffarian M and Fahimi F (2009) Non-iterative nonlinear model predictive approach applied to the control of helicopters' group formation. *Robotics and Autonomous Systems* 57(6–7): 749–757.
- Saska M (2013) Movies of simulations and experiments of the formation driving approach. Available at: <http://imr.felk.cvut.cz/formationsijrr/> (accessed 19 September 2013).
- Saska M, Krajník T and Přeučil L (2012a) Cooperative micro UAV-UGV autonomous indoor surveillance. In: *IEEE conference on systems, signals and devices (SSD)*.
- Saska M, Krajník T, Faigl J, et al. (2012b) Low cost MAV platform AR-Drone in experimental verifications of methods for vision based autonomous navigation. In: *IEEE/RSJ International conference on intelligent robots and systems (IROS)*.
- Saska M, Vonásek V and Přeučil L (2011) Roads sweeping by unmanned multi-vehicle formations. In: *Proceedings of the IEEE International conference on robotics and automation (ICRA)*.
- Saska M, Vonásek V, Krajník T, et al. (2012c) Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach. In: *IEEE/RSJ International conference on intelligent robots and systems (IROS)*.
- Shin J and Kim H (2009) Nonlinear model predictive formation flight. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 39(5): 1116–1125.
- Sira-Ramirez H and Castro-Linares R (2010) Trajectory tracking for non-holonomic cars: A linear approach to controlled leader-follower formation. In: *IEEE conference on decision and control (CDC)*.
- Stipanović DM, Hokayem PF, Spong MW, et al. (2007) Cooperative avoidance control for multi-agent systems. *Journal of Dynamic Systems, Measurement, and Control* 129: 699–707.
- Tanner H and Christodoulakis D (2007) Decentralized cooperative control of heterogeneous vehicle groups. *Robotics and Autonomous Systems* 55(11): 811–823.
- Turpin M, Michael N and Kumar V (2011) Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots* 33(1–2): 143–156.
- Xiao F, Wang L, Chen J, et al. (2009) Finite-time formation control for multi-agent systems. *Automatica* 45(11): 2605–2611.
- Yang H, Zhu X and Zhang S (2010) Consensus of second-order delayed multi-agent systems with leader-following. *European Journal of Control* 16: 188–199.
- Zhang X, Duan H and Yu Y (2010) Receding horizon control for multi-UAVs close formation control based on differential evolution. *Science China Information Sciences* 53: 223–235.

## Appendix: Low-level controller designed for Ar. Drone

In the kinematic model described in Section 3, it is assumed that the MAVs can follow a trajectory containing segments with a given curvature. Further, we assume that MAVs

are controlled using forward speed  $v$ , curvature  $K$ , and ascent  $w$ . We used the Ar.Drone quadcopter, which allows changes in the speed of its rotors and consequently its pitch, yaw, and roll angles (see Figure 1(b) for the definition of the angles).

The forward speed of the drone can be controlled by changing the pitch  $\theta$ . Let us assume the simplified model of the quadcopter depicted in Figure 13. Force  $F$  is generated by the rotors.  $F_g$  denotes the gravitational force. The forward force is then  $F_o = -F_g \tan \theta$ . Assuming that the drone moves slowly, we can use an approximation  $F_o = c_0 v$ , where  $c_0$  is a constant that may be simply identified experimentally. The forward speed of the drone is then controlled by changing pitch  $\theta$ :

$$\tan \theta = -\frac{c_0 v}{F_g} \quad (27)$$

When the drone moves along a circular segment, the centrifugal force has to be compensated for by changing roll  $\eta$ . The centrifugal force can be expressed as  $F_c = mv^2/r$ , where  $m$  is the weight of the drone and  $r = 1/K$  is the radius of the circular segment with curvature  $K$ . The centrifugal force is compensated for by the lateral force  $F_d = -F_c$ .  $F_d$  is controlled by changing roll as  $F_d = -F_g \tan \eta$ , which

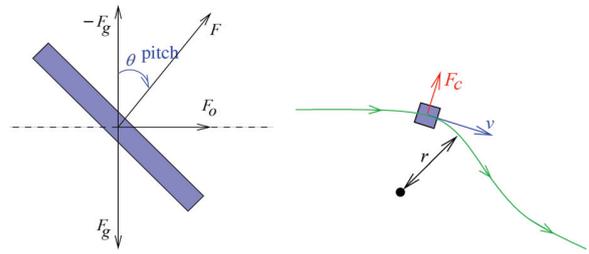


Fig. 13. Simplified model of the quadcopter.

gives  $-F_g \tan \eta = mv^2/r$ . The lateral speed is then controlled by

$$\tan \eta = -\frac{v^2 K}{g} \quad (28)$$

When the pitch and roll angles are small, the approximation  $\tan x \sim x$  can be used in (27) and (28). To control the drone along a trajectory with the given curvature  $K$  and known forward speed  $v$ , pitch  $\theta$  and roll  $\eta$  are controlled. Further details on Ar.Drone control can be found in Krajník et al. (2011).

# B

---

KEY ARTICLE [2] - JOURNAL OF INTELLIGENT AND ROBOTIC  
SYSTEMS 2014

---

©[2014] Springer. This article was published in Journal of Intelligent and Robotic Systems: **M. Saska**, T. Krajník, V. Vonásek, Z. Kasl, V. Špurný, and L. Preucil: Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups, 2014.



# Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups

Martin Saska · Tomáš Krajník · Vojtěch Vonásek ·  
Zdeněk Kasl · Vojtěch Spurný · Libor Přeučil

Received: 31 August 2013 / Accepted: 23 September 2013 / Published online: 12 October 2013  
© Springer Science+Business Media Dordrecht 2013

**Abstract** A fault-tolerant method for stabilization and navigation of 3D heterogeneous formations is proposed in this paper. The presented Model Predictive Control (MPC) based approach enables to deploy compact formations of closely cooperating autonomous aerial and ground robots in surveillance scenarios without the necessity of a precise external localization. Instead, the proposed method relies on a top-view visual relative localization provided by the micro aerial vehicles

flying above the ground robots and on a simple yet stable visual based navigation using images from an onboard monocular camera. The MPC based schema together with a fault detection and recovery mechanism provide a robust solution applicable in complex environments with static and dynamic obstacles. The core of the proposed leader-follower based formation driving method consists in a representation of the entire 3D formation as a convex hull projected along a desired path that has to be followed by the group. Such an approach provides non-collision solution and respects requirements of the direct visibility between the team members. The uninterrupted visibility is crucial for the employed top-view localization and therefore for the stabilization of the group. The proposed formation driving method and the fault recovery mechanisms are verified by simulations and hardware experiments presented in the paper.

---

M. Saska (✉) · V. Vonásek · Z. Kasl ·  
V. Spurný · L. Přeučil  
Department of Cybernetics, Faculty of Electrical  
Engineering, Czech Technical University in Prague,  
Technická 2, 166 27 Prague 6, Czech Republic  
e-mail: saska@labe.felk.cvut.cz

V. Vonásek  
e-mail: vonasek@labe.felk.cvut.cz

Z. Kasl  
e-mail: kaslzden@fel.cvut.cz

V. Spurný  
e-mail: spurnvoj@fel.cvut.cz

L. Přeučil  
e-mail: preucil@labe.felk.cvut.cz

T. Krajník  
Lincoln Centre for Autonomous Systems,  
Faculty of Science, University of Lincoln,  
Brayford Pool, Lincoln, LN6 7TS,  
Lincolnshire, UK  
e-mail: tkrajnik@lincoln.ac.uk

**Keywords** Mobile robots · Micro aerial vehicles ·  
Formation driving · Fault detection  
and recovery · Model predictive control ·  
Leader-follower · Trajectory planning

## 1 Introduction

Integration of fault detection and recovery mechanisms into unmanned aerial systems is crucial

for improving their robustness and to enable their deployment in large closely cooperating teams. The identification of a sensor or actuator fault or even a failure of a team member makes possible to adapt the group behaviour, keeping the system operational with limited capabilities. This approach is especially appealing for formations and swarms of autonomous aerial, but also terrestrial vehicles, where the possibility of redundancy in robots' deployment is one of the key properties.

In this paper, a scenario of multi-robot surveillance is investigated. In the mission, a formation of autonomous vehicles has to repeatedly drive through a workspace in a phalanx to cover a large operating space. We propose to employ heterogeneous teams of autonomous micro-scale vertical take-off and landing vehicles (so called Micro Aerial Vehicles—MAVs) and Autonomous Ground Robots (UGVs). This allows us to consider their deployment in missions, which are impossible for solely MAVs or UGVs teams or in which these teams would not be efficient.

The MAVs can reach locations inaccessible by the UGVs. Beyond, they may provide a top view survey of the scene, which gives an important overview for human supervisors. On the contrary, the UGVs may operate in workspaces constrained by obstacles (e.g. in abundant vegetation). They can carry much heavier payload, which allows to use more powerful sensors. The UGVs have larger operational range and they may even provide an additional power source for the MAVs through a mobile heliport. These aspects attract us to take advantage of both platforms and to employ a heterogeneous MAVs-UGVs team. Besides, the co-existence of ground and flying robots can provide efficient solutions of fundamental formation driving problems, as is a precise and reliable relative localization of team members closely cooperating together. This approach reduces probability of collisions within the robotic group.

Usually, robots in reconnaissance and surveillance missions may not rely on pre-installed precise global localization infrastructures and commonly available systems (as GPS) lack required precision for control of compact formations. Besides, GPS lacks sufficient reliability mainly in

urban and indoor environments. The proposed formation driving approach is suited for an on-board visual relative localization. The employed localization system uses simple light-weight cameras mounted on MAVs and identification patterns placed on UGVs and MAVs. The distance between the vehicles is then provided due to the known size of the patterns. Details on the visual based relative localization together with description of its precision and reliability is provided in [1]. With this top-view concept, one may better tackle the problem of loss of direct visibility that frequently occurs in the visual relative localization of ground robots operating in a workspace with obstacles. The possibility of team members' relative localization from above increases precision and reliability of the localization and brings another perspective to see the scene by operators supervising the mission.

Beyond the visual relative localization of individual robots, we propose to use a simple vision based technique also for the formation navigation in the environment. The presented formation driving method relies on a navigation approach called GeNav [2]. GeNav method uses features detected in images that are gathered by a monocular camera carried by a leader of the formation. This very simple method enables to robustly navigate the group along a pre-learned path consisting of a sequence of straight segments (a proof of stability of this method can be found in [2]).

The combination of the top-view relative localization and the visual navigation provides a light-weight, low-cost, easy-to-deploy and efficient solution, which may act as an enabling technique for an extensive utilization of simple micro-scale robots. This paper is focussed on theoretical and technical aspects of the formation driving mechanism suited for the real-world deployment of autonomous robots under the GeNav navigation and the top-view localization, while technical details on the visual relative localization are available in [3] and the GeNav navigation in [2]. In addition, the paper addresses issues of the fault identification and recovery to increase robustness of the method. A mechanism to detect (and correct if possible) a malfunction of a single robot as well

as an inadvisable breakup of the group is proposed and experimentally verified.

## 2 State of the Art

The research endeavor in the formation driving community is aimed mainly at tasks of formation stabilization [4–6] and formation following a predefined path [7–10]. For example in [4], the task of formation stabilization and its convergence into a desired pattern is tackled for formations with communication delays. In [5], a multi-agent control system using artificial potential based on bell-shaped functions is proposed. The work in [6] employs a distributed iterative learning scheme for solving the formation control with switching strategy in the virtual structure and virtual leader-follower schemes.

The path following problem is tackled by designing a nonlinear formation control law based on the virtual structure approach via propagation of a virtual target along the path in [8]. In [7], the path following is investigated for groups of robots with limited sensing ranges. In [9], according to the leader-follower concept, the leader robot is forced to follow a given path, while the followers track the leaders's path with a fixed time delay. In [10], beyond the trajectory tracking, a possibility of an autonomous design of geometric pattern of the desired formation is discussed.

Beside the methods of the formation driving for UGVs, we should mention few approaches designed for UAVs [11–14]. In [11], the formation stabilization and keeping in the desired shape are treated as a dynamic 3D tracking problem, where the relative geometry of multiple UAVs is kept via a cascade-type guidance law under the leader-follower concept. A leader-follower approach for stabilization of helicopter's formations using a nonlinear model predictive control is proposed in [12] and optimized for on-line embedded solution enabling a response to the fast dynamic of UAVs in [13]. In [14], the formation stabilization of vertical take-off and landing unmanned aerial vehicles in presence of communication delays is

addressed. Finally, let us mention work in [15] aimed at stabilization of a heterogeneous formation of UAVs above UGVs in circular orbits.

The above mentioned techniques are suited for utilization of robots under a precise external global localization system (for example approaches [13] are verified with the VICON system), for UGV formations they often rely on a dead reckoning with its cumulative error [8] or they provide theoretical solutions verified only by simulations [4, 5, 7, 9–12, 14], where a known position of robots may be assumed. In our work, the necessity of utilization of on-board systems for robots' localization and navigation is inherently included in the essence of the formation driving approach. The stabilization of the robot in the team is suited for requirements of available robust localization and navigation techniques, which enables its utilization in real-world scenarios.

The fault detection and recovery is an important and actual topic in the UAV (MAV) control nowadays due to the recent boom in the deployment of small unmanned vehicles. The fault detection techniques designed for a single vehicle can be divided into two categories: model-free and model-based. The model-free methods are based on analysis of the signal from sensors and do not rely on the model of the underlying system. As an example, let us mention an approach using neural networks to perform the signal analysis in order to acquire the information about a fault [16].

More frequently, methods for fault detection in unmanned aerial systems employ the model-based approach. They utilize residuals (difference between the sensor readings and expected values derived from a model of the monitored system) for the detection of occurrence of a fault. An example of an actuator fault diagnostic system designed for the nonlinear model of mini-quadrotors is available in [17]. Deployment of a methodology for actuator and sensor fault detection in an autonomous helicopter is presented in [18]. An autonomous actuator fault recovery mechanism based on an incorporation of a post-fault model of the actuator is proposed in [19]. This approach is extended for response to multiple actuator faults in [20].

Beyond the actuator faults in a single vehicle, this paper deals with the fault detection and recovery in the formation control. This problem is addressed for formations of terrestrial robots in [21]. In the paper, the formation is represented as a cellular automaton, where each formation member is represented as an individual cell, and the formation recovery is realized through a distributed auction-based mechanism. A fault tolerant approach designed for formations of quadrotor UAVs is presented in [22]. The virtual structure based method is used for the formation trajectory planning, while the fault recovery is realized by a replanning to be able to respond to a failure of a formation member. The most related work to the proposed paper is published in [23] and [24], where a vision-based relative position estimation for a team of UAVs is used in case of detection of fault in one of the onboard inertial systems. Although, the team of UAVs is not originally coordinated as a formation, a formation driving mechanism is employed in order to view the same scene from two UAVs (the faulty UAV and a faultless UAV used for its assistance) at the same time. Finally, let us mention two examples of the fault recovery in robotic swarms. In [25], an immune system reaction is employed. It solves the fault recovery problem through an isolation of the faulty robot from the swarm by neighbouring robots to protect the entire group. A fault detection method inspired by a light-based communication of fireflies, which spontaneously synchronize their rhythmic flashes, is presented in [26]. The method is based on analysing of anomalies from the synchronized light pulses of robots equipped with LEDs and light detectors.

Here, our contribution is an approach being able to detect faults in formation driving of heterogeneous MAVs-UGVs groups stabilized under the top view localization. Beyond, we aim at re-coupling of inadvertently splitted formations caused by a fault in the system, but also by surrounding environment.

In our method, we rely on the Model Predictive Control (MPC) to be able to involve constraints imposed by the inter vehicle relations (shape of the formation feasible for the top-view relative localization), by vehicles (mobility constraints), by obstacles (environment constraints) and by the

employed GeNav navigation of the entire group (straight line segments of the desired path) into the formation driving. The MPC approach is often used for stabilizing nonlinear systems with control constraints. In [12] and [13] it was shown, that the computational power of microprocessors available onboard of unmanned helicopters enables to employ MPC techniques also for the formation control of such a high dynamic systems, similarly as it is proposed here.

For descriptions and a general survey of MPC methods see [27–30] and references reported therein. In the formation driving, researchers take advantage of MPC mainly to respond to changes in dynamic environment [12, 13, 31–33]. In [31], authors introduce a new cost penalty in MPC optimization to guarantee a simple obstacle avoidance. Decentralized receding horizon motion planner introduced in [32] is developed for coordination of UGVs based on a motion planning independent to neighbors. The trajectory tracking mechanism developed in [33] is based on integration of a differential evolution algorithm into the MPC concept.

In our approach, we go beyond these papers in the following aspects. We apply the MPC method for the stabilization of the formation with included requirements of the top-view relative localization, which could be an enabling technique for deployment of heterogeneous MAVs-UGVs teams outside the laboratories. We present a novel obstacle avoidance function with a simple and effective representation of the 3D formation included. This approach provides a robust solution of the formation driving in environments with dynamic obstacles. Our formation driving method is designed for the purpose of simple yet stable visual navigation developed in [2], which is well suited for the reconnaissance and surveillance missions in environments without precise external localization. Beside the dynamic obstacles avoidance, the proposed method provides an inter-vehicle avoidance, which is crucial for failure tolerance of the system. All these behaviours and abilities are numerically and experimentally verified at the end of this paper.

This paper extends our previous work [34], where the basic formation driving mechanism based on the visual relative localization was

introduced. The main extension consists in the fault diagnosis and recovery mechanism that is introduced in Fig. 3 and described in details in Section 5. Beyond, results of new experiments and simulations verifying the proposed concept are published in Section 6.

### 3 Problem Statement and Preliminaries

Development of method presented in this paper is motivated by reconnaissance applications, where a team of robots has to autonomously follow a desired path given by a supervising expert. During the movement along the path, the robots have to keep a formation suited for the mission requirements. The robots can form a searching phalanx (a line formation) to be able to search for victims or intruders in large areas or a compact fleet of vehicles can be used for transportation purposes. We assume that one robot of the group (UGV or MAV), called GeNav leader in this paper, is capable of autonomous navigation along such a path. We will employ a navigation system based on detection of SURF features in an image provided by an on-board camera. The system was developed for navigation of a single UGV robot [2] and later extended for a quadcopter [35]. This system (called as GeNav) is suited for guidance of robots along path that consists of a sequence of straight segments. Its precision ( $\sim 20$  cm) and reliability enables a robust navigation of a single robot, but it is not sufficient for the coordination of the robots within a compact formation with small relative distances between robots.

Beside the GeNav leader, we assume to use a group of simple UGV followers without any on-board sensors for their localization and a group of MAV followers (quadrotors) equipped with a bottom camera and a system for visual relative localization [3]. This system provides information on the relative position between the camera of MAVs and center of an identification pattern. The identification patterns are carried by all UGVs and MAVs except the one flying in the highest altitude. The precision of the employed visual relative localization system ( $\sim 1$  cm) is sufficient for the formation stabilization in the desired shape. We assume that the shape is designed to satisfy the

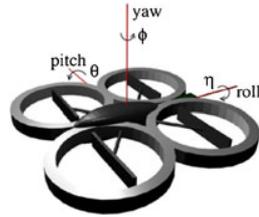
condition that all robots, except the MAV flying in the highest altitude, are in the field of view of at least one bottom camera mounted on an MAV.

Now, let us describe preliminaries important for description of the method, in which the heterogeneous 3D formation of MAVs and UGVs has to follow the desired path, while requirements of the formation driving and the top-view relative localization are satisfied. It means that (1) the movement of the formation has to be smooth also in the unsmooth connections of straight path segments, where the GeNav leader is turning around on the spot, and (2) the direct visibility between the vehicles have to be kept during the formation deployment.

Let  $\psi_j(t) = \{x_j(t), y_j(t), z_j(t), \varphi_j(t)\}$ , where  $j \in \{GL, VL, 1, \dots, nf\}$ , denote configurations of the GeNav leader *GL*, a virtual leader *VL*, and *nf* followers at time *t*. The GeNav leader is equipped with the on-board visual navigation to follow the pre-learned path segments. It is positioned in front of the formation and it is used as a reference point for the coordinate system used by the top-view relative localization. Whereas, the virtual leader is a reference point for the proposed formation driving technique. Virtual leader is initially placed in the same position and orientation as the GeNav leader. Using the trajectory following approach described in Section 4.3, it keeps the same position as GeNav leader except the deviation caused by obstacles that could break the top view localization or to cause collisions. Significant deviation of *GL* and *VL* positions can be also seen in connections of line segments of the desired path. In these points, the path is not feasible for the formation of nonholonomic robots, which forces the virtual leader to temporarily leave the path to be able to follow a smooth trajectory feasible for the formation.

The Cartesian coordinates  $x_j(t)$ ,  $y_j(t)$  and  $z_j(t)$  define positions  $\bar{p}_j(t)$  of all robots (leaders and followers) and  $\varphi_j(t)$  denotes their heading. Both MAVs and UGVs (except the robot assigned as the GeNav leader) are denoted as followers in the presented approach. For the MAVs, the heading  $\varphi_j(t)$  becomes directly the yaw (see Fig. 1 for the coordinate system of MAVs). Roll together with pitch do not need to be included in the kinematic model employed in MPC, but they depend on the

**Fig. 1** MAV coordinate system



type of utilized MAVs as shown for a quadrotor in [36].

Let us assume that the environment contains a finite number  $n_0$  of compact obstacles. The obstacles can be static (as part of a known map) or dynamic and unknown (detected during the formation movement by on-board sensors). These updates of the map are shared by all robots via a Wi-Fi communication. A follower or even more followers of the formation can become dynamic obstacles if deviating from their desired positions as demonstrated in the failure tolerance simulation in Section 6.

The kinematics for any robot  $j$  in 3D is described by the simple nonholonomic kinematic model:  $\dot{x}_j(t) = v_j(t) \cos \varphi_j(t)$ ,  $\dot{y}_j(t) = v_j(t) \sin \varphi_j(t)$ ,  $\dot{z}_j(t) = w_j$  and  $\dot{\varphi}_j(t) = K_j(t)v_j(t)$ , where feed-forward velocity  $v_j(t)$ , curvature  $K_j(t)$  and ascent velocity  $w_j(t)$  represent control inputs denoted as  $\bar{u}_j(t) = \{v_j(t), K_j(t), w_j(t)\}$ . We assume that UGVs operate in a flat surface and that  $z_j(\cdot) = 0$  and  $w_j(\cdot) = 0$  for each of the UGVs. In case of MAVs,  $v_j(\cdot)$ ,  $K_j(\cdot)$  and  $w_j(\cdot)$  values are inputs for the low level controller, as shown in [36].

Let us now describe a discretization of the kinematic model as it is used in the proposed formation driving with the model predictive trajectory following included. Let us define a time interval  $[t_0, t_{\text{end}}]$  consisting of a sequence of elements of increasing times  $\{t_0, t_1, \dots, t_{\text{end}-1}, t_{\text{end}}\}$ , such that  $t_0 < t_1 < \dots < t_{\text{end}-1} < t_{\text{end}}$ . We will refer to  $t_k$  using its index  $k$  in this paper. For the model predictive planning, the control inputs are held constant over each time interval  $[t_k, t_{k+1})$ , where  $k \in \{0, \dots, \text{end}\}$ . We will call the points at which the control inputs change as *transition points*. By integrating the kinematic model over these

intervals, the following discretized model may be obtained:

if  $K_j(k+1) \neq 0$ :

$$x_j(k+1) = x_j(k) + \frac{1}{K_j(k+1)} \times [-\sin(\varphi_j(k)) + \sin(\varphi_j(k) + K_j(k+1)v_j(k+1)\Delta t)],$$

$$y_j(k+1) = y_j(k) - \frac{1}{K_j(k+1)} \times [-\cos(\varphi_j(k)) + \cos(\varphi_j(k) + K_j(k+1)v_j(k+1)\Delta t)],$$

$$z_j(k+1) = z_j(k) + w_j(k+1)\Delta t$$

$$\varphi_j(k+1) = \varphi_j(k) + K_j(k+1)v_j(k+1)\Delta t$$

and if  $K_j(k+1) = 0$ :

$$x_j(k+1) = x_j(k) + v_j(k+1) \cos(\varphi_j(k)) \Delta t,$$

$$y_j(k+1) = y_j(k) + v_j(k+1) \sin(\varphi_j(k)) \Delta t,$$

$$z_j(k+1) = z_j(k) + w_j(k+1)\Delta t$$

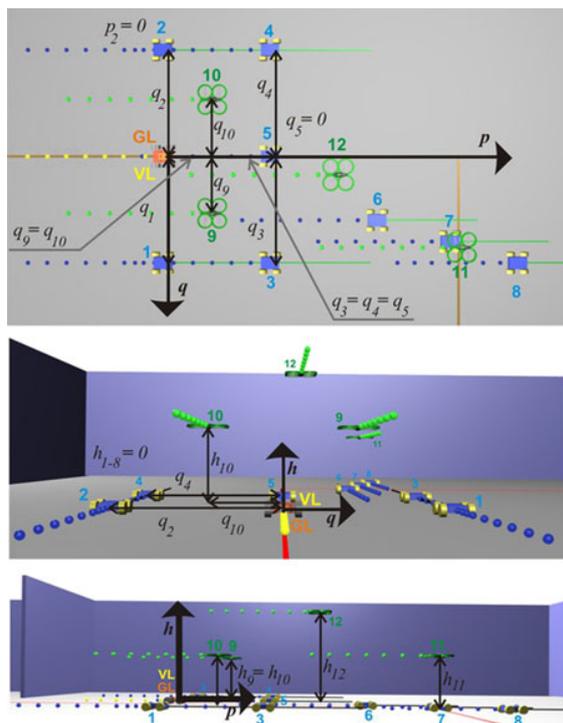
$$\varphi_j(k+1) = \varphi_j(k), \quad (1)$$

where  $x_j(k)$ ,  $y_j(k)$  and  $z_j(k)$  are the rectangular coordinates and  $\varphi_j(k)$  the heading angle at the transition point with index  $k$ .  $\Delta t$  is a sampling time, which is uniform in the whole interval  $[t_0, t_{\text{end}}]$ . The control inputs  $v_j(k+1)$ ,  $K_j(k+1)$  and  $w_j(k+1)$  are constant between transition points with index  $k$  and  $k+1$ .

As mentioned in the problem statement, we assume a heterogeneous 3D formation of a given shape, which satisfies the requirements given by the formation driving and the top-view localization: (1) robots are in a safe relative distance; (2) each robot, except the MAV flying in the highest altitude, is observed by at least one MAV. In this paper, the shape of the entire formation is maintained with a leader-follower technique derived from the approach [37], which was designed for formations of UGVs working in a planar

environment. For the heterogeneous MAVs-UGVs formations, we have extended the notation from [37] to 3D as visualized in Fig. 2. Besides, we have extended the technique in [37], which is designed for following smooth splines (continuity of second-order is required), for utilization of paths consisting of straight line segments, which are required by the GeNav navigation.

In our method, both types of followers, MAVs and UGVs, follow the trajectory of the virtual leader in distances defined in  $p, q, h$  curvilinear coordinate system. The position of each follower  $i$  is uniquely determined by states  $\psi_{VL}(t_{p_i})$  in travelled distance  $p_i$  from the actual position of the virtual leader along the virtual leader’s trajectory, by offset distance  $q_i$  from the trajectory in perpendicular direction and by elevation  $h_i$  above the trajectory.



**Fig. 2** The desired shape of the formation described in curvilinear coordinates

behind its actual position. To get states of follower  $i$  in rectangular coordinates, states of the virtual leader at time  $t_{p_i}$ , which is  $\psi_{VL}(t_{p_i}) = \{x_{VL}(t_{p_i}), y_{VL}(t_{p_i}), z_{VL}(t_{p_i}), \varphi_{VL}(t_{p_i})\}$ , have to be shifted with vector  $V(t_{p_i})$  as follows:

$$\psi_i(t) = \psi_{VL}(t_{p_i}) + V(t_{p_i}). \tag{2}$$

The vector  $V(t_{p_i})$  consists of four components:  $V(t_{p_i}) = (-q_i \sin(\varphi_L(t_{p_i})), q_i \cos(\varphi_L(t_{p_i})), h_i, 0)$ .

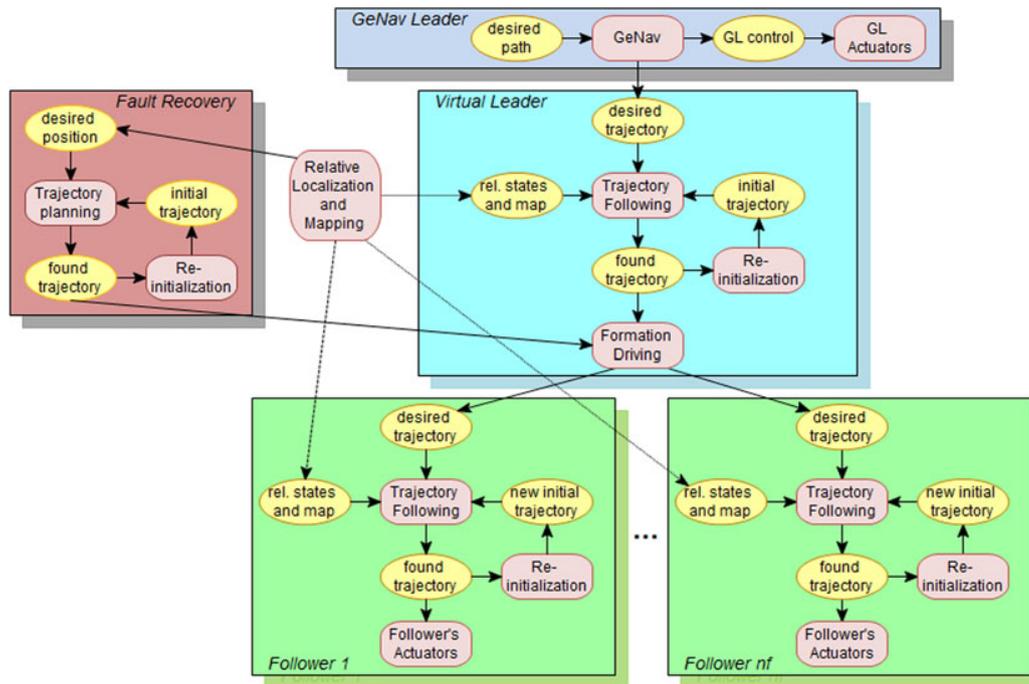
### 4 Integrated Trajectory Planning and Formation Stabilization

#### 4.1 Method Overview

The system designed for the stabilization of heterogeneous MAVs-UGVs formations is divided into four main blocks as you can see in the scheme depicted in Fig. 3. The first block, *GeNav Leader*, is responsible for navigation of the entire formation in the environment. It provides control inputs for the GeNav leader based on image features gained by an onboard camera. The GeNav method enables to navigate a robot or a group of robots along a pre-learned path consisting of straight segments. The requirements on the piecewise straight desired path is important for stability of the method as analysed in [2].

From the formation stabilization perspective, an important output of the *GeNav Leader* module is a prediction of GeNav leader’s states. For the prediction, it is assumed that the GeNav leader follows the desired path without any perturbation in both, the desired speed and the position on the path. The perturbations, which occur in real robotic systems, will be diminished by the presented receding horizon control technique. The predicted trajectory, which consists of  $n$  states derived with constant sampling time  $\Delta t$ , acts as an input of the *Virtual Leader* block.

This part is important for avoidance of obstacles that could affect the relative localization within the group or that could collide with robots of the formation. Besides, it enables to follow the GeNav leader in connections of straight line segments of the desired path. In the *Virtual*



**Fig. 3** Schema of the complete planning and control system

Leader part, the *Trajectory Following* block provides control inputs for the virtual leader, which are feasible for the entire formation and respect the requirements of the top-view relative localization via the model of the formation. In the straight segments of the desired path, the trajectory found by the *Trajectory Following* block follows the desired trajectory with minimal deviation and it is only employed to diminish possible perturbations. A significant difference between the desired and found trajectory occurs mainly due to appearing obstacles or near to line segment connections. Details on the trajectory following mechanism with emphasis on incorporation of the 3D heterogeneous formation stabilized under the top-view localization are presented in Section 4.3.

The resulting trajectory obtained in the *Trajectory Following* block is described by a sequence of configurations of the virtual leader  $\psi_L(k)$ , where  $k \in \{1, \dots, N\}$ , and by constant control inputs applied in between the transition points. According the MPC concept, only a portion of

the computed control actions is applied on the interval  $(t_0, t_0 + n\Delta t)$ , known as the receding step. This process is then repeated on the interval  $(t_0 + n\Delta t, t_0 + N\Delta t + n\Delta t)$  as the finite horizon moves by *time steps*  $n\Delta t$ , yielding a state feedback control scheme strategy. The unused part of the trajectory can be employed for re-initialization of the planning process in each planning step, since the plan of the formation between two consequent steps is usually changed only slightly. To summarize this,  $n$  is number of transition points in the part of the planning horizon, which is realized by robots in each planning step, and  $N$  is the total number of transition points in the planning horizon.

In the proposed formation driving system, the trajectory obtained in the *Trajectory Planning* block is used as an input for the *Formation Driving* module, which transforms the plan to desired configurations of followers (using Eq. 2). The core of the third main block, which is multiplied for MAVs and UGVs followers, is also

the *Trajectory Following* module. This part is responsible for avoiding of impending collisions with obstacles or other members of the team and it corrects deviations from the desired trajectory provided by the virtual leader. In real applications with dynamic obstacles and disturbances caused by the imprecise model of sensors and actuators, the desired trajectories provided by the *Formation Driving* cannot be directly applied for control of particular followers. They have to be adapted to ensure the stability of the group and non-collision movement. Similarly as in the leader’s trajectory following, the unused part of the found trajectory can be employed for the initialization of the planning process.

The fourth main block, labelled as *Fault Recovery*, is employed only if an unwished splitting of the formation is detected. In such a case, a new virtual leader is created to lead the unstuck part of the former group. Its aim is to navigate the sub-group back to its desired position within the main formation. It uses the extended MPC trajectory planning approach described in details in Section 5.2.

A communication (via WiFi) is required only between the GeNav leader and particular followers. It is assumed that the *GL* and *VL* modules are realized on the same vehicle. Also the data from the relative localisation processes are stored there. Therefore, the communication between the GeNav leader and followers is limited to sending the desired trajectory and actual data from the visual relative localization.

Finally, let us remark that the trajectories of virtual leader and followers are given in the local frame of the GeNav leader, since all members of the formation know its relative position provided by the top-view localization.

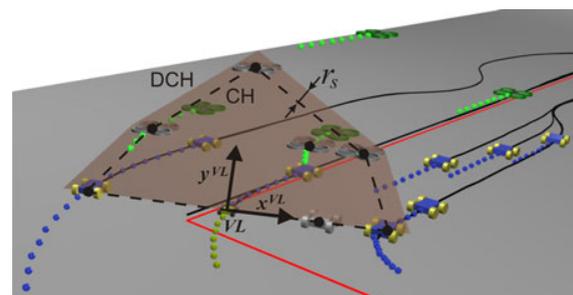
#### 4.2 3D Formation Representation for the Obstacle Avoidance

One of the main contribution of this work is the ability of the system to ensure formation stabilization under the top-view visual relative localization in environments with dynamic obstacles. This requires to design an obstacle avoidance

function included into the trajectory following method, which is introduced in Fig. 3. The core of the avoidance function is a proper representation of the entire formation, which incorporates the requirement on the direct visibility between the robots into the formation stabilization process.

In our approach, the 3D formation is represented by a convex hull of positions of followers projected into a plane  $\mathcal{P}_{VL}$ , which is orthogonal to the trajectory of the virtual leader in its actual position (see Fig. 4). The projection of the position of  $i$ -th follower into the plane  $\mathcal{P}_{VL}$  can be simply obtained as  $x_i^{VL} := q_i$  and  $y_i^{VL} := h_i$ , where  $\{x^{VL}; y^{VL}\}$  is coordinate system in the plane  $\mathcal{P}_{VL}$  as sketched in Fig. 4. The convex hull of the set of points  $\{x_i^{VL}; y_i^{VL}\}$ , where  $i \in \{1, \dots, nf\}$ , is an appropriate representation of the 3D formation under the top-view relative localization by two reasons: (1) Each follower  $i$  intersects the plane  $\mathcal{P}_{VL}$  at point  $\{x_i^{VL}; y_i^{VL}\}$  in future. (2) The convex hull of such a set of points denotes borders of the area, which should stay obstacle free. This ensures that the direct visibility between MAVs and UGVs, which is crucial for the presented top-view visual localization, is satisfied.

Moreover for the obstacle avoidance function presented in Section 4.3, the convex hull (CH) needs to be dilated by a detection boundary radius  $r_s$  to keep obstacles in a desired distance from followers. Only obstacles that are closer to the convex hull than  $r_s$  are considered in the avoidance



**Fig. 4** Dilated convex hull. The shaded contours with black balls represent projections of followers into the plane of virtual leader

function. In the trajectory following process applied for control of followers, the DCH is reduced to a circle with radius equal to  $r_s$  to represent a single robot.

### 4.3 Trajectory Planning and Control Mechanism

Let us now describe the trajectory following mechanism with obstacle avoidance function more in details. As mentioned above, the aim of the method is to find a control sequence that steers the virtual leader along the desired path followed by the GeNav leader and consequently to find control sequences that stabilize the followers behind the virtual leader in desired relative positions. The intention of the method is to find such control sequences that keep the virtual leader as close as possible to the GeNav leader and followers as close as possible to their desired position behind the virtual leader, while satisfying the requirements given by the non-collision formation driving and the top-view relative localization. By applying this concept, the group is able to respond to changes in workspace, which can be dynamic or newly detected static obstacles, and to failures of a robot of the team.

To define the trajectory planning problem in a compact form, we need to gather states  $\psi_j(k)$ , where  $k \in \{1, \dots, N\}$  and  $j \in \{VL, 1, \dots, n_r\}$ , into vector  $\Psi_j \in \mathbb{R}^{4N}$  and control inputs  $\bar{u}_j(k)$  into vector  $\mathcal{U}_j \in \mathbb{R}^{3N}$ . Then all variables describing the trajectory of the virtual leader or a follower can be collected in an optimization vector:  $\Omega_j = [\Psi_j, \mathcal{U}_j] \in \mathbb{R}^{7N}$ . Let us now transform the trajectory planning to minimization of a cost function  $J_j(\Omega_j)$ ,  $j \in \{VL, 1, \dots, n_r\}$ , subject to sets of

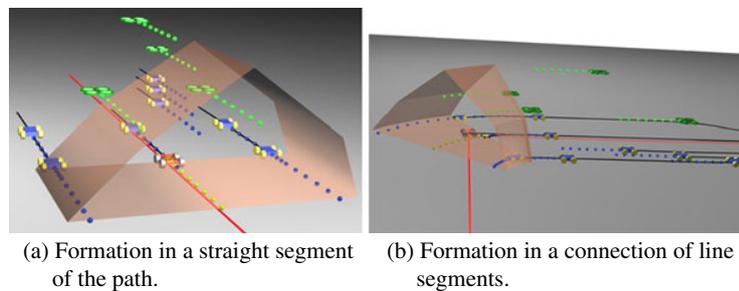
equality constraints  $h_j(k) = 0, \forall k \in \{0, \dots, N - 1\}$ , and inequality constraints  $g_j(k) \leq 0, \forall k \in \{1, \dots, N\}$ . The proposed cost function consists of four components:

$$\begin{aligned}
 J_j(\Omega_j) = & \sum_{k=1}^N \left\| (\bar{p}_{d,j}(k) - \bar{p}_j(k)) \right\|^2 \\
 & + \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{d_{DCH}(\Omega_j, o_l)}{d_{DCH}(\Omega_j, o_l) - R_{DCH}} \right\} \right)^2 \\
 & + \frac{1}{N} \sum_{k=1}^N (v_j(k) - \bar{v}_j)^2 + (K_j(k) - \bar{K}_j)^2 \\
 & + \sum_{f \in \bar{n}_n} \left( \min \left\{ 0, \frac{d_{j,f}(\Omega_j, \Omega_f^\circ) - r_{s,j}}{d_{j,f}(\Omega_j, \Omega_f^\circ) - r_{a,j}} \right\} \right)^2.
 \end{aligned} \tag{3}$$

The first part penalizes solutions with states deviated from the desired states  $\bar{p}_{d,j}(k)$ , where  $k \in \{1, \dots, N\}$ . In the virtual leader’s trajectory tracking, the desired states are obtained by the prediction of the movement of the GeNav leader. In the followers’ trajectory planning, the desired states are derived from the result of the virtual leader’s trajectory tracking using the formation driving concept for each of the followers.

The second term of  $J_j(\Omega_j)$  contributes to the final cost when an obstacle is inside the projection of the dilated convex hull along the planned trajectory. As mentioned, the convex hull represents the formation in case of the virtual leader’s trajectory planning or a single robot in case of the followers’ trajectory planning. Examples of the projected convex hull are shown in Fig. 5. The value of the second term of  $J_j(\Omega_j)$  will be increasing

**Fig. 5** The dilated convex hull projected along the planned trajectory of the virtual leader



as the obstacle is approaching to the centre of the convex hull. The constant  $R_{DCH}$  is equal to half of the maximal width of the dilated complex hull measured in the  $x^L$  coordinate ( $R_{DCH} = r_s$  in the followers' trajectory planning). The function  $d_{DCH}(\Omega_j, o_l)$  provides distance from the dilated convex hull to obstacle  $o_l$  again in the direction of  $x^L$  coordinate. The function value is negative if the obstacle is outside the dilated convex hull and positive if the obstacle is in the hull. The direction of the gradient of such defined avoidance function is to the side of the hull in the  $x^L$  coordinate. This is important since the formation, which is "fixed" by UGVs to the ground, cannot avoid obstacles by change of its altitude.

The third term is important for the reducing of undesirable oscillations in movement of robots and it eliminates needless aggressive manoeuvres. This term penalises high variance of control inputs. During the optimization process, solutions with control inputs deviating from their mean values,  $\bar{v}_j = \frac{1}{N} \sum_{k=1}^N v_j(k)$  and  $\bar{K}_j = \frac{1}{N} \sum_{k=1}^N K_j(k)$ , are penalized, which results into smooth trajectories.

Finally, the last part of the cost function  $J_j(\Omega_j)$  is crucial for the failure tolerance of the system. This term is a sum of avoidance functions in which the other members of the team are considered also as dynamic obstacles. This part has to protect the robots in case of an unexpected behaviour of a defective neighbour. Function  $d_{j,f}(\Omega_j, \Omega_f^\circ)$  provides minimal distance between the planned trajectory  $\Omega_j$  of  $j$ -th follower and the recent plan  $\Omega_f^\circ$  of  $f$ -th robot. The  $(\cdot)^\circ$  symbol denotes the last results of the optimization process for the particular robot. The minimal distance is provided for all  $f \in \bar{n}_n$ , where  $\bar{n}_n = \{1, \dots, j-1, j+1, \dots, n_r\}$ . The detection radius  $r_{s,j}$  is usually smaller than the basic detection radius  $r_s$  used for the dilation of the convex hull, because the follower should not try to avoid a close neighbour if both are at the desired position. Beside the detection radius, we need to define a circular avoidance boundary with radius  $r_{a,j}$ , where  $r_{s,j} > r_{a,j}$ . While, single robots should not respond to other followers detected outside the region with radius  $r_{s,j}$ , distance between the robots and their neighbours less than  $r_{a,j}$  is considered as inadmissible (it could cause a collision).

The equality constraints  $h(k)$  represent the kinematic model (1) for all  $k \in \{0, \dots, N-1\}$  with initial conditions given by the actual state of the leader. This ensures that the obtained trajectory stays feasible with respect to kinematics of utilized robots. It means that these constraints are satisfied if  $\psi_j(k+1)$  is obtained by substituting the vectors  $\psi_j(k)$  and  $\bar{u}_j(k+1)$  into the Eq. 1 for all  $k \in \{0, \dots, N-1\}$ .

The sets of inequality constraints  $g(k)$  characterize bounds on control inputs  $\bar{u}_j(k)$  for all  $k \in \{1, \dots, N\}$ . For all followers, the control inputs are limited by vehicle mechanical capabilities (i.e., chassis and engine) as  $v_{\min,i} \leq v_i(k) \leq v_{\max,i}$ ,  $|K_i(k)| \leq K_{\max,i}$  and for MAVs also  $w_{\min,j} \leq w_j(k) \leq w_{\max,j}$ . These values may differ for each of the followers. For the virtual leader, these limits have to be extended, since the constraints of the entire formation need to be included. The trajectory of the virtual leader must be feasible for all followers in their desired positions. For the virtual leader, the admissible control set can be determined using the leader-follower approach as  $\max_{i=1, \dots, n_r} \left( \frac{-K_{\max,i}}{1-q_i K_{\max,i}} \right) \leq K_{VL}(k) \leq \min_{i=1, \dots, n_r} \left( \frac{K_{\max,i}}{1+q_i K_{\max,i}} \right)$  and  $\max_{i=1, \dots, n_r} \left( \frac{v_{\min,i}}{1+q_i K_L(t)} \right) \leq v_{VL}(k) \leq \min_{i=1, \dots, n_r} \left( \frac{v_{\max,i}}{1+q_i K_L(t)} \right)$ . These restrictions must be applied to respect different values of curvature and speed of robots in different positions within the guided formation. Intuitively, e.g. the robot following the inner track during a turning movement goes slower but with a bigger curvature than the robot further from the center of the turning.

### 5 Fault Diagnosis and Recovery in Formation Control

Faults in multi-robot systems and especially in formations or swarms of various aerial, ground, water, or underwater robots can be investigated in several levels of abstraction. In the most general case, the compact group as a whole can fail in carrying out its task or mission. The second case of faults in multi-robot applications represents examples, where the group is able to continue with performing its task in a limited way, e.g. a

robot or even several robots (a sub-group) from the original group is lost. Finally, we should mention the situation in which all robots can continue towards fulfilment their task, but some of them in a limited way. This case is referred as component/components failure of a robot, where the component may be either a sensor or an actuator.

Similarly, the following faults can occur in the proposed approach of heterogeneous 3D formation driving due to numerous reasons.

- (1) The entire formation can fail if the GeNav leader loses its path to follow.
- (2) The compact formation is not able to follow the GeNav leader, e.g. due to the motion constraints or constraints given by surrounding environment.
- (3) An undesirable separation of a follower or a sub-group of followers from the main formation can be caused by several reasons: lost of the relative localization, influences of the environment or serious failure of robot's motion abilities to name few.
- (4) Finally, a fault of a robot's component, which influences its ability to follow the formation, may occur in both, MAV and UGV platforms.

The performance and stability of the GeNav technique, which is employed by the GeNav leader, is investigated and sufficiently described in [2] and therefore the first item of the list will be skipped in the following analyses. In case of a component failure (the last item of the list), which may be a malfunction of a sensor or actuator that is not fatal for the ability to follow the group, the MPC correction mechanism in the replanning loop takes place. Even in a case of strong disturbances, which significantly change the system in comparison with the model applied in the predictive control, the deviation from the desired position is continuously corrected due to the periodical replanning. If the disturbances exceed a tolerable limit and the MPC mechanism is not able to stabilize the robot within the formation, the situation may be considered as the undesirable separation of a follower (the second item of the list).

Therefore, only two points have to be resolved to fully analyse the behaviour of the presented

formation driving mechanism and to enable recovery in case of failures.

- Fault-detection and recovery of the virtual leader's trajectory tracking mechanism (the second item of the list).
- Fault-detection and recovery of the followers' trajectory tracking mechanism (the third item of the list).

Remaining types of faults, which do not lead to the separation of a robot/robots from the formation, can be compensated by the MPC replanning as it is usual.

### 5.1 Failure Detection

Failures of the virtual leader's as well as followers' trajectory tracking mechanisms can be detected simply by observing the progress of values of function (3) since the same optimization function is used for solving both problems. The deviation of the system (virtual leader or a follower) from its desired position is penalized only by the first term of function (3). The functional value of this term corresponds not only to the actual deviation of the system, but it characterizes also its progress in future. Therefore, an increase of this value in a longer time period indicates that the stabilization mechanism is not able to compensate the deviation. The particular system (a follower or the virtual leader) can be considered as a lost entity and a recovery mechanism has to be activated. It is worth to mention that the value of the first term of Eq. 3 can be temporary increased also due to an obstacle avoidance manoeuvre. Therefore, the threshold signalling the separation of the controlled system has to be higher than the peaks caused by obstacles or a reasoning mechanism with included information given in the second and fourth terms of Eq. 3, where the obstacle proximity is penalized, has to be employed. In the presented experiments, a threshold exceeding the peaks of the value of the first term in Eq. 3, which are caused by the obstacles, has been used with sufficient reliability. Exceeding of this threshold then indicates the formation decay and it is not caused by the regular obstacle avoidance. Once the splitting of the formation or the malfunction of

the virtual leader is detected, the failure recovery mechanisms introduced in Fig. 3 and described in details in the following paragraph needs to be run.

### 5.2 Fault-Tolerant Formation Control

The purpose of this section is not to present the well known MPC technique being able to compensate partial faults and uncertainties of sensors and actuators, but to describe the novel mechanism developed to recovery the splitted formation. The proposed approach is based on creating an ad-hoc virtual leader for commanding the unstuck group back to its desired position (referred to as *DP*) within the former formation. To be accurate, we should clarify that the desired position *DP* corresponds to the desired position of the virtual leader that leads the new formation.

The standard MPC scheme with the limited control horizon with a constant sampling time  $\Delta t$  in-between of  $N$  transition points may not be sufficient for the navigation of the unstuck part of the formation back to its position if the distance to the rest of the group significantly exceeds the length of the horizon. A simple prolonging of the horizon would quickly touch the limits of available computational resources. Therefore, we propose to extend the control scheme of the virtual leader with an additional planning horizon with variable sampling time in-between of transition points. This horizon is used for the trajectory planning of the separated formation/robot into its desired position *DP*. The entire horizon is then divided into two segments, the standard control horizon and the planning horizon. In the planning horizon, lengths of time intervals between transition points are also variables taking part in the planning problem. This planning algorithm again respects constraints given by the top view localization and by kinematics of followers to be sure that the plan is feasible in case of a sub-formation breakaway.

To define the trajectory planning problem with two time intervals in a compact form we need to gather states  $\psi_j(k)$ , where  $k \in \{N + 1, \dots, N + M\}$ , and control inputs  $\bar{u}_j(k)$ , where  $k \in \{N + 1, \dots, N + M\}$ , into vectors  $\Psi_{j,M} \in \mathbb{R}^{4M}$  and  $\mathcal{U}_{j,M} \in \mathbb{R}^{3M}$ , similarly as it was done with  $\Psi_j \in \mathbb{R}^{4N}$  and  $\mathcal{U}_j \in \mathbb{R}^{3N}$  in Section 4.3. The variable

$M$  denotes number of the transition points employed in the sparse planning horizon, while  $N$  is number of transition points in the short control horizon, which was used also in the trajectory following algorithm described in Section 4.3. Also values  $\Delta t(k)$ ,  $k \in \{N + 1, \dots, N + M\}$ , that become variables in the planning horizon need to be gathered into vector  $\mathcal{T}_{j,M}^\Delta$ . All variables describing the complete trajectory (with both control and planning horizons) from the actual position of the separated sub-formation until the desired position within the original formation can be collected into the optimization vector  $\Omega_{j,2} = [\Psi_j, \mathcal{U}_j, \Psi_{j,M}, \mathcal{U}_{j,M}, \mathcal{T}_{j,M}^\Delta] \in \mathbb{R}^{7N+8M}$ .

The fault-recovery mechanism can be realized through the minimization of cost function  $J_{j,2}(\Omega_{j,2})$  subject to equality constraints  $h_j(k) = 0$ ,  $\forall k \in \{0, \dots, N + M - 1\}$ ,  $g_S(\psi_j(N + M)) = 0$  and inequality constraints  $g_j(k) \leq 0$ ,  $\forall k \in \{1, \dots, N\}$ . The stability constraint  $g_S(\psi_j(N + M))$  guarantees that the found trajectory for the formation will reach its desired position *DP*. The stability constraint is given by  $g_S(\psi_j(N + M)) := \|\bar{p}_j(N + M) - DP\|$ , where  $\bar{p}_j(N + M)$  is position of the last transition point in the trajectory. The constraints  $h_j(\cdot)$  and  $g_j(\cdot)$  are described in Section 4.3.

The cost function  $J_{j,2}(\Omega_{j,2})$ , employed in the trajectory planning and obstacle avoidance problem, is equivalent to the cost function  $J_j(\Omega_j)$  from Eq. 3 except the first term penalizing deviation from the desired states. Here, this term is replaced by summation  $\sum_{k=N+1}^{N+M} \Delta t(k)$ , which minimizes the total time to reach the desired location *DP*. The value of the sum correlates with the estimated time of the formation movement in the planning horizon if the target would be static. In case of the formation re-coupling, even a moving main formation (and so moving *DP*) can be reached by the unstuck robots due to the periodical MPC replanning.

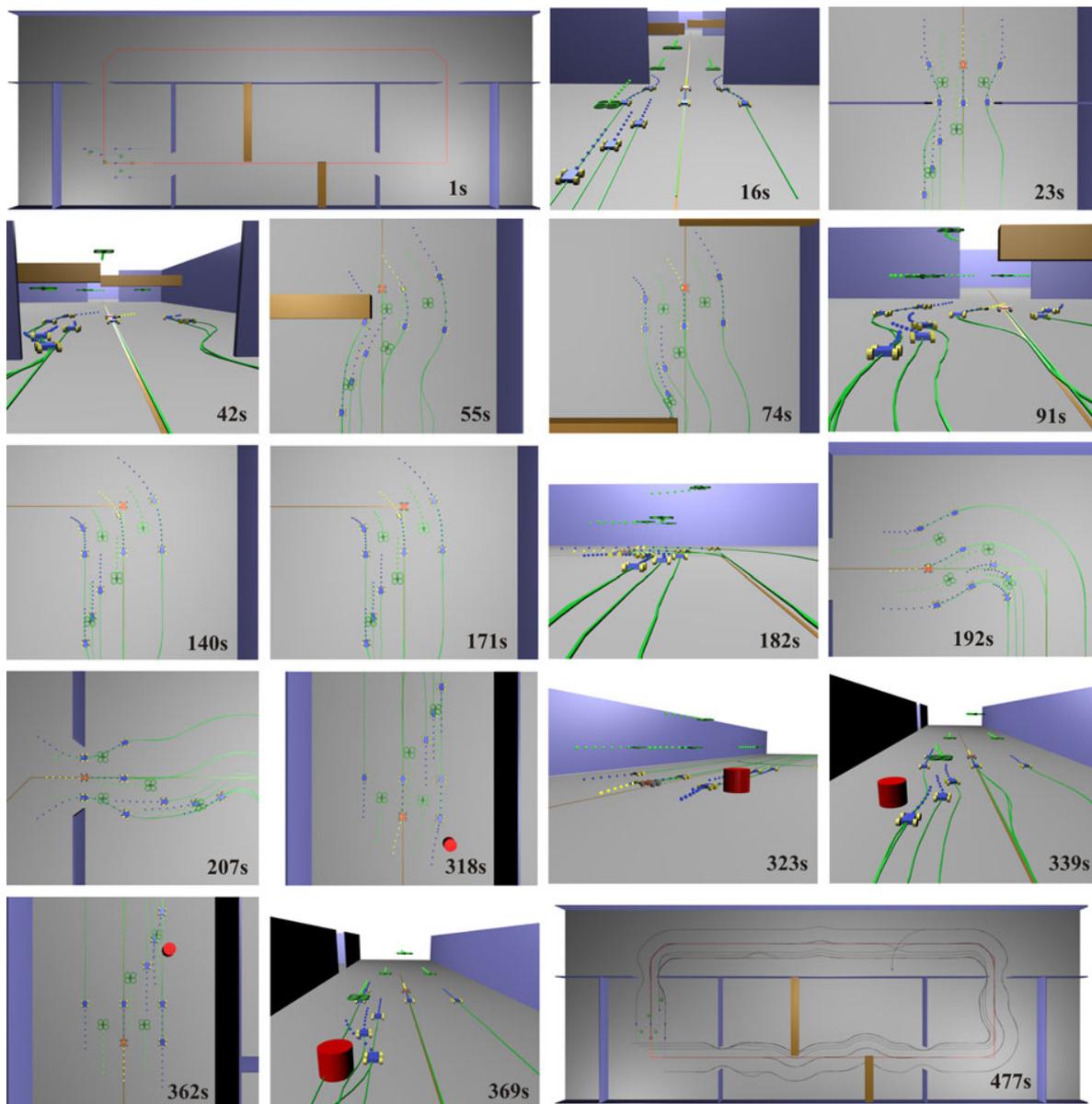
## 6 Experimental Results

Results presented in this section have been obtained by the proposed algorithm with the Sequential Quadratic Programming (SQP) method [38] employed for solving the optimization problems

used in the virtual leader trajectory tracking and for the stabilization and obstacle avoidance of followers. This solver provided the best performance from the tested available algorithms. Nevertheless, one can use any optimization method, which is able to solve the optimization problems defined in this paper.

### 6.1 Simulation of the Formation Movement with Obstacle Avoidance Tasks

The performance of the presented method in an environment with static and dynamic obstacles is shown in the simulation in Fig. 6. The method is used with parameters:  $n=2$ ,  $N=8$  and  $\Delta t=0.25$  s.

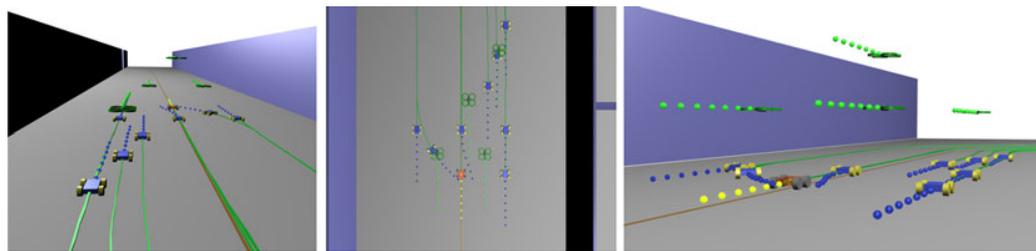


**Fig. 6** Snapshots of the formation movement simulation

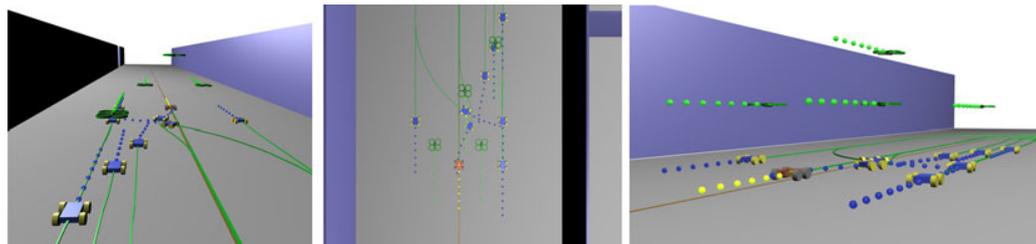
**Table 1** Curvilinear coordinates of followers within the formation used in the experiment presented in Figs. 6 and 7

$i$	1	2	3	4	5	6	7	8	9	10	11	12
$p_i$	0	0	1	1	1	2	2.7	3.4	.5	.5	2.7	1.6
$q_i$	1	-1	1	-1	0	.6	.8	1	.5	-.5	.8	.2
$h_i$	0	0	0	0	0	0	0	0	.5	.5	.5	1

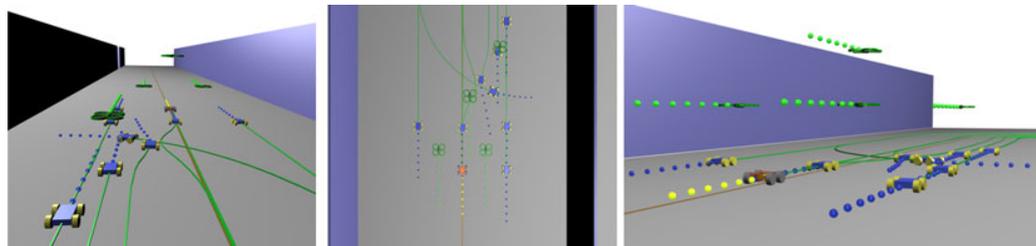
The experiment presents performance of this approach in scenarios inspired by a real world mission. The formation driving technique is employed in a surveillance application, in which a heterogeneous team of MAVs and UGVs has to periodically move through three rooms connected by a corridor. The objective of the mission is to follow a given path and to keep a desired shape of the



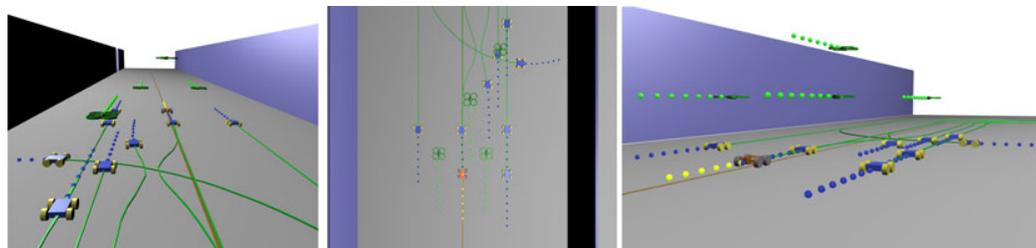
(a) Failure of follower 2 deviating from its desired position. 289s



(b) Follower 5 avoiding the broken robot. 298s



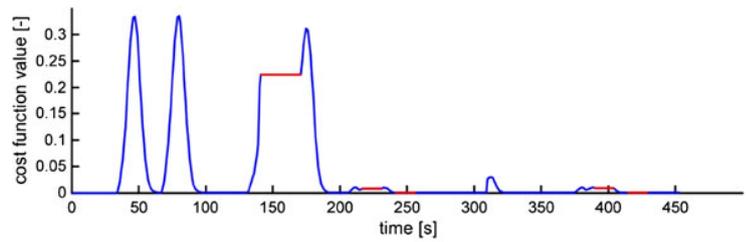
(c) Follower 6 avoiding the broken robot. 303s



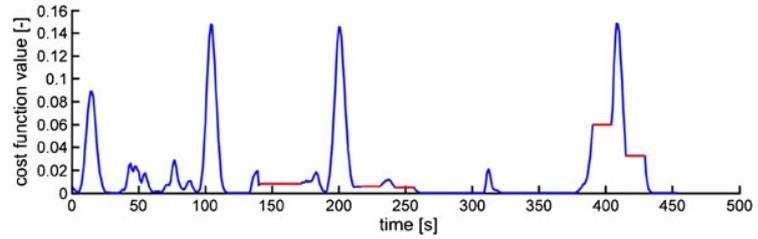
(d) All followers successfully avoided the broken robot. 308s

**Fig. 7** Simulation of a response of the formation driving algorithm to a failure of one of the followers. The manoeuvre was recorded by three virtual cameras

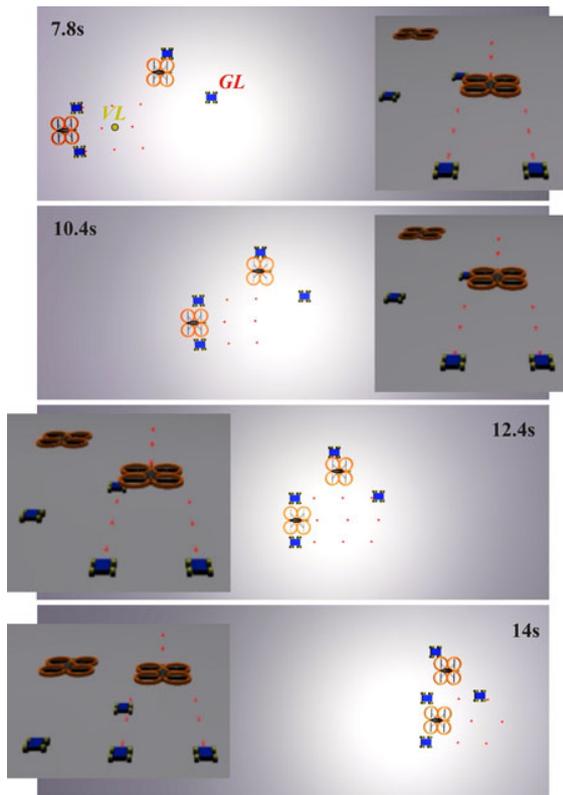
**Fig. 8** Progress of the cost function employed in the trajectory planning method during the movement presented in Figs. 6 and 7



(a) Virtual leader.



(b) Follower 1.



**Fig. 9** MAVs-UGVs formation recovery after its undesired splitting into two independent units

formation. The formation can be autonomously temporarily shrunk in narrow passages (e.g. in doorways) or due to dynamic obstacles forcing followers to perform avoidance manoeuvres. The team (described in Fig. 2) consists of the GeNav leader (the orange robot denoted by *GL*), the virtual leader (the yellow robot denoted by *VL*), 8 UGV followers and 4 MAV followers. Three of the MAVs are positioned in a lower altitude to be able to relatively localize the ground robots. The fourth MAV is flying above them to provide relative positions of the lower MAVs. The MAV flying in the highest altitude could detect also the UGVs, but with much lower precision and reliability due to the greater relative distance and possibility of visibility interruption by one of the lower MAVs. Besides, the desired relative positions of the MAVs in the formation are determined in

**Table 2** Performance of the fault-recovery mechanism with different ratio between the speed of the main formation ( $v_{main} = 1$  in the experiments) and the speed of the unstuck formation  $v_{unstuck}$

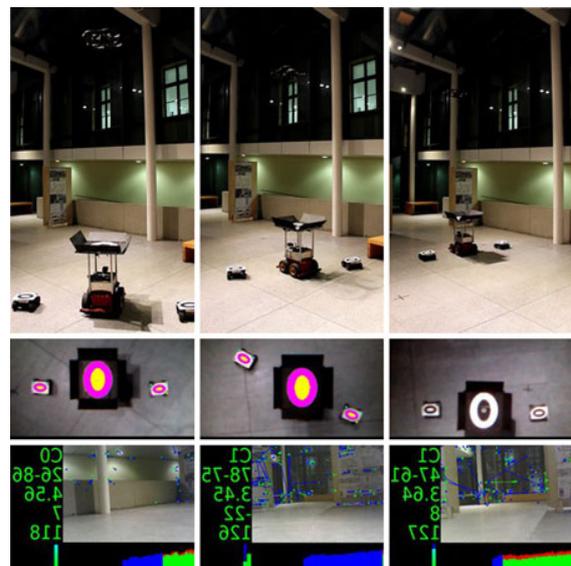
$v_{unstuck}$	1	1.1	1.2	1.3	1.4	1.5	1.6
<i>Time</i>	inf	41 s	21 s	14 s	10 s	8 s	7 s

The value of *Time* indicates time required for re-coupling of the sub-formation that was deviated from its desired position in distance 2 map units in *x* coordinate and 5 map units in *y* coordinate

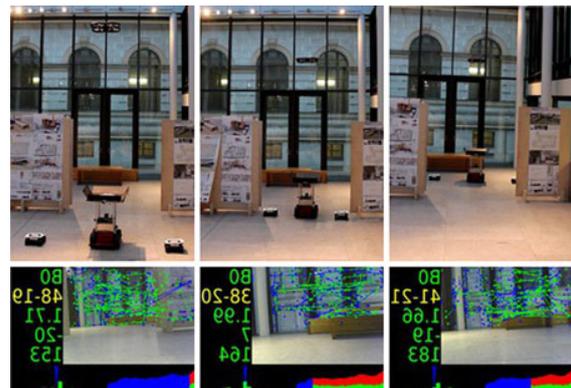
such a way that they are not mutually influenced by air flow effects. The followers' coordinates relative to the virtual leader are presented in Table 1.

The initial position of the group is depicted in the first snapshot in Fig. 6 captured at time 1 s. In snapshots 16–23 s, the outer followers of the formation temporarily deviate from their desired positions to pass through the narrow passage towards the second room. The original shape of the formation is restored and the group starts avoiding the overhead obstacle in snapshots 42–55 s. The obstacle is sufficiently high to be passed under by all robots except the MAV flying in the highest altitude. The GeNav leader can be navigated without any influence of the obstacle, but the rest of the formation has to move away from the desired path to keep the constraints given by the relative localization. In the snapshot captured at time 55 s, one can see the deviation of the position of the virtual leader from the position of the GeNav leader. This enables to avoid the obstacle in a way that the obstacle is always situated outside the dilated convex hull of the formation. In the 74th second, the formation returns back on the desired path, but it is again forced to avoid the second overhead obstacle (snapshot at time 91 s). At time 140 s, the GeNav leader is approaching into the first connections of line segments of the path. The virtual leader and the followers are waiting for the GeNav leader, which has to turn on the spot. They are already deviated from the path to be able to smoothly continue without any complicated manoeuvring. Once the turning of the GeNav leader is finished (171 s), the complete formation continues back on the desired path (182 s, 192 s). At time 318 s, an unknown obstacle is detected by the formation. The obstacle is avoided, using the virtual leader's obstacle avoidance function, at the price of temporarily leaving of the desired path (snapshots at 323 s and 339 s). The second obstacle is detected by the followers, see snapshot at time 362 s. This dynamic obstacle cannot be avoided by the virtual leader's re-planning, since it was detected too late. Therefore, the avoidance function included in the follower's trajectory following method is utilized here. The shape of the formation is temporarily changed to keep the obstacle outside the dilated convex hull (369 s).

Cost-function values of the virtual leader's and the 1st follower's trajectory planning during the movement presented in Fig. 6 are depicted in Fig. 8. The peaks in the course of the leader's cost values correspond with the places of connections of line segments forming the desired path that has to be followed. In these connections, the virtual leader is forced to deviate from the path to be able to pass the sharp edges of the path smoothly.



(a) Formation going through a connection of straight segments of the path.



(b) Formation is temporarily shrinking to get through the narrow passage.

**Fig. 10** Formation driving using the GeNav algorithm for the navigation and the top-view relative localization for the stabilization

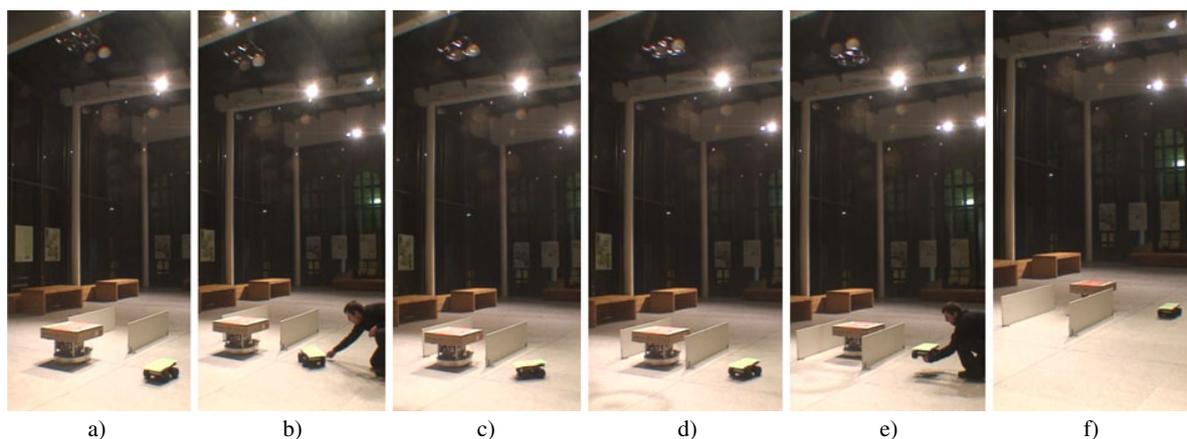
Also the first unknown obstacle is contributing into the virtual leader's cost values. The temporal increase of cost values of the trajectory tracking of follower 1, which was chosen as an interesting example, is caused by the proximity of the obstacles. The obstacles force the robot to leave the desired position in the formation. The deviation from the desired state is penalized by the first term in the Eq. 3. The red lines in the courses of the values denote parts, in which the GeNav leader is turning to be able to follow the next path segment and the rest of the formation is waiting in static positions.

## 6.2 Simulations of the Fault Tolerance and Recovery

Beside the obstacle avoidance abilities, we have tested also the proposed fault-tolerant mechanism included in the formation driving scheme. As a part of the complex experiment presented in Fig. 6, a failure of one of the followers (its steering was blocked) has been simulated. The response of the formation to the undesired motion of the broken follower is in details shown in Fig. 7. In the snapshots, a successful avoidance manoeuvre of followers 5 and 6 as a response to prediction of the collision is demonstrated (see the last part of Eq. 3 for details on the applied avoidance function).

The second simulation (Fig. 9), verifying the proposed fault-recovery technique, presents the

re-coupling of an inadvertently disconnected formation using the trajectory planning approach introduced in Section 5.2. In the experiment, the formation led by the robot equipped with the GeNav system (denoted as *GL* in the picture) follows a straight path segment. A sub-formation of two UGVs and one MAV has been separated from the group. In this unstuck group, a new virtual leader with position denoted as *VL* in the picture is created. Physically, the planning procedure of *VL* is run on a robot with sufficient computational power. It can be any robot from the original group, since it is assumed that the range of the visual relative localization (which cannot be interrupted) is significantly lower than the range of WiFi used for transfer of the plans for particular followers. The new formation starts its movement to reach the rest of the group from a location, which is deviated 2 map units sideways and 5 map units behind its desired position. As expected, the performance of the algorithm varies depending on the difference between the speed of the former formation and the maximal allowed speed of the unstuck group. This relation is shown in Table 2, where the performance of the formation recovery is expressed as the time needed for the re-coupling. It is obvious that in the case of same maximal allowed speed of both formations (the first column of the table), the unstuck formation cannot reach its desired position *DP* (denoted by the *inf* sign in the table).



**Fig. 11** Demonstration of the mechanism providing fault-tolerance in the MAVs-UGVs formation driving

### 6.3 Hardware Experiments

The experiment in Fig. 10a demonstrates the ability of the obstacle avoidance by temporary shrinking of the formation and it verifies the formation movement in a connection of path segments (Fig. 10b). In the experiment, the Pioneer 3-AT robotic platform is employed as the GeNav leader and two MMP5 platforms and the Ar.Drone MAV act as followers. Beside the pictures of the formation movement, images used for the GeNav visual navigation and for the top-view relative localization are shown in Fig. 10 and in a video record of the experiment [39].

In the final experiment, the failure recovery mechanism is shown in practice (see Fig. 11). The employed formation consists of a G2Bot-Testbed of the Czech Technical University employed as the leader and the MMP5 robot and the Ar.Drone used as followers. In the experiment, the MMP5 follower is firstly slightly pushed from its position within the formation, which is corrected based on the information from the top view relative localization only. The failure of the formation integrity is demonstrated by the forced shift of the robot behind the formation, which is again compensated to achieve the desired shape of the formation.

## 7 Conclusion

In this paper, a novel fault-tolerant formation driving approach developed for heterogeneous MAVs-UGVs teams was proposed. The core of the method consists in stabilization of the compact 3D formation by the top-view visual relative localization in control feedback. Besides, the proposed method is suited for utilization of a simple visual navigation of the formation based on detection of features in images obtained by an onboard camera. It was shown that these simple on-board vision based systems enable to deploy teams of closely cooperating unmanned ground and aerial vehicles in environments without any pre-installed infrastructure for robots' localization. Beyond the description and experimental verification of the proposed method, the fault diagnosis and recovery mechanisms were provided in the paper.

**Acknowledgement** This work was supported by GAČR under M. Saska's postdoc grant no. P10312/P756.

## References

1. Faigl, J., Krajník, T., Chudoba, J., Preucil, L., Saska, M.: Low-cost embedded system for relative localization in robotic swarms. In: Proc. of IEEE International Conference on Robotics and Automation (2013)
2. Krajník, T., Faigl, J., Vonásek, M., Kulich, V., Košnar, K., Přeucil, L.: Simple yet stable bearing-only navigation. *J. Field Robot.* **27**(5), 511–533 (2010)
3. Saska, M., Krajník, T., Přeucil, L.: Cooperative micro uav-ugv autonomous indoor surveillance. In: IEEE SSD (2012)
4. Dong, W.: Robust formation control of multiple wheeled mobile robots. *J. Intell. Robot. Syst.* **62**(3–4), 547–565 (2011)
5. Hengster-Movrić, K., Bogdan, S., Draganjac, I.: Multi-agent formation control based on bell-shaped potential functions. *J. Intell. Robot. Syst.* **58**(2), 165–189 (2010)
6. Liu, Y., Jia, Y.: An iterative learning approach to formation control of multi-agent systems. *Syst. Control Lett.* **61**(1), 148–154 (2012)
7. Do, K.D., Lau, M.W.: Practical formation control of multiple unicycle-type mobile robots with limited sensing ranges. *J. Intell. Robot. Syst.* **64**(2), 245–275 (2011)
8. Ghommam, J., Mehrjerdi, H., Saad, M., Mnif, F.: Formation path following control of unicycle-type mobile robots. *Robot. Auton. Syst.* **58**(5), 727–736 (2010)
9. Sira-Ramiandrez, H., Castro-Linares, R.: Trajectory tracking for non-holonomic cars: a linear approach to controlled leader-follower formation. In: IEEE Conf. on Decision and Control (CDC) (2010)
10. Xiao, F., Wang, L., Chen, J., Gao, Y.: Finite-time formation control for multi-agent systems. *Automatica* **45**(11), 2605–2611 (2009)
11. No, T.S., Kim, Y., Tahk, M.-J., Jeon, G.-E.: Cascade-type guidance law design for multiple-uav formation keeping. *Aerosp. Sci. Technol.* **15**(6), 431–439 (2011)
12. Saffarian, M., Fahimi, F.: Non-iterative nonlinear model predictive approach applied to the control of helicopters group formation. *Robot. Auton. Syst.* **57**(67), 749–757 (2009)
13. Liu, C., Chen, W.-H., Andrews, J.: Piecewise constant model predictive control for autonomous helicopters. *Robot. Auton. Syst.* **59**(78), 571–579 (2011)
14. Abdessameud, A., Tayebi, A.: Formation control of vtol unmanned aerial vehicles with communication delays. *Automatica* **47**(11), 2383–2394 (2011)
15. Tanner, H., Christodoulakis, D.: Decentralized cooperative control of heterogeneous vehicle groups. *Robot. Auton. Syst.* **55**(11), 811–823 (2007)
16. Isermann, R.: Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Springer (2006)
17. Freddi, A., Longhi, S., Monteriu, A.: Actuator fault detection system for a mini-quadrotor. In: IEEE Inter-

- national Symposium on Industrial Electronics (ISIE), pp. 2055–2060. IEEE (2010)
18. Heredia, G., Ollero, A., Bejar, M., Mahtani, R.: Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics* **18**(2), 90–99 (2008)
  19. Ranjbaran, M., Khorasani, K.: Fault recovery of an under-actuated quadrotor aerial vehicle. In: 49th IEEE Conference on Decision and Control (CDC), pp. 4385–4392. IEEE (2010)
  20. Ranjbaran, M., Khorasani, K.: Generalized fault recovery of an under-actuated quadrotor aerial vehicle. In: American Control Conference (ACC), pp. 2515–2520. IEEE (2012)
  21. Mead, R., Long, R., Weinberg, J.B.: Fault-tolerant formations of mobile robots. In: IEEE/RSJ IROS, pp. 4805–4810. IEEE (2009)
  22. Chamseddine, A., Zhang, Y., Rabbath, C.A.: Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles. In: American Control Conference (ACC), pp. 3291–3296. IEEE (2012)
  23. Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A.: Multi-uav cooperative fault detection employing vision based relative position estimation. In: Proceedings of the 17th IFAC World Congress, pp. 12 093–12 098 (2008)
  24. Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A.: Multi-unmanned aerial vehicle (UAV) cooperative fault detection employing differential global positioning (DGPS), inertial and vision sensors. *Sensors* **9**(9), 7566–7579 (2009)
  25. Ismail, A.R., Timmis, J.: Aggregation of swarms for fault tolerance in swarm robotics using an immunoeengineering approach. In: UK Workshop on Computational Intelligence (2009)
  26. Christensen, A.L., O’Grady, R., Dorigo, M.: From fireflies to fault-tolerant swarms of robots. *IEEE Trans. Evol. Comput.* **13**(4), 754–766 (2009)
  27. Barambones, O., Etxebarria, V.: Robust adaptive control for robot manipulators with unmodelled dynamics. *Cybern. Syst.* **31**(1), 67–86 (2000)
  28. Alamir, M.: Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes. Ser. Lecture Notes in Control and Information Sciences, vol. 339. Springer, Berlin/Heidelberg (2006)
  29. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. *Automatica* **36**(6), 789–814 (2000)
  30. Boscariol, P., Gasparetto, A., Zanotto, V.: Model predictive control of a flexible links mechanism. *J. Intell. Robot. Syst.* **58**(2), 125–147 (2010)
  31. Chao, Z., Zhou, S.-L., Ming, L., Zhang, W.-G.: Uav formation flight based on nonlinear model predictive control. *Math. Probl. Eng.* **2012**(1), 1–16 (2012)
  32. Defoort, M.: Distributed receding horizon planning for multi-robot systems. In: IEEE International Conference on Control Applications (CCA), pp. 1263–1268 (2010)
  33. Zhang, X., Duan, H., Yu, Y.: Receding horizon control for multi-uavs close formation control based on differential evolution. *Sci. China Inf. Sci.* **53**(2), 223–235 (2010)
  34. Saska, M., Krajník, T., Vonásek, V., Vanek, P., Preucil, L.: Navigation, localization and stabilization of formations of unmanned aerial and ground vehicles. In: ICUAS (2013)
  35. Krajník, T., Nitsche, M., Pedre, S., Přeučil, L., Mejail, M.: A simple visual navigation system for an UAV. In: International Multi-Conference on Systems, Signals and Devices, p. 34. IEEE, Piscataway (2012)
  36. Krajník, T., Vonásek, V., Fišer, D., Faigl, J.: AR-drone as a platform for robotic research and education. In: Research and Education in Robotics: EUROBOT 2011. Springer, Heidelberg (2011)
  37. Barfoot, T.D., Clark, C.M.: Motion planning for formations of mobile robots. *Robot. Auton. Syst.* **46**, 65–78 (2004)
  38. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (2006)
  39. Movie: Movie of hw experiment and simulation presented in this paper. Online: <http://imr.felk.cvut.cz/formation/> (2013). cit. 2013-2-22



---

KEY ARTICLE [3] - AUTONOMOUS ROBOTS 2017

---

©[2016] IEEE. This article was published in journal Autonomous Robots: **M. Saska**, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar: System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization, 2017.



# System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization

Martin Saska<sup>1</sup> · Tomas Baca<sup>1</sup> · Justin Thomas<sup>2</sup> · Jan Chudoba<sup>1</sup> · Libor Preucil<sup>1</sup> · Tomas Krajnik<sup>3</sup> · Jan Faigl<sup>4</sup> · Giuseppe Loianno<sup>2</sup> · Vijay Kumar<sup>2</sup>

Received: 27 January 2015 / Accepted: 12 April 2016  
© Springer Science+Business Media New York 2016

**Abstract** A complex system for control of swarms of micro aerial vehicles (MAV), in literature also called as unmanned aerial vehicles (UAV) or unmanned aerial systems (UAS), stabilized via an onboard visual relative localization is described in this paper. The main purpose of this work is to verify the possibility of self-stabilization of multi-MAV groups without an external global positioning system. This approach enables the deployment of MAV swarms outside laboratory conditions, and it may be considered an enabling technique for utilizing fleets of MAVs in real-world scenarios. The proposed visual-based stabilization approach has been designed for numerous different multi-UAV robotic applications (leader-follower UAV formation stabilization, UAV swarm stabilization and deployment in surveillance scenarios, cooperative UAV sensory measurement) in this paper. Deployment of the system in real-world scenarios truthfully verifies its operational constraints, given by limited onboard sensing suites and processing capabilities. The performance of the presented approach (MAV control, motion planning, MAV stabilization, and trajectory planning) in multi-MAV applications has been validated by experimental results in

indoor as well as in challenging outdoor environments (e.g., in windy conditions and in a former pit mine).

**Keywords** Micro aerial vehicles (MAVs) · Unmanned aerial vehicles (UAVs) · Formations · Swarms · Visual relative localization · Stabilization · Control · Trajectory planning

## 1 Introduction

The proposed approach relies strictly on onboard sensors and aspires to be an enabling technique for using closely cooperating MAV-groups in workspaces that are not equipped with motion capture systems (e.g. VICON<sup>1</sup>), which usually provide very precise and fast global localization of MAVs. With the proposed method, the utilization of closely cooperating MAVs is possible without installing any global localization infrastructure prior to the MAVs deployment in a GPS-denied environment. Besides, it enables applicability of multi-MAV teams in tasks requiring flight operations in close proximity between neighbors, where precision and reliability of GPS are not sufficient. The proposed approach is also especially appealing for missions in which the GPS signal may be jammed.

The robot localization being restricted to the onboard sensory system also significantly reduces the amount of communication necessary for the robots' coordination. In some applications, the group stabilization and control towards mission objectives can be achieved without explicit communication, as shown later in this paper where examples of the applicability of the system are presented. Disabled communication is crucial for MAVs operating in workspaces where

---

✉ Martin Saska  
martin.saska@fel.cvut.cz

<sup>1</sup> Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague, Czech Republic

<sup>2</sup> GRASP Lab, University of Pennsylvania, 3330 Walnut Street, 19104 Philadelphia, PA, USA

<sup>3</sup> Lincoln Centre for Autonomous Systems, University of Lincoln, Brayford Pool, Lincoln, UK

<sup>4</sup> Dept. of Computer Science, Faculty of Electrical Engineering Czech Technical University in Prague, Technicka 2, 166 27 Prague, Czech Republic

<sup>1</sup> <http://www.vicon.com/>.

radio transmissions are not feasible due to the structure of the environment or due to safety rules. Besides, current communication technologies do not provide sufficient bandwidth for large communities of robots operating in relatively small areas. In the proposed method, robots can share the information required for self-stabilization through observation of states of neighbors, i.e. by the onboard (in our case, visual) relative localization.

This paper presents a control system designed for multi-MAV teams, its overall structure, and a description of its components. An important part of the paper is an overview of three commonly used planning approaches for multi-MAV system (formation control, environment monitoring by swarm control, and MAV-group deployment in a surveillance scenario), which were designed for using with this system. In the description of the methods, it is highlighted how to deal with constraints given by the visual relative localization and how to integrate them into motion planning in specific multi-MAV applications. This should provide a guideline for developing high level planning algorithms in specific multi-MAV applications, since satisfying constraints of the onboard relative localization is crucial for achieving reliable behaviour by the MAV-group. Unlike the external global positioning system, where the precision and reliability of the robots' localization is independent to mutual positions of MAVs and the shape of the swarm, the operational space of the onboard relative localization sensors (for the vision sensor, mainly the range and the view angle of cameras) significantly limits the deployment of robots.

We rely on a light-weight embedded vision system using monocular cameras with a limited view angle. The system takes advantage of the possibility to equip all team members with black and white (B/W) patterns, which enables us to achieve sufficient precision on the order of centimeters if the actual distance between neighboring vehicles is on the order of meters. The detection of simple patterns with known shape and size also significantly speeds up the image processing. The localization system may therefore provide relative position measurements up to 60 times per second, and may be directly employed in the feedback loop for control and stabilization of the MAV-group.

The proposed control scheme integrates information from an onboard camera module with data from an inertial measurement unit and a commercially available PX4Flow<sup>2</sup> smart sensor employed to measure the altitude and velocities of particular MAVs in the swarm. The MAV-group is then stabilized in three levels. The lowest level is the fastest control loop realized by the OEM MikroKopter's attitude stabilization board.<sup>3</sup> Above this loop, we have developed a position stabilization mechanism that leverages data from the visual relative local-

ization unit in the control feedback. On the top of that, we show three examples of swarm motion planning. The motion planning acts as the third control level designed for navigation of the whole MAV-group and its stabilization in required shapes, which may be dynamically changed. The methods employ a concept of adaptively evolving group behaviors that are established to decrease the uncertainty of the relative localization. These approaches are novel in the way how the constraints of vision based localization are incorporated into the control scheme. The operational constraints of the relative localization describe where neighboring particles or an object of interest equipped with the identification pattern may be detected and localized with a required precision and reliability. Plans that consider a model of the localization precision and reliability may decrease the overall uncertainty and increase the reliability of the complex autonomous system, as it was shown in our previous work on this topic (Faigl et al. 2012). Therefore, the proposed group motion planning approaches use a model of the localization system arising from theoretical analyses of the vision system and from an experimental evaluation of the system performance in real scenarios.

The paper is organized as follows. The related work and the contribution of the proposed MAV-group stabilization systems with respect to the state-of-the-art are presented in Sect. 2. In Sect. 3, the hardware components of the localization module and the pattern detector approach are presented. The control scheme suited for onboard visual relative localization is proposed in Sect. 4. Section 5 presents an experimental verification of the system. Section 6 summarizes three examples of high level motion control with integrated MAV motion constraints, obstacle avoidance, and constraints of the relative localization. These approaches and the performance of the onboard relative localization system are verified in real flight conditions. Finally, concluding remarks are stated in Sect. 7.

## 2 State-of-the-art

### 2.1 Swarms of autonomous vehicles

Recent research on multi-MAV systems has focused on aspects of communication and maintenance of connectivity within the team members (Teacy et al. 2010; Schmickl and Crailsheim 2008), modeling of the swarm behavior by predicting individual behaviors (Winfield et al. 2008; Hamann and Worn 2008), task allocation and strategies for solving multiple tasks (Berman et al. 2009; Liu et al. 2007; Fazli et al. 2013), and control and collision avoidance within the swarm (Sharma and Ghose 2009; Kumar et al. 2010; Yu and Beard 2013; Turpin et al. 2012). Topics covered in this paper are related mainly to control and stabilization of MAV teams.

<sup>2</sup> <https://pixhawk.org/modules/p4flow>.

<sup>3</sup> <http://www.mikrokoetter.de/>.

In literature, one can find papers describing control methodologies for swarms of both autonomous ground vehicles (Kloetzer and Belta 2007; Cai et al. 2011; Cheah et al. 2009; Marjovi and Marques 2013) and unmanned aerial vehicles (Bennet and McInnes 2009; Barnes et al. 2008; Holland et al. 2005; Doitsidis et al. 2012). These methods are often inspired by nature (e.g., by flocks of birds Leonard & Fiorelli (2001) or molecules forming crystals Balch and Hybinette (2000)), and they try to fulfil various requirements of swarm robotics. Since the proposed approach follows the requirements of swarms as listed in Trianni (2008): *scalability for large groups, high redundancy and fault tolerance, usability in tasks unsolvable by a single robot and locally limited sensing and communication abilities*, examples of studies investigating these domains should also be mentioned. In particular, a hierarchical framework for planning and control of arbitrarily large swarms is proposed in Kloetzer and Belta (2007). Considerations influencing the fault tolerance of teams are discussed in Christensen et al. (2009) and various co-operation strategies for teams of MAVs solving multi-robot tasks are published in Buerkle and Leuchter (2009). Finally, controllers for swarms of robots with limited communication requirements are described in Cai et al. (2011) and Cheah et al. (2009), where the necessary conditions for swarm stability are described using a direct graph topology in Cai et al. (2011), and a Lyapunov-like function is employed for convergence analysis of multi-robot systems in Cheah et al. (2009).

The work in Cheah et al. (2009), which investigates swarming behaviors of ground robots in a planar environment, is the most closely related to the research proposed in this paper. We also aim to develop a system for stabilization of swarms in a desired shape while maintaining a close distance among swarm members. Beyond the method designed in Cheah et al. (2009) for ground robots, 3D swarm principles and swarming rules adapted for the requirements of visual relative localization are established in this paper.

In general, most of the state-of-the-art algorithms mentioned above have been verified only via numerical simulations, using ground vehicles, or rarely with MAVs, but in laboratory conditions (usually with VICON in control feedback). These approaches therefore often omit realistic constraints given by the real outdoor deployment of compact MAV-groups, which is the aim of this paper. The proposed system goes beyond these works mainly by incorporating the requirements of relative visual positioning into the MAV-group motion planning, stabilization, and coordination. This improvement makes it possible to deploy large multi-MAV systems flying in compact formations or swarms outside of laboratories equipped with positioning systems. Besides, the possibility of direct interactions by perceiving neighboring robots in the MAV-group brings artificial swarms closer to the initial ideas and theoretical studies of swarming principles observed in nature.

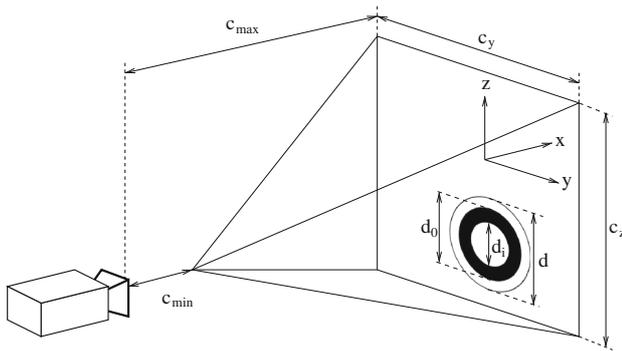
## 2.2 Systems of relative localization of autonomous robots

Let us now briefly describe the state-of-the-art methods of geometric pattern detection, since the employed visual relative localization system based on B/W pattern detection is instrumental in the presented control approach. A basic method for geometric pattern detection is the Generalized Hough Transform (Ballard 1981) used for finding the parameters of the expected geometrical shapes, which is unfortunately computationally demanding. The computational complexity issue is investigated e.g. in Cai et al. (2004), where the RANSAC algorithm is applied, in Rad et al. (2003), which is aimed at tracking objects easily separable from the background, and in Jia et al. (2011), where the method is constrained to finding ellipses. These methods are sufficiently fast when using a standard PC, which may be placed onboard more powerful ground robots. However, these methods cannot be considered real-time for light-weight MAVs equipped with small embedded processors. One can find algorithms suited for embedded systems with real-time performance, but their limitations restrict their utilization in real-world applications (e.g. the system in Carreras et al. (2003), which is based on detecting color segments, and the approach in Masselli and Zell (2012), which uses a pattern of four tennis balls, suffers in varying lighting conditions).

If we omit methods with image processing performed on an external desktop PC (e.g. Garca Carrillo et al. (2011); Bošnjak et al. (2012)), the most relevant approach to our vision system is proposed in Lange et al. (2009). The method (Lange et al. 2009) uses white rings for MAV positioning during landing, but provides a relative position update at only 0.1Hz. In addition, a more powerful onboard PC is required for the real-time control in Lange et al. (2009). The same problem arises in Yang et al. (2012), where the “H” shape landing pattern is detected in real-time, but with a powerful onboard PC. Our solution provides sufficient sensitivity of detection and precision for the MAV-group stabilization and satisfies computational requirements of onboard embedded systems carried by lightweight MAVs.

## 3 System for relative localization

As mentioned in the introduction, the core technique for the proposed stabilization, coordination, and navigation of MAVs is the visual relative localization based on the pattern detection by onboard cameras. The two main requirements, fast localization and onboard usability, require low computational demands for the image processing part. Therefore, we use an algorithm that allows for rapid detection and localization of simple circular patterns composed of concentric black and white circles of known diameter. Our algorithm



**Fig. 1** The localization pattern and the operational space of the relative localization system

(details described in [Krajník et al. \(2014\)](#)) outperforms common black-and-white pattern detectors in terms of speed by an order of magnitude while achieving similar precision and robustness. An example of the localization pattern with a sketch of the possible operational space of the relative localization module is depicted in [Fig. 1](#).

The detection algorithm searches the image for circular patterns using a combination of flood-fill techniques, on-demand thresholding, and on-the-fly statistics calculation. The statistical information gathered on-the-fly is used to test whether the continuous areas of pixels are likely to represent the searched pattern, and quickly reject false candidates. The main advantage of the method is that it can be initiated from any position in the image without a performance penalty, which allows for a simple implementation of pattern tracking. In a typical situation, the algorithm processes only the area that is occupied by the pattern itself, which results in a significant performance boost.

In the initial phase of the pattern detection, the image is scanned for a continuous segment of black pixels. Segmentation of the pixels into black and white classes employs an adaptive thresholding that ensures good performance of the algorithm under variable light conditions, which is especially important in real-world outdoor experiments. Once a continuous segment of black pixels is found by the flood-fill method, it is tested for minimum size and roundness. A pattern with outer and inner diameters  $d_o$ ,  $d_i$ , bounding box dimensions  $b_u$ ,  $b_v$  and area  $s$  is considered circular if its roundness  $\rho_{out}$  is smaller than a predefined constant  $\rho_{max}$ , i.e.

$$\rho_{max} > |\rho_{out}| = \left| \frac{\pi}{4s} b_u b_v \frac{d_o^2 - d_i^2}{d_o^2} - 1 \right|. \quad (1)$$

If a black region passes the roundness test, the flood-fill algorithm is initiated from the region's centroid in order to search for the inner white segment. Since the inner segments are circles and not rings, the roundness test for the inner white segments is simpler than (1):

$$\rho_{max} > |\rho_{in}| = \left| \frac{\pi}{4s} b_u b_v - 1 \right|. \quad (2)$$

Then, the concentricity of segments and the ratio of their areas are tested. After passing these tests, the positions of the segments' pixels  $u_i$ ,  $v_i$  that were stored during the flood-fill are used to calculate the ellipse center  $u$ ,  $v$  and covariance matrix  $\mathbf{C}$  as follows:

$$\mathbf{C} = \frac{1}{s} \sum_{i=0}^{s-1} \begin{pmatrix} u_i u_i & u_i v_i \\ u_i v_i & v_i v_i \end{pmatrix} - \begin{pmatrix} uu & uv \\ uv & vv \end{pmatrix}. \quad (3)$$

Note that  $u_i$ ,  $v_i$  are integers, and the computationally most expensive part of (3) is calculated using integer arithmetic.

Finally, the ellipse semiaxes  $e_0$ ,  $e_1$  are obtained from eigenvalues  $\lambda_0$ ,  $\lambda_1$  and eigenvectors  $v_0$ ,  $v_1$  of the covariance matrix  $\mathbf{C}$  as follows:

$$\begin{aligned} e_0 &= 2\lambda_0^{\frac{1}{2}} v_0, \\ e_1 &= 2\lambda_1^{\frac{1}{2}} v_1. \end{aligned} \quad (4)$$

Knowing the length of the ellipse semiaxes, the final segment test is performed:

$$\xi > |\pi e_0 e_1 s - 1|. \quad (5)$$

The constant  $\xi$  represents a tolerance value much lower than  $\rho_{max}$ , because the ellipse dimensions  $e_0$ ,  $e_1$  are obtained from the covariance matrix with the sub-pixel precision. If the detected segments satisfy (4), they represent the localization pattern, and the obtained information is used to calculate the spatial dimensions of the pattern.

To obtain the relative distance of the pattern, we calculate the image coordinates of the ellipse (co-)vertices and transform these into canonical camera coordinates. This transformation takes into account not only the camera length and optical center, but also its radial distortion. The transformed vertices are then used to calculate the centre and axes of the ellipse in the canonical camera form. From the vertices, we calculate a conic  $\mathbf{Q}$  such that the ellipse points  $u'$ ,  $v'$  satisfy

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T \mathbf{Q} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0. \quad (6)$$

Then, we calculate the eigenvalues  $\lambda_0$ ,  $\lambda_1$ ,  $\lambda_2$  and eigenvectors  $q_0$ ,  $q_1$ ,  $q_2$  of the conic  $\mathbf{Q}$  and use them to obtain the position of the pattern in space by the equations presented in [Yang et al. \(2012\)](#):

$$\mathbf{x} = \frac{d_o}{\sqrt{-\lambda_0 \lambda_2}} \left( q_0 \lambda_2 \sqrt{\frac{\lambda_0 - \lambda_1}{\lambda_0 - \lambda_2}} + q_2 \lambda_0 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_0 - \lambda_2}} \right), \quad (7)$$

where  $d_o$  is the circular pattern diameter.

**Table 1** Performance of the relative localization

Resolution	320 × 240		480 × 360		640 × 480		752 × 480	
FPS	60		46		30		27	
L (m)	$L_e$ (cm)	$L_\delta$ (%)						
0.5	1.2	0.1	0.9	0.4	3.6	0.6	4.3	1.2
1.0	0.1	0.1	0.3	0.1	1.5	0.3	2.3	0.6
1.5	0.6	0.1	0.9	0.1	0.1	0.1	0.4	0.5
2.0	0.2	0.5	1.0	0.1	1.1	0.1	0.8	0.1
2.5	2.0	0.2	0.0	0.2	0.7	0.2	1.8	0.2
3.0	1.2	0.2	0.7	0.3	0.0	0.2	4.0	0.2
3.2	3.0	0.2	1.8	0.7	3.5	0.2	2.3	0.2
3.5	–	–	1.8	0.9	0.8	0.2	2.2	0.2
4.0	–	–	–	–	5.4	0.4	3.3	0.4
4.5	–	–	–	–	2.7	0.3	2.5	0.2
5.0	–	–	–	–	2.4	0.6	3.4	0.6
5.5	–	–	–	–	6.6	0.5	6.5	0.7

### 3.1 Relative localization system performance

The aim of this section is to show the performance of the relative localization system and to empirically specify its operational space. For details and experiments identifying the sensor model, see [Faigl et al. \(2013\)](#) and [Krajnc et al. \(2014\)](#). Except the viewing angle, which can be clearly defined for each optical system (based on the lens), the most important factors that need to be considered in swarm stabilization and motion planning are the measurement accuracy and reliability. Both of these depend on the distance of the measured object, which provides a maximum measurable distance with acceptable system properties. The maximum measurable distance is then considered to be the range of the relative visual localization. This a priori obtained sensor model is crucial for the proposed multi-MAV motion planning and coordination. The detection reliability was measured with a pattern (with outer diameter  $d_o = 0.18$  m) placed on the camera optical axis at a distance  $L$  from the camera and compared with the ground truth (see [Table 1](#)).

Four different resolutions of the Caspa camera (used in all presented experiments) have been tested. The higher resolutions provide significantly better results, but at the cost of a decreasing measurement rate. The presented frame rates measured as Frames per Second (FPS) are obtained when the pattern is tracked (i.e. the blob is continuously detected without failures).<sup>4</sup> If the pattern is not detected on the basis of its position in the previous image and the whole picture needs to be processed, the frame rate sinks to 50–60% of

<sup>4</sup> For the 320 × 240 resolution, the frame rate is limited by the camera, which can provide images at 60 Hz.

the previous value. However, this lower value is not significant for the proposed control approach, since the measured relative distance is considered in the control loop only if the pattern is repeatedly detected. In the error and reliability data, we assume a systematic error proportional to the measured distance, which may be identified using the real distances and the Least Square Method (LSM). We present an average distance corrected by the systematic error (denoted as  $\hat{L}$ ), since this value is more relevant for control and stability than the actual measured values of the distance in swarm applications. The corrected error in the distance estimate is obtained as  $L_e = |L - \hat{L}|$ , where  $L$  is the ground truth. The standard deviation,  $L_\delta$ , presented as percentage of the measured distance, describes the repeatability of the measurements.

In addition to the variable resolution of the processed images, another aspect influencing the performance is the size of the pattern. As expected, with smaller patterns, the distance measurement error increases and the maximum measurable distance significantly decreases. For example, 480 × 360 image resolution allows the maximum measurable distances  $L_{max} = \{3.5, 2.0, 1.5, 1.0, 0.5\}$  m with pattern diameters  $d = \{18, 9, 8, 7, 5\}$  cm.

In addition to this analysis, we conducted an experiment to evaluate the performance of the vision-based relative localization and to characterize its operational limits in flight conditions (see [Fig. 2](#)). During the experiment, two MAVs hovering in approximately static positions aim to localize the third MAV, which is following a predefined trajectory (see [Fig. 3](#) for the ground truth positions of all MAVs obtained using VICON). All vehicles are equipped with cameras and identification patterns. This measurement was crucial for experimental evaluation of the limits of the space in which neighboring MAVs can be relatively localized.

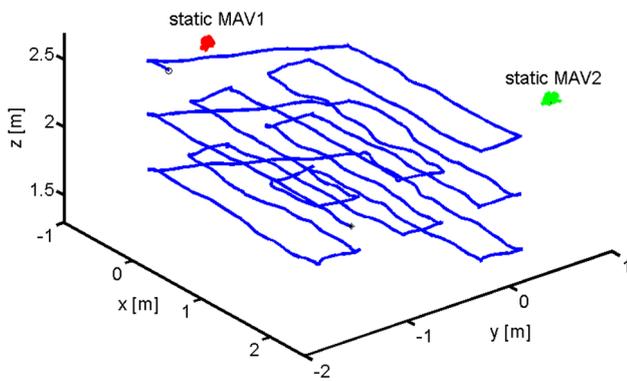
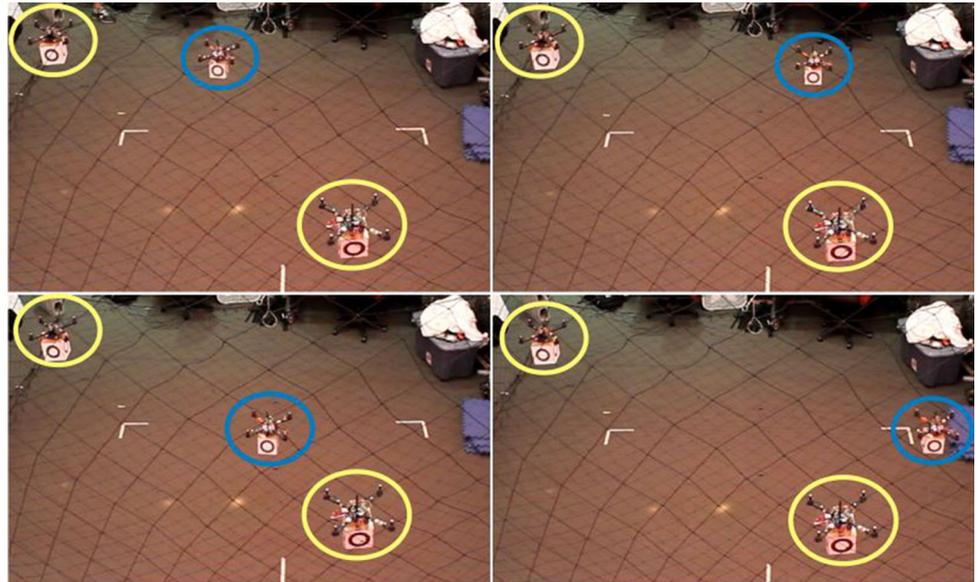
## 4 MAV model and control system

### 4.1 MAV model

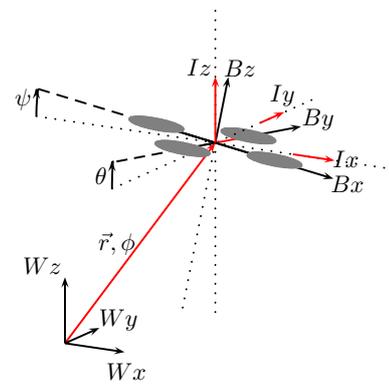
In the proposed approach, a suitable model of the quadcopters is essential for use in simulations of MAVs movement, in motion planning, and in inter-vehicle coordination. This ensures that the motion constraints are satisfied during the planning process and that the obtained solution is feasible for the MAV-group. In this work, we rely on a simplified, decoupled dynamical model described as follows:

$$\begin{aligned}
 \ddot{x}^W &= \frac{U}{m} \left( \sin \psi^I \cos \phi^W - \sin \theta^I \sin \phi^W \right), \\
 \ddot{y}^W &= \frac{U}{m} \left( \sin \theta^I \cos \phi^W + \sin \psi^I \sin \phi^W \right), \\
 \ddot{z}^W &= \frac{U}{m} \cos \theta^I \cos \psi^I - g,
 \end{aligned} \tag{8}$$

**Fig. 2** Snapshots from measurements of the operational space of the visual relative localization system, which is important for specifying constraints for the planning of swarm movement



**Fig. 3** Positions of MAVs captured by VICON during the experiment from Fig. 2

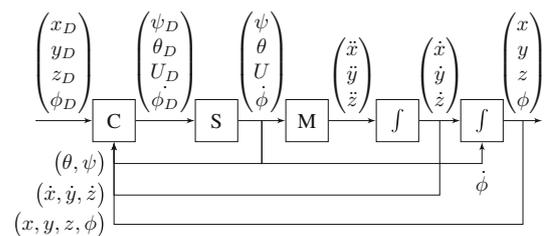


**Fig. 4** The reference frames used in description of MAV control scheme.  $W$  world frame,  $B$  body frame,  $I$  IMU frame

where  $\phi$  is the *yaw* angle,  $\theta$  is the *pitch* angle,  $\psi$  is the *roll* angle,  $U$  is collective thrust,  $m$  is the mass of the MAV, and  $g$  is the gravitational acceleration. We consider 3 frames of reference (Fig. 4). The world frame ( $W$ ) that is fixed in the workspace, the body frame ( $B$ ) that coincides with particular MAV and the IMU frame ( $I$ ) in which the *roll* and *pitch* angles are measured.

### 4.2 Control and stabilization scheme

The complete system used for stabilizing the group members at desired relative distances (keeping the required shape of the group) and for motion simulation at the motion planning level is depicted in Fig. 5. The system consists of a controller (block C), the stabilization unit (S), and the model from (8). For deployment of the system, parameters of the linear model are identified using Least Squares Method from the measured flight data.



**Fig. 5** Scheme of the system together with a controller. Position, its derivatives and  $\phi$  are meant in the world frame,  $\theta$  and  $\psi$  in the IMU frame

The flow of data within the proposed swarm stabilization system is shown in Fig. 6. The control scheme is suited for the MikroKopter quadcopter platform used for experimental evaluation of the visual relative localization based stabilization of the multi-MAV system in Sect. 5. The commercially available MikroKopter set includes a proprietary attitude stabilization board (Flight-CRTL) using an onboard

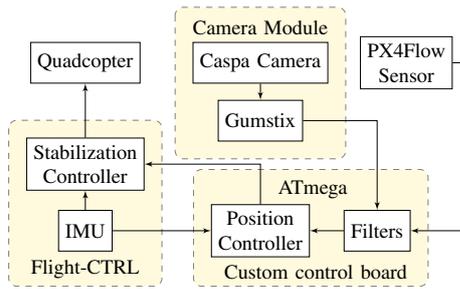


Fig. 6 Scheme of the data flow

Inertial Measurement Unit (IMU) for control feedback. The vision-based stabilization is built upon this lowest level and controls the relative positions between neighboring swarm entities. The solution is based on a custom board with the ATmega  $\mu$ -controller, which also serves as a communication hub between all onboard modules. Data received from the visual system, together with the output from IMU and from the PX4Flow smart camera sensor, serve as the control feedback at this level. The PX4Flow sensor provides information on the altitude and velocities relative to the surrounding environment. This setup is crucial for suppressing the motion oscillations within the group that are caused by the cumulative position error. The IMU provides angles  $\theta^I$ ,  $\psi^I$ , PX4Flow provides  $\dot{x}^W$ ,  $\dot{y}^W$ , and  $\dot{z}^W$ , and the camera module provides relative coordinates to the particular neighbor  $x_n^I$ ,  $y_n^I$ ,  $z_n^I$ . The position controller computes the desired control outputs  $\phi_D^I$ ,  $\psi_D^I$ , and  $U_D^I$ .

Three controllers are integrated in block C; the forward and lateral controllers are identical due to the system decoupling. The following equation denotes the controllers outputs:

$$\begin{aligned} \theta_D^I &= K_P e_x + K_D \frac{de_x}{dt} + K_I \int_0^t e_x d\tau, \\ \psi_D^I &= K_P e_y + K_D \frac{de_y}{dt} + K_I \int_0^t e_y d\tau, \\ U_D &= L_P e_z + L_D \frac{de_z}{dt} + L_I \int_0^t e_z d\tau, \end{aligned} \quad (9)$$

where  $K_P$ ,  $K_D$ ,  $K_I$ ,  $K_A$ ,  $L_P$ ,  $L_D$ , and  $L_I$  denote the controller constants that need to be identified during the system setup. The control errors,  $e_x^I$ ,  $e_y^I$ , and  $e_z^I$ , define the difference in the IMU coordinate system between the actual position of the controlled MAV and the desired position. The desired position is determined by the relative position to the circular pattern (resp. patterns) measured by the onboard visual localization, and by the desired relative position to the pattern (resp. patterns) given by a high-level planning method (see Sect. 6 for examples of various planning approaches). The desired position may be dynamic in the sense of moving localization pattern (resp. patterns), which is placed on

neighboring MAV (resp. MAVs), and/or in the sense of altering desired relative positions. In experiments with a static hovering MAV, the desired position is determined relatively to an initial position by the PX4Flow sensor.

### 5 Experimental verification of the system with visual relative localization in control feedback

In the first experiment, which was performed to demonstrate the performance of the control scheme, a single MAV is stabilized at a fixed relative distance to a static localization pattern. In this case, the MAV is also equipped with the localization pattern for its off-line global localization using an external fixed camera (see Fig. 7). The data from the external camera is used for experiment recording and off-line analysis, while the MAV control relies on onboard sensors only. The results from this external camera are plotted in Fig. 8. The mean control deviation from the desired equilibrium was 0.11, 0.12, and 0.04m with standard deviations 0.14, 0.15, and 0.05m, measured in the  $x$ ,  $y$ , and  $z$  coordinates. The slight motion oscillations are caused mainly by the noise in the sensor data. See Fig. 9 for data from the camera module with Gumstix, optical flow obtained from the PX4Flow sensor, and the outputs of the designed controllers. A detailed view of a sample from Fig. 9 is presented in Fig. 10.

The stabilization of neighboring vehicles with a predefined mutual distance is shown in the outdoor experiment in a former pit mine (see Fig. 11 for pictures from the experiment). The experiment verifies the ability of the system to follow a moving “leader” MAV with an attached localization pattern. The first MAV (the leader) is controlled along a pre-planned trajectory based on the visual odometry from a PX4Flow sensor (the pose estimate is obtained by integrating

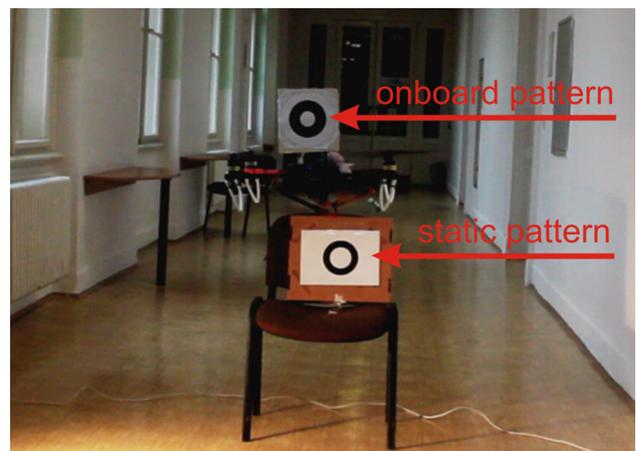
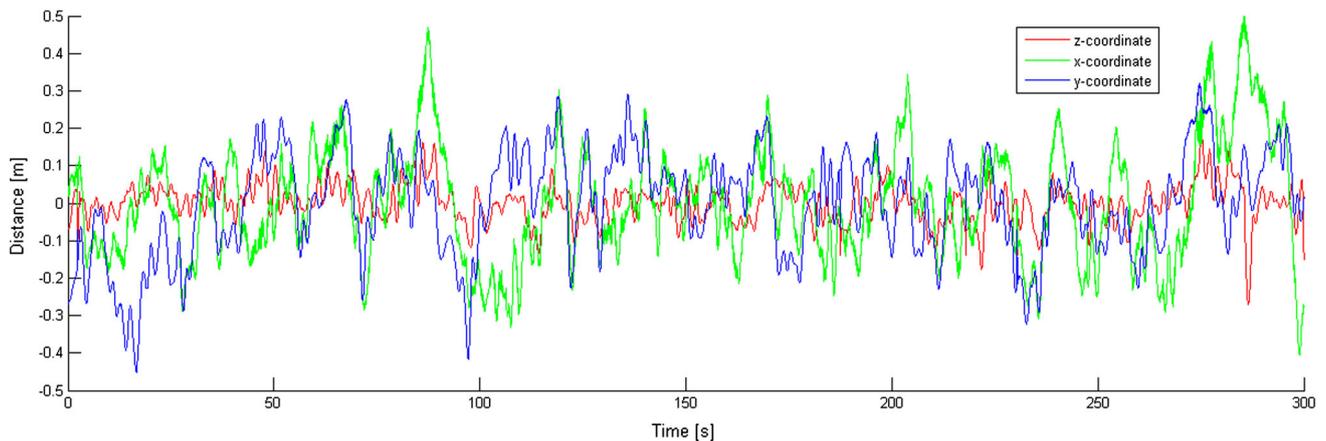
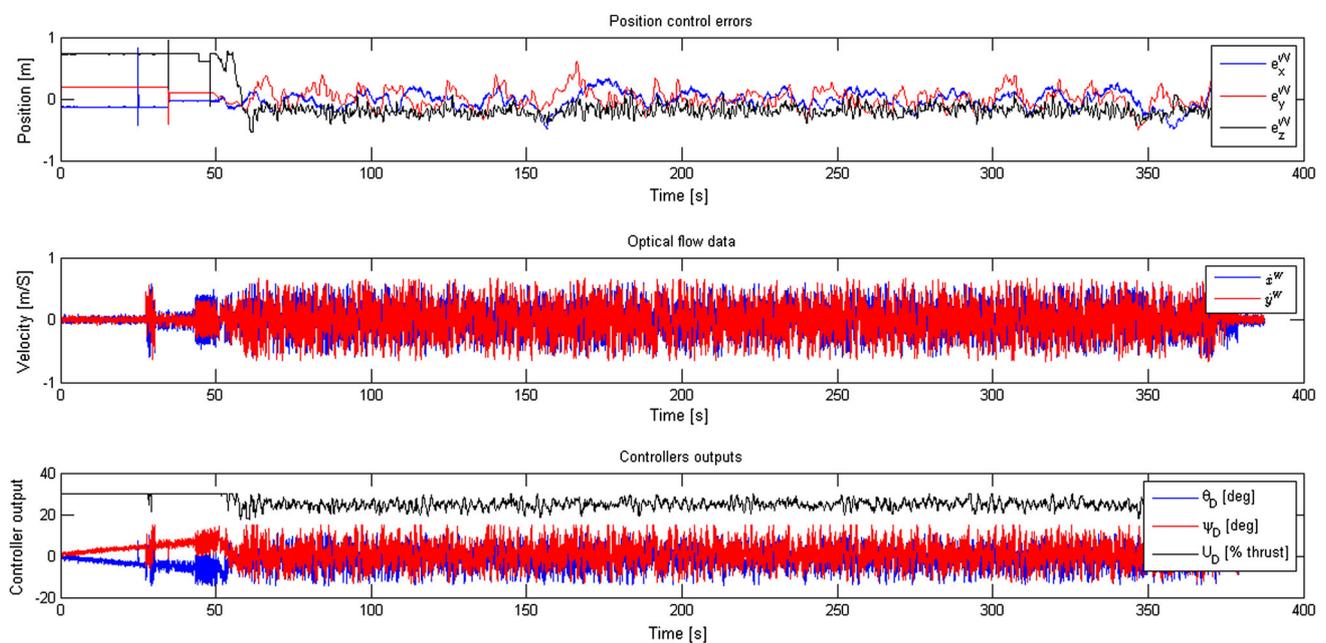


Fig. 7 MAV stabilized at a fixed relative distance to the static pattern. The onboard pattern is used for external localization, which gives the ground-truth for experiment evaluation



**Fig. 8** Deviation from the desired equilibrium located at 2.5 m from the static circular pattern (experiment in Fig. 7). Data obtained from the record of the external camera



**Fig. 9** Sensor data and controller output during the experiment shown in Fig. 7. The first picture presents the output (in  $x$ ,  $y$ , and  $z$  coordinates) from the onboard relative localization module. The output from

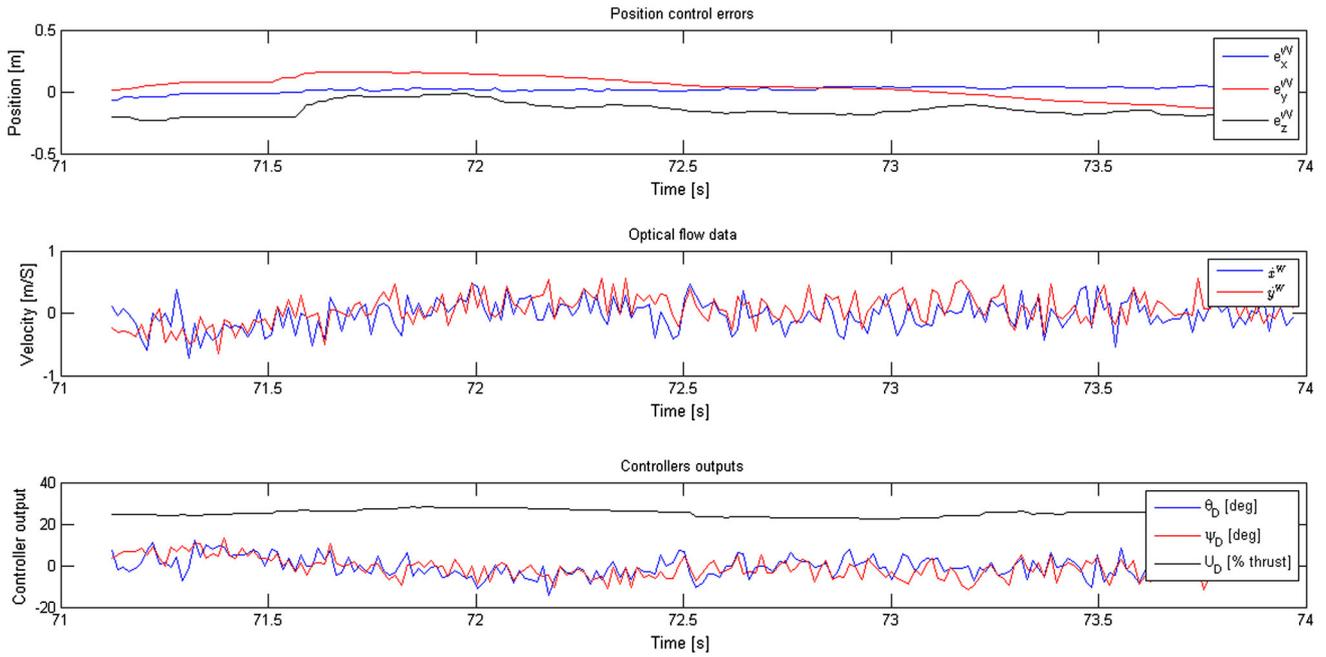
the PX4Flow sensor is shown in the second picture, while the outputs of the controllers are presented in the third plot

the optical flow from the down-looking camera). The second MAV follows the first one at a fixed desired spacing based on feedback from the onboard visual relative localization system. The elevation above the slope of the mine is fixed for both MAVs based on the feedback from the sonar.

The same experiment was repeated in the presence of wind on a flat road. We demonstrated that the system is also able to stabilize the formation with minimal influence of the wind on the performance of the system (see Fig. 12 for pictures taken during the experiment). Videos of these experiments can be downloaded from [Multimedia \(2015\)](#).

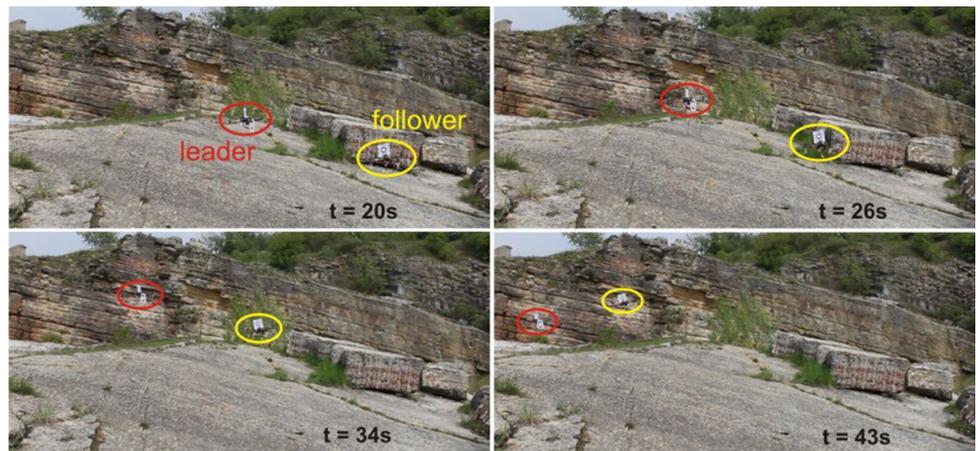
The aim of the experiment presented in Fig. 13 is to demonstrate flying in strings of the relatively stabilized MAVs using the proposed system. In the case of stabilization of large groups of MAVs, it is difficult to ensure that all MAVs are stabilized directly to the same MAV (a common leader). Naturally, more complex networks arise in swarms or formations of MAVs, in which always some robots need to be stabilized relatively to neighbors that are already stabilized relatively to another robot, etc.

Data in Table 2 with results of experiments from Fig. 13 show only a slight increase of motion oscillation of an MAV

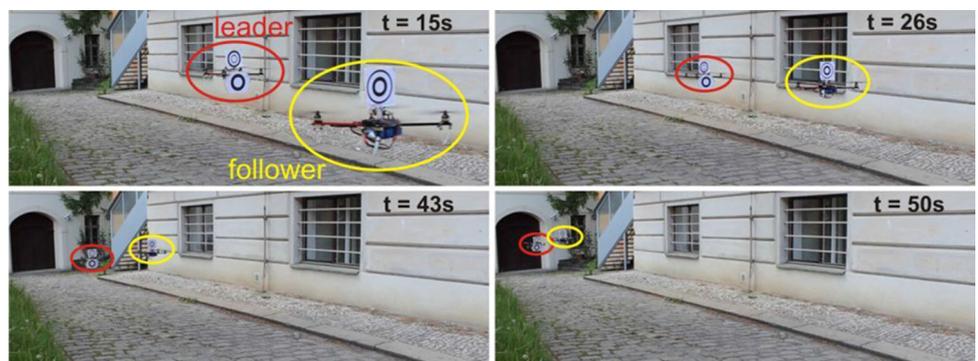


**Fig. 10** Zoomed view on data from Fig. 9

**Fig. 11** Leader-follower formation flying on a slope surface

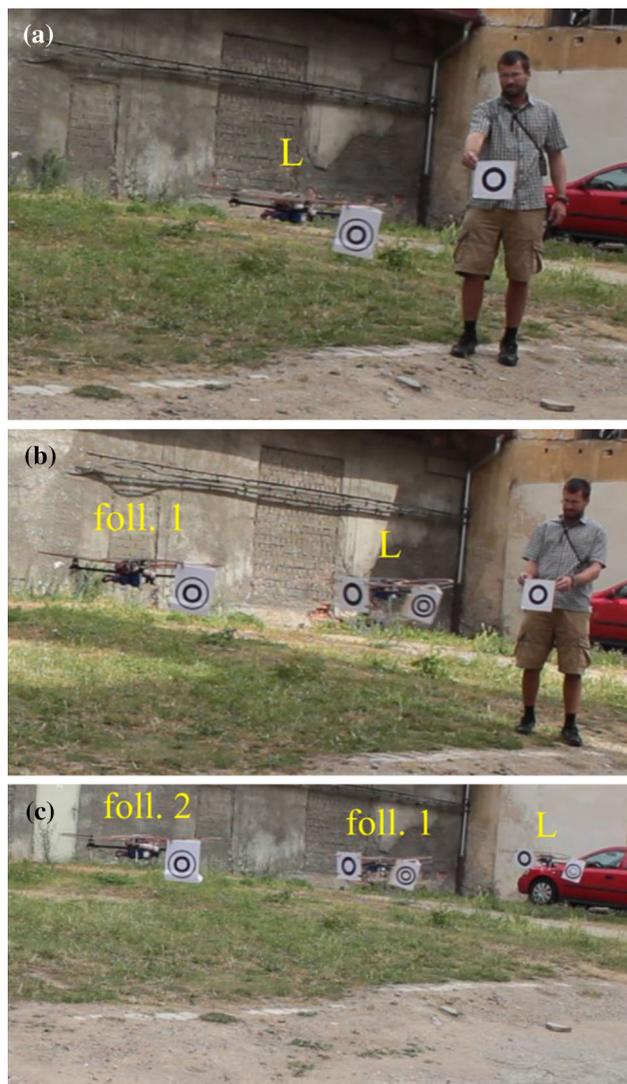


**Fig. 12** Leader-follower formation in windy conditions



following another MAV in a comparison with the situation where the same MAV is stabilized relatively to a static pattern. The moving pattern introduces additional noise into

the measurement of the relative localization and furthermore slightly decreases performance of the vision system as described in Faigl et al. (2012). See the second and the



**Fig. 13** Verification of the proposed approach for stabilization of MAV-groups based on the visual relative localization. Experiments show (see data in Table 2) that due to the measured relative speed between the MAVs and surface, which is employed in control feedback, position oscillations do not increase with the size of the group being relatively stabilized **a** 1-MAV. MAV relatively stabilized to a static pattern using the visual relative localization in control feedback **b** MAV pair. The MAV on the right (follower 1) is relatively stabilized to a static pattern. The second MAV (follower 2) is relatively stabilized to the follower 1 **c** 3 MAVs in line. The MAV in the most right (leader) hovers on spot using the PX4Flow sensor. The middle MAV (follower 1) is relatively stabilized to the leader. The MAV in the most left (follower 2) is relatively stabilized to the follower 1

fourth columns of the table that show flight performance of the MAV with id 3 in two different roles: (1) as a leader stabilized relatively to the static pattern, (2) as a follower stabilized relatively to another leader. Similar comparison can be seen in the third and the sixth columns for the MAV with id 2.

In addition, the motion oscillations of the MAV with id 3 are comparable in the **MAV pair** experiment (fourth column

**Table 2** Tests of the flight performance of MAVs in a static platoon-like formation hovering on a spot

Exp. type	1-MAV	MAV pair		3 MAVs in line		
MAV id	3	2	3	1	2	3
MAV role	L	L	fol. 1	L	fol. 1	fol. 2
Mean error (cm)	13.1	19.6	14.6	14.7	20.0	14.9
Stand. dev. (cm)	6.9	11.2	8.2	7.8	12.5	8.3

Snapshots from the experiments are shown in Fig. 13 and videos are available at [Multimedia \(2015\)](#). The mean error and standard deviation are measured from a fixed equilibrium by the external video system ([Krajník et al. 2014](#)). The equilibrium is defined by a fixed relative position to the onboard pattern in case of followers, by a fixed relative distance to the static pattern in case of the leader in the **1-MAV** and **MAV pair** experiments, or the equilibrium is set as a fixed initial position in case of the leader in the **3 MAVs in line** experiment. The relative distance between neighbouring MAVs and between the MAV and the static pattern was 2.5 m in all experiments. The statistics were obtained from approximately 900 samples for each of the experiments. Approximately 30s long records with the rate 30 frames per seconds were analysed for each experimental flight

of the table), where the MAV with id 3 acts as the first follower that is directly stabilized to the leader, and in the **3 MAVs in line** experiment (seventh column), where the MAV with id 3 acts as the second follower that is indirectly stabilized to the leader; over another follower. Taking into consideration the data that describes the motion performance of the MAVs with id 2 and 3, one can see that the difference between these two robots is more significant than the difference between motion of MAV 3 in roles of the follower 1 and 2. Even though all MAVs are controlled by the same systems with the same parameters and sensors, the small differences during their manufacturing influence the flight performance. The independence of the control performance from the number of robots in the controlled string was observed also in case of moving relatively stabilized formations (see Figs. 11 and 12), which is crucial in most of the applications requiring the group stability. In all experiments in Fig. 13, the relative distance between neighbouring MAVs and between the MAV and the static pattern was 2.5 m.

In addition to these outdoor experiments, the vision system was tested in numerous experiments of various multi-robot applications with the VICON motion capture system as a reference. The precision and reliability of the external motion capture system is sufficient to consider the obtained data as the ground truth ([Michael et al. 2010](#)). Details on the tested multirobot scenarios and the obtained results are given in Sect. 6. In the formation driving experiments (Figs. 16, 17, 18, 19, 20, 21), where the relative distance between quadcopters is almost constant, the reliability of the measurement is approximately 98 %. If we exclude the outliers caused by identification of a “wrong” MAV, or by occlusions, the mean error of the relative distance is 1.1 cm (with standard

deviation 0.9 cm) at a distance of 1.5 m between neighboring MAVs. The maximum error is always  $< 4$  cm in these experiments. In the swarm experiments (Figs. 24, 25, 28, 29, 30), the relative distance between quadcopters differs from 1–2.5 m and the measurement reliability is approximately 95 %. The mean error of the relative distance is 1.3 cm (with a standard deviation of 1.7 cm). This slightly worse performance is caused by a longer relative distance between the robots and by their relative motion, which may spoil some pictures captured by the onboard cameras (the camera and the objects in the images move independently). The frame rate exceeds 30 frames per second if the images are processed online and pictures are not stored in the memory of the camera module. If unprocessed images need to be stored for a later evaluation of the experiment, the rate has to be reduced to 10–15 fps.

## 6 Multi-robot scenarios demonstrating the practical usability of the system

The aim of this section is to present examples of practical utilization of the proposed system. We have chosen three general approaches to MAV deployment that are currently solved in state-of-the-art literature (leader-follower formation flying, swarm-inspired stabilization, and multi-MAV surveillance) to show how these scenarios can be solved using the proposed system. Each of the multi-robot scenarios proposes a different approach to motion planning and coordination of the MAV-group. The common challenge lies in the necessity to satisfy the MAV motion constraints and the constraints imposed by the relative localization. Therefore, the description of these methods is focused on integration of the relative localization constraints into the planning algorithms. In addition, in the experimental parts of this section, the tests with the VICON motion capture system in control feedback verify that the trajectories of the MAVs obtained by the proposed high-level planning systems are feasible for real MAV-groups. This means that the obtained trajectories respect the MAV dynamics, the localization constraints, and the environment constraints (obstacles and no-fly zones). Besides, the aim of these experiments is to evaluate the performance of the camera module and the localization algorithm in multi-MAV applications. The results of the experiments are compared with the ground truth and are attached to the description of each method in the following subsections.

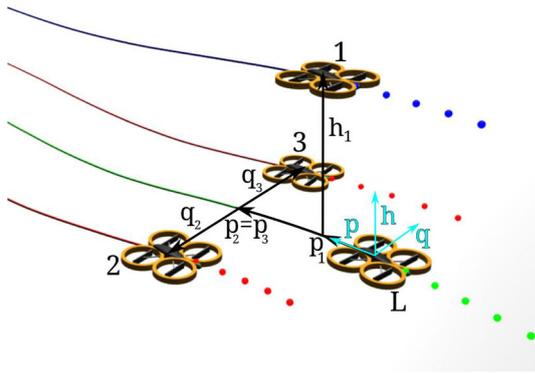
### 6.1 Scenario 1: leader-follower formation flight

In this scenario, a formation of multiple MAVs reaches a desired target region in a complex environment with obstacles, while maintaining predefined relative positions. The desired shape of the formation can be temporarily changed

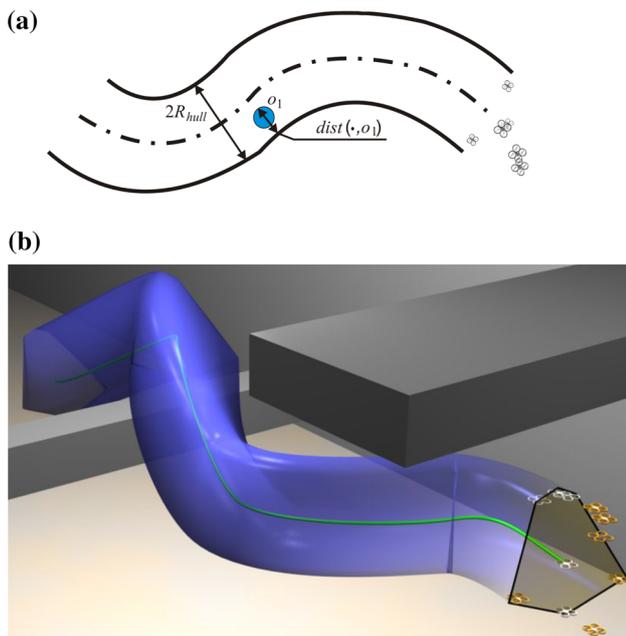
only if it is enforced by environmental constraints (e.g. in narrow passages). The proposed formation control mechanism is suited for the real-world deployment of autonomous robots relying on the onboard visual relative localization, which brings additional movement constraints to the MAV team. The method is based on a leader-follower technique, where the team of robots is stabilized by sharing knowledge of the leader's position within the formation (see the original leader-follower approach (Barfoot and Clark 2004) designed for a group of ground robots (UGVs) and the extension of the leader-follower approach for heterogeneous MAVs-UGVs teams in Saska et al. (2014c, d) for details). The method presented in this section is an extension of our work introduced in conference paper (Saska et al. 2014b), where only simulation results were presented and where the requirements on the onboard relative localization necessary for the HW experiments, which is the main contribution of this paper, were not included.

We do not rely on following a given trajectory, as in most of the state-of-the-art methods (Chao et al. 2012; No et al. 2011; Liu et al. 2011). We propose to integrate the stabilization of followers in the desired positions behind the leader together with the trajectory planning into a desired goal area with obstacle avoidance ability for the entire formation. The global trajectory planning is directly integrated into the formation control mechanism, which is important for finding a feasible solution for the proposed approach using the relative visual localization of the team members. For stabilization of the MAV group via the onboard relative localization, it is crucial that direct visibility between team members is not interrupted by an obstacle. Thus, in the trajectory planning process, direct visibility is ensured by penalizing collisions between obstacles and a 2D convex hull of the positions of followers, which represents the 3D formation. The 2D convex hull is obtained as a projection of positions of followers into a plane that is orthogonal to the trajectory of the virtual leader in its current position (see Fig. 15a). For the obstacle avoidance function described in Eq. (11), the convex hull is dilated by a safety radius, which is considered around each MAV, to keep obstacles at a desired distance from the followers. The trajectory planning into the desired goal region and the immediate control of the formation is then integrated in a single optimization process with this obstacle avoidance function. The method can continuously respond to changes in the vicinity, while keeping the cohesion of the immediate control inputs with the directions of movement of the MAV formation in the future.

In the algorithm, followers follow the trajectory of the leader at distances defined in the  $p$ ,  $q$ ,  $h$  curvilinear coordinate system, as visualized in Fig. 14. The position of each follower  $i$  is uniquely determined: (1) by states  $x_L(t_{p_i})$  in the traveled distance  $p_i$  from the actual position of the leader along its trajectory, (2) by the offset distance  $q_i$  from the



**Fig. 14** The desired shape of the formation described in curvilinear coordinates



**Fig. 15** An example of the dilated convex hull projected along a trajectory. This trajectory would be infeasible for the formation stabilized by the presented approach, since an obstacle appears inside the convex hull a contours of the convex hull projected along the leader's trajectory. An obstacle is denoted inside the hull to clarify the meaning of function  $dist(X_L(\cdot), o_l)$  b 3D visualization of the convex hull projected along the leader's trajectory (the circular obstacle is hidden inside the hull)

leader's trajectory in the perpendicular direction and, 3) by the elevation  $h_i$  above the leader's trajectory, as follows:

$$\begin{aligned} x_i(t) &= x_L(t_{p_i}) + (-q_i \sin(\theta_L(t_{p_i})), q_i \cos(\theta_L(t_{p_i})), h_i)^T, \\ \theta_i(t) &= \theta_L(t_{p_i}), \end{aligned} \quad (10)$$

where  $x_L(t_{p_i})$  is the position of the leader at the time when the virtual leader was at traveled distance  $p_i$  behind the current position and  $\theta_L$  is the yaw of the leader at time  $t_{p_i}$ .

The short-term trajectory planning responding to the local workspace of the robots and the long-term trajectory planning

providing a plan to the target location are solved together in a single optimization step. The leader's trajectory encoded into a vector of constant control inputs at time  $t$  is used as the optimization vector  $X_L(t) = [v_{L,1}, v_{L,1}, k_{L,1}, \dots, v_{L,N}, v_{L,N}, k_{L,N}, v_{L,N+1}, v_{L,N+1}, k_{L,N+1}, \delta_{L,N+1}, \dots, v_{L,N+M}, v_{L,N+M}, k_{L,N+M}, \delta_{L,N+M}]$  to include both, the local and the global trajectory planning. The vector  $X_L(\cdot)$  consists of the normal velocity  $v_L(m \cdot s^{-1})$ , the tangential velocity  $v_L(m \cdot s^{-1})$ , the curvature  $k_{L,\cdot} [m^{-1}]$ , and the length of the time interval  $\delta_L(s)$ . The curvature  $k_{L,\cdot}$  of the trajectory followed by the leader is constant within each control segment and may vary along the whole trajectory. The time interval  $\delta_{L,j}$  is constant if  $j \in \{1 \dots N\}$  and becomes variable if  $j \in \{N+1 \dots N+M\}$ . The constant time interval is denoted as  $\Delta t$  and is set as  $\delta_{L,j} := \Delta t = 0.1s, j \in \{1 \dots N\}$ , in the experiments.  $N$  is the number of transition points in the short control horizon with the constant  $\Delta t$  between the transition points.  $M$  is the number of transition points in the long planning horizon with variable  $\delta_{L,j}$  between the transition points. The trajectory is obtained from the optimization vector by applying the constant control inputs into the model in Sect. 4.

The leader's control problem with the obstacle avoidance ability can then be transformed to minimization of the multi-objective cost function  $F_L(X_L(\cdot))$  as follows:

$$\begin{aligned} F_L(X_L(\cdot)) &= \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{dist(X_L(\cdot), o_l)}{dist(X_L(\cdot), o_l) - R_{hull}} \right\} \right)^2 \\ &+ \alpha \sum_{j=N+1}^{N+M} \delta_{L,j}. \end{aligned} \quad (11)$$

The first part of the function prevents the formation from colliding with obstacles. The number of considered obstacles is denoted as  $n_o$ , and  $o_l$  denotes the  $l$ -th obstacle. Its value is zero if all obstacles are outside the projected convex hull, which is formed by MAVs following the leader in their desired positions within the formation.  $R_{hull}$  is the radius of the convex hull depicted in Fig. 15a. The value goes to infinity as an obstacle approaches into the center of the hull. This ensures that direct visibility between the robots will not be broken by an obstacle located among them. The value of the second term is based on an estimation of the total time to reach the desired target region, which must be minimized. The influence of the obstacle avoidance function and the endeavour of the trajectory planning to reach the target region in minimum time are weighted by constant  $\alpha$ .

To ensure feasibility of the obtained solution, the optimization process is subject to a set of constraints. The first constraint, which is necessary for a convergence of the formation driving process into the desired equilibrium (the desired target region), requires that the final transition point of the

planning horizon is inside the target region. In addition, control inputs have to be constrained since the planning approach for the leader must respect the constraints given by mechanical capabilities of all followers. The admissible control set for the leader can be determined by applying the leader-follower approach for  $i = 1, \dots, n_r$  as  $k_{L,max} = \min(k_{i,max}/(1 + q_i k_{i,max}), k_{L,min} = \max(-k_{i,max}/(1 - q_i k_{i,max}), v_{L,max} = \min(v_{i,max}/(1 + q_i k_L), v_{L,min} = \max(v_{i,min}/(1 - q_i k_L)), v_{L,max} = \min(v_{i,max}), v_{L,min} = \max(v_{i,min})$ , where  $k_{i,max}$ ,  $v_{i,max}$ ,  $v_{i,min}$ ,  $v_{i,max}$  and  $v_{i,min}$  are limits on the control inputs of the  $i$ -th follower. These restrictions must be applied to satisfy different values for the curvature and the speed of the robots in different positions within the formation. For example, the robot following the inner track during turning goes more slowly but with a bigger curvature than the robot further from the center of the turning, due to the fact that the followers turn around the same Instantaneous Center of Curvature (ICC) and at the same angular speed.

The states specified by the trajectory of the leader of the formation obtained as a result of the optimization are transformed for the followers using the transformation in Eq. (10). These desired states are used for the trajectory tracking algorithm with the obstacle avoidance function, which enables responses to events that occur in the environment behind the actual position of the leader. The trajectory is encoded into a vector of constant control inputs and it is used as the optimization vector  $X_i(\cdot) = [v_{i,1}, v_{i,1}, k_{i,1}, \dots, v_{i,N}, \tau_{i,N}, k_{i,N}]$  for the  $i$ -th follower. For the motion planning of the followers, only the short-term horizon with a constant sampling time is employed. The discrete-time trajectory tracking for each follower is transformed to minimization of the multi-objective cost function  $F_i(X_i(\cdot), X_L)$  subject to set of constraints as follows:

$$\begin{aligned}
 F_i(X_i(\cdot), X_L) &= \sum_{l=1}^{n_o} \left( \min \left\{ 0, \frac{\text{dist}(X_i(\cdot), o_l) - r_s}{\text{dist}(X_i(\cdot), o_l) - r_a} \right\} \right)^2 \\
 &+ \sum_{j \in \bar{n}_n} \left( \min \left\{ 0, \frac{d_{i,j}(X_i(\cdot), X_j) - r_s}{d_{i,j}(X_i(\cdot), X_j) - r_a} \right\} \right)^2 \\
 &+ \beta \left( \sum_{j=1}^N |d_{x_{i,j}} - x_j|^2 + \sum_{j=1}^N (d_{\theta_{i,j}} - \theta_j)^2 \right). \quad (12)
 \end{aligned}$$

The first sum penalises solutions with a distance to an obstacle less than the detection radius  $r_s$ . The penalty function goes to infinity as an obstacle approaches a distance equal to the avoidance radius  $r_a$ . If the distance between an obstacle and the trajectory is less than  $r_a$ , the solution is considered infeasible (the obstacle proximity constraint of the optimization is violated). In the second sum of the cost function, the other members of the team are consid-

ered as dynamic obstacles in case of an unexpected behavior of defective neighbors deviating from their desired positions within the formation. Function  $d_{i,j}(X_i(\cdot), X_j)$  returns the minimal distance between the planned trajectory of follower  $i$  and the plan of other followers  $j \in \bar{n}_n$ , where  $\bar{n}_n = \{1, \dots, i - 1, i + 1, \dots, n_r\}$ .

The last term of the cost function penalizes a growing Euclidean distance between the desired positions  $d_{x_{i,j}}$ ,  $j \in \{1 \dots N\}$ , obtained from the actual leader's trajectory  $X_L$ , and the positions of the  $i$ -th follower. Also the differences between the desired yaw angles  $d_{\phi_{i,j}}$ ,  $j \in \{1 \dots N\}$ , and the actual yaw of follower  $i$  are penalized. The influences of the obstacle avoidance function and the trajectory following term are weighted by constant  $\beta$ . Values  $\alpha = 1$  and  $\beta = 1$  were used in all experiments in this article, but the approach does not require fine tuning of these parameters and the same values can be efficiently used in different scenarios.

In addition to the constraint, which is satisfied if the distance between the trajectory corresponding to the particular solution of the optimization and all obstacles is greater than  $r_a$ , the control inputs are constrained to satisfy the motion constraints of the employed MAVs (limits on forward and ascending velocities etc.).

### 6.1.1 Experimental evaluation of the planning technique in flight conditions

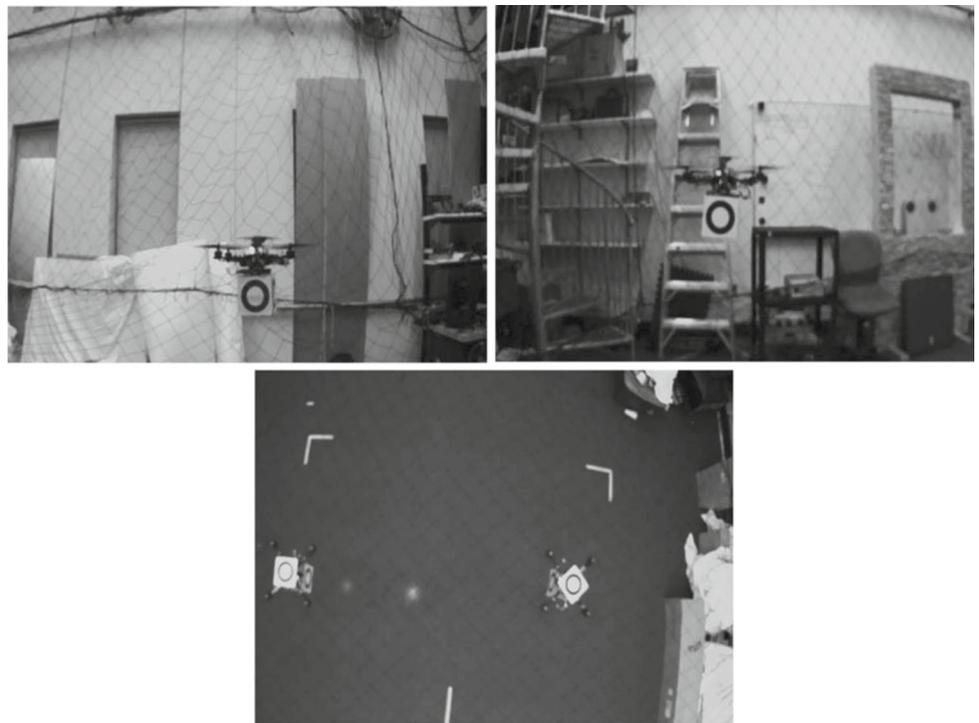
In this section, the feasibility of results of the formation planning approach is verified by experiments with multiple MAVs. Two virtual obstacles, the no-fly zones depicted in Fig. 18, are considered in the workspace to demonstrate the obstacle avoidance ability. Three MAVs equipped with the visual relative localization modules (Faigl et al. 2013) are stabilized in triangular and line formations. In the triangular formation (see Figs. 16, 17, 18, 19), an MAV with a camera pointed down is flying above two other MAVs with side looking cameras. The experiment with the line formation of three MAVs equipped with cameras oriented to the side (see Figs. 20, 21) is realized repeatedly to show the robustness of the method. The formation flies twice to the target region and back to the initial position. The initial position from the first flight is considered the centre of the target region for the return flight, etc. The multi-criteria optimization problems defined in Eqs. (11) and (12) were solved by sequential programming method (CFSQP toolbox Version 2.0) in the experiments. All MAVs are equipped with identification circle patterns for fast relative localization. The independent motion capture system (VICON) is used as a ground truth to evaluate the performance of the visual relative localization during the formation driving experiments. Complete records of the experiments are available in Multimedia (2015).

The main purpose of the experiment was to verify the ability of the system to relatively localize MAVs in a compact

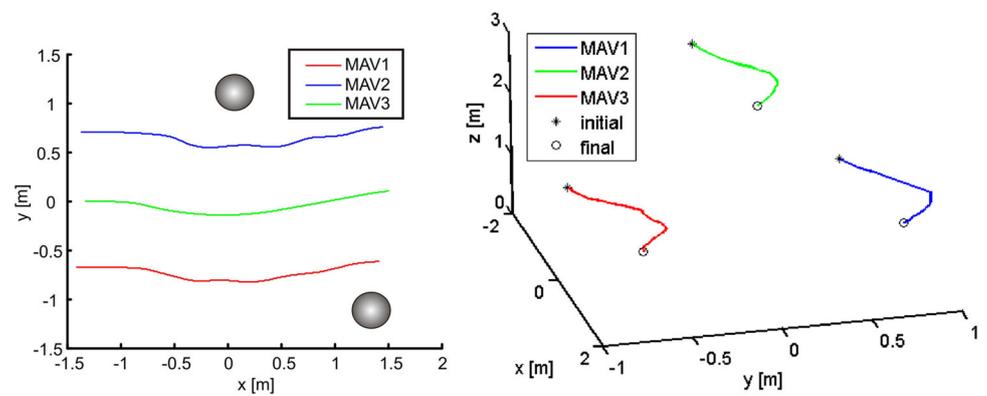
**Fig. 16** Experiment with a triangular formation of 3 MAVs

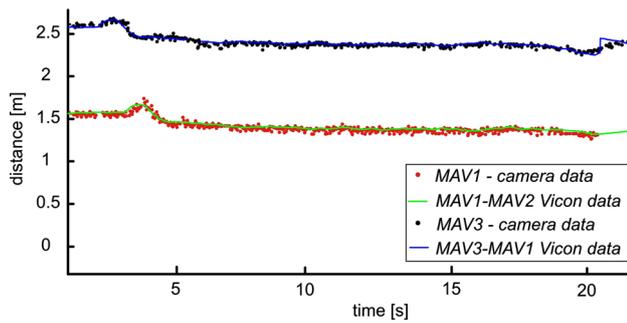


**Fig. 17** Examples of pictures obtained by the onboard cameras for the relative localization (exp. in Fig 16)

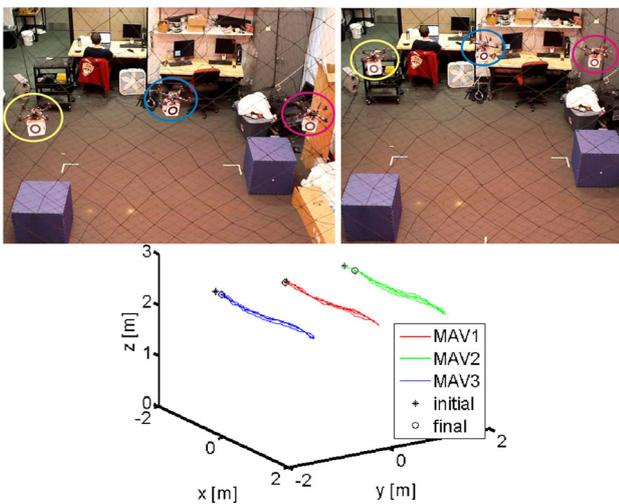


**Fig. 18** Trajectories of MAVs in the experiment from Fig. 16 recorded by the VICON system. Positions of the virtual obstacles are denoted by the *circles*





**Fig. 19** Relative distances between MAV1-MAV2 and MAV3-MAV1 in the experiment from Fig. 16. The *dots* correspond to raw data obtained from the visual relative localization, and the *curves* are reference values provided by the VICON motion capture system

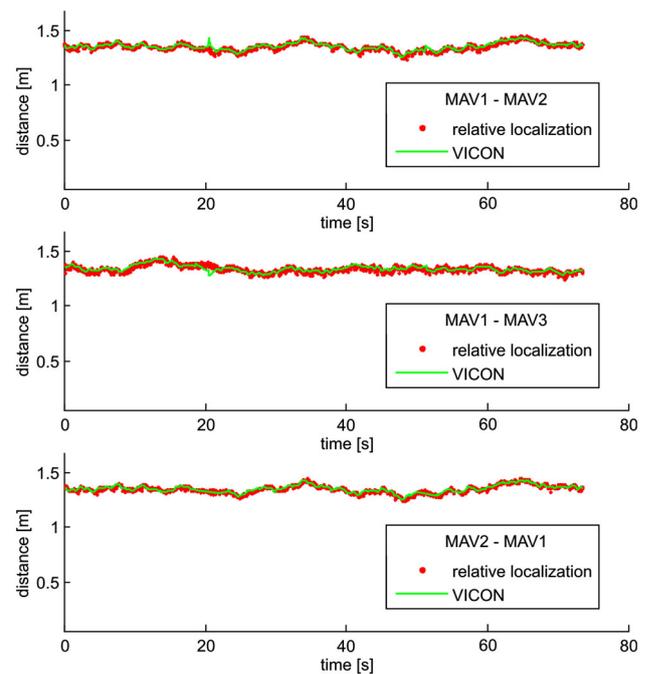


**Fig. 20** Experiment with the line formation of 3 MAVs

formation using VICON as a ground truth, and therefore, the experiment was realized in simple laboratory conditions. Nevertheless, the trajectory planning and formation stabilization mechanisms may be efficiently employed in more complex situations as was shown in our previous research with Unmanned Ground Vehicles (UGVs).

See our results (Saska et al. 2009), where performance of trajectory planning for UGV formations is shown in a complex office-like environment. In Saska et al. (2013a), complex maneuvers of the formation controlled by the MPC were presented. Usage of the MPC-based stabilization and trajectory planning in task of airport snow shoveling by fleets of autonomous ploughs is presented in Saska et al. (2010) with stability analyses in Saska et al. (2013b). The work in Saska et al. (2016) is focused on testing the ability of the approach to avoid dynamic obstacles by integration of its motion prediction into the MPC trajectory planning.

The approach presented in this paper is an extension of the methods designed for UGVs taking into account constraints of the visual relative localization system, which is used for



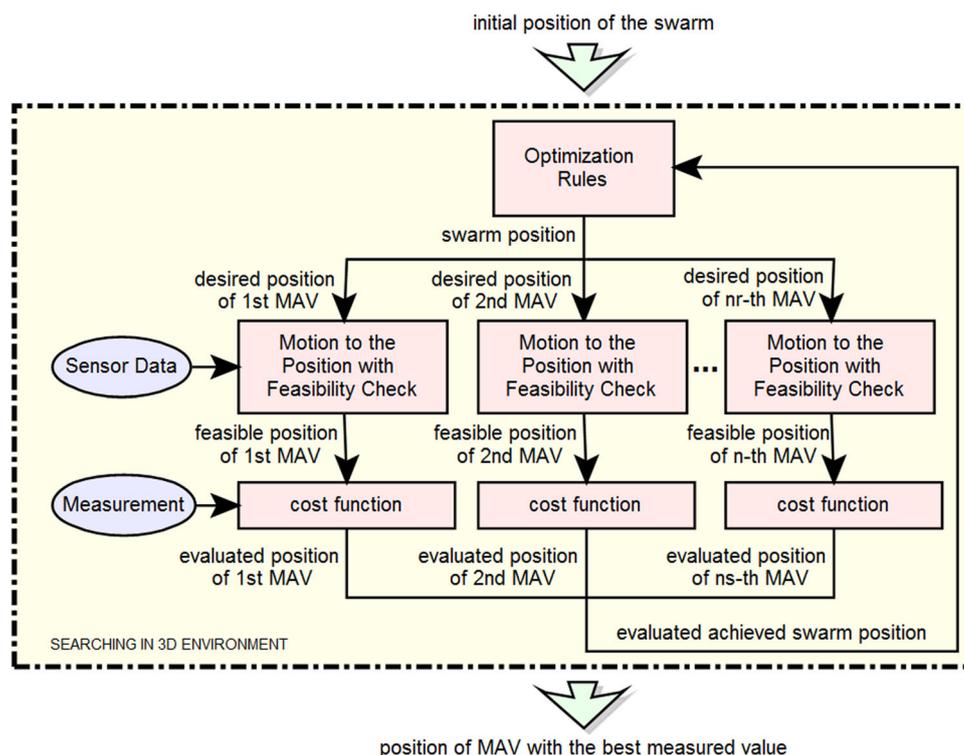
**Fig. 21** Relative distances between MAVs obtained by the onboard cameras for formation stabilization. VICON data record is plotted as a reference

stabilization of MAV formations in 3D shapes. Due to the employed convex hull that represents the entire formation in the planning process, the trajectory planning ability of the system is not limited and it achieves a similar performance as was presented for UGVs. In case of limited computational power onboard of MAVs, where the complexity of the optimization is increased by the third dimension, the applicability of the method in real-time could be limited. In this case, the additional planning horizon needs to be decomposed as was proposed by the hierarchical approach in Saska et al. (2007) for UGVs in convex environments.

## 6.2 Scenario 2: cooperative searching for extremes in a field of a measured physical value

The second scenario deals with searching for locations in a 3D environment with an extreme in a field of a measured physical value. In particular, the investigated scenario is motivated by searching for locations with a minimum GSM signal in mountain areas, which are hard to reach, but which need to be sufficiently covered for safety reasons. Another example can be monitoring the intensity of WiFi signal in industrial complexes, shopping malls or large office buildings. The signal coverage and interfaces from multiple transmitters can hardly be modelled in such complex 3D environments, and physical measurements are therefore unavoidable. With their fast deployment and operability, MAVs are especially appealing to provide the desired data in these large and complex

**Fig. 22** Scheme of the system for feasible navigation of MAV swarms stabilized by visual relative localization based on the FSS algorithm



areas. Moreover, swarm intelligence can speed up the process of searching for extremes in the measured intensity and can enable more autonomy within the system.

In the proposed system, we rely on a Fish Search School (FSS) technique (Filho and Lima 2009), which allows us to define the swarm motion based on the actual state of particular particles. Each particle in the FSS defines its future movement based only on its current state and the states of neighbors obtained by onboard systems. This is preferable to methods such as Particle Swarm Optimization (PSO), where the new desired positions of MAVs are determined based on the best achieved position of a particle of the swarm so far (the global best) and the best achieved positions of each particle (the personal best). This requires to remember or denote these locations in the environment. The FSS method can be directly used for control of a swarm of MAVs with the proposed relative localization considering each MAV as an FSS particle. In such a tangible FSS, MAVs may use odometry from IMU for the short term localization in the environment during the displacement between two consequent positions generated by the FSS rules. The required information on the position of neighbors is achieved by visual relative localization. Information about the global position of MAVs in the environment is not necessary, as the robots are, in a matter of fact, steered by the distribution of the measured signal intensity.

The FSS control rule is created by three simple operators:

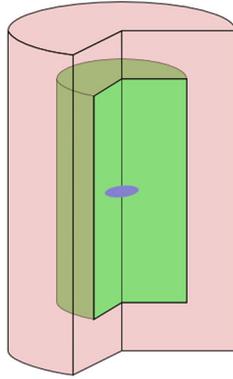
- (1) individual movement,
- (2) collective-instinctive movement,

- (3) collective-volitive movement that depends on a factor describing the recent success of the swarm.

The success of the swarm is determined by the progress of the cost function values, which are provided by the sensory measurement in this application example (see a mathematical expression of these rules in Filho and Lima (2009)). In the proposed tangible FSS algorithm, the optimization vector represents the position of one MAV simply as  $X = [x, y, z]$ , in contrast to the PSO environment coverage presented in Sect. 6.3, where the positions of all  $n_r$  MAVs are encoded into a unique optimization vector (the PSO particle). The number of FSS particles is equal to the number of physical robots  $n_r$  in the swarm (in the PSO environment coverage algorithm the number of PSO particles is equal to the number of virtual MAV swarms).

A scheme of the tangible FSS algorithm is shown in Fig. 22. The core of the motion planning and group stabilization algorithm is in the *Optimization Rules* block, where the FSS control rules are implemented according to Filho and Lima (2009). The input of this block is an estimate of the current relative positions of particles within the swarm and values of the cost function obtained for each MAV. In each optimization step, desired new positions of all MAVs are computed using the FSS rules based on this information. Then, the swarm is controlled into the new positions using the position control described in Sect. 4, while the localization and motion constraints are checked on the basis of the avail-

**Fig. 23** Safety zones around a quadcopter



able sensor data (the *Motion to the Position with Feasibility Check* blocks).

The most important part of the *Sensor Data* block for swarm stabilization and inter-vehicle collision avoidance is an estimation of relative positions of neighbors provided by the onboard localization system. Based on this information, the actual shape of the swarm is considered feasible if none of the MAVs are within the safety zones of another MAV. This means that the shape is feasible regarding the inter-vehicle collisions and the air-flow effect from propellers of neighboring MAVs. Two safety zones are considered (see Fig. 23) for collision avoidance. Robots can temporarily (conditionally) enter the red outer zone, but once they reach the blue inner zone, which is considered forbidden, they have to return back to release from both zones before the next FSS iteration. The concept of two zones prevents the system from oscillations and deadlocks in applications with dense swarms, where close proximities of MAV pairs and even multiple MAVs occur frequently.

A similar concept is employed for keeping the robots within the range of the relative localization. Again, two limits on the maximal distance between relatively stabilized MAVs are considered. The weaker restriction can be temporarily broken. Both limits have to be satisfied before the next FSS step. This approach decreases the likelihood that the swarm evolution gets stuck if several MAVs move close to the borders of their safety zones or close to the limits of their relative localization. Once all MAVs approach the locations obtained by the FSS rules or reach the last feasible constellation, sensor measurements are taken in the new positions of swarm particles. The measured values act as the cost values of the FSS optimization. The cost function evaluation is represented by the *cost function* block in the figure. The obtained cost function values and the information on the relative positions of neighbours are used as input of the *Optimization Rules* block in the new FFS iteration.

As a stopping criteria a predefined maximum number of iterations is used in experiments presented in this article.

According to Filho and Lima (2009), the progress of the total mass of FSS swarm and rate of the cost function values changes may be applied to detect termination of the searching process or deadlocks, but these studies go beyond the scope of this paper.

### 6.2.1 Experimental evaluation of the planning technique in flight conditions

The experiment in Figs. 24 and 25 demonstrates the use of the proposed tangible FSS method with onboard relative localization for searching in a 3D environment. In the experiments, MAVs cooperatively search for locations with the lowest intensity of a signal transmitted from four transmitters distributed in the environment at different altitudes. The intensity of the signal is simulated in the experiment based on known locations and the transmission power of the virtual transmitters. Instead of a real measurement of the signal strengths, the cost value for the  $i$ -th particle is then obtained as

$$F(X) = \sum_{j=1}^4 |x_i - s_j|^{-2}, \quad (13)$$

where  $s_j$  is the location of the  $j$ -th transmitter.

The progress of minimal cost value (13) “measured” by an MAV of the group in the particular iteration is shown in Fig. 26. The temporary increase of the cost values at the beginning of the experiment is caused by the initial stabilization of the group into a shape that satisfies the constraints given by size of the MAVs and their relative localization. For evaluation of the FSS algorithm with tangible particles (the real MAVs), results of a simulation with dimensionless particles are also presented in Fig. 26. The simulation was run using the same map and initial setup as in the real experiment. In the simulation, the initial stabilization of the group is not necessary, since the basic FSS method without motion constraints is used and the cost function values decrease from the beginning of the searching process.

The results presented in Fig. 25 show that the requirements on the maximal relative distance between particular pairs of MAVs (2.5 m) are kept during the experiment, and that the neighboring MAVs are always in the view angle of the onboard cameras. The relative distance  $|x_i - s_j|$  is obtained on the basis of data from the VICON motion capture system. In real-world deployment, knowledge of the global positions of MAVs, denoted as  $x_i$  here, would not be necessary, since the tangible FSS technique requires only knowledge on the positions relative to neighbors (and obstacles) and the actual measured intensity, which can all be obtained by onboard sensors.

**Fig. 24** Experiment with a swarm of 3 MAVs controlled by the feasible FSS rules

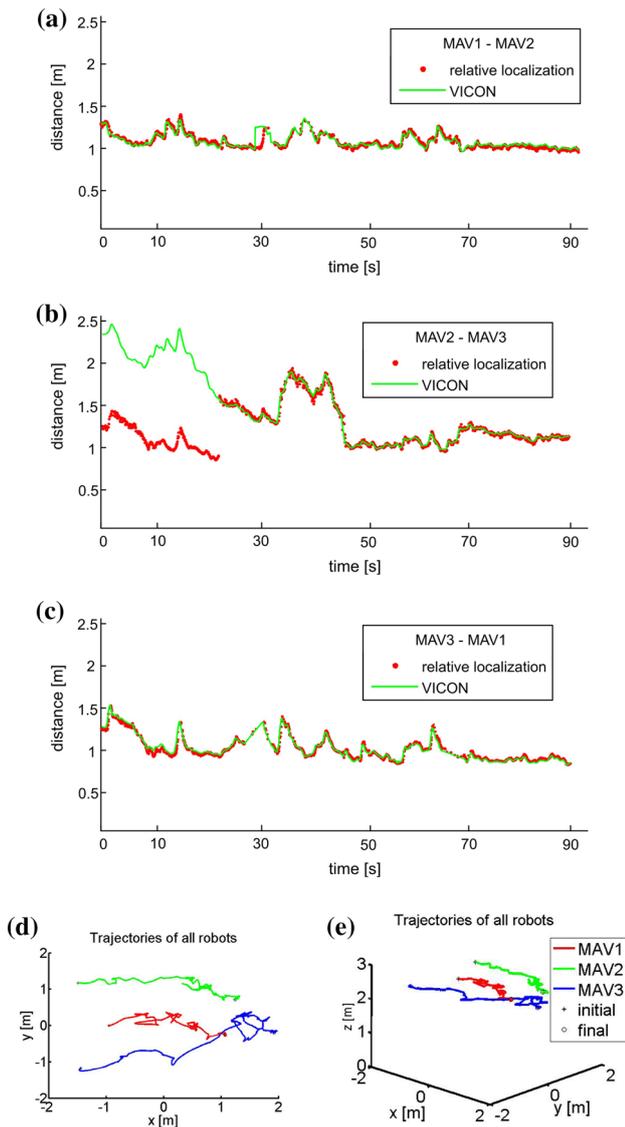


### 6.3 Scenario 3: environment coverage for cooperative surveillance

The third scenario demonstrates deployment of the proposed system in the task of cooperative surveillance (presence of MAVs at locations of interest). This section is a summary of the approach originally published by our team in the conference paper (Saska et al. 2014a). Here, the description of the method is put into the context of the presented control and localization system and it is used as an example of the system deployment in scenarios, where the trajectories of the robots have to be purposely computed prior the mission for their verification by the operator. In the scenario, a set of goals (areas) is assigned to a limited number of autonomous robots (MAVs) with the aim to find a static swarm configuration that can guard the areas. Let us call the set of static positions of all MAVs in the surveillance areas a *swarm distribution*, and let us call the complete task of the motion of MAVs from the initial depot into the static swarm distribution a *swarm deployment*.

Again, the MAV swarm has to respect the motion, localization, and sensing constraints of MAVs. These constraints

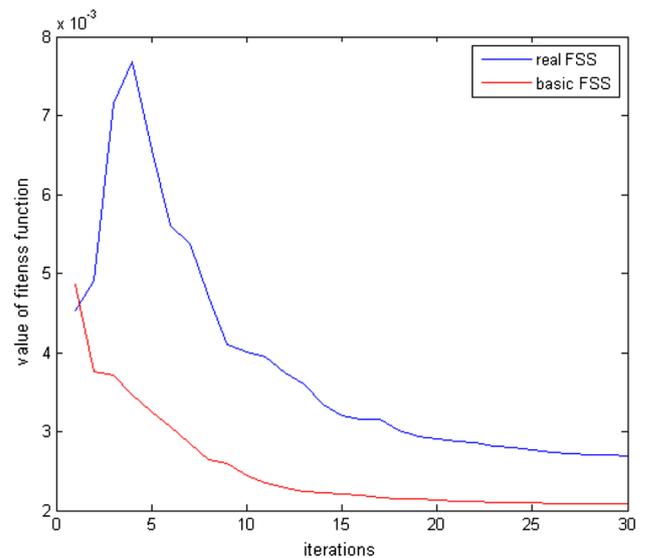
have to be applied in the final static swarm distribution and also during the swarm deployment. In the case that the swarm is not capable of covering the given set of locations of interest completely, for example because of an insufficient number of entities available or constraints on sensing, the coverage by the team members is maximised in the searching process. In this manner, we tackle the problem of static coverage of a set of areas by spreading a swarm of MAVs, while the swarm constraints are guaranteed for all obtained trajectories between the initial location of the MAVs into the achieved swarm distribution. So, we are looking for both: (1) the feasible static shape of the swarm (locations of particular swarm entities - the swarm distribution) and (2) a feasible plan of motion from the initial configuration to this target shape (trajectories for all MAVs—the swarm deployment). This leads us to a swarm-shape optimization with the need to keep the history (a feasible MAV movement) of swarm shape evolution from its initial state. This can be understood as a novel approach to multi-objective optimization, where a motion planning technique is integrated directly into the core of the optimization engine. The 3D pose of all MAVs in the swarm is then encoded into a unique optimization vector



**Fig. 25** Tangible Fish Search School (FSS) optimization. MAVs steered by FSS rules towards a location corresponding to a minimum of a signal transmitted from multiple transmitters. (slower movement of MAVs) **a** Comparison of the relative distances between MAVs captured by the onboard vision system and by data obtained by VICON **b** Comparison of the relative distances between MAVs captured by the onboard vision system and by data obtained by VICON **c** Comparison of the relative distances between MAVs captured by the onboard vision system and by data obtained by VICON **d** Positions of MAVs captured by VICON during the experiment **e** 3D view of the positions of MAVs captured by VICON during the experiment

as  $X = [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{n_r}, y_{n_r}, z_{n_r}]$ , where  $n_r$  is the number of robots in the swarm. The Particle Swarm Optimization (PSO) technique (Kennedy et al. 1995) is employed as the optimization method in this application.

A simple scheme of the proposed approach is shown in Fig. 27. In comparison with standard optimization techniques, where in each optimization step the actual solution (or several solutions) is directly evaluated by a cost function,

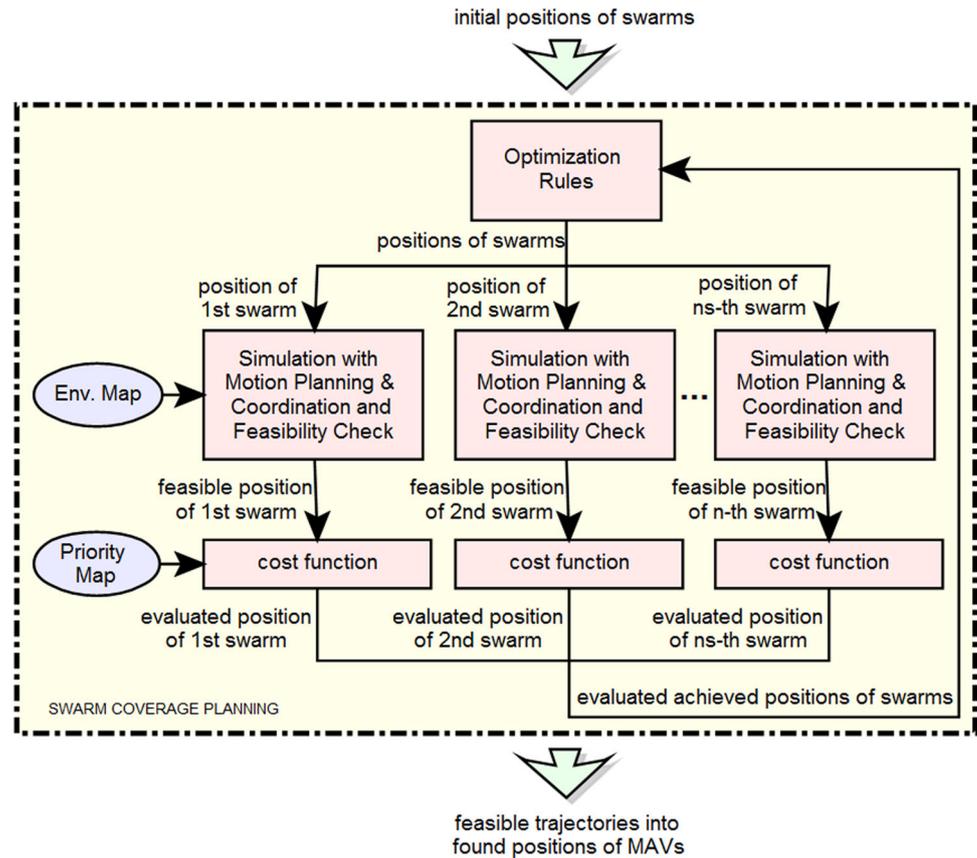


**Fig. 26** Progress of values of the lowest intensity measured by a swarm member at the particular iteration (the cost function values) in the experiment with real FSS particles and in a simulation with basic FSS method using dimensionless particles

here, the optimization vector is suited to respect the swarm constraints before the optimization continues. In each step of the optimization, the new shape of the swarm encoded into the optimization vector is used as an input to a motion planning approach, which generates collision-free trajectories connecting the desired positions with the actual state for each single MAV. The given plan is realized in a simulation using the trajectory tracking mechanism (Lee et al. 2010) with the MAV model introduced in Sect. 4. The simulation is run until the desired positions are reached or a violation of the swarm constraints is detected. If a mutual collision between MAVs is detected, the plan can often be corrected by a proper permutation of the goals assigned to particular vehicles. This does not influence the optimization process, since the MAVs are considered to be identical swarm particles. Any multi-robot coordination approach may be utilized in this phase of the planning mechanism if the permutation of goals is not sufficient. If a violation of the relative localization constraints (range, viewing angle, mutual MAV heading, etc.) is detected, the simulation is reversed into the last state considered as a feasible swarm distribution, and the optimization vector is replaced by this result. The achieved optimization vector is evaluated by the cost function, and the optimization continues in the next step from this state. An uncertainty in the optimization (e.g. the randomly weighted vectors addition in PSO) is crucial to increase the probability that the optimization will not end up in the same constraints violation, but it escapes from this potential deadlock.

In the experiments that are presented in this paper, the areas of interest are polygons and circles. The set of all these areas is represented by a square grid  $AoI$  that covers the

**Fig. 27** Scheme of the planning system for environment coverage by MAV swarms stabilized by visual relative localization



entire workspace with size of each cell 10 cm (the experimental workspace is shown in Fig. 30f). The cells of  $AoI$  that represent the areas of interest are initialized with the value 1, while the zero cells represent regions not assigned as areas of interest. The no-fly zones and the borders of the operational area are denoted by the mission operator as a set of convex polygons. These polygons are dilated and represented by the *Environment Map* of the same size ( $n, m$ ) as the size of the  $AoI$  matrix.

The cost function that evaluates particular solutions of the swarm spreading problem (position of all MAVs of the swarm) can be then expressed as

$$f(X) = \sum_{x=1, y=1}^{m, n} \max \left( 0, AoI_{x,y} - \sum_{i=1}^{n_r} R_{x,y,i} \frac{h_{opt}^2}{h_i^2} \right), \quad (14)$$

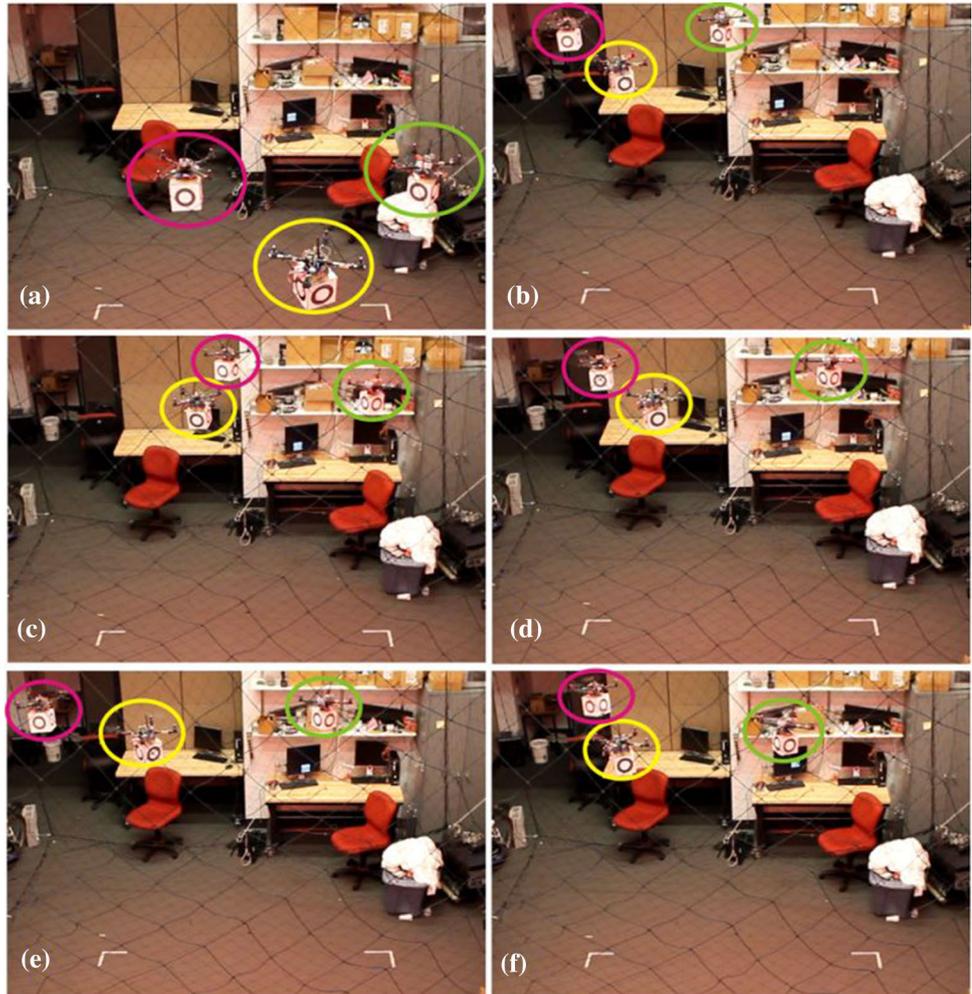
where  $h_i$  is the height of the  $i$ -th MAV above the ground (the altitude) and  $h_{opt}$  is the altitude determined as the “optimal” for the particular surveillance application. An MAV at lower altitude than  $h_{opt}$  does not gain more information per square unit. The value of the variable  $R_{x,y,i}$  is 1 if the cell of the workspace represented by the element  $AoI_{x,y}$  is completely observed by the surveillance sensor of the  $i$ -th MAV in its position in the swarm and 0 in the opposite case.

Finally, we should emphasize that the proposed method does not guarantee to find the optimal distribution of the swarm and the optimal trajectories from the initial positions into the found locations. What is guaranteed is the feasibility of the solution with respect to the motion and localization constraints. Regarding the presented relative visual localization, it is important that the plan of the swarm distribution in the environment satisfies constraints given by the range of the relative localization and viewing angle of the on-board cameras, and that it respects the mutual heading of the MAVs.

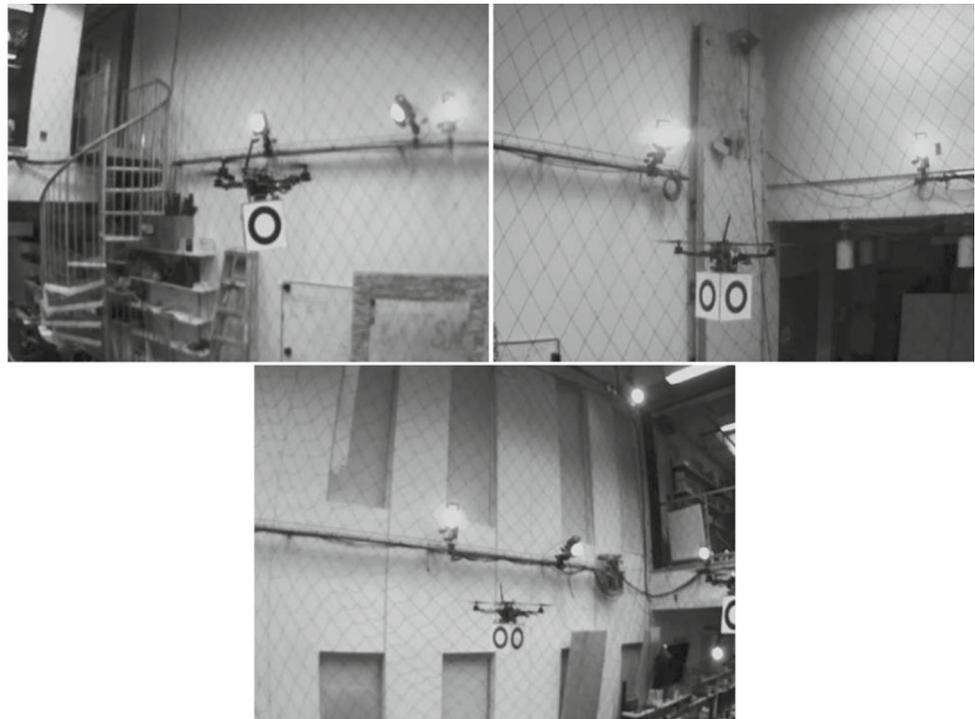
### 6.3.1 Experimental evaluation of the planning technique in flight conditions

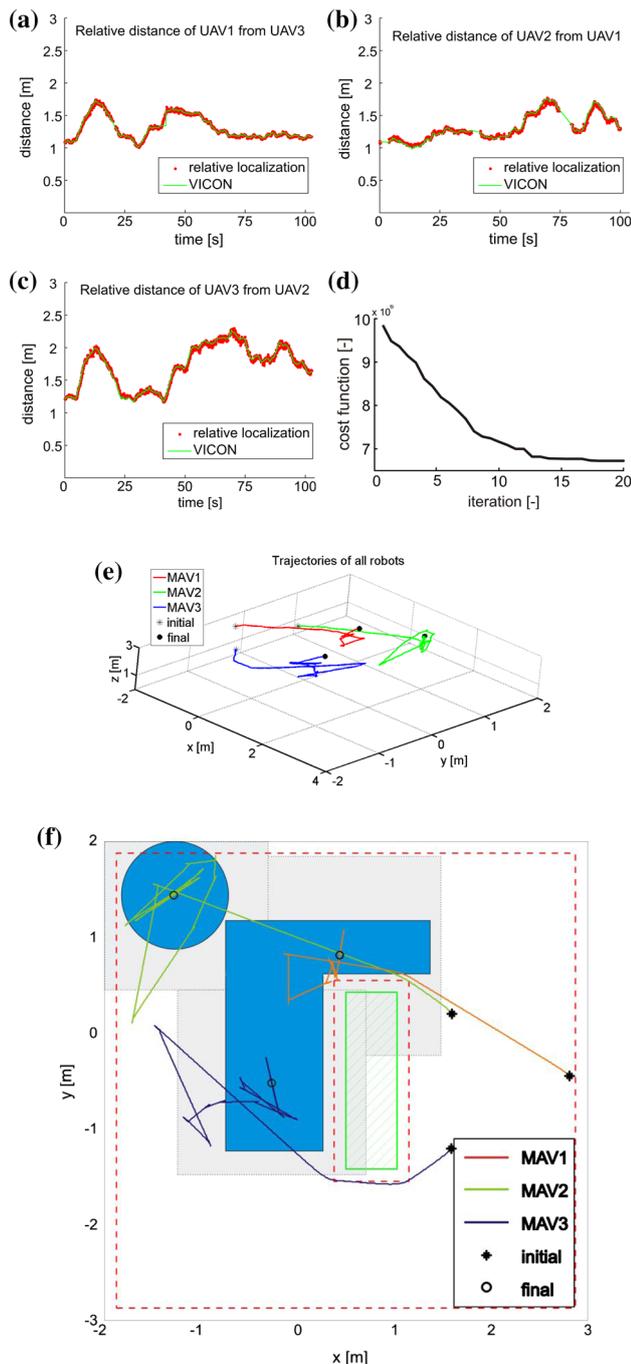
The aim of the experiment in Figs. 28, 29, 30 is to demonstrate deployment of the proposed system in a surveillance task, where locations of interest with different priorities are covered by a self-stabilized swarm of MAVs. The feasibility of the plan for swarm distribution in the environment with known sets of areas of interest, no-fly zones and initial positions of the MAVs is verified in the experiment. The plan has to satisfy the constraints given by the range of the relative localization, the viewing angle of the on-board cameras, the mutual heading of the MAVs, and the movement constraints during deployment of the system. Fig. 30 shows that a guess

**Fig. 28** Experiment with swarm of 3 MAVs following trajectories obtained off-line by the proposed planning algorithm. MAVs are denoted by circles of different colours (Color figure online)



**Fig. 29** Pictures taken by the onboard localization systems of all MAVs in the same moment (experiment in Fig. 28)





**Fig. 30** Swarm deployment in the environment to cover selected areas of interest. (experiment in Fig. 28) **a** Comparison of relative distances between MAVs captured by the onboard vision system and by the external motion capture system (VICON). **b** Comparison of the relative distances between MAVs captured by the onboard vision system and by VICON. **c** Comparison of the relative distances between MAVs captured by the onboard vision system and by VICON. **d** Progress of the cost function values of the best PSO particle during off-line optimization of the swarm deployment found for the experiment in Fig. 28. **e** 3D view of positions of MAVs captured by VICON during the experiment. **f** Positions of MAVs captured by VICON during the experiment, with denoted areas of interest (blue regions) and the no-fly zone (green rectangle) extended with a safety zone due to localization and control uncertainty) (Color figure online)

of the relative position of neighboring vehicles is continuously provided during the flight, and that the limit on the relative distances within the swarm entities (2.5 m) is kept.

Finally, we should mention that a global localization system (such as GPS) is necessary to reach the surveillance locations of the group in applications of the approach designed for swarm deployment. In most of the scenarios with compact MAV swarms, such positioning system has lower precision in comparison with the relative distances between MAVs. Therefore, the more precise onboard relative localization needs to be employed to protect the swarm members from mutual collisions. Moreover in our approach, such a global localization technique may be used to localize only few robots of the group. In the experiment presented in this section, the global position is estimated from the visual odometry of one of the MAVs using the PX4Flow sensor, while the entire group is stabilized using the onboard relative localization system.

#### 6.4 Comparison of performance of the system in scenarios 1–3

The purpose of this section was to demonstrate possibility of deployment of multi-MAV teams in different robotic scenarios and to show advantages and disadvantages of the onboard relative localization system in different techniques of control of MAV-groups.

In the scenario 2, outputs of the visual relative localization system may be used directly in the FSS rules, and therefore, the performance of the system directly influences the planning process. The advantage of this approach is that it is very robust to inaccuracy of measurement of the relative distances, but the FSS method is sensitive to drop out of the system. Longer malfunctions of the onboard localization system cause interruption of the searching process and may even lead to inter-vehicle collisions.

On the contrary, the control system is robust to a drop out of the localization method and sensitive to inaccuracy in measurements of the relative positions in scenarios 1 and 3. In these scenarios, the onboard visual relative localization approach is used to unify local reference frames of the MAVs. In case of a temporary drop out of the relative localization, the MAVs can safely continue in their mission based on the visual odometry. The allowed duration of the drop out depends on the cumulative error of the odometry, the current distance between MAVs, the safety distance between MAVs, and the range of the onboard localization system. If the precision of the relative localization is low, the performance of formation flying and swarm deployment may be even worse than if the system relies only on the odometry of particular MAVs.

Finally, let us describe computational complexity and communication load required by these approaches.

The most computationally demanding is the scenario 1, where the initial plan for the leader and also the control inputs for the followers need to be computed with sufficiently powerful PC to be able to get result between two planning steps. In the presented experiments, the initial plan was obtained in approximately 900ms and each MPC step required 20–40ms. The plan was computed on an external PC (Intel Core i7, 8GB RAM) and then wirelessly sent into MAVs. This setup was sufficient for testing of the relative localization system, but MAVs already may be equipped by sufficiently small and powerful HW solutions to enable onboard computing. The data flow is low in this application, since only few control commands need to be sent into the MAVs in each control step. In case of onboard computing, which is expected in real applications, only the plan of the leader needs to be distributed within the team.

In the second scenario, the swarming algorithm is not computationally intensive and can be run onboard on the  $\mu$ -controller. Also the data flow is very low. Only the cost function value needs to be distributed within the team after each measurement, which is done with low update rate.

In the third scenario, the plan is purposely computed prior the mission to enable its verification by a human operator of the surveillance mission. Therefore, this plan may be obtained using standard PC and then sent into MAVs. The trajectory following process is run using the onboard ATmega  $\mu$ -controller as described in Sect. 4. During the flight, no communication is required except the initial synchronization command.

## 7 Conclusion

A complex system for stabilization and control of MAV-groups based on onboard visual relative localization has been presented in this paper. The aim of the system is to provide a tool for autonomous deployment of teams of unmanned quadcopters in real world scenarios without the need for external localization. An onboard camera module with a fast image processing algorithm suited for the requirements of the group stabilization was described together with a simple controller using this module in the control feedback. As the core of the presented system, three various planning approaches have been proposed to solve specific multi-MAV scenarios. The common factor of these methods is the endeavour to solve the group stabilization, motion planning and coordination tasks with the specific requirements given by the employed vision-based relative localization. The performance and feasibility of the motion planning methods presented here have been verified and evaluated by experiments with a fleet of MAVs. In the experiments, the performance of the onboard relative localization system in particular applications has been numerically evaluated with

respect to an external motion capture system used as a ground truth.

In all motion planning approaches presented in this paper, the constraints on relative positions of the MAVs in the group are considered to satisfy the direct visibility among them and therefore to continuously keep the relative localization linkages during their motion. Nevertheless, various experiments of the system have shown that the relative localization of neighboring vehicles can be temporarily interrupted without any negative influence on the overall system stability. In our future work, we will integrate the possibility of temporal disconnection of the localization linkages, due to obstacles appearing in between of MAVs or due to a temporary enlargement of the group size, into the motion planning method. This significantly increases applicability of the system in GPS-denied environment, where the GPS signal is blocked by obstacles that may be present in such a high density that it is impossible to avoid them by the entire group, and where the temporary occurrence of the obstacles in between of the MAVs has to be allowed.

**Acknowledgments** This work has been supported by the Ministry of Education of the Czech Republic under project no. LH11053 and the experimental works required for paper revisions by Project No. HS 13167/830/8301616C000 founded by Khalifa University for the MBZIRC competition, both projects supporting the joint research of the Czech Technical University in Prague and the University of Pennsylvania. In addition, Martin Saska has been supported by the Grant Agency of the Czech Republic under postdoc Grant No. P103-12/P756. The work of Jan Faigl has been also partially supported by the Czech Science Foundations (GACR) under the research Project No. 13-18316P. Tomas Krajnik has been supported by the EU project STRANDS (ICT-600623). Tomas Baca has been supported by CTU grant no. SGS15/157/OHK3/2T/13. Final experiments done by Martin Saska for revisions of the paper have been supported by the Czech Science Foundations (GACR) under the research Project No. 16-24206S.

## References

- Balch, T., & Hybinette, M. (2000). Social potentials for scalable multi-robot formations. In *Proceedings of IEEE conference on robotics and automation*.
- Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111–122.
- Barfoot, T. D., & Clark, C. M. (2004). Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46, 65–78.
- Barnes, L., Garcia, R., Fields, M., & Valavanis, K. (2008). Swarm formation control utilizing ground and aerial unmanned systems. In *IEEE/RSJ international conference on intelligent robots and systems*.
- Bennet, D. J., & McInnes, C. R. (2009). Verifiable control of a swarm of unmanned aerial vehicles. *Journal of Aerospace Engineering*, 223(7), 939–953.
- Berman, S., Halasz, A., Hsieh, M., & Kumar, V. (2009). Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4), 927–937.
- Bošnak, M., Matko, D., & Blažič, S. (2012). Quadcopter control using an on-board video system with off-board processing. *Robotics and Autonomous Systems*, 60(4), 657–667.

- Buerkle, A., & Leuchter, S. (2009). Development of micro UAV swarms. *Autonome mobile systeme 2009* (pp. 217–224)., Informatik aktuell series Berlin: Springer.
- Cai, W., Yu, Q., & Wang, H. (2004). A fast contour-based approach to circle and ellipse detection. In *5th world congress on intelligent control and automation (WCICA)*.
- Cai, N., Xi, J.-X., & Zhong, Y.-S. (2011). Swarm stability of high-order linear time-invariant swarm systems. *Control Theory Applications IET*, 5(2), 402–408.
- Carreras, M., Rida, P., García, R., & Nicosevici, T. (2003). Vision-based localization of an underwater robot in a structured environment. In *ICRA*.
- Chao, Z., Zhou, S.-L., Ming, L., & Zhang, W.-G. (2012). UAV formation flight based on nonlinear model predictive control. *Mathematical Problems in Engineering*, 2012(1), 1–16.
- Cheah, C. C., Hou, S. P., & Slotine, J. J. E. (2009). Region-based shape control for a swarm of robots. *Automatica*, 45(10), 2406–2411.
- Christensen, A., O'Grady, R., & Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4), 754–766.
- Doitsidis, L., Weiss, S., Renzaglia, A., Kosmatopoulos, E., Siegart, R., Scaramuzza, D., et al. (2012). Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision. *Autonomous Robots*, 33(1–2), 173–188.
- Faigl, J., Krajník, T., Chudoba, J., Preucil, L., Saska, M. (2013). Low-cost embedded system for relative localization in robotic swarms. In *Proceedings of IEEE international conference on robotics and automation*.
- Faigl, J., Krajník, T., Vonásek, V., & Přeucil, L. (2012). On Localization Uncertainty in an Autonomous Inspection. In *IEEE international conference on robotics and automation (ICRA)*.
- Fazli, P., Davoodi, A., & Mackworth, A. (2013). Multi-robot repeated area coverage. *Autonomous Robots*, 34(4), 251–276.
- Filho, C., Lima Neto, F., Lins, A., Nascimento, A., & Lima, M. (2009). fish school search. *Nature-inspired algorithms for optimisation, studies in computational intelligence* (pp. 261–277). Berlin: Springer.
- Garca Carrillo, L., Sanchez, A., Dzul, A., & Lozano, R. (2011). Stabilization and trajectory tracking of a quad-rotor using vision. *Journal of Intelligent & Robotic Systems*, 61, 103–118.
- Hamann, H., & Worn, H. (2008). A framework of spacetime continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2, 209–239.
- Holland, O., Woods, J., Nardi, R., & Clark, A. (2005). Beyond swarm intelligence: The UltraSwarm. In *IEEE swarm intelligence symposium*.
- Jia, L.-Q., Liu, H.-M., Wang, Z.-H., & Chen, H. (2011). An effective non-HT circle detection for centers and radii. In *ICMLC*.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4).
- Kloetzer, M., & Belta, C. (2007). Temporal logic planning and control of robotic swarms by hierarchical abstractions. *IEEE Transactions on Robotics*, 23(2), 320–330.
- Krajník, T., Nitsche, M., Faigl, J., Vanek, P., Saska, M., Přeucil, L., Duckett, T., & Mejail, M. (2014). A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, Online first, 2014. <http://dx.doi.org/10.1007/s10846-014-0041-x>.
- Kumar, M., Garg, D., & Kumar, V. (2010). Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3), 743–748.
- Lange, S., Sunderhauf, N., & Protzel, P. (2009). A vision based onboard approach for landing and position control of an autonomous multi-rotor uav in GPS-denied environments. In *International conference on advanced robotics (ICAR)*.
- Lee, T., Leoky, M., & McClamroch, N. (2010). Geometric tracking control of a quadrotor UAV on se(3). In *49th IEEE conference on decision and control (CDC)*.
- Leonard, N., & Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control*.
- Liu, C., Chen, W.-H., & Andrews, J. (2011). Piecewise constant model predictive control for autonomous helicopters. *Robotics and Autonomous Systems*, 59(78), 571–579.
- Liu, W., Winfield, A., Sa, J., Chen, J., & Dou, L. (2007). Strategies for energy optimisation in a swarm of foraging robots. *Swarm Robotics*, 4433, 14–26.
- Marjovi, A., & Marques, L. (2013). Optimal spatial formation of swarm robotic gas sensors in odor plume finding. *Autonomous Robots*, 35(2–3), 93–109.
- Masselli, A., & Zell, A. (2012). A novel marker based tracking method for position and attitude control of MAVs. In *Proceedings of international micro air vehicle conference and flight competition*.
- Michael, N., Mellinger, D., Lindsey, Q., & Kumar, V. (2010). The grasp multiple micro-UAV testbed. *IEEE Robotics Automation Magazine*, 17(3), 56–65.
- Multimedia. (2015). Various experiments with multi-MAV system verifying the proposed approach. <http://mrs.felk.cvut.cz/data/mavgroups/> Retrieved from 8 August 2015.
- No, T. S., Kim, Y., Tahk, M.-J., & Jeon, G.-E. (2011). Cascade-type guidance law design for multiple-UAV formation keeping. *Aerospace Science and Technology*, 15(6), 431–439.
- Rad, A. A., Faez, K., & Qaragozlou, N. (2003). Fast circle detection using gradient pair vectors. In *DICTA*.
- Saska, M., Chudoba, J., Preucil, L., Thomas, J., Loianno, G., Tresnak, A., Vonasek, V., & Kumar, V. (2014a). Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. In *Proceedings of 2014 international conference on unmanned aircraft systems (ICUAS)*.
- Saska, M., Hess, M., & Schilling, K. (2007). Hierarchical spline path planning method for complex environments. In *Proceedings of the 4th international conference on informatics in control, automation and robotics*. Angers, France.
- Saska, M., Kasl, Z., Preucil, L. (2014b). Motion planning and control of formations of micro aerial vehicles. In *Proceedings of the 19th world congress of the international federation of automatic control*.
- Saska, M., Mejia, J. S., Stipanovic, D. M., Schilling, K. (2009). Control and navigation of formations of car-like robots on a receding horizon. In *Proceedings of 3rd IEEE multi-conference on systems and control*.
- Saska, M., Vonasek, V., & Preucil, L. (2010). Control of ad-hoc formations for autonomous airport snow shoveling. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 4995–5000). Taipei: IEEE Industrial Electronics Society.
- Saska, M., Krajník, T., Vonasek, V., Kasl, Z., Spurny, V., & Preucil, L. (2014c). Fault-tolerant formation driving mechanism designed for heterogeneous MAVs-UGVs groups. *Journal of Intelligent and Robotic Systems*, 73(1–4), 603–622.
- Saska, M., Mejia, J., Stipanovic, D., Vonasek, V., Schilling, K., & Preucil, L. (2013a). Control and navigation in manoeuvres of formations of unmanned mobile vehicles. *European Journal of Control*, 19(2), 157–171.
- Saska, M., Spurny, V., & Vonasek, V. (2016). Predictive control and stabilization of nonholonomic formations with integrated spline-path planning. *Robotics and Autonomous Systems*, 75, 379–397.
- Saska, M., Vonasek, V., Krajník, T., & Preucil, L. (2014d). Coordination and navigation of heterogeneous MAV&UGV formations localized by a hawk-eye-like approach under a model predictive control scheme. *International Journal of Robotics Research*, 33(10), 1393–1412.

- Saska, M., Vonasek, V., & Preucil, L. (2013b). Trajectory planning and control for airport snow sweeping by autonomous formations of ploughs. *Journal of Intelligent and Robotic Systems*, 72(2), 239–261.
- Schmickl, T., & Crailsheim, K. (2008). Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25, 171–188.
- Sharma, R. K., & Ghose, D. (2009). Collision avoidance between UAV clusters using swarm intelligence techniques. *International Journal of Systems Science*, 40, 521–538.
- Teacy, W., Nie, J., McClean, S., & Parr, G. (2010). Maintaining connectivity in UAV swarm sensing. In *IEEE GLOBECOM Workshops*.
- Trianni, V. (2008). *Evolutionary swarm robotics*. New York: Springer.
- Turpin, M., Michael, N., & Kumar, V. (2012). Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1–2), 143–156.
- Winfield, A., Liu, W., Nembrini, J., & Martinoli, A. (2008). Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2, 241–266.
- Yang, S., Scherer, S. A., & Zell, A. (2012). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. In *International conference on unmanned aircraft systems (ICUAS'12)*.
- Yang, S., Scherer, S., & Zell, A. (2012). An Onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1–4), 499–515.
- Yu, H., & Beard, R. (2013). A vision-based collision avoidance technique for micro air vehicles using local-level frame mapping and path planning. *Autonomous Robots*, 34(1–2), 93–109.



**Martin Saska** received his M.Sc. degree at Czech Technical University in Prague, 2005, and his Ph.D. degree at University of Wuerzburg, Germany, within the Ph.D. program of Elite Network of Bavaria, 2009. Since 2009, he is a research fellow at Czech Technical University in Prague, where he founded and heads the Multi-robot Systems lab (<http://mrs.felk.cvut.cz/>) and co-founded Center for Robotics and Autonomous Systems with more than 40 researchers cooperating in

robotics (<https://robotics.fel.cvut.cz/cras/>). He was a visiting scholar at University of Illinois at Urbana-Champaign, USA in 2008 and at University of Pennsylvania, USA in 2012, 2014 and 2016, where he worked with Vijay Kumar's group within GRASP lab. He is an author or co-author of more than 40 publications in peer-reviewed conferences and more than 10 publications in impacted journals, including IJRR, AURO, RAS, ASC, EJC. He received best paper award at ICUAS 2013 and he was best paper award honored finalist at ICRA 2008. He was program committee member of RA 2007, FLAIRS 27, IDC 2013, FLAIRS 28, ICUAS 2015, ICRA 2016, ICUAS 2016 and serves as reviewer in IEEE trans. on automatic control, IEEE trans. on automation science and engineering, Journal of intell. & robotic syst., Autonomous robots, Swarm intelligence, European journal of control, Journal of the franklin institute, Engineering and applied mathematics. His interests include cooperation and control of multi-robot systems, formation flying, and bio-inspired swarms.



**Tomas Baca** is a Ph.D. student at Czech Technical University in Prague, with topic: Model predictive control of groups of unmanned aerial vehicles. His interests include control and stabilization of multi-robot systems, and application of aerial vehicles in photography.



**Justin Thomas** received his B.S. in Mechanical Engineering from LeTourneau University in May 2011 and graduated Summa Cum Laude. While at LeTourneau University, Justin researched propulsion systems for fixed-wing aircraft and control systems for biomechanical testing fixtures. In the Fall of 2012, Justin joined the University of Pennsylvania as a Ph.D. student in the department of Mechanical Engineering and Applied Mechanics and as a member of the GRASP Lab under the guidance of Professor Vijay Kumar. At the University of Pennsylvania, Justin is researching aerial manipulation and vision-based control using Micro Aerial Vehicles.



**Jan Chudoba** received his M.Sc. degree at Czech Technical University in Prague, 2004. Currently, he is a research fellow at Department of Cybernetics. He has >20 publications in robotics. His interests include navigation of mobile robots, mapping and sensor data fusion.



**Libor Preucil** received Ph.D. in Computer Vision and Medical Image Processing, 1993. Since 1996 he has acted as the Managing Researcher at the Gerstner Laboratory (a part of the Department of Cybernetics), CTU Prague, where he is a head of Intelligent and mobile robotics group.



**Tomas Krajník** is a research fellow at the Lincoln Center of Autonomous Systems, UK and Intelligent Mobile Robotics Group of the Czech Technical University in Prague. He has received his Ph.D. degree in Artificial Intelligence and Biocybernetics from the Czech Technical University, Prague, Czech Republic, in 2012. His research interests include autonomous navigation, environment modelling, computer vision and aerial robots.



**Jan Faigl** received the Ing. degree in technical cybernetics and the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University in Prague (CTU), in 2003 and 2010, respectively. Since 2013, he is an assistant professor at the Agent Technology Center, Department of Computer Science, where Jan leads computational robotics laboratory focused on intelligent data processing, mission and motion planning for complex autonomous vehicles in

surveillance, exploration, and environment monitoring missions. In 2013/2014 he acted as Fulbright Visiting Scholar at the University of Southern California for six months. He is an author or co-author of more than 40 publications in peer-reviewed journals and conferences.



**Giuseppe Loianno** received B.Sc and M.Sc in Automation Engineering both with honours at University of Naples Federico II in December 2007 and February 2010, respectively. From August 2008 till March 2009 he has been an exchange student at KTH (Royal Institute of Technology) in Stockholm, then from September 2009 till March 2010 he developed his master thesis at ETH

Zurich at the ASL laboratory focusing on Micro Aerial Vehicles under the supervision of Prof. Dr. Davide Scaramuzza. He received his Ph.D in computer and automatic engineering focusing in Robotics in May 2014 in the PRISMA Lab group, led by Prof. Dr. Bruno Siciliano. From April 2013 he worked for 14 months with the Grasp Lab at University of Pennsylvania, supervised by Prof. Dr. Vijay Kumar, where currently he is a postdoctoral researcher. His research interests include visual odometry, sensor fusion and visual servoing for micro aerial vehicles.



**Vijay Kumar** is the UPS Foundation Professor in the School of Engineering and Applied Science at the University of Pennsylvania. He received his Ph.D. from The Ohio State University in 1987. He served as the Deputy Dean for Research in the School of Engineering and Applied Science from 2000-2004. He directed the GRASP Laboratory, a multidisciplinary robotics and perception laboratory, from 1998-2004. He was the Chairman of the Department of Mechanical Engineering

and Applied Mechanics from 2005-2008 and the Deputy Dean for Education in the School of Engineering and Applied Science from 2008-2012. He was the assistant director for robotics and cyber physical systems at the White House Office of Science and Technology Policy from 2012-2014. Dr. Kumar served on the editorial boards of the IEEE Transactions on Robotics and Automation, IEEE Transactions on Automation Science and Engineering, ASME Journal of Mechanical Design, the ASME Journal of Mechanisms and Robotics and the Springer Tract in Advanced Robotics (STAR). He is the recipient of the 1991 National Science Foundation Presidential Young Investigator award, the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, the 2012 ASME Mechanisms and Robotics Award, the 2012 IEEE Robotics and Automation Society Distinguished Service Award and a 2012 World Technology Network Award. He has won best paper awards at DARS 2002, ICRA 2004, ICRA 2011, RSS 2011, and RSS 2013, and his doctoral students won Best Student Paper Awards at ICRA 2008, RSS 2009, and DARS 2010.

# D

---

KEY ARTICLE [4] - ICARCV 2016

---

©[2018] IEEE. This paper is available in proceedings of the 14th International Conference on Control, Automation, Robotics and Vision: **M. Saska**, T. Baca, and D. Hert: Formations of unmanned micro aerial vehicles led by migrating virtual leader, 2016. *An extended journal version of the paper is under review of Autonomous Robots.*



# Formations of Unmanned Micro Aerial Vehicles Led by Migrating Virtual Leader

Martin Saska

Faculty of Electrical Engineering  
Czech Technical University in Prague  
Email: martin.saska@fel.cvut.cz

Tomas Baca

Faculty of Electrical Engineering  
Czech Technical University in Prague  
Email: bacatoma@fel.cvut.cz

Daniel Hert

Faculty of Electrical Engineering  
Czech Technical University in Prague  
Email: hertdani@fel.cvut.cz

**Abstract**—A novel approach for control and motion planning of formations of multiple unmanned micro aerial vehicles (MAVs), also referred to as unmanned aerial vehicles (UAVs) - multirotor helicopters, in cluttered GPS-denied environments is presented in this paper. The proposed method enables autonomously to design complex maneuvers of a compact MAV team in a virtual-leader-follower scheme. The feasibility of obtained results of the motion planning approach and the required stability of the formation is achieved by migrating the virtual leader along a hull surrounding the formation. This enables us to suddenly change formation motion in all directions, independently of actual orientation of the formation.

## I. INTRODUCTION

Control and stabilization of MAVs are currently intensively studied research fields due to their high maneuverability and possibility to reach locations hardly accessible by unmanned ground vehicles (UGVs) and by fixed-wing unmanned aerial vehicles (UAVs) that are usually larger and too fast for flying in a straitened environment with obstacles. MAVs are especially appealing for applications in such cluttered workspace since they can fly close to obstacles relatively safely and they enable complex maneuvers in a low speed. The same abilities are required for control of compact closely cooperating multi-MAV teams, where also inter-vehicle coordination of MAVs flying in small relative distances needs to be tackled (see Fig. 1 for examples of our target applications). In addition, the requirements on operation in compact MAV formations in the cluttered workspace close to the obstacles exclude utilization of global navigation satellite systems, such as GPS, with their insufficiently low precision in comparison with required mutual distances between neighbouring MAVs and its low reliability if the signal from satellites is partly or completely disabled by the obstacles.

This paper proposes to solve this problem by a novel motion planning and stabilization approach for control of teams of MAVs flying in compact formations in a cluttered environment without using GPS. The proposed method is focused on key properties required for deployment of the groups of aerial robots in narrow spaces, i.e. efficient motion planning that enables to design autonomously complex maneuvers of the formation and coordination of possibly large MAV groups flying in small relative distances. The required high maneuverability is achieved by inclusion of model predictive control (MPC) and trajectory planning, which is feasible for the MAV formations,



Fig. 1. Examples of practical deployment of parts of the proposed formation flying method. a) Cooperative localization of active RFID chips by a formation of relatively stabilized MAVs of a given shape. b) Documentation of hardly accessible places in historical sites by a formation of MAVs. One MAV carries a camera and its neighbors light sources with possibility to set relative angle between the camera axis and the lights, as it is required by the classical film lighting techniques, such as the three-point lighting approach [3].

into a single optimization process and by employing a concept of a virtual leader migrating along a convex hull surrounding the formation. Such extension of the classical leader-follower (L-F) approach enables to arbitrary change motion direction of the formation enforced by the surrounding environment and by the obstacles, while the requirement given by the L-F technique, in which the leader is always in front of the formation, is satisfied. This enables us to rely on the well-conducted theory of the L-F control technique and to guarantee the stability of the group.

To be able to localize the MAVs in the formation, we rely on an extension of the onboard visual relative localization system using a single camera that we have described in [1], [2]. The extended system, which may be used for real deployment of the theory presented in this paper, enables to estimate relative positions of neighboring robots in the formation using a 360° panoramic view provided by a ring of four onboard monocular cameras and localization patterns carried by each formation member (see Fig. 2). The relative pose is estimated from the detected size of the pattern and its position in the image.

In the presented approach, we will rely on a virtual-leader-follower approach, where all MAVs are considered as followers following a virtual robot in front of the formation. The virtual-leader approach is often used for multi-robot control due to its high robustness (see e.g. [4]), since it is not threatened by a failure of the real leader. In our paper, the motivation for using the virtual leader is different. We take advantage of the possibility to change the virtual leader position relatively to the formation, which enables suddenly change its motion direction while keeping the requirements of the L-F scheme specified



Fig. 2. Examples of the outdoor experiments of stabilization of MAV formations using the onboard vision-based relative localization system [1], [2] and picture of the 360° relative localization module.

in [5].

In recent works, the sudden change of the position of the formation leader is usually realized by switching the leadership between different team members to increase the system robustness [6], to enable splitting the team [7], and to reconfigure the formation shape [8]. While all of these works are focused on a study of shape stability of the formation simply following a given path, we will focus on the integration of switching the positions of the virtual leader into the trajectory planning process and on aspects and limitations of deployment of MAVs in close relative distances.

For other examples of recent methods designed for control of MAV formations see [9], [10], [11]. In all of these works, the desired trajectory that is followed by the formation is given or obtained by an external path planning method modified for the formation requirements. An integration of trajectory planning into the formation stabilization process can be found only in [12], where a combination of the L-F approach with a potential field technique is presented. Nevertheless, this algorithm is designed for ground holonomic robots, and it suffers from the usual problems of potential field techniques, such as oscillations near the local extremes, deadlocks in U-shape obstacles and poor performance if avoiding clusters of obstacles. In the presented method, we enable to include a global information about the environment, which is not possible with the method in [12], and to plan complex trajectories in a cluttered environment with obstacles.

As mentioned, we propose to include the ability of global trajectory planning into the MPC scheme within a single optimization process. This approach enables us to integrate the relative-localization constraints, which are crucial for robots deployment in a GPS-denied environment, directly into the formation stabilization process. MPC solutions [13] and [14] have demonstrated that the MPC method can be used for stabilization of a single MAV using an onboard embedded processor. Although these pioneering methods can realize only simple control tasks (such as hovering in ideal windless conditions) due to a short prediction horizon, our solution introduced in [15] enables to control a single MAV using embedded

processor and onboard sensors with prediction control horizon longer than 2s, as shown in numerous indoor and outdoor experiments <https://youtu.be/IPy7w-GUbw4>. Such prediction interval is sufficient for following arbitrary dynamically changing 3D trajectories with the precision and stability required in multi-MAV scenarios (see Fig. 2 for examples of stabilization of MAV formations using the visual relative localization [1], [2]).

This work is built on our achievements in control and stabilization of formations of ground vehicles [16], [17] and heterogeneous teams of ground and aerial robots [18], [19], where the formations were able to follow only simple trajectories without the possibility of changing the motion direction. Moreover, the planning was limited to a plain space due to applied ground robots, which were used for stabilization of the formation relatively to the ground. The possibility of motion direction alternation was introduced in our work [20], [21], which was also designed for the ground (car-like) robots and UGV-UAV teams. Due to the limited motion capabilities of the car-like robots, the formations in [20], [21] may change the motion direction only from the forward motion to the backward motion (i.e. only the 180-degree change of the motion direction). The proposed approach enables to plan maneuvers with the arbitrary change of the motion direction (the change of the motion direction can be done immediately into any angle) of the formation keeping its desired shape in a cluttered workspace. In the proposed MPC approach, the trajectory planning into the desired target, the immediate formation control, and the planning of positions of the virtual follower relatively to the formation are integrated into a single optimization process. Due to this cohesion, the formation can smoothly respond to changes in its vicinity and to consider also the global information about the environment in the control loop. The local control inputs are then optimized not only regarding the short-term obstacle avoidance but also regarding the future movement of the formation.

## II. PROBLEM DEFINITION

In this paper, a problem of control of an MAV formation of a given shape from its initial position into a spherical target region is tackled. We assume a fixed number  $N_{MAV}$  of the MAVs in the formation. It is assumed that the required formation shape, which is specified by curvilinear coordinates  $p_i, q_i, h_i$  for each MAV  $i$  relatively to the virtual leader, satisfies constraints on the relative localization of all formation members. It means that the maximal distance between any two robots in the formation is shorter than the sensing range of the relative localization system as specified in [2]. Due to the 6-pages limit of this paper, we have to omit description of this curvilinear coordinate system, and we refer to Fig. 4 and to [18] for details. We assume that a finite number of compact obstacles, which are represented in 3D as complex polyhedrons, is located in the workspace. Considering only the polyhedrons, the minimal distance between an obstacle and a convex shape representing the formation can be efficiently obtained by the GilbertJohnsonKeerthi distance algorithm (GJK),

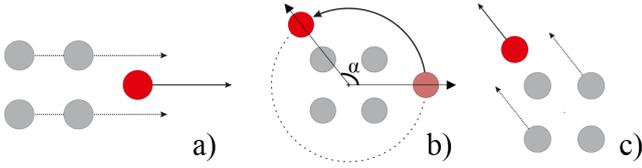


Fig. 3. Migration of the virtual leader to enable sudden change of the flight direction, if the maneuvering in a clustered environment is necessary. Virtual leader - red, followers - grey. a) Flight before the migration. b) Migration of the virtual leader. c) Flight in the new direction.

which is important for fast obstacle avoidance due to its low computational complexity [22]. The obstacles and the target region are represented on a global map, which is shared by all robots. During the mission, new obstacles can be added to the map once they are detected by a follower. A Wi-Fi communication is employed for sharing the map updates and for distribution of desired trajectories to the followers. For localization of MAVs, we assume to rely only on onboard sensors (IMU, PX4flow, and the relative visual localization system [1]) and no external positioning system is available. The global position of the formation in the environment can be obtained relatively imprecisely using any onboard global localization, for example [23].

### III. METHOD DESCRIPTION

#### A. Migrating virtual leader concept

In our method, the virtual “migrating” leader, which is employed to extend the maneuverability of the formation driving approach from [5], is initialized or reinitialized on a 3D convex hull of positions of followers in the direction of further movement of the formation (see Fig. 3). It enables to change the direction of the formation flight suddenly while keeping the virtual leader in front of the formation, which is the main assumption of the L-F approach from [5]. The position of the virtual leader ahead the formation is necessary for the coordination of followers, which required states be computed from the motion history of the virtual leader. If the virtual leader is not positioned in front of all followers, their non-collision flight is not ensured (see Fig. 2 of [24] for example) and the ability of the system to avoid dynamic obstacles is reduced. For simplification of the description of our approach, we use a circumscribed sphere of the 3D hull in Fig. 3 and then the virtual leader is placed at the intersection of the sphere and the ray from its center in the direction of the formation movement.

As mentioned in the introduction, we propose to integrate the short term control of the formation and its trajectory planning with the turning maneuvers into a single MPC-based optimization process. In MPC techniques, a finite horizon optimization control problem is solved for the kinematic model of the system from the current states over a time interval (called *control horizon*) with  $N$  transition states distributed with constant sampling time  $\Delta t$ . The control inputs are constant in the interval between two consequent transition points. We extend this scheme by another time interval (*planning horizon*)

with additional  $M$  transition states. The *planning horizon* has the variable time difference between the transition states and it is used for trajectory planning into the desired target region.

The overall trajectory of the formation from its actual state into the target region can be then encoded into a vector of constant control inputs as  $X = [v_{v,1}, v_{h,1}, \omega_1, \alpha_1, \dots, v_{v,N}, v_{h,N}, \omega_N, \alpha_N, v_{v,N+1}, v_{h,N+1}, \omega_{N+1}, \delta_{N+1}, \alpha_{N+1}, \dots, v_{v,N+M}, v_{h,N+M}, \omega_{N+M}, \delta_{N+M}, \alpha_{N+M}]$ , which is used as the optimization vector under the MPC scheme in each planning step. The optimization is frequently restarted from the current state and therefore only the begging of the vector  $X^*$  obtained as result of the optimization is used for MAV control. The vector  $X$  consists of the vertical velocity  $v_{v,\cdot}$  [ $\text{m}\cdot\text{s}^{-1}$ ], the horizontal velocity  $v_{h,\cdot}$  [ $\text{m}\cdot\text{s}^{-1}$ ], the angular velocity around the vertical axis with origin in the position of the virtual leader  $\omega$  [ $\text{rad}\cdot\text{s}^{-1}$ ], the time interval between consequent transition points  $\delta$  [s], and the angle  $\alpha$  [ $\text{rad}$ ] by which the formation heading is rotated (see Fig. 3 for visualization the rotation maneuver).

Then, we solve trajectory planning and control of the formation as minimization of the cost function  $CF(X)$ , subject to sets of constraints  $C_{inputs}$ ,  $C_{target}$ , and  $C_{acc}$ , such as

$$C_{inputs} := \begin{cases} v_{v,min} \leq v_{v,k} \leq v_{v,max}, \\ |\omega_k| \leq v_{h,k}/R_{min}, \\ \text{if } \alpha_k = 0, 0 \leq v_{h,k} \leq v_{h,max}, \text{ else, } v_{h,k} = 0, \end{cases} \quad (1)$$

where  $k \in \{1 \dots N + M\}$ . The limits  $v_{v,min}$  and  $v_{v,max}$  are given by the motion capabilities of the MAVs in the formation. The angular velocity is limited by the minimal allowed turning radius  $R_{min}$  of the formation that ensures smooth movement of all followers in the desired shape. As shown in [5],  $R_{min}$  depends on the size of the formation and it is the main bottleneck of using the L-F technique [5] in a cluttered environment with obstacles, where the limited turning radius decreases maneuverability of the formation. Solving this limitation is the motivation of the proposed migrating virtual leader approach. Also, the limit  $v_{h,max}$  depends on the motion capabilities of MAVs and the desired shape of the formation since the robot following the outer track goes faster than the robot closer to the center of the turning if  $\omega_k \neq 0$ , as described in [5], [18]. The constraint  $C_{target} := dist_{target} - r_{target} \leq 0$  ensures that the distance  $dist_{target}$  of the last point of the trajectory and the center of the target region is smaller than its radius  $r_{target}$ . The set of constraints  $C_{acc}$  is defined as

$$C_{acc} := \begin{cases} a_{max} - |v_{h,k+1} - v_{h,k}|/\Delta t \leq 0, k \in 1 \dots N - 1 \\ a_{max} - |v_{h,k+1} - v_{h,k}|/\delta_k \leq 0, \\ k \in N \dots N + M - 1, \end{cases} \quad (2)$$

where  $a_{max}$  is a limit on the acceleration of the formation in the horizontal plane. Constraints  $C_{inputs}$  and  $C_{acc}$  are important for integration of the rotating maneuvers into the overall trajectory. These sets ensure that the formation stops

before the virtual leader migration, as required for the L-F technique.

The cost function is defined as

$$\begin{aligned}
 CF(X) = & \min \left\{ 0, \frac{dist_{obst}(X) - r_s}{dist_{obst}(X) - r_a} \right\}^2 + \sum_{j=N+1}^{N+M} \delta_j \\
 & + \sum_{j=2}^{N+M} ((v_{v,j} - v_{v,j-1})^2 + (v_{h,j} - v_{h,j-1})^2 + (\omega_j - \omega_{j-1})^2) \\
 & + \sum_{j=1}^{N+M} \ln(|c_1 \cdot \alpha_j| + 1),
 \end{aligned} \tag{3}$$

where the first term is the obstacle avoidance function originally described in [25], which is zero if the minimal distance  $dist_{obst}(X)$  between the trajectory (described by the vector of control inputs  $X$  applied on the kinematic model for the actual state) and an obstacle is greater than safety radius  $r_s$  and increases if an obstacle approaches into the sphere defined by the safety radius. The spherical representation of the formation is important due to the employed visual relative localization, and it ensures the direct visibility between the followers. The value of the first term of  $CF(X)$  goes to infinity as an obstacle approaches the avoidance radius  $r_a$ . The time of flight penalization in the second term depends only on the planning horizon since the transition states are distributed with the constant frequency in the control horizon. The third term penalizes movement oscillations, which may occur in MPC control scheme, and it is aimed to prevent the system from an undesired deviation in control inputs. The last term, which penalizes the turning maneuvers, can be used (with its influence set by constant  $c_1$ ) in applications, where trajectories without the turning maneuver are preferred even at the cost of a longer time to the target. In this paper, a nonzero weight  $c_1$  of this term is used only in the experiment in table I.

### B. Stabilization of followers

The desired trajectories for each of the followers are simply obtained from the result of the optimization  $X^*$  using the L-F method from [5] as sketched in section II. As depicted in Fig. 4, the only issue that needs to be considered is the modification of  $p_i, q_i$  coordinates after each turning maneuver for all MAVs:

$$\begin{aligned}
 p_i &= \cos(\phi + \alpha)(P_x - P_{ix}) + \sin(\phi + \alpha)(P_y - P_{iy}), \\
 q_i &= \cos(\phi + \alpha)(P_y - P_{iy}) + \sin(\phi + \alpha)(P_x - P_{ix}),
 \end{aligned}$$

where  $i \in 1 \dots N_{MAV}$ ,  $P := [P_x, P_y]$  is the new position of the virtual leader,  $\phi$  is its heading before its migration, and  $P_i := [P_{ix}, P_{iy}]$  is position of the  $i$ -th follower, which is obtained as  $P_i := P_0 + \cos(\phi)[p_i, q_i] - \sin(\phi)[q_i, p_i]$ .  $P_0$  is position of the virtual leader before its migration. Coordinate  $h_i$  stays unchanged during the turning maneuver.

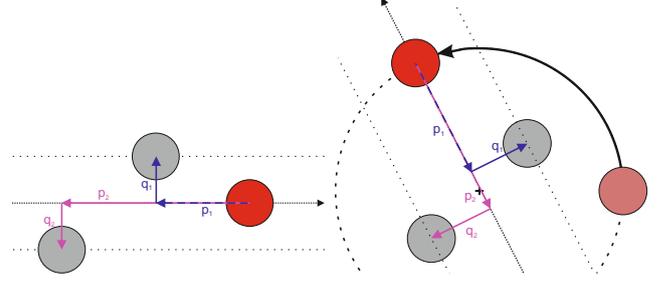


Fig. 4. A formation before (left) and after (right) the virtual leader migration. The virtual leader is represented by the red circle and followers by gray circles.

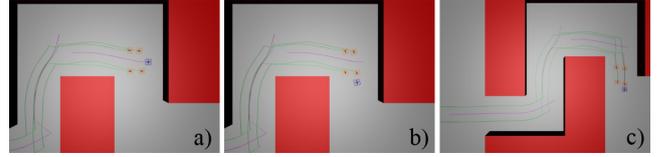


Fig. 5. Formation flying in a narrow corridor, which requires sudden changes of formation velocity to be able to fly around corners without collisions with obstacles and mutual collisions. MAVs are denoted by orange and the virtual leader by blue contours. a) The formation before the third migration of the virtual leader. b) Migration finished. c) Mission accomplished.

The desired trajectories of each MAV can be then directly followed by the onboard embedded MPC solver described in [15]. Unfortunately, the solution in [15] does not include an obstacle avoidance function. If a dynamic environment is considered, inter-vehicle collision avoidance in case of a follower failure is required or the ability to temporarily change the formation shape is needed (as shown in Fig. 6), the algorithm described in section VI.C of our previous work [16] on UGV-formations control needs to be applied in a slower planning loop before employing the fast onboard MPC. Due to the space limit of this paper, we have to skip the description of the extension of the algorithm in [16] for using with MAVs and into 3D, but it is straightforward.

## IV. NUMERICAL SIMULATIONS

In the presented experiments, a C Feasible Sequential Quadratic Programming (CFSQP) library [26] is used for solving the optimization problem defined in III-A. Initial formation parameters in the simulations with snapshots shown in Fig. 5 and 6 and results in table I are  $p_i = \{0.5; 0.5; 0.9; 0.9\}$ ,  $q_i = \{0.4; -0.4; 0.4; -0.4\}$ , and  $r_i = \{0; 0; 0.2; 0.2\}$ , while for the simulation in Fig. 8  $p_i = \{0.55; 0.55; 0.3; 0.3; 0.3; 0.3; 0.8; 0.8; 0.8; 0.8\}$ ,  $q_i = \{0; 0; -0.4; -0.4; 0.4; 0.4; 0.4; 0.4; -0.4; -0.4\}$ , and  $r_i = \{-0.2; 0.2; 0.2; -0.2; 0.2; -0.2; 0.2; -0.2; 0.2; -0.2\}$ .

In the first scenario in Fig. 5, the width of the corridor in the workspace does not allow to use the basic L-F approach from [5] and the turning maneuvers with the sudden change of the position of the virtual leader are required. During these turning maneuvers, the formation is forced to stop its movement, the position of the virtual leader is changed, and the formation continues in the new direction.

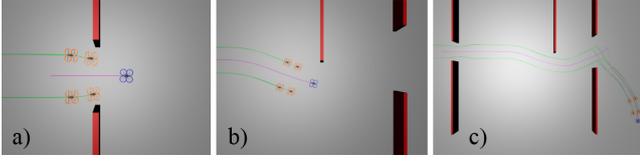


Fig. 6. Ability of deformation of the formation shape enforced by the narrow passage (a) and avoidance of a lately detected obstacle (b).

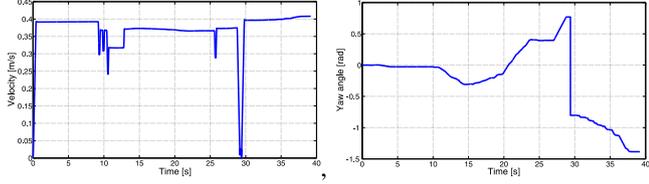


Fig. 7. Progress of velocity (left) and yaw (right) of the virtual leader in the experiment presented in Fig. 6.

In the map in Fig. 6, which is inspired by an office-like environment, the “door” in the first obstacle is smaller than the formation in its desired shape, and the formation has to be temporarily narrowed there to be able to pass through. A response of the formation to lately detected obstacle is demonstrated by the straight obstacle inside the room, which is considered as unknown at the beginning of the simulation. The turning maneuver at the end of the experiment (see the yaw and velocity profiles in Fig. 7 shortly before the 30th second of the simulation) is not necessary due to the environment constraints, but this solution is evaluated by a cheaper cost value than a longer solution without the maneuver. The temporary decrease of speed of the formation motion between the 9. and 13. second of the simulation is caused by the later detected obstacle.

In the simulation in Fig. 8, the system scalability is tested using the 10-MAVs formation. The progress of cost function values of solutions obtained in each planning step is shown in Fig. 9. The cost is linearly decreasing due to the constant speed of the formation movement (the trajectory is shortened by the approximately same distance in each planning step), except the intervals along the displacement of the virtual leader position, where the decrease of cost values is faster. The progress of heading direction of the virtual leader during the simulation is shown in Fig. 9. The airflow influence of the top MAVs to the bottom MAVs, which is observed in real-world experiments, is neglected in this simulation. This enables us to verify the ability of the system to stabilize large compact formations of 3D shape. Real-world applicability of such a compact 3D formation would be possible only with smaller MAVs (as shown in [27]) than using the platforms considered in this paper.

The influence of weight of penalization of the rotation maneuvers  $c_1$  in the cost function (3) is shown in table I. The aim of this statistic, which was obtained by 2100 simulations in 300 randomly generated scenarios (see examples of the scenarios in Fig. 10), is to verify the reliability of the method and its ability to choose between smooth longer trajectories and shorter trajectories with included turning maneuvers based on

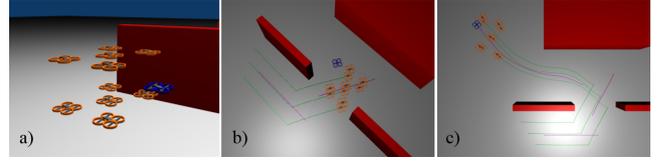


Fig. 8. A formation of 10 MAVs flying in a narrow corridor. Airflow effect from MAV propellers to neighboring vehicles was not considered in this simulation. a) Initial position. b) The formation after the second migration. c) Target reached.

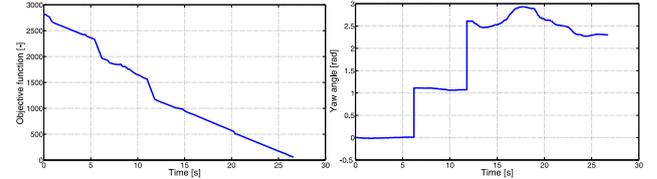


Fig. 9. Progress of cost function values of solutions of the virtual leader motion planning problem (left) and yaw (right) of the virtual leader in the simulation with snapshots presented in Fig. 8.

the particular application. In the table, the trajectory curvature value is obtained as mean curvature along the 300 trajectories followed by the formation. The flight time is the mean total time of flight to reach the desired target location and the minimum obstacle distance is mean minimal distance between an MAV and an obstacle in the 300 runs of the algorithm for each setting of weight  $c_1$ .

Results of this analysis are also displayed as a box plot in Fig. 11, where the red line represents the median and edges of the blue rectangle are the 25-th and 75-th percentiles. The whiskers are extended to the most extreme data points that are not considered as outliers, which are plotted individually as the red crosses. Points are considered as outliers if they are larger than  $q_3 + w(q_3 - q_1)$  or smaller than  $q_1 - w(q_3 - q_1)$ , where  $q_1$  and  $q_3$  are the 25-th and 75-th percentiles, respectively, and  $w$  is the whisker length.

Results confirmed that the number of turning maneuvers (counted in the first three rows of the table) significantly decreases with increasing value of the weight of penalization of these maneuvers in the cost function. With the increasing number of the turning maneuvers, the curvature of the obtained trajectories increases and also the minimal distance between the formation and obstacles decreases if the trajectories without turning maneuvers are preferred, which both increase flight safeness. This shows the usefulness of the method in cluttered

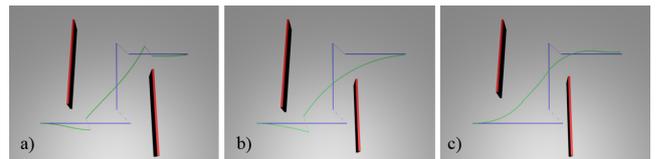


Fig. 10. Examples of solutions of different scenarios found by the proposed algorithm in the statistics presented in Fig. 11 and table I. The initial trajectory is plotted in blue, obtained trajectories in green and obstacles in red color. a) A solution with 2 maneuvers, b) with 1 maneuver, c) without a maneuver.

weight	0	25	50	75	150	300	1000
2 manoeuvres	299	257	194	140	109	92	38
1 manoeuvre	1	38	85	129	150	151	175
0 manoeuvres	0	5	21	31	41	57	87
min. obst. dist.	1.25	1.24	1.23	1.22	1.19	1.16	1.09
traj. curvature	2.56	3.08	3.47	3.89	4.17	4.43	5.04

TABLE I

INFLUENCE OF THE WEIGHT  $c_1$  OF THE TERM IN THE COST FUNCTION (3) THAT PENALIZES THE ROTATION MANEUVERS IN THE SOLUTION. EACH SETTING OF THE PARAMETER  $c_1$  WAS TESTED IN 300 SIMULATIONS OF RANDOMLY GENERATED SCENARIOS.

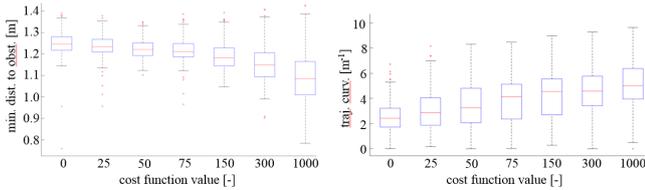


Fig. 11. Minimal distance to obstacles (a) and curvature of the trajectory (b) of solutions obtained by the algorithm with different setting of the parameter  $c_1$ . Each statistic was obtained from 300 simulations of randomly generated scenarios.

environments, additionally to the ability to solve situations, where a feasible trajectory does not exist if the turning maneuvers are not enabled, as was shown in the previous simulations.

## V. CONCLUSION

In this paper, a novel concept of motion planning and stabilization of formations of micro aerial vehicles based on a dynamic virtual leader was designed and experimentally evaluated. This concept is suited for utilization of onboard visual relative localization of neighboring MAVs, which can be considered as an enabling technique for deployment of large teams of unmanned helicopters in cluttered and GPS-denied environments. The results of theoretical analyzes of system stability and convergence of the MPC-based planning and results of the numerical simulations show that the proposed method is capable of flexible formation flying in cluttered GPS-denied environments and it is robust to external disturbances caused by the wind, air turbulence close to walls, and imprecise sensors.

## ACKNOWLEDGMENT

The work has been supported by contract with Khalifa University of Science via CTU HS No. 13167/830/8301616C, by MSMT under grant No. 7AMB16FR017, and by GACR via grant No. 16-24206S.

## REFERENCES

- [1] T. Krajník, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, vol. 76, pp. 539–562, 2014.
- [2] J. Faigl, T. Krajník, J. Chudoba, L. Preucil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *IEEE ICRA*, 2013.
- [3] Y. Zhang and K. Ma, "Lighting design for globally illuminated volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 2946–55, 2013.

- [4] G. Droge, "Distributed virtual leader moving formation control using behavior-based mpc," in *ACC*, 2015.
- [5] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Auton. Syst.*, vol. 46, pp. 65–78, 2004.
- [6] J. Wook Kwon, J. H. Kim, and J. Seo, "Article multiple leader candidate and competitive position allocation for robust formation against member robot faults," *Sensors*, vol. 15, no. 5, pp. 10771–10790, 2015.
- [7] S. Swaminathan, M. Phillips, and M. Likhachev, "Planning for multi-agent teams with leader switching," in *IEEE ICRA*, 2015.
- [8] P. A. Sperandio Giacomini and E. M. Hemerly, "Reconfiguration between longitudinal and circular formations for multi-uav systems by using segments," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 2, pp. 339–355, 2014.
- [9] O. Cetin and G. Yilmaz, "Real-time autonomous uav formation flight with collision and obstacle avoidance in unknown environment," *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2016.
- [10] K. A. Ghamry and Y. Zhang, "Formation control of multiple quadrotors based on leader-follower method," in *ICUAS*, 2015.
- [11] A. Monteriu, "Nonlinear decentralized model predictive control for unmanned vehicles moving in formation," *Information Technology and Control*, vol. 44, no. 1, pp. 89–97, 2015.
- [12] S. Garrido, L. Moreno, and P. U. Lima, "Robot formation motion planning using fast marching," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 675 – 683, 2011.
- [13] C. Papachristos, K. Alexis, and A. Tzes, "Model predictive hovering-translation control of an unmanned tri-tiltrotor," in *IEEE ICRA*, 2013.
- [14] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust explicit model predictive flight control of unmanned rotorcrafts: Design and experimental evaluation," in *ECC*, 2014.
- [15] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *MMAR*, 2016.
- [16] M. Saska, V. Spurny, and V. Vonasek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, no. B, pp. 379–397, 2016.
- [17] M. Saska, V. Vonasek, and L. Preucil, "Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs," *Journal of Intelligent and Robotic Systems*, vol. 72, no. 2, pp. 239–261, 2013.
- [18] M. Saska, V. Vonasek, T. Krajník, and L. Preucil, "Coordination and Navigation of Heterogeneous MAV&UGV Formations Localized by a "hawk-eye"-like Approach Under a Model Predictive Control Scheme," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, September 2014.
- [19] M. Saska, T. Krajník, V. Vonasek, Z. Kasl, V. Spurny, and L. Preucil, "Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1-4, pp. 603–622, 2014.
- [20] M. Saska, V. Vonasek, and L. Preucil, "Roads Sweeping by Unmanned Multi-vehicle Formations," in *IEEE ICRA*, 2011.
- [21] V. Spurny, T. Baca, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *MMAR*, 2016.
- [22] D. J. E.G Gilbert and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 2, pp. 193 – 203, 1988.
- [23] D. Scaramuzza and at al., "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.
- [24] M. Saska, J. Mejia, D. Stipanovic, V. Vonasek, K. Schilling, and L. Preucil, "Control and Navigation in Manoeuvres of Formations of Unmanned Mobile Vehicles," *European Journal of Control*, vol. 19, no. 2, pp. 157–171, March 2013.
- [25] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Šiljak, "Cooperative avoidance control for multi-agent systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, pp. 699–707, 2007.
- [26] J. Z. C. Lawrence and A. Tits, "User's guide for cfsqp version 2.5," University of Maryland, 1997.
- [27] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

# E

---

KEY ARTICLE [5] - ICRA 2014

---

©[2018] IEEE. This paper is available in proceedings of 2014 IEEE International Conference on Robotics and Automation: **M. Saska**, J. Vakula, and L. Preucil: Swarms of Micro Aerial Vehicles Stabilized Under a Visual Relative Localization, 2014.



# Swarms of micro aerial vehicles stabilized under a visual relative localization

Martin Saska, *Member, IEEE*, Jan Vakula, Libor Přeučil, *Member, IEEE*

**Abstract**—A stabilization and control technique developed for steering swarms of unmanned micro aerial vehicles is proposed in this paper. The presented approach based on a visual relative localization of swarm particles is designed for utilization of multi-robot teams in real-world dynamic environments. The core of the swarming behaviour is inspired by Reynold’s BOID model proposed for 2D simulations of schooling behaviour of fish. The idea of the simple BOID model, with three simple rules: Separation, Alignment and Cohesion, is extended for swarms of quadrotors in this paper. The proposed solution integrates the swarming behaviour with the relative localization and with a stabilization and control mechanism, which respects fast dynamics of unmanned quadrotors.

The proposed method aspires to be an enabling technique for deployment of swarms of micro areal vehicles outside laboratories that are equipped with precise positioning systems. The swarming behaviour as well as the possibility of swarm stabilization with the visual relative localization in the control feedback are verified by simulations and partly by an experiment with quadrotors in this paper.

## I. INTRODUCTION

The possibility of deployment of large groups of unmanned Micro Aerial Vehicles (MAVs) closely cooperating together brings new potentialities for autonomous robotics. MAV swarms are beneficial in numerous applications including cooperative surveillance, reconnaissance and monitoring tasks, search and rescue missions, searching for sources of pollution, sensory data acquisition and various military applications. To be more specific, swarms of MAVs can be employed for monitoring of natural disasters (floods, forest fires), patrolling of objects or protected areas (ammunition depots, borders), surveillance of crowds (cultural and sport events, demonstrations), monitoring of industrial accidents (plume of toxic gas tracking, concentration of pollutants measuring), sensing in large environments (measuring of signal coverage, smog concentration) and many more.

Most of these tasks involve the utilization of swarms of MAVs in environments without any pre-installed infrastructure for precise localization of robots. Although available global localization systems (such as GPS) can be used for a rough positioning of the whole swarm, the precision of these systems is insufficient for relative localization of closely operating MAVs. Knowledge of the relative position of neighbouring MAVs in the swarm is crucial for collision avoidance within the team and for any coordination of MAVs,

The authors are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague 6, Czech Republic  
{saskam1,vakuljan}@fel.cvut.cz,  
preucil@labe.felk.cvut.cz

if it is required by the application (cooperative actions). The swarm control approach presented in this paper is suited for utilization of an onboard visual system for relative localization in large teams of unmanned micro quadrotors. The employed localization method provides precise information on relative positions of neighbouring robots with an update rate, which is sufficient for the swarm stabilization.

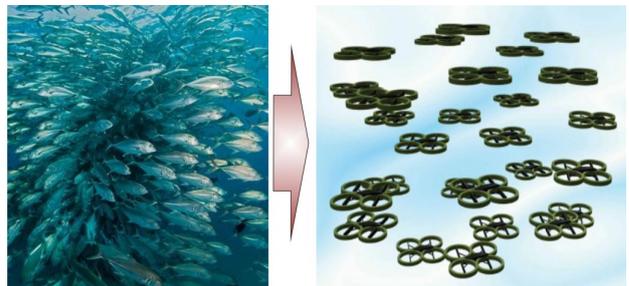


Fig. 1. Swarm of quadrotors controlled by rules derived from behaviours observed in schools of fish.

The presented algorithm designed for the swarm control with obstacle avoidance ability is inspired by the Reynold’s BOID model [1], which was developed to simulate schooling behaviour of fish. In our method, the Reynold’s basic rules (originally designed for control of 2D holonomic particles) are interpreted for stabilisation of swarms of MAVs under the relative localization as follows:

- Separation - avoid crowding neighbours (a short range repulsion to avoid collisions and to reduce mutual airflow effects caused by propellers).
- Cohesion - keep swarm compact to enable its stabilisation using the relative localization (a long range attraction to keep swarm particles in the range of the visual localization).

The alignment is realized via a target attraction, which steers swarm particles towards a common target. Besides, we consider another rule, which is important in applications in environment with obstacles or with possible human-swarm interaction, where the swarm has to keep a sufficient safe distance from people:

- External avoidance - avoid obstacles (a short range repulsion based on flight direction).

In nature, flocks (fish schools) can avoid obstacles or escape from predators very fast in a cooperative way and without mutual collisions. In analogy to the school of fish, also MAVs equipped with the onboard sensors for relative

localization are able to detect neighbours within a short detection range and to determine their position and change of the position. This information employed in the control feedback enables the swarm to keep compact and avoid obstacles without any explicit communication. Moreover, even if not all robots of the group are able to detect the particular obstacle, its avoidance is ensured via a sequential transfer of motion changes through the swarm by continual observing of relative positions of neighbours (the visual relative localization system that we have introduced in [2] is employed). The presented swarm stabilization approach is an extension of work in [3], which was designed for control of swarms of ground robots.

From the robotic point of view, such an approach based on flocking behaviour has two important advantages regarding the large MAV swarms: 1) It is strictly decentralized based only on the information gathered by onboard sensors and therefore it is scalable. 2) The swarming rules are computationally efficient and so the control algorithm can be run in a fast loop even using the computational power available onboard of simple MAVs.

1) *State of the art*: Recently, the research of swarms of autonomous vehicles covers broad areas of robotics including aspects of task allocation and strategies for solving multiple tasks [4], communication and maintenance of connectivity within the team [5], a modeling of the swarm behavior by predicting of individual behaviours [6], or a collision avoidance within the swarm [7]. The topic involved in this paper is related mainly to control and stabilization of swarms of MAVs [8], [9]. The most related to the research proposed in this project is presented in [10], where swarming behaviours of ground robots in a planar environment are investigated. The aim of our approach is also to stabilize swarms of autonomous robots (in our case MAVs) in a desired shape while maintaining a small distance among themselves. Beyond the research presented in [10], we design principles of swarming rules to satisfy requirements on the visual system of relative localization in 3D, which enables *to take swarms of flying robots outdoor*. This is one of the most important contributions of our method in comparison with the aforementioned algorithms that have been verified usually via numerical simulations or rarely using ground vehicles in laboratories. In literature, one can find also number of works based on the Reynold's model, e.g. [11], [12]. These algorithms are also mostly designed to steer ground robots or 3D particles, which are often considered as dimensionless points. There is lack of approaches considering limitations of MAV multi-robot systems or even investigating possibility of deployment of swarms of aerial robots in real-world missions.

## II. PROBLEM STATEMENT AND PRELIMINARIES

Let us assume a group of  $N$  quadrotor MAVs equipped with omnidirectional vision system capable of relative localization of neighbouring robots and obstacles in a limited range. The robots are identical to each other (the localization does not provide the identification of neighbours) and there is

no communication within the swarm available. The obstacles are considered as a set of simple objects (spheres in the experiments) with known relative positions from the MAVs. A complex map, could be represented by such objects with arbitrary precision, but this paper is not focussed on the environment representation/mapping.

The problem solved in this paper is to stabilize the robots in a compact swarm in an environment with obstacles and, if it is required by the application, to reach a target region. In the case of the target following, we also assume that the robots are able to detect the target position or a direction into the target. The control algorithm must respect both the relative positions of neighbouring MAVs and the constraints of the relative localization (local range). Clearance between neighbouring MAVs and between MAVs and obstacles must ensure a collision free movement.

### A. Quadcopter model and control

In this paper, we use a quadrotor vehicle model [13] with four identical propellers located at vertices of a square (see Fig. 2 b)). Each of the propellers  $j$  generate a thrust  $f_i^j$  along its axis. For each MAV  $i$ , we consider an inertial reference frame and a body-fixed frame with origin located at the center of mass of the MAV. The relative position of these frames is defined by the location of the center of mass  $x_i \in \mathbb{R}^3$  in the inertial frame and by the rotation matrix  $R_i \in \mathbb{R}^{3 \times 3}$  from the body-fixed frame to the inertial frame. The inertial reference frame is different for each MAV since they cannot communicate with each other, which would be necessary for unification of the reference frames.

The motion model of MAVs according to [13] is

$$\begin{aligned} \dot{x}_i &= v_i, \\ m_i \dot{v}_i &= m_i g e - f_i R_i e, \\ \dot{R}_i &= R_i \hat{\Omega}_i, \\ J_i \dot{\hat{\Omega}}_i + \Omega_i \times J_i \hat{\Omega}_i &= M_i, \end{aligned} \quad (1)$$

where  $v_i \in \mathbb{R}^3$  is velocity of the center of mass in the inertial frame,  $m_i \in \mathbb{R}$  is weight of the MAV,  $\Omega_i \in \mathbb{R}^3$  is angular velocity in the body-fixed frame,  $J_i \in \mathbb{R}^3$  is inertia matrix with respect to the body frame. The hat symbol  $\hat{\cdot}$  is defined by the condition  $\hat{x}y = x \times y$  for all  $x, y \in \mathbb{R}^3$ ,  $g$  is the gravity acceleration and  $e = [0, 0, 1]$ . The total moment  $M_i \in \mathbb{R}^3$  along all axes of the body-fixed frame and the thrust  $f_i \in \mathbb{R}$  are control inputs of the plant. The total thrust,  $f_i = \sum_{j=1}^4 f_i^j$ , acts in the direction of the axis of the body-fixed frame which is orthogonal to the plane defined by the centres of the four propellers. The control inputs are obtained by the tracking controller presented in [13], which is employed to reach the new locations of MAVs given by the swarming approach described in section III.

### B. Visual relative localization of swarm particles

The swarming principles investigated in this paper are designed for using the light-weight vision based embedded system of the relative localization of particles within the

robotic group. The system developed within our team (see [2] for technical details and performance analyses) is based on a detection of black and white patterns with precision in units of centimeters for distances in units of meters. This operational range and precision are sufficient for the stabilization of groups of MAVs cooperatively acting in close swarms.

Although, this system is sufficiently robust and precise, it has a drawback concerning stabilization of large groups of MAVs, since its operational range is limited (depends on resolution of the employed cameras and size of the pattern). Therefore, it is crucial to incorporate the operational constraints into the swarming rules and to keep MAVs in appropriate relative positions regarding the relative localization. The operational constraints are described by a model of the localization arising from theoretical analyses of the vision system and experimental evaluation of the system performance in real scenarios presented in [2].

In addition, the localization system (in its simple version) may not identify which MAV is recognized in robot's neighbourhood. The possibility of particular MAVs identification would require more complicated patterns or patterns of different colours, which would be at the cost of decreased reliability, precision and operational range. The proposed nature inspired swarm control technique is especially appealing to deal with this limitation, since the swarm theory assumes utilization of homogeneous particles. Therefore, also the relative interaction of swarm particles (described in Section III-A) considers the neighbouring MAVs as anonymous entities.

### III. FLOCKING BEHAVIOUR WITH OBSTACLE AVOIDANCE ABILITY

A novel MAV flocking approach based on the Reynold's model of the coordinated animal motion is proposed in this section. Taking the steering behaviours of Reynold's boids as an inspiration, we have incorporated the local interactive forces from neighbouring robots into the dynamics of MAVs. We have also included effects given by proximate obstacles and a force pushing the overall swarm into the desired target region. This active force would not be necessary in case of employment of the algorithm as a component of higher level motion planning algorithms. Then the flocking MAV control can be used for swarm stabilization and for emergency intervention due to its avoidance and escaping behaviour, while the high level planner would solve the swarm navigation.

The proposed swarming approach, which provides control inputs based on surrounding quadrotors and near obstacles in a decentralized manner, enables employment of large swarms of relatively localized MAVs. This control scheme provides a sufficiently robust and fast solution, which requires minimal computational power and simple sensors available onboard of MAVs. The swarming behaviour is designed as a combination of the relative interactions of swarm particles, the attraction of the target and the interaction with surrounding environment as follows.

#### A. Relative interaction of swarm particles

The core of the proposed swarming behaviour is based on the relative interaction between particles of the group. The relative positions of neighbouring robots are composed into the separate rules for each of the individuals independently, which results in the required behaviour of the entire group. This decentralized concept is suitable for the proposed scheme, which is based on the relative visual localization of MAVs. In such a swarm, each robot is capable of localizing only neighbours in its limited surroundings. The effects of neighbouring robots are integrated into the MAV control by the individual force

$$\mathbf{F}_{ind_i} = \sum_{j,j \neq i}^N e_{ij} \mathbf{F}_{ind_{ij}}, \quad (2)$$

where  $e_{ij}$  is a distance weight function and  $\mathbf{F}_{ind_{ij}}$  is an interactive force.

The distance function emulates the sensor range of the employed visual relative localization. It ensures that the interactive force is considered in the MAV control rule only if the neighbouring vehicles are in relative positions that enable their confident localization. The distance function is important mainly for realistic simulations of the MAV swarm behaviour. In the experiments with real quadrotors, this function is replaced by a tag indicating validity of the sensory data. Such a tag is provided by the utilized visual relative localization system [2]. The magnitude of the distance weight function depends on the relative distance  $\mathbf{L}_{ij}$  between two neighbouring quadrotors,  $i$  and  $j$ , as:

$$e_{ij} = \frac{1}{e^{a\mathbf{L}_{ij}-b} + c} + \frac{1}{e^{0.5a\mathbf{L}_{ij}-b} + c}. \quad (3)$$

The weight function is used with constant values  $a = 5$ ,  $b = 4$ ,  $c = 0.6$  in this paper. The interactive force

$$\mathbf{F}_{ind_{ij}} = K_d(\|\mathbf{L}_{ij}\| - L_r)\mathbf{L}_{ij} + D_d \frac{d\mathbf{L}_{ij}}{dt} \quad (4)$$

is designed as a spring-damper model. It enables stabilization of the robots in a relative distance equal to a desired intra-robot distance  $L_r$ . The constant  $L_r$  has to be chosen smaller than the range of the relative localization to ensure the stabilization of the group. On the contrary, too small value of  $L_r$  can increase possibility of collisions and mutual disturbances by air streams from propellers.  $\mathbf{L}_{ij}$  is a vector between robots  $i$  and  $j$ , which is given by the visual relative localization. The constants, which are used as  $K_d = 1.5$  and  $D_d = 2$  in experiments in this paper, affect the rate of the convergence into the equilibrium. The required derivative of the relative position of neighbours is obtained by Kalman filtering of the data from the relative localization system [2].

#### B. Swarm attractivity

The rule of the swarming behaviour presented in this subsection deals with the attraction of the swarm to the target position. This ability of the swarm movement into the desired goal position is achieved by integration of an attractive force

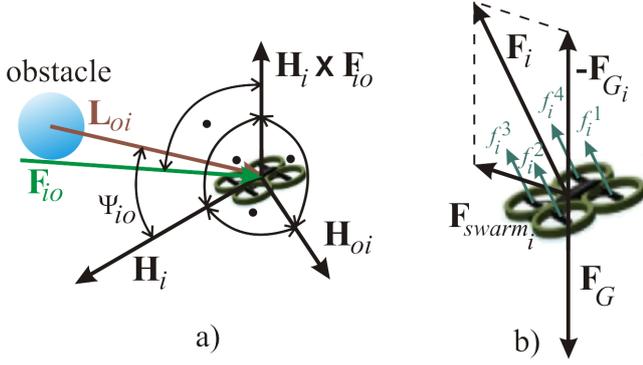


Fig. 2. a) Forces of interaction with surrounding environment. b) Integration of the total swarming force into the MAV control.

to the MAV control scheme. The attractive force is pointing at the goal and again it works as a spring-damper model:

$$\mathbf{F}_{goal_i} = K_g \frac{\mathbf{L}_{ig}}{\|\mathbf{L}_{ig}\|} + D_g \frac{d\mathbf{L}_{ig}}{dt}. \quad (5)$$

Here, the required equilibrium that has to be achieved is a zero relative distance between  $i$ -th robot and the goal position.  $\mathbf{L}_{ig}$  is the relative position vector from the  $i$ -th MAV to the goal. The constants, which are  $K_g = 1.1$  and  $D_g = 2$  in experiments in this paper, control the rate of convergence of the group. In comparison with the eq. (4), where the force magnitude is changing based on the relative distance to the neighbours, the attractive force is normalized to ensure the same swarming behaviour along the whole trajectory to the goal.

As it is well known in robotics, such a simple attractive force suffers from local extremes in complex environments with obstacles, which may result in undesirable oscillations or a dead-lock. Nevertheless, this navigational approach is sufficient for the verification of swarming principles based on the relative visual localization, which is the main purpose of this paper.

### C. Interaction with environment

The obstacle avoidance behaviour is essential for the real-world deployment of autonomous robots. Incorporation of an obstacle avoidance function directly into the swarming rules enables a very fast response to changing environment. In the proposed approach, the avoidance manoeuvre is realized by a translational reshaping of the swarm, which is caused by the interactions between particles of the group. Thus some individuals in the group may perform obstacle avoidance even without sensing the obstacle directly. This results in the required evasive action of the entire swarm without any centralized command.

The arising reshaping of swarms of quadrotors caused by detected obstacles is initialized by incorporating the equation

$$\mathbf{F}_{obs_i} = \sum_{o \in \mathcal{O}} \delta e_{oi} \frac{\mathbf{H}_{oi}}{\|\mathbf{H}_{oi}\|} \quad (6)$$

into the steering rules of each MAV. Each obstacle  $o$  in the set of detected obstacles  $\mathcal{O}$  is considered in eq. (6). The magnitude of  $\mathbf{F}_{obs_i}$  depends on the direction dependence function  $\delta$  and the exponential distance function  $e_{oi}$ .

The function  $e_{oi}$ , which is designed as

$$e_{oi} = b_o e^{a_o \|\mathbf{L}_{oi}\|}, \quad (7)$$

is exponentially growing with decreasing distance  $\|\mathbf{L}_{oi}\|$  between the  $i$ -th MAV and the obstacle. The parameters of the exponential function are used as  $a_o = -3$  and  $b_o = 100$  in experiments presented in this paper.

The direction dependency function  $\delta$ ,

$$\delta = (1 + \cos(\Psi_{io})), \quad (8)$$

is important due to the fast dynamics of the MAV swarms. This function enables to generate a repulsion according to the relative angle  $\Psi_{io}$  between the vector  $\mathbf{L}_{oi}$  and the direction vector of the  $i$ -th MAV.  $\mathbf{L}_{oi}$  is the relative position vector between the MAV and the obstacle. MAVs flying towards a collision with the obstacle or in a direction which is close to the collision are influenced by the avoidance function more intensively.

$\mathbf{H}_{oi}$  is vector perpendicular to the direction vector of the  $i$ -th MAV, which is pointing away from the obstacle as shown in Fig. 2 a). The vector  $\mathbf{H}_{oi}$  is defined as

$$\mathbf{H}_{oi} = (\mathbf{H}_i \times \mathbf{F}_{io}) \times \mathbf{H}_i, \quad (9)$$

where  $\mathbf{H}_i$  is the direction vector of the MAV. The force  $\mathbf{F}_{io}$  is employed to keep the MAV in a sufficient distance from the obstacle. Also, it enables us to incorporate a prediction of the obstacle movement into the avoidance function. The force, which is defined as

$$\mathbf{F}_{io} = K_o \mathbf{L}_{oi} + D_o \frac{d\mathbf{L}_{oi}}{dt}, \quad (10)$$

deviates slightly from the vector  $\mathbf{L}_{oi}$  oriented from the center of the obstacle to center of the mass of the quadrotor, as denoted in Fig. 2a). The deviation correlates with the movement of the obstacle (included in the first-order derivative of  $\mathbf{L}_{oi}$ ). Again, the constants, which are used as  $K_o = 1.5$  and  $D_o = 2$  in experiments in this paper, influence speed of the respond to detected dynamic or static obstacles.

The total force that acts on the  $i$ -th MAV is determined as a sum of particular contributions as:

$$\mathbf{F}_{swarm_i} = \mathbf{F}_{ind_i} + \mathbf{F}_{goal_i} + \mathbf{F}_{obs_i}. \quad (11)$$

For the stabilization of the quadrotor in the required orientation, a ray from the center of mass of the MAV in the direction of  $\mathbf{F}_i$  is employed as the reference for the single MAV low-level control. The control approach addressed in [13] is used to achieve the new orientation of the MAV in the direction of  $\mathbf{F}_i$ . The low-level control is used to follow the direction of  $\mathbf{F}_i$  until the update of the swarming rules is initiated, based on the frequency of the high level control loop of the swarm stabilization.

#### IV. EXPERIMENTAL RESULTS

The experiments presented in this section have been designed to show performance of the proposed method and to analyse its key properties. The simulations shown in Fig. 5 and 8 demonstrate the ability of the algorithm to maintain a compact swarm of numerous MAVs and to avoid obstacles simultaneously. Movies of these simulations and two additional experiments are available in [14]. In the scenarios, 27 MAVs (experiments 1 and 2) or 10 MAVs (experiments 3 and 4) are initialized in a compact formation.

At the beginning of the simulation, the quadrotors autonomously increase their relative distances based on the swarming rules described above. The desired equilibrium is achieved very fast (in less than one second) as can be seen in the graph of distances from quadrotors to their closest neighbour shown in Fig. 4. Together with the swarm stabilization, the group is moving towards the desired goal position. After few steps of the first simulation (Fig. 5), the outer MAVs of the swarm detect an obstacle. The obstacle is positioned in the middle of the line connecting the start and goal positions. The quadrotors smoothly avoid the obstacle to keep the sufficient distance from the detected object during the whole manoeuvre. The distance between the obstacle and all MAVs is shown in Fig. 3. Once the obstacle is avoided, the group converges back into the flock shape, which is depicted in the last snapshot of Fig. 5, and continues towards the goal position. Reintegration of the sub-swarms is possible only if there exists at least one relative interaction between both groups. It means that the value of term  $\|e_{ij}\mathbf{F}_{ind_{ij}}\|$  is greater than zero for at least one pair of robots  $i$  and  $j$ , which belong to different sub-swarms.

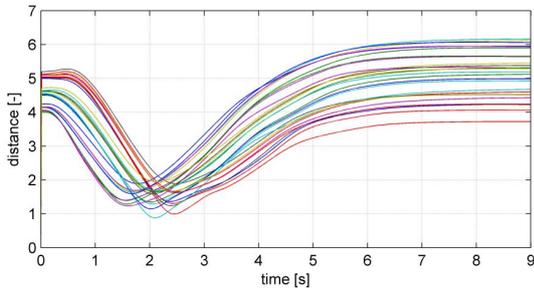


Fig. 3. Graph of distances from MAVs to the obstacle in the simulation 1.

The second simulation is initialized identically as the first one, but two additional obstacles are placed in the environment to show robustness of the approach. In snapshots of the experiment in Fig. 8 and in the video in [14], one can see that the swarm is again deformed due to the obstacles as the “avoidance signal” is propagated through the group. During the whole avoidance manoeuvre, sufficiently safe relative distances within robots of the team as well as between robots and all obstacles are kept. The values of distances between the obstacles and MAVs are plotted in Fig. 6 and the values of distances between MAVs and their closest neighbours in Fig. 7.

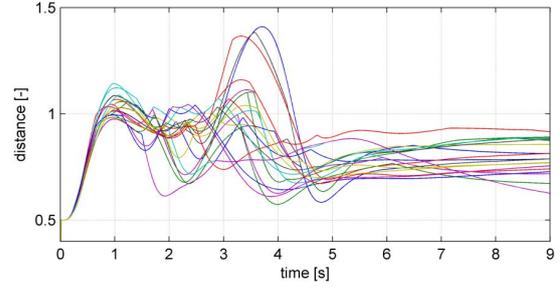


Fig. 4. Graph of distances between MAVs and their closest neighbour during the simulation 1.

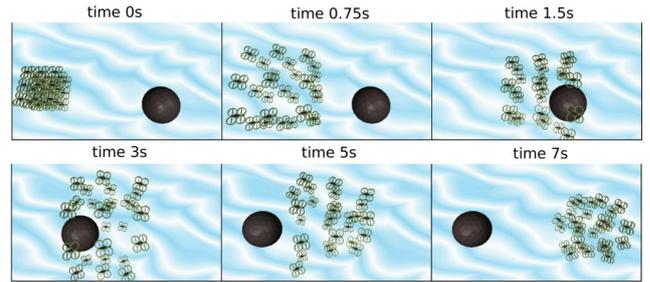


Fig. 5. Snapshots of simulation 1.

The experiments with multiple MAVs show that the swarm of quad-rotors can be stabilized using the visual relative localization in the control feedback. In the experiment shown in Fig. 9, two MikroKopters L4-ME and one AR-drone are stabilized using only the feedback from onboard cameras.

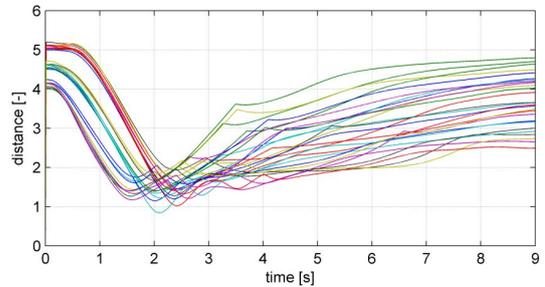


Fig. 6. Graph of distances between MAVs and the closest obstacle during the simulation 2.

#### V. CONCLUSION

A nature inspired approach for control and stabilization of swarms of unmanned micro aerial vehicles was presented in this paper. The proposed method is designed with the onboard visual relative localization of swarm members integrated into the MAVs control. The algorithm is based on simple swarming rules using an information on position of neighbours in the team and obstacles in limited sensory range without the necessity of communication. It was verified via simulations that the proposed swarming principle, computed onboard of MAVs in a strictly decentralized manner, enables

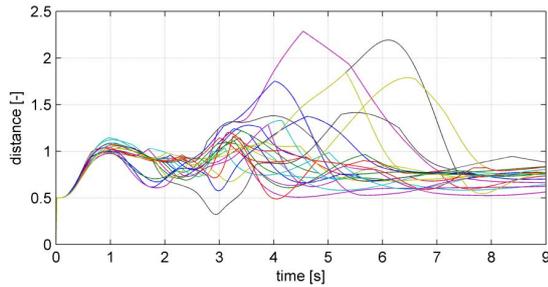


Fig. 7. Graph of distances between MAVs and their closest neighbour during the simulation 2.

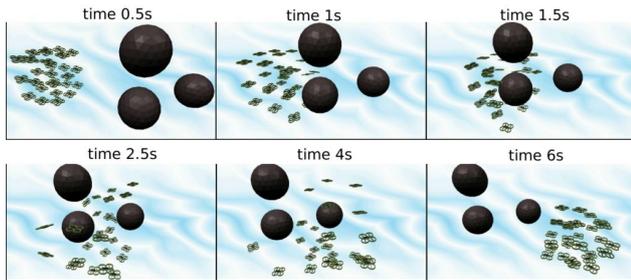


Fig. 8. Snapshots of simulation 2.

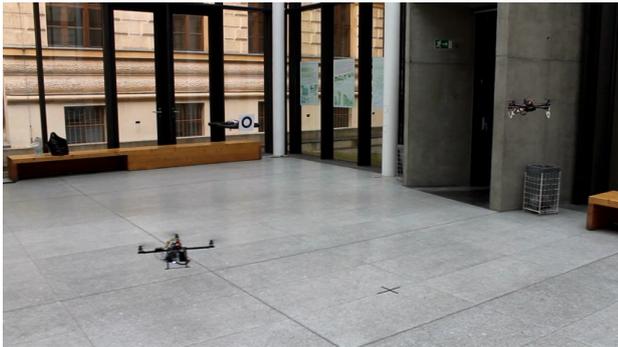


Fig. 9. Verification of the proposed swarm stabilization approach using only the relative interaction between swarm members.

stabilization of the swarm of micro quadrotors. With these desirable characteristics, the proposed approach could be an enabling technique for employing large swarms of MAVs outside laboratories equipped with a precise global position system. Such a swarming behaviour is appealing in scenarios in which a large team of robots has to move closely together: e.g. to form a distributed sensor for environment measurement, monitoring or surveillance.

## VI. ACKNOWLEDGEMENTS

This work was of M. Saska supported by GAČR under M. Saska's postdoc grant no. P10312/P756 and L. Přeučil was supported by MŠMT project Kontakt II no. LH11053.

## REFERENCES

[1] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," in *Computer Graphics*, 1987, pp. 25–34.

[2] J. Faigl, T. Krajník, J. Chudoba, L. Přeučil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[3] H. Min and Z. Wang, "Design and analysis of group escape behavior for distributed autonomous mobile robots," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[4] S. Berman, A. Halasz, M. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[5] T. Schmickl and K. Crailsheim, "Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm," *Autonomous Robots*, vol. 25, pp. 171–188, 2008.

[6] A. Winfield, W. Liu, J. Nembrini, and A. Martinoli, "Modelling a wireless connected swarm of mobile robots," *Swarm Intelligence*, vol. 2, pp. 241–266, 2008.

[7] M. Kumar, D. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 743–748, 2010.

[8] D. J. Bennet and C. R. McInnes, "Verifiable control of a swarm of unmanned aerial vehicles," *Journal of Aerospace Engineering*, vol. 223, no. 7, pp. 939–953, 2009.

[9] L. Barnes, R. Garcia, M. Fields, and K. Valavanis, "Swarm formation control utilizing ground and aerial unmanned systems," in *IEEE/RSSJ International Conference on Intelligent Robots and Systems.*, 2008.

[10] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[11] D. Gu and H. Hu, "Using fuzzy logic to design separation function in flocking algorithms," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 826–838, 2008.

[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[13] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor uav on  $se(3)$ ," in *49th IEEE Conference on Decision and Control (CDC)*, 2010.

[14] Movie, "Movie of simulations of the swarming behaviour [online]. <http://imr.felk.cvut.cz/swarm/> [cit. 2013-9-14];" 2013.

---

KEY ARTICLE [6] - ECMR 2017

---

©[2018] ECMR. This paper is available in proceedings of 2017 European Conference on Mobile Robots: D. Brandtner and **M. Saska**: Coherent swarming of unmanned micro aerial vehicles with minimum computational and communication requirements, 2017. *An extended version of the paper was submitted to journal Robotics and Autonomous Systems.*



# Coherent swarming of unmanned micro aerial vehicles with minimum computational and communication requirements

Daniel Brandtner and Martin Saska

**Abstract**—An algorithm designed for stabilization and control of large groups of micro aerial vehicles (MAVs) - multirotor helicopters - without any explicit communication is proposed in this paper. The presented algorithm enables a swarm of MAVs to maintain its coherence and perform a compact motion in complex environments while avoiding obstacles with only very limited computational and sensory requirements. The method is very robust to incomplete sensory information, it enables a fully distributed applicability, and it is highly scalable. Increasing amount of MAVs even improves the required coherence behaviour. Numerous simulations in different environments were conducted to verify the algorithm, show its potential, and explore its various configurations.

## I. INTRODUCTION

Decreasing size and increasing robustness of micro aerial vehicles (MAVs) allow us to consider deployment of large multi-MAV groups instead of heavy and well-equipped unnamed aerial vehicles. Distribution of sensory power to small flexible units increases reliability of the system and enables applications where distributed sensing or acting is necessary and which would not be possible with a single vehicle. Substitution of a large vehicle by a team of light MAVs is especially important due to safety reasons in scenarios where the autonomous system may interact with humans and where the large platforms may cause injury and damage objects in their proximity. The applications with presence of people are often located in urban and indoor environment, where global navigation satellite systems (such as GPS) are not available or their precision is not sufficient for group stabilization, and onboard sensors are required for mutual localization of individuals in the group. The proposed swarm stabilizing algorithm is designed for using vision based relative localization [1] by onboard cameras, carried by MAVs, which was already successfully employed by our team for cooperative surveillance by a team of self-localized and stabilized MAVs and for compact formations flying [2], [3], [4]. Although we call the group in [2] as a swarm, the method requires centralized planning and MAV coordination. In this paper, we propose a novel method that allows to control the visually stabilized MAV groups in a fully decentralized way and without explicit communication, which is the main requirement of swarm robotics research (see Fig. 1 for motivation).

The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. martin.saska@fel.cvut.cz



Fig. 1. Motivation: A group of MAVs above the sand dunes following the sun-set direction.

### A. Related Work

Swarm robotics, which is aimed to implement a collective behaviour without an explicit central control law, usually arises from principles of swarm intelligence dealing with decentralized, self-organized multi-agent systems [5]. Many of these swarming algorithms are inspired by behaviours observed on animal interactions. For example, the BOID model is inspired by bird flocking [6], algorithm in [7] by nesting and foraging habits of various species of insects, and approaches in [8], [9] by fish schooling.

The methods [7], [8], [9] as well as for example PSO (Particle Swarm Optimization)-based algorithms [10], [11] and methods [12], [13], [14] require global knowledge and communication unlike the approach discussed in this paper, which is strictly distributed. Also the potential field-based swarming method in [15] relies on explicit wireless communication between neighbour MAVs, while the cognitive-based algorithm in [16] uses global positioning systems and broadcasting of MAV positions. Frequently used BOID models [17], [18], similarly as the proposed approach, rely on mutual localization and stabilization of neighbouring particles, but require a full information on the relative position and even velocity of neighbours, whereas our method uses only a binary information on the presence of the neighbours in proximity of the particular MAV. This is a very important capability for control of swarms of light-weight and simple MAVs as it significantly reduces requirements on the onboard mutual localization system.

The proposed method, which is inspired by a study of coordination of ground robots in [19], enables the swarm of MAVs to perform a coherent motion towards a given target using only an onboard binary light sensor that can recognize whether the target is in sight or not. Although in [19], the group members share this information with neighbours to get a vague common idea of the target location, our method uses

individual on-board estimation of the target position without the need of the explicit communication.

To sum up the contribution of this paper, the presented algorithm enables to stabilize an MAV swarm in environments with obstacles without the need of global positioning systems and an explicit wireless communication, and with significantly limited computational and sensor requirements. Moreover, the important contribution of this paper is a comprehensive analysis of the achieved swarming behaviour aimed at confirmation/disapproval of a set of hypothesis assumed in swarm robotics literature or being compiled based on our experience with the BOID models used for MAV swarm control.

### B. Problem Definition and MAV set-up

The task solved in this paper is stabilization of a swarm of  $M$  MAVs and its navigation into a desired location in environments with static and dynamic obstacles. For the swarm coherence, it is required that all swarm members are relatively stabilized with at least  $\alpha$  neighbours in the group, which means that at least  $\alpha$  helicopters are mutually localized by each MAV using onboard sensors. Therefore, we assume that all  $M$  MAVs in the swarm are equipped with the sensors enabling to discern the adjacent robots (called "neighbours"), localize the obstacles, and detect direction of the required target. We suppose that the neighbours detection (not necessarily full localization - only a binary information on a presence of a neighbour) is ensured within a range of detection with radius  $R$ . Let us denote the actual number of detected neighbours within radius  $R$  by variable  $N$ .

Also the position of the target does not need to be observed precisely. In the presented experiments, a set of 4 simple vision sensors is deployed around the robot providing information in which quadrant corresponding to the particular sensor the target is located (no precise information on the bearing and distance of the target is required). The obstacles are localized in a detection range  $R_a$ , where  $R_a < R$ , and again the knowledge of distance to obstacles is not required and also the bearing estimation error can be very high (30° error was considered in the experiments).

## II. SWARMING MODEL

The proposed method is inspired by principles of the swarming model introduced in [19] to control a group of ground robots. We adopted the basic swarming rules, modified them, and integrated them into a multi-MAV visually-stabilized system. For MAV control strictly without communication (in [19] a communication channel was used to share the information among the robots), constraints of the onboard visual relative localization mechanism have to be considered. Additionally, motion constraints of the low level onboard model predictive control technique [20], which is used to deal with the high dynamics of MAVs, are integrated into the swarming approach.

The proposed swarming model used for MAV-swarm stabilization is composed of three behavioural states: *forward* (default setting for all MAVs), *coherence* and *avoidance*.

MAVs in the *forward state* fly straight with a constant velocity as long as they are forced to enter into another state. If the required connection with at least  $\alpha$  neighbours is lost (i.e.  $N < \alpha$ ) for an MAV, it enters into the *coherence state*, in which turns and flies back until the connection is restored. Once the MAV enters back into the localization range ( $N \geq \alpha$ ), it performs a random turn and returns back into the *forward state*. The coherence manoeuvre can be triggered only once per  $cf$  cycles to allow the MAV to regain its visual connection. The random turn is applied if only the swarm coherence is required and the swarm movement direction is not controlled. If a desired target is required to reach by the swarm members, the turning is influenced by the estimated position of the target as mathematically described in Algorithm 1. A proper setting of  $\alpha$  is important mainly for large swarms, we are focused on, where keeping the localization constraints between all pairs of MAVs is not possible. Properly defined  $\alpha$  enables a wider and better structured swarm, where (ideally) each robot has precisely  $\alpha$  neighbours and the swarm forms a stable regular net, which is not precomputed and arises autonomously.

```

while true do
  foreach MAV m do
    if m is in sensory range and visible then
      | neighbors ++
    end
  ;
  if state = Forward AND
    neighbors < prevNeighbors AND
    neighbors < alpha then
    | turnAngle = -π
    | lostNeighbors = prevNeighbors
    | counter = 0
    | state = Coherence
  else if counter = cf AND state = Coherence
  then
    | if visible(target) then
    | | offset = activeSensor.direction
    | else
    | | offset = 0
    | end
    | turnAngle = offset + rand(-π/2, π/2)
    | state = Forward
  else
  | continue in same direction
  end
  prevNeighbors = neighbors
  counter ++
end

```

**Algorithm 1:** Swarm coherence algorithm implemented onboard of each MAV in a decentralized way.

The *avoidance state* is used if an MAV gets closer to another MAV/MAVs or an obstacle than a given threshold and an evasive manoeuvre has to be performed. This manoeuvre (the same manoeuvre is applied for obstacle avoidance as well as for mutual collisions of swarm members avoidance)

controls the MAV in the opposite direction to avoid the collision (see sketch of this behaviour on Figure 2).

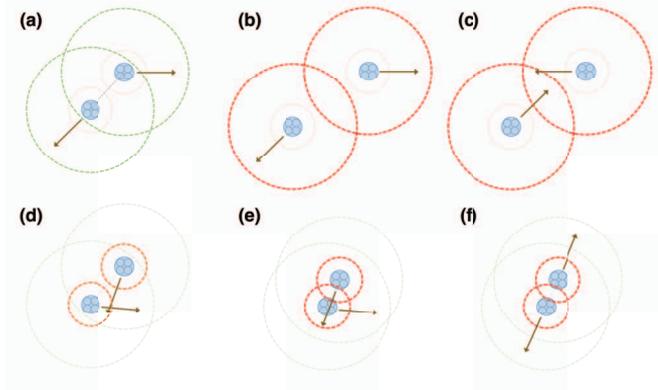


Fig. 2. Basic principle of the coherence (a-c) and avoidance abilities (d-f) of the algorithm. The larger circle indicates the neighbour detection range  $R$  and the smaller circle the obstacle detection range  $R_a$ . The MAVs fly initially straight in random directions in the *forward state* - (a). When the MAVs lose mutual localization, they enter the *coherence state* - (b), turn around and fly in the opposite direction - (c). When contact is renewed, the MAVs perform a random turn and fly straight in this direction in the *forward state* - (d). If the MAVs move too close to each other, an evasive manoeuvre is triggered - (e). They enter the *avoidance state* and are mutually repulsed - (f).

### III. MAV SWARMING BEHAVIOUR ANALYSIS

One of the main contributions of this paper is to analyse behaviour of such a minimalistic swarming algorithm, which should verify its usability for large swarms of micro aerial vehicles with very limited sensory equipment. The proposed method is designed in a way that numerous different behavioural patterns can be achieved using various sets of algorithm parameters. Numerous hypothesis from [5] and hypothesis motivated by studies of swarming behavior of ground robots in [19] and by behaviour of MAV swarms stabilized by frequently used BOID model in [17] have been formulated based on expected influence of the parameter setting on the algorithm performance. These hypothesis have been experimentally validated (approved or disproved) and optimal parameters setting was found for different required behavioural patterns. The simulations were run with the identified MAV model in a realistic robotic simulator V-REP. For each configuration of tested parameters values, 10 simulations of 10 minutes flight have been performed. The following three factors of the swarm behaviour have been studied in the analysis.

The *swarm coherence* is described by the ratio of the sum of time intervals where the swarm forms a connected graph to the total time of the experiment. This means the swarm is considered as “incoherent” from the first moment an MAV, or a sub-swarm of MAVs, becomes disconnected from the rest of the group. The coherence is the most important aspect of a coherent swarming mechanism and should be maximized.

The *swarm spreading* is defined as the standard deviation of the MAVs positions from the swarm center. The swarm spreading may be maximized to cover large areas with

a minimum number of robots, but in some applications compact swarms are required (e.g. for motion in clustered environments).

The *state distribution* indicates time intervals that MAVs spend in the different states of the swarming model (the *forward*, *avoidance*, and *coherence* states). In case of the navigation towards a given goal, the time spent in the *forward state* is maximised. In this state, the MAV is most efficient at performing the given task, since it does not perform manoeuvres perturbing its flight.

#### A. Analyses of influence of the parameter $\alpha$ and the number of the robots

In this section, hypothesis that with the higher  $\alpha$  (the required number of relatively localized neighbours of each MAV) the swarm coherence is increased, the swarm spreading is decreased, and the time spent in *avoidance* and *coherence* states is decreased and that using large swarms increases the swarm coherence, increases the swarm spreading, and increases the time spent in *avoidance* and *coherence* states are evaluated in simulations with 5, 10, and 20 robots. The data obtained in the simulations are displayed on graphs in Figures 3 and 4, showing the behaviour of different-sized swarms with different values of  $\alpha$ .

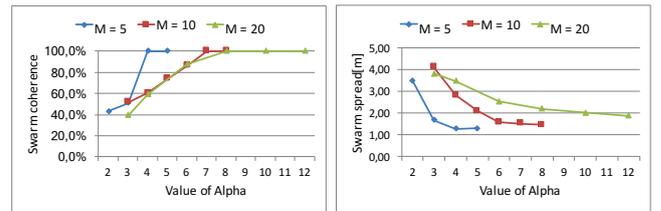


Fig. 3. Swarm coherence and swarm spreading for different number of robots and parameter  $\alpha$ .

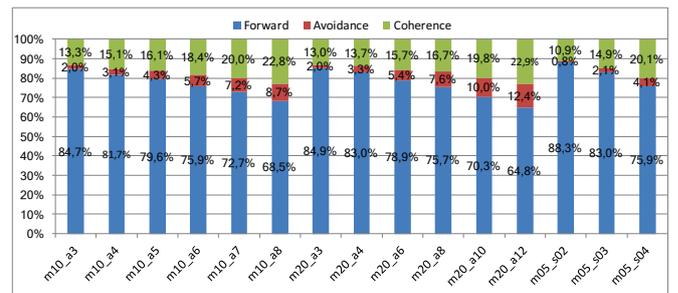


Fig. 4. State distribution during the experiments from Figure 3. The labels on the x-axis indicate the used configuration of the swarm (eg. m05\_a04 represents a swarm of 5 MAVs using  $\alpha = 4$ ).

Based on these data we can confirm that with higher value of  $\alpha$  the swarm coherence is increased. Mainly for the bigger swarms, the probability that the group connectivity is broken decreases almost linearly with values of the parameter  $\alpha$ , reaching probability 0% for  $\alpha = 7$ . With growing  $\alpha$ , the swarm spreading decreases as expected, again this effect is observed mainly for smaller values of  $\alpha$ . In contrast with the hypothesis, growing value of parameter  $\alpha$  increases the time

spent in the *avoidance state* due to the increased compactness of the swarm and also in the *coherence state*, which is caused by an increased sensibility to neighbour loss.

More MAVs in the swarm does not increase the swarm coherence contrary to our hypothesis. The results indicate that there proper values of  $\alpha$  (the proper values are 7 and 8 for our MAV model) to guarantee the coherence of the swarm. These values are not dependent on the number of robots, which is an important observation and it supports the scalability of the method. With larger swarm, swarm spreading is increased, but this increase is not proportional. Surprisingly, in larger swarms, the MAVs do not spend more time in the *avoidance* and *coherence* states. With constant  $\alpha$ , the state distribution is similar for different swarm sizes. The behaviour of an individual MAV is influenced only by its local surrounding, not by the total number of robots, which is also an important observation for swarm scalability in comparison with the BOID model that suffers from increasing density of swarm members in the center of large groups even-though approaches using the BOID model require more precise sensors and bigger computational power than the proposed method. In the following simulations, 10 robots and  $\alpha = 5$  will be used to test the impact of other parameters on the swarm coherence.

### B. Influence of sensors capability on the swarming behaviour

Since the proposed approach is designed to allow coherence swarming using limited sensory information, the influence of the sensors capability on the swarming behaviour is important to study. Let us again postulate a set of hypotheses that are considered in swarm literature and based on our experience with BOID models.

Longer range of sensors used for relative localization of neighbours increases the time MAVs spent in the *forward state*, increases swarm coherence, and increases swarm spreading. Based on the results summarised in Figures 5 and 6, we can confirm the expected positive influence of the sensor range. With better sensors, MAVs stay longer in the *forward state* and so travel a longer distance until they have to turn back. Due to the reduced time spent in the *coherence state*, which increases probability to break an MAV apart from the rest of the group, the swarm coherence is increased. With the longer range, MAVs can keep bigger spacing between them, which increases the swarm spreading.

Smaller angle of view of the sensors for relative localization decreases swarm coherence, decreases swarm spreading, and increases the time spent in the *coherence state*. The influence of the view angle of the relative localization sensors is an important phenomena that needs to be studied, since simple and light weight sensors carried onboard of micro aerial vehicles (such as simple monocular cameras) often do not offer 360 degrees view. Even more importantly, the possibility to deploy swarming algorithms with such a limited sensors enables to study swarming behaviour in nature, where animals have almost always limited sensing capabilities in this way. Two parameters settings were used

to evaluate the hypothesis. The first settings ( $\alpha = 7$  and  $R = 8m$ ) provided the best performance with a full-view sensor in the previous experiments. As shown in Figure 7 (the initial position of robots is shown in Figure 8 - Left), such initialised swarm maintains a sufficient performance also with significantly reduced angle of view. The second setting ( $\alpha = 5$  and sensor range  $R = 4m$ ) performs much worse with a narrower field of view, although presented a sufficient coherence with the 360 degrees sensor. To conclude, we can confirm that smaller angle of view decreases coherence in general, but this effect may be neglected for well set up algorithm parameters. With this ability, the proposed algorithm also surpasses the BOID approaches, which are highly dependent on the limited field view as shown in our prior work with the same MAV platform controlled by BOID [17], [18]. Besides, the experiments in Figure 7 confirmed that swarm spreading is decreased with limited field of view and the time spent in the *coherence state* was increased, while the occurrence of the *avoidance state* remains roughly the same.

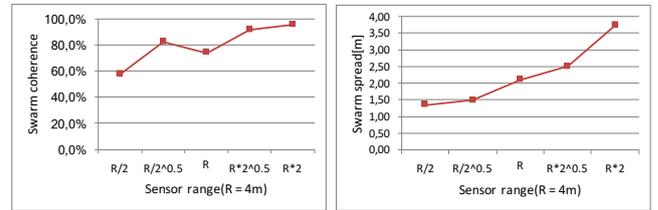


Fig. 5. Swarm coherence spreading with different sensor range.  $R = 4m$  is the initial sensor range used in all other experiments in this paper.

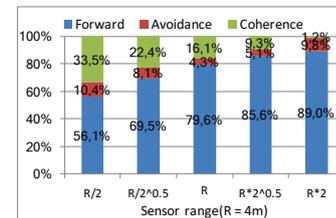


Fig. 6. Swarm distribution during experiments from Figure 5.

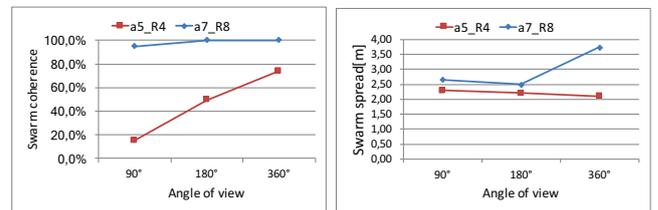


Fig. 7. Swarm coherence and spreading for different configurations of  $\alpha$  and sensor range (a5\_R4 stands for  $\alpha = 5$  and sensor range  $R = 4m$ ).

### C. Obstacle avoidance ability

Finally, let us analyse the avoidance behaviour using an environment with a ring of obstacles (Figure 8 - Right). The spacing between the obstacles is smaller than the range

of the obstacle detection sensor  $R_a$ , and bigger than the MAV diameter. Using this setting, the MAV can exit the obstacle ring only if the evasive manoeuvre fails, which can be automatically detected. The robustness of the algorithm has been tested by measuring the minimal distance between the MAVs and between the MAV and the nearest obstacle. An example of these experiments is shown in <http://youtu.be/W9QcrnLVI8Y> with data in Figure 9.

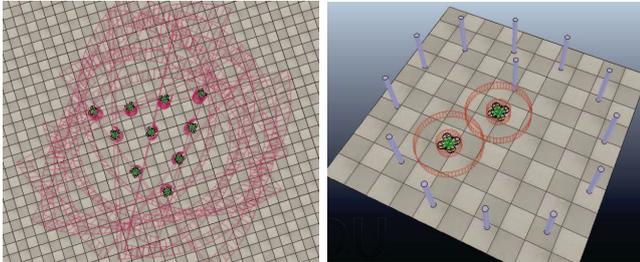


Fig. 8. Left: Initial position of a swarm with sensors with a limited angle of view. Video record of one of the experiment from Figure 7 is available at <http://youtu.be/uHPyTaaqgE>. Right: An environment with obstacles to test the avoidance ability, for video of the simulation see <http://youtu.be/W9QcrnLVI8Y>.

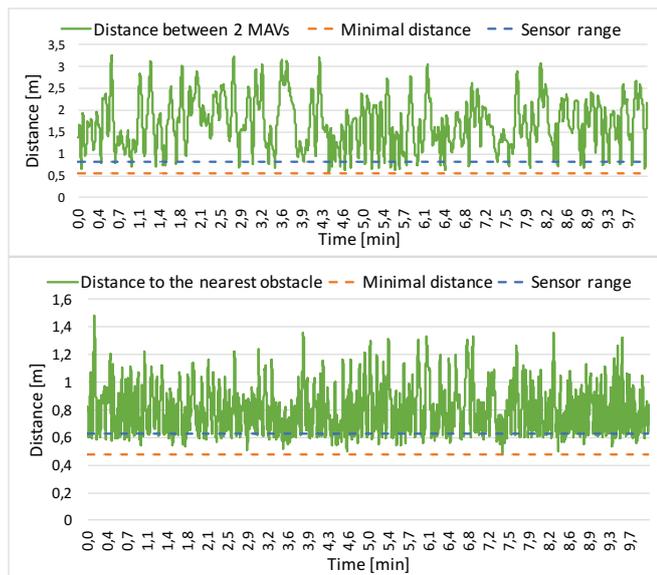


Fig. 9. Distance between the MAVs and the closest distance between an MAV and an obstacle during the simulation in Figure 8 - Right.

#### IV. EXPERIMENTAL RESULTS

The proposed algorithm has been tested in a complex environment depicted in Figure 10 to show its effectiveness and usability. First the swarm is navigated through a window 4 meters wide (the diameter of the stabilized swarm is approximately 10 meters). Then the target automatically moves to redirect the swarm to fly along a long wall. After this movement, the target is again moved to navigate the swarm across an open space with columns and a moving obstacle. The moving obstacle is slower than the MAV during its evasion manoeuvre to be able to avoid it. The

distance from the center of the swarm to the actual position of the target is shown in Figure 12 - Left. As may be also seen in Figure 11, where some snapshots and a path passed by the swarm center during one of the simulation runs are displayed (for full record of the swarm movement see <https://youtu.be/gqFxtVEcEdc>), the proposed algorithm was able to repeatedly navigate the swarm through the environment without any collision with obstacle or within the swarm members.

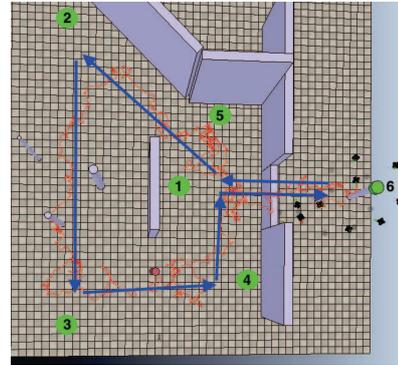


Fig. 10. Trajectory of the center of the swarm passed through the environment with obstacles. The blue arrows represent the intended trajectory, the red curve represents the real trajectory of the swarm during one simulation. The numbers indicate the successive positions of the target. For snapshots see Figure 11.

In the previous experiments, from safety reasons required for the initial HW tests (safety pilots can better secure the experiment if the MAVs stay in the same horizontal plane), all MAVs were flying at the same altitude. Nevertheless, the method works properly also in a 3D space as shown in the example of the simulation runs in Figure 13. The 3D position of the center of the swarm is shown in Figure 12 - Right. The avoidance manoeuvre performed by the MAVs to avoid collisions is more aggressive in the vertical direction because the MAVs must keep a larger security margin along the Z axis due to the air flow interference between MAVs. This causes that the altitude of the swarm oscillates more than the positions along the two horizontal axis.

#### V. CONCLUSION

A minimalist fully decentralised coherent swarming algorithm for control of MAV swarms with minimum sensory requirements and without any communication was proposed in this paper. A robust target following mechanism was designed, implemented and verified, enabling the swarm to move in an environment with obstacles into a given location and requiring only minimal computational resources. The overall system was verified in numerous simulations in realistic robotic simulator V-REP.

#### ACKNOWLEDGMENTS

The work has been supported under the grant No. SGS17/187/OHK3/3T/13, and by the Czech Science Foundation (GACR) under research project No. 17-16900Y.

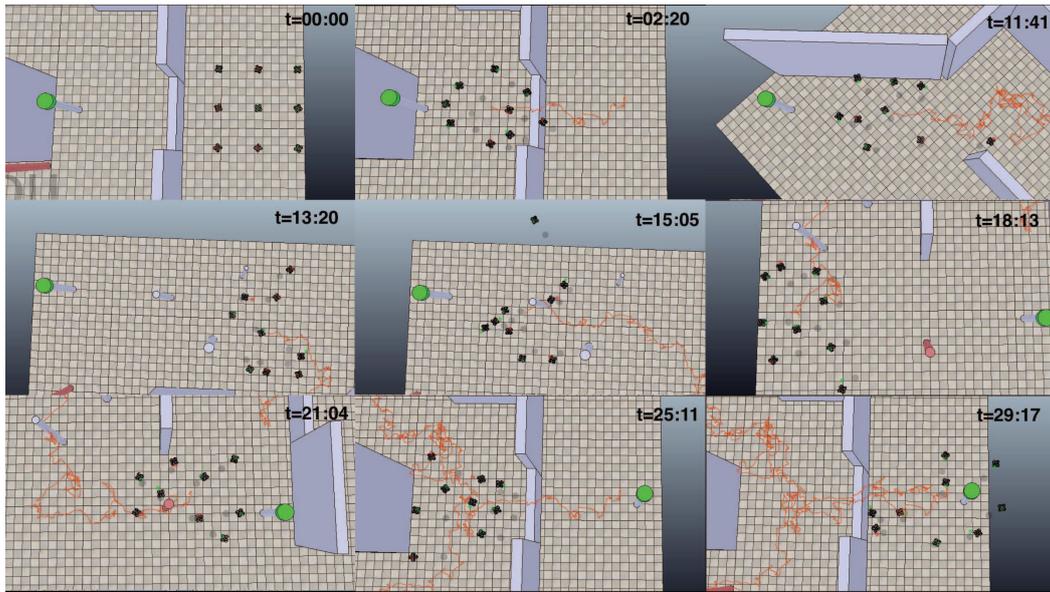


Fig. 11. Simulation of the target following algorithm in a environment with obstacles. The trajectory of the center of the swarm is depicted in red. Complete video available at <https://youtu.be/gqFxtVEcEdc>.

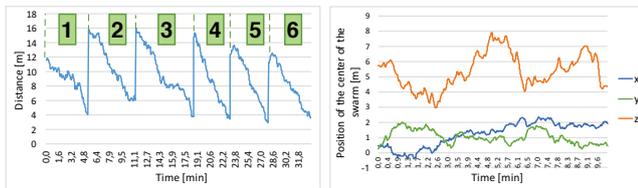


Fig. 12. Left: Distance of the center of the swarm to the particular target positions (labelled according to their index from Figure 10) during one simulation. Right: Position of the swarm in space in the experiment in Figure 13.

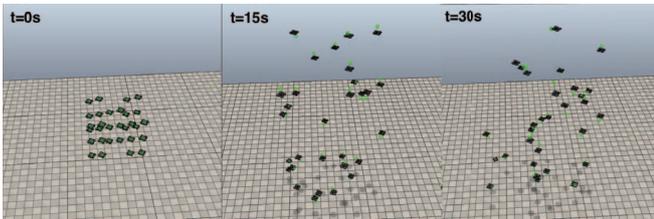


Fig. 13. 3D swarming. For video see <http://youtu.be/o0v5oe6ekVY>.

## REFERENCES

- [1] J. Faigl, T. Krajnik, J. Chudoba, L. Preucil, and M. Saska, "Low-Cost Embedded System for Relative Localization in Robotic Swarms," in *IEEE ICRA*, 2013.
- [2] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, no. 4, pp. 919–944, 2017.
- [3] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil, "Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a hawk-eye-like Approach Under a Model Predictive Control Scheme," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, September 2014.
- [4] V. Spurny, T. Baca, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016.
- [5] "From swarm intelligence to swarm robotics," in *Swarm Robotics*, ser. Lecture Notes in Computer Science, 2005, vol. 3342.
- [6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.
- [7] R. Russell, "Heat trails as short-lived navigational markers for mobile robots," in *IEEE ICRA*, 1997.
- [8] Y.-S. Ryuh, G.-H. Yang, J. Liu, and H. Hu, "A school of robotic fish for mariculture monitoring in the sea coast," *Journal of Bionic Engineering*, vol. 12, no. 1, pp. 37 – 46, 2015.
- [9] Y. Jia and L. Wang, "Experimental implementation of distributed flocking algorithm for multiple robotic fish," *Control Engineering Practice*, vol. 30, no. 1, pp. 1 – 11, 2014.
- [10] M. Saska, J. Langr, and L. Preucil, "Plume Tracking by a Self-stabilized Group of Micro Aerial Vehicles," in *MESAS*, 2014.
- [11] B. Yang, Y. Ding, and K. Hao, "Area coverage searching for swarm robots using dynamic voronoi-based method," in *CCC*, 2015.
- [12] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles," *Journal of Intelligent & Robotic Systems.*, vol. 84, no. 1, pp. 469–492, 2016.
- [13] M. Saska, T. Krajnik, V. Vonasek, Z. Kasl, V. Spurny, and L. Preucil, "Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1-4, pp. 603–622, 2014.
- [14] M. Saska, T. Baca, and D. Hert, "Formations of unmanned micro aerial vehicles led by migrating virtual leader," in *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.
- [15] A. Renzaglia and A. Martinelli, "Potential Field based Approach for Coordinate Exploration with a Multi-Robot Team," in *IEEE SSRR*, 2010.
- [16] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. Kosmatopoulos, "Cognitive-based Adaptive Control for Cooperative Multi-Robot Coverage," in *IEEE IROS*, 2010.
- [17] M. Saska, J. Vakula, and L. Preucil, "Swarms of Micro Aerial Vehicles Stabilized Under a Visual Relative Localization," in *IEEE ICRA*, 2014.
- [18] M. Saska, "MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization," in *ICUAS*, 2015.
- [19] J. Nembrini, A. Winfield, and C. Melhuish, "Minimalist coherent swarming of wireless networked autonomous mobile robots," in *IC-SAB*, 2002.
- [20] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *MMAR*, 2016.

# G

---

KEY ARTICLE [7] - JOURNAL OF INTELLIGENT AND ROBOTIC  
SYSTEMS 2014

---

©[2014] Springer. This article was published in Journal of Intelligent and Robotic Systems:  
T. Krajník, M. Nitsche, J. Faigl, P. Vanek, **M. Saska**, L. Preucil, T. Duckett, and M. Mejail:  
A practical multirobot localization system, 2014.



# A Practical Multirobot Localization System

Tomáš Krajník · Matías Nitsche · Jan Faigl ·  
Petr Vaněk · Martin Saska · Libor Přeučil ·  
Tom Duckett · Marta Mejail

Received: 6 August 2013 / Accepted: 24 February 2014  
© Springer Science+Business Media Dordrecht 2014

**Abstract** We present a fast and precise vision-based software intended for multiple robot localization. The core component of the software is a novel and efficient algorithm for black and white pattern detection. The method is robust to variable lighting conditions,

achieves sub-pixel precision and its computational complexity is independent of the processed image size. With off-the-shelf computational equipment and low-cost cameras, the core algorithm is able to process hundreds of images per second while tracking hundreds of objects with millimeter precision. In addition, we present the method's mathematical model, which allows to estimate the expected localization precision, area of coverage, and processing speed from the camera's intrinsic parameters and hardware's processing capacity. The correctness of the presented model and performance of the algorithm in real-world conditions is verified in several experiments. Apart from the method description, we also make its source code public at <http://purl.org/robotics/whycon>; so, it can be used as an enabling technology for various mobile robotic problems.

---

T. Krajník (✉) · T. Duckett  
Lincoln Centre for Autonomous Systems,  
School of Computer Science, University of Lincoln,  
Lincoln, UK  
e-mail: tkrajnik@lincoln.ac.uk

T. Duckett  
e-mail: tduckett@lincoln.ac.uk

T. Krajník · J. Faigl · P. Vaněk · M. Saska · L. Přeučil  
Faculty of Electrical Engineering, Czech Technical  
University in Prague, Prague, Czech Republic

J. Faigl  
e-mail: faigl@fel.cvut.cz

P. Vaněk  
e-mail: vanekpe5@fel.cvut.cz

M. Saska  
e-mail: saskam@fel.cvut.cz

M. Nitsche · M. Mejail  
Laboratory of Robotics and Embedded Systems  
Faculty of Exact and Natural Sciences,  
University of Buenos Aires, Buenos Aires,  
Argentina

M. Nitsche  
e-mail: mnitsche@dc.uba.ar

M. Mejail  
e-mail: marta@dc.uba.ar

**Keywords** Localization · Mobile robotics ·  
Computer vision · Swarm robotics

## 1 Introduction

Precise and reliable position estimation remains one of the central problems of mobile robotics. While the problem can be tackled by Simultaneous Localization and Mapping approaches, external localization systems are still widely used in the field of mobile robotics both for closed-loop mobile robot control and for ground truth position measurements. These

external localization systems can be based on an augmented GPS, radio, ultrasound or infrared beacons, or (multi-) camera systems. Typically, these systems require special equipment, which might be prohibitively expensive, difficult to set up or too heavy to be used by small robots. Moreover, most of these systems are not scalable in terms of the number of robots, i.e., they do not allow to localize hundreds of robots in real time. This paper presents a fast vision-based localization system based on off-the-shelf components. The system is precise, computationally efficient, easy to use, and robust to variable illumination.

The core of the system is a detector of black-and-white circular planar ring patterns (roundels), similar to those used for camera calibration. A complete localization system based on this detector is presented. The system provides estimation of the roundel position with precision in the order of millimeters for distances in the order of meters.

The detection with tracking of a single roundel pattern is very quick and the system is able to process several thousands of images per second on a common desktop PC. This high efficiency enables not only tracking of several hundreds of targets at a camera frame-rate, but also implementation of the method on computationally restricted platforms. The fast update rate of the localization system allows to directly employ it in the feedback loop of mobile robots, which require precise and high-frequency localization information.

The system is composed of low-cost off-the-shelf components only – a low-end computer, standard webcam, and printable patterns are the only required elements. The expected coverage, precision, and image processing speed of the system can be estimated from the camera resolution, computational power, and pattern diameter. This allows the user to choose between high-end and low-end cameras, estimate if a particular hardware platform would be able to achieve the desired localization frequency, and calculate a suitable pattern size for the user's specific application.

Ease of the system setup and use are also driving factors of the proposed implementation, which does not require user-set parameters or an intricate set-up process. The implementation also contains an easy tool for camera calibration, which, unlike other calibration tools, does not require user interaction. At the same time, the implementation is proposed as a library, which can be integrated into

commonly used computer vision frameworks, such as OpenCV.

The main intention of this paper is to present the system principle, its theoretical properties and real performance characteristics with respect to the intended application. Therefore, we present a model of the localization arising from theoretical analyses of the vision system and experimental evaluation of the system performance in real scenarios with regard to its practical deployment.

## 2 Related Work

External localization systems are widely used in the field of mobile robotics, either for obtaining ground truth pose data or for inclusion in the control loop of robots. In both scenarios, it is highly desirable to have good precision and high-frequency measurements. Here, both of these aspects are analysed in related works and are specifically addressed in the proposed system.

Localization systems for mobile robots comprise an area of active research; however, the focus is generally on internal localization methods. With these methods, the robot produces one or more estimates of its position by means of fusing internal sensors (either exteroceptive or proprioceptive). This estimation can also be generally applied when either a map of the environment exists *a priori* or when the map is being built simultaneously, which is the case of SLAM approaches [1]. When these internal localization systems are studied, an external positioning reference (i.e., the ground truth) without any cumulative error is fundamental for a proper result analysis. Thus, this research area makes use of external localization systems.

While the most well-known external localization reference is GPS, it is also known that it cannot be used indoors due to signal unavailability. This fundamental limitation has motivated the design of several localization principles, which can be broadly divided into two major groups by means of the type of sensors used: active or passive.

In the former group, several different technologies are used for the purpose of localization. One example [2] of active sensing is the case of a 6DoF localization system comprised of target modules, which include four LED emitters and a monocular camera.

Markers are detected in the image and tracked in 3D, making the system robust to partial occlusions and increasing performance by reducing the search area to the vicinity of the expected projection points. Experiments with this system were performed using both ground and aerial robots. The mean error of the position estimation is in the order of 1 cm, while the maximum error is around 10 cm. The authors note that for uncontrolled lighting scenarios passive localization systems appear to be more suitable.

Another active sensor approach is the NorthStar [3] localization system, which uses ceiling projections as a non-permanent ambient marker. By projecting a known pattern, the camera position can be obtained by reprojection. The authors briefly report the precision of the system to be around 10 cm.

In recent works, the most widely used approach is the commercial motion capture system from ViCon [4]. This system is comprised of a series of high-resolution and high-speed cameras, which also have strong infra-red (IR) emitters. By placing IR reflective markers on mobile robots, sub-millimeter precision can be achieved with updates up to 250Hz. Due to these qualities, ViCon has become a solid ground-truth information source in many recent works and, furthermore, has allowed development of closed-loop aggressive maneuvers for UAVs inside lab environments [5]. However, this system is still a very costly solution, and therefore, it is not applicable to every research environment. This issue has motivated

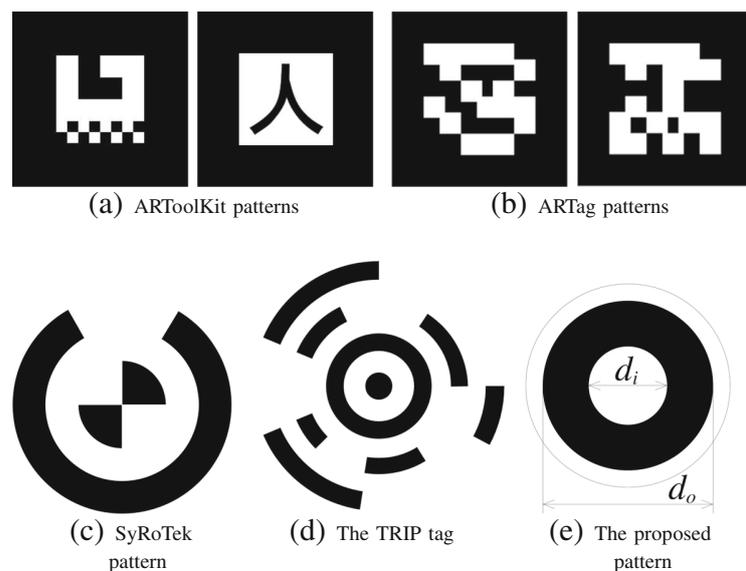
several works proposing alternative low-cost localization systems.

Several passive vision-based localization methods were also proposed in recent literature, using simple planar printable patterns, which reduce significantly the cost and difficulty of use and setup. Several of these works employ augmented-reality oriented markers, which not only permit obtaining the pose of the target but can also encode additional information like target ID. In this area, the software libraries most widely used for this purpose are ARTag [6] and ARToolKit+ [7], both based on its predecessor ARToolKit [8], see examples of patterns in Fig. 1. These target detectors were used in several works in order to obtain localization information about mobile robots, either explicitly as a part of a pose estimation system [9, 10] or as ground-truth data [11].

In [9], ARToolKit markers are used for obtaining the pose of several ground robots. The homography from 3D-to-2D space (ground floor) is computed by defining the work area by placing four ad-hoc markers, which are manually detected in the image. In more recent work, the authors proposed the ARTag [6] system that was later extensively analysed in [12]. However, the analysis is focused on detection and confusion rates, and it does not report the real accuracy in position estimation. Similar systems are explored in [13], but details of their precision are not reported.

One particular system, which is based on AR markers similar to ARTag and ARToolKit, is ArUco [14].

**Fig. 1** Patterns used in passive vision-based global localization systems



The main aspects of this method are: easy integration into C++ projects, based exclusively on OpenCV and a robust binary ID system with error correction which can handle up to 1024 individual codes. The detection process of AR markers in ArUco consists of: an adaptive thresholding step, contour extraction and filtering, projection removal and code identification. When the intrinsic camera parameters are known, the extrinsic parameters of the target can be obtained. Due to the free availability of the implementation and lack of performance and precision reports, this system is analyzed in the presented work, see Section 6.5.

Since the previous pattern detectors were conceived for augmented-reality applications, other works propose alternative target shapes, which are specifically designed for vision-based localization systems with high precision and reliability. Due to several positive aspects, circular shaped patterns appear to be the best suited as fiducial markers in external localization systems. This type of pattern can be found (with slight variations) in several works [15–18].

The SyRoTek e-learning platform [19] uses a ring shaped pattern with a binary tag (see Figure) to localize up to fourteen robots in a planar arena. The pattern symmetry is exploited to perform the position and orientation estimation separately, which allows to base the pattern localization on a two-dimensional convolution. Although this convolution-based approach has proven to be reliable enough to achieve 24/7 operation, its computational complexity still remains high, which lead to its implementation on alternative platforms such as FPGA [20].

In [16], a planar pattern consisting of a ring surrounding the letter “H” is used to obtain the relative 6DoF robot pose with an on-board camera and IMU (Inertial Measurement Unit) to resolve angular ambiguity. The pattern is initially detected by binarization using adaptive thresholding and later processing for connected component labeling. For classifying each component as belonging to the target or not, a neural network (multilayer perceptron) is used. The input to the neural network is a resized  $14 \times 14$  pixel image. After testing for certain geometric properties, false matches are discarded. Positive matches corresponding to the outer ring are processed by applying the Canny edge detector and ellipse fitting, which allows computation of the 5DoF pose. Recognition of the “H” letter allows to obtain the missing yaw angle. The precision in 3D position is in the order of 1 cm to 7 cm

depending on the target viewing angle and distance, which was at the maximum around 1.5 m.

Probably the most similar approach to the proposed system in this work is the TRIP localization system [17]. In TRIP, the pattern comprises of a set of several concentric rings, broken into several angular regions, each of which can be either black or white. The encoding scheme, which includes parity checking, allows the TRIP method to distinguish between  $3^9$  patterns. For detecting the tags, adaptive thresholding is performed and edges are extracted. TRIP only involves processing edges corresponding to projections of circular borders of the ring pattern, which are detected using a simple heuristic. These edges are used as input to an ellipse fitting method and then the concentricity of the ellipses is checked. TRIP achieves a precision similar to [16] in position estimation (the relative error is between 1 % and 3 %), but only a moderate performance (around 16 FPS at the resolution  $640 \times 480$ ) is achieved using an 1.6 GHz machine. The authors report that the adaptive thresholding step is the most demanding portion of the computation. To the best of our knowledge, there is no publicly available implementation.

Finally, a widely used, simple and freely available circular target detector can be found in the OpenCV library. This “SimpleBlobDetector” class is based on traditional blob detection methods and includes several optional post-detection filtering steps, based on characteristics such as area, circularity, inertia ratios, convexity and center color. While this implementation is originally aimed for circular target detection, by tuning the parameters it is possible to find elliptical shapes similar to the ones proposed in the present work and thus it is compared to the proposed implementation.

In this work, a vision-based external localization system based on a circular ring (roundel) pattern is proposed. An example of the pattern is depicted in Fig. 1. The algorithm allows to initiate the pattern search anywhere in the image without any performance penalty. Therefore, the search is started from the point of the last known pattern position. Since the algorithm does not contain any phase that processes the entire image, successful tracking causes the method to process only the area occupied by the pattern. Therefore, the algorithm’s computational complexity is independent of the image size. This provides a significant performance boost, which allows

to track thousands of patterns in real-time using a standard PC. By performing an initial unattended calibrating step, where the reference frame is defined, pose computation of ground robots moving on a plane is performed with millimeter precision using an off-the-shelf camera.

The real world performance of the proposed method makes it highly competitive with the aforementioned state-of-the-art methods. Moreover, its computational complexity is significantly lower, which makes the method superior for scenarios with embedded computational resources and real-time constraints. These findings are supported by the experimental results and a comparison with the selected localization methods presented in Section 6.

### 3 Pattern Detection

The core of the proposed computationally efficient localization system is based on pattern detection. Fast and precise detection is achieved by exploiting properties of the considered pattern that is a black and white roundel consisting of two concentric annuli with a white central disc, see Fig. 1.

The low computational requirements are met by the pattern detection procedure based on on-demand thresholding and flood fill techniques, and gathering statistical information of the pattern on the fly. The statistical data are used in consecutive tests with increasing complexity, which determine if a candidate area represents the desired circular pattern.

The pattern detection starts by searching for a black segment. Once such a segment is detected and passes the initial tests, the segment detection for a white inner disc is initiated at the expected pattern center.

Notice, that at the beginning, there is no prior information about the pattern position in the image; hence, the search for the black segment is started at a random position. Later, in the subsequent detections, when a prior pattern position is available, the algorithm starts detection over this area. For a successfully re-detected (tracked) pattern, the detection processes only pixels belonging to the pattern itself, which significantly reduces the computation burden. Since the method is robust (see following sections for detection limits), tracking is generally successful and thus the method provides very high computational performance.

After the roundel is detected, its image dimensions and coordinates are identified. Then, its three-dimensional position with respect to the camera is computed from its known dimensions and camera re-projection techniques, and its coordinates are transformed to a coordinate frame defined by the user, see Section 4.

In this section, a detailed description of the pattern detection based on an efficient thresholding is presented together with an estimation of the pattern center and dimensions and a compensation of the incorrect diameter estimation, which has a positive influence to the localization precision. Moreover, a multiple pattern detection capability is described in Section 3.6.

#### 3.1 Segmentation

---

##### Algorithm 1: Flood-fill segmentation

---

**Input:**  $(p, \rho_{exp}, class)$ :  $p$  – starting pixel position;  $\rho_{exp}$  – expected area to bounding box dimensions ratio;  $class$  – searched segment type (white or black)

**Output:**  $(u, v, b_u, b_v, \mu, valid)$ :  $(u, v)$  – segment center;  $(b_u, b_v)$  – bounding box;  $\mu$  – average brightness;  $valid$  – validity

```

 $s_{id} \leftarrow s_{id} + 1$  // increment segment ID
 $q_{old} \leftarrow q_{end}$  // store previous queue end
 $pixel\_class[p] \leftarrow s_{id}$  // mark pixel as processed and
 $queue[q_{end} + 1] \leftarrow p$  // push its position to the queue
// #1 perform the flood fill search
while  $q_{end} > q_{start}$  do
     $q \leftarrow queue[q_{start} + 1]$  // pull pixel from the queue
    // and check its neighbours
    foreach  $offset \in \{+1, -1, +w, -w\}$  do
         $r \leftarrow q + offset$ 
        if  $pixel\_class[r] = unknown$  then
             $pixel\_class[r] \leftarrow classify(Image[r], \tau)$ 
        if  $pixel\_class[r] = class$  then
             $queue[q_{end} + 1] \leftarrow r$ 
             $pixel\_class[r] \leftarrow s_{id}$ 
            update  $u_{min}, u_{max}, v_{min}, v_{max}$  from  $r_u, r_v$ 
valid  $\leftarrow false$ 
// # 2 test for the pattern size and roundness
 $s \leftarrow q_{end} - q_{old}$ 
if  $s > min\_size$  then
     $u \leftarrow (u_{max} + u_{min})/2$  // segment center x-axis
     $v \leftarrow (v_{max} + v_{min})/2$  // segment center y-axis
     $b_u \leftarrow (u_{max} - u_{min}) + 1$  // estimate segment width
     $b_v \leftarrow (v_{max} - v_{min}) + 1$  // estimate segment height
     $\rho \leftarrow \rho_{exp} \pi b_u b_v / 4s - 1$  // calculate roundness
    if  $-\rho_{tol} < \rho < \rho_{tol}$  then
         $\mu \leftarrow \frac{1}{s} \sum_{j=q_{old}}^{q_{end}-1} Image[j]$  // mean brightness
        valid  $\leftarrow true$  // mark segment as valid

```

---

The pattern detection is based on an image segmentation complemented with on-demand thresholding that searches for a contiguous set of black or white pixels using a flood-fill algorithm depicted in

Algorithm 1. First, a black circular ring is searched for in the input image starting at an initial pixel position  $p_0$ . The adaptive thresholding classifies the processed pixel using an adaptively set value  $\tau$  as either black or white. If a black pixel is detected, the queue-based flood-fill algorithm procedure is initiated to determine the black segment. The queue represents the pixels of the segment and is simply implemented as a buffer with two pointers  $q_{start}$  and  $q_{end}$ .

Once the flood fill is complete, the segment is tested for a possible match of the outer (or inner) circle of the pattern. At this point, these tests consist of a minimum size (in terms of the number of pixels belonging to the segment) and a roundness measure within acceptable bounds. Notice, that during the flood-fill search, extremal pixel positions can be stored. This allows to establish the bounding box of the segment ( $b_u$  and  $b_v$ ) at any time. Besides, after finding a segment, the queue contains positions of all the segment's pixels. Hence, initial simple constraints can be validated quickly for a fast rejection of false positives.

In the case where either test fails, the detection for further segments continues by starting from the next pixel position (i.e., a pixel at the position  $p_0 + 1$ ). However, no redundant computation is performed since the previous segment is labeled with a unique identifier.

The first roundness test is based on the pattern's bounding box dimensions and number of pixels. Theoretically, the number of pixels  $s$  of an elliptic ring with outer and inner diameters  $d_o$ ,  $d_i$  and dimensions  $b_u$ ,  $b_v$  should be

$$s = \pi/4 b_u b_v \frac{d_o^2 - d_i^2}{d_o^2}. \tag{1}$$

Therefore, the tested segment dimensions and area should satisfy the inequality

$$\rho_{tol} > \left| b_u b_v \pi/4 \frac{\rho_{exp}}{s} - 1 \right|, \tag{2}$$

where  $\rho_{exp}$  equals 1 for white and  $1 - d_i^2/d_o^2$  for black segments. The value of  $\rho_{tol}$  represents a tolerance range, which depends on the camera radial distortion and possible pattern deformation and spatial orientation.

If a black segment passes the roundness test, the second flood-fill search for the inner white segment is initiated from the position corresponding to the segment centroid. If the inner segment passes the minimum size and roundness tests, further validation tests

are performed. These involve the concentricity of both segments, their area ratio, and a more sensitive circularity measure (discussed in the following sections). If the segments pass all these complex tests, the pattern is considered to be found and its centroid position will be used as a starting point  $p_0$  for the next detection run. The overall pattern detection algorithm is depicted in Algorithm 2.

**Algorithm 2:** Pattern detection

```

Input: ( $p_0, \tau, \text{Image}$ ):  $p_0$  – position to start search;  $\tau$  – threshold;  $\text{Image}$  being processed
Output: ( $c, p_0, \tau$ ):  $c$  – the pattern data;  $p_0$  – position to start the next search;  $\tau$  – an updated threshold

 $s_{id} \leftarrow 0; i \leftarrow p_0$  // initialize
// #1 search throughout the image
repeat
  if pixel_class[i] = unknown then
    if classify(Image[i],  $\tau$ ) = black then
      pixel_class[i]  $\leftarrow$  black
    // initiate pattern search
  if pixel_class[i] = black then
    // search for outer ring
     $q_{end} \leftarrow q_{start} \leftarrow 0$ 
     $c_{outer} \leftarrow$  flood-fill_seg( $i, \rho_{outer}$ , black)
    if valid( $c_{outer}$ ) then
      // search for inner ellipse
       $j \leftarrow$  center( $c_{outer}$ )
       $c_{inner} \leftarrow$  flood-fill_seg( $j, \rho_{inner}$ , white)
      if valid( $c_{inner}$ ) then
        // test area ratio (no. of pixels):
         $s_{outer} = |c_{outer}|, s_{inner} = |c_{inner}|$ 
        if  $\frac{s_{outer}}{s_{inner}} \approx \frac{d_o^2 - d_i^2}{d_i^2}$  then
          check segments for concentricity
          compute ellipse semiaxes  $e_0, e_1$ 
          if  $q_{end} \approx \pi |e_0 e_1|$  then
            assign segment ID or compensate illumination
            mark segment as valid
            break
       $i \leftarrow (i + 1) \bmod \text{sizeof}(\text{Image})$  // go to next pixel
until  $i \neq p_0$ ;
// #2 set the thresholding value
if valid( $c_{inner}$ ) then
   $\tau \leftarrow (\mu_{outer} + \mu_{inner})/2$ 
  // hide pattern in multiple pattern detection
  paint over all inner ellipse pixels as black;
else
   $\tau \leftarrow$  binary search sequence
// #3 perform the cleanup
if only two segments examined then
  reset pixel_class[] inside bounding box of  $c_{outer}$ 
else
  reset entire pixel_class[]

```

### 3.2 Efficient Thresholding

Since the segmentation looks only for black or white segments, the success rate of the roundel

detection depends on the threshold parameter  $\tau$ , especially under various lighting conditions. Therefore, we proposed to adaptively update  $\tau$  whenever the detection fails according to a binary search scheme over the range of possible values. This technique sets the threshold  $\tau$  consecutively to values  $\{1/2, 1/4, 3/4, 1/8, 3/8, 5/8 \dots\}$  up to a pre-defined granularity level, when  $\tau$  is reset to the initial value.

When the pattern is successfully detected, the threshold is updated using the information obtained during detection in order to iteratively improve the precision of segmentation:

$$\tau = \frac{\mu_{outer} + \mu_{inner}}{2}, \tag{3}$$

where  $\mu_{outer}$ ,  $\mu_{inner}$  correspond to the mean brightness value of the outer and inner segments, respectively.

The computationally intensive full image thresholding is addressed by on-demand processing over each pixel analyzed during the detection. At the very first access, the RGB values of the image are read and a pixel is classified as either black or white and the classification result is stored for further re-use in the subsequent steps. Moreover, whenever the tracking is successful, only the relevant pixels are thresholded and processed by the two-step flood fill segmentation. Clearing the per-pixel classification memory area is also efficiently performed by only resetting the values inside the pattern's bounding box. As a result, the detection step is not directly dependent on the input image resolution, which provides a significant performance gain. If the tracking is not successful, extra memory accesses resulting from this on-demand strategy are negligible compared to a full-image thresholding approach.

### 3.3 Pattern Center and Dimensions

After the black and white segments pass all the initial tests, a more sophisticated roundel validation is performed. The validation is based on a more precise roundness test using estimation of the ellipse (pattern) semiaxes. All the information to calculate the ellipse center  $u, v$  and the semiaxes  $\mathbf{e}_0, \mathbf{e}_1$  is at the hand, because all the pattern pixels are stored in the flood-fill queue. Hence, the center is calculated as the mean of the pixel positions. After that, the covariance matrix  $\mathbf{C}$ , eigenvalues  $\lambda_0, \lambda_1$ , and eigenvectors  $\mathbf{v}_0, \mathbf{v}_1$  are

established. Since the matrix  $\mathbf{C}$  is two-dimensional, its eigen decomposition is a matter of solving a quadratic equation. The ellipse semiaxes  $e_0, e_1$  are calculated simply by

$$\mathbf{e}_i = 2\lambda^{1/2}_i \mathbf{v}_i. \tag{4}$$

The final test verifying the pattern roundness is performed by checking if the inequality

$$\rho_{prec} > \left| \pi \frac{|\mathbf{e}_0||\mathbf{e}_1|}{s} - 1 \right| \tag{5}$$

holds, where  $s$  is the pattern size in the number of pixels. Unlike in the previous roundness test (2), the tolerance value of  $\rho_{prec}$  can be much lower because (4) establishes the ellipse dimensions with subpixel precision.

Here, it is worth mentioning that if the system runs on embedded hardware, it might be desirable to calculate  $\mathbf{C}$  using integer arithmetic only. However, the integer arithmetic might result in a loss of precision, therefore  $\mathbf{C}$  should be calculated as

$$\mathbf{C} = \frac{1}{s} \sum_{i=0}^{s-1} \begin{pmatrix} u_i u_i & u_i v_i \\ u_i v_i & v_i v_i \end{pmatrix} - \begin{pmatrix} uu & uv \\ uv & vv \end{pmatrix}, \tag{6}$$

where  $u_i$  and  $v_i$  are the pattern's pixel coordinates stored in the queue and  $u, v$  denote the determined pattern center.

### 3.4 Pattern Identification

The ratio of the patterns' inner and outer diameters does not have to be a fixed value, but can vary between the individual patterns. Therefore, the variable diameter ratio can be used to distinguish between individual circular patterns. If this functionality is required, the system user can print patterns with various diameter ratios and use these ratios as ID's.

However, this functionality requires to relax the tolerance ranges for the tests of inner/outer segment area ratio, which might (in an extreme case) cause false positive detections. Variable inner circle dimensions also might mean a smaller inner circle or a thinner outer ring, which might decrease the maximal distance at which the pattern is detected reliably. Moreover, missing a priori knowledge of the pattern's diameter ratio means that compensation for incorrect diameter estimation is not possible, which might slightly decrease the method's precision.

### 3.5 Compensation of Incorrect Diameter Estimation

The threshold separating black and white pixels has a significant impact on the estimation of the pattern dimensions. Moreover, the pixels on the black/white border are affected by chromatic aberration, nonlinear camera sensitivity, quantization noise, and image compression, see Fig. 2. As a result, the borderline between the black ring and its white background contains a significant number of misclassified pixels.

The effect of pixel misclassification is observed as an increase of the ratio of white to black pixels with increasing pattern distance. The effect causes the black ring to appear thinner (and smaller), which has a negative impact on the distance estimation. However, the inner and outer diameters of the pattern are known, and therefore, the knowledge of the true  $d_o$  and  $d_i$  can be used to compensate for the aforementioned effect. First, we can establish the dimensions of the inner white ellipse  $e_{0i}$  and  $e_{1i}$  in the same way as in Section 3.3. We assume the pixel misclassification enlarges the inner ellipse semiaxes  $e_{0i}$ ,  $e_{1i}$  and shrinks the outer semiaxes  $e_{0o}$ ,  $e_{1o}$  by a value of  $t$ . Since the real inner  $d_i$  and outer  $d_o$  pattern diameters are known, the true ratio of the areas can be expressed as

$$\frac{d_i^2}{d_o^2} = r = \frac{(e_{0i} - t)(e_{1i} - t)}{(e_{0o} + t)(e_{1o} + t)}, \tag{7}$$

where  $t$  can be calculated as a solution of the quadratic equation

$$(1 - r)t^2 - t(e_{0i}e_{1i} + re_{0o}e_{1o}) + e_{0i}e_{1i} - re_{0o}e_{1o} = 0. \tag{8}$$

The ambiguity of the solution can be resolved simply by taking into account that the corrected semiaxes lengths  $e_{0i} - t$ ,  $e_{1i} - t$  must be positive. The compensation of the pattern diameter reduces the average localization error by approximately 15 %.

### 3.6 Multiple Target Detection

The described roundel detection method can also be used to detect and track several targets in the scene. However, a single threshold  $\tau$  is not well suited to detecting more patterns because of illumination variances. Besides, other differences presented across the working area may affect the reflectance of the pattern and thus result in different gray levels for different patterns, which in turn requires a different  $\tau$  value for each pattern. Individual thresholding values not only provide detection robustness but also increase precision by optimizing pixel classification for each target individually.

Multiple targets can be simply detected in a sequence one by one, and the only requirement is to avoid detection of the already detected pattern. This can be easily avoided by modifying the input image after a successful detection by painting over the corresponding pixels, i.e., effectively masking out the pattern for subsequent detection runs.

Detection of multiple targets can also be considered in parallel, e.g., for obtaining additional performance gain, using a multi core processor. In this case, it is necessary to avoid a possible race condition and mutual exclusion has to be used for accessing the classification result storage.

An initial implementation of the parallel approach using OpenMP and multi-processor system did not yield a significant speedup. Furthermore, due to the



Fig. 2 Undesired effects affecting the pattern edge

high performance of detection of a single pattern, the serial implementation provides better performance than the parallel approach. Therefore, all the presented computational results in this paper are for the serial implementation.

#### 4 Pattern Localization

The relative pattern position to the camera module is calculated from the parameters established in the previous step. We assume that the radial distortion of the camera is not extreme and the camera intrinsic parameters can be established by the method [21] or similar. With this assumption, the pattern's position is computed as follows:

1. The ellipse center and semiaxes are calculated from the covariance matrix eigenvectors and transformed to a canonical camera coordinate system.
2. The transformed parameters are then used to establish coefficients of the ellipse characteristic equation, which is a bilinear form matrix (also called a cubic).
3. The pattern's spatial orientation and position within the camera coordinate frame is then obtained by means of eigen analysis of the cubic.
4. The relative coordinates are transformed to a two- or three-dimensional coordinate frame defined by the user.

A detailed description of the pattern position estimation is presented in the following sections.

##### 4.1 Ellipse Vertices in the Canonical Camera System

The ellipse center  $u'_c, v'_c$  and semiaxes  $e'_0, e'_1$  are established in a canonical camera form. The used canonical form is a pinhole camera model with unit focal lengths and no radial distortion. The transformation to a canonical camera system is basically a transform inverse to the model of the actual camera.

First, we calculate the image coordinates of the ellipse vertices  $\mathbf{a}_{0,1}$  and co-vertices  $\mathbf{b}_{0,1}$ , and transform them to the canonical camera coordinates  $\mathbf{a}'_{0,1}, \mathbf{b}'_{0,1}$ . The canonical coordinates of the (co)vertices are then used to establish the canonical center and canonical semiaxes. This rather complicated step is performed to compensate for the radial

distortion of the image at the position of the detected ellipse.

Since the ellipse center  $\mathbf{u}$  and semiaxes  $\mathbf{e}_0, \mathbf{e}_1$  are known, calculation of the canonical vertices  $\mathbf{a}'_{0,1}$  and co-vertices  $\mathbf{b}'_{0,1}$  is done simply by adding the semiaxes to the ellipse center and transforming them:

$$\begin{aligned} \mathbf{a}'_{0,1} &= g'((u \pm e_{0x} - c_x)/f_x, (v \pm e_{0y} - c_y)/f_y) \\ \mathbf{b}'_{0,1} &= g'((u \pm e_{1x} - c_x)/f_x, (v \pm e_{1y} - c_y)/f_y) \end{aligned}$$

where  $g'$  is the radial undistortion function and  $f_{x,y}, c_{x,y}$  are the camera focal lengths and optical center, respectively. Using the canonical position of the ellipse vertices, the ellipse center  $u', v'$  and axes  $\mathbf{e}'_0, \mathbf{e}'_1$  are then calculated as

$$\begin{aligned} \mathbf{e}'_0 &= (\mathbf{a}'_0 - \mathbf{a}'_1)/2 \\ \mathbf{e}'_1 &= (\mathbf{b}'_0 - \mathbf{b}'_1)/2 \\ \mathbf{u}'_c &= (\mathbf{a}'_0 + \mathbf{a}'_1 + \mathbf{b}'_0 + \mathbf{b}'_1)/4 \end{aligned}$$

After this step, we have all essential variables to calculate the ellipse characteristic equation.

##### 4.2 Ellipse characteristic equation

Notice that each point  $u, v$  lying on an ellipse satisfies the characteristic equation of an ellipse:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T \begin{pmatrix} q_a & q_b & q_d \\ q_b & q_c & q_e \\ q_d & q_e & q_f \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \mathbf{X}^T \mathbf{Q} \mathbf{X} = 0, \quad (9)$$

where  $\mathbf{Q}$  is called a conic. Thus, the parameters of the matrix  $\mathbf{Q}$  are calculated from the ellipse center and axes as follows:

$$\begin{aligned} q_a &= +e'_{0u}e'_{0u}/|e'_0|^2 + e'_{0v}e'_{0v}/|e'_1|^2 \\ q_b &= +e'_{0u}e'_{0v}/|e'_0|^2 - e'_{0u}e'_{0v}/|e'_1|^2 \\ q_c &= +e'_{0u}e'_{0u}/|e'_1|^2 + e'_{0v}e'_{0v}/|e'_0|^2 \\ q_d &= -u'_c q_a - v'_c q_b \\ q_e &= -u'_c q_b - v'_c q_c \\ q_f &= +q_a u'^2_c + q_c v'^2_c + 2q_b u'_c v'_c - 1 \end{aligned} \quad (10)$$

##### 4.3 Pattern Position

Once the conic parameters  $\mathbf{Q}$  are known, the position and orientation of the pattern can be obtained

by means of eigenvalue analysis [22]. Let the  $\mathbf{Q}$  matrix eigenvalues and eigenvectors be  $\lambda_0, \lambda_1, \lambda_2$  and  $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2$ , respectively. Since  $\mathbf{Q}$  represents an ellipse, its signature is (2, 1) and we assume that  $\lambda_0 \geq \lambda_1 > 0 > \lambda_2$ . According to [16], the position of the circle can be calculated as:

$$\mathbf{x}_c = \pm \frac{d_o}{\sqrt{-\lambda_0 \lambda_2}} \left( \mathbf{q}_0 \lambda_2 \sqrt{\frac{\lambda_0 - \lambda_1}{\lambda_0 - \lambda_2}} + \mathbf{q}_2 \lambda_0 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_0 - \lambda_2}} \right),$$

where  $d_o$  is the circular pattern diameter. The ambiguity of the sign can be resolved by taking into account that the pattern is located within the camera field of view. Thus, if the first component of the  $\mathbf{x}_c$  vector is negative, the vector  $\mathbf{x}$  is simply inverted.

#### 4.4 Transformation to the Global Coordinates

The position  $\mathbf{x}_c$  of the circular pattern established in the previous step is in a camera centered coordinate frame. Depending on the particular application scenario, our system allows to transform the pattern coordinates to a 3D or 2D coordinate frame defined by the user. The user just places four circular patterns in the space covered by the camera and provides the system with their real positions.

##### 4.4.1 Global Coordinate Frame – 3D Case

In the case of the 3D localization, the three patterns at positions  $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$  define the coordinate origin and  $x$  and  $y$  axes, respectively. The transformation between the global  $\mathbf{x} = (x, y, z)^T$  and camera centered  $\mathbf{x}_c = (x_c, y_c, z_c)^T$  coordinate systems can be represented as

$$\mathbf{x} = \mathbf{T} (\mathbf{x}_c - \mathbf{t}_0),$$

where  $\mathbf{t}_0$  equals  $\mathbf{x}_0$  and  $\mathbf{T}$  is a similarity transformation matrix.

The user can define the coordinate system simply by putting three “calibration” patterns in the camera field of view and designating the pattern that defines the coordinate system origin  $\mathbf{t}_0$  and the  $x$  and  $y$  axes. Using the pattern positions (let us define them as  $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$ , respectively), the system calculates the transformation between the camera and global coordinate systems, i.e., the vector  $\mathbf{t}_0$  and matrix  $\mathbf{T}$ . Establishing the translation vector  $\mathbf{t}$  is straightforward – it corresponds to the camera coordinates of the pattern at the global coordinate origin, i.e.,  $\mathbf{t} = \mathbf{x}_0$ .

The  $x$  and  $y$  axes of the coordinate frame are defined by vectors  $\mathbf{t}_1 = \mathbf{x}_1 - \mathbf{x}_0$  and  $\mathbf{t}_2 = \mathbf{x}_2 - \mathbf{x}_0$ , respectively. Since we assume an orthonormal coordinate system, the  $z$  axis vector can be simply calculated as a cross product  $\mathbf{t}_3 = \mathbf{t}_1 \times \mathbf{t}_2$ . From an algebraic point of view, the matrix  $\mathbf{T}$  represents a transformation of the vector  $\mathbf{x}' = \mathbf{x} - \mathbf{t}$  to a coordinate system defined by the basis  $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ . Therefore, the matrix  $\mathbf{T}$  can be calculated simply as

$$\mathbf{T} = \begin{pmatrix} t_{1x} & t_{2x} & t_{3x} \\ t_{1y} & t_{2y} & t_{3y} \\ t_{1z} & t_{2z} & t_{3z} \end{pmatrix}^{-1}. \tag{11}$$

Having established the vector  $\mathbf{t}$  and matrix  $\mathbf{T}$ , any point in the camera coordinate frame can be transformed to the coordinate frame defined by the user.

When the user places four patterns in the camera field of view, four independent coordinate transformations are calculated using each pattern triplet. The pattern position  $x'$  is then calculated as their mean, which results in increased system accuracy.

##### 4.4.2 Global Coordinate Frame – 2D Case

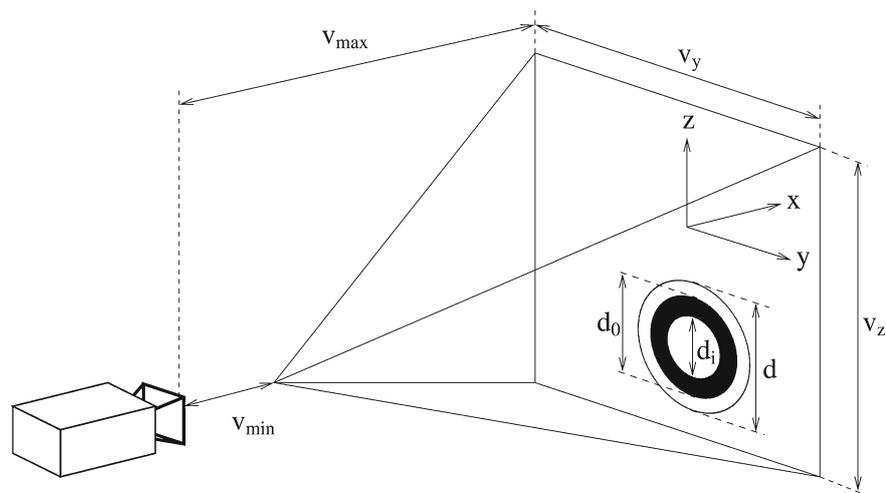
Two-dimensional localization can be generally more precise than full three-dimensional localization. This is because the estimation of the pattern position depends mainly on the pattern distance, especially in cases when the pattern image is small. Estimation of the pattern distance can be simply avoided if all the patterns are located only in a plane, e.g., ground robots operating on a floor.

In this case, the transformation from the image coordinates to an arbitrary world plane is a homography, and (homogeneous) spatial coordinates  $\mathbf{x}$  of the patterns can be calculated directly from their canonical coordinates  $\mathbf{u}'$  simply by  $\mathbf{x} = \mathbf{H}\mathbf{u}'$ , where  $\mathbf{H}$  is a  $3 \times 3$  homography matrix. Similarly to the case of three-dimensional localization, the user can define  $\mathbf{H}$  just by placing four patterns in the camera field of view and providing the system with their positions in the desired coordinate frame.

## 5 Sensor Model

In this section, we present three mathematical models that can be used to estimate the expected performance of the system. The main purpose of these models is

**Fig. 3** Geometry of the operational space



to support the selection of the most suitable camera, processing hardware, and pattern size according to the particular application scenario. The first model calculates the localization system coverage from the pattern dimensions, camera resolution, and field of view. The second set of equations provides estimation of the localization precision based on the camera parameters, pattern dimensions, and required coverage. Finally, the third model estimates the necessary computational power to track the given number of patterns at the desired frame rate.

### 5.1 Localization System Coverage

Regarding the practical deployment of the localization system, its most critical property is its coverage or “operational space”, i.e., the space where the pattern is reliably detected and localized. The dimensions of the operational space are affected by the camera focal length and radial distortion, image resolution, pattern diameter, and pattern spatial orientation.

For the sake of simplicity, the effect of radial distortion on the shape of the operational space is neglected and an ideal pinhole camera is assumed. Considering this ideal model, the operational space has a pyramidal shape with its apex close to the camera, see Fig. 3. The parameters of the operational space are the minimal and maximal detectable distances  $v_{min}$ ,  $v_{max}$  and base dimensions  $v_y$ ,  $v_z$ .

A pattern can be detected if it “fits” in the image and its central part and black ring are recognizable. Therefore, the pattern image dimensions must be

lower than the camera resolution, but higher than a certain value. To estimate the dimensions, we assume the camera focal lengths  $f_x$ ,  $f_y$  and radial distortion parameters have been established by a calibration tool<sup>1</sup>, e.g., MATLAB calibration toolbox or similar software based on [21]. Then, the width and height  $w_p$ ,  $h_p$  of the pattern in pixels can be calculated by

$$w_p = f_x \frac{d_o}{x} \cos(\varphi), \quad h_p = f_y \frac{d_o}{x} \cos(\psi), \quad (12)$$

where  $x$  is the pattern distance from the image plane,  $d_o$  is the pattern diameter, and  $\varphi$  and  $\psi$  represent the pattern tilt.

#### 5.1.1 Minimal Localization Distance

The minimal distance  $v_{min}$ , at which the pattern can be detected regardless of its orientation, is given as

$$v_{min} = d_o \max\left(\frac{f_x}{w}, \frac{f_y}{h}\right), \quad (13)$$

where  $w$  and  $h$  is the image horizontal and vertical resolution in pixels, respectively. One has to realize that the fractions  $f_x/w$  and  $f_y/h$  correspond to the camera field of view. Hence, the camera field of view remains the same regardless of the current resolution settings and the distance  $v_{min}$  can be considered as independent of the camera resolution.

<sup>1</sup>Such a tool is also a part of the proposed system available online at [23].

### 5.1.2 Maximal Localization Distance

The pattern has to be formed from a sufficient number of pixels to be detected reliably. Therefore, the pattern pixel dimensions have to exceed a certain value that we define as  $D$ . The value of  $D$  has been experimentally established as 12. We also found that  $D$  might be lower than this threshold for exceptionally good lighting conditions; however,  $D = 12$  represents a conservative value. Having  $D$ , the maximal detectable distance  $v'_{max}$  of the pattern can be calculated as

$$v'_{max} = \frac{d_o}{D} \min(f_x \cos(\varphi), f_y \cos(\psi)). \quad (14)$$

Notice a higher camera resolution increases the focal lengths  $f_x$  and  $f_y$ ; so, setting the camera resolution as high as possible maximizes the area covered by the localization system.

On the other hand, (14) does not take into account the camera radial distortion and it is applicable only when the pattern is located near the optical axis. The radial distortion causes the objects to appear smaller as they get far away from the optical axis. Thus, the distance  $v'_{max}$  at which the pattern is detected along the optical axis is higher than the maximal detectable distance  $v_{max}$  of the pattern located at the image corners. Therefore, the dimension  $v_{max}$  of the operational space is smaller than  $v'_{max}$  by a certain factor and  $v_{max}$  can be calculated as

$$v_{max} = \frac{d_o}{D} \min(k_x f_x \cos(\varphi), k_y f_y \cos(\psi)), \quad (15)$$

where  $k_x$  and  $k_y$  represent the effect of the radial distortion. The values of  $k_x$  and  $k_y$  can be estimated from the differential of the radial distortion function close to an image corner:

$$\begin{aligned} k_x &= 1 + \frac{dg(r_x, r_y)}{dx} \\ &= 1 + 2k_1 r_x + 4k_2(r_x^3 + r_x r_y^2) + \dots, \\ k_y &= 1 + \frac{dg(r_x, r_y)}{dy} \\ &= 1 + 2k_1 r_y + 4k_2(r_y^3 + r_y r_x^2) + \dots \end{aligned}$$

where  $r_x$  and  $r_y$  can be obtained from the camera optical axis and focal lengths as  $c_x/f_x$  and  $c_y/f_y$ . For a consumer grade camera, one can assume that the radial

distortion would not shrink the pattern more than by 10 %; so, a typical value of  $k_{x,y}$  would be between 0.9 and 1.0.

### 5.1.3 Base Dimensions

Knowing the maximal detectable distance  $v_{max}$ , the dimensions of the localization area “base”  $v_y$  and  $v_z$  can be calculated as

$$v_y = w \frac{v_{max}}{f_x} - 2d_o, \quad v_z = h \frac{v_{max}}{f_y} - 2d_o, \quad (16)$$

where  $w$  and  $h$  are the horizontal and vertical resolutions of the camera used, respectively. Considering a typical pattern, the value of  $d_o$  is much smaller than the localization area and can be omitted.

With Equations (13), (15), and (16) the user can calculate the diameter of the pattern and camera parameters from the desired coverage of the system. It should be noted that the presented model considers a static configuration of the module and the detected pattern. Rapid changes of the pattern’s relative position may cause image blur, which might affect  $v_{max}$  and restrict the operational space.

## 5.2 Localization System Precision

Another important property of the localization system is the precision with which the system provides estimation of the pattern position. The precision of the localization is directly influenced by the amount of noise in the image and uncertainty in the camera parameters. The position estimation error also depends on the system operational mode, i.e., it is different for the three-dimensional and two-dimensional position estimations. The expected localization precision is discussed in the following sections for both the 2D and 3D cases.

### 5.2.1 Two-dimensional Localization by Homography

For the 2D localization, the pattern position is estimated simply from its center image coordinates. In the case of an ideal pinhole camera, the calibration procedure described in Section 4.4 should establish the relation between the image and world planes. Therefore, the precision of the position estimation is affected mainly by the image radial distortion. Since the uncertainties of the radial distortion parameters are

known from the camera calibration step, the error of radial distortion for  $x$  and  $y$  can be estimated from the differential of the radial distortion function

$$\begin{aligned} \eta_x &= x(\epsilon_1 r + \epsilon_2 r^2 + \epsilon_5 r^3 + 2\epsilon_3 y) + \epsilon_4(r + 2x^2) \\ \eta_y &= y(\epsilon_1 r + \epsilon_2 r^2 + \epsilon_5 r^3 + 2\epsilon_4 x) + \epsilon_3(r + 2y^2), \end{aligned} \quad (17)$$

where  $\eta_x$  and  $\eta_y$  are the position relative errors,  $k_i$  are camera distortion parameters,  $\epsilon_i$  are their uncertainties, and  $r = x^2 + y^2$ . The overall relative error of the two-dimensional localization can be expressed as

$$\eta_{hom} = \eta_{rad} = \sqrt{\eta_x^2 + \eta_y^2}. \quad (18)$$

Note that (17) does not take into account the camera resolution. Therefore, the model suggests that higher resolution cameras will not necessarily achieve better localization precision. This is further investigated in Section 6.2.2, where experimental results are presented. Also, note that in the standard camera calibration implementations, values of  $\epsilon_i$  are meant as 99.7 % confidence intervals. To calculate the average error, i.e., the standard deviation, one has to divide  $\eta_{hom}$  by three.

### 5.2.2 Full Three-dimensional Localization

In the full 3D localization, the main source of the localization imprecision is incorrect estimation of the pattern distance. Since the pattern distance is inversely proportional to its diameter in pixels, smaller patterns will be localized with a higher error. The error in the diameter measurement is caused by quantization noise and by the uncertainty in the identification of the camera's intrinsic parameters, especially in the parameters of the image radial distortion. One can roughly estimate the expected error in the pattern distance estimation as

$$\eta_{3D} = \frac{\Delta f}{f_x} + \Delta e_0 \frac{x f_x}{d_0} + \eta_{rad}, \quad (19)$$

where  $\Delta f$  is the error of the focal length estimation,  $\Delta e$  represents the error of the ellipse axis estimation due to image noise, and  $\eta_{rad}$  is the relative error of the radial distortion model. While  $\Delta f$  and  $\eta_{rad}$  can be calculated from the camera calibration parameters,  $\Delta e_0$  is influenced by a number of factors that include camera thermal noise, lighting, motion blur etc. However, its current value can be estimated on the fly from the variance of the calibration (see Section 4.4.1) patterns' diameters.

In our experiments, the typical value of  $\Delta e_0$  was around 0.15 pixels. This means that for a well-calibrated camera, the major source of distance estimation error is the ratio of image noise to the pattern projection size. Since the pattern image size (in the number of pixels) grows with the camera resolution, the precision of localization can be increased simply by using a high resolution camera or a larger pattern.

### 5.3 Computational Requirements

From a practical point of view, it is also desirable to estimate the necessary computational hardware needed to achieve a desired frame rate, especially for an embedded solution. The time needed to process one image can be roughly estimated from the number of patterns, their expected size, image dimensions, tracking failure rate, and the computer speed. For the sake of simplicity, we can assume that the time to process one frame is a linear function of the amount of processed pixels:

$$t = (k_0 + k_1(s_p(1 - \alpha) + s_i \alpha)) n o, \quad (20)$$

where  $k_0$  represents the number of operations needed per pattern regardless of its size (e.g., a coordinate transformation),  $k_1$  is a constant corresponding to the number of operations per pixel per pattern,  $s_p$  is the average size of the pattern in pixels,  $\alpha$  is the expected failure rate of the tracking,  $s_i$  is the image size in pixels,  $n$  is the number of tracked patterns, and  $o$  is the number of operations per second per processor core given as a ratio  $o = c/m$  of the entire processor MIPS (Million Instructions Per Second)  $m$  and the number of processor cores  $c$ . The constant  $k_0$  has been experimentally estimated as  $5 \cdot 10^5$  and  $k_1$  as 900. The average size  $s_p$  of a pattern can be calculated from the camera parameters, pattern diameter, and average distance from the camera by (12). Thus, if the user wants to track 50 patterns with 30 pixel diameter using a machine with two cores and 53 GIPS (Giga Instructions Per Second), the expected processing time per image would be 1.2 ms, which would allow to process about 800 images per second.

The speed of the localization algorithm depends on the failure rate of the tracking  $\alpha$ . Typically, if the pattern displacement between two frames is smaller than the pattern radius, the tracking mechanism causes the method to process only the pixels belonging to the pattern. This situation corresponds to  $\alpha$  being equal to

zero. Thus, assuming that the pattern is not moving erratically, the method's computational complexity is independent of the processed image size. Moreover, the smaller the pattern image dimensions, the faster the processing rate is. Of course, equation (20) gives only a coarse estimate, but it might give the user a basic idea of the system processing speed. Equation (20) has been experimentally verified and the results are presented in Section 6.3.

## 6 Experiments

This section is dedicated to presentation of the experimental results verifying the mathematical models established in Section 5. First, the model of the operational space defined by (15) and (16) is tested to see if it corresponds to a real situation. After that, the real achievable precision of the localization is evaluated according to the model (17) and (19). Then, the real computational requirements of the algorithm are measured using different computational platforms and the model in (20) is validated. Finally, the performance of the proposed localization system is also evaluated according to the precise motion capture system and compared with the AR tag based approach ArUco [14] and the simple OpenCV circle detector.

### 6.1 Operational Space for a Reliable Pattern Detection

The purpose of this verification is to validate the model describing the area covered by the localization system. In Section 5.1, the covered space is described as a pyramid with base dimensions  $v_y$ ,  $v_z$  and a height denoting the maximal detectable distance  $v_{max}$ .

#### 6.1.1 Maximal Detection Distance

A key parameter of the operational space is the maximal distance for reliable pattern detection  $v_{max}$  that is described by (14). The following experimental setup has been used to verify the correctness of this model. Two different cameras have been placed on a mobile platform SCITOS-5 with precisely calibrated odometry. The proposed localization system was set up to track three circles, each with a different diameter. The platform has been set to move away from the circles at a constant speed and its distance from the patterns was recorded whenever a particular pattern was not

detected. The recorded distances are considered as the limit  $v'_{max}$  of the system operational space. The same procedure was repeated with the patterns being slanted by forty degrees (Table 1).

During this experiment, the patterns were located approximately at the image center. As previously noted in Section 5.1.2, additional correction constants  $k_x$ ,  $k_y$  have been introduced in Section 5.1.2 to take into account radial distortion effects, which cause the detected pattern to appear smaller when located at the image edges. The augmented model considering the radial distortion was verified in an additional experiment using a pattern with diameter 2.5 cm positioned at the image corner. In this case, the maximal detected distance was reduced by 7 % for a Logitech QuickCam Pro camera and by 7 % for an Olympus VR-340. These results are in a good accordance with the model introduced in Section 5.1.2, where the values of  $k_x$  and  $k_y$  were estimated to be between 0.9 and 1.0.

#### 6.1.2 Base Dimensions

The dimensions of the coverage base are modeled by (16), which provides the dimensions of the expected coverage  $v_y$  and  $v_z$ . This model was verified using a similar setup to the previous experiment. The camera was placed to face a wall at a distance established in the previous experiment and four patterns were placed at the very corners of the image. This procedure was repeated for three different sizes of the pattern. The operation space dimensions, both measured and calculated by (16), are summarized in Table 2.

**Table 1** Maximal distance for a reliable pattern detection

Camera type	Pattern $d_o[cm]$	Distance [m]			
		Measured		Predicted	
		0°	40°	0°	40°
Logitech QC Pro	2.5	1.4	1.3	1.6	1.3
	5.0	3.3	2.8	3.2	2.5
Olympus VR-340	7.5	4.4	3.9	4.9	3.7
	2.5	6.8	6.2	6.7	5.1
	5.0	13.2	11.4	13.4	10.3
	7.5	19.8	16.8	20.1	15.4

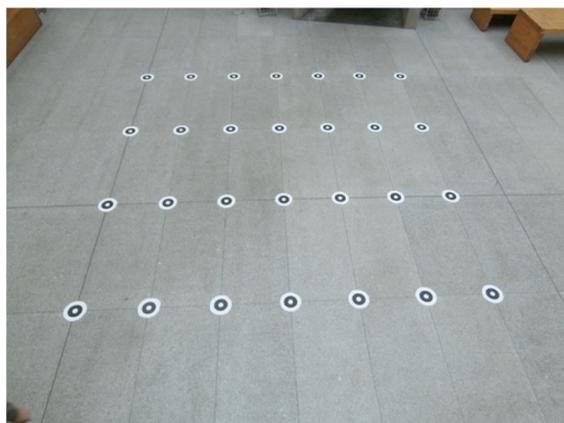
**Table 2** Dimensions of the operational space

$d_o$ [cm]	Dimensions [m]					
	Measured			Predicted		
	$v_{max}$	$v_y$	$v_z$	$v_{max}$	$v_y$	$v_z$
2.5	1.6	2.1	1.6	1.5	1.9	1.4
5.0	3.0	3.9	3.0	3.0	4.0	3.0
7.5	4.5	5.9	4.5	4.4	5.8	4.4

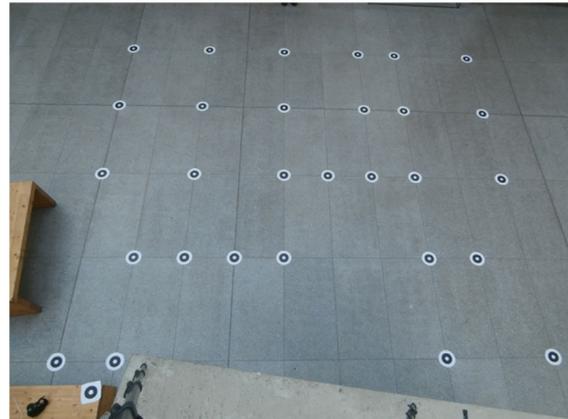
### 6.2 Localization Precision

The real localization precision, which is probably the most critical parameter of the localization system, was established experimentally using a dataset collected in the main entrance hall of the Faculty of Mechanical Engineering at the Charles square campus of the Czech Technical University. The entrance hall offers enough space and its floor tiles form a regular rectangular grid with dimensions  $0.625 \times 1.250$  m. The regularity of the grid was verified by manual measurements and the established precision of the tile placement is around 0.6 mm.

We placed several patterns on the tile intersections and took five pictures with three different cameras from two different viewpoints (Figs. 4 and 5). The cameras used were a Creative Live! webcam, Olympus VR-340, and Canon 550D set to  $1280 \times 720$ ,  $4608 \times 3456$ , and  $5184 \times 3456$  pixel resolutions, respectively. The viewpoints were chosen at two different heights; so, the images of the scene were taken from a “side” and a “top” view.



**Fig. 4** Side view of the experiment



**Fig. 5** Top view of the experiment

First, three or four of the patterns in each image were used to define the coordinate system. Then, the resulting transformation was utilized to establish the circle global positions. Since the circles were placed on the tile corners, their real positions were known precisely. The Euclidean distances of these known positions to the ones estimated by the system were considered as the measure of the localization error.

#### 6.2.1 Three-dimensional Localization Precision

In this test, the system was set to perform full three-dimensional localization. In this model, the most significant cause of the localization error is the wrong distance estimation of the pattern (as noted in Section 5.2.2). The distance measurement is caused by an imperfect estimation of the pattern semiaxes lengths, see (19). The equation indicates that a camera with a higher resolution would provide a better precision.

The measured and predicted average and maximal localization errors for the individual pictures are shown in Table 3. The table also contains the predicted average localization error  $\eta_{pred}$  calculated by (3) for a comparison of the model and the real achieved precision.

#### 6.2.2 Two-dimensional Localization Precision

In the case of indoor ground robot localization, we can assume that the robots move in a plane. The plane where the robots move and the image plane form a homography, which was previously defined by

**Table 3** Precision of 3D position estimation

Image		Abs. [cm]		Rel. [%]		
camera	view	$\epsilon_{avg}$	$\epsilon_{max}$	$\eta_{pred}$	$\eta_{avg}$	$\eta_{max}$
Webcam	side	5.7	19.5	1.04	0.90	2.96
Webcam	top	3.7	12.1	0.68	0.61	1.83
VR-340	side	1.9	6.5	0.47	0.35	1.02
VR-340	top	3.2	11.0	0.54	0.50	1.39
C-550D	top	2.5	7.4	0.30	0.43	1.46

four reference patterns during the system setup. The real achievable precision of two-dimensional localization was measured within the same experimental scenario as the previous full 3D case. The average and maximal measured localization errors are depicted in Table 4. Similar to the previous case, the table contains the predicted mean error  $\eta_{pred}$  calculated by (17).

The results indicate that the assumption of ground plane movement increases the precision by an order of magnitude. Moreover, the results also confirm that increasing the image resolution does not necessarily increase the localization precision. Rather, the precision of localization is influenced mostly by the camera calibration imperfections. This fact confirms the assumptions presented in Section 5.2.1.

### 6.3 Computational Requirements

The purpose of this experiment was to evaluate the estimation of the computational requirements provided by the model proposed in Section 5.3. Thus, the hypothesis is to test if the algorithm processing speed estimation (20) conforms to the proposed assumptions. Moreover, in this experiment, we also verify if

**Table 4** Precision of 2D position estimation

Image		Abs. [cm]		Rel. [%]		
camera	view	$\epsilon_{avg}$	$\epsilon_{max}$	$\eta_{pred}$	$\eta_{avg}$	$\eta_{max}$
Webcam	side	0.23	0.62	0.03	0.04	0.08
Webcam	top	0.18	0.68	0.04	0.03	0.09
VR-340	side	0.64	1.40	0.11	0.12	0.22
VR-340	top	0.68	2.08	0.19	0.11	0.32
C-550D	top	0.15	0.33	0.03	0.03	0.07

the algorithm complexity depends only on the pattern size rather than on the image resolution.

#### 6.3.1 Processing Time vs. Image and Pattern Dimensions

The model of computational requirements assumes that once the circles are reliably tracked, the system processing time is independent of the image size. In such a case, the image processing time is a linear function of the overall number of pixels belonging to all the patterns. Three synthetic datasets were created to verify this assumption. The first dataset consists of images with variable resolution and one circular pattern with a fixed size. The image resolutions of the second dataset are fixed, but the pattern diameter varies. Both pattern and image dimensions of the third dataset images are fixed; however, the number of patterns in each image ranges from one to four hundred. Each image of each dataset was processed one thousand times and the average time to track all the roundels in the image was calculated. The average processing time is shown in Fig. 6.

The presented results clearly show that the image processing time is proportional to the number of pixels occupied by the tracked circular patterns and does not depend on the processed image dimensions. Moreover, the results demonstrate the scalability of the algorithm, which can track four hundred robots more than one hundred times per second. The aforementioned tests were performed on a single core of the Intel iCore5 CPU running at 2.5 GHz and accompanied with 8 GB of RAM.

#### 6.3.2 Processing Time Using Different Platforms

From a practical point of view, processing images at a speed exceeding the camera frame rate is not necessary. Rather, the algorithm might be deployed on systems with slower processing units. Thus, one should be able to establish what kind of computational hardware is needed for a particular setup. This can be roughly estimated using the time to process one image by means of (20). Three real world datasets and five different platforms, including two credit-card sized computers, were used to verify the model in a realistic setup.

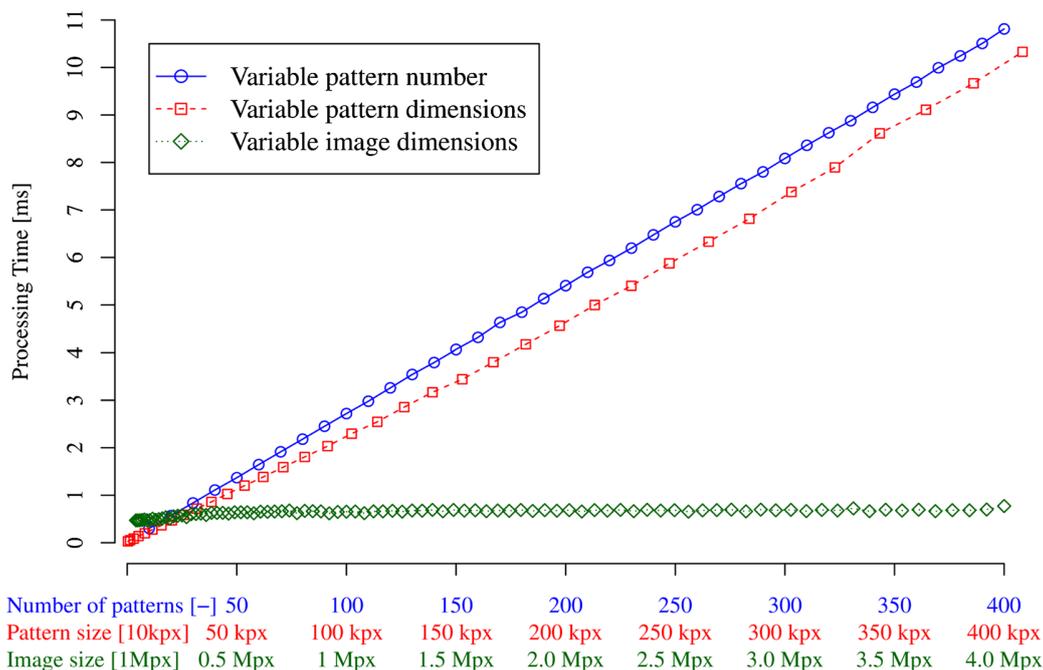


Fig. 6 Influence of the number of tracked patterns, pattern and image sizes on the method’s speed

- The “small” dataset consists of one thousand images of a static pattern, which occupies approximately seven hundred pixels, i.e., 0.1 % of the image’s total area.
- The “large” dataset is similar, but with a larger, sixty-pixel diameter pattern, occupying approximately 0.3 % of image pixels.
- The algorithm performance with these two datasets (“small” and “large”) is relevant in scenarios where the tracked objects are moving slowly and the camera is in a static position.
- The “fast” dataset contains 130 images of a fast moving pattern with a variable size. The dataset was tailored to cause failure of the tracking mechanism in one case. Thus, the performance of the algorithm with this dataset is similar to cases when the camera is not stationary or the tracked objects are moving quickly.

The average processing time per image for each dataset was measured and calculated by (20). The results summarized in Table 5 indicate the correctness of the model described in Section 5.3.

#### 6.4 Comparison with a Precise Localization System

The real achievable precision of the localization system has been reported in Section 6.2; however, only for experiments with static targets, where the patterns were placed at the predefined positions. Such a setup provides verification of the precision for scenarios where the system tracks slowly moving robots. On the other hand, rapid movement of the tracked targets introduces additional effects, which might have a considerable impact on the system precision. First, the

Table 5 Required image processing time

CPU	Processing time [ms]					
	Measured			Predicted		
	small	large	fast	small	large	fast
i-5 2450M	0.04	0.10	0.37	0.04	0.12	0.35
Atom N270	0.30	0.72	3.25	0.33	0.89	2.68
Pentium M	0.20	0.45	1.44	0.17	0.48	1.45
Odroid U2	0.27	0.89	2.76	0.29	0.79	2.86
Raspb. Pi	1.10	4.00	15.8	1.34	3.66	11.0

**Table 6** Localization accuracy of a moving target

Mode	Abs. [cm]		Rel. [%]	
	$\epsilon_{avg}$	$\epsilon_{max}$	$\eta_{avg}$	$\eta_{max}$
2D	1.2	4.2	0.4	1.5
3D	3.1	11.2	1.2	4.4

captured images can be affected by motion blur and deformation caused by the camera's rolling shutter. Besides, there might be a delay in position estimation because standard USB cameras deliver the images with a delay caused by the interface's limited bandwidth. Therefore, we consider an additional experiment to evaluate the impact of these factors on the real performance of the presented global localization system. We consider a precise reference system and set up our localization system in an area where a high-precision motion capture system is installed and which is able to track multiple targets<sup>2</sup>. The motion capture system provides positions of the tracked targets 250 times per second with a precision up to 0.1 mm; so, it can be considered as a ground truth for our position measurements.

Four reference targets were placed in the area and a common coordinate system was calculated for both systems. After that, four sequences of targets moving at speeds up to 1.2 m/s were recorded by a Logitech QuickCamPro and the commercial motion capture system. Euclidean distances of target positions provided by both systems were taken as a measure of our system accuracy. The mean precisions of two- and three-dimensional localization were established as 1.2 cm and 3.1 cm, respectively (Table 6). Although the system's relative accuracy is lower than in the static tests presented in Section 6.2, centimeter precision is still satisfactory for many scenarios. The error is caused mostly by the image blur because of a long exposure rate set by the camera internal control. Careful setting of the camera exposure and gain parameters might suppress this effect. In fact, such a tuning has been made for localization of flying quadrotors, see Section 7.1.

It is also worth to mention that even though the commercial system is able to localize rapidly moving targets with a higher precision, its setup took more

than thirty minutes while the presented system is prepared in a couple of minutes (just placing four patterns to establish the coordinate system).

## 6.5 Comparison with Other Visual Localization Systems

The advantages and drawbacks of the presented localization system are demonstrated by a comparison of its performance with the well-established localization approaches based on AR markers and OpenCV. The performance of AR-based markers has been measured using the ArUco [14] library for detection and localization of multiple AR markers (similar to the ones used in ARTag and ARToolKit systems). A comparison with the OpenCV circular pattern detection is based on the OpenCV's "SimpleBlobDetector" class. The precision, speed, and coverage of all three systems were established in a similar way as described in the previous sections. For the sake of simplicity, we will refer to the presented system as *WhyCon*.

### 6.5.1 Precision Comparison

The localization precision of the ArUco-, OpenCV-, and WhyCon-based localization methods was obtained experimentally by the method described in Section 6.2. The comparison was performed on  $4608 \times 3456$  pixel pictures taken by an Olympus VR-340 Camera from two different (*side* and *top*) viewpoints.

The achieved results are presented in Table 7. The WhyCon position estimation error is significantly lower than the error of ArUco and OpenCV in both the two- and three-dimensional localization scenarios.

**Table 7** Localization precision comparison

mode	view	Relative error [%]					
		WhyCon		ArUco		OpenCV	
		<i>avg</i>	<i>max</i>	<i>avg</i>	<i>max</i>	<i>avg</i>	<i>max</i>
2D	side	0.12	0.19	0.20	0.41	0.52	1.03
	top	0.12	0.32	0.22	0.37	0.77	1.62
3D	side	0.31	1.10	0.63	2.52	–	–
	top	0.33	1.04	1.08	2.90	–	–

<sup>2</sup>Human Performance Centre at the University of Lincoln

Moreover, we found that the OpenCV's blob radius calculation was too imprecise to reliably estimate the pattern distance and could not be used for the full 3D localization.

### 6.5.2 Performance Comparison on Different Platforms

The computational performance of the three evaluated systems was compared for three different platforms. The methods' performance was compared using two datasets similarly to the evaluation scenario described in Section 6.3.2. The *slow* dataset contains an easy-to-track pattern while for the *fast* datasets about 1 % of the images are tailored to cause a tracking failure.

The results presented in Table 8 indicate that the proposed algorithm is capable of finding the patterns approximately one thousand times faster than the traditional methods. Even in the unfavorable case where the patterns cannot be reliably tracked, the method outperforms ArUco and OpenCV hundred times. The performance ratio is even better for small embedded platforms with limited computational power. This property is favorable for deployment in the intended applications, especially under real-time requirements.

### 6.5.3 Range and Coverage Comparison

The AR fiducial markers are primarily intended for augmented reality applications and in a typical scenario, the localized marker is situated close to the camera. Therefore, the AR marker-based systems are not tuned for a reliable detection of distant patterns with

small image dimensions. Thus, the range and coverage of the AR marker-based systems would be lower compared to WhyCon. On the other hand, OpenCV's circular blob detector can detect small circular patterns.

To estimate the ArUco and OpenCV detectors maximal range, we have established the minimal size (in pixels) that the tags need to have in order to be detected reliably. The sizes that correspond to the minimal pattern diameter  $D$  in the Equation (15) were established in a similar way as described in Section 6.1.1. While the OpenCV detector can find blobs larger than 12 pixels, the ArUco detector requires the AR marker side to be longer than 25 pixels. Therefore, ArUco's maximal detection range is less than a half of WhyCon's or OpenCV's range.

## 7 Practical Deployment

In this section, we present an overview of several research projects where the proposed circle detection algorithm has been successfully employed. This practical deployment demonstrates the versatility of the presented localization system. A short description of each project and comment about the localization performance is presented in the following sub-sections.

### 7.1 UAV Formation Stabilization

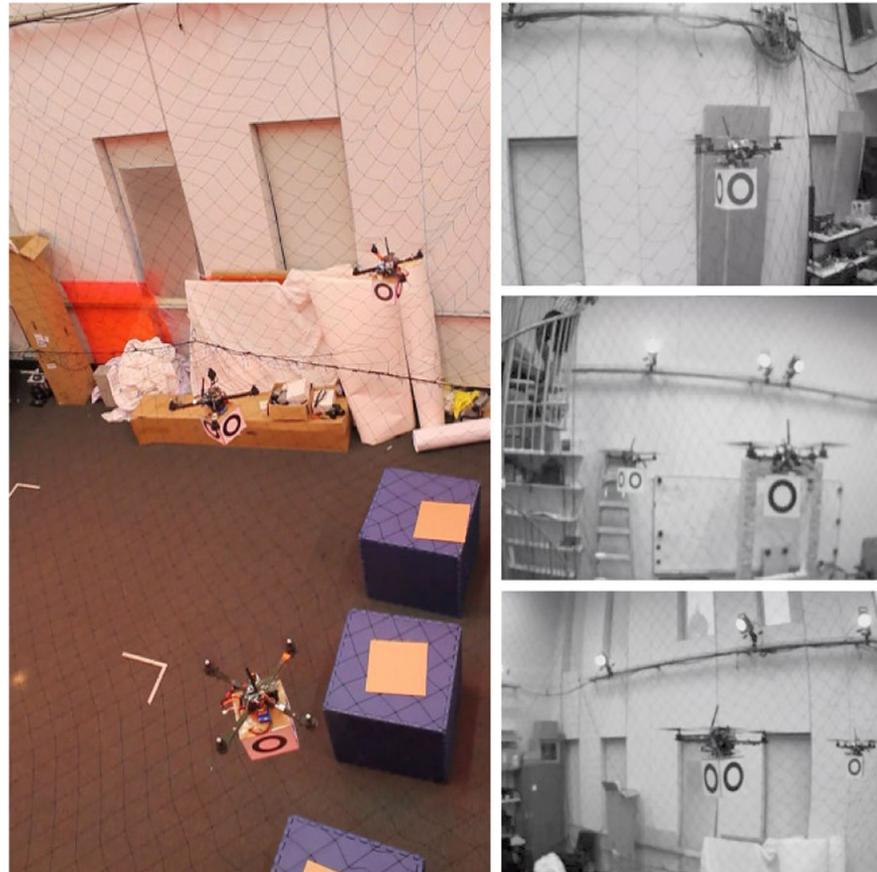
In this setup, the circle detection algorithm was considered for a relative localization and stabilization of UAV formations operating in both indoor and outdoor environments. A group of quadrotors are supposed to maintain a predefined formation by means of their relative localization. Each quadrotor UAV carries a circular pattern and an embedded module [24] running the localization method, see Fig. 7.

Thus, each UAV is able to detect other quadrotors in its vicinity and maintain a predefined relative position. Although the UAV's movements are relatively fast, we did not observe significant problems caused by image blur and the system detected the patterns reliably. This scenario demonstrates the ability to reliably detect circular patterns despite their rapid movements and variable lighting conditions. Moreover, it proved its ability to satisfy real-time constraints when running

**Table 8** Image processing time comparison

CPU	Processing time [ms]					
	ArUco		OpenCV		WhyCon	
	fast	slow	fast	slow	fast	slow
i5 2450M	19	19	63	62	0.35	0.04
Pentium M	121	119	329	329	1.00	0.18
Odroid U2	148	149	371	366	0.93	0.28
Raspb. Pi	875	875	1795	1759	6.59	1.21

**Fig. 7** Decentralized localization of quadrotor formation performed by the presented method. Courtesy of the GRASP laboratory, PENN



on computationally constrained hardware. The precision of the relative localization was in the order of centimeters [24].

### 7.2 Birds-eye UAV-based Localization System

The algorithm has also been used for relative localization of ground robots, which were supposed to maintain a predefined formation shape even if they lack direct visibility among each other. In this setup, one robot of the formation carried a heliport with the Parrot AR.Drone [25] quadrotor, which can take off and observe the formation from above using a downward-pointing camera. Each ground robot had a roundel pattern, which is elliptical rather than circular to provide also an estimate of the robot orientation. Using the roundel detection algorithm, the position and heading of the ground robots are provided by the flying quadrotor while it maintains its position above the formation.

Moreover, the heliport was designed by a circular pattern, which makes it possible to autonomously land the quadrotor after the mission end, see Fig. 8. Despite the relatively low resolution ( $168 \times 144$ ) of the UAV's downward-looking camera and its rapid movements,



**Fig. 8** Mixed UAV-UGV robot formation

the overall localization precision was approximately 5 cm.

### 7.3 Autonomous Docking of Modular Robots

The Symbion and Replicator projects [26] investigate and develop novel principles of adaptation and evolution of symbiotic multi-robot organisms based on bio-inspired approaches and modern computing paradigms. The robot organisms consist of large-scale swarms of robots, which can dock with each other and symbiotically share energy and computational resources within a single artificial life form. When it is advantageous to do so, these swarm robots can dynamically aggregate into one or many symbiotic organisms and collectively interact with the physical world via a variety of sensors and actuators. The bio-inspired evolutionary paradigms combined with robot embodiment and swarm-emergent phenomena enable the organisms to autonomously manage their own hardware and software organization.

In these projects, the proposed localization algorithm has been used as one of the methods for detecting power sources and other robots, see Fig. 9. The method demonstrated its ability to position the robot with a sub-millimeter precision, which is essential for a successful docking. The method's deployment in this scenario demonstrated not only its

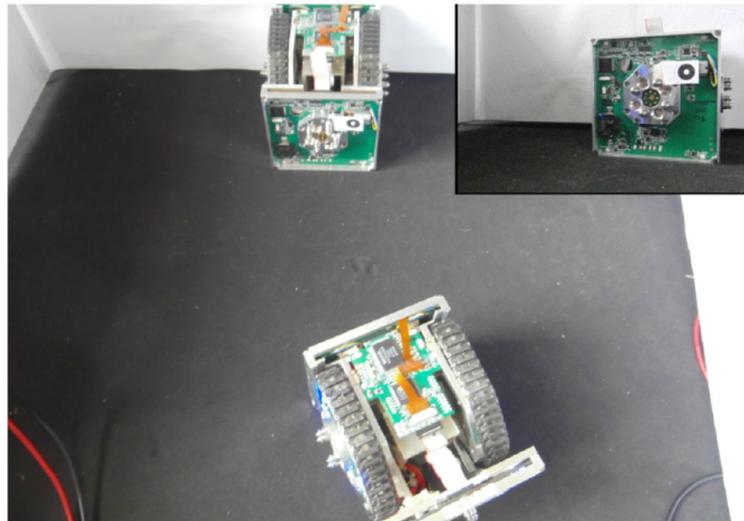
precision, but also its ability to run on computationally constrained hardware.

### 7.4 Educational Robotics

SyRoTek [19] is a remotely accessible robotic laboratory, where users can perform experiments with robots using their Internet connectivity. The robots operate within a flat arena with reconfigurable obstacles and the system provides an overview of the arena from an overhead camera. The project has been used for education and research by several institutions in Europe and Americas. An important component of SyRoTek is the localization system providing estimation of the real robots' positions.

Originally the localization was based on a convolution algorithm. Even though it is computationally demanding and rather imprecise, it demonstrated suitability for 24/7 operation. After replacement of this original localization system by the presented roundel-based system, the precision of the localization was improved. Moreover, the computational requirements were decreased as well [27]. In this deployment, the roundel pattern is formed from ellipses where the inner ellipse has slightly different dimensions, see Fig. 10, which allows to distinguish between individual robots. This use case demonstrates the ability of the system to operate in 24/7 mode. In addition, using different dimensions of

**Fig. 9** Symbion/Replicator robots during docking



**Fig. 10** A top-down view to the SyRoTek arena



the inner ellipse allows to distinguish between 14 SyRoTek robots.

#### 7.5 Ground Truth Assessment in Mobile Robot Navigation

BearNav (originally SURFNav) is a visual based navigation system for both ground [28] and aerial mobile [29] robots. The method is based on convergence theorem [30], which states that map-based monocular navigation does not need full localization, because if the robot heading is continuously adjusted to turn the robot towards the desired path, its position error does not grow above certain limits even if the position estimation is based only on proprioceptive sensing affected by drift. The aforementioned principle allows to design reliable and

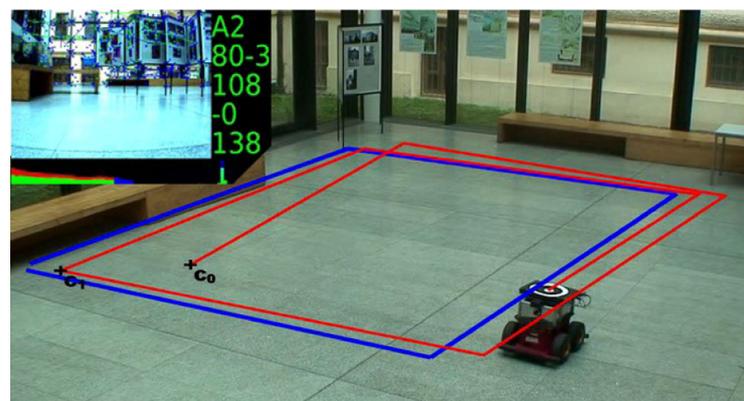
computationally inexpensive camera-based navigation methods.

The presented roundel based localization system was used to provide a continuous and independent measurement of the robot position error, which allowed to verify the convergence theorem and benchmark the individual navigation algorithms in terms of their precision, see Fig. 11. The system proved to be useful especially for aerial robots [29], which, unlike the ground robots, cannot be simply stopped for a manual position measurement.

#### 7.6 Autonomous Charging in Long-term Scenarios

The STRANDS project [31] aims to achieve intelligent robot behaviour in human environments through adaptation to, and the exploitation of, long-term

**Fig. 11** Reconstructed trajectory of a mobile robot





**Fig. 12** SCITOS-5 platform near its charging station. Notice the three o's of the label

experience. The project approach is based on a deeper understanding of ongoing processes affecting the appearance and structure of the robot's environment. This will be achieved by extracting qualitative spatio-temporal knowledge from sensor data gathered during months of autonomous operation. Control mechanisms that will exploit these structures to yield adaptive behaviour in highly demanding scenarios will be developed.

The circle detection method is used in the project as an initial solution of localization-related problems before more sophisticated implementations take its place. One of such deployments is the localization of the robot during its approach to a charging station, which has been solved by placing three patterns in the charging area, see Fig. 12.

## 8 Conclusion

We present a fast and precise vision-based system intended for multiple robot localization. The system's core component is based on a novel principle of circular roundel detection with computational complexity independent of the processed image size. The resulting system allows to localize swarms composed of

several hundreds of robots with millimeter (2D) or centimeter (3D) precision, while keeping up with standard camera frame rates. In addition, we provide a model to calculate the sufficient camera and computer parameters to achieve the desired localization precision, coverage and update rate, which support potential users to decide which kind of equipment is needed for their particular setup.

The most notable features of the system are its low computational requirements, ease of use, and the fact that it works with cheap, off-the-shelf equipment. The system has been deployed already in a number of international mobile robotic projects concerning distributed quad rotor localization [24], visual based autonomous navigation [30], decentralized formation control [25], long-term scenarios [31], evolutionary swarm [26], and educational [19] robotics. Since the system has already proved to be useful in a variety of applications, we publish its source code [23]; so, other roboteers can use it for their projects. The experiments indicate that the presented system is three orders of magnitude faster than traditional methods based on OpenCV or AR markers while being more precise and capable of detecting the markers at a greater distance.

In the future, we plan to increase the precision and coverage of the system by using multiple cameras. We will plan to improve the tracking success rate by predicting the position of the target by considering the dynamics of the tracked object.

**Acknowledgments** The European Union supported this work within its Seventh Framework Programme projects ICT-600623 "STRANDS" and 216240 "Replicator". The Ministry of Education of the Czech Republic has given support by 7AMB12AR022 and 7E08006. The Ministry of Science of Argentina supported this work by project ARC/11/11. The work of J. Faigl was supported by the Czech Science Foundation (GAČR) under research project No. 13-18316P. Christian Dondrup and David Mullineaux are acknowledged for help with experiments.

## References

1. Thrun, S., Burgard, W., Fox, D., et al.: Probabilistic robotics, vol. 1. MIT press Cambridge (2005)
2. Breitenmoser, A., Kneip, L., Siegwart, R.: A monocular vision-based system for 6D relative robot localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 79–85 (2011)

3. Yamamoto, Y., et al.: Optical sensing for robot perception and localization. In: IEEE Workshop on Advanced Robotics and its Social Impacts, pp. 14–17. IEEE (2005)
4. Vicon: Vicon MX Systems. <http://www.vicon.com>. [cited 8 Jan 2014]
5. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* **31**(5), 664–674 (2012)
6. Fiala, M.: 'ARTag', an improved marker system based on artoolkit (2004)
7. Wagner, D., Schmalstieg, D.: ARToolKitPlus for pose tracking on mobile devices. In: Proceedings of 12th Computer Vision Winter Workshop, pp. 139–146 (2007)
8. Kato, D.H.: ARToolKit. <http://www.hitl.washington.edu/artoolkit/>, [cited 8 Jan 2014]
9. Fiala, M.: Vision guided control of multiple robots. In: First Canadian Conference on Computer and Robot Vision, pp. 241–246 (2004)
10. Rekleitis, I., Meger, D., Dudek, G.: Simultaneous planning, localization, and mapping in a camera sensor network. *Robot. Auton. Syst.* **54**(11) (2006)
11. Stump, E., Kumar, V., Grocholsky, B., Shiroma, P.M.: Control for localization of targets using rangeonly sensors. *Int. J. Robot. Res.* (2009)
12. Fiala, M.: Comparing ARTag and ARtoolkit plus fiducial marker systems. In: Haptic Audio Visual Environments and their Applications, pp. 6–pp. IEEE (2005)
13. Bošnjak, M., Matko, D., Blažič, S.: Quadcopter hovering using position-estimation information from inertial sensors and a high-delay video system. *J. Intell. Robot. Syst.* **67**(1), 43–60 (2012)
14. ArUco: a minimal library for augmented reality applications based on opencv. <http://www.uco.es/investigacion/grupos/ava/node/26>. [cited 8 Jan 2014]
15. Ahn, S.J., Rauh, W., Recknagel, M.: Circular coded landmark for optical 3d-measurement and robot vision. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1128–1133. IEEE (1999)
16. Yang, S., Scherer, S., Zell, A.: An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **69**(1–4), 499–515 (2012)
17. Lo, D., Mendonça, P.R., Hopper, A., et al.: TRIP: A low-cost vision-based location system for ubiquitous computing. *Pers. Ubiquit. Comput.* **6**(3) (2002)
18. Pedre, S., Krajník, T., Todorovich, E., Borensztein, P.: Hardware/software co-design for real time embedded image processing: A case study. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 599–606. Springer (2012)
19. Kulich, M., et al.: Syrotek - distance teaching of mobile robotics. *IEEE Trans. Educ.* **56**(1), 18–23 (2013)
20. Pedre, S., Krajník, T., Todorovich, E., Borensztein, P.: Accelerating embedded image processing for real time: a case study. *J. Real-Time Image Process.* (2013)
21. Heikkilä, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1106–1112 (1997)
22. Yang, S., Scherer, S.A., Zell, A.: An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **69**(1–4), 499–515 (2013)
23. Krajník, T., Nitsche, M., Faigl, J.: The WhyCon system. <http://purl.org/robotics/whycon>, [cited 8 Jan 2014]
24. Faigl, J., Krajník, T., Chudoba, J., Přeučil, L., Saska, M.: Low-cost embedded system for relative localization in robotic swarms. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pp. 985–990, IEEE, Piscataway (2013)
25. Saska, M., Krajník, T., Přeučil, L.: Cooperative Micro UAV-UGV Autonomous Indoor Surveillance. In: International Multi-Conference on Systems, Signals and Devices, p. 36, IEEE, Piscataway (2012)
26. Kernbach, S., et al.: Symbiotic robot organisms: Replicator and symbion projects. In: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 62–69. ACM (2008)
27. Cajtler, V.: Syrotek localization system. Bachelor thesis, Dept. of Cybernetics, CTU (2013). In Czech
28. Krajník, T., Přeučil, L.: A simple visual navigation system with convergence property. In: Proceedings European Robotics Symposium (EUROS), pp. 283–292 (2008)
29. Krajník, T., Nitsche, M., Pedre, S., Přeučil, L., Mejail, M.: A Simple Visual Navigation System for an UAV. In: International Multi-Conference on Systems, Signals and Devices, p. 34, IEEE, Piscataway (2012)
30. Krajník, T., et al.: Simple, yet stable bearing-only navigation. *J. Field Robot.* **27**(5), 511–533 (2010)
31. Hawes, N.: STRANDS - Spatial-Temporal Representations and Activities for Cognitive Control in Long-Term Scenarios. <http://www.strands-project.eu>, [cited 8 Jan 2014]

# H

---

KEY ARTICLE [8] - IEEE ROBOTICS AND AUTOMATION LETTERS  
(RA-L)

---

©[2018] IEEE. This paper has been accepted for publication in the IEEE Robotics and Automation Letters (RA-L): V. Walter, N. Staub, A. Franchi, and **M. Saska**: UVDAR System for Visual Relative Localization with application to Leader-Follower Formations of Multirotor UAVs, 2019.



# UVDAR System for Visual Relative Localization with application to Leader-Follower Formations of Multirotor UAVs

Viktor Walter<sup>1</sup>, Nicolas Staub<sup>1</sup>, Antonio Franchi<sup>2</sup> and Martin Saska<sup>1</sup>

**Abstract**—A novel onboard relative localization method, based on ultraviolet light, used for real-time control of a leader-follower formation of multirotor UAVs is presented in this paper. A new smart sensor, UVDAR, is employed in an innovative way, which does not require communication and is extremely reliable in real-world conditions. This innovative sensing system exploits UV spectrum and provides relative position and yaw measurements independently of environment conditions such as changing illumination and presence of undesirable light sources and their reflections. The proposed approach exploits this retrieved information to steer the follower to a given 3D position and orientation relative to the leader, which may be considered as the main building block of any multi-UAV system operating with small mutual distances among team-members. The proposed solution was verified in demanding outdoor conditions, validating usage of UVDAR in real flight scenario and paving the way for further usage of UVDAR for practical multi-UAV formation deployments.

## I. INTRODUCTION

The growing interest in compact cooperative flights of Unmanned Aerial Vehicles (UAVs) [1] motivates an ongoing pursuit for efficient and embeddable onboard source of mutual relative localization.

In our previous work [2], we proposed a novel approach to tackle this issue, relying on vision in the unconventional ultraviolet spectrum. We named this new onboard sensor *UVDAR* for UltraViolet Direction And Ranging, and together with blinking ultraviolet markers used on its associated targets these comprise the *UVDAR system*. Its main advantages w.r.t. other solutions are twofold. First, the use of UV significantly increases robustness to challenges of outdoor environments regardless of the time of day, and second, its use of active markers allows for retrieval of orientation or identity of a target. The availability of such robust sensor is a prerequisite for decentralized outdoor formation flights and swarming and is especially crucial when a sufficiently precise absolute localization source is unavailable, or when it is unfeasible to prepare the necessary infrastructure [3], such as a motion-capture system (MoCap) or a base-station for *Real-time kinematic - Global Navigation Satellite System* (RTK-GNSS).

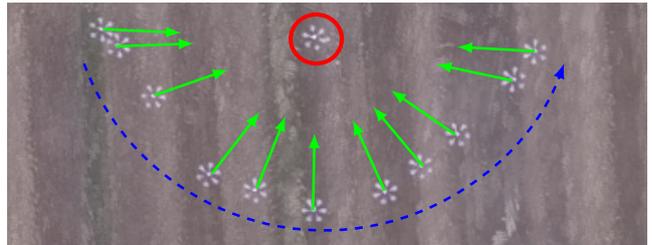


Fig. 1: Top view of a *directed leader-follower* experiment. The leader rotates by  $180^\circ$ . This is detected by UVDAR sensor carried by the follower, triggering it to create a trajectory as per the proposed algorithm in order to preserve its pose in the leader frame.

A typical example of multi-UAV flights is the leader-follower formation, consisting of two members, one following the other. Typically, the goal in such flight is for the follower to keep a constant distance from the leader, or to follow its trajectory [4]. Such following is applicable for various tasks such as cooperative mapping of historical buildings, cooperative carrying of objects or cooperative localization of a moving transmitter [3], [5]. In this paper, we show how the leader-follower approach has to be designed to be able to perform the required behavior using the UVDAR sensor. The presented *directed leader-follower* method, which leverages relative orientation information, can be considered as a guideline for designing complex multi-UAV systems working in real conditions with this sensor.

The literature on classical leader-follower formations is rich, see, *e.g.*, [4], [6], [7] for theoretical works backed by simulation. Works addressing the challenges of real experiments are limited, especially relying on onboard relative localization. The experimentally validated approaches often rely on either absolute localization source, *e.g.*, MoCap in [8], or RTK-GNSS in [9], [10]. As is known, MoCap is not practical for real-world deployment (neither outdoor or indoor) as it requires the installation of an expensive infrastructure. The absolute localization sources can provide full pose of the leader to the follower, which oversimplifies the problem. Even if only partial information is passed to the follower motion controller, like distance or bearing, this type of system provides continuous stream of such information without errors, which is unrepresentative of real-world deployment.

Some more practical approaches consider infrastructure-less sensing, like ranging based on radio signal [11], which only allows for distance-based following without any orientation information. Another approach [12], for the 2D case, wirelessly communicates the leader intents, which proves

<sup>1</sup> CTU in Prague, FEE, Department of Cybernetics, Czech Republic  
{viktor.walter|nicolas.staub}@fel.cvut.cz  
martin.saska@fel.cvut.cz

<sup>2</sup> LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France  
antonio.franchi@laas.fr

This research was partially supported by the ANR, Project ANR-17-CE33-0007 MuRoPhen, by CTU grant no. SGS17/187/OHK3/3T/13, by the Grant Agency of the Czech Republic under grant no. 17-16900Y and by project no. DG18P02OVV069 in programme NAKI II.



Fig. 2: Comparison between visible and UV camera footage from UVDAR, collected during the experiment. The UV image is significantly easier to process to retrieve UAV information.

feasible since there are less degrees of freedom and less drift than in the presented 3D case. These two approaches rely on radio transmissions, which is subject to the effects of network congestion and interference. This is why we consider vision-based approaches more suitable for the multi-robot groups, especially in uncontrolled outdoor environments.

This direction has been previously explored by the Multi-robot Systems (MRS) group at CTU-Prague, relying on true outdoor relative localization, see [13], [14]. The source of relative localization was an onboard vision-based system using passive circular markers as described in [15]. That came with drawbacks: high sensitivity to lighting conditions and partial occlusion, and substantial size for an acceptable detection range. A similar approach has been proposed [16], that extended the usability of passive markers for low light in short distances by leveraging the infra-red reflection. However, in all other respects it suffers from the same drawbacks as the visible passive marker approach. Furthermore, it was tested only for stable ground vehicles. This motivated the development of the UVDAR system, which is more robust to real-world conditions, due to optically filtering out visual information that is not of interest, reducing the computational load, see Fig. 2 for comparison with visible spectrum. UVDAR also provides relative orientation measurements and target identities, and the whole system is small and lightweight.

Our contribution is threefold. We first show how UVDAR can be used to obtain both the relative position and orientation. We then propose a *directed leader-follower* algorithm that works interactively with the UVDAR sensor and measurement method. Finally, we validate the performance of the presented method in outdoor experiments.

## II. POSE RETRIEVAL PRINCIPLE FOR UVDAR

The UVDAR sensor, presented in our previous work [2] [17], retrieves image positions and frequency-based IDs for individual blinking ultraviolet markers from a modified camera. This data is used to obtain the relative pose of the leader.

The blinking markers carried by the leader UAV have a known layout. We found that six markers arranged in a pattern of a regular hexagon pose as a good compromise, that ensures that at least two markers are visible from each direction and the markers not being too close to each

other. This means that they provide a source of a distance estimation without their images tending to merge in the operational distances. We instantiate this arrangement on a regular hexarotor platform with the markers attached to the ends of its arms, but the arrangement is easily reproduced for any similar rotorcraft, by *e.g.*, mounting the six markers onto a horizontal ring attached to the vehicle. If a different number of markers was to be used, the calculations used in this section need to be adjusted accordingly. In particular, if the arrangement will result in different number of markers being visible from different directions, each case needs separate equations according to the given geometry. The relative yaw is obtained by giving these markers two distinct IDs, retrievable by the UVDAR, one for the three port side markers and another for the three starboard ones. For other shapes of UAVs, different configurations may be preferable, see [17].

In this section we introduce the calculations necessary to retrieve the relative bearing, the mutual distance and the relative yaw when UVDAR system is used in conjunction with regular distribution hexarotors. The relative bearing is the direction towards the leader in the follower body-frame and the mutual distance is the distance between the geometrical centers of the two UAVs. The relative yaw is the angle between the horizontal components of their connecting line and the *tailing direction*, which is in our case the backwards direction in the leader body-frame.

Note, that for the mutual distance and relative yaw estimation we are assuming a horizontal alignment between the sensor and the target, since the height difference has negligible effect on the presented distance estimates and no effect on the relative yaw estimates

Two basic cases of the UVDAR output occur in practice, see Fig. 3, it either sees simultaneously two markers (case A, Fig. 3a) or three markers (case B, Fig. 3b). This depends on the relative yaw of the leader, because of the Lambertian radiation pattern of the markers, leading to two different calculations to retrieve the values of interest. In both cases the distance is first retrieved based on geometrical considerations and then the relative yaw based on marker IDs.

### A. Distance Retrieval – case A

The pixel coordinates  $\mathbf{m}_i$  of the origin points (the current expected image position of a blinking marker obtained by UVDAR, see [2]) is first translated into a 3D unit vector  $\mathbf{v}_i$  pointing towards the marker,

$$\mathbf{v}_i = c2w(\mathbf{m}_i), \quad (1)$$

where  $c2w(\cdot)$  is a standard function available in a number of vision libraries, such as the *OCamCalib* toolbox [18], provided that the camera has been properly geometrically calibrated.

As we consider only two markers, the angle their corresponding vectors form is denoted  $\alpha$  and obtained via cross-product,  $\alpha = \arccos(\mathbf{v}_1 \cdot \mathbf{v}_2)$ .

This angle is used to calculate the distance  $l_c$  between the target UAV geometrical center and the sensor, while

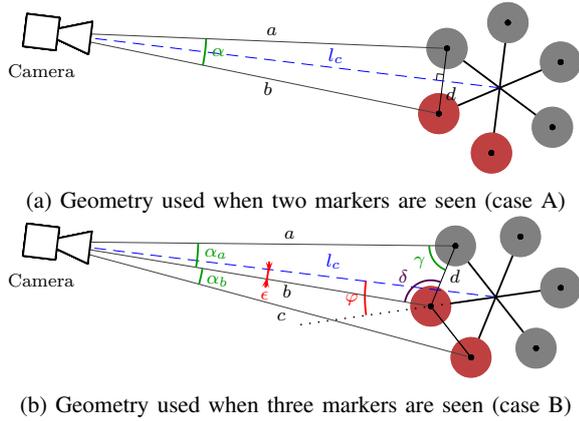


Fig. 3: The notation used in relative UAV pose estimation.

presuming that the line segment between the two markers is perpendicular to the line connecting the camera with its center point. This yields

$$l_c = \left(\frac{d}{2}\right) \cot\left(\frac{\alpha}{2}\right) + \sqrt{0.75d}, \quad (2)$$

where  $d$  is the length of the hexarotor arm, see Fig. 3a.

The relative bearing vector  $\mathbf{v}_c$ , is obtained through the conversion in (1) applied on the point in between the two origin points visible. The distance  $l_c$  and vector  $\mathbf{v}_c$  describe the relative position of the leader w.r.t. the UVDAR sensor.

Note that with only two markers visible, there is an ambiguity on  $\alpha$  arising from the simultaneous influence of distance and relative yaw (orthogonality assumption), which is explored in depth in [17]. This ambiguity disappears if three markers are visible.

### B. Distance Retrieval – case B

When three markers are visible, see Fig. 3b, the angles  $\alpha_a$  and  $\alpha_b$  are computed via (1) from two adjacent origin points. The distance  $l_c$  and angles  $\varphi$  and  $\epsilon$  are expressed as

$$l_c = \sqrt{b^2 + d^2 - 2bd \cos\left(\delta + \frac{\pi}{3}\right)} \quad (3)$$

$$\epsilon = \arcsin\left(\frac{d}{l_c} \sin\left(\delta + \frac{\pi}{3}\right)\right) \quad (4)$$

$$\varphi = \arcsin\left(\frac{b}{l_c} \sin\left(\delta + \frac{\pi}{3}\right)\right). \quad (5)$$

Here,  $b$  and  $\delta$  are common terms which correspond respectively to the distance from the sensor to the middle marker and the angle formed by the left marker, middle marker and the sensor. They can be expressed as follows

$$b = \frac{d \sin(\pi - (\delta + \alpha_a))}{\sin(\alpha_a)} \text{ and } \delta = 2 \arctan\left(\frac{P}{\sqrt{3B+3}}\right).$$

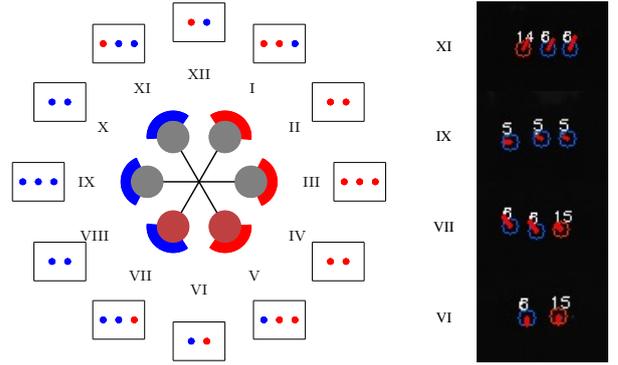


Fig. 4: Left – The layout of the two marker IDs on the considered hexarotor, denoted as blue (6 Hz) and red (15 Hz). Around it the frames illustrate the UVDAR view from their corresponding direction. On the right, the actual view from four different directions from experiments.

The last equation is a compact form of the analytical solution of a set of non-linear equations, where

$$\begin{aligned} A &= \cot(\alpha_a) \quad B = \cot(\alpha_b) \\ P &= B \left( 2\sqrt{\frac{O}{(B^2 + 2\sqrt{3} + 3)}} - 1 \right) \\ &\quad + \left( 6\sqrt{\frac{O}{(\sqrt{3}B + 3)^2}} \right) + (2A + \sqrt{3}) \\ O &= A^2 - AB + \sqrt{3}A + B^2 + \sqrt{3}B + 3. \end{aligned}$$

The relative position is then estimated from  $l_c$  and the relative bearing  $\mathbf{v}_2$  of the middle marker rotated by  $\epsilon$  in yaw.

### C. Relative Yaw Retrieval – case A

The relative yaw in camera frame  $\Delta\psi_c$  is retrieved from the IDs of the UVDAR markers. Only a finite number of patterns can be observed, as seen in Fig. 4, which we numbered from I to XII.

When only two origin points are seen, it corresponds to six possible relative orientations. If the two IDs differ, the leader is seen either from the front ( XII and  $\Delta\psi_c = \pi$ ) or from the back (VI and  $\Delta\psi_c = 0$ ), *i.e.*, the tailing direction. If the IDs are identical, the orientation is ambiguous (II–IV and VII–X). We resolve this with an heuristic, by averaging the two possible interpretations of such observation. Namely,  $\Delta\psi_c = \pm\pi/3$  and  $\pm 2\pi/3$ , so the average is  $\Delta\psi_c = \pi/2$  on starboard side or  $-\pi/2$  on port side. Note, that resolving the ambiguity based on previous observations is precluded by the ability of the target to independently change its rotation rate at any moment.

### D. Relative Yaw Retrieval – case B

When three origin points are seen, we consider the other six possible relative orientations, see Fig. 4. They correspond with relative orientation s.t. the follower is roughly facing one specific arm ( $\Delta\psi_e = \pm(\pi/6 + k(\pi/3))$ ):  $k \in \{0, 1, 2\}$ ).

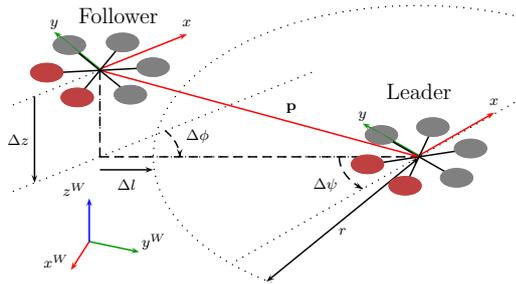


Fig. 5: Variables defining our requirements on the leader-follower system. Red propellers denote the back of the UAV. We want the leader to be on the circular perimeter with the radius  $r$  around the leader ( $\Delta l \rightarrow 0$ ) at the height of the leader ( $\Delta z \rightarrow 0$ ), while facing it ( $\Delta \phi \rightarrow 0$ ) from its back ( $\Delta \psi \rightarrow 0$ ).

Imperfect sensor alignment with the corresponding arm is accounted for by using the angle  $\varphi$  obtained from (5) as  $\Delta \psi_c = \Delta \psi_e - \varphi$ . Note that typically, for the considered operational distances greater than 5 m the origin points are very close in the UVDAR image, which means that  $\alpha_a$  and  $\alpha_b$  are almost identical, and  $\epsilon$  and  $\varphi$  had negligible effects.

In both cases, before the information retrieved from the UVDAR, *i.e.*, estimates of the relative position and yaw are used in our *directed leader-follower* they are transformed into the IMU frame of the follower UAV, which compensates not only for the offset of the sensor but also for the tilt (*i.e.*, roll and pitch) of the unit. Thus corrected relative yaw is denoted as  $\Delta \psi$ , and is used in the following section as *tailing error*.

### III. DIRECTED LEADER-FOLLOWER

The goal of *directed leader-follower* formation is to simultaneously regulate the mutual distance towards a pre-set *tailing distance* and to let the follower always face a given leader side (*tailing direction*), *e.g.*, its back. Our proposed algorithm solves such task and takes also into account constraints of vision-based sensing: 1) forward facing directional sensor and 2) sensitivity to rapid image motion. To address both, the follower behavior is such that it always attempts to face the leader. This guarantees that the leader is in the field of view (FoV) of the sensor and additionally that the leader image position will not change greatly over short periods of time. The requirements of the system are therefore to minimize the control errors illustrated in Fig. 5 as  $\Delta l$  - the distance error,  $\Delta z$  - the height error,  $\Delta \phi$  - the heading error and  $\Delta \psi$  - the tailing error. These are equal to zero if the follower is in what we call the *target pose* w.r.t. the leader. The heading error is the horizontal angle between the bearing of the leader and the frontal direction of the follower.

#### A. Trajectory Generation Strategies

The goal is to steer the follower to the target pose, located on the back of the leader, at a distance  $r$ , *tailing distance*, by which we also define a safety perimeter around the leader.

If the follower is steered only with the currently observed leader pose, changes in the observation lead to rapid changes

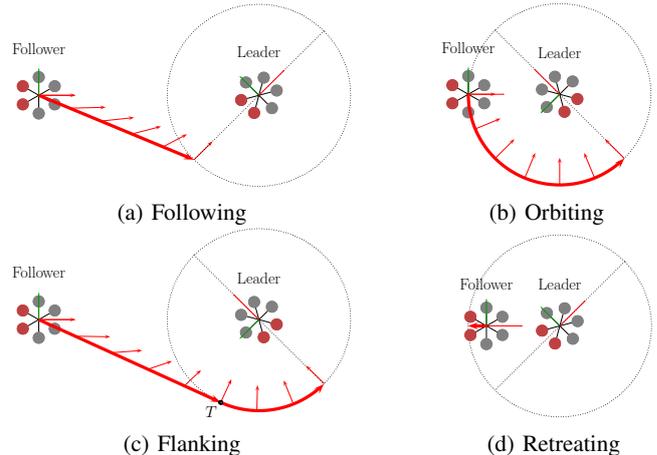


Fig. 6: The four strategies used in our directed leader-follower experiment. The red propellers denote the back of the UAV.

in the follower target pose. This is detrimental as we consider under-actuated platforms which have high coupling between their translational and rotational dynamics. Hence higher translational acceleration means higher tilting, likely to perturb the visual localization. To avoid this, we design the algorithm such that it repeatedly constructs a short-term trajectory, at fixed rate, whose time horizon is at most 4 s. The trajectory consists of isochronous points defined by their position and yaw. Trajectories are naturally constructed in the follower body-frame, if the follower is localized in the world-frame it is possible to convert them to world-frame to accommodate for low-level trajectory trackers.

The trajectory is constructed according to one of four distinct strategies; 1) following, 2) orbiting, 3) flanking, and 4) retreating, as depicted in Fig. 6.

For each strategy we consider the height error and the lateral position errors separately, as the height error does not play into the trajectory selection, and we attempt to bring it to zero as fast as possible in all four cases. We do this, by setting the  $z$  component of the whole generated trajectory directly to the  $z$  component of the estimated relative position of the leader, which forces our trajectory tracker to reach this height as fast as it can, bringing  $\Delta z$  close to zero.

The appropriate lateral strategy is selected based on the current situation which is described by the tailing error  $\Delta \psi$ , the tangential angle  $\beta = \arccos(r/l_c)$  and the distance error  $\Delta l$ , see Fig. 7. The decision map is as follows:

$$\Delta l < -h \quad \rightarrow \text{Retreating} \quad (6)$$

$$\Delta l \in [0, -h] \quad \rightarrow \text{Orbiting} \quad (7)$$

$$(\Delta l > 0) \wedge (|\Delta \psi| < |\beta|) \quad \rightarrow \text{Following} \quad (8)$$

$$(\Delta l > 0) \wedge (|\Delta \psi| \geq |\beta|) \quad \rightarrow \text{Flanking} \quad (9)$$

where  $h$  is a tolerance factor, introduced to prevent rapid switching in boundary cases by creating some hysteresis.

In the *following* strategy, Fig. 6a, the follower flies directly to the target pose on the perimeter at its maximum admissible horizontal speed.

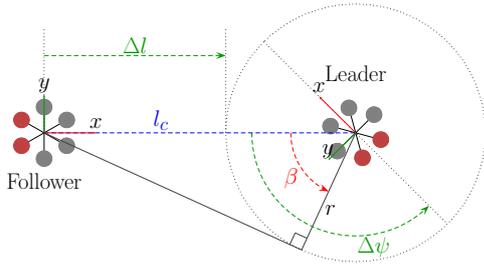


Fig. 7: Parameters of the observed situation for strategy selection.

In the *orbiting* strategy, Fig. 6b, as the follower is already at the tailing distance, it slides along the perimeter to reach the desired tailing direction.

In the *flanking* strategy, Fig. 6c, the follower flies straight to the tangent point  $T$  of the perimeter closer to the target pose, which brings the follower in the *orbiting* configuration.

In the *retreating* strategy, Fig. 6d, the follower is inside the safety perimeter of the leader, hence it is navigated outside of it radially, at its maximum admissible horizontal speed.

These strategies are devised to generate a fast path toward the target pose that is continuous up to the first derivative. Generation of each trajectory is based on the measured relative poses of the leader and the follower, from which estimates of the safety perimeter and the target pose lying on it are calculated. Depending on the current strategy, linear, circular or a combined trajectory is generated. The employed sampling of the trajectory ensures that the euclidean distance between two consecutive positions corresponds to the distance traversed at the maximum admissible speed during a single time-step, enforcing constant tangential speed. In order to accommodate for a leader motion, strategy selection and trajectory generation are triggered at a fixed rate.

Additionally, for each strategy we enforce that the follower yaw is such that its camera always faces the estimated leader position, considering the error  $\Delta\phi$ , by setting the reference yaw in each step of the trajectory to face the currently estimated leader position. This ensures continuous observation without rapid movements in the image as well as preventing loss of the leader from view in case of limited FoV, in our case  $180^\circ$  in the horizontal axis.

The generated trajectories are not accounting for real-world dynamics of the UAV and should be filtered before being sent to the low-level trajectory tracker. In our experimental setup, we leveraged the model predictive control present in our system [19], making the final trajectory smooth. This alters the original trajectory, but the optimization procedure used in [19] minimizes these differences, so that the resulting trajectory differs from the original only in four specific cases. Firstly, at the start of the trajectory after the leader was first discovered, the follower first accelerates to reach the desired tangential speed. This does not happen if the leader was already being tracked, since in such case the initial state already includes the tangential speed. Secondly, if the target pose is reached within a single trajectory generation period, the follower will decelerate, since abrupt stopping

is unfeasible. The third situation occurs during the transition between the linear and circular phase of the flanking strategy, when the trajectory is adjusted to achieve continuous acceleration. In this case the resulting trajectory resembles turns in automobile roads, eliminating step change in acceleration. This result is possible, because the flanking strategy contains both phases. An additional benefit is that if the next strategy is orbiting, after reaching the perimeter, the initial state will already include appropriate tangential speed so that the original trajectory will be followed with minimal change. Lastly, when retreating the trajectory is set such that the follower retreats according to its maximum speed, without regards to other conditions, which the model predictive control interprets by applying the maximum admissible acceleration. As the trajectory is re-generated asynchronously, following one of the four policies, the current state of the model is fused with the new trajectory to ensure a smooth transition. One useful addition for initialization of the leader-follower task or if the leader is lost, is setting the follower to slowly spin in place if it has not detected the leader yet, or has not seen it for pre-defined time.

### B. Constraints on the leader motion

In order for the follower not to lose the leader and to prevent collisions, the motion of the leader must conform to a set of restrictions.

The blinking signal retrieval in UVDAR limits the maximum component of the marker velocity perpendicular to the associated camera optical ray, in order to ensure consistent tracking. With our typical frame-rate of *approx.* 72 Hz, 23 frame signal sample and maximum allowed marker shift between frames of *approx.* 1 pixel, this limit is  $0.3 * l \text{ m s}^{-1}$ , scaling with the real distance  $l$  between the UAVs. The limit also defines the maximum yaw rotation rate of the leader, corresponding to *approx.*  $0.3 * l/d \text{ rad s}^{-1}$ . Additionally, linearity assumption in the UVDAR [2] limits the maximum acceleration of the leader in this direction to  $0.3 * l \text{ m s}^{-2}$ .

While tracking is unaffected by the component of the velocity along the camera optical rays, the distance measurement is less precise than the relative bearing. In particular, earlier experiments [17], showed that in rare conditions the distance measurement error could get close to 20%.

The distance estimate is important for the follower to successfully retreat from the leader in case of breached perimeter. This breach must be detected in time despite the distance estimate possible error. In the adverse case where both UAVs are flying directly towards each other, the follower at its following speed of  $v_{Fmax}$  and the leader at  $v_L$ , the follower registers a perimeter breach with delay. Additionally, this delay is extended by filtering the distance estimate with a moving average filter of time window  $t_a$ , when the detection is delayed by  $t_d = t_a/2$ . Note that the detriments of the moving average filter in this case are balanced by enhanced performance of the bearing estimate. The perimeter breach is detected at the distance

$$l_{brake} = (r - h) * 0.8 - r_{coll} - (v_L + v_{Fmax})t_d,$$

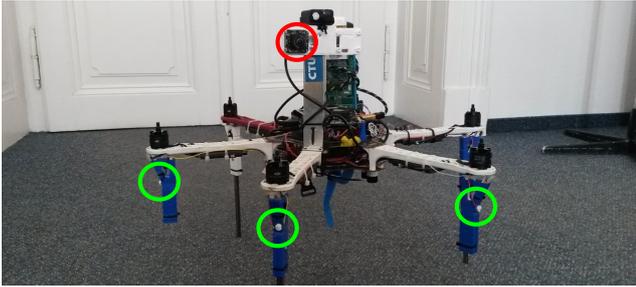


Fig. 8: The UAV platform used in our experiments, here equipped both with a ultraviolet camera (red) and active markers (green) comprising the hardware components of UVDAR system.

where  $r_{coll}$  is the collision distance. If  $a_{Fmax}$  is the maximum feasible acceleration of the follower in the case of retreating, then collision in the worst case can be avoided if

$$v_L < \sqrt{2l_{brake}a_{Fmax}} - v_{Fmax},$$

when  $l < r$ . In our experimental setting, this translates to limiting the approaching speed to  $v_L < 0.61 \text{ m s}^{-1}$  for distances smaller than  $r = 5 \text{ m}$ . A violation of this restriction is shown in the red zone of Fig. 10, resulting in the follower not retreating fast enough.

Evidently for greater safety, the leader should avoid approaching the follower. The rough direction in which the follower lies is implicitly known to the leader, since the follower is set to face a specific side of the leader. If the leader needs to fly in this direction, a simple way to prevent approaching the follower is to first rotate, ideally by  $90^\circ$ , and thus to steer the follower out of the way into a relative pose from which it can easily follow in a sideways manner.

Lastly, since the maximum distance for reliable detection by the UVDAR is  $15 \text{ m}$ , the leader, when it is further than  $12.5 \text{ m}$  from the follower, must not retreat from it faster than  $v_{Fmax} \text{ m s}^{-1}$ . This will ensure that the error of distance measurement will not lead to the follower losing the leader from sight. In most cases, the following algorithm already accounts for this, if the following distance is set to less than  $12.5 \text{ m}$  and enough time is provided for the follower to reach the target pose at the start of the mission.

## IV. OUTDOOR EXPERIMENTAL VALIDATION

### A. Experimental Platform

In order to validate the performances of the proposed formation algorithm, we conducted a campaign of real-world outdoor flights with two *DJI f550*-based hexarotors, see Fig. 8. They are each fitted with a *Intel NUC7* computer, a *PixHawk* flight controller and a *Tersus* GPS receiver, used with a RTK-GNSS system to obtain ground truth.

The two units were each equipped with a part of the UVDAR system. The leader was equipped with ultraviolet markers attached on the ends of its arms. The markers can be set with a blinking ID or not. In our setup two IDs are used as depicted in Fig. 4. Apart from providing IDs, blinking markers ensure robustness against reflections of the sun.

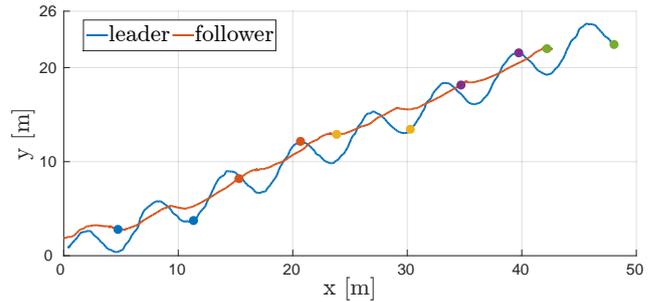


Fig. 9: Top view of the leader and follower trajectories in the preliminary experiment without marker identity. For a sense of time, selected simultaneous positions are marked with the same color.

The follower was equipped with a front facing UVDAR sensor, with a fisheye lens, allowing for  $180^\circ$  of horizontal FoV. Resolution and typical frame-rate are  $752 \times 480 \text{ pix}$  and  $72 \text{ Hz}$ , respectively. With the current UVDAR settings the detection range is around  $15 \text{ m}$ , see [17]. The relative positions and yaws of the leader are cyclically estimated at the rate of  $10 \text{ Hz}$ .

In order to increase the precision and to suppress the effect of any spurious errors of detection on the flight, we use a moving average filter of window 10, on the relative distance and relative yaw estimate. The relative bearing does not need filtering, as it is derived from the image position of the target, which we consider to be sufficiently reliable and precise.

During the experiments we noted that the UVDAR is highly sensitive to the lens focus. Indeed repeated manipulation altering the focus made the detection range drastically decrease to around  $6 \text{ m}$ , insufficient for practical purposes. Fortunately, focus can be monitored and adjusted easily.

For visualization, comparison and future analysis, the follower also carried a front-facing color camera. The views from the two onboard cameras are recorded a low frame-rate, so as not to impede the UVDAR.

### B. Preliminary Flight – Without Marker Identity

In a first set of experiments, we validated the UVDAR performances as a distance and relative bearing sensor, before going further. To do so, a simple leader-follower formation was tested. The markers of the leader were not blinking and the follower set to only approach the leader up to a desired tailing distance. This was implemented as a simple proportional position controller. Such behavior has been demonstrated with various other sensors. The distance and relative bearing are obtained as described in Sec II.

The leader tracked a waving trajectory retreating from the follower, see Fig. 9. The follower successfully managed to tail the leader during the whole trajectory, of length  $214 \text{ m}$ , demonstrating that UVDAR provides sufficient distance and relative bearing measuring capabilities for real-world flight.

### C. Real-World Flight of the Directed Leader-follower

Since the UVDAR is also able to provide a useful relative orientation estimate, we have conducted a second real-world flight where this information is used.

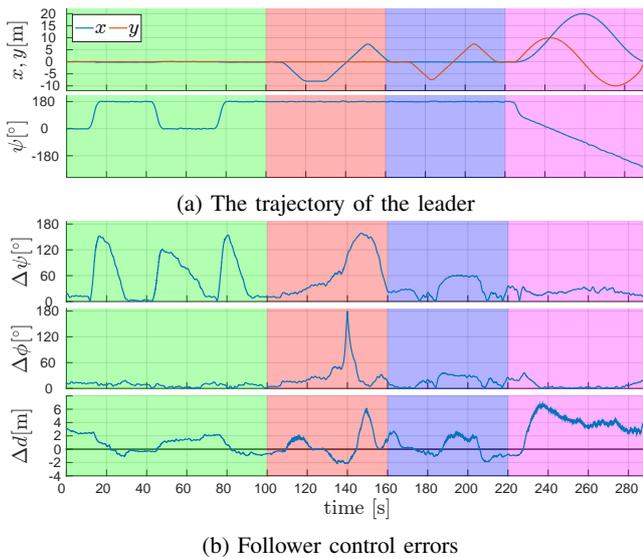


Fig. 10: (a) The leader trajectory defined by position and yaw  $\psi$ . The height was constant, set to 8 m. (b) From top to bottom - tailing error, heading error and distance error. Colored zones correspond to specific leader motions - rapid rotation (green), retreat-approach (red), left-right (blue), and circle following (purple).

The markers on the leader were set to blink, at 6 and 15 Hz following the pattern depicted in Fig. 4. The distance, relative bearing and orientation are obtained as described in Sec II, and used in the *directed leader-follower* from Sec. III. The tailing distance was set to  $r = 5$  m, with maximum hysteresis  $h = 2$  m. The temporary trajectories are generated at 2 Hz and their tangential speed is kept at  $2 \text{ m s}^{-1}$ .

Before the leader started its trajectory, it waited for the follower to reach its target pose. The leader trajectory was devised to highlight the system behavior in four representative cases. First, the leader makes three rapid rotations in yaw by  $180^\circ$  with 30 s of static hovering in between. Second, the leader moves linearly at  $0.8 \text{ m s}^{-1}$  with static yaw, going forward 8 m, backward 16 m and then forward 8 m again.

Third, the same retreat-approach motion was performed from left to right. The fourth case was a car-like following of a circular trajectory with a radius of 10 m. The height of the leader was fixed to 8 m for the whole experiment. The leader motion and follower control errors are plotted for the full trajectory, with the four cases, in Fig. 10a.

Video of the experiment can be seen online<sup>1</sup> and an external view of the experiment is shown in Fig. 11.

1) *Rapid rotation*: This highlights the importance and usage of relative orientation. Every time the follower detected a change in the leader orientation, it flew around the leader, see Fig. 1, to successfully reach the target pose again, demonstrating that relative yaw retrieval with UVDAR is reliable enough for real-world applications.

2) *Retreat-approach*: This can be seen as a classical leader-follower formation. The follower uses the relative position estimate to maintain a set distance from the leader.

<sup>1</sup><http://mrs.felk.cvut.cz/directed-following-with-uvdd>

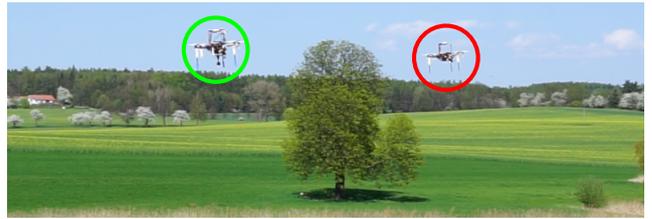


Fig. 11: View of *directed leader-follower* experiment, the leader (red) is retreating from the follower (green).

The observed performance is good overall. However, note that due to the granularity of the distance estimate from vision, combined with observation averaging, the reaction of the follower can be delayed, see Fig. 10b, around the 140 s mark. This engaged the follower collision avoidance mechanism, see [19], forcing it to fly over the follower and then to resume directed following by turning around and orbiting, which is the origin of the observed peak in  $\Delta\psi$ . This demonstrates that the good following performance can be jeopardized if the leader flies towards the follower faster than the admissible limit of  $0.61 \text{ m s}^{-1}$  estimated in III-B, since in this case we set the speed of approach to  $0.8 \text{ m s}^{-1}$ . A larger perimeter can be set to mitigate this, trading off visual distance estimation precision, so a compromise needs to be found for each application.

3) *Left-right*: When the leader moves side to side, both the relative distance and orientation estimates performances are evaluated. As depicted in Fig. 10b the performances are good as the max. relative yaw was around  $60^\circ$  and the distance error around 2 m, demonstrating that the follower was able to deal with a continuous disturbance in both quantities simultaneously.

4) *Circular following*: The last part of the trajectory demonstrated the ability of our system to follow a leader along an extended trajectory, by tailing a leader flying along a circular trajectory in a car-like manner. As the plot implies, while the follower lagged behind the tailing distance by 4 m on average, it did not lose track of the leader for the whole trajectory, both in terms of relative yaw and heading.

## V. POTENTIAL FUTURE EXTENSIONS

The system performances in the experiments validated our approach and more importantly pave the way to a wider use of UVDAR for multi-UAV relative localization.

Other formations can be explored and tested, such as train-like formation where multiple units are following another one in front of them. Thanks to the marker IDs provided by the UVDAR, that can map to leader identities, keeping the leader-follower order should prove easy. Such formation needs to guarantee that leader motion can not force followers further down the line to reach speed or acceleration limits.

Additional studies on leveraging the marker layout are necessary. The current layout, with adjacent triplets of same-ID markers has the drawback that for a pair of same-ID (Fig. 4-II,IV,VIII,X), the relative yaw is ambiguous, *e.g.*, II

and IV appear identical. In Sec. II-D we used a heuristic, averaging the two possible interpretations. Another available option is presuming one of the two interpretations based on which leads to the more favorable dynamics. With the current layout, leader starboard and port directions (III and IX) can not be chosen as alternative tailing directions since they are surrounded by ambiguous observations.

However, with the current layout it is possible to steer a follower to any of the other alternative tailing directions that can be uniquely located (Fig. 4-V,VI,VII,XII,XI,I). This allows for multiple directed followers for a single leader, separated by different tailing directions. A simpler way to allow for more followers is to assign them to different relative heights, although the aerodynamic coupling between the followers must be taken into account in that case.

For steering towards one of the unique positions the current layout was sufficient, but for truly arbitrary static formations a third identity must be introduced, using up more of the limited number of available IDs.

Filtering distance estimation with simple averaging proved to be a weak point, imposing strong motion restriction on the leader, and more advanced filtering techniques, such as a Kalman filter, should be considered in the future.

## VI. CONCLUSION

In this paper, we demonstrated the applicability of our novel vision-based relative localization system UVDAR for cooperative UAV flights on a specific implementation of the leader-follower formation. This *directed leader-follower* formation control exploited the relative leader pose obtained by the UVDAR sensor, comprising position and yaw, to steer the follower to a *target pose* pose w.r.t. the body of a moving leader, while also preserving the conditions for continued observation by this vision system. The cooperative combination of UVDAR with a specialized control algorithm was shown to maintain the desired following behavior, without direct communication between the two UAVs.

The encouraging performance of the system for various motions is shown through outdoor experiments. In particular, the use of UVDAR for a real application is demonstrated for the first time, in demanding outdoor situations. More complex formations have to be addressed in future work. Overall, the UVDAR performance in outdoor conditions should lead to its wider adoption.

## REFERENCES

- [1] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, Aug 2018.
- [2] V. Walter, N. Staub, M. Saska, and A. Franchi, "Mutual localization of uavs based on blinking ultraviolet markers and 3d time-position hough transform," in *CASE 2018*, 2018.
- [3] M. Saska, V. Kratký, V. Spurný, and T. Bába, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control." in *ETFA*. IEEE, 2017.
- [4] Y. Zou and Z. Meng, "Leader-follower formation control of multiple vertical takeoff and landing uavs: Distributed estimator design and accurate trajectory tracking." in *ICCA 2017*, July 2017, pp. 764–769.
- [5] M. Saska, "Large sensors with adaptive shape realised by selfstabilised compact groups of micro aerial vehicles," in *International Symposium on Robotic Research*, 2017.
- [6] A. S. Brando, I. H. B. Pizetta, M. Sarcinelli-Filho, and R. Carelli, "High-level nonlinear underactuated controller for a leader-follower formation involving a miniature helicopter and a ground robot," in *(SBR-LARS 2012)*, Oct 2012, pp. 168–173.
- [7] J. Choi and Y. Kim, "Fuel efficient three dimensional controller for leader-follower uav formation flight," in *(ICCAS 2007)*, Oct 2007, pp. 806–811.
- [8] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *(ECC 2013)*, July 2013, pp. 3858–3863.
- [9] S. Montenegro, Q. Ali, and N. Gageik, "A review on distributed control of cooperating mini uavs," 2015.
- [10] J. Vilca, L. Adouane, and Y. Mezouar, "Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration," in *Distributed Autonomous Robotic Systems*, 2016, pp. 237–254.
- [11] T. Chen, Q. Gao, and M. Guo, "An improved multiple uavs cooperative flight algorithm based on leader follower strategy," in *(CCDSC)*, June 2018, pp. 165–169.
- [12] T. Bhavana, M. Nithya, and M. Rajesh, "Leader-follower co-ordination of multiple robots with obstacle avoidance," in *(SmartTechCon 2017)*, Aug 2017, pp. 613–617.
- [13] M. Saska, T. Bába, J. Thomas, J. Chudoba, L. Preucil, T. Krajník, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, no. 4, pp. 919–944, 2017.
- [14] V. Spurný, T. Bába, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *(MMAR 2016)*. IEEE, 2016, pp. 998–1003.
- [15] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.
- [16] H. Park, I. Choi, S. Park, and J. Choi, "Leader-follower formation control using infrared camera with reflective tag," in *(URAI 2013)*, Oct 2013, pp. 321–324.
- [17] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of uavs using ultraviolet led markers," in *(ICUAS 2018)*, 2018.
- [18] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *(ICVS'06)*, 2006.
- [19] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *(IROS 2018)*. IEEE, 2018.

---

KEY ARTICLE [9] - IEEE ETFA 2017

---

©[2018] IEEE. This paper is available in proceedings of the 22nd International Conference on Emerging Technologies and Factory Automation: **M. Saska**, V. Kratky, V. Spurny, and T. Baca: Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control, 2017. *An extended journal version of the paper is under preparation and expected to be submitted into IEEE Robotics and Automation Magazine in 2019.*



# Documentation of Dark Areas of Large Historical Buildings by a Formation of Unmanned Aerial Vehicles using Model Predictive Control

Martin Saska and Vít Krátký and Vojtěch Spurný and Tomáš Báča  
Department of Cybernetics, Faculty of Electrical Engineering  
Czech Technical University in Prague  
email: martin.saska@fel.cvut.cz, tel: +420224357634

**Abstract**—A system designed for a unique multi-robot application of closely flying formations of Unmanned Aerial Vehicles (UAVs) in indoor areas is described in this paper. The proposed solution is aimed as a tool for historians and restorers working in large historical buildings such as churches to provide an access to areas that are difficult to reach by humans. In these objects, it is impossible to keep a large scaffolding for a long time due to regular services, which is necessary for studying a long-term influence of restorations works, and some parts of the churches were even not reached by people for decades and need to be inspected. To provide the same documentation and inspection techniques that are used by the experts in lower easily accessible parts of the buildings, we employ a formation of autonomous UAVs, where one of the robots is equipped by a visual sensor and the others by source of light, which provides the required flexibility for control of lightening.

The described system in its full complexity has been implemented with achieved robustness and reliability required by deployment in real missions. The technology demonstration has been provided with real UAVs in historical objects to help restorers and conservationists with achieved valuable results used in plans of restoration works. In these missions, UAVs were autonomously hovering at designated locations to be able to demonstrate usefulness of such robotic lightening approach.

## I. INTRODUCTION

A system for autonomous documentation of dark areas in large historical buildings by a formation of Unmanned Aerial Vehicles (UAVs) is presented in this paper. In the proposed approach, a self-stabilized formation of multi-rotor helicopters is employed for filming and visual inspection in dark conditions, where one of the UAVs carries a camera and neighboring UAVs a source of light. This setup aims to fully autonomously realize two techniques often used by historians and restorers for manual inspection of interiors of historical monuments nowadays. The first one, so-called *Three point lighting* approach [1], [2], is a filming technique in which 1-3 sources of light are used in different locations relatively to the camera optical axis. The method enables to create an illusion of a three-dimensional subject in a two-dimensional image and to illuminate the subject being shot (such as sculptures in historical buildings) while controlling the shading and shadows produced by lighting. This is essential for the presentation of historical monuments in interiors to the broad public, as it

removes the boring flatness from images and videos, and it adds a value to the analysis of gathered results by historians.

The second technique frequently used by restorers employs a strong side-light for illumination of flat objects, such as walls with parget and mosaics. In this method, the strong light needs to be placed as close as possible to the scanned plain, which makes visible shadows in the image in a case of a roughness of the surface. Restorers and conservationists can detect from such illuminated pictures if a tile in the mosaic is not fixed properly or if a painting is affected by a humidity indicated by buckling of the wall surface.

We implemented these two methods using autonomous unmanned helicopters, which enables their usage in places in the interior of large historical objects (e.g. churches) that are not accessible without installation of a costly staging. In cooperation with restorers and filmmakers, we defined the filming *Three point lighting* technique and the *Strong side-lightening* method as a multi-objective optimization problem in the Model Predictive Control (MPC) framework. The MPC approach is used for control of the formation members taking into account task objectives as well as constraints of the formation flying (obstacle avoidance, mutual collision avoidance), low level UAV stabilization [3] (motion constraints), filming (limited camera field of view, keeping all UAVs out of the taken images), and illumination (providing intensity of the lightening in a required range, keeping the recommended angle between the light and camera axes).



Fig. 1. Demonstration of the presented formation flying approach to realize a flying “film crew” in a church in Sternberk, Moravia.

It is not our intention to design a system that would decide where and what to film or scan. This is a job of the filmmakers and experts from the field of restoring and historical sciences. To be able to reach the direct usability of the system by the experts, we employ the artificial intelligence and the autonomy of UAVs only for solving technical tasks, such as the lightening, and the artistic intent is left at hands of the professionals. Also the decision, which part of the church needs to be inspected and which technique has to be used is still taken by historians and the proposed system is designed only as a smart tool to facilitate their job. Therefore, we assume an input from these experts for the system in a form of a given path that has to be followed by the UAV with a camera and a given set of objects that have to be filmed. From the safety reason, we expect a given map and known positions of the objects of interest in the building. Although, online mapping techniques are provided by robotic community nowadays, they lack 100% reliability, and also filmmakers as well as historians require visualisation of the planned path prior the mission to make sure that their intention will be fulfilled for which the map given prior the mission is essential. In addition, the revision of the overall plan increases probability of mission failure, which is crucial for deployment of autonomous systems flying close to objects with high historic and financial costs. Fortunately, in the presented application scenario, the knowledge of a precise map prior the mission is a realistic assumption since the precise map is always used by the historians and it is a main part of the restoration and historical survey for which these techniques are targeted.

In practice, the experts denote which part of the path has to be dedicated to which object of interest (OoI) and the camera orientation is simply computed for each point of the path using the known position of the OoI. The obtained profile of the camera heading does not need to be smooth and feasible since it is considered only as one of the desired objectives of the MPC optimization. Usually, the task requirements do not respect the UAV movement constraints, mainly if the input for the system is provided by the historians. The film lighting technicians have usually better understanding of motion constraints of a camera doing a dynamic shot, but still their demands are often in contrast with limitations of UAVs and self-localized formations. Therefore, the proposed system is designed to find a solution close to the usually infeasible (and even non-continuous) desired trajectory for the leader and the camera orientation, and to find a compromise between the requirements given by the experts and the artificial system of closely cooperating UAVs. Besides, we expect that the positions of the objects of interest, obstacles and neighbouring robots may not be known absolutely precisely, and the planning and control of the multi-robot team is then driven by actual results of UAV perception (obstacle detection and object recognition). In the current implementation, the solution is prepared for using the system of neighboring objects/UAVs localization that we have described in [4], [5], but any state-of-the-art implementation of proper obstacle detection and object recognition techniques can be used.

#### A. Literature review

Autonomous systems used for documentation of heritage sites are mostly employed for 3D modelling of exteriors and interiors of objects by laser scanners or by the photogrammetry, which is being popular recently as it allows to use cheap and lightweight monocular cameras for 3D object reconstruction. Numerous techniques exist to facilitate and speed up the scanning process by increasing its automatization. Static terrestrial laser scanners were used in [6], [7] for 3D modelling of historical sites. Using these techniques, the scanners have to be placed into predefined geo-referenced locations and the obtained 3D point cloud is fused using the known positions of different measurements. In [8], a long-range 3D laser scanner was used to scan large historical sides and to provide an inter-relationship between outdoor and indoor profiles using a technique for storing and processing big amount of data. A faster scanning process is allowed by a hand-held mobile mapping system called Zebedee [9] that was designed for semi-autonomous gathering of 3D point cloud models in cultural heritage applications. Another level of autonomy has been added using a ground mobile robot carrying the laser scanner that enables 3D modelling of large-scale environments with minimal human intervention [10].

Applying UAVs provides a possibility to acquire data from measurement locations inaccessible by ground robots or hand-held systems in addition to the autonomy provided by UGVs. Nowadays, UAVs are deployed mainly for 3D modelling of outdoor archaeological locations by the photogrammetry of geo-referenced images and historical buildings by lightweight lasers [11], [12], [13], [14]. All these techniques, which are offered by numerous private companies, use GPS data to reference the images and to initialize the 3D cloud construction process by this information. Only a limited number of approaches allows 3D modelling by UAVs in GPS-denied environment inside historical buildings, although indoor localization and 3D mapping techniques are well investigated by robotic community [15], [16]. We have found only the work in [17] that uses UAVs in the context of documentation of interiors of historical buildings. The method is based on a state-of-the-art online visual simultaneous localization and mapping and an offline visual structure-from-motion method in order to obtain the 3D model of the building.

Our method goes much further than [17] and it aims to fully exploit the abilities of UAVs, which is mainly flying in places hardly accessible by people and other types of robots. It was identified during the first deployment of the system in large churches that in these hardly accessible locations light conditions are often not sufficient and external light sources are required. The proposed system enables to set direction of the light sources dependently on the position of the camera, which is a very useful tool being used by restorations and filmmakers. This is realised by a leader-follower formation flying technique providing the *Three point lighting* and *Strong side-lightening* approaches in an autonomous way.

The proposed formation stabilization and navigation algo-

algorithm, which arises from our theoretical work on formation stabilization of ground robots [18], [19], [20] and UAVs [21], [22], [23], [24], consists of two main components. In the first one, an optimization-based model predictive control mechanism is employed for the formation leader, which is the UAV carrying the camera. A sequence of control inputs in the receding horizon fashion is provided for the leader in two optimization steps. Firstly, a multi-objective optimization is used to control the 3D position of the robot along the predefined path. Deviation from the desired path that is given by the restorers, too aggressive control inputs (changing velocity and angular acceleration), and too small distance from the obstacles and the objects of interest are penalised in this optimization. The second optimization step of the planning method proposed for the leader is aimed to provide a smooth and feasible profile of the camera orientation. Motion planning and stabilization of followers are also realized under the model predictive control scheme in two optimization processes to control their position within the formation and to control orientation of their light sources. In each control step, a new control sequence is obtained as a result of the multi-objective optimization procedures, which is applied in the receding horizon fashion. It means that in each step, only a small portion of the plan is sent to the actuators and for the consequent re-planning a new piece of the same size is added at the end of the plan.

## II. PRELIMINARIES

### A. Task definition and assumptions

The multi-robot task solved in this paper is to realize the *Three point lighting* and *Strong side-lightening* approaches in indoor GPS-denied environment by a team of cooperating UAVs (multi-rotor helicopters). In this scenario, one of the UAV (the leader) is equipped with a calibrated camera and  $n_F$  UAVs (the followers) with light sources. Parameters of the camera and light sources (orientation, field of view, light intensity, dispersion, etc.) are known prior the mission.

In the description of the proposed method, we suppose a known map of the environment (a contour of walls) with a set of convex obstacles  $O$ . Although no dynamic obstacles usually occur in the real deployment of our approach for documentation of historical buildings, the method in general is able to deal with a dynamic environment and with an unknown or partial known map as we have shown in a theoretical work in [25]. We also suppose that each UAV is equipped with a system providing its localization in the given map and relatively to its neighbours. For the real deployment in GPS-denied environment, we propose to rely on a fusion (the mechanism of the fusion is described in [26]) of an odometry obtained from an optical flow in images captured by a down-looking camera and outputs of visual relative localization of neighbours (for description of the employed localization method see [4], [5]). In the case of the *Strong side-lightening* technique, we assume to have a precise information on the distance to the scanned wall. A communication link between the robots is assumed during the mission for distribution of the

actual leader's plan to followers, which is used for planning their optimal positions and orientations of the light sources in the future.

As mentioned, our intention is not to contribute to the art and to jobs of the restorers and we suppose that a desired path for the leader is given by these experts. In addition, we suppose that the path is labelled by an information which OoI has to be observed in which part of the path. Using the information on known positions of OoI, the desired orientation of the camera may be simply derived. The desired path and camera orientation given by the experts, later denoted by the  $(\cdot)^{ex}$  symbol, does not need to be feasible with respect of kinematics of UAVs.

Let us note that the proposed method is capable of working with unknown a priori knowledge of positions of obstacles and objects of interest (they can be detected by state-of-the-art methods such as the one we proposed in [5]) and work in a full perception driven mode. Also the given desired path is not necessary for the proposed formation flying method and a trajectory planning method can be integrated as in our previous work in [21]. Nevertheless, such a higher level of autonomy is not required in this application and it adds an additional source of uncertainties, which would decrease robustness and so the applicability of the method.

### B. Model predictive control on a receding horizon

MPC-based approach is used in the proposed system to independently control the position of the leader as well as the followers in the formation, but also to control the orientation of the carried camera and the light sources to enable stability analyses of the complete system. In the MPC scheme, the control task is transformed into an optimization with constraints imposed by a model of the controlled system. We will use the kinematic UAV model from [21] for both, the leader and the followers. The optimization solves a finite horizon optimization control problem starting from current state over a control horizon. The result of the MPC method is then an optimal trajectory defined by  $N$  transition points and constant control inputs between them. The optimization is initialized in each control step based on the solution obtained in the previous MPC step. In each MPC step, once a solution of the optimization problem is obtained, only the first  $n$  vectors of the computed control inputs are applied to control the system. The optimization process is then repeated on a new interval as the finite horizon moves by the time horizon with length  $n\Delta t$ , where  $\Delta t$  is the time difference between two subsequent transition points.

## III. MOTION PLANNING AND FORMATION STABILIZATION SYSTEM

In this section, a formation flying method based on the MPC framework employed for control and navigation of an autonomous team of a cameraman and illuminators will be described. In the method description, the more general setup of a camera and lights mounted on a gimbal is used. This enables to control the Yaw angle independently to the task, which

enables a very important safety mechanism. If all UAVs are oriented in the direction of their movement along the desired trajectories (independently to positions of the OoI and so the camera/gimbal orientation), it is much simpler to manually overtake their control in a case of a system malfunction. This is important mainly during the initial debugging of the methods and for testing flights preceding each practical deployment of the system.

For the method description, let us denote the UAV position in the cartesian coordinates as  $p_j(k) := \{x_j(k), y_j(k), z_j(k)\}$ , its heading  $\theta_j(k)$ , and the camera heading  $\varphi_j(k)$ , where  $k \in \{1, \dots, N\}$  denotes indexes of transition points in the MPC control horizon. For MPC techniques, let us use control inputs  $U_j(k) := \{v_j(k), w_j(k), K_j(k)\}$ ,  $k \in \{1, \dots, N\}$ , where  $v_j(k)$  is forward velocity,  $w_j(k)$  ascent velocity, and  $K_j(k)$  curvature of the trajectory, for control of the  $j$ -th UAV and angular velocity  $\omega_j(k)$  for control of its gimbal with the camera. These control inputs are constant in-between of the transition points and lead the UAV from one transition point into a following one applying the model from [21]. This notation is used for the leader as well as the followers in the formation, such that  $j \in \{L, 1, \dots, n_F\}$ .

#### A. Trajectory tracking of the leader with camera

As mentioned, two control problems are solved independently (control of UAV states  $\psi_L(k) := \{p_L(k), \theta_L(k)\}$ ,  $k \in \{1, \dots, N\}$ , and camera headings  $\varphi_L(k)$ ,  $k \in \{1, \dots, N\}$ ) to reduce the overall computational complexity in the proposed method.

Using the MPC methodology, we can formulate the first problem as an optimization of a vector  $\Omega_{L,1} = [U_L(k)] \in \mathbb{R}^{3N}$ ,  $k \in \{1, \dots, N\}$ . The objective function  $J_{L,1}$ , which is minimised and subject to a set of inequality constraints, is designed as

$$\begin{aligned}
J_{L,1}(\Omega_{L,1}) &= \frac{\alpha}{N} \sum_{k=1}^N \|p_L^{ex}(k) - p_L(k)\|^2 \\
&+ \sum_{k=1}^N \frac{\beta_1}{N} K_L^{diff}(k)^2 + \frac{\beta_2}{N} v_L^{diff}(k)^2 + \frac{\beta_3}{N} w_L^{diff}(k)^2 \\
&+ \frac{\gamma}{N} \sum_{k=1}^N \left( \min \left\{ 0, k \frac{\min_{j \in \{1 \dots n_F\}} \text{dist}(p_L(k), \Omega_{j,1}^*) - r_s}{\min_{j \in \{1 \dots n_F\}} \text{dist}(p_L(k), \Omega_{j,1}^*) - r_a} \right\} \right)^2 \\
&+ \frac{\delta}{N} \sum_{k=1}^N \left( \min \left\{ 0, k \frac{\text{dist}(p_L(k), O) - r_s}{\text{dist}(p_L(k), O) - r_a} \right\} \right)^2 \\
&+ \frac{\varepsilon}{N} \sum_{k=1}^N (\min \{0, \|p_{OoI}^{ex}(p_L^{ex}(k))) - p_L^{ex}(k)\| \\
&\quad - \|p_{OoI}^{ex}(p_L^{ex}(k))) - p_L(k)\|\})^2. \tag{1}
\end{aligned}$$

The first component of the cost function penalises deviations of the leader's position  $p_L(k)$  along the planning interval  $N$  from the desired position on the path given by the experts  $p_L^{ex}(k)$  as one of the inputs of the method. The second

component is employed to suppress big deviations in control inputs (and so too aggressive behaviour of the system), where the notation  $(\cdot)^{diff}(k) := (\cdot)(k-1) - (\cdot)(k)$  represents the difference of the particular control parameter from its previous value. The control inputs at  $k=0$  represent values that are implemented by controllers at the time of initialization of the optimization process. The third term ensures avoidance of other robots in the formation that are considered as dynamic obstacles for the leader. Function  $\text{dist}(p_L(k), \Omega_{j,1}^*)$  returns the shortest distance between the position of the leader being optimized and the plan of the  $j$ -th follower obtained in the last call of the optimization process described in section III-B (let us denote results of the optimization processes by the notation  $(\cdot)^*$ ). Constant  $r_s$  is radius of a safety area around the robot in which the obstacles are considered in the avoidance function and  $r_a$  is radius of an avoidance area. The value of the third term approaches infinity if distance to an obstacle equal to  $r_a$  is reached. The fourth component, which protects the leader from collisions with static obstacles, uses the same avoidance function. Also the influence of this term is increased for an obstacles detected at the end of the control horizon (based on index  $k$  of the position  $p_L(k)$  that contributes to the cost function), which ensures the obstacle avoidance functionality and protects the system from oscillations in a close proximity to obstacles. Function  $\text{dist}(p_L(k), O)$  returns the shortest distance between the position of the leader and the obstacles. The fifth term is important due to the filming task as it penalises trajectories that are in a bigger distance to the position of the object of interest  $p_{OoI}^{ex}(p_L^{ex}(k))$  in case of a deviation of the leader from the desired path by an obstacle. Avoiding an obstacle in a free space closer to the OoI, which is selected by the experts as the one that has to be observed by the follower at position  $p_L^{ex}(k)$ , decreases probability that the obstacle appears in the camera field of view.

The values of coefficients  $\alpha$ ,  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ ,  $\gamma$ ,  $\delta$ , and  $\varepsilon$  in equation (1) influence behaviour of the system between often antagonistic requirements. For example with increasing values of  $\beta(\cdot)$ , a smoother and less aggressive flight performance will be provided, while with increasing values of  $\gamma$  and  $\delta$  the obstacle avoidance ability of the system is preferred.

The inequality constraints  $r_a - \text{dist}(p_L(k), O) \leq 0$ ,  $k \in \{1, \dots, N\}$ , and  $r_a - \min_j \{\text{dist}(\Omega_{L,1}, \Omega_{j,1}^*)\} \leq 0$ ,  $j \in \{1, \dots, n_F\}$ , which have to be ensured in the optimization, support the avoidance function to ensure that a plan of the leader does not touch an obstacle by its avoidance region. In addition, control inputs in the optimization vector are limited by the following inequality constraints  $v_L(k) - v_{L,max} \leq 0$ ,  $v_{L,min} - v_L(k) \leq 0$ ,  $w_L(k) - w_{L,max} \leq 0$ ,  $w_{L,min} - w_L(k) \leq 0$ ,  $K_L(k) - K_{L,max} \leq 0$ ,  $K_{L,min} - K_L(k) \leq 0$ , where  $k \in \{1, \dots, N\}$ , including upper and lower bounds of the UAV controllers.

The second optimization problem is employed to obtain a smooth and feasible control of the camera heading (instructions for the gimbal or UAV controller in the case of the fixed mounting). Let us define the optimization vector in this

problem as  $\Omega_{L,2} = [\omega_L(k)] \in \mathbb{R}^N$ ,  $k \in \{1, \dots, N\}$ , and the cost function, which is again minimised in the optimization, as

$$J_{L,2}(\Omega_{L,2}) = \zeta \sum_{k=1}^N \text{diffang}(\varphi_L^{ex}(k), \varphi_L(k))^2 + \eta \sum_{k=1}^N \omega_L^{diff}(k)^2, \quad (2)$$

where  $\varphi_L^{ex}(k)$  is the desired camera heading at  $k$ -th transition point. The heading is obtained from the position of the object of interest  $p_{OoI}^{ex}(p_L^{ex}(k))$  that has to be filmed/scanned at this moment, and the position of the leader at  $k$ -th transition point of the plan  $\Omega_{L,1}^*$  obtained as a solution of the first optimization problem. Although, the heading  $\varphi_L^{ex}(k)$  is not explicitly given by the experts, we used the  $(\cdot)^{ex}$  notation here, since they provide the desired position of the leader that should be as close as possible to the planned one and also the position of the OoI. The desired heading  $\varphi_L^{ex}(k)$  should follow their intention as the orientation of the camera is naturally one of the most important factors influencing the required result in both filming techniques being solved in this paper.

The function  $\text{diffang}(\psi_1, \psi_2)$  is defined for  $\psi_1, \psi_2 \in (0, 2\pi)$  as

$$\text{diffang}(\psi_1, \psi_2) = \begin{cases} \psi_1 - \psi_2 & \text{if } \psi_1 - \psi_2 \leq \pi, \\ 2\pi - (\psi_1 - \psi_2) & \text{if } \psi_1 - \psi_2 > \pi. \end{cases} \quad (3)$$

The constants  $\zeta$  and  $\eta$  set influence of the objective trying to follow the desired camera orientation (the first term of the multi-objective function in eq. (2)) and the objective achieving not aggressive change of the orientation (the second term), respectively. The only constrain functions that have to be satisfied for optimization of the orientation are used to limit the UAV angular velocity as  $\omega_L(k) - \omega_{L,max} \leq 0$  and  $\omega_{L,min} - \omega_L(k) \leq 0$ ,  $\forall k \in \{1, \dots, N\}$ . Although the camera heading control could be done by any simpler controller combined together with an interpolation of desired heading values, we rely on MPC to be able to synchronise it with the position control under the recoding horizon.

### B. Trajectory tracking for followers with the light source

The desired positions of the  $n_F$  followers  $p_{j,d}(k)$ ,  $j \in \{1, \dots, n_F\}$ ,  $k \in \{1, \dots, N\}$ , in the formation are obtained from the positions of the leader (derived from its actual plan - the last result  $\Omega_{L,1}^*$  of the MPC position control) and the position of the currently filmed object of interest. The direction of the source of light (from the position of the follower to the OoI) and the optical axis of the camera have to form predefined angles  $\pm 45^\circ$  in case of the *Three point lighting* approach and  $\pm 90^\circ$  for the *Strong side-lightening* technique. The desired distance of the light is given by the requirements on the intensity of illumination. Naturally, such defined desired trajectory for the followers is not feasible for the UAVs (mainly when the leader suddenly switches its attention between two objects of interest). Therefore, the trajectory cannot be directly used

as a desired control equilibrium, but its following needs to be integrated into the MPC framework taking into account motion and formation driving constraints applied in the optimization.

For followers control, we again assume that the orientation of the light source can be controlled independently to the UAV position control and so both these problems will be tackled in two separate optimization processes. In the trajectory tracking of the  $j$ -th follower, the optimization vector  $\Omega_{j,1} = [v_j(k), w_j(k), K_j(k)] \in \mathbb{R}^{3N}$ ,  $k \in \{1, \dots, N\}$ , collecting control inputs over the control horizon with  $N$  transition points is obtained by minimizing the multi-objective function

$$J_{j,1}(\Omega_{j,1}) = \frac{\chi}{N} \sum_{k=1}^N \|p_{j,d}(k) - p_j(k)\|^2 + \sum_{k=1}^N \frac{l_1}{N} K_j^{diff}(k)^2 + \frac{l_2}{N} v_j^{diff}(k)^2 + \frac{l_3}{N} w_j^{diff}(k)^2 + \frac{\kappa}{N} \sum_{k=1}^N \left( \min \left\{ 0, k \frac{\min_{neigh_j} \text{dist}(p_j(k), \Omega_{neigh_j,1}^*) - r_s}{\min_{neigh_j} \text{dist}(p_j(k), \Omega_{neigh_j,1}^*) - r_a} \right\} \right)^2 + \frac{\lambda}{N} \sum_{k=1}^N \left( \min \left\{ 0, k \frac{\text{dist}(p_j(k), O) - r_s}{\text{dist}(p_j(k), O) - r_a} \right\} \right)^2 + \frac{\mu}{N} \sum_{k=1}^N \left( \min \left\{ 0, k \frac{\|p_b(p_j(k), p_L^*(k), \varphi_L^*(k)) - p_j(k)\| - d_s}{\|p_b(p_j(k), p_L^*(k), \varphi_L^*(k)) - p_j(k)\| - d_a} \right\} \right)^2. \quad (4)$$

The first term in the objective function penalises deviation from the desired positions  $p_{j,d}(k)$  of the  $j$ -th follower. The desired position depends on the position of the leader in the  $k$ -th transition point using its actual plan  $\Omega_{L,1}^*$ , position of the object of interest that is filmed by the leader in the  $k$ -th transition point, and the requirements of the *Three point lighting* or *Strong side-lightening* approaches. The second term penalises too aggressive control behaviour as in the case of the leader's control. The third part of the cost function represents the mutual avoidance function, where set  $neigh_j := \{L, 1 \dots n_F\} \setminus j$  collects indexes of all remaining robots in the formation, which are the neighbours of the  $j$ -th follower. The fourth part ensures collision avoidance as for the leader. The last term, which is aimed to avoid having the UAV "illuminator" in the view of the camera, penalises the solutions in which the follower gets closer to the border of a pyramid representing space currently observed by the camera. The apex of the pyramid is roughly defined by the expected position of the leader  $p_L^*(k)$  following its actual plan  $\Omega_{L,1}^*$ . Orientation of the pyramid corresponds with the planned orientation of the camera  $\varphi_L^*(k)$  according to  $\Omega_{L,2}^*$ . In the cost function, distances between the position of the follower  $p_j(k)$  and position  $p_b(p_j(k), p_L^*(k), \varphi_L^*(k))$ , which is the closest point on the pyramid (defined by  $p_L^*(k)$  and  $\varphi_L^*(k)$ ) from the position  $p_j(k)$ , smaller than a safety distance  $d_s$  are

penalised. While  $d_s$  denotes the minimum desired distance from the border of the pyramid that is considered as safe, distance  $d_a$  is infeasible. The values of coefficients  $\chi$ ,  $\iota_1$ ,  $\iota_2$ ,  $\iota_3$ ,  $\kappa$ ,  $\lambda$ , and  $\mu$  set system behaviour similarly as in the case of the leader.

In the optimization process, the same inequality constraints protecting collisions with obstacles and other team members as in the position control of the leader are applied. The limits of control inputs for the followers may differ from the bounds used for the leader control, but the same structure of inequality constraints is used. Similarly, also the last term of the cost function (4) is supported by the inequality constraints  $d_a - \|p_b(p_j(k), p_L^*(k), \varphi_L^*(k)) - p_j(k)\| \leq 0, \forall k \in \{1, \dots, N\}$ .

As mentioned, movement of the light source on a gimbal or change of UAV heading in case of a firm attachment of the light source is limited by motion constraints. Moreover, mainly for the *Three point lighting* approach, smooth changes of direction of the light source carried by the followers are required. We have to solve this problem in each control step under the MPC framework for each follower  $j$  and therefore we formalize it as an optimization problem with vector  $\Omega_{j,2} = [\omega_j(k)] \in \mathbb{R}^N$ , where  $k \in \{1, \dots, N\}$ . The cost function that is minimised in the optimization is defined as

$$J_{j,2}(\Omega_{j,2}) = \nu \sum_{k=1}^N \text{diffang}(\varphi_{j,d}(k), \varphi_j(k))^2 + \xi \sum_{k=1}^N \omega_j^{\text{diff}}(k)^2, \quad (5)$$

where  $\varphi_{j,d}(k)$  is desired heading of the light source at the  $k$ -th transition point obtained from the given position of the object of interest and the position of the  $j$ -th follower at the  $k$ -th transition point of the plan  $\Omega_{j,1}^*$ , which is obtained as a solution of the first optimization problem. In comparison with the cost function used for the MPC control of the camera heading in eq. (2), here the desired orientation of the light is not given by the experts directly, but it is computed based on results of subsequently realized MPC steps for the leader and the followers composed into a compact formation via the rules of the *Three point lighting* and *Strong side-lightening* techniques. Nevertheless, the motivation of both parts of the cost function, meaning of the constants  $\nu$  and  $\xi$ , and also the employed inequality constraints are the same as for control of the camera direction.

#### IV. EXPERIMENTAL VERIFICATION AND SYSTEM DEPLOYMENT

The designed formation flying and stabilization system has been developed and verified in two robotic simulators, and the principles of lightening by a pair of UAVs were employed in two historical buildings including two large churches with a cooperation of historians, restorers, and filmmakers. The output of the system deployment was a set of images used by restorers for a plan of restoration works and two professionally edited documents broad-coasted by the main Czech TV and several other news providers (see the list

at <http://mrs.felk.cvut.cz/projects/cesnet>). The parameters have been used as  $n = 1$ ,  $N = 8$ ,  $t = 0.5$ ,  $\alpha = 1.5$ ,  $\beta_1 = 0.2$ ,  $\beta_2 = 0.2$ ,  $\beta_3 = 0.2$ ,  $\gamma = 0.2$ ,  $\delta = 0.01$ ,  $\varepsilon = 0.1$ ,  $\zeta = 0.1$ ,  $\eta = 1$ ,  $\chi = 1.5$ ,  $\iota_1 = 0.5$ ,  $\iota_2 = 0.5$ ,  $\iota_3 = 0.1$ ,  $\kappa = 0.1$ ,  $\lambda = 0.1$ ,  $\mu = 0.1$ ,  $\nu = 0.1$ , and  $\xi = 1$  in all experiments.

The first simulator, V-Rep, was used for simulation of both lightening techniques implemented by the UAV formation since it provides a useful tool for visualization of shadows on 3D objects caused by onboard lights (see Fig. 5 for an example of the formation flying around a 3D statue). The second simulator, Gazebo, enables a very realistic verification of the formation stabilization and control approach. In this simulator, we have implemented a plugin that emulates a firmware of the Pixhawk low-level stabilization, which is used in our platform being primarily designed for the MBZIRC competition in Abu Dhabi (<http://mrs.felk.cvut.cz/projects/mbzirc>) together with Vijay Kumar Lab, University of Pennsylvania, <http://kumarrobotics.org/> (for the HW system see [26]).

The simulator Gazebo enables to test the method even in more challenging conditions than are in historical buildings. For example in Fig. 2, a scenario inspired by an indoor space of Saint Nicholas Church in Prague was built to evaluate the system prior its using there. The testing scene includes six objects of interests and fourteen obstacles (including walls), which is much more complicated work-space in comparison with the tasks solved in the Saint Nicholas Church. The scenario in Fig. 2 with snapshots from one of the simulations in Fig. 4 verifies smooth transition between several consequent objects of interest (e.g. Fig. 4 (a-c)), the obstacle avoidance function (Fig. 4 (d-f)) and avoidance of collisions with other robots in the formation (Fig. 4 (g-h)). For the obstacle avoidance verification, the initial path was designed purposely infeasible for the formation by adding additional obstacles too close to the path and a failure of one of the follower was simulated (the follower with fill light, the robot on the right side of the formation, was suddenly stopped and the other UAVs were forced to avoid it).

The real deployment of the system in the Saint Nicholas Church located on the Old Town Square in Prague <http://www.svmikulas.cz/en/> is shown in Fig. 6 and in document at <https://youtu.be/g1NuPnLCFTg>. The picture show formation stabilization aimed to film the statue of Jesus Christ and the pulpit providing illumination by the *Three point lighting* technique. The second deployment of the proposed UAV formation lightening approach in the Virgin Mary Church in Sternberk, Moravia, is presented in Fig. 1 and in video at [https://youtu.be/-sTUwzFf\\_Mk](https://youtu.be/-sTUwzFf_Mk).

#### V. CONCLUSION

In this paper, a system designed for visual documentation and inspection of interiors of historical buildings by a formation of cooperating UAVs was described. The proposed solution of this application arises from our long-term basic research work in the field of UAV formation flying and it is mature enough to provide a solution that is already in use by historians and restorers to complement their work

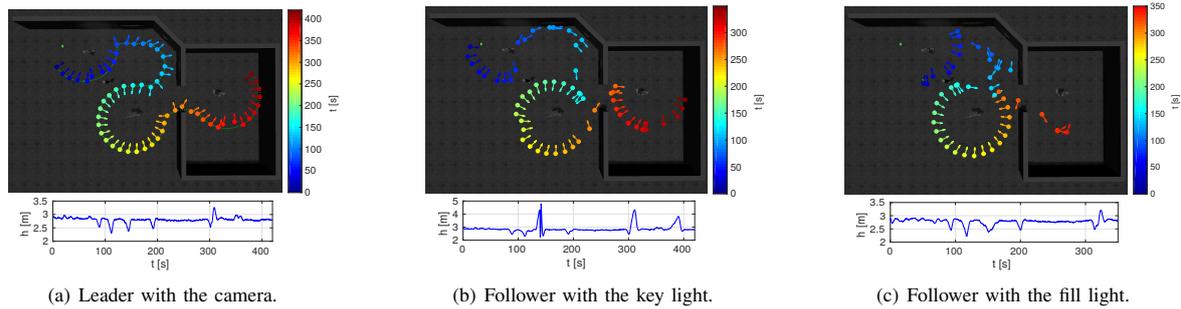


Fig. 2. Trajectories of all formation members in the experiment in the complex environment. The arrows denote the heading of the camera in different time, and the green dotted line shows the desired trajectory. In the graph,  $h$  is the height which corresponds with the  $z$  coordinate of the leader.

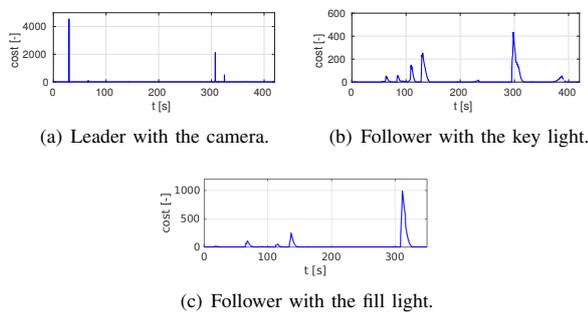


Fig. 3. The progress of values of the cost function used for trajectory tracking.

in inaccessible places of large churches. From this point of view, the basic functionalities of the system achieved TRL 8 (the system has been tested and launched in real operations), while the full-scale system operability is currently in TRL 7-8 levels. The complete technology including obstacle avoidance and failure recovery has been developed and demonstrated in realistic robotic simulations.

#### ACKNOWLEDGMENTS

The work has been supported by the Khalifa University in Abu Dhabi via the sponsorship for participation in the MBZIRC challenge, by the CESNET project for UAV mapping of historical locations, and by the Czech Science Foundation (GACR) under research project No. 17-16900Y.

#### REFERENCES

- [1] Y. Zhang and K.-L. Ma, "Lighting design for globally illuminated volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2946–2955, 2013.
- [2] J. Birn, *Digital Lighting and Rendering, 2013*. New Riders, year=, ch. Three-Point Lighting for 3D Renderings.
- [3] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *MMAR*, 2016.
- [4] J. Faigl, T. Krajník, J. Chudoba, L. Preucil, and M. Saska, "Low-Cost Embedded System for Relative Localization in Robotic Swarms," in *IEEE ICRA*, 2013.
- [5] T. Krajník, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.
- [6] W. Sheng, A. Okamoto, and S. Tanaka, "Visual point-based analysis of laser-scanned historical structures," in *International Conference on Culture and Computing*, 2015.
- [7] P. K. Allen, I. Stamos, A. Troccoli, B. Smith, M. Leordeanu, and Y. C. Hsu, "3d modeling of historic sites using range and image data," in *IEEE ICRA*, 2003.
- [8] N.-J. Shih, H.-J. Wang, C.-Y. Lin, and C.-Y. Liu, "3d scan for the digital preservation of a historical temple in taiwan," *Advances in Engineering Software*, vol. 38, no. 7, pp. 501 – 512, 2007.
- [9] R. Zlot, M. Bosse, K. Greenop, Z. Jarzab, E. Juckes, and J. Roberts, "Efficiently capturing large, complex cultural heritage sites with a handheld mobile 3d laser mapping system," *Journal of Cultural Heritage*, vol. 15, no. 6, pp. 670 – 678, 2014.
- [10] P. S. Blaer and P. K. Allen, "View planning and automated data acquisition for three-dimensional modeling of complex sites," *Journal of Field Robotics*, vol. 26, no. 11-12, pp. 865–891, 2009.
- [11] J. Fernandez-Hernandez, D. Gonzalez-Aguilera, P. Rodriguez-Gonzalez, and J. Mancera-Taboada, "Image-based modelling from unmanned aerial vehicle (uav) photogrammetry: An effective, low-cost tool for archaeological applications," *Archaeometry*, vol. 57, no. 1, pp. 128–145, 2015.
- [12] N. Hallermann, G. Morgenthal, and V. Rodehorst, "Vision-based monitoring of heritage monuments: Unmanned aerial systems (uas) for detailed inspection and high-accuracy survey of structures," *WIT Transactions on The Built Environment*, vol. 153, no. 12, p. 621, 2015.
- [13] K. Themistocleous, M. Ioannides, A. Agapiou, and D. G. Hadjimitsis, "The methodology of documenting cultural heritage sites using photogrammetry, uav, and 3d printing techniques: the case study of asinou church in cyprus," 2015.
- [14] M. Hess, V. Petrovic, D. Meyer, D. Rissolo, and F. Kuester, "Fusion of multimodal three-dimensional data for comprehensive digital documentation of cultural heritage sites," in *Digital Heritage*, 2015.
- [15] C. Teuliere, E. Marchand, and L. Eck, "3-d model-based tracking for uav indoor localization," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 869–879, 2015.
- [16] S. Minaeian, J. Liu, and Y. J. Son, "Vision-based target detection and localization via a team of cooperative uav and ugvs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 1005–1016, 2016.
- [17] A. L. Majdik, L. Tizedes, M. Bartus, and T. Szirnyi, "Photogrammetric 3d reconstruction of the old slaughterhouse in budapest," in *IWCIM*, 2016.
- [18] M. Saska, V. Spurny, and V. Vonasek, "Predictive control and stabilization of nonholonomic formations with integrated spline-path planning," *Robotics and Autonomous Systems*, vol. 75, no. Part B, pp. 379–397, 2016.
- [19] M. Saska, J. S. Mejia, D. Stipanovic, V. Vonasek, K. Schilling, and L. Preucil, "Control and Navigation in Manoeuvres of Formations of Unmanned Mobile Vehicles," *European Journal of Control*, vol. 19, no. 2, pp. 157–171, 2013.
- [20] M. Saska, J. S. Mejia, D. M. Stipanovic, and K. Schilling, "Control and navigation of formations of car-like robots on a receding horizon," in *Proc of 3rd IEEE Multi-conference on Systems and Control*, 2009.
- [21] M. Saska, V. Vonasek, T. Krajník, and L. Preucil, "Coordination and Navigation of Heterogeneous MAV&UGV Formations Localized by a

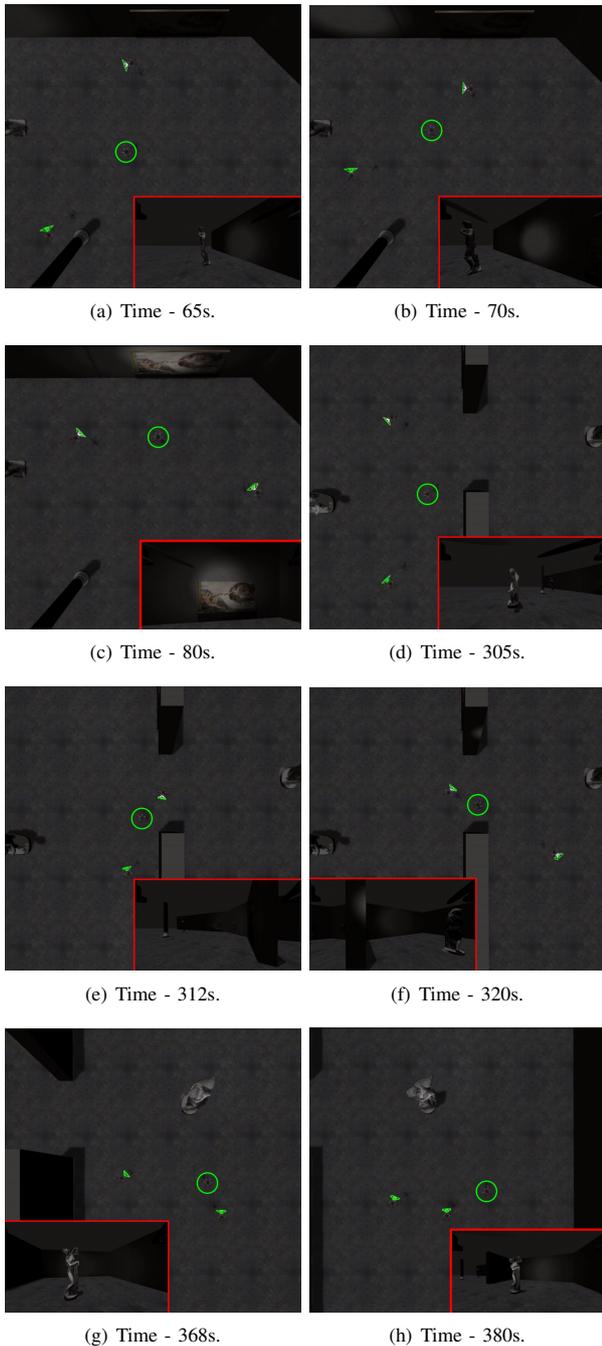


Fig. 4. Snapshots of the formation during the transition to another object of interest (a-c), during the transition to another object of interest and concurrent avoiding the obstacle (d-f), and during avoiding the broken follower carrying the fill light (g-h). The three UAVs are flying in a formation to realize a film crew consisting of a cameraman (the robot denoted by the green circle) and two illuminators (the robots with pyramids representing the lights) realizing the *Three point lighting* approach in an environment inspired by the interior of the Saint Nicholas church.

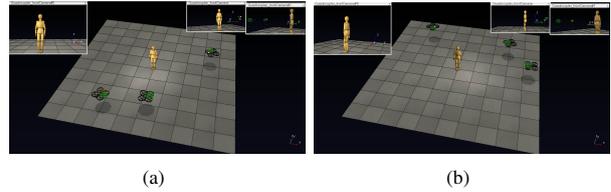


Fig. 5. Simulation of the *Three point lighting* technique in V-Rep.

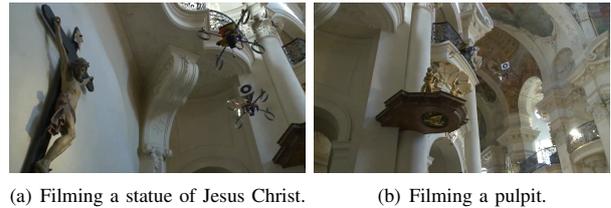


Fig. 6. A hovering formation performing part of the *Three point lighting* technique in the Saint Nicholas church.

- 'hawk-eye'-like Approach Under a Model Predictive Control Scheme," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.
- [22] M. Saska, Z. Kasl, and L. Preucil, "Motion Planning and Control of Formations of Micro Aerial Vehicles," in *The 19th IFAC World Congress*, 2014.
- [23] M. Saska, V. Vonásek, T. Krajník, and L. Přeucil, "Coordination and navigation of heterogeneous uavs-ugvs teams localized by a hawk-eye approach," in *IEEE/RSJ IROS*, 2012.
- [24] V. Spurny, T. Baca, and M. Saska, "Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control," in *MMAR*, 2016.
- [25] M. Saska, V. Spurny, and L. Preucil, "Trajectory Planning and Stabilization for Formations Acting in Dynamic Environments," in *Progress in Artificial Intelligence*, 2013.
- [26] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajník, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots. First online.*, 2016.

©[2018] Wiley Online Library. This article was accepted for publication in Journal of Field Robotics: T Baca, P Stepan, V Spurny, D Hert, R Penicka, **M Saska**, J Thomas, G Loianno, and V Kumar: Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle, 2018.



# Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle

---

**Tomas Baca\***

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
tomas.baca@fel.cvut.cz

**Petr Stepan**

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
petr.stepan@fel.cvut.cz

**Vojtech Spurny**

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
vojtech.spurny@fel.cvut.cz

**Daniel Hert**

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
vojtech.spurny@fel.cvut.cz

**Robert Penicka**

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
robert.penicka@fel.cvut.cz

**Martin Saska**

Department of Cybernetics  
Faculty of Electrical Engineering  
Czech Technical University in Prague  
martin.saska@fel.cvut.cz

**Justin Thomas<sup>†</sup>**

GRASP Laboratory  
University of Pennsylvania  
Philadelphia, US  
jut@seas.upenn.edu

**Giuseppe Loianno**

Department of ECE and MAE  
Tandon School of Engineering  
New York University  
New York City, New York  
loiannog@seas.upenn.edu

**Vijay Kumar**

GRASP Laboratory  
University of Pennsylvania  
Philadelphia, US  
kumar@seas.upenn.edu

## Abstract

This paper addresses the perception, control and trajectory planning for an aerial platform to identify and land on a moving car at 15 km/h. The hexacopter Unmanned Aerial Vehicle (UAV), equipped with onboard sensors and a computer, detects the car using a monocular camera and predicts the car future movement using a nonlinear motion model. While following the car, the UAV lands on its roof, and it attaches itself using magnetic legs. The proposed system is fully autonomous from takeoff to landing. Numerous field tests were conducted throughout the year-long development and preparations for the MBZIRC 2017 competition, for which the system was designed. We propose a novel control system in which a Model Predictive Controller is used in real time to generate a reference trajectory for the UAV, which are then tracked by the nonlinear feedback controller. This combination allows to track predictions of the car motion with minimal position error. The evaluation presents three successful autonomous landings during the MBZIRC 2017, where our system achieved the fastest landing among all competing teams.

---

\*<http://mrs.felk.cvut.cz>

<sup>†</sup><http://www.grasp.upenn.edu>

# 1 Introduction

Autonomous take-off and landing are the key components and also the most challenging components of all fully autonomous UAV systems. Precise landing ability is important for autonomous docking of UAV platforms (mainly Micro Aerial Vehicles - UAVs) into a recharging station in missions requiring repeated flight operations, and also in information gathering and delivery applications, where it is required to reach a precise, desired position and then return to a base. Even more challenging abilities are required for landing on a moving platform, especially if the platform may not be equipped with a precise localization system. Although the use of a moving helipad introduces uncertainty and a source of possible failures into the UAV system, it extends the application domain of UAVs and especial of multi-rotor helicopters. These platforms benefit from high robustness and maneuverability. However, they suffer from a short operational time, and a cooperation with another vehicle is often required. A UAV system capable of vertical take-off and landing on a moving vehicle may be deployed from boats, trains or cars in areas close to the target locations of the UAV mission. Short-term flights of this kind efficiently exploit the abilities of UAVs, and combining them with a moving platform extends their operational range.

Hundreds of works dealing with autonomous landing on static and dynamic helipads have been published in this decade in the robotics literature describing advanced control and landing pattern detection algorithms and showing promising simulations and laboratory experiments. However, only a few of these works have demonstrated deployment in real-world conditions, and none of them have presented a reliable performance that enables repeated landing on a fast-moving helipad in a demanding outdoor environment. This huge reality gap was identified by the scientific board of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017 competition, organized by the Khalifa University of Science in Abu Dhabi. The aim of this board of top scientists in robotics was to select tasks on the edge of the current state-of-the-art to provide a significant impact on the robotic community. Automatic landing on a fast-moving vehicle was the first challenge on their list.

The existence of the reality gap was confirmed in the MBZIRC competition, where only five teams (out of 142 registered teams from almost all the best robotic groups worldwide) successfully landed during the competition on a car moving at a speed of 15 km/h and only two teams (CTU-UPenn-UoL and the team of the University of Bonn) landed precisely in both trials of Challenge 1 of the competition. The CTU-UPenn-UoL system is presented here. Most importantly, the MBZIRC competition can be considered as a relevant and objective benchmark of this task, which is currently being investigated by the robotic community, since several competitive solutions were compared in the same experimental setup. The same car, the same landing pattern and the same trajectory and velocity profile of the car were used for all competitors in the same environment. The criterion for success was the shortest time of landing. Moreover, the successful teams had to achieve the goal after only a few minutes of preparation, without the option of postponing the beginning of their time slot. By standard practice in most laboratory experiments, no repeated tests were allowed, and moreover, the system robustness was exhibited in the current environmental conditions (light and windy), since the teams could not influence the start of their trial.

The solution described in this paper presented the best reliability among all teams and it achieved the fastest performance in the entire competition. Only our system was able to land three times in the competition, without a failure in autonomous mode - see Table 1. The fastest time of landing was achieved by the proposed system during the grand challenge, where all MBZIRC challenges were solved simultaneously. This even increased the demands on system robustness and immediate deployment without any preparation. The key components of the system (HW and SW) that provided this high reliability and performance in comparison with state-of-the-art works are described in the following paper. A novel UAV state estimation approach is presented together with a predictive trajectory tracking technique that enables us to track and predict an estimated position of the landing pattern, with the necessary precision and maneuverability to be able to follow the car even in turns of its path. For precise ground vehicle state estimation, which is crucial information for the landing UAV, fast and robust visual localization of the landing pattern is proposed. The detected positions of the car are filtered using an Unscented Kalman Filter (UKF) based technique with an

assumed car-like model of the vehicle, while the prediction of the car position in future takes into account a known profile of the track that is followed by the moving helipad. The model predictive control based approach applied for car tracking using an estimate of its movement in future is the most important element of the proposed system that enables the UAV to land precisely on a platform following a speed profile that is close to the speed limit of the UAVs and on a non-straight path.

## 1.1 State-of-the-art

The academic community has identified great interest in the task of autonomous landing of Unmanned Aerial Vehicles on ground or marine vehicles. The survey in (Jin et al., 2016) provides an overview of techniques used for vision-based autonomous landing. A list of various visual markers with the corresponding detection techniques is referenced, as well as hardware design, control and estimation methods for both indoor and outdoor tasks. Similarly, methods for more general autonomous landing of an unmanned aerial system are described in (Kong et al., 2014).

Vision-based estimation of a ground vehicle states using an only UAV onboard sensors is proposed in (Benini et al., 2016). Robust marker detection using onboard GPU (Graphics Processing Unit) provides precise pose information even in occluded and cluttered environments. The authors of (Lin et al., 2017) also propose a method for automatic detection and estimation of a landing marker onboard a ship deck. They also aim to provide a robust pose estimate when the marker is partially occluded, or when the scene contains marker reflections. However, both solutions lack an experimental evaluation that would test the system during fully autonomous landing.

The authors of (Jung et al., 2015), (Jung et al., 2016), (Fu et al., 2016) and (Ghommam and Saad, 2017) deal with simulations of landing marker detection. They also propose guidance laws for autonomous landing, but also in simulation. Simulated autonomous landing on a ship is presented in (Tan et al., 2016).

Many of indoor experiments on autonomous landing on a slow-moving target are presented in works by (Ghamry et al., 2016), (Araar et al., 2017), (Lee et al., 2012) and (Bi and Duan, 2013). An indoor solution with a motion capture system is presented in (Ghamry et al., 2016). A ground robot and a UAV are both controlled by a centralized system to fulfill missions which include autonomous takeoff and landing of the UAV, atop the ground vehicle. The system presented in (Lee et al., 2012) uses sensors onboard the UAV and relies on the Vicon motion capture system and external computational unit. Thanks to the motion capture system, the UAV is able to conduct a patrol search for the ground vehicle. When the ground vehicle is located, it switches to relative localization based on visual marker detection. Similarly, a system for indoor autonomous tracking and landing is presented in (Hui et al., 2013). Camera images are also processed off board on an external computer.

Multiple works describe systems capable of autonomous outdoor flight while tracking a static or moving marker. In (Yang et al., 2013), (Masselli et al., 2014) and (Yang et al., 2015), UAV systems capable of hovering and landing on a static target are proposed. Autonomous landing on a target moving at slow speed up to 1 m/s is presented in (Kim et al., 2014) and in (Lee et al., 2016). The authors of (Xu and Luo, 2016) present a solution capable of landing on a moving car at speeds of 7 m/s. The presented system was tested in scenarios with the ground vehicle moving along a straight line.

The most similar approach to our work is presented by (Borowczyk et al., 2017) and (Hoang et al., 2017). The authors of (Borowczyk et al., 2017) propose a system that utilizes a vision-based approach combined with inertial and GPS measurements from a cell phone placed on the ground vehicle. Experiments show landings at speeds up to 50 km/h. However, it is unclear whether the system is capable of landing during non-linear motion of the car. Moreover, precise knowledge of the global position of the car is an assumption that is problematic in most applications. Successful landing on a moving vehicle in an outdoor environment is also described in (Hoang et al., 2017). Only onboard sensory data and computation power are used. The proposed solution is able to track and land on a car moving at a speed of up to speed 2 m/s.

The competitive solutions in the MBZIRC competition were presented by the team of the Beijing Institute of Technology and by the team of the University of Bonn (Beul et al., 2017). They also landed multiple times, but their systems have not yet been published. It is therefore not possible to compare the two systems and to highlight differences. Nevertheless, all three solutions can be considered as a valuable contribution to the field of robotics since according to our knowledge no other system exists that can offer a complete solution to this very demanding and complex challenge in these outdoor conditions (which was also the reason why this task was selected by respected leaders in the field of robotics for the competition).

## 1.2 Contributions

This manuscript presents a complete system for automatic detection, estimation, tracking and landing on a moving car with an unmanned aerial vehicle. The proposed method enables the UAV to detect a landing pattern in images from a single onboard camera and to calculate the position of the car relative to the UAV. The computer vision algorithm provides fast and robust detection of the landing marker using a SuperFisheye camera lens. Position and velocity of the car are estimated and predicted in a global frame of reference using a non-linear motion model by an Unscented Kalman Filter. The proposed control architecture is tailored specifically to the challenge of following fast dynamic objects with a UAV by leveraging the non-linear state controller in conjunction with a novel model predictive control tracker. Future predictions of the car movement are reshaped by the MPC tracker into a feasible state reference, which is reflected on the states of the real UAV by the non-linear state controller. The novelty of this approach is in the combination of the state-of-the-art methods to accomplish a robust execution of the demanding task in real-world experimental conditions. Moreover, the novel MPC tracker is a contribution which allows to track arbitrarily long reference trajectories without solving an MPC task for the whole reference. The experimental results show that the UAV can follow a car moving at 15 km/h autonomously and land on its roof, while attaching itself using magnetic legs. The system is robust to very challenging outdoor conditions with a wind speed of up to 10 m/s, varying light conditions and blowing sand in the air. The approach presented in the manuscript provides precision and repeatability in the landing task, which is a crucial element for fully autonomous missions (such as periodical surveillance, reconnaissance, object carrying, and monitoring), for which UAVs are an especially appealing option.

## 1.3 Problem definition

The task, as it is described by the rules of the competition, consists of an autonomous search and landing on a moving ground vehicle by an unmanned aerial vehicle, a multirotor helicopter. The competition takes place in an arena with dimensions of  $90 \times 60$  meters, as shown in Figure 1a. A track in the shape of figure 8 is marked on the ground where the car is supposed to drive starting at a random place and heading in a random direction. The landing area is a square with dimensions of  $1.5 \times 1.5$  m with a marker X, as shown in Figure 1b, placed 1.5 m above the ground on the roof of the vehicle. A magnetic or suction gripper can be used to attach the UAV to the surface, which is made of a ferrous material. The moving vehicle starts at a constant speed of 15 km/h. It reduces the speed to 10 km/h after 6 minutes and to 5 km/h after 12 minutes from the start. However, our system was designed to land independently of the speed level. No human intervention is allowed in the fully autonomous mode.

## 2 Experimental hardware platform

The experimental platform was designed from off-the-shelf parts, with the aim to simplify reproducibility and potential maintenance. The same platform was also successfully used for the treasure hunt challenge – MBZIRC challenge No. 3 (our team won this challenge, as described in (Spurny et al., 2018)), where three UAVs cooperatively collected small objects. More importantly, we intended to reuse the platform for future research activities, which introduced a need for simple potential modifications to the system.

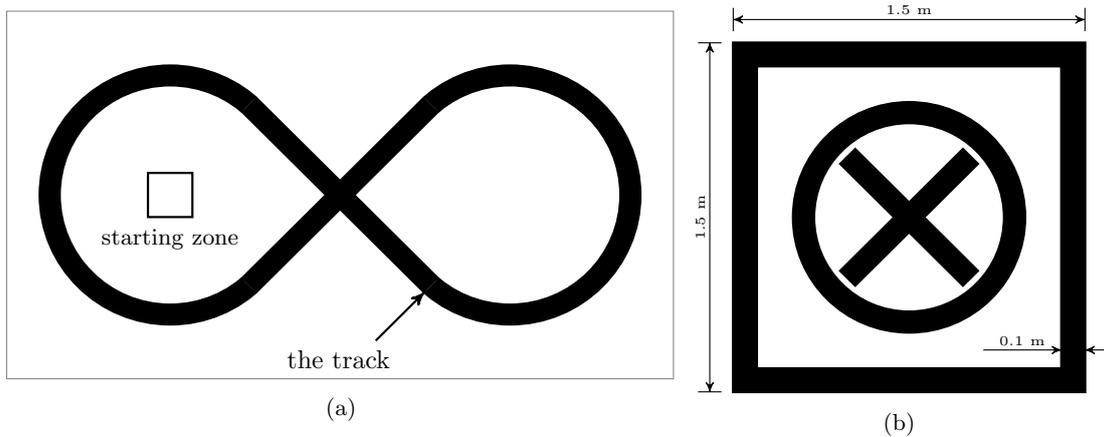


Figure 1: (a) a schematic image of the arena, showing the track for the ground vehicle, (b) the visual marker attached to the helipad of the ground vehicle, as described in the rules of the competition.

The proposed platform is a multirotor vehicle based on a DJI F550 hexacopter frame equipped with the DJI E310 propulsion system. Most components were chosen as individual and commercially available parts, to maximize the simplicity of the system, minimize the cost and to allow custom modifications if needed for any particular task. Key components are shown in Figure 2. See (Spurny et al., 2018) for a different configuration of the system, proposed for the MBZIRC treasure hunt challenge. A flight controller board is required to allow basic flight capability. The PixHawk flight controller (Meier et al., 2012) was chosen for its open source firmware and for its well-documented interface, which allows us to connect it to a high-level onboard computer. PixHawk contains sensors such as gyroscopes, accelerometers, an atmospheric pressure sensor, a magnetometer, and GPS, and it produces a single position, velocity and orientation estimate of the UAV in global world frame by their measurements.

Onboard computations are performed on an Intel NUC-i7 computer with an Intel i7 processor and 8 GB of RAM. The computer is installed with GNU Linux Ubuntu 16.04 and the Robot Operating System (ROS) in the Kinetic version. The Robot Operating System is a middleware library for C++ and Python programming languages. It provides a convenient way of building a complex system of applications with the asynchronous exchange of messages. An ecosystem of existing programs exists covering functionalities such as visualization, logging and data sharing, geometric transformations, etc. Sensor drivers are often found with the ROS interface already integrated, which makes them simpler to integrate.

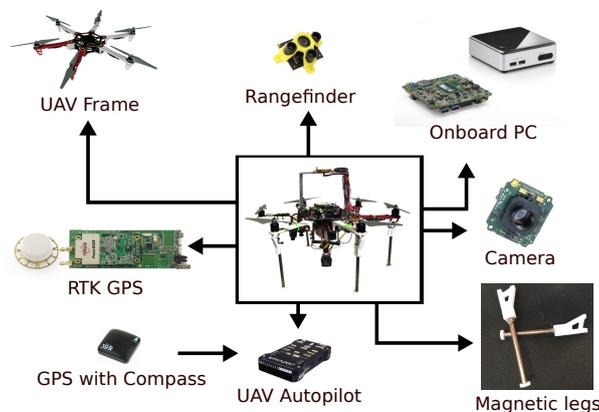


Figure 2: Schematic of individual hardware modules on the UAV

To improve the localization accuracy of the UAV in space, we integrated the PRECIS-BX305 GNSS RTK BOARD differential GPS receiver (Tersus-GNSS, 2017). Differential RTK (Real Time Kinematics) GPS uses a ground base station to transmit corrections to the UAV, which practically eliminates GPS drift. The TeraRanger time-of-flight laser rangefinder (Ruffo et al., 2014) serves two purposes. During a flight, it measures the distance to the ground, which is used to improve the estimation of the UAV height. In the landing task, during touchdown on the ground vehicle, it serves as a trigger for switching off the propellers. To detect the car, a single Matrix-vision mvBlueFOX-MLC200w camera is mounted on a fixed, down-facing mount beneath the UAV. A SuperFisheye lens was chosen to maximize the chance of detection in the final stages of landing when the landing pattern is close to the camera. Its global shutter provides images free of the rolling shutter effect.

### 3 System structure

The guidance law presented in this paper is a modular pipeline consisting of components which are depicted in Figure 3. The following paragraphs give a list of the components, which are subsequently described in sections 4 to 9 of this paper.

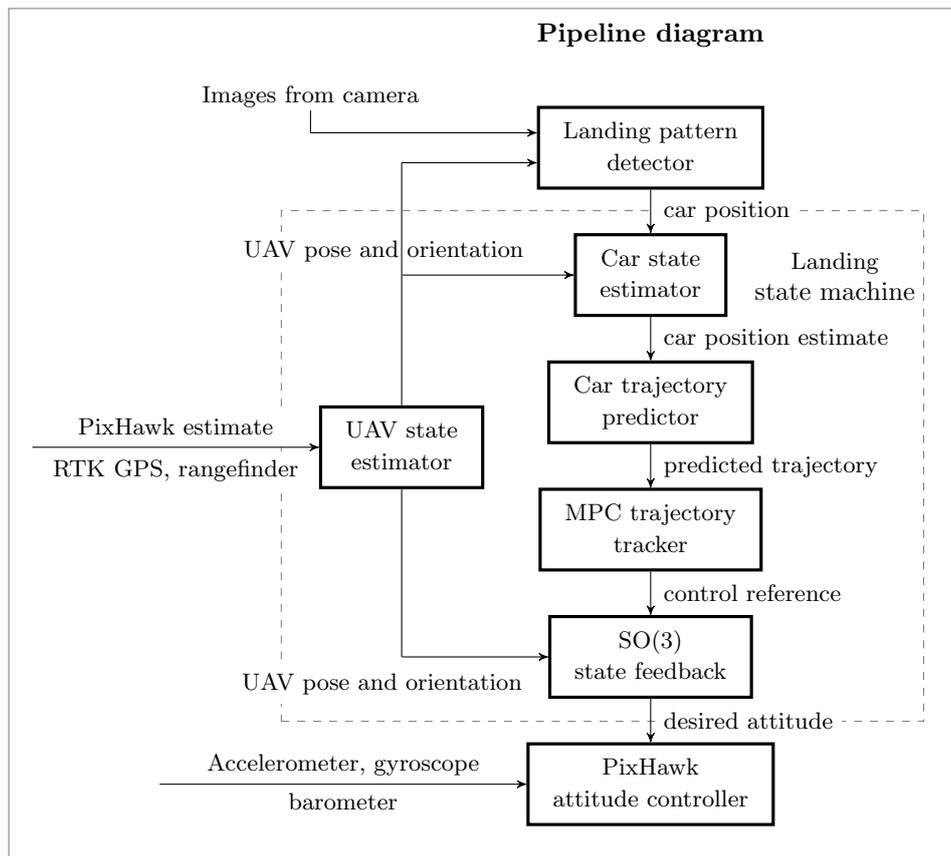


Figure 3: Scheme of the software pipeline for landing on a moving vehicle. See Section 4 for *Landing pattern detector*, Section 6 for *Car state estimator* and *Car state predictor*, Section 7 for *UAV state estimator*, Section 8 for *MPC trajectory tracker* and Section 9 for *SO(3) state feedback*. The dashed line surrounds the parts which are controlled by a landing state machine, later discussed in Section 5.

The first component is the **landing pattern detector** (presented in Section 4), which provides measure-

ments of car position in the world frame coordinate system. Position measurements are processed by the **car state estimator** (Section 6), using an Unscented Kalman Filter. Unmeasured states such as acceleration and heading are required to fully predict the future trajectory of the car. The **car state predictor** calculates the future trajectory of the car, starting from the latest state estimate and using the same model and the same non-holonomic model as is used for the estimation. The predicted future trajectory serves as a reference for the **MPC tracker** (Section 8), which minimizes a quadratic error of UAV future states over a prediction horizon to fly precisely above the car given the dynamical constraints of the aircraft. The MPC tracker then outputs desired states (position, velocity, and acceleration) to the **state feedback controller** (Section 9). The state feedback controller, being the last part of the pipeline implemented in the high-level computer, produces attitude and thrust commands for the PixHawk flight controller.

The **car state estimator** serves two purposes within our pipeline. First, it filters the incoming signal from the **landing pattern detector**, the measurement variance of which is adjusted with respect to the UAV height. Secondly, it estimates unmeasured states (velocity, acceleration, heading, turn curvature), which are required to predict the car future movement. The resulting estimate is outputted at 100 Hz. Using the information from the car state estimator, we predicted the future movement of the car, using the same dynamic model as during the estimation. The curvature of the predicted trajectory is biased using a known map of the arena and the track on which the car was driven. The predicted trajectory is updated at 30 Hz.

In our pipeline, a trajectory tracker is responsible for generating a set of desired states of the UAV (position, velocity, and acceleration) to follow the trajectory generated by the car state estimator. It uses decoupled, third order translational dynamics to simulate a virtual UAV at 100 Hz. The virtual UAV is then controlled by Model Predictive Control with a 8 s prediction horizon, also at 100 Hz. States of the virtual UAV are sampled and are handed out to the state feedback controlled as a reference. Thanks to the MPC, the tracker provides the necessary feed-forward action to follow the known future path. The particular MPC control approach is based on previous work presented in (Baca et al., 2016), further extended to support the state constraints in velocity and acceleration.

## 4 Visual localization of the landing platform

Robust, precise and fast detection of the landing pattern is a crucial ability to achieve reliable landing on moving vehicles. In the system designed for the MBZIRC 2017 competition, we relied on a color mvBlueFOX-MLC200w camera with a global shutter, which is important for recognizing moving objects from a camera on the fast-moving UAV. Another advantage of this light camera is the fast frame rate, 93 images per second, with resolution  $752 \times 480$ . Although a color camera was used, the image analysis was conducted after converting the obtained images to greyscale, thanks to the landing pattern being black and white. Using a Sunex DSL215 miniature SuperFisheye lens, the camera provides a horizontal field of view of  $185^\circ$ . It observes the car under the UAV even in the event of UAV tilting, so it is not necessary to use a gimbal camera stabilizer. This reduces the complexity and weight of the system. This scheme provides a very simple, cheap, and robust solution that can be applied in various landing scenarios beyond the MBZIRC competition.

Let us now briefly describe the image processing algorithm that was used for landing on the moving helipad in the MBZIRC competition. In this paper, we focus on general approaches that could be re-used for detecting landing patterns similar to the pattern used in MBZIRC 2017 to provide a complete system for autonomous vision-based landing. For special details on the technique adapted for localizing the MBZIRC pattern, see (Stepan et al., 2018), where all vision approaches employed by our team in the MBZIRC competition are summarized.

As was mentioned above, to ensure outdoor deployment in real scenarios, the detection procedure has to be robust to various weather conditions, changes in light intensity, and direct sunshine with shadows cast by the aircraft and other objects in the environment, such as the support structure of the MBZIRC arena.

Other requirements are low computational complexity to be able to use small and simple platforms, fast response, and the use of standard computer vision libraries, e.g., OpenCV, to provide simple implementation and reproducibility. Mainly the very fast response (50 FPS and more) and the availability of low computational power are contradictory requirements that are hardly achievable by state-of-the-art computer vision approaches and require the design of new methods suited for this special application.

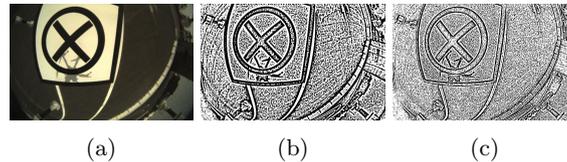


Figure 4: (a) Image taken by the camera, (b) result of the adaptive threshold with box size 11 pixels, (c) adaptive threshold with box size 5 pixels. The box size is variable and depends on the thickness of the expected segments in the landing pattern and decreases with the height of the UAV.

In the proposed pattern detection approach, the first step is adaptive thresholding with a variable box, the size of which depends on the UAV height and the known parameters of the landing pattern (for examples with box size 5 pixels and 11 pixels, see Figure 4b and 4c, respectively). During the experiments, the size of the box spanned on the interval  $[5, 21]$ , where it was defined as  $2(w/12) + 1$ , where  $w$  stands for the width of the detected white square of the landing pattern in pixels. The advantage of the adaptive threshold is its robustness to light intensity. The contours of the painted pattern (the circle and the lines in our case) are then simply detected in these segmented images.

An important part of the algorithm is the undistort procedure. That needs to be applied to compensate the distortion caused by the SuperFisheye lens. The lens parameters can be identified using OpenCV and its fish-eye model. However, the undistort function provided in the library is too slow, and a new method needs to be designed. The employed approach, which is described in details in (Stepan et al., 2018), relies on the fact that the distortion coefficients are known in advance, and the scales required for computing undistorted coordinates can be precomputed.

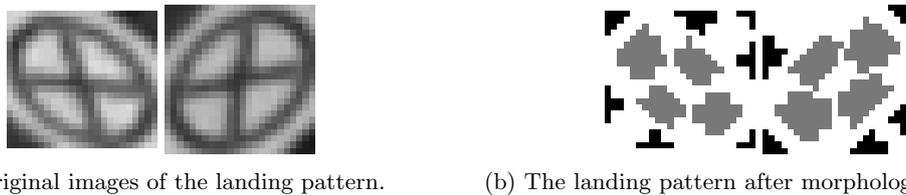


Figure 5: Operation of morphological closing was applied as a part of the recognition pipeline. The original camera images were cropped, which results in the images with resolution of  $24 \times 24$  px.

Robust detection of the MBZIRC 2017 landing pattern is based on detecting the outer circle and then the inner cross, to exclude false positive detections. Based on our experience, this design enables the pattern to be detected robustly in all phases of the landing approach. We can recommend it for other projects, where the autonomous UAV landing is required. A combination of the circle and the inner cross should also be used, if possible, in designs of landing patterns. In the initial phase of the approach, where the length of one of the axes of the ellipse (the detected circle) is shorter than 30 pixels, due to the long distance between the helipad and the UAV, the lines of the cross cannot be detected reliably. Then the cross is detected using the morphology operation *closing* and searching for areas similar in size. Circle detection is positively confirmed if four closed areas similar in size are found within the circle, see Figure 5 for example.

Later, if the UAV approaches closer to the helipad and the circle size is 30-150 pixels, the cross can be detected by Guo Hall thinning (Guo and Hall, 1989), which enables the lines inside the circle to be detected robustly. Positive detection of the landing pattern is confirmed if the crossing point of the two biggest lines

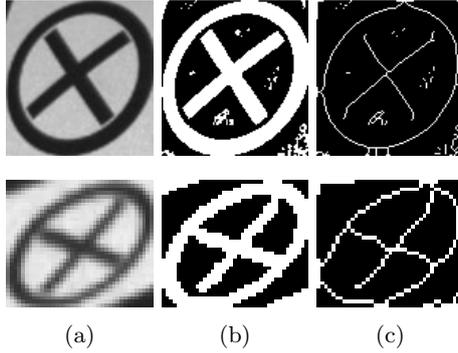


Figure 6: (a) Image taken by the camera, (b) application of the adaptive threshold with box size 11 pixels, (c) results of Guo Hall thinning.

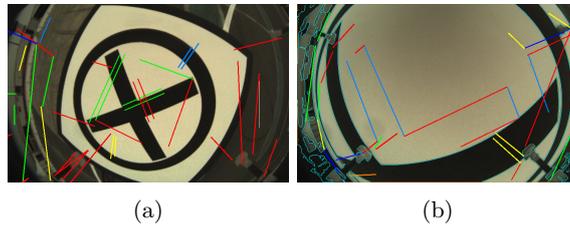


Figure 7: Line detection (a) if the cross is inside the circle and (b) if only a part of the landing pattern is detected.

of the cross inside the circle is detected near the center of the ellipse. Results of the landing pattern detection process using the Guo Hall thinning algorithm are shown in Figure 6.

Finally, if the circle is larger than 150 pixels, the cross is detected by recognizing its border, i.e., by detecting two pairs of parallel lines (the red and green lines in Figure 7a). This approach provides robust detection of the landing pattern if the entire circle is not visible in the image. Figure 7b depicts the pattern reconstructed only from two visible lines of the cross.

An estimate of the height of the UAV is required to select the proper approach for pattern detection. In the proposed system, this information is obtained using the laser rangefinder and its fusion with the onboard IMU (Inertial Measurement Unit) to compensate the deviation of the measurement caused by a vehicle with a variable height profile, when it appears under the approaching UAV. If the pattern is detected, its known parameters (the known dimension of the circle and the cross) can be used to make a precise measurement of the relative distance between the helipad and the UAV. This information is used for estimating the position of the landing pattern in the global world coordinate system, which is used as the desired state of the position controller.

## 5 Landing state machine

The autonomous task of the UAV is driven by a single state machine from autonomous takeoff to landing. The state machine (Figure 8) takes control of the UAV after a signal has been given by an operator or after the start time is reached. After taking off, it moves the UAV above the center of the map by switching to the *Fly to waiting point* state. The UAV waits for the car to appear in the field of view, while it hovers at the height of 8 m above the crossing of the two roads. Several factors have influenced the choice this strategy, e.g., the real speed and acceleration constraints of the UAV and the known trajectory and velocity profile of

the car. This approach minimizes the complexity of the system and also provides the shortest mean time for locating the target, given the mentioned constraints. By using this strategy, we also maximize the possible quality of the images being captured onboard the UAV since any movement of the UAV introduces a motion blur, which negatively influences the initial spotting of the target.

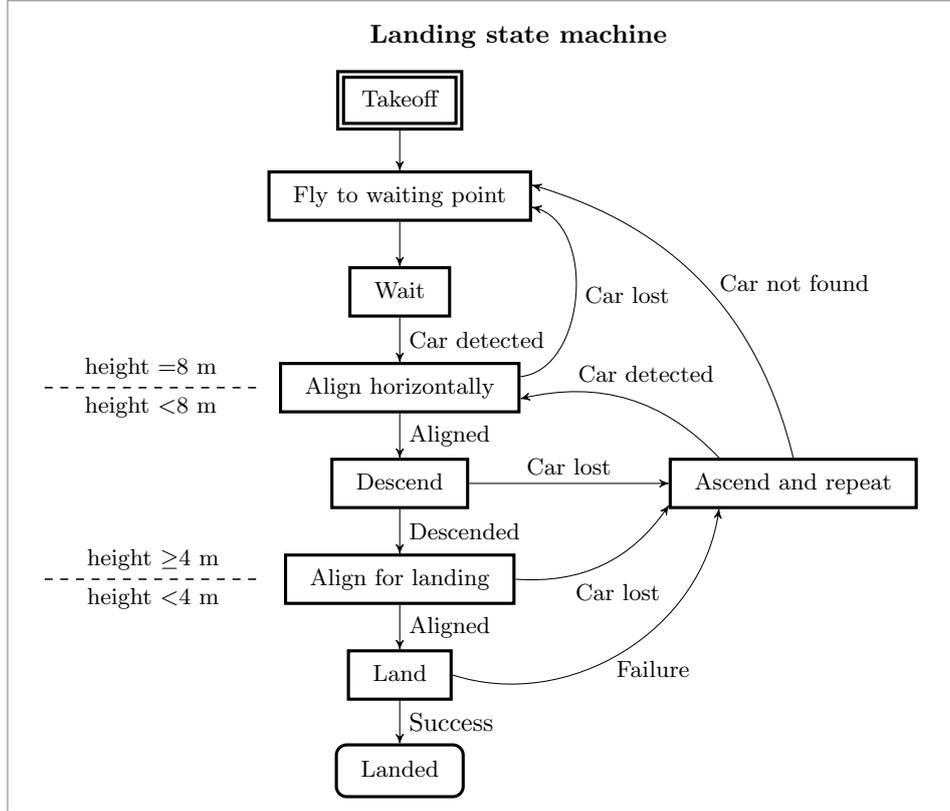


Figure 8: The autonomous flight, from takeoff to landing, is controlled by the *landing state machine*. The linear passage from the *Takeoff* state to the *Landed* state can be divided to three groups of states, based on the current height above ground – 8 m or higher, between 4 and 8 m and below 4 m. The state *Ascend and repeat* handles situations when the car was lost from the field of view of the camera.

When the target is first spotted, and the covariance of its state estimate exceeds a defined threshold, the UAV starts to align its horizontal position with the car while maintaining 8 m height (*Align horizontally* state). The aligning uses an approach strategy, which exploits the fact that the car velocity vector points towards the center of the map and thus towards the UAV. The approaching trajectory is created around a mutual meeting point  $\mathbf{M}_{[t]}$ , which is the closest point of the current position of the UAV  $\mathbf{U}_{[t]}$  to the predicted trajectory of the car. The trajectory meets the following properties, where  $\mathbf{C}_{[t]}$  is the current position of the car and  $\mathbf{E}_{[t]}$  is the last point of the car prediction:

- the portion of the UAV trajectory in between  $\mathbf{M}_{[t]}$  and  $\mathbf{E}_{[t]}$  is found by sampling the predicted trajectory of the car starting at  $\mathbf{M}_{[t]}$ ,
- the other portion of the UAV trajectory in between  $\mathbf{U}_{[t]}$  and  $\mathbf{M}_{[t]}$  requires creating a trajectory of the same time duration as the portion of the predicted trajectory of the car from  $\mathbf{C}_{[t]}$  to  $\mathbf{M}_{[t]}$ ,
- the resulting trajectory does not require motion (and thus a control action) in the direction parallel the current car motion, which means that the trajectory will not lose it from the field of view because the onboard camera is tilted away from the car.

In the special case, where  $\mathbf{M}_{[t]}$  does not reside on the predicted trajectory of the car, the whole prediction is used as a reference for the MPC tracker. This situation may occur if the car is first spotted while driving away from the UAV.

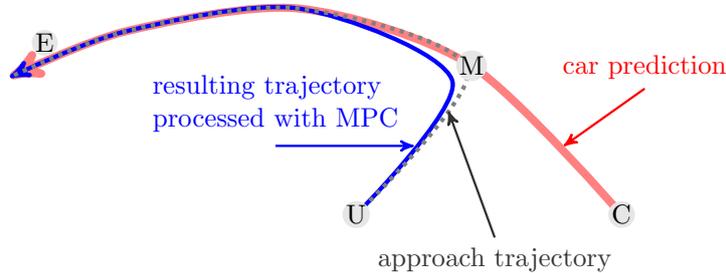


Figure 9: Illustration of the approach strategy. Point  $\mathbf{M}$  represents the common meeting point of the UAV and the car,  $\mathbf{U}$  is the current position of the UAV,  $\mathbf{C}$  is the current position of the car and  $\mathbf{E}$  marks the final point in the car prediction. The car predicted trajectory is shown in red, the approach trajectory is marked as dotted and the resulting feasible trajectory, optimized by MPC, is shown in blue.

After the UAV is aligned with the car horizontally within 1.5 m, the state machine switches to the *Descend* state. While in the *Descend* state, the height decreases to 4 m, the lowest height at which it is still possible to follow the car continuously given the particular UAV and camera configuration. Once a height of 4 m is reached, the state machine transitions to the *Align for landing* state, where it waits for two conditions to be met to initiate the final landing on the moving car. First, the UAV has to be aligned horizontally within 0.3 m of the center of the landing pattern. Second, transition to the *Land* state is allowed only above the straight parts of the track. Finally, the landing maneuver is executed, in which a fast descent is made to the roof of the car. During the landing, the motors are cut off by a signal from the down-facing laser range finder, or the whole landing is aborted due to a low height threshold being met (1.5 m above the ground). If the car is lost from sight during any of the previously mentioned phases of the landing, the state machine transitions to the *Ascend and repeat* state. In the *Ascend and repeat* state, the UAV ascends while it follows the car prediction based solely on the estimate. If the car is not detected again, the state machine transitions back to the *Fly to waiting point* state. If the car is detected while in the state *Ascend and repeat*, the horizontal alignment process is repeated via the state *Align horizontally*.

## 6 Ground vehicle state estimation and prediction

Several sub-problems have to be solved to follow a moving object with an autonomous helicopter. The first part of the pipeline, which provides visual detection of the landing surface, was presented in section 4. Motion estimation is necessary to compensate for inherent flaws in the data that are extracted from camera images. Information provided by the *landing pattern detector* is naturally skewed by phenomena such as *signal noise*, *false positive detections*, *irregular detection rate* and *time delay*. These issues are common for most real-world sensors and are usually addressed by filtration and fusion with other available data. Moreover, since the dynamical system of the vehicles is known and can be described by a mathematical model, we can use the knowledge to maximize the information we gain from camera observations of the car. In particular, we can estimate unknown states that are difficult or even impossible to measure directly, namely velocity, heading and curvature of the turn. Estimation of hidden states further allows us to predict the future movement of the vehicle.

### 6.1 LKF with a liner model

The simplest model that can be used to estimate and predict the motion of the car is a linear model of 2nd order translational dynamics. This model does not impose any constraints on the holonomy of the system,

and therefore lacks an estimate of the turning radius ( $1/K$ , where  $K$  is the turning curvature). States can easily be estimated using the Linear Kalman Filter as

$$\begin{aligned}\mathbf{x}_{[n+1]} &= \mathbf{A}\mathbf{x}_{[n]} + \mathbf{B}\mathbf{u}_{[n]}, \\ \mathbf{y}_{[n+1]} &= \mathbf{C}\mathbf{x}_{[n+1]} + \mathbf{D}\mathbf{u}_{[n]},\end{aligned}\tag{1}$$

where  $\mathbf{x}_{[n]} \in \mathbb{R}^n$  is the state vector and  $\mathbf{u}_{[n]} \in \mathbb{R}^k$  is the input vector in sample  $n$ . We assume that  $\mathbf{C} = \mathbf{I}$ ,  $\mathbf{D} = \mathbf{0}$ . The lateral motion of the car is captured by the matrices

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},\tag{2}$$

where the state vector is defined as  $\mathbf{x}_{[n]} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y})^T$ . Testing with the linear model demonstrated satisfactory performance during linear motion of the car, but showed a significant tracking error when the car was turning. The scenario is showcased in the video <http://mrs.felk.cvut.cz/jfr2018landing-video1>.

## 6.2 UKF with a car-like model

To improve the car state estimation for non-linear motion, a different model is required, e.g., the nonholonomic car-like model

$$\begin{aligned}\mathbf{x}_{[n+1]}^o &= \mathbf{x}_{[n]}^o + \dot{\mathbf{x}}_{[n]}^o \Delta t, \\ \dot{\mathbf{x}}_{[n+1]}^o &= \begin{pmatrix} \cos \phi_{[n+1]} \\ \sin \phi_{[n+1]} \end{pmatrix} v_{[n+1]}, \\ \phi_{[n+1]} &= \phi_{[n]} + \dot{\phi}_{[n]} \Delta t, \\ \dot{\phi}_{[n+1]} &= K_{[n]} v, \\ v_{[n+1]} &= v_{[n]} + a_{[n]} \Delta t, \\ K_{[n+1]} &= K_{[n]} + \dot{K}_{[n]} \Delta t,\end{aligned}\tag{3}$$

where  $\mathbf{x}_{[n]}^o = (x, y)^T_{[n]}$  is the position of the car in the global coordinate system,  $\phi_{[n]}$  is its heading,  $K_{[n]}$  is the curvature of its turn,  $v_{[n]}$  is its scalar velocity,  $a_{[n]}$  is its scalar acceleration, and  $\Delta t$  is the time difference. Car-like model better reflects the physics of the car motion thanks to adding non-holonomic constraints and effectively by coupling the heading with the curvature of its turn. Estimates of the heading of the vehicle allow its motion to be tracked while the onboard camera is oriented properly to maximize successful detection even, while the car is in turn. The Unscented Kalman Filter (UKF) (Wann and van der Merwe, 2000) was used as a filtration method and as a predictor. In contrast with LKF, UKF utilizes a general function as a form of model iteration. Covariance of the estimated hypothesis is transformed as a set of sampled points of the ellipsoid, which is later reconstructed using the known prior distribution of the points. Therefore, it is not required to differentiate the model function, as it would be using nowadays obsolete Extended Kalman Filter (EKF), which uses a linear approximation of the model. Due to the high speed of the car, UKF is needed for its robustness and better performance, comparing to EKF and LKF.

## 6.3 Ground vehicle state prediction

As we will discuss below in section 8, knowing the future trajectory of the ground vehicle is a key element in tracking its motion with the unmanned aircraft. To predict the future trajectory, the same model as for its state estimation is applied, creating a discrete, time-parameterized trajectory in 2D space. The output of the UKF estimator is directly used as an initial condition for the prediction. Additionally, thanks to the

knowledge of all states of the dynamical model in Equation (3), the predicted trajectory can be offset to compensate for the delay in the vision system. The prediction for  $n$  future steps takes form of

$$\mathbf{q}_{[n+1]} = f(\mathbf{q}_{[n]}), \forall n = 0, \dots, n-1, \quad (4)$$

where  $f()$  is the model function according to Equation (3),  $\mathbf{q}_{[n]}$  is a complete state vector defined as

$$\mathbf{q}_{[n]} = \left[ \mathbf{x}_{[n+0]}^o{}^T, \phi_{[n+1]}, K_{[n+1]}, v_{[n+1]}, a_{[n+1]} \right]^T \quad (5)$$

and  $\mathbf{q}_{[0]}$  is the initial condition provided in real-time by the UKF estimator.

In general, the car can ride with changes in the curvature and acceleration depending on the driver. However, the competition rules specify the shape of the track in the arena (see Figure 1), and also the speed profile, which allows us to bias the estimate or the prediction to achieve more accurate trajectory tracking.

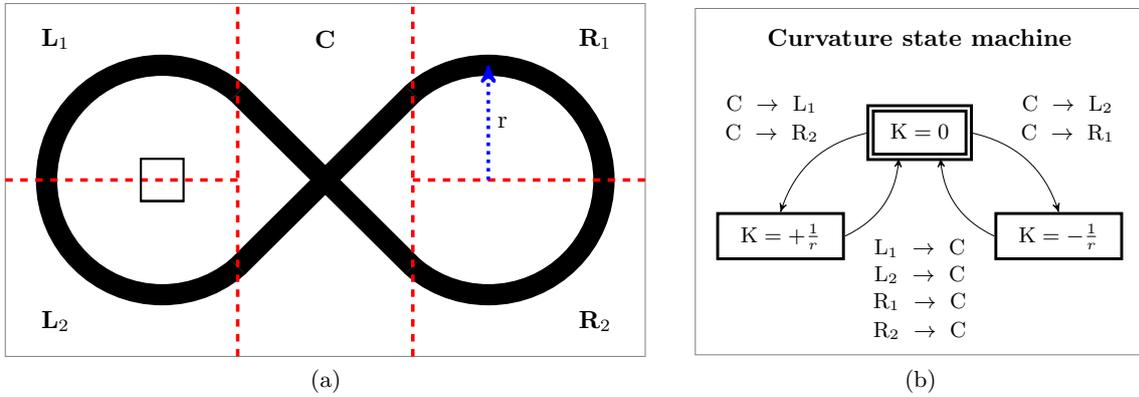


Figure 10: (a) The competition arena is divided into five areas. The center area (C) contains track with zero curvature. The four corner areas (L<sub>1</sub>, L<sub>2</sub>, R<sub>1</sub>, R<sub>2</sub>) contain track with curvature  $|K| = 1/r$ . (b) State machine producing the bias for the curvature of the car. Inputs to the state machine are transitions of the car between the different parts of the arena.

The first level of biasing the curvature relies solely on the known curvature in different parts of the track. Knowledge of coordinates of only a few pre-defined points is required to identify the curvature for any given coordinates on the map. To bias the curvature, the arena is divided into five separate parts. Figure 10a shows the partitioning into the corner parts L<sub>1</sub>, L<sub>2</sub>, R<sub>1</sub>, R<sub>2</sub>, where the curvature of the turn is  $|K| = 1/r$ , and the center area C, where  $K = 0$ . However, the sign of the curvature depends on the direction in which the car is driving in the particular part. To solve the estimation of the direction, we designed a simple state machine (Figure 10b), which describes all possible transitions between the different parts of the map that correspond to a change in the curvature. Later, when creating the prediction of the car using the model (Equation 3), the initial curvature  $K_{[0]}$  is set based on the current state of the curvature state machine. Further states of the prediction undergo the same process with an identical temporary state machine, to ensure that the curvature is correctly biased throughout the whole future.

#### 6.4 Biasing the predicted trajectory to the measured track

To further improve the performance of the tracking in the particular scenario of the competition, the predicted trajectory of the car was biased towards the known global GPS coordinates of the track. The predicted trajectory of the car is snapped towards the analytically described track. The snapping is proportional to the covariance of the predicted points. This leaves space for the car to drive off center of the track since the initial part of the prediction is close to the estimate. Figure 11 also illustrates a typical example of a predicted and snapped trajectory.

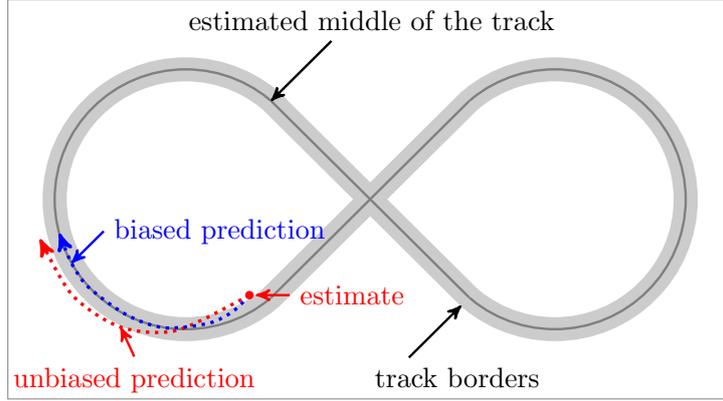


Figure 11: Scheme of the real track, on which the car was supposed to drive, with the path denoted in the middle of the track, to which the car prediction was biased. An example of a predicted (red) and a biased trajectory (blue) is shown.

## 7 UAV state estimation

Autonomous UAV control relies on an estimate of the states of the UAV dynamical system. Namely, knowledge of position and velocity (both vertical and horizontal) is required to control the movement for precise landing on the moving vehicle. Our platform is equipped with several independent sources of information, which are fused to obtain a single, reliable and smooth estimate of the UAV pose. It is essential to ensure smoothness of the resulting signal since SO(3) (Section 9) state feedback is sensitive to noise.

The main source of data for both the vertical axis and the horizontal axis in the proposed system is the PixHawk flight controller. Its Extended Kalman Filter fuses traditional inertial sensors – a three-axis accelerometer and a gyroscope with an height pressure sensor and a GPS receiver. Although the aircraft is already capable of autonomous flight with this off-the-shelf setup, we make use of other sensors, a time-of-flight laser rangefinder (Ruffo et al., 2014) and a differential GPS receiver, to provide more precise localization,

### 7.1 Horizontal position estimation

Position estimation in the lateral axes is based on the estimate provided by PixHawk, namely position  $\mathbf{x}^p$ , and velocity  $\dot{\mathbf{x}}^p$ . Although its precision may be satisfactory locally for short periods of time, it is prone to heavy drift in time spans of minutes. To correct this drift and thus to ensure repeatability of the experiments and, e.g., locating the dropping zone, the horizontal position from PixHawk is corrected by differential RTK GPS. Position measurements from the RTK GPS receiver are fused using the Linear Kalman Filter with the model

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \Delta t \\ \Delta t \end{pmatrix}, \quad (6)$$

where  $\mathbf{x}_{[n+1]}^e = \mathbf{A}\mathbf{x}_{[n]}^e + \mathbf{B}\mathbf{u}_{[n]}$  is the linear system equation,  $\mathbf{x}_{[n]}^e = (x, y)_{[n]}^T$  is the state vector finally used for control, and  $\mathbf{u}_{[n]}$  is the system input. According to our experience,

$$\mathbf{x}_{[0]}^p + \sum_{n=0}^k \dot{\mathbf{x}}_{[n]}^p \Delta t_{[n]} = \mathbf{x}_{[k]}^p, \forall k \in \mathbb{N} \quad (7)$$

does not hold for the position and velocity estimate provided by PixHawk. This is a very useful observation for somebody building a fully autonomous UAV system using an off-the-shelf controller. The input vector  $\mathbf{u}$  consists of velocities obtained by integrating differentiated positions  $\mathbf{x}^p$ , which ensures that our filter does not introduce any more drift into the resulting estimate when no RTK GPS corrections are involved. In

situations when the position is not being corrected, the resulting estimate follows the same relative state trajectory as  $\mathbf{x}^p$ , just shifted according to the latest correction.

In other words, the position estimate fused by the PixHawk is used as a main source of information, regardless of whether the RTK GPS is currently available. The difference is, the PixHawk position estimated is updated by input

$$\mathbf{u}_{[k]} = \left( \mathbf{x}_{[k]}^p - \mathbf{x}_{[k-1]}^p \right) / \Delta t_{[k]} \quad (8)$$

which results in position update

$$\mathbf{x}_{[k+1]}^e = \mathbf{x}_{[k]}^e + \left( \mathbf{x}_{[k+1]}^p - \mathbf{x}_{[k]}^p \right), \quad (9)$$

that follows the PixHawk estimate when left uncorrected. However, when the RTK GPS is available, the estimate  $\mathbf{x}_{[k+1]}^e$  can be freely corrected by the LKF, effectively adding an offset using the more precise source of information. By using such approach, the correction will be still applied even during long outages of the RTK GPS system.

## 7.2 Vertical position estimation

UAV state estimation relies less on the PixHawk in the vertical axis than in the horizontal axis. Height corrections come not only from differential RTK GPS but also from the down-facing TeraRanger rangefinder and from the landing pattern detector, which can provide height data when flying above the car. The estimator provides an option to switch between these sources of data, depending on the current state of the landing state machine. The PixHawk height is fused by the same technique as in the horizontal system, as denoted by Equation 8.

It is feasible to correct the height using the TeraRanger rangefinder when flying above uneven ground, but it cannot be used reliably when the down-facing sensor is obstructed by the car. However, RTK GPS can provide precise relative height measurements, but only when RTK FIX has been established. RTK FIX is one of several precision states of RTK GPS, which provides the best accuracy and guarantees a correct position signal. The standard states include RTK FLOAT and DGPS, but only RTK FIX guarantees the precision need for actually correcting the built-in PixHawk GPS. Finally, correcting the height using data from the *landing pattern detector* might bring unexpected steps in the signal due to false positive detections or signal dropouts. Since none of the additional sources is completely reliable, we employed a safety mechanism for detecting anomalies, which can toggle off any of the above-mentioned sensors.

## 8 Predictive trajectory tracking

While the state feedback described in section 9 provides precise position and velocity control, it requires a smooth and feasible reference. The reference consists of all states of the translational dynamics – position, velocity, and acceleration it is provided at 100 Hz, the same rate as the resulting control signal. There are various ways of creating the reference. Typically, thanks to the differential flatness of the UAV dynamical system, QP optimization can be performed to find a polynomial, given the initial and final state conditions (Mellinger and Kumar, 2011), which can then be derived and sampled to create the reference. In our case, we chose to generate the reference using a Model Predictive Control approach. The following text describes a simpler variant of the original approach designed for multiple vehicles, which was proposed in (Baca et al., 2018).

The Model Predictive Control Tracker (MPC Tracker) uses a QP formulation of the minimal sum-of-squares problem, where the optimal control action  $\mathbf{u}$  is found for a future prediction horizon of states

$\mathbf{x}_{[n]} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z})_{[n]}^T$  by minimizing the function

$$\begin{aligned} V(\mathbf{x}_{[0, \dots, m-1]}, \mathbf{u}_{[0, \dots, m-1]}) &= \frac{1}{2} \sum_{i=1}^{m-1} \left( \mathbf{e}_{[i]}^T \mathbf{Q} \mathbf{e}_{[i]} + \mathbf{u}_{[i]}^T \mathbf{P} \mathbf{u}_{[i]} \right), \\ \text{s.t.} \quad & \mathbf{x}_{[0, \dots, m-1]} \geq \mathbf{x}_L, \\ & \mathbf{x}_{[0, \dots, m-1]} \leq \mathbf{x}_U, \end{aligned} \quad (10)$$

where  $\mathbf{e}_{[n]} = \mathbf{x}_{[n]} - \tilde{\mathbf{x}}_{[n]}$  is the control error,  $\tilde{\mathbf{x}}_{[n]}$  is the setpoint for the MPC,  $m$  is the length of the prediction horizon, and  $\mathbf{x}_L$  and  $\mathbf{x}_U$  represent box constraints on states. The control error  $\mathbf{e}_{[n]}$  requires the formation of a general prediction of  $\mathbf{x}_{[n]}$ , which has been described in (Baca et al., 2016). In our case, the optimized control action is not directly used to control the real UAV. Instead, it controls a model of the UAV translational dynamics in real-time simulation. States of the simulated model are then sampled at 100 Hz to create the reference for the state feedback.

An important notion is a difference between the trajectory setpoint  $\tilde{\mathbf{x}}$  and the reference, which is generated by the MPC tracker. The trajectory setpoint  $\tilde{\mathbf{x}}$  is provided by high-level planning or, in this case, by the car predictor. No requirements are imposed on  $\tilde{\mathbf{x}}$  in general. By contrast, the reference produced by the MPC tracker is feasible, satisfies UAV dynamics and state constraints, and serves as a control reference for the SO(3) state feedback (see section 9). The inherent predictive nature of MPC provides trajectory tracking optimizing actions over the future, and this makes it ideal for tracking moving targets.

The simulated model is an LTI system covering the 3rd order translational dynamics of the UAV with the system matrices

$$\mathbf{A} = \begin{pmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta t \end{pmatrix}, \quad (11)$$

where  $\Delta t = 0.01$  s. The same matrices are used to formulate the MPC prediction. In our MPC formulation,  $\Delta t$  is different for the first iteration ( $\Delta t = 0.01$  s) and for all the other iterations ( $\Delta t = 0.2$  s). This allows the simulation to be controlled smoothly if the MPC is executed at 100 Hz, while there is a relatively sparse distribution of further states. Sparse distribution provides a much longer prediction horizon than these would normally be with  $\Delta t$  being constant. As in traditional MPC, only the control action in the first step is used to control the model in the simulation. Before action from the second step would be required, a new instance of the optimization task is formulated and solved, starting from new initial conditions. This results in a fresh control action for the next simulation step. This method is valid only if the MPC can be solved repeatedly within the 0.01 s simulation step.

Penalization parameters  $\mathbf{Q}$  and  $\mathbf{P}$  in (10) have been found empirically as

$$\mathbf{Q} = \begin{pmatrix} 5000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 800 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 800 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 800 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 800 & 0 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} 500 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 500 \end{pmatrix}. \quad (12)$$

As in our previous work (Baca et al., 2016), we used the *move blocking* technique to effectively prolong the prediction horizon while maintaining the computational complexity. The particular control action distribution for the MBZIRC competition was

$$\mathbf{U} = (1 \ 1 \ 1 \ 1 \ 1 \ 5 \ 5 \ 5 \ 5 \ 5 \ 10), \quad (13)$$

which results in an 8 s prediction horizon with only 33 variables in the optimization task. Without *move blocking* 120 variables would be required to solve the control problem.

As defined in Equation (10), MPC handles state constraints as linear inequalities. We impose maximum acceleration and velocity box constraints on the UAV to ensure safe and feasible resulting trajectories. The optimization being solved lies in the family of Linearly Constrained Quadratic Programming, which acquires a global optimum in a convex polytope. A custom solver based on sequential closed-form solution (Algorithm 1) is implemented to ensure guaranteed real-time performance, while sacrificing optimality. According to (Rossiter, 2003), in control design we should, if possible, rely on input pre-shaping rather than more complicated control-oriented solutions. Thus in every iteration, a reference trajectory is first pre-shaped to satisfy the velocity constraints by a low-pass filter as well as to initiate it in the current state of the UAV. Then the unconstrained problem is solved analytically as in (Baca et al., 2016). Although the resulting trajectory approximately satisfies the velocity constraints, the acceleration constraints are in general violated. In the second step, the acceleration part of the optimized trajectory is again pre-shaped to satisfy the acceleration constraints. Then we solved the unconstrained MPC again, however now with acceleration double-integrated to serve as the new position reference. The result of the second MPC step is a solution which, according to our empirical results, satisfies both acceleration and position constraints within 10% margin of error, which is a tolerable trade-off for complete control over the deterministic execution of the algorithm.

---

**Algorithm 1** Sequential closed-form MPC

---

```

1: procedure ITERATEMPC
2: input:
3:   reference  $\leftarrow$  desired reference
4:   current_state  $\leftarrow$  current state of the UAV
5:   max_v  $\leftarrow$  maximum velocity constraint
6:   max_a  $\leftarrow$  maximum acceleration acceleration
7: execution:
8:   # the first iteration of the MPC generates trajectory not violating velocity constraints
9:   reference  $\leftarrow$  preshapeVelocity(current_state, reference, max_v)
10:  trajectory, control_input  $\leftarrow$  analyticMPC(current_state, reference)
11:
12:  # the second iteration of the MPC generates trajectory not violating acceleration constraints
13:  reference  $\leftarrow$  preshapeAcceleration(current_state, trajectory, max_a)
14:  trajectory, control_input  $\leftarrow$  analyticMPC(current_state, reference)
15:
16:  return trajectory

```

---

MPC-based trajectory tracking operates in two modes. The first simple positioning mode, used mainly for short-distance position changes, accepts either relative or absolute position commands and tries to reach a given position in the fastest way with respect to the MPC scheme. The second trajectory-following mode utilized by *high-level trajectory planning* uses a precomputed path plan. It tries to track the trajectory precisely while respecting the plan waypoints schedule, which is crucial for precise landing on the moving vehicle.

## 9 Feedback control

The position controller uses the estimated state as feedback to follow the trajectories given as an output of the high-level trajectory planner. In many previous works, a backstepping approach is used for UAV control, because the attitude dynamics can be assumed to be faster than the dynamics governing the position, so linearized controllers are used for both loops (Mellinger et al., 2013; Weiss et al., 2011; Heriss et al., 2012). However, we need the system to be capable of large deviations from the hover configuration during operations like fast mapping of objects, or for heavy wind compensation. We therefore use a nonlinear controller. Let us consider an inertial reference frame denoted by  $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$  and a body reference frame centered in the center of mass of the vehicle with an orientation denoted by  $\mathbf{R} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ , where  $\mathbf{R} \in SO(3)$ . The dynamic model of the vehicle can be expressed as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + mg\mathbf{e}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, \\ \mathbf{J}\hat{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} &= \mathbf{M},\end{aligned}\tag{14}$$

where  $\mathbf{x} \in \mathbb{R}^3$  is the Cartesian position of the vehicle expressed in the inertial frame,  $\mathbf{v} \in \mathbb{R}^3$  is the velocity of the vehicle in the inertial frame,  $m \in \mathbb{R}$  is the mass,  $f \in \mathbb{R}$  is the net thrust,  $\boldsymbol{\Omega} \in \mathbb{R}^3$  is the angular velocity in the body-fixed frame, and  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  is the inertia matrix with respect to the body frame. The hat symbol  $\hat{\cdot}$  denotes the skew-symmetry operator according to  $\hat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ ,  $g$  is the standard gravitational acceleration, and  $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$ . The total moment  $\mathbf{M} \in \mathbb{R}^3$ , with  $\mathbf{M} = [M_1 \ M_2 \ M_3]^\top$ , along all axes of the body-fixed frame and the thrust  $\tau \in \mathbb{R}$  are control inputs of the plant. The dynamics of the rotors and propellers are neglected, and it is assumed that the force of each propeller is directly controlled. The total thrust,  $f = \sum_{j=1}^6 f_j$ , acts in the direction of the z axis of the body-fixed frame, which is orthogonal to the plane defined by the centers of the four propellers. The relationship between a single motor thrust  $f_j$ , the net thrust  $f$ , and the moments  $\mathbf{M}$  can be written as

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ sd & 1 & sd & -sd & -1 & -sd \\ -cd & 0 & cd & cd & 0 & -cd \\ -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}\tag{15}$$

where  $c = \cos(30^\circ)$ ,  $s = \sin(30^\circ)$  and  $d$  is the distance from the center of mass to the center of each rotor in the  $\mathbf{b}_1, \mathbf{b}_2$  plane. For non-zero values of  $d$ , eq. (15) can be inverted using the right pseudo-inverse.

For control, we build on the work in (Lee et al., 2013) and in (Mellinger and Kumar, 2011) with control inputs  $f \in \mathbb{R}$  and  $\mathbf{M} \in \mathbb{R}^3$  chosen as

$$\mathbf{M} = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} - \mathbf{J} \left( \hat{\boldsymbol{\Omega}} \mathbf{R}^T \mathbf{R}_c \boldsymbol{\Omega}_c - \mathbf{R}^T \mathbf{R}_c \dot{\boldsymbol{\Omega}}_c \right),\tag{16}$$

$$f = - \left( -k_x \mathbf{e}_x - k_{ib} \mathbf{R} \int_0^t \mathbf{R}(\tau)^T \mathbf{e}_x d\tau - k_{iw} \int_0^t \mathbf{e}_x d\tau - k_v \mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d \right) \cdot \mathbf{R}\mathbf{e}_3,\tag{17}$$

with  $\ddot{\mathbf{x}}_d$  the desired acceleration, and  $k_x, k_v, k_R, k_\Omega$  positive definite terms. We extend the referenced controllers by including two integral terms which accumulate error in the body frame and in the world frame, respectively. We include both terms to provide the opportunity to capture external disturbances (*e.g.*, wind) separately from internal disturbances (*e.g.*, an inefficient prop or a payload imbalance), particularly when the vehicle is permitted to yaw or rotate about the vertical axis. The thrust and the moments are then

converted to motor rates according to the characteristic of the proposed vehicle. Subscript  $C$  denotes a commanded value, and  $R_C = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$  is calculated as

$$\begin{aligned} \mathbf{b}_{2,des} &= [-\sin \psi_{des}, \cos \psi_{des}, 0]^\top, \quad \mathbf{b}_{3,C} = \frac{\mathbf{f}}{\|\mathbf{f}\|}, \quad \mathbf{b}_{1,C} = \frac{\mathbf{b}_{2,des} \times \mathbf{b}_3}{\|\mathbf{b}_{2,des} \times \mathbf{b}_3\|}, \quad \mathbf{b}_{2,C} = \mathbf{b}_3 \times \mathbf{b}_1, \\ \dot{\mathbf{b}}_{2,des} &= [-\cos \psi_{des} \dot{\psi}_{des}, -\sin \psi_{des} \dot{\psi}_{des}, 0]^\top, \quad \dot{\mathbf{b}}_{3,C} = \mathbf{b}_{3,C} \times \frac{\dot{\mathbf{f}}}{\|\mathbf{f}\|} \times \mathbf{b}_{3,C}, \\ \dot{\mathbf{b}}_{1,C} &= \mathbf{b}_{1,C} \times \frac{\dot{\mathbf{b}}_{2,des} \times \mathbf{b}_{3,C} + \mathbf{b}_{2,des} \times \dot{\mathbf{b}}_{3,C}}{\|\mathbf{b}_{2,des} \times \mathbf{b}_{3,C}\|} \times \mathbf{b}_1, \quad \dot{\mathbf{b}}_{2,des} = \dot{\mathbf{b}}_{3,C} \times \mathbf{b}_{1,C} + \mathbf{b}_{3,C} \times \dot{\mathbf{b}}_{1,C}, \\ \hat{\Omega}_C &= R_C^\top \dot{R}_C. \end{aligned} \tag{18}$$

Note that here we have to define  $\mathbf{b}_{2,des}$  based on the yaw, instead of defining  $\mathbf{b}_{1,des}$  as it was defined in (Mellinger and Kumar, 2011), due to a different Euler angle convention (we use the ZYX convention instead of ZXY). The definition of the tracking errors can be found in (Spurny et al., 2018).

## 10 Experimental evaluation

The platform was thoroughly tested during all stages of development. All features were developed and verified in simulation before experiments on the real hardware. The Gazebo simulator and the Robot Operating System allowed the same instances of software to be implemented and tested in simulation and also in the field. To ensure safety, all experiments were performed in unoccupied rural areas, and the UAV was supervised by a human operator at all times.

### Supplementary multimedia material

A video attachment to this work is available at website <http://mrs.felk.cvut.cz/jfr2018landing>.

#### 10.1 Experiments prior to the competition

The initial experiments involved testing the landing pattern detection and landing on a full-sized static target, as depicted in Figure 12a. State estimation based on the linear model of the moving target showed that following a non-linear motion requires a more descriptive car-like model. Figures 12a and 12b show the UAV following a car with the visual marker attached to its roof. Figure 13 depicts a test using a Linear Kalman Filter and a linear motion prediction of the future trajectory of the car. During numerous test runs, the UAV was able to follow the ground vehicles. However a significant position error was observed in turns of the path. The car-like model, which was tested later, exhibited significantly better performance than the linear model for following general trajectories, which include turning (see Figure 14).

The vision and guidance system were verified without taking of weather conditions into account. Although it was known that the competition would be held on concrete surface in summer weather, we tuned the landing pattern detection algorithm to various surfaces, including grass, snow, concrete and asphalt throughout the seasons to achieve maximal possible reliability and to account for unforeseen conditions during the competition. Figure 15 shows snapshots from experiments in winter conditions.

Experiments on autonomous landing, including the *Landing State Machine* (Section 5), are portrayed in Figure 14. Initially, the rate of a successful landing was 54% out in 22 trials. These experiments showed the need for a camera equipped with a SuperFisheye lens to improve pattern detection during the final stage of landing.



Figure 12: Photos from the first experiments in the field, a) shows a UAV hovering above a static target, b) shows a UAV tracking a vehicle moving along a linear trajectory, and c) depicts first attempts with tracking a ground vehicle moving in circles. The Linear Kalman Filter was used to estimate the states of the ground vehicle during this stage of development. Additional video material documenting these experiments can be found at <http://mrs.felk.cvut.cz/jfr2018landing-video1>.



Figure 13: Figures showing a UAV following the ground vehicle at a speed of 10 km/h. The motion of the car was estimated by the Linear Kalman Filter. The system was thoroughly tested on general trajectories of the ground vehicle. Video material for this experiment can be found at <http://mrs.felk.cvut.cz/jfr2018landing-video2>.



Figure 14: Experiments aimed at the autonomous following and landing on the ground vehicle using car-like motion model. Various stages of the Landing State Machine (see Section 5) are shown, a) the UAV tracking the car as it aligns horizontally, b) the descending phase, c) the UAV aligns for the second time before turning off the propellers and d) the UAV after the successful landing. Video summary of the experiments can be found at <http://mrs.felk.cvut.cz/jfr2018landing-video3>.



Figure 15: The performance of the system was tested in all weather conditions. Experiments in snowy and desert environments helped to fine-tune the computer vision for various lighting conditions and ground textures. A video showing a landing in a desert car be seen at <http://mrs.felk.cvut.cz/jfr2018landing-video4>.

## 10.2 The competition trials

Each team that participated in the competition in Abu Dhabi had an option to take part in two rehearsal trials ( $2 \times 30$  minutes), two competition trials ( $2 \times 15$  minutes) and two rounds of the Ground challenge



Figure 16: A sequence of images from the second trial during the competition. The whole videos, as well as additional material including onboard footage, can be found at <http://mrs.felk.cvut.cz/jfr2018landing>.

( $2 \times 25$  minutes). Our team competed in autonomous mode in both trials of the first challenge and the first round of the Grand Challenge. The second round of the Grand challenge was performed in manual mode, to allow manual operation on the ground robot at the same time. A combined manual and autonomous mode for different robots in the same trial was not allowed. In both trials of the landing challenge, we experienced a successful landing. Touchdown during trial 1 took place 143.2 s after the start. In trial 2 it took the UAV 84.6 s to land, which brought us the second place among all teams in the autonomous landing challenge, just behind the UAV of Beijing Institute of Technology, with a time of 63.4 s. Figure 16 depicts the autonomous landing using the proposed system during the first trial of the competition. Table 1 compares our results with these of other teams.

Independently of the three separate robotic challenges, in the Grand Challenge, the teams competed in all three challenges simultaneously. During the Grand Challenge, our system scored the fastest landing ever performed among all teams in the entire MBZIRC competition, with 25.1 s time from the start. Figure 19 shows the relative position and control error plots from the fastest trial. Figure 17 shows top-down plots of two trials from the landing challenge, and also the first trial from the Grand Challenge. The same trials are presented in Figures 18a, 18b and 18c, with the states of the landing state machine (Section 5) color-coded in the trajectory of the UAV.

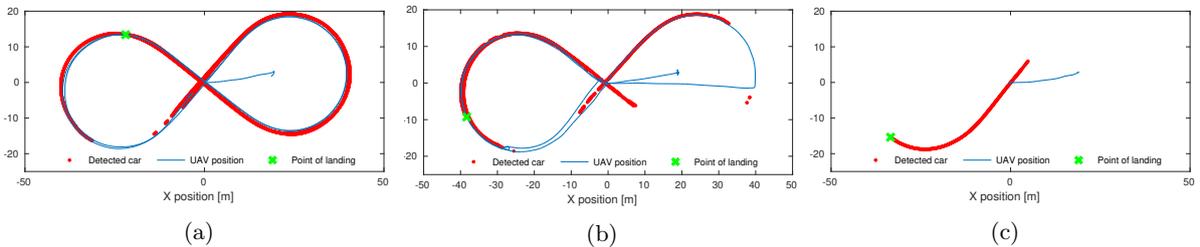


Figure 17: Top-down view of all three successful trials: (a) 1:44 min flight, (b) 1:28 min flight, (c) 0:25 min flight.

Team	TRIAL 1	TRIAL 2	GRAND 1	GRAND 2
Beijing Institute of Technology	63,4	63,4	NQ	NQ
<b>CTU in Prague, UPENN and UoL</b>	143,2	84,6	<b>25,1</b>	M
University of Bonn	110,5	<i>not landed</i>	58,6	42,3
University of Catania	134,5	<i>falling off</i>	NQ	NQ

Table 1: Time in seconds of all successful autonomous landing attempts in the MBZIRC 2017 competition, when the car was moving at its maximum speed of 15 km/h. NQ - not qualified for the final challenge, M - manual mode applied due to other robots in the arena, *not landed* - UAV not landed due to a crash or time limit of 15 minutes exceeding, *falling off* - UAV touched the mobile landing platform but was not able to fix there, TRIAL 1 & 2 - trials of the MBZIRC Challenge 1, GRAND 1 & 2 - trials of the MBZIRC Grand Challenge.

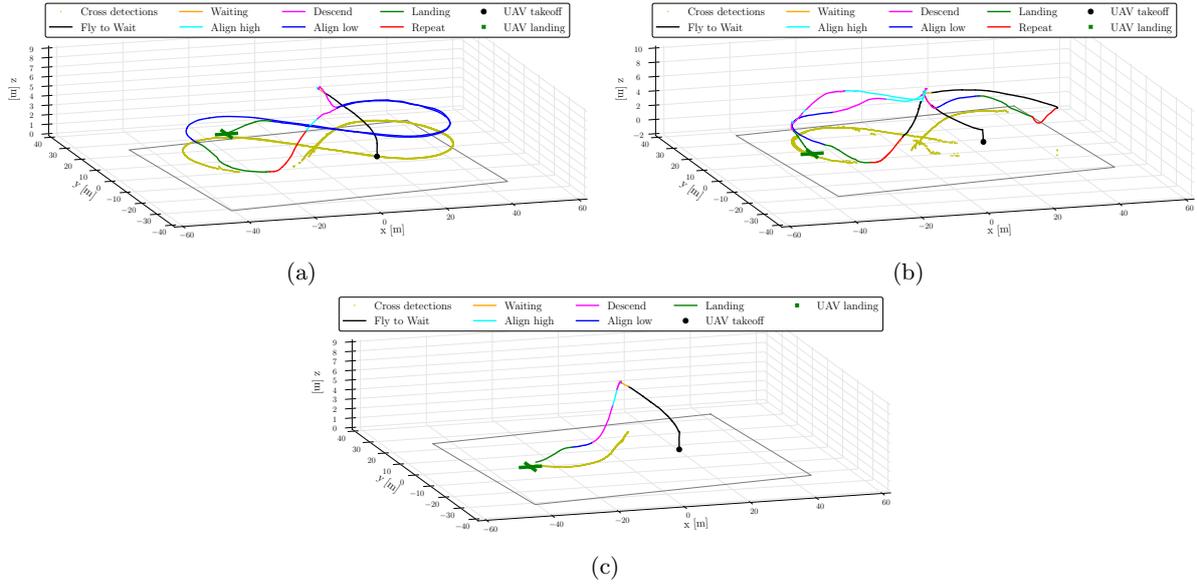
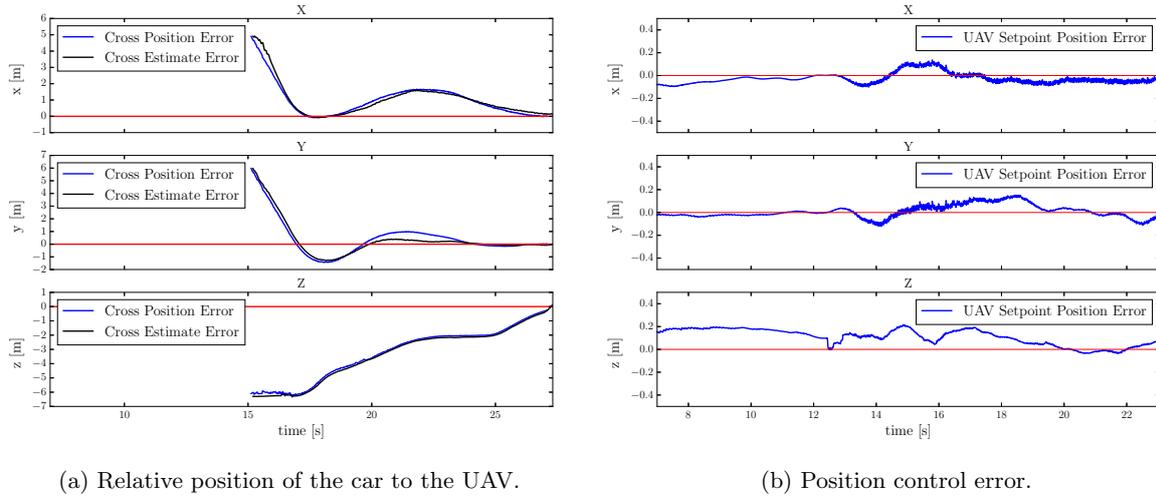


Figure 18: Three-dimensional plot of positions of the ground vehicle, as detected by the UAV, and the UAV during the (a) first competition trial, (b) second competition trial and (c) first Grand challenge trial. The trajectory of the UAV is color-coded according to the states of the *Landing State Machine*.



(a) Relative position of the car to the UAV.

(b) Position control error.

Figure 19: Plots of (a) the relative position of the detected and estimated ground vehicle, and the position of the UAV during the third (the fastest) landing and (b) the position control error during the flight.

## 11 Lessons learned

Although the competition results can be considered a major success, it was not without hurdles, mainly during implementation and testing of the proposed system. The proposed control pipeline consisting of estimator, predictor, tracker, and controller showed to be depended mainly on the performance of the car estimator. Tuning of the estimator parameters on real data was an essential factor which influences the overall performance of the remaining components in the pipeline. Hence we stress the significance of the real-world outdoor experiments above simulation, to obtain real sensor data. Finally, we cannot stress enough how important is the team and the dedication of the individual members of the team. We value the skill of

*getting things done* and a capability of delivering performance on time when it is needed together with the mindset of a scientist. The ability to transfer scientific concepts in robotics to a working prototype is vital for experimental evaluation such as the one in this competition.

### 11.1 Toward a more general solution

Despite our best effort to develop a general solution capable of autonomous landing on a moving car, couple sub-systems have been tailored specifically to the competition scenario. The computer vision system was designed to locate and track the landing pattern specified by the rules of the competition. We can speculate that similar pattern could be used in practice to mark a landing spot on a vehicle, which is designed to receive the UAV. In such case, the proposed vision system would be a viable option. However, in the case of an unmarked and possibly arbitrary vehicle, a different approach to localization and tracking would be required, e.g., based on nowadays popular artificial neural networks. Estimation and prediction of the car movement using a non-linear car-like model provide a framework suitable for tracking and landing on most common vehicles. A more precise model could be used to better estimate state of a specific vehicle. Our approach to bias the prediction of the future car movement based on the known parameters of the arena is optional and can be omitted in the case of a general area and car trajectory. Moreover, all field experiments, before the competition trials, were performed without particular bias towards the known trajectory of the car. See Figure 13 for an example of such experiment. The presented state machine is also designed around the competition scenario, however, the need to customize it is apparent. Besides those two cases, the presented approach can be applied to a general scenario of locating, tracking and landing on a vehicle in an outdoor environment.

## 12 Conclusion

We proposed, developed and experimentally evaluated an unmanned aerial system for autonomous landing on a fast-moving vehicle. The solution described in this paper is a multirotor helicopter platform equipped with sensors and a computer, capable of onboard image processing and state estimation of the ground vehicle and also predictive planning and automatic control. Images from an onboard camera are processed online to extract the position of the landing marker on top of the vehicle. States of the car-like dynamical model are estimated using the Unscented Kalman Filter, based on image processing and the known parameters of the car trajectory. The same model is then used to predict the future trajectory of the vehicle. The Model Predictive Control tracker creates an optimal feed-forward reference in third order dynamics based on the predicted trajectory. Non-linear  $SO(3)$  state feedback controls the UAV along the reference. A state machine controls the UAV from takeoff, through finding the target and tracking it, to finally turning off its propellers during the touchdown. The proposed system utilizes state-of-the-art techniques from control engineering in a unique combination to solve the difficult challenge, which only a handful of teams from all over the world were able to tackle. The proposed control pipeline relies on the novel MPC Tracker, which was proposed specifically for this challenge (Baca et al., 2018), to achieve the high accuracy of the autonomous landing.

The system was extensively tested in the course of more than one year of development. The experiments showed that the UAV is capable of autonomous tracking and landing on a car moving at a speed of 15 km/h. In the MBZIRC 2017 challenge, the system proved to be robust by successfully landing in both trials of the challenge 1 of the competition. During the Grand Challenge of the competition, it landed in 25 s, which is the shortest time among all teams during the entire competition, which may be considered as a relevant and objective benchmark of this task.

### Acknowledgments

The outstanding results of this project could not have been achieved without the full cooperation of each member of our team, comprising people from the Czech Technical University in Prague, the University of

Pennsylvania and the University of Lincoln, UK (Figure 20). The work has been supported by CTU grant no. SGS17/187/OHK3/3T/13, the Grant Agency of the Czech Republic under grant no. 17-16900Y, the Office of Naval Research Global. Grant Numbers: N000140710829, N000141410510, by Army Research Laboratory. Grant Number: W911NF1720181, and by Khalifa University of Science, Technology and Research. Grant Number: MBZIRC 2017.



Figure 20: Team members after the award ceremony of the MBZIRC competition in Abu Dhabi, United Arab Emirates.

## References

- Araar, O., Aouf, N., and Vitanov, I. (2017). Vision based autonomous landing of multirotor uav on moving platform. *Journal of Intelligent & Robotic Systems*, 85(2):369–384.
- Baca, T., Hert, D., Loianno, G., Saska, M., and Kumar, V. (2018). Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
- Baca, T., Loianno, G., and Saska, M. (2016). Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles. *International Conference on Methods and Models in Automation and Robotics (MMAR)*.
- Benini, A., Rutherford, M. J., and Valavanis, K. P. (2016). Real-time, gpu-based pose estimation of a uav for autonomous takeoff and landing. In *International Conference on Robotics and Automation (ICRA)*, pages 3463–3470. IEEE.
- Beul, M., Houben, S., Nieuwenhuisen, M., and Behnke, S. (2017). Fast autonomous landing on a moving target at mbzirc. In *The European Conference on Mobile Robotics (ECMR)*. IEEE.
- Bi, Y. and Duan, H. (2013). Implementation of autonomous visual tracking and landing for a low-cost quadrotor. *International Journal for Light and Electron Optics*, 124(18):3296–3300.
- Borowczyk, A., Nguyen, D.-T., Phu-Van Nguyen, A., Nguyen, D. Q., Saussié, D., and Le Ny, J. (2017). Autonomous landing of a quadcopter on a high-speed ground vehicle. *Journal of Guidance, Control, and Dynamics*, 40:2378–2385.
- Fu, M., Zhang, K., Yi, Y., and Shi, C. (2016). Autonomous landing of a quadrotor on an ugv. In *International Conference on Mechatronics and Automation (ICMA)*, pages 988–993. IEEE.

- Ghamry, K. A., Dong, Y., Kamel, M. A., and Zhang, Y. (2016). Real-time autonomous take-off, tracking and landing of uav on a moving ugv platform. In *24th Mediterranean Conference on Control and Automation (MED)*, pages 1236–1241. IEEE.
- Ghommam, J. and Saad, M. (2017). Autonomous landing of a quadrotor on a moving platform. *Transactions on Aerospace and Electronic Systems*, 53:1504–1519.
- Guo, Z. and Hall, R. W. (1989). Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373.
- Heriss, B., Hamel, T., Mahony, R., and Russotto, F. X. (2012). Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *Transactions on Robotics*, 28(1):77–89.
- Hoang, T., Bayasgalan, E., Wang, Z., Tsechpenakis, G., and Panagou, D. (2017). Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle. In *American Control Conference (ACC)*, pages 5580–5585. IEEE.
- Hui, C., Yousheng, C., Xiaokun, L., and Shing, W. W. (2013). Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision. In *Chinese Control Conference (CCC)*, pages 5895–5901. IEEE.
- Jin, S., Zhang, J., Shen, L., and Li, T. (2016). On-board vision autonomous landing techniques for quadrotor: A survey. In *35th Chinese Control Conference (CCC)*, pages 10284–10289. IEEE.
- Jung, W., Kim, Y., and Bang, H. (2016). Target state estimation for vision-based landing on a moving ground target. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 657–663. IEEE.
- Jung, Y., Lee, D., and Bang, H. (2015). Close-range vision navigation and guidance for rotary uav autonomous landing. In *International Conference on Automation Science and Engineering (CASE)*, pages 342–347. IEEE.
- Kim, J., Jung, Y., Lee, D., and Shim, D. H. (2014). Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1243–1252. IEEE.
- Kong, W., Zhou, D., Zhang, D., and Zhang, J. (2014). Vision-based autonomous landing system for unmanned aerial vehicle: A survey. In *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, pages 1–8. IEEE.
- Lee, D., Ryan, T., and Kim, H. J. (2012). Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In *International Conference on Robotics and Automation (ICRA)*, pages 971–976. IEEE.
- Lee, H., Jung, S., and Shim, D. H. (2016). Vision-based uav landing on the moving vehicle. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1–7. IEEE.
- Lee, T., Leok, M., and McClamroch, N. H. (2013). Nonlinear robust tracking control of a quadrotor uav on  $se(3)$ . *Asian Journal of Control*, (2):391–408.
- Lin, S., Garratt, M. A., and Lambert, A. J. (2017). Monocular vision-based real-time target recognition and tracking for autonomously landing an uav in a cluttered shipboard environment. *Autonomous Robots*, 41(4):881–901.
- Masselli, A., Yang, S., Wenzel, K. E., and Zell, A. (2014). A cross-platform comparison of visual marker based approaches for autonomous flight of quadrocopters. *Journal of Intelligent & Robotic Systems*, 73(1-4):349–359.

- Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2012). Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1):21–39.
- Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *International Conference on Robotics and Automation (ICRA)*, pages 2520–2525. IEEE.
- Mellinger, D., Shomin, M., Michael, N., and Kumar, V. (2013). Cooperative grasping and transport using multiple quadrotors. In *The 10th International Symposium Distributed Autonomous Robotic Systems*, pages 545–558. Springer Berlin Heidelberg.
- Rossiter, J. A. (2003). *Model-based predictive control: a practical approach*. CRC press.
- Ruffo, M., Di Castro, M., Molinari, L., Losito, R., Masi, A., Kovermann, J., and Rodrigues, L. (2014). New infrared time-of-flight measurement sensor for robotic platforms. In *18th International Workshop on ADC Modelling and Testing*. IMEKO.
- Spurny, V., Baca, T., Saska, M., Penicka, R., Krajnik, T., Thomas, J., Dinesh, T., Loianno, G., and Kumar, V. (2018). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*.
- Stepan, P., Krajnik, T., Petrlik, M., and Saska, M. (2018). Vision techniques for on-board detection, following, and mapping of moving targets. *Journal of Field Robotics*, pages 1–18.
- Tan, C. K., Wang, J., Paw, Y. C., and Liao, F. (2016). Autonomous ship deck landing of a quadrotor using invariant ellipsoid method. *Transactions on Aerospace and Electronic Systems*, 52(2):891–903.
- Tersus-GNSS (2017). PRECIS-BX305 GNSS RTK Board. Available: <https://www.tersus-gnss.com> (cited on 2017-07-17).
- Wann, E. A. and van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158. IEEE.
- Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874.
- Xu, L. and Luo, H. (2016). Towards autonomous tracking and landing on moving target. In *International Conference on Real-time Computing and Robotics (RCAR)*, pages 620–628. IEEE.
- Yang, S., Scherer, S. A., and Zell, A. (2013). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, pages 1–17.
- Yang, S., Ying, J., Lu, Y., and Li, Z. (2015). Precise quadrotor autonomous landing with sruf vision perception. In *International Conference on Robotics and Automation (ICRA)*, pages 2196–2201. IEEE.

# K

---

KEY ARTICLE [11] - JOURNAL OF FIELD ROBOTICS 2018

---

©[2018] Wiley Online Library. This article was published in Journal of Field Robotics: V. Spurny, T. Baca, **M. Saska**, R. Penicka, T. Krajnik, J. Thomas, D. Thakur, G. Loianno, and V. Kumar: Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs, 2018.



# Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles

Vojtěch Spurný<sup>1</sup> | Tomáš Báča<sup>1</sup> | Martin Saska<sup>1</sup> | Robert Pěnička<sup>1</sup> |  
Tomáš Krajník<sup>2</sup> | Justin Thomas<sup>3</sup> | Dinesh Thakur<sup>3</sup> | Giuseppe Loianno<sup>4</sup> |  
Vijay Kumar<sup>3</sup>

<sup>1</sup>Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic

<sup>2</sup>Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic

<sup>3</sup>GRASP Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania

<sup>4</sup>Department of ECE and MAE, Tandon School of Engineering, New York University, New York City, New York

## Correspondence

Vojtěch Spurný, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic.  
Email: vojtech.spurny@fel.cvut.cz

## Funding information

České Vysoké Učení Technické v Praze, Grant/Award Number: SGS17/187/OHK3/3T/13; Office of Naval Research Global, Grant/Award Numbers: N00014-07-1-0829, N00014-14-1-0510; Army Research Laboratory, Grant/Award Number: W911NF-17-2-0181; Grantová Agentura České Republiky, Grant/Award Number: 17-16900Y; Research Center for Informatics project, Grant/Award Number: CZ.02.1.01/0.0/0.0/16\_019/0000765; Khalifa University of Science, Technology and Research, Grant/Award Number: MBZIRC 2017

## Abstract

This paper addresses the problem of autonomous cooperative localization, grasping and delivering of colored ferrous objects by a team of unmanned aerial vehicles (UAVs). In the proposed scenario, a team of UAVs is required to maximize the reward by collecting colored objects and delivering them to a predefined location. This task consists of several subtasks such as cooperative coverage path planning, object detection and state estimation, UAV self-localization, precise motion control, trajectory tracking, aerial grasping and dropping, and decentralized team coordination. The failure recovery and synchronization job manager is used to integrate all the presented subtasks together and also to decrease the vulnerability to individual subtask failures in real-world conditions. The whole system was developed for the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017, where it achieved the highest score and won Challenge No. 3—Treasure Hunt. This paper does not only contain results from the MBZIRC 2017 competition but it also evaluates the system performance in simulations and field tests that were conducted throughout the year-long development and preparations for the competition.

## KEYWORDS

aerial robotics, cooperative robots, mobile manipulation, planning

## 1 | INTRODUCTION

Small autonomous unmanned aerial vehicles (UAVs) are widely used in numerous applications of data collection due to their potential for rapid deployment and their ability to reach locations inaccessible by ground robots. While fixed wing UAVs have the advantage of stable flight at high speeds, long range, and long flight time, rotary wing UAVs (such as the popular multicopter helicopters) benefit from their capacity for high manoeuvrability, vertical take off and landing, flight in cluttered environments in close proximity to obstacles, and hovering in a desired position in a 3D environment. The ability to

precisely reach a desired 3D position and hover in place is crucial for long-term information gathering, and especially for physical interaction with objects in the workspace. Delivery applications composed of acquisition, transport, and drop-off provide an example requiring interaction with the environment during autonomous flight. This is the topic discussed in our paper.

A multiple cooperative delivery mission (called Treasure Hunt) was the most complex task in the 2017 Mohamed Bin Zayed International Robotics Challenge (MBZIRC<sup>1</sup>). In the competition, the

<sup>1</sup><http://www.mbzirc.com/> - Accessed: July 17, 2018.

delivery task was solved in its full complexity, including searching for objects with unknown positions, grasping moving objects, and cooperation among multiple UAVs working in concert. The deployment of a team of UAVs was motivated by the limited total mission time, and by including large objects with weights exceeding the maximum payload of the individual robots. In the mission, 23 objects (10 static, 10 dynamic, and 3 large) had to be localized in an outdoor arena and collected by three UAVs of limited size. While the small objects (static and dynamic) could be lifted by a single UAV, the large objects required two UAVs to transport them.

The system that exhibited the best performance among all participants in the MBZIRC competition in the Treasure Hunt challenge is presented in this paper<sup>2</sup>. The system design is driven by the specific task proposed and precisely specified by the organizers. The approach is tailored to provide high robustness and performance to solve the challenging task by modification of available robotic methods and designing new algorithms where necessary. Nevertheless, the proposed system is easily reusable in a large set of multi-UAV scenarios as shown in Section 1.2. The core of the system is the failure recovery and synchronization jobs manager (FSM), which is crucial for managing all subsystems and for coordinating all UAVs sharing the same workspace. The FSM is also needed to achieve the reliability required for the deployment of UAVs in real-world conditions, which requires the ability to recover from UAV failures and also from a malfunction of the localization and communication infrastructure. For example, the robots can easily collide with the objects being grasped due to a wind gust which, in combination with the ground effect, can create a hardly predictable external force on the UAV in the final phase of the approach to an object. Such a collision could result in a UAV crash, deadlock, or an overturned object. Moreover, malfunctions of UAV subsystems such as camera dropouts, incorrect range-finder measurements, gripper failure or gripper feedback failure, and imprecise object gripping, can be expected in demanding outdoor conditions. All these eventualities need to be considered by the system to enable undisturbed operation of the remaining robots in the event of a UAV failure, limited operation of a UAV with a faulty subsystem, or an unsuccessful or interrupted grasping task. From this point of view, the proposed FSM concept can be considered as a hierarchical state machine with included synchronization and failure recovery abilities, which may be effectively reused in any complex multi-UAV task involving environment interaction.

Although the rules of the MBZIRC competition allowed the use of global navigation satellite system (GNSS) and the even more precise differential global positioning system (DGPS) for UAV localization, the availability of these systems was not guaranteed. For example, GNSS information was available only intermittently, due to interference with other transmitters located at the competition site and occlusion of satellites by the surrounding

buildings and infrastructure. The provided Wi-Fi infrastructure was even less reliable and therefore the proposed FSM approach leverages the combination of different modes of the system based on the availability of Wi-Fi, GNSS, and DGPS. In addition to the FSM, a sensor fusion mechanism is presented for combining information from various onboard sensors (onboard IMU, GPS, DGPS, rangefinder, and camera) which must be considered as potentially unreliable at any time. It is vital that the UAV may continue with the task despite lacking some sensor data (e.g., precise measured altitude above ground), because the competition rules did not allow any human intervention or debugging during the trials, and which is also the case in most of the real-world UAV applications.

Another important subsystem, which is crucial in tasks requiring interaction with the environment, is relative detection and estimation of the state of the objects requiring interaction. In the presented system, the relative localization technique relies on onboard vision, since the objects in the competition were designed to support such an approach. The shape and color of the objects were specified before the mission and a color-based key was used to identify the score for collecting the particular object and to distinguish the object type. Static, dynamic, and long objects were labeled by different colors, all easily distinguishable from the background. Therefore, the vision approach is the simplest way to acquire all data required for the high-level planning (the score, type, and position estimate), and also for the visual servoing in the grasping task (precise relative positions of objects). However, any alternative relative localization system can be easily integrated based on the application. State estimation of the object is necessary mainly for dynamic objects, where a velocity estimate of the object needs to be taken into account by the UAV control modules.

Two flight behaviours are required in the Treasure Hunt task: Trajectory following and precise visual servoing. The trajectory tracking mode is used to search for the object in the environment, to approach the vicinity of the object, and to transport the object to the required location. The most important property of this controller is rapid and smooth movement along the trajectory provided by the high-level planning. The visual servoing applied in the final phase of grasping can be realized more slowly, but the requirements on precision are much higher. In the paper, we will present a novel model predictive control (MPC)-based approach that allows integration of the UAV state estimation (including external forces produced by the wind and ground effect) and target state estimation (a position and velocity estimate of the currently observed object), enabling our robots to reach the target with a maximum position error of 8 cm, which is determined by the diameter of the object and the size of the gripper.

## 1.1 | State-of-the-art

Rotorcraft or rotor-wing UAVs are suitable for tasks with object manipulation, due to their ability to hover on the spot. Their usage in

<sup>2</sup><http://mrs.felk.cvut.cz/projects/mbzirc> - Accessed: July 17, 2018.

this field has already been investigated in several publications, mainly for a single UAV, in particular subparts such as gripper or manipulator design, control techniques, and object detection.

The design of a manipulator for use in industrial applications, for aerial inspection by contact, and also for aerial manipulation is described in Fumagalli, Stramigioli, and Carloni (2016). The design of a multidegree arm manipulator placed on UAVs is presented in Morton and Toro (2016) and in Korpela, Danko, and Oh (2012). The idea of using a suction-based gripper for versatile aerial grasping is presented and experimentally verified in Kessens, Thomas, Desai, and Kumar (2016). Other gripper designs are presented in Mellinger, Lindsey, Shomin, and Kumar (2011) and Pounds, Bersak, and Dollar (2011b).

A study about determining stability bounds, in which the changing mass-inertia parameters of the system due to the grasped object will not destabilize a proportional-integral-derivative flight controller for helicopters, is presented in Pounds, Bersak, and Dollar (2011a). The authors of Thomas, Loianno, Polin, Sreenath, and Kumar (2014) introduce a controller and a planner for high-speed aerial grasping, using a quadrotor UAV with a claw-like gripper. Their approach is used for grasping a cylindrical object relying on feedback from a monocular camera and an inertial measurement unit onboard the aerial robot. Images from the camera are used for computing the desired pitch angle, and the remaining axes (roll and yaw) are controlled using feedback from the vision motion capture system. In Ghadiok, Goldin, and Ren (2012), a system for autonomous grasping of objects using a monocular IR camera is introduced. Detection of the objects is based on finding an IR beacon, which has to be placed on the objects. The authors also rely only on onboard sensors, but the position and yaw estimation is computed offboard on the ground station. A methodology for controlling a multiarm manipulating aerial vehicle is presented in Orsag, Korpela, Pekala, and Oh (2013). The control of a system where the control input is generated for the UAV and the manipulator joints simultaneously is described in Heredia et al. (2014), Kamel, Comari, & Siegwart (2016), and Kannan, Quintanar-Guzman, Dentler, Olivares-Mendez, & Voos (2016). The papers (Kim, Seo, Choi, & Kim, 2016; Lippiello et al., 2016; Santamaria-Navarro, Grosch, Lippiello, Sola, & Andrade-Cetto, 2017) present a vision guidance approach using an image-based visual servo for an aerial manipulator. A method for planning a time-optimal trajectory for a quadrotor with the goal of grasping a moving target is introduced in Spica, Franchi, Oriolo, Bühlhoff, and Giordano (2012). However, the solution is presented only by simulations.

Detecting and estimating the object is a challenging task that needs to be investigated for autonomous grasping. Online detection of the known object and estimation of its position using features from images are described in RamonSoria, Arrue, and Ollero (2017). Another method for onboard object extraction based on stereo vision for autonomous grasping of objects is presented in RamonSoria, Bevec, Arrue, Ude, & Ollero (2016). However, the aforementioned methods rely on stereo or depth sensors, which are not used on our UAVs. To detect the colored objects, we modified a computationally efficient method (Krajník et al., 2014), which already proved its reliability and accuracy in real-world conditions.

Ways of transporting large objects by multiple UAVs have already been investigated in Gioioso, Franchi, Salvietti, Scheggi, and Prattichizzo (2014), Mellinger, Shomin, Michael, and Kumar (2013) and Parra-Vega, Sanchez, Izaguirre, Garcia, and Ruiz-Sanchez (2013). A control scheme for cooperative simultaneous manipulation of an object by a team of UAVs is described in Parra-Vega et al. (2013). The idea of grasping and manipulating objects by a swarm of UAVs has been also studied in Gioioso et al. (2014), where the swarm is teleoperated using the free motion of a human hand. Both these works lack experimental verification, because the systems were tested only in simulations. Transport of large objects by multiple UAVs had been achieved in Mellinger et al. (2013). However, the experiments were done in an indoor environment under the Vicon<sup>3</sup> motion capture system.

Solutions for the Treasure Hunt scenario have already been presented by two teams participating in the MBZIRC competition which had worked on this scenario autonomously. The approach used by the team from ETH Zurich is described in Bähmann, Schindler, Kamel, Siegwart, and Nieto (2017), and the approach used by the team from the University of Bonn is presented in Nieuwenhuisen et al. (2017). Both teams relied on an electropermanent magnetic gripper for grasping ferrous objects, which are recognized using a color blob detection algorithm. They also used a similar approach for locating the objects. First, the arena is cooperatively searched by UAVs to create a map of the objects, and then an attempt is made to grasp and deliver each object in the map. However, the solution in Bähmann et al. (2017) relies on a Wi-Fi communication infrastructure, and the authors do not propose any alternative in the event of communication blackout. They also do not explain how they solve the problem of multiple UAVs coordination over the drop-off zone. In Nieuwenhuisen et al. (2017), the authors mention a conservative solution for a disturbed communication network. However, this solution is not explained in detail, and therefore, their approach cannot be directly replicated and evaluated. Furthermore, their controller does not compensate for external factors such as wind, which is a common disturbance in an outdoor environment.

## 1.2 | Contribution

The contribution of this paper correlates directly with the expected contribution of the MBZIRC challenge. A board of respected scientists<sup>4</sup> from leading robotic groups worldwide selected the Treasure Hunt scenario as the most challenging task in the MBZIRC event for numerous reasons. This scenario extends state-of-the-art systems in various ways: Deployment of multiple UAVs in the same outdoor workspace, multirobot scanning of the environment with no prior information on the position of objects, online distribution of tasks to UAVs based on the obtained information, and physical interaction with the environment. Indeed, physical interaction of UAVs with objects in an unknown outdoor environment, especially

<sup>3</sup><http://www.vicon.com/> - Accessed: July 17, 2018

<sup>4</sup><http://www.mbzirc.com/committee> - Accessed: July 17, 2018

cooperatively (some objects require the cooperation of multiple UAVs), is a challenging and innovative task, mainly if it must be solved in demanding windy environments, such as the MBZIRC 2017 venue in Abu Dhabi. The strong wind gusts present in the location between the coast and the desert significantly influence the precision and the stability of the UAV controllers, particularly in the final phase of object grasping, where they are combined with the ground effect. Further, the light conditions (e.g., the strong and variable sunshine) make the vision task more complicated than in a laboratory environment. The multirobot aspect requires rapid communication and coordination of UAVs, which seemed to be a bottleneck for the approaches presented by most of the other teams. Our solution to the challenges caused by unreliable communication is also a contribution to robotic research.

The overall contribution of our paper goes beyond the MBZIRC challenge, as it contains a comprehensive description of all components of the system that can be used in various collaborative multi-UAV missions, including physical interaction of robots and the environment. Although the system is primarily designed for outdoor deployment with a GNSS signal available, it can be used in GNSS-denied conditions with only a slight modification, since object grasping is realised by visual servoing, which relies on relative localisation only. Besides object grasping and delivery tasks, the system has been successfully deployed in numerous multi-UAV applications, including detection of sources of radiation and electromagnetic fields (Saska, 2017), inspection and documentation of historical sites (Saska, Kratky, Spurny, & Baca, 2017), reconnaissance and surveillance missions (Pěnička, Faigl, Váňa, & Saska, 2017; Pěnička, Saska, Reymann, & Lacroix, 2017), etc.

Another contribution of this paper for the robotic community is based on the fact that the next MBZIRC event intends to build on the achievements of MBZIRC 2017, and to propose even more challenging tasks that are beyond the current state-of-the-art in robotics. Although 143 teams applied to participate in the 2017 contest, including the best robotic labs worldwide, only four groups were able to grasp at least one object autonomously during the competition. To maximize the impact of future MBZIRC events and to encourage more competition, which will again push the limits of robotic systems, it is necessary for more teams to succeed in solving the challenging scenarios. A logical starting point is to use, or at least be inspired by, the approach that demonstrated the best performance in the 2017 MBZIRC, which is presented in this paper. Moreover, we would like to share and highlight the parts of the system and the phases in its development that brought added value in comparison with the systems of our competitors. Our experience and our solutions to the proposed challenges should be beneficial in further MBZIRC contests, in other robotic competitions, and also for the design of autonomous UAV systems for deployment in emergency applications. The rules of the competition forced teams to design a system for immediate deployment (the preparation time was only 20 min for the multi-UAV challenge) and for operation within a given time, without the option of postponing the start of the mission. This contrasts with

most robotic experiments presented in the literature, where only successful trials and demos are presented. Short preparation time and a successful start on demand, without the possibility of repeated trials, are required by industry and in emergency applications, and the MBZIRC competition was designed to force teams to achieve these requirements.

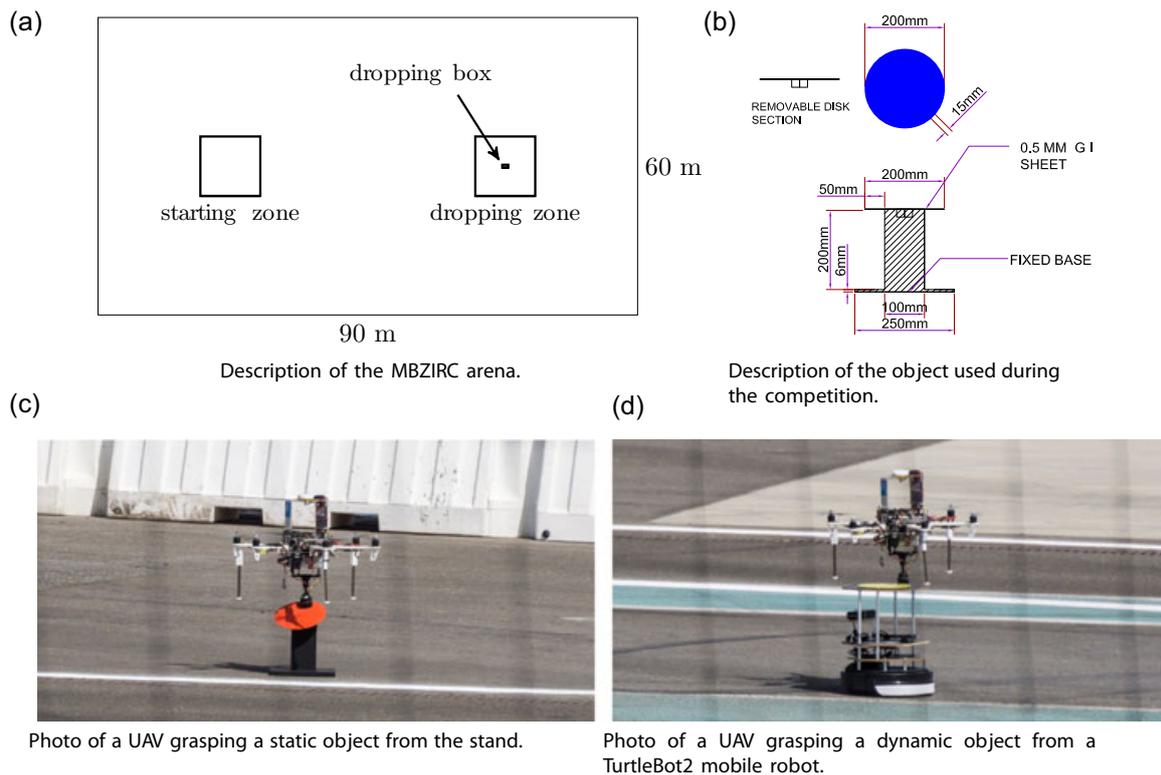
### 1.3 | Problem statement

In the MBZIRC 2017 Treasure Hunt challenge, three UAVs (with a maximum size of 120 cm × 120 cm × 50 cm) must locate, grasp, deliver, and drop a set of objects into a given box within 20 min. The set should contain 10 moving and 10 stationary small objects, as well as three stationary large objects, all of which are randomly placed inside the arena. The small objects were approximately 0.370 kg ferrous disks on a stationary stand or moving TurtleBot2 robot, as shown in Figure 1b–d. Different colors of the objects—green, blue, and red for the static objects, and yellow for the dynamic objects—were associated with different scores, one, two, three, and five points, respectively. The nonstationary objects were moving at random velocities not exceeding 5 km/hr. Three large orange objects not exceeding 200 cm in length, and not exceeding 2 kg in weight, were valued at ten points each on successful transport and delivery by at least two cooperating UAVs into the dropping zone depicted in Figure 1a. If a large object was moved into the dropping zone by a single robot, the team obtained five points. The small objects could be grasped by a single UAV and dropped into a box placed inside the dropping zone. The objects could be picked up by a magnetic gripper, a suction gripper, or another device carried onboard the UAVs. Before the start of each trial, the three UAVs had to be in the start location.

## 2 | HARDWARE

The specifications of the MBZIRC challenge described above influence the decision on which UAV platform to use. Our intention was to reuse the platforms and the entire system in our follow-up research, and to achieve simple replicability of the system in the future. Therefore, we tried to maximize the use of commercially available off-the-shelf UAV components, and only a few 3D printed specialized tools (sensor holders and the gripper). This approach reduced development time, increased reliability, and now enables our system to be used by other universities with a minimum overhead for technology transfer. It also increases the impact of this paper, which can be considered as a comprehensive manual for building a robust multi-UAV system, even for research groups without any experience with UAVs.

The proposed UAV platform is a complex system composed of integrated active members, computational resources, and sensor modules, shown in a schematic view of the system in Figure 2. The main structure of each UAV consists of a DJI hexacopter F550 frame and E310 DJI motors. This choice satisfies the size limitations of the MBZIRC event, the flight time, and the payload capability that is necessary for additional sensors, and also for carrying the objects. The

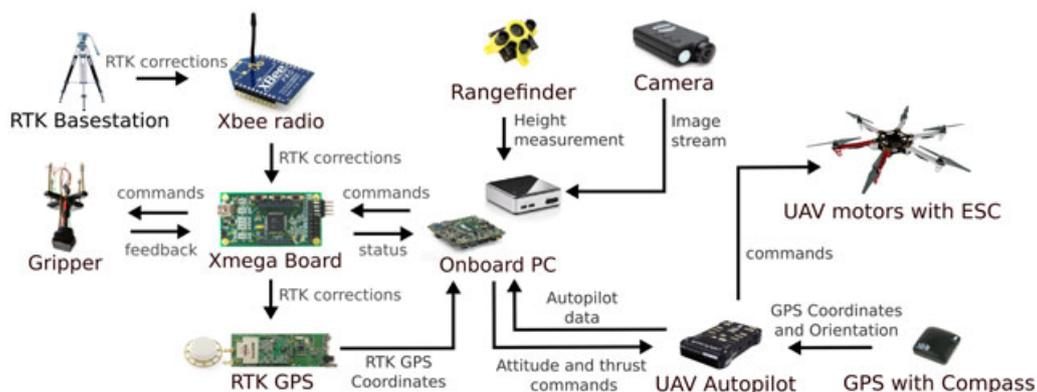


**FIGURE 1** Description of the MBZIRC 2017 competition. For more information, visit <http://www.mbzirc.com>. UAV: unmanned aerial vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

system is controlled at the lowest level by a PixHawk flight controller (Meier et al., 2012) that contains a set of sensors, such as accelerometers, gyroscopes, and magnetometers, which are necessary for stable UAV flight. The open-hardware and open-software architecture is advantageous for the MBZIRC competition, and also for research on multirobot systems. An Intel NUC-i7 PC provides sufficient computation power to solve all the required onboard image processing tasks, and also UAV coordination, state estimation, and motion planning in the complex Treasure Hunt challenge. Transport of messages between the onboard PC and PixHawk autopilot is performed over a serial line using MAVlink protocol. Communication between the UAVs, which is important for their

coordination, is provided by the Wi-Fi module embedded in the PC. A high-resolution Mobius ActionCam (2018) camera is used for object detection, and for relative visual localization.

The rules of the competition allowed the use of GNSS and even more precise navigation systems for localization. To maximize the accuracy and to increase reliability, our system uses a combination of the real-time kinematic (RTK) satellite, which enhances the precision of position data derived from satellite-based positioning systems (e.g. GPS, GLONASS, Galileo, and BeiDou), and a classical GNSS module attached to the PixHawk controller. Information on the position is provided in the RTK system by a PRECIS-BX305



**FIGURE 2** Description of components in our UAV platform. ESC: electronic speed controller; GPS: global positioning system; RTK: real-time kinematic; UAV: unmanned aerial vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

GNSS RTK BOARD (GPS L1L2/GLONASS G1/BEIDOU B1B3) (Tersus-GNSS, 2017), with accuracy of  $10 \text{ mm} \pm 1$  parts per million (ppm) horizontally and  $15 \text{ mm} \pm 1$  ppm vertically when the RTK device is in the most accurate state, RTK FIX. This RTK system requires a stationary GNSS receiver, called RTK basestation, which is placed on a known location. The RTK basestation then broadcasts its position and measurements from all visible satellites (RTK corrections) to the UAVs using XBee Pro radio modules (Digi International, 2017). A custom board was designed to provide communication of the XBee module with the RTK device.

In principle, the vertical position (altitude) provided by the RTK GPS is measured above the mean sea level. However, the UAV does not have any information about the ground-level profile or the distance to the objects that are to be grasped, based on the GPS. This information is obtained using the onboard *TeraRanger One* laser rangefinder, which is mounted face-down and is connected directly to the onboard PC, where its data are filtered and used for precise height control. Finally, the objects are grasped using an OpenGrab EPM v3 electropermanent magnet, which combines the advantages of electro and permanent magnets and creates a very strong magnetic contact with ferrous objects (NicaDrone, 2017). Our custom board (previously mentioned for managing communication from the XBee module into the RTK device) also provides a low-level interface between the main computer and the gripper.

### 3 | SOFTWARE SYSTEM STRUCTURE

The proposed solution relies on the robot operating system (ROS), which is an open-source set of software libraries and tools commonly used in the robotic community. Using ROS, the complex MBZIRC tasks can easily be divided into smaller subtasks (nodes). This also improves and clarifies the structure of the proposed solution. Furthermore, the Gazebo robotic simulator can be used for simulation in the loop, together with firmware from PixHawk, which provides a very realistic testbed and significantly simplifies testing of the whole system. Using this realistic simulator, hardware experiments could be carried out in a shorter time and in a safer way than if direct HW is used. Because changes were double-checked in the

simulator, we did not experience any serious crash during more than 1 year of intensive preparation for the MBZIRC event.

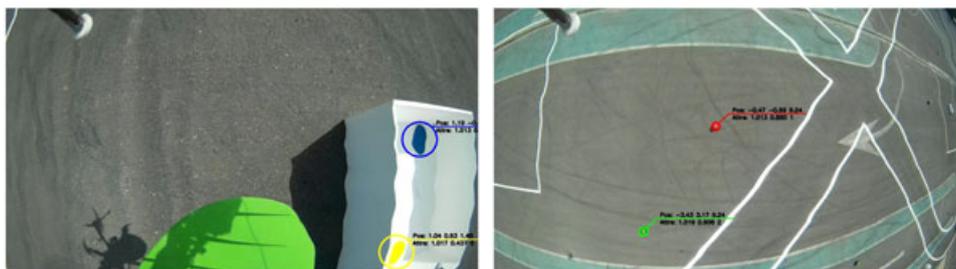
In this section, the subcomponents of the proposed system are described. The first two parts explain object detection, object estimation, and motion prediction for dynamic objects. In the next subsection, the estimation of the UAV state from all available sensors is introduced, followed by details on communication in the multirobot network. Further, the nonlinear controller used for UAV control is explained, together with the novel MPC-based approach used for online design of a feasible and smooth reference for the nonlinear controller. This is followed by details of high-level planning built upon MPC-based trajectory tracking, which is used for UAV coordination and collision avoidance when the same workspace is shared. Lastly, the FSM, which is crucial for managing all subsystems and for coordinating all UAVs sharing the same workspace, is described. All these subcomponents are executed on the onboard PC Intel NUC-i7.

#### 3.1 | Object detection

Since the camera that is used to detect the colored objects has a rolling shutter, vibrations induced by the drone motors cause the acquired images to be subject to a specific ‘jelly’ or ‘wobble’ effect, which makes the use of geometry-based methods for object detection (e.g., the Hough transform) problematic (Afolabi, Man, Liang, Guan, & Krilavičius, 2015; see Figure 3). We therefore designed a computationally efficient ellipse detection algorithm, which relies on the use of statistics that are robust to this type of noise (Krajník et al., 2014). However, the original method described in Krajník et al. (2014), which used adaptive thresholding to detect black-and-white patterns, had to be extended to process the color information.

Since the perceived colors are influenced by the light conditions, and the exact colors of the objects were not known until the actual contest, we created a semiautomatic autocalibration method that can learn a Gaussian-mixture-based model (GMM) of each color during a short hover over the objects. Once the GMMs are learned, they are used to create an RGB color map, which allows the image pixels to be classified rapidly into object candidates and the background.

The color map is then used in the method (Krajník et al., 2014), which searches for continuous segments of object-colored pixels,



**FIGURE 3** Object detection in onboard camera images affected by the ‘jelly’ or ‘wobble’ effect, which deforms lines (left image), as well as circular and square objects (right image). The detection results indicate the 3D relative position (top line) and attributes like roundness, eccentricity and type (1,2,3 for red, green, and blue static objects and 5 for the yellow moving object) [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

establishes their bounding box, the number of pixels, the centroid, convexity, and compactness and uses these statistics to reject segments that cannot correspond to circular objects. Then, using the known object size and camera parameters, the method calculates the relative 3D position of the object. This position is then transformed to a global 3D coordinate frame, and objects that do not appear to be close to the ground plane are rejected as false positives. Finally, global 3D positions of the detected objects are forwarded to a mapping module, which integrates multiple detections of the objects into a single 3D representation, which is then used by the planning system.

The performance of the method during tests and in the contest itself indicated computational efficiency and robustness to changing illumination, which was one of the key factors in the robustness of the entire system used in the MBZIRC competition.

### 3.2 | Object estimation and motion prediction

Localization of targets with onboard cameras tends to provide data that are inherently embedded with flaws. The data may be skewed by phenomena such as *signal noise*, *false positive detections*, *irregular detection rate*, *data blackouts*, etc. These issues can hardly be mitigated during the detection, and some of them (e.g., data blackouts) also depend on the external environment. Moreover, several moving targets appear in the MBZIRC challenge and so estimates of unobserved states such as velocities and heading may help to follow their position precisely. This leads to a need to filter the detected position of the targets. We also required the filtration system to be capable of sorting out measurements belonging to targets that have been marked as *unreliable*, for example, due to data blackout being too frequent. Another requirement comes from the multirobot nature of the task. A UAV should share information about parts of the map that are currently occupied. Targets in those areas should then be filtered out in other UAVs to prevent unrequired grasping of the same target by multiple UAVs.

In the event that there is a single target in the field of view (FOV) of the UAV, an Unscented Kalman Filter (UKF) is used as a filter and as a predictor in conjunction with the car-like motion model

$$\begin{aligned} \mathbf{x}_{[n+1]}^o &= \mathbf{x}_{[n]}^o + \dot{\mathbf{x}}_{[n]}^o \Delta t, \\ \dot{\mathbf{x}}_{[n+1]}^o &= \begin{pmatrix} \cos \phi_{[n+1]} \\ \sin \phi_{[n+1]} \end{pmatrix} \mathbf{v}_{[n+1]}, \\ \phi_{[n+1]} &= \phi_{[n]} + \dot{\phi}_{[n]} \Delta t, \\ \dot{\phi}_{[n+1]} &= K_{[n]} \mathbf{v}_{[n]}, \\ \mathbf{v}_{[n+1]} &= \mathbf{v}_{[n]} + \mathbf{a}_{[n]} \Delta t, \\ K_{[n+1]} &= K_{[n]} + \dot{K}_{[n]} \Delta t, \end{aligned} \quad (1)$$

where  $\mathbf{x}_{[n]}^o = (x, y)^T$  is the position of the object in the global coordinate system,  $\phi_{[n]}$  is its heading,  $K_{[n]}$  is the curvature of its turn,  $\mathbf{v}_{[n]}$  is its scalar velocity,  $\mathbf{a}_{[n]}$  is its scalar acceleration, and  $\Delta t$  is the time difference. An estimate of the target heading allows its motion to be tracked, while the onboard camera is oriented with its wider FOV in favor of detecting sudden changes of the object's heading.

However, real-world scenarios might contain several objects in the FOV, while some of them are moving. In that case, the UAV needs to track a particular object independently of the movement of all the objects in the scene. This requires a local map of the objects to be actively maintained. Our map model was based on Equation (1) for an arbitrary number of independent objects. Another state has been included to cover the type of the object (its color and whether it is moving) as well as the time of its last update and whether it is currently active. Manipulation of the objects in the map obeys the following principles:

- An object that has not been seen for more than 5 s is *deactivated*. Deactivated objects stay in the map, but their movement is no longer predicted by the UKF.
- Objects that are deactivated for more than 3 s are deleted from the map.
- Measurements from the *object detector* (Section 3.1) are paired with objects in the map using min-distance bipartite graph matching, constrained by the color of the objects.
- Objects located outside of the competition arena or in any of the locally banned areas (near the dropping zone or around other UAVs) are deleted from the map, and new measurements in these areas are discarded.

Additionally, it can be anticipated that grasping attempts may not be successful at all times. The filter allows a temporarily ban on an area around a particular object, to avoid deadlock in the grasping state machine. Such a ban is valid for 30 s in a radius of 4 m around the object.

### 3.3 | UAV position estimation

Automatic control of UAVs relies on estimates of the states of the UAV dynamical system. Namely, knowledge of position and velocity (both vertical and horizontal) is required to coordinate the movement for precise picking up and delivery of the object. Our platform is equipped with several independent sources of information, which are fused to obtain a single, reliable and smooth estimate of the UAV pose. An important requirement is to ensure smoothness of the resulting signal, since the  $SO(3)$  state feedback is sensitive to noise.

The main source of data for both the vertical and the horizontal axes is the PixHawk flight controller. Its extended Kalman filter fuses present-day inertial sensors—a three-axis accelerometer and a gyroscope with an altitude pressure sensor and a GPS receiver. Although the aircraft is already capable of autonomous flight with this off-the-shelf setup, we make use of other sensors to provide more precise localization and thus better precision of object manipulation.

#### 3.3.1 | Horizontal position estimation

The position estimates in the lateral axes are based on the estimate provided by PixHawk, namely positions  $\mathbf{x}^p$ , and velocities  $\dot{\mathbf{x}}^p$ . Although the precision of the estimates may be satisfactory locally for short time intervals, they are prone to significant drift in time spans of minutes.

To correct this drift, and thus to ensure repeatability of the experiments and, for example, locating the dropping zone, the horizontal position from PixHawk is corrected by differential RTK GPS. Position measurements from the RTK GPS receiver are fused using the linear Kalman filter with the model

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} \Delta t \\ \Delta t \end{pmatrix}, \quad (2)$$

where  $\mathbf{x}_{[n+1]}^e = \mathbf{A}\mathbf{x}_{[n]}^e + \mathbf{B}\mathbf{u}_{[n]}$  is the linear system equation,  $\mathbf{x}_{[n]}^e = (x, y)^T_{[n]}$  is the state vector finally used for control, and  $\mathbf{u}_{[n]}$  is the system input. According to our experience,

$$\mathbf{x}_{[0]}^p + \sum_{n=0}^k \dot{\mathbf{x}}_{[n]}^p \Delta t_{[n]} = \mathbf{x}_{[k]}^p, \quad \forall k \in \mathbb{N} \quad (3)$$

does not hold for the position and velocity estimate provided by PixHawk. This is a very useful observation for somebody building a fully autonomous UAV system using an off-the-shelf controller. The input vector  $\mathbf{u}$  consists of velocities obtained as differentiated positions  $\mathbf{x}^p$  (later integrated by the filter), which ensures that the proposed filter does not introduce any drift into the resulting estimate when no RTK GPS corrections are received. In situations when the position is not being corrected, the resulting estimate follows the same relative state trajectory as  $\mathbf{x}^p$ , just shifted according to the latest correction. The final tuning of the filter resulted in process covariance  $\mathbf{Q} = \text{diag}(1, 1)$  and measurement covariance  $\mathbf{R} = \text{diag}(10e3, 10e3)$ . Moreover, the RTK GPS corrections were saturated to ever impose maximally 0.25 m difference from the internal state of the filter. Such technique limits sudden changes of the estimated position, which was necessary for safety of the flight.

The multirobotic scenario requires a coordinate space to be shared among all three UAVs. The base of our Cartesian system is set to predefined GPS coordinates and its orientation is according to the East-North-Up convention. Therefore, the first, second, and third axis point to the east, north and upwards, respectively. A point of origin is measured using the RTK GPS, to which all independent coordinate systems of all UAVs are then shifted after each of them is powered up. The common base station of the differential RTK GPS then ensures that all UAV estimates are corrected to lie within the same global coordinates.

### 3.3.2 | Vertical position estimation

In contrast with the horizontal position, estimates of the height rely much less on PixHawk. The linear Kalman filter for the vertical axis also uses the differentiated PixHawk height in the same manner as the horizontal axis. However, height corrections come not only from the differential RTK GPS, but also from the down-facing TeraRanger rangefinder and the object detector, which is able to provide an estimate of the relative distance, when flying above an object. The estimator provides an option to switch between these sources of data, depending on the current task and the circumstances.

It is feasible to correct the height using the TeraRanger rangefinder, when flying above uneven ground, but it cannot be

used reliably when the down-facing sensor is obstructed, for example, when carrying an object, or when there might be a foreign object on the ground, namely the dropping box. RTK GPS can provide precise relative height measurements, but only when RTK FIX has been established. This depends on the strength of the GNSS signal and on the quality of the communication link between the base station and the UAV. Finally, correcting the altitude using data from the *object detector* may bring in unexpected steps in the signal due to false-positive detections. Since none of the additional sources is completely reliable, we implemented a safety mechanism for detecting anomalies, which can toggle off any of the above-mentioned sensors from being fused.

## 3.4 | Communication between UAVs

In multirobot systems, reliable communication is required mainly if there is a need for direct cooperation between multiple autonomous vehicles, as in the case when large objects are to be carried cooperatively. However, a reliable communication channel is a crucial tool even for coordinating the UAV team sharing the same workspace for grasping small objects individually, as was demonstrated in the MBZIRC competition. The rules of the MBZIRC event specified that all teams are obliged to share the same 5 GHz Wi-Fi network, the reliability of which was influenced by interference occurring during transmission. This may easily lead to packet loss, which can interrupt the connection. Decreased reliability of the communication link during the entire mission is not limited to the MBZIRC case. It is a typical feature of most UAV applications in demanding outdoor conditions. The MBZIRC contest therefore provided an interesting and realistic evaluation scenario for multi-UAV systems, in which it cannot be assumed that a complete communication network is available at all times. In our opinion, our system achieved significantly better performance in the multi-UAV scenario than the other teams, due to the following strategy. We attempted to maximize utilization of the communication channel, if it was available, to achieve optimal behavior of the system. However, it was important to be able to degrade into a system not relying on the communication infrastructure at all. This was done at the cost of decreased performance, but our system still provided safe flight operation of multiple UAVs solving the given task. A smooth and possibly repeated transition between the optimal behavior relying on communication and the nonoptimal but safe and still working system without communication, and back, is provided by the FSM approach described in Section 3.8.

The software part responsible for managing communication between UAVs is based on the ROS master within the ROS network. To increase the robustness of the communication net in the event of a failure of the robot that is the leader in the ROS master scheme, the proposed method relies on multiple independent ROS masters assigned to each of the UAVs. The ROS package `multimaster_fkie` (Tiderko, 2017) is used to maintain communication between these ROS masters. This package offers a set of nodes to establish and

manage a multimaster network, which is necessary for such tasks with the team of UAVs in the event of an unreliable communication infrastructure.

To reduce the load of the communication channels managed by the ROS master network, only selected information (topics) are exchanged between the team members:

- the actual position of the UAV in the global coordination system,
- the actual state of the high-level state machine being part of the FSM,
- the estimated position of the object during grasping,
- the planned trajectory.

These topics are used in nodes for proactive collision-free planning, fail-safe reactive collision avoidance, and object estimation. The bandwidth of the Wi-Fi network necessary for transmission of all mentioned information for a single UAV is approximately 10 kB/s.

### 3.5 | Low-level UAV control

The position controller uses the estimated state as feedback to follow the trajectories given as an output of the high-level trajectory planner. In many previous works, a backstepping approach is used for UAV control, because the attitude dynamics can be assumed to be faster than the dynamics governing the position, so linearized controllers are used for both loops (Herissé, Hamel, Mahony, & Russotto, 2012; Mellinger et al., 2013; Weiss, Scaramuzza, & Siegwart, 2011). However, we need the system to be capable of large deviations from the hover configuration during operations like fast mapping of objects, or for strong wind compensation. We therefore use a nonlinear controller. Let us consider an inertial reference frame denoted by  $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$  and a body reference frame centered in the center of mass of the vehicle with an orientation denoted by  $\mathbf{R} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ , where  $\mathbf{R} \in \text{SO}(3)$ . The dynamic model of the vehicle can be expressed as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + m\mathbf{g}\mathbf{e}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, \\ \mathbf{J}\dot{\hat{\boldsymbol{\Omega}}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} &= \mathbf{M}, \end{aligned} \quad (4)$$

where  $\mathbf{x} \in \mathbb{R}^3$  is the Cartesian position of the vehicle expressed in the inertial frame,  $\mathbf{v} \in \mathbb{R}^3$  is the velocity of the vehicle in the inertial frame,  $m \in \mathbb{R}$  is the mass,  $f \in \mathbb{R}$  is the net thrust,  $\boldsymbol{\Omega} \in \mathbb{R}^3$  is the angular velocity in the body-fixed frame, and  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  is the inertia matrix with respect to the body frame. The hat symbol  $\hat{\cdot}$  denotes the skew-symmetry operator according to  $\hat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$  for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ ,  $\mathbf{g}$  is the standard gravitational acceleration, and  $\mathbf{e}_3 = [0 \ 0 \ 1]^T$ . The total moment  $\mathbf{M} \in \mathbb{R}^3$ , with  $\mathbf{M} = [M_1 \ M_2 \ M_3]^T$ , along all axes of the body-fixed frame and the thrust  $\tau \in \mathbb{R}$  are control inputs of the plant. The dynamics of the rotors and propellers are neglected, and it is assumed that the force of each propeller is directly controlled. The total thrust,  $f = \sum_{j=1}^6 f_j$ , acts in the direction of the z-axis of the body-fixed frame,

which is orthogonal to the plane defined by the centers of the six propellers. The relationship between a single motor thrust  $f_j$ , the net thrust  $f$ , and the moments  $\mathbf{M}$  can be written as

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ sd & 1 & sd & -sd & -1 & -sd \\ -cd & 0 & cd & cd & 0 & -cd \\ -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \quad (5)$$

where  $c = \cos(30^\circ)$ ,  $s = \sin(30^\circ)$ , and  $d$  is the distance from the center of mass to the center of each rotor in the  $\mathbf{b}_1, \mathbf{b}_2$  plane. For nonzero values of  $d$ , Equation (5) can be inverted using the right pseudoinverse.

For control, we build on the work in Lee, Leok and McClamroch (2013) and in Mellinger and Kumar (2011) with control inputs  $f \in \mathbb{R}$  and  $\mathbf{M} \in \mathbb{R}^3$  chosen as

$$\mathbf{M} = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} - \mathbf{J}(\hat{\boldsymbol{\Omega}}\mathbf{R}^T \mathbf{R}_C \boldsymbol{\Omega}_C - \mathbf{R}^T \mathbf{R}_C \dot{\boldsymbol{\Omega}}_C), \quad (6)$$

$$\begin{aligned} f &= -\left(-k_x \mathbf{e}_x - k_{ib} \mathbf{R} \int_0^t \mathbf{R}(\tau)^T \mathbf{e}_x d\tau - k_{iw} \int_0^t \mathbf{e}_x d\tau - k_v \mathbf{e}_v - m\mathbf{g}\mathbf{e}_3 \right. \\ &\quad \left. + m\ddot{\mathbf{x}}_d\right) \cdot \mathbf{R}\mathbf{e}_3 = \mathbf{f} \cdot \mathbf{R}\mathbf{e}_3, \end{aligned} \quad (7)$$

with  $\ddot{\mathbf{x}}_d$  the desired acceleration, and  $k_{iw}, k_{ib}, k_x, k_v, k_R, k_\Omega$  positive definite terms. We extend the referenced controllers by including two integral terms which accumulate the error in the body frame and in the world frame, respectively. We include both terms to provide the opportunity to capture external disturbances (e.g., wind) separately from internal disturbances (e.g., an inefficient prop or a payload imbalance), particularly when the vehicle is permitted to yaw or rotate about the vertical axis. The thrust and the moments are then converted to motor rates according to the characteristic of the proposed vehicle. Subscript C denotes a commanded value, and  $\mathbf{R}_C = [\mathbf{b}_{1,C}, \mathbf{b}_{2,C}, \mathbf{b}_{3,C}]$  is calculated as

$$\begin{aligned} \mathbf{b}_{2,des} &= [-\sin \psi_{des}, \cos \psi_{des}, 0]^T, \quad \mathbf{b}_{3,C} = \frac{\mathbf{f}}{\|\mathbf{f}\|}, \\ \mathbf{b}_{1,C} &= \frac{\mathbf{b}_{2,des} \times \mathbf{b}_3}{\|\mathbf{b}_{2,des} \times \mathbf{b}_3\|}, \quad \mathbf{b}_{2,C} = \mathbf{b}_3 \times \mathbf{b}_1. \end{aligned} \quad (8)$$

Note that here we have to define  $\mathbf{b}_{2,des}$  based on the yaw, instead of defining  $\mathbf{b}_{1,des}$  as it was defined in Mellinger and Kumar, 2011, due to a different Euler angle convention (we use the ZYX convention instead of ZXY). The quantities

$$\begin{aligned} \mathbf{e}_R &= \frac{1}{2}(\mathbf{R}_C^T \mathbf{R} - \mathbf{R}^T \mathbf{R}_C)^V, \quad \mathbf{e}_\Omega = \boldsymbol{\Omega} - \mathbf{R}^T \mathbf{R}_C \boldsymbol{\Omega}_C, \\ \mathbf{e}_x &= \mathbf{x} - \mathbf{x}_d, \quad \mathbf{e}_v = \dot{\mathbf{x}} - \dot{\mathbf{x}}_d, \end{aligned} \quad (9)$$

represent the orientation, the angular rate errors, and the translation errors, respectively. The symbol  $\cdot^V$  represents the *vee* map  $\text{so}(3) \rightarrow \mathbb{R}^3$ . If the initial attitude error is less than  $90^\circ$ , the zero equilibrium of the tracking error is exponentially stable, that is,  $[\mathbf{e}_x^T \ \mathbf{e}_v^T \ \mathbf{e}_\Omega^T \ \mathbf{e}_R^T]^T \equiv [\mathbf{0}^T \ \mathbf{0}^T \ \mathbf{0}^T \ \mathbf{0}^T]^T$ .

### 3.6 | Trajectory tracking

The state feedback, described in Section 3.5, which provides precise position and velocity control, requires a smooth and feasible reference. The reference consists of all states of the translational dynamics—position, velocity, and acceleration—and is provided at 100 Hz, the same rate as the resulting control signal. There are various ways to create the reference. Typically, thanks to the differential flatness of the UAV dynamical system, a QP optimization can be solved to find a polynomial given the initial and final state conditions (Mellinger & Kumar, 2011), which can then be sampled to create the reference. In our case, we chose to generate the reference using a MPC approach. MPC ensures that the resulting trajectory satisfies a given model as well as the dynamical constraints, which are imposed on the model. As it optimizes control actions over a prediction horizon, it can react adequately to unfeasible changes in the reference trajectory, and can also create proper feed-forward proactions to minimize the control error in the future.

The MPC tracker uses a QP formulation of a minimal sum-of-squares problem, where the optimal control action  $\mathbf{u}$  is found for a future prediction horizon of states  $\mathbf{x}_{[n]} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z})_{[n]}^T$  by minimizing the function

$$\begin{aligned} V(\mathbf{x}_{[0, \dots, m-1]}, \mathbf{u}_{[0, \dots, m-1]}) &= \frac{1}{2} \sum_{i=1}^{m-1} (\mathbf{e}_{[i]}^T \mathbf{Q} \mathbf{e}_{[i]} + \mathbf{u}_{[i]}^T \mathbf{P} \mathbf{u}_{[i]}), \\ \text{s.t. } \mathbf{x}_{[0, \dots, m-1]} &\geq \mathbf{x}_L, \\ \mathbf{x}_{[0, \dots, m-1]} &\leq \mathbf{x}_U, \end{aligned} \quad (10)$$

where  $\mathbf{e}_{[n]} = \mathbf{x}_{[n]} - \bar{\mathbf{x}}_{[n]}$  is the control error,  $\bar{\mathbf{x}}_{[n]}$  is the setpoint for the MPC,  $m$  is the length of the prediction horizon, and  $\mathbf{x}_L$  and  $\mathbf{x}_U$  represent box constraints on states. The control error  $\mathbf{e}_{[n]}$  requires the formation of a general prediction of  $\mathbf{x}_{[n]}$ , which was described previously in Baca, Loianno, and Saska (2016). In our case, the optimized control action is not directly used to control the real UAV. Instead, it controls a model of the UAV translational dynamics in real-time simulation. States of the simulated model are then sampled at 100 Hz to create the reference for the state feedback. This is a novel approach in UAV control, where benefits of both nonlinear control and linear MPC are used together.

An important notion is the difference between the trajectory setpoint  $\bar{\mathbf{x}}$  and the reference, which is generated by the MPC tracker. The trajectory setpoint  $\bar{\mathbf{x}}$  is provided by an operator or a program. No requirements are imposed on  $\bar{\mathbf{x}}$ . In contrast, the reference produced by the MPC Tracker is feasible, satisfies the UAV dynamics and state constraints, and serves as a control reference for the  $SO(3)$  state feedback.

The simulated model is a linear time-invariant system covering the third-order translational dynamics of the UAV with sampling of  $\Delta t = 0.01$  s. In our MPC formulation,  $\Delta t$  is different for the first iteration ( $\Delta t = 0.01$  s) and for all the other iterations ( $\Delta t = 0.2$  s). This allows smooth control of the simulation, if the MPC is executed at 100 Hz, while there is a relatively sparse distribution of further states, which allows us to have a much longer prediction horizon than there would normally be with  $\Delta t$  being constant. As in traditional

MPC, only the control action in the step is used to control the model in the simulation. In the meantime, a new instance of the optimization task is formulated, starting from new initial conditions, which results in a fresh control action for the next step. This method is valid only if the MPC can be solved repeatedly within 0.01 s.

The penalization parameters  $\mathbf{Q}$  and  $\mathbf{P}$  in Equation (10) were found empirically. As in our previous work (Baca et al., 2016), we used the *move blocking* technique to effectively prolong the prediction horizon while maintaining the computational complexity. The particular control action distribution for the MBZIRC competition was as

$$\mathbf{U} = (1 \ 1 \ 1 \ 1 \ 1 \ 5 \ 5 \ 5 \ 5 \ 5 \ 10), \quad (11)$$

which results in an 8-s prediction horizon with only 33 variables in the optimization task.

Creating the control reference for the state feedback with MPC has several advantages over conventional solutions. It produces a reference that is feasible according to the specified model, which makes it safe to execute. If the setpoint for MPC is not feasible, the resulting reference is feasible with respect to Equation (10). The inherent predictive nature of MPC provides trajectory tracking optimizing actions over the future, which makes it ideal for tracking moving targets, such as the moving objects in the competition.

As defined in Equation (10), MPC handles state constraints as linear constraints. We impose maximum acceleration and velocity box constraints on the UAV to ensure safe and feasible resulting trajectories. The optimization being solved lies in the family of linearly constrained quadratic programming, which acquires a global optimum in a convex polytope. A custom solver, based on a sequential closed-form solution, has been implemented to ensure guaranteed real-time performance.

MPC-based trajectory tracking operates in two modes, as follows. The first simple positioning mode, used mainly for short distance position changes, applies either relative or absolute position commands, and tries to reach a given position in the fastest way with respect to the MPC scheme. The second trajectory-following mode used by high-level trajectory planning (Section 3.7) uses a precomputed path plan, and tries to precisely track the trajectory while respecting the plan waypoints schedule, which is crucial for multirobot collision-free operation.

Having the predictions of the future movement for all UAVs allows us to extend the capability of the MPC tracker to avoid future collisions. When communication between the aircraft is established, they exchange their future trajectory predictions and act according to a decentralized mechanics, which will alter their courses to avoid the collision, based on sorting the UAVs by priorities. If there is a potential collision between two UAVs, the UAV with lower priority will avoid the other UAV by changing to a higher flight level. The system also allows priorities to be reassigned dynamically in the following cases:

- UAV should be avoided at all times (its priority is higher by definition). This may occur when it is currently grasping an object, or when its avoidance mechanism is accidentally turned off.

- UAV should avoid the other aircraft even if it has higher priority. Such a situation occurs when the other machine does not comply with the mechanics for any reason.

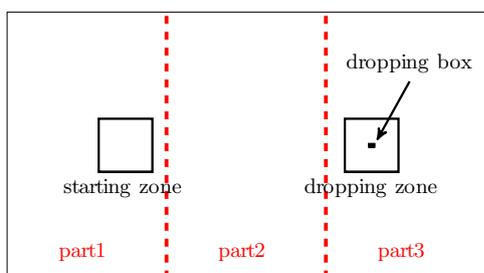
### 3.7 | High-level trajectory planning

High-level trajectory planning is built on top of MPC-based *Trajectory tracking*, which is used for precise tracking of the planned trajectories. The onboard online trajectory planning mechanism is used in two main parts of the Treasure Hunt scenario. The first task is Sweeping of the arena, where the team of UAVs is required to localize the objects within the arena, and either save their locations to the global map (at the beginning of the mission) or immediately try to grasp the first detected object (later in the mission, once all objects detected in the initial map have been processed, the grasping was successful, or failed repeatedly). The second online trajectory planning is utilized in Proactive collision-free planning, which is involved in cases where one UAV has to fly into another position. For example, when it holds the grasped object and wants to drop it into the dropping box.

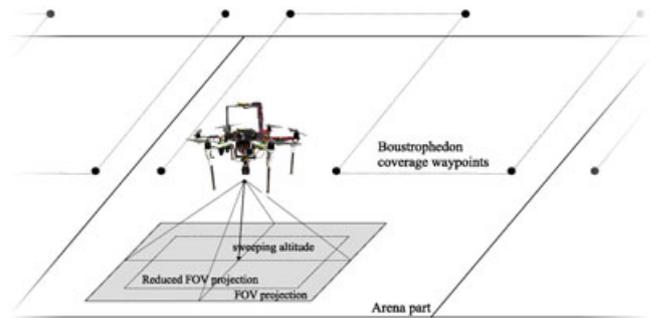
#### 3.7.1 | Sweeping

Sweeping the arena designed for the MBZIRC Treasure Hunt challenge involves localizing both dynamic and static objects. The trajectory planning for so-called sweeping can be described as coverage path planning (CPP; Galceran & Carreras, 2013), where for a given area the CPP should find a path from which the entire workplace can be scanned with an onboard sensor, in our case an onboard camera.

The proposed multirobot CPP algorithm is based on simple area decomposition into three equally large zones that split the area along the larger side (Figure 4). Each arena zone has one UAV assigned to localize and pick up the objects from. All UAVs then plan the coverage path using Boustrophedon coverage (Choset & Pignon, 1998) in each part of the area separately. Using Boustrophedon coverage, we create zigzag paths, as shown in Figure 5, such that the reduced FOV entirely covers the particular arena zone. The reduced FOV is set based on the required overlap in the coverage (set to 20% during the competition) and on the real FOV camera projection to the ground plane with respect to the sweeping altitude that is used.



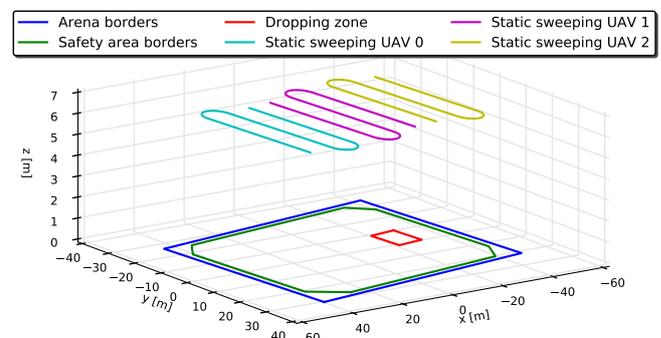
**FIGURE 4** Decomposition of the Mohamed Bin Zayed International Robotics Challenge arena into three equal zones [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 5** Boustrophedon coverage of the decomposed competition arena. FOV: field of view [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

To produce smooth trajectories for constant speed object detection, the Dubins vehicle model (Dubins, 1957) is used to create the final path between the waypoints. The minimal turning radius  $\rho = v_c^2 / a_{\max}$  of the Dubins vehicle was selected based on the desired constant velocity  $v_c$  ( $\sim 3 \text{ ms}^{-1}$ ) and the maximal acceleration of the UAV  $a_{\max}$  ( $\sim 2 \text{ ms}^{-2}$ ), using an equation of circular motion with constant speed. The sweeping high-level trajectory planning is summarized in Figure 6, where the shown trajectories for all three UAVs were further used in the two following approaches in different stages of the Treasure Hunt scenario.

In the first approach, called static sweeping, the UAVs follow the created trajectories at a height ( $\sim 7 \text{ m}$ ) and simultaneously detect the colored objects while the global map of the static objects is being created. After this initial coverage, the approximate positions of the detected static objects are estimated based on multiple detections of the same object. The second approach, called dynamic sweeping, is applied later in the schedule of the task, and the UAVs use similar paths as in the static sweeping. However, the sampled trajectories are used repeatedly (not just once, as in the static sweeping) and the UAVs do not create a global map. Instead, each UAV tries to find and estimate the position of any object while following the sweeping trajectory. When any object is located, the trajectory following is stopped and the UAV tries to grasp the object immediately. Either after successful grasping and dropping of the object, or after a number of unsuccessful



**FIGURE 6** Sweeping trajectories based on Boustrophedon coverage using the Dubins vehicle and decomposition of the arena into three distinct parts, one for each UAV. UAV: unmanned aerial vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

grasps, the UAV continues with dynamic sweeping from the last trajectory sample.

### 3.7.2 | Proactive collision-free planning

Our strategy for covering the Treasure Hunt competition arena is based on decomposition into three equally large zones for each of the UAVs (Figure 4). Unfortunately, the dropping zone is located in one-third of the competition arena. After successful grasping, the UAV in part 1 therefore has to fly through the remaining zones to drop the object. Because there is a possibility of colliding with another UAV during this flight through the remaining zones, proactive collision-free planning has to be used. The actual positions of the UAVs are known due to information sharing, as was explained in Section 3.4. However, the Wi-Fi communication infrastructure is not reliable and, as mentioned, a multirobot system deployed in real world conditions should be robust to losing Wi-Fi communication. Therefore, we decided to use different flying heights for each of the UAVs, which minimizes the possibility of a collision, without any additional planning. Unfortunately, while completing this task the UAVs cannot maintain only these heights during the mission, as they have to descend for events such as grasping the objects and then dropping them. These events take most of the overall flight time, because they require a complicated grasping manoeuvre and hovering in front of the dropping zone, if it is sharing with other UAVs. Moreover, the grasping manoeuvre can be repeated several times before the object is gripped.

The proposed solution for finding a collision-free trajectory uses four assumptions derived from the MBZIRC rules, which are, however, valid for most cooperative transport applications:

- A Collision can occur only if a UAV leaves its dedicated height.
- The position of the UAV in the x-axis and in the y-axis does not alter rapidly in the event that it flies out of its safe altitude (the grasping and dropping manoeuvres are carried out following strictly vertical trajectories that accept grasping of dynamic objects, but where the lateral movement is also minor).
- The shape of the competition workspace is convex.
- At most three UAVs are used in the environment (this assumption is valid only for the MBZIRC Treasure Hunt task, but an extension of the approach is straightforward for different numbers of robots).

Thanks to these assumptions, the method for very rapidly computing a collision-free trajectory can be simplified to finding a collision-free

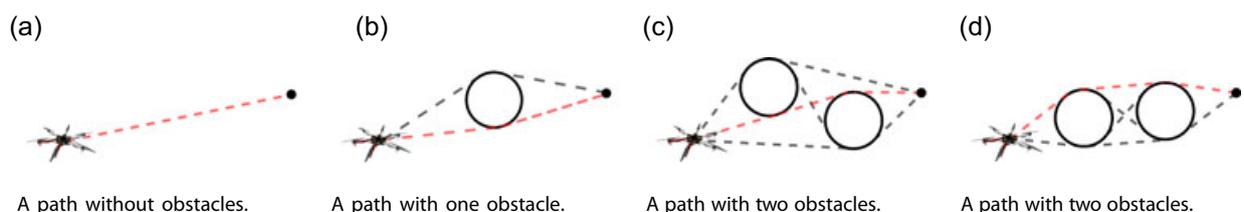
path in 2D (at the dedicated height) between two points, where only two obstacles can occur. These obstacles are circles centered on the x and y coordinates of neighboring UAVs with safety radius  $r_a$ . It is prohibited to encroach on these circles. The safety radius of the circles depends on the speed of the UAVs which, for security reasons in the MBZIRC competition, was restricted to a maximum of 30 km/hr. We used a detection radius (the relative distance between UAVs in which the avoidance maneuver is initiated) of 5 m radius during the competition, while the critical radius in which the UAVs are considered to be in a collision is 0.8 m.

Based on the previously realized experimental comparison of available path planning approaches (Saska, Kulich, & Preucil, 2006), a visibility graph method (Lozano-Pérez & Wesley, 1979) was applied to solve the collision-free planning problem. The method provides the shortest path and it is sufficiently fast in simple situations including limited number of obstacles. Only four possible paths in the graph consisting of tangent lines to circles, which represent the obstacle, and the circle segments can be considered as a candidate solution in our case of two obstacles. The solution can, therefore, be found analytically in a very short time (possibly in each control step) with negligible burden on the processor. See examples of trajectories generated by proactive collision-free planning in Figure 7a–d.

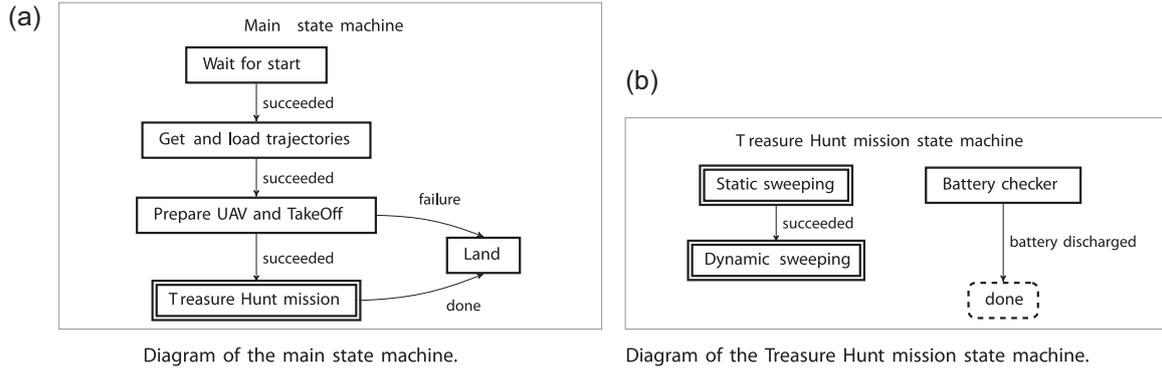
A collision-free trajectory exists only for described planning when the start points or the end points are not inside the safety radius  $r_a$  of another UAV. In situations when a UAV is already inside the safety radius  $r_a$  of another UAV, the UAV finds a plan into the nearest position that is not in conflict with a UAV, and the collision-free planning procedure is initiated. If the high-level planning system requires to fly into a position, which is occupied by another UAV, then a temporary goal position is set instead. This position is the closest feasible position to the original goal such that it lies on the original trajectory. The UAV then waits for up to 1.5 min until the goal position is available again. If the goal position is not freed within this time, it is assumed that the information about the occupation of the goal position is incorrect. During the MBZIRC competition, the planning was repeated five times per second, and in the event of a communication interruption, the last received states of other UAVs were considered as correct for 5 s.

### 3.8 | Failure recovery and synchronization jobs manager

The main core of the system is the FSM concept, which is used for managing all subsystems. It increases the robustness of the entire



**FIGURE 7** Examples of trajectories generated by fast proactive collision-free planning. The black circles denote obstacles. The red path shows the shortest collision-free trajectory, and the gray paths denote other collision-free paths [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]



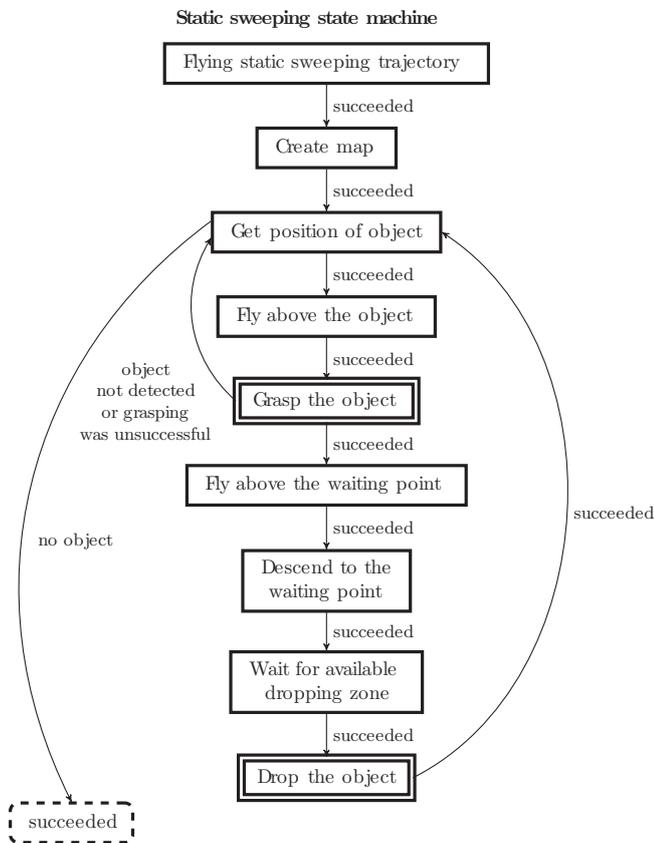
**FIGURE 8** The structure of the FSM tool. FSM: failure recovery and synchronization jobs manager; UAV: unmanned aerial vehicle

code structure resolving the remaining few subsystem failure cases due to wrong sequential and concurrent operations. In the proposed system, the FSM is designed using SMACH (Bohren, 2017), a ROS-independent Python library, and it is fully integrated into the designed ROS framework.

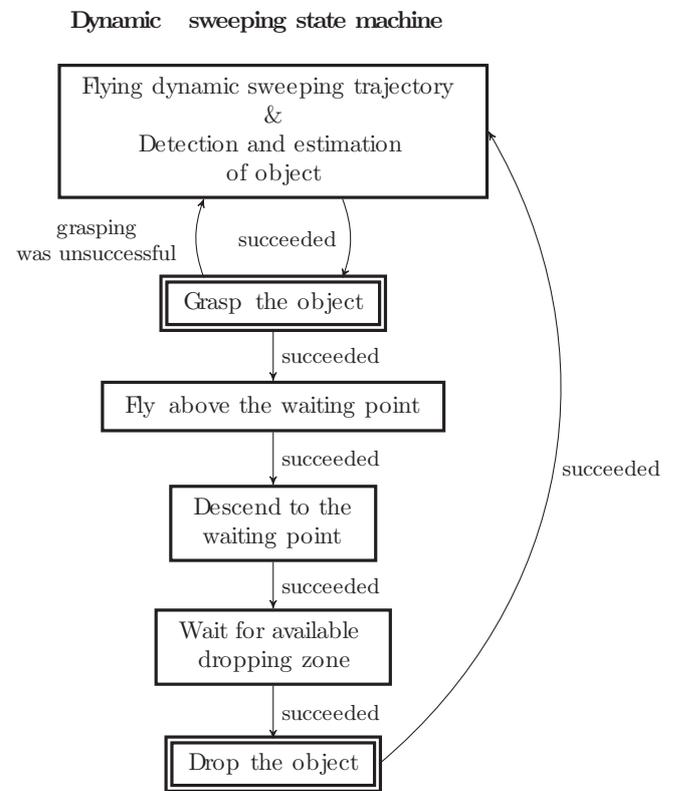
As was mentioned in the introduction, the entire FSM structure may be considered as a hierarchical state machine with synchronization and failure recovery abilities. For simplicity, we will refer to the components of the FSM as state machines in this section. In Figures 8, 9, 10, and 12, the internal states of the FSM levels (the so-called state machines) are visualized by rectangles, and the nested lower-level

state machines are visualized by double-line rectangles, such as the *Treasure Hunt mission* state machine introduced in Figure 8a by the *Treasure Hunt mission* rectangle, and described in detail in Figure 8b. Transitions between two states and from one state to a lower-level state machine are marked by the arrow with a label of an outcome describing the transition. Dotted terminal states represent the transition that is called after returning to a higher level state machine. The land event is called whenever any state produces an outcome that means that the UAV cannot continue in its mission.

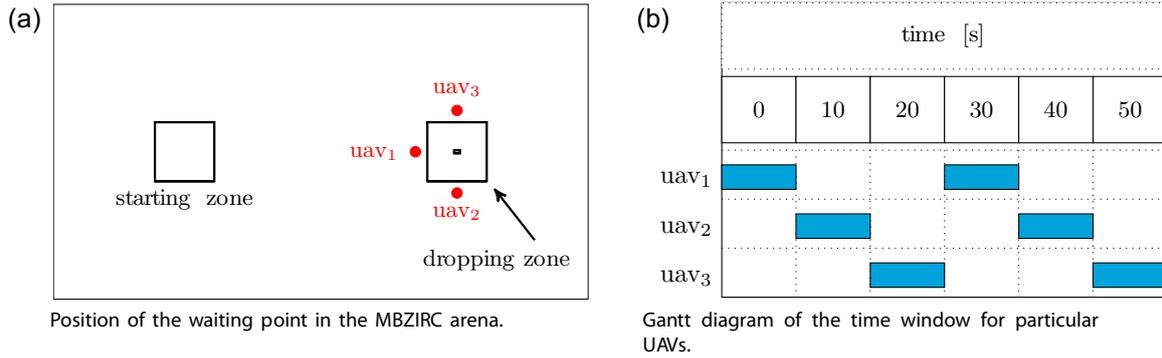
The diagram of the main state machine is visualized in Figure 8a. In the first step, the trajectories for static sweeping and also for dynamic sweeping in the predefined part of the competition arena (see Section 3.7.1) are loaded, and an automatic take off is called. Once the UAV is



**FIGURE 9** Diagram of the static sweeping state machine



**FIGURE 10** Diagram of the dynamic sweeping state machine

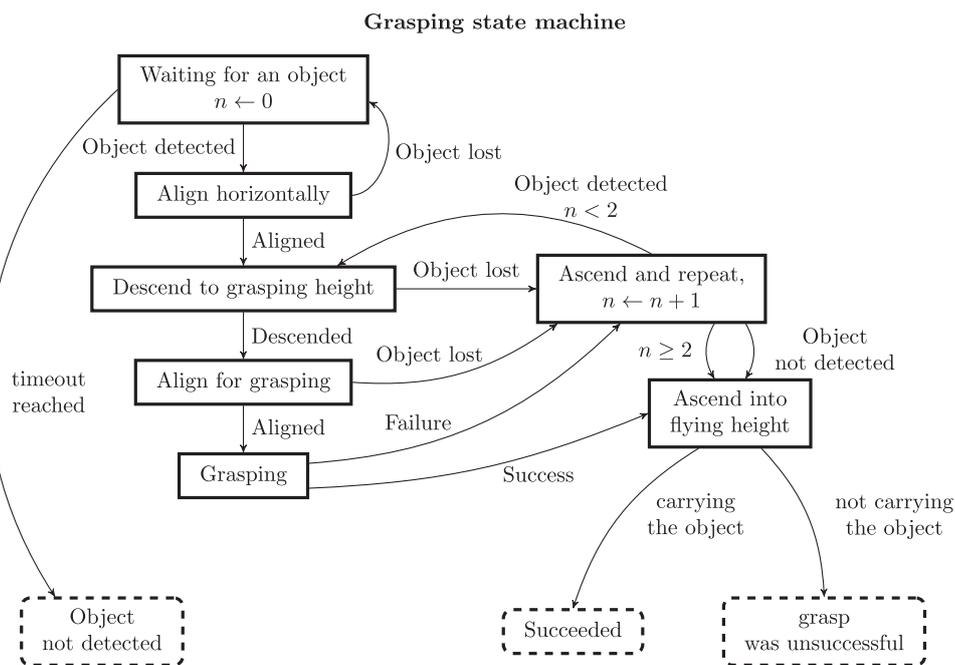


**FIGURE 11** Waiting position around the dropping zone, and a Gantt diagram of the proposed time window strategy. The index of the UAV indicates to which part of the arena the UAV belongs. (a) Position of the waiting point in the MBZIRC arena. (b) Gantt diagram of the time window for particular UAVs. MBZIRC: Mohamed Bin Zayed International Robotics Challenge; UAV: unmanned aerial vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

in the air, the mission state machine is activated (Figure 8b). The mission is a concurrent state machine that sequentially runs the static sweeping procedure and the dynamic sweeping procedure, while simultaneously controlling the voltage of the battery. If the battery is discharged, the state machine terminates all currently executed tasks of the UAV, and a land event is called. The level of voltage for battery discharge was set experimentally for each battery type.

The static sweeping state machine (Figure 9) starts by following the sweeping trajectory and creating a map with approximate positions of the static objects. After this initial coverage of the competition arena, an attempt is made to grasp the nearest estimated static object in the map. The grasping procedure is shown in Figure 12. Initially, the state machine starts with the object detection mechanism. Whenever an object is located, the UAV tries to align itself horizontally above the estimated position of the object and then to

descend to the grasping height of 1.5 m above the ground. Once the UAV has reached the desired height and it is aligned above the object, it tries to grasp the object. Whenever the object is lost in the steps after descending to the grasping height, the UAV ascends and repeats these steps again. The steps are also repeated if the grasping fails. Only two attempts are made to grasp the estimated object. If the UAV was not successful in these attempts, the state machine returns the UAV to the safe flying height and it is terminated with the outcome that the grasping was unsuccessful. After a successful grasp, the UAV also ascends to the safe flying height, but the grasping state machine outputs that the grasp was successful. The decision as to whether the UAV is carrying an object is made via a feedback from the Hall effect sensors that are placed on the gripper. To avoid deadlock, the state machine is terminated in the first node if the object is not found within a certain time.



**FIGURE 12** Diagram of the grasping state machine

The static sweeping state machine reacts to unsuccessful outcomes from the grasping procedure by selecting a new object for grasping from the map. When the grasping attempt was successful and the UAV is carrying the object, the UAV flies at its safe height to a position above the waiting point. The waiting point is selected based on the part of the competition arena assigned to the UAV. During the MBZIRC competition, the safe flying heights for part 1, part 2, and part 3 were 3, 4, and 5 m, respectively. The waiting points were located 7 m (measured in  $x, y$  plain) from the center of the dropping box (Figure 11a). Once the UAV reaches the position above the waiting point at its safe height, it descends to the dropping height of 1.5 m above the ground. At this waiting point, the UAV hovers until the moment when the dropping zone is not occupied by any other UAV, if the communication infrastructure is available, or until the UAV has access to the dropping zone based on the time windows, if the communication channel cannot be used for negotiation and for sharing the status of the dropping zone.

The negotiation about access to the dropping zone is based on queries of the current UAV on its waiting position addressed to neighboring UAVs. The neighboring UAV responds with confirmation that allows the current UAV to access the zone, but only when the neighboring UAV is not inside this zone, or if the neighboring UAV has not been waiting for access for longer than the current UAV. The current UAV starts with the dropping maneuver only when it receives confirmations from all neighboring UAVs. The negotiation about access to the dropping zone is carried out repeatedly until the UAV receives confirmations.

If communication has been lost for more than a predefined time during the mission, all UAVs will switch to a strategy with time windows for accessing the zone to avoid collisions in the dropping zone. Time windows 10 s in length are used for each UAV. This range of time windows provides two time intervals for dropping for each UAV per minute. The UAV in part 1 can be in the restricted area around the dropping zone in the 0–9 s time interval, the UAV in part 2 can be there in the 10–19 s time interval, and the UAV in the part 3 can be there in the 20–29 s time interval. This strategy is the same for accessing the dropping area in the second half of the minute, so the intervals are offset by 30 s (Figure 11b). The UAV can call the dropping procedure only when it is in the waiting position at the dropping height, and its time window starts. This strategy is not as effective as negotiation and sharing of the status of the dropping zone, but it is safer in the case of a problematic communication network. This strategy requires the clocks on the UAVs to be initially synchronized within a few milliseconds using *chrony*—an implementation of the network time protocol.

The dropping maneuver is done in sequence: Flying above the dropping box at the dropping height, dropping the object, and returning to the UAV safe height above the waiting position. After dropping the object, the state machine initializes the grasping procedure with the next estimated object in the map. This is done until all detected objects have been grasped, or an attempt has been made to grasp them, in the case of a grasping failure.

In the dynamic sweeping state machine (Figure 10), the UAV flies the dynamic sweeping trajectory, and when any object is detected and its position is estimated, the UAV immediately tries to grasp it. After successful grasping and dropping, the UAV flies back into the dynamic sweeping trajectory and continues with dynamic sweeping while simultaneously looking for the remaining objects. This approach is not as effective as the initial static sweeping procedure, where the UAVs could fly for another object in the map directly, and minimize the overall flight time, but it is more robust. In the ideal case of perfect mapping and grasping procedures, all static objects are grasped during the static sweeping part, and only the dynamic objects are hunted during the dynamic sweeping. In the demanding real-world conditions of the MBZIRC arena, with changing light conditions and wind gusts, many objects were not grasped in the first phase of the mission. This was due to a safety procedure that allowed a limited number of grasping attempts per object to avoid a deadlock. These missed objects could be grasped later, in the dynamic sweeping part, as the local environment conditions changed.

Another interesting property of this approach is the possibility to exchange the sweeping trajectories, and therefore the operational zones and waiting positions between the UAVs after a given period of the mission. This increases the robustness and the performance of the overall system in the event of a failure or a malfunction of a UAV subsystem. Even if all components of all UAVs are fully functional as designed, each UAV in the team behaves differently in different tasks, and it often happened that a UAV could accomplish a task in which another team member failed, and vice versa. This is another useful lesson learned during the MBZIRC event that should be adapted for designing multirobot systems, if possible. Finally, splitting the static object grasping in the initial sweeping part and the subsequent grasping of dynamic objects and the remaining static objects increases the overall system robustness. There is a much lower probability of a UAV crash during static object grasping. This has been shown in numerous realistic complex simulations, and also during system testing and its deployment in the competition.

## 4 | EXPERIMENTAL RESULTS

In this section, we present both the experimental results achieved while preparing for the Treasure Hunt scenario, and also the performance of the system during the MBZIRC competition. The remainder of this section is divided into main parts, where we present the experimental results achieved in the simulator, during the preparations for the competition in South Bohemia, in the final tests in a challenging desert environment and in the course of the MBZIRC competition. A video attachment to this paper is available at website <http://mrs.felk.cvut.cz/jfr2018treasurehunt>.

### 4.1 | Robotic simulator

The system was initially developed using the Gazebo robotic simulator, which was used as the simulation in the loop, together

**TABLE 1** Percentiles of duration of the experiment before the first collision occurred

Percentile	0.5	0.75	0.95	0.99
Without the avoidance	104 s	152 s	264 s	431 s
With the avoidance	–	–	–	–

Note. The results were obtained in two 24-hr simulated flights (one with and one without the collision avoidance mechanism used) with five UAVs, conducting a 2D random walk on  $100 \times 100$  m area. The total of 495 collisions were recorded if the collision avoidance mechanism was not used.

with the PixHawk firmware. Using the robotic simulator, the process of developing the subsystems and integrating the entire system was carried out significantly faster and more safely than when using the real system directly. In addition, by modeling the whole scenario in the simulator and by testing the behavior of the complex FSM approach in it, the complete system achieved the necessary level of reliability for deployment in tasks such as the Treasure Hunt.

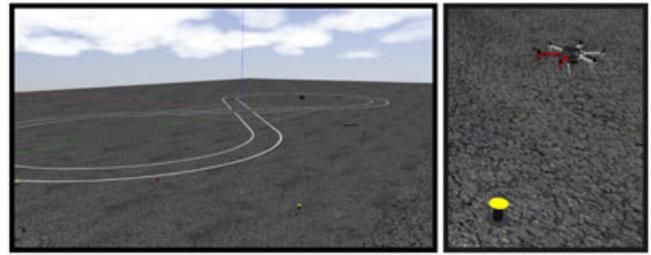
The underlying layers of the control pipeline, namely, the UAV state estimation, control, tracking, and predictive collision avoidance, were extensively tested using the Gazebo simulator. To show the system robustness, we conducted 24-hr simulated flights of five UAVs in an area of  $100 \times 100$  m. Each UAV followed an independent random walk reference in the same height. Without the collision avoidance technique, the median time of the first collision between any of the UAVs was 104 s, from total of 495 simulated scenarios (simulation was always restarted after the first collision). With the collision avoidance mechanism, there was not a single collision within the 24 hr of the experiment, while the minimal registered distance between the UAVs was 1.21 m, which is still 50% more than the collision distance 0.8 m of the used platforms. See Table 1 for the comparison of percentiles of duration of the experiment before the first collision occurred.

The results from 20 simulations of the complete MBZIRC 2017 Treasure Hunt scenario are shown in Table 2. Each of these simulations contained 10 static and 10 dynamic objects, which were randomly placed in a simulated MBZIRC arena. Snapshots from simulation are shown in Figure 13. We expected that the dynamic objects will move according to some motion model that is predictable

**TABLE 2** Results from 20 simulations of Challenge 3, in which the objects (10 static and 10 moving) were randomly placed

	Mission time (min)	Time needed for grasping of the static object (s)	Time needed for grasping of the dynamic object (s)	Smallest distance between UAVs (m)
Min	12.1	23.7	35.0	1.9
Max	17.4	36.4	51.2	3.3
Mean	13.6	30.6	43.6	2.5

Note. UAVs in a distance closer than 0.8 m are colliding in the simulation as well as in the real system, which never happened in simulations and real flights if the collision avoidance approach was used. UAV: unmanned aerial vehicle.

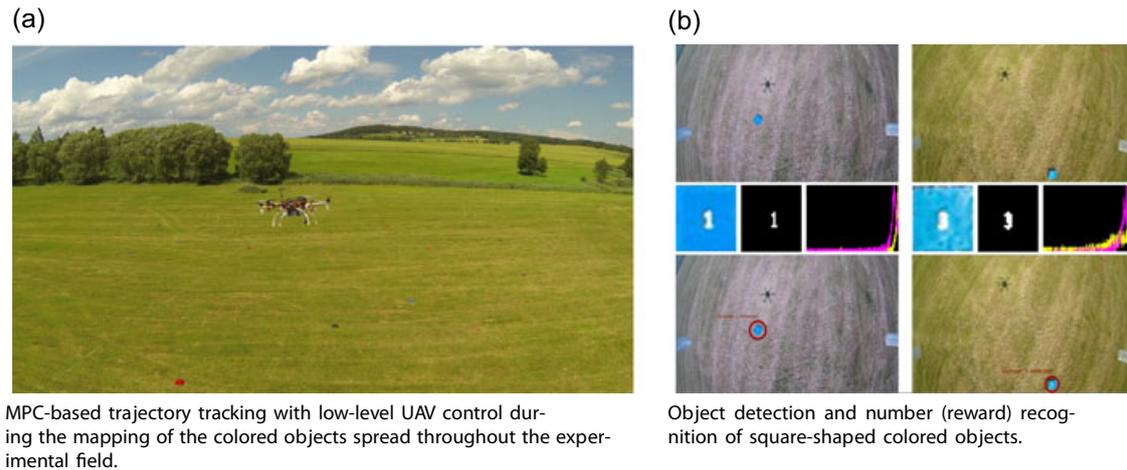
**FIGURE 13** Snapshots from the simulation developed for the Mohamed Bin Zayed International Robotics Challenge competition [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

and smooth. Therefore, we modeled the movement of the dynamic objects in simulation using the car-like motion model, where the velocity of the object did not exceed 0.3 m/s. Due to the movement type of dynamic objects being uncertain, the mission was divided into two parts. The first part is the safe part of the mission, where only the static objects are attempted to be grasped and delivered. After this part is finished, the rest of objects will be targeted regardless of whether they are static or dynamic. Results from the simulations in Table 2 show that the system is capable of collecting all targets to the dropping box in the competition time interval of 20 min. The best time of finishing the mission was 12.1 min and the worst was 17.4 min. The average time needed was 13.6 min. Results in Table 2 also show that all static objects were grasped faster than the fastest dynamic object. Furthermore, thanks to using collision avoidance methods, there was no collision between members of the team during the mission. The closest any UAVs got to each other was 1.9 m, which only happened in one of the simulations, and in general the mutual distances were higher than that.

## 4.2 | Experimental camps in the Czech Republic

Key parts of the proposed system were tested in the course of experimental camps held in the countryside of South Bohemia in the Czech Republic throughout the year before the competition (Figure 14). Repeated experimental verification of key parts of the proposed system was necessary to test phenomena that are difficult to simulate, and also to discover issues that were not present in our previous hardware experiments without physical interaction of the robot with the real-world environment. One issue that was discovered was the influence of the force produced by the propellers on the carried objects. This exposed the need for a stronger magnetic gripper, which we then designed. Another discovered issue was the ground effect caused by the objects. This manifested itself as turbulence in the last phase of the grasping maneuver.

The most crucial parts of the system were the low-level UAV control and the MPC-based trajectory tracking, used for precise positioning of the UAV. These were thoroughly tested to obtain the centimeter precision required for the grasping task. The MPC-based trajectory tracking used during the colored object mapping is shown in Figure 14a. In addition, initial testing of the object detection was carried out. However, in accordance with the initial specification of the



**FIGURE 14** Experimental verification of the MPC-based trajectory tracking method and the object detection algorithm during the experimental camps in the countryside of South Bohemia in the Czech Republic. <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video1>. MPC: model predictive control; UAV: unmanned aerial vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

shape of the object expected in the competition, we designed square shaped objects with numbers describing the rewards (Figure 14b). Videos showing initial attempts for grasping and dropping of the object, and trajectory following are available at <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video1>.

One of the experimentally verified subsystems was the MPC-based collision avoidance implemented for reactive avoidance of collisions between multiple UAVs in the MBZIRC competition. Using the MPC predictions of the future parts of the trajectory of other UAVs (discussed in Section 3.6), each UAV can avoid collisions with other UAVs by a simple change of flight height in potential collision parts of the future trajectory. It is necessary to employ this method in scenarios with a problematic communication network. This is because after reestablishing communication the proactive collision-free planning may not be able to deal with a suddenly discovered imminent collision, or may not even be active in the current phase. This safety mechanism is implemented on the lowest level of control in all phases of the mission. Figure 15 shows the verification of MPC-based collision avoidance, with two UAVs exchanging their position and one hovering UAV between the two positions. A video showing this verification is available at <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video2>. Such collision avoidance requires only a small number of messages to be shared between UAVs. These messages contain the MPC future trajectory predictions of each UAV, and are distributed with a very low frequency of 2 Hz. Although the proposed collision avoidance technique requires only a low communication bandwidth ( $\sim 6$  kB/s for three UAVs), the collision avoidance was not always used during the competition, due to dropouts of communication between UAVs, which was observed by all teams in the competition.

Another evaluated subsystem was the object detection and mapping. In particular, the datasets gathered were used to compare computational efficiency of our object detection method to the Maximally Stable Extremal Regions (MSER) (Matas, Chum, Urban, & Pajdla, 2004) and “SimpleBlobDetector” methods included in the

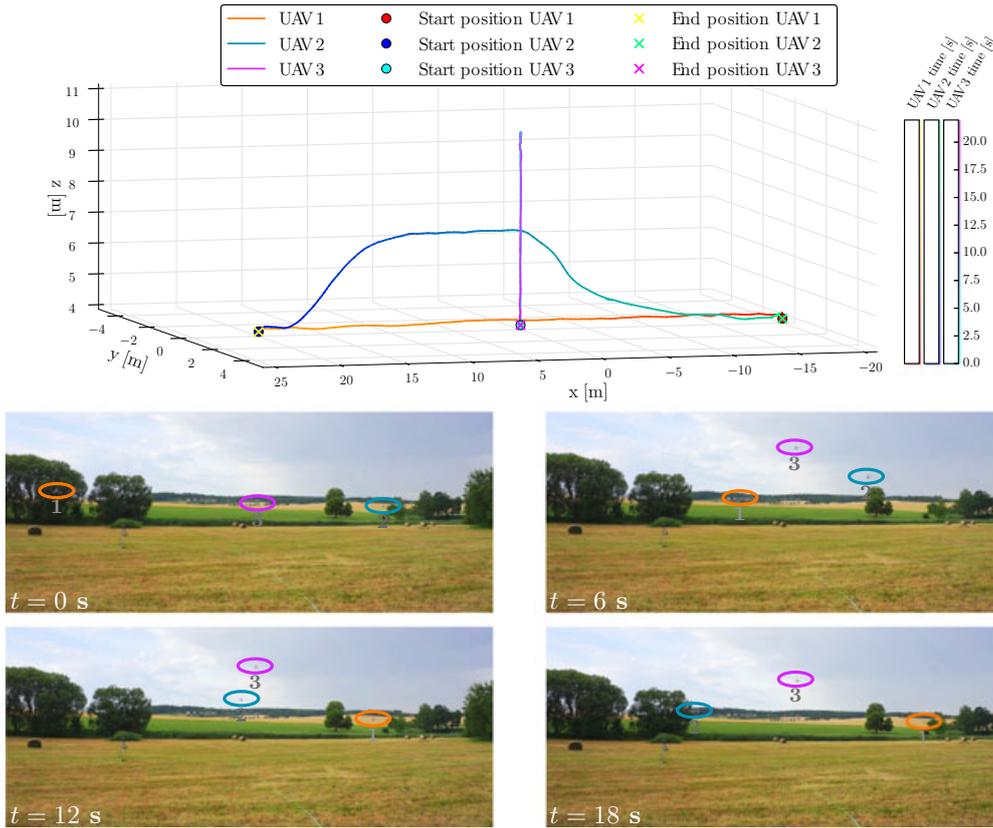
OpenCV library (Bradski, 2000). The results indicated that the system presented achieved significantly higher frame rates compared to the aforementioned two methods. This confirmed the experiments in Krajník et al. (2014), which introduced an algorithm our detection was based on.

### 4.3 | Desert testing in the United Arab Emirates

Finally, the complete system was thoroughly tested for a period of three weeks just before the competition, in the desert near Abu Dhabi in the United Arab Emirates. The desert environment was challenging, due to the uneven terrain and the rapidly changing wind conditions. By tuning the system for such weather and terrain conditions, our system was better prepared for the environment at the Yas Marina Circuit in Abu Dhabi, where the competition was held. The rapidly changing terrain profile in the dunes of the desert also had an influence on the quality of the communication network. The frequent interruptions of the connection inspired our solution, which does not rely on the communication network.

As we have mentioned, several important features of our system were, in our opinion, the dominant factors that led to our winning performance in all trials of the Treasure Hunt challenge in the MBZIRC competition. Most of the other teams did not take into consideration external disturbances such as wind in their controller. Surprisingly, the MBZIRC competition arena was not perfectly flat, and some teams had relied on its flatness. Finally, relying on a robust communication network was the main bottleneck of the competitive solutions.

Photos from the tests of the system in the desert are shown in Figure 16. The grasping procedure is captured in the image on the right, and the dropping maneuver is shown in the image on the left. A video showing the behavior of the complete system with three UAVs in this environment is available at <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video3>. During this testing, the yellow objects were stationary as opposed to the competition, where they were dynamic. This means, that in this phase, the system was tested for the static



**FIGURE 15** MPC-based collision avoidance between three drones. Two drones (UAV 1 and UAV 2) exchange their positions, while the third UAV 3 hovers in a position colliding with their trajectories. Using MPC future trajectory prediction, the UAVs avoid a collision by changing their trajectory height. <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video2>. MPC: model predictive control; UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

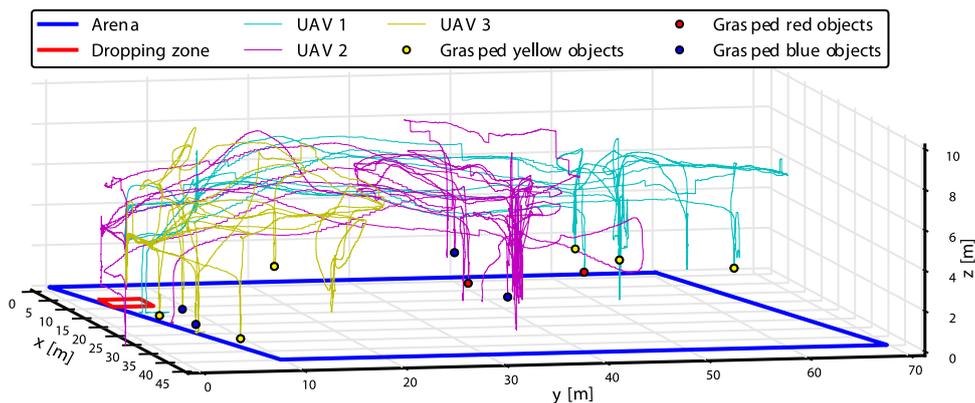
objects only. In addition, the paths traveled by the UAVs during the experiment presented in the video are shown in Figure 17. In this figure, the z-axis denotes the height above the level of the starting position as measured by the differential RTK GPS. The UAVs were kept at constant height above the ground and therefore the graph shows how uneven the terrain was. Furthermore, Figure 17 depicts the positions and colors of the objects that were collected.

#### 4.4 | Results from the MBZIRC competition

Our system was applied four times in the Treasure Hunt scenario during the final MBZIRC competition. During the competition, the number of dynamic (yellow) objects was decreased from announced 10 to 3 for this scenario for organizational reasons. The results, that is, the number of colored objects that were collected, are shown in Table 3. The first two



**FIGURE 16** Photos from the tests of the proposed system in the desert near Abu Dhabi, United Arab Emirates. The grasping procedure is captured in the image on the right, and the dropping maneuver is shown in the image on the left. <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video3> [Color figure can be viewed at wileyonlinelibrary.com]



**FIGURE 17** The paths traveled by individual UAVs during the desert experiment. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

**TABLE 3** Numbers of the objects collected in the Treasure Hunt scenario during the MBZIRC 2017 competition

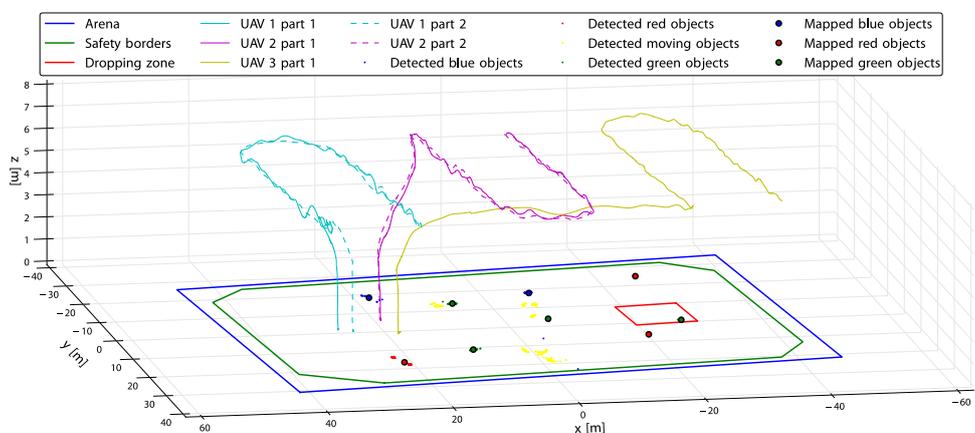
	Placed into the box	Placed outside the box but inside the dropping area
TRIAL 1	2R, 2G	1G
TRIAL 2	2R, 3G	
GRAND 1	1R, 1G, 2B, 1Y	1G, 1B
GRAND 2	2R, 3G, 1B, 1Y	1G

Note. B: blue static object; G: green static object; GRAND 1 and 2: trials of the MBZIRC Grand Challenge; R: red static object; TRIAL 1 and 2: trials of MBZIRC Challenge 3; Y: yellow nonstationary object.

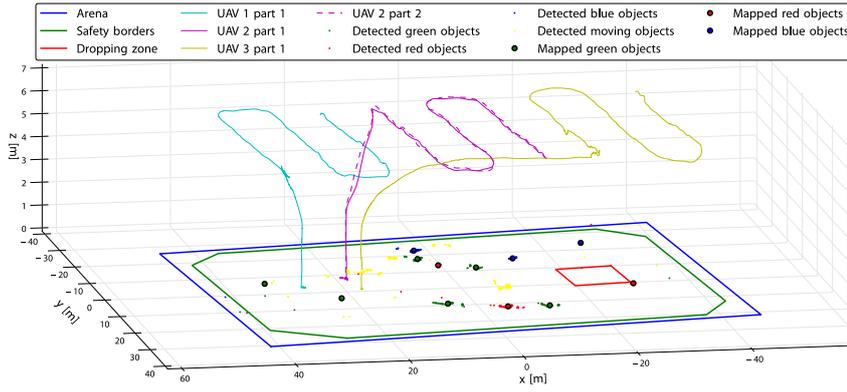
attempts, denoted as TRIAL 1 and TRIAL 2, are the results from Challenge 3, which contained only the Treasure Hunt scenario. The remaining two trials (GRAND 1 and GRAND 2) were a part of the Grand Challenge, where the Treasure Hunt scenario was undertaken simultaneously with the scenario of landing on a moving ground vehicle (Challenge 1), and the scenario where a ground robot had to locate and reach a panel, and further physically operate a valve located on the panel (Challenge 2). During these four trials within the competition, 25 objects overall were successfully placed into the dropping zone. The best

performance according to the number of grasped and placed objects was achieved during the second trial of the Grand Challenge, when eight objects, including a nonstationary object were brought into the dropping zone. The system described in this paper won first place in Challenge 3, and contributed to our third place in the Grand Challenge. A video showing results from the MBZIRC competition is available at <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video4>.

One part of the system for the Treasure Hunt scenario involved localizing objects using sweeping trajectories (described in Section 3.7.1). The static sweeping paths traveled by UAVs in the trials of Challenge 3 are shown in Figures 18 and 19. The flight time of the described UAV platform with fully charged four cell batteries with 6,750 mAh capacity is up to 15 min, which is less than allowed time per trial. The organizers allowed to change the batteries during the trial without any penalization. The trajectories before changing the batteries are labeled in the graphs as part 1, and after the batteries are changed, they are labeled as part 2. Furthermore, on these graphs, the colored points denote the detections of the objects that were observed, and the larger circles denote the estimated positions of these objects. After processing the data from the first trial, we decided to decrease the sweeping trajectory height from 7 to 6.5 m. This modification made objects more visible in camera images, which improved object detection. A disadvantage of this change



**FIGURE 18** Mapping sweep during the first trial of Challenge 3. The colored points denote the detections of the objects that were observed, and the larger circles denote the estimated positions of these objects. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]



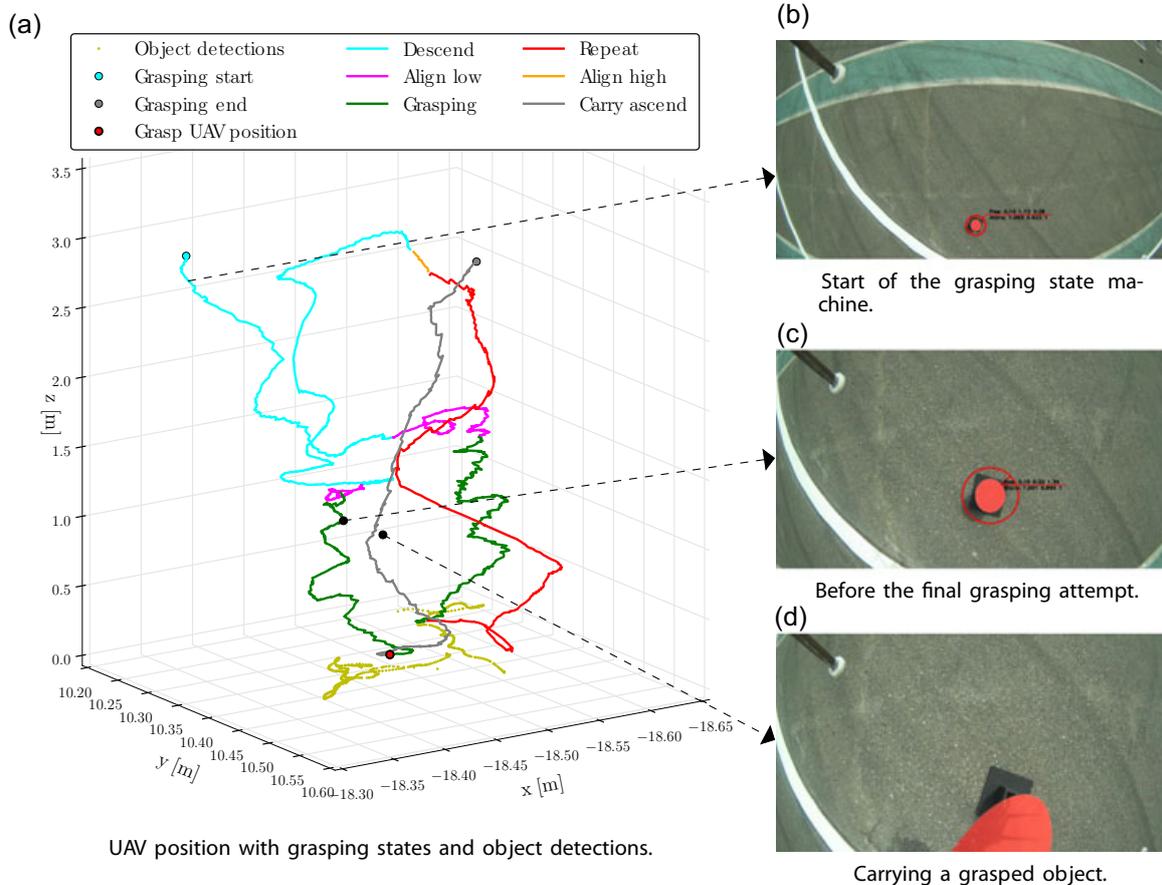
**FIGURE 19** Mapping sweep during the second trial of Challenge 3. The colored points denote the detections of the objects that were observed, and the larger circles denote the estimated positions of these objects. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

was that it prolonged the trajectories, because the condition of at least 20% of overlap in the coverage could not be satisfied by following the same trajectory (in the xy plane). For this reason, the sweeping trajectories differ between these two trials.

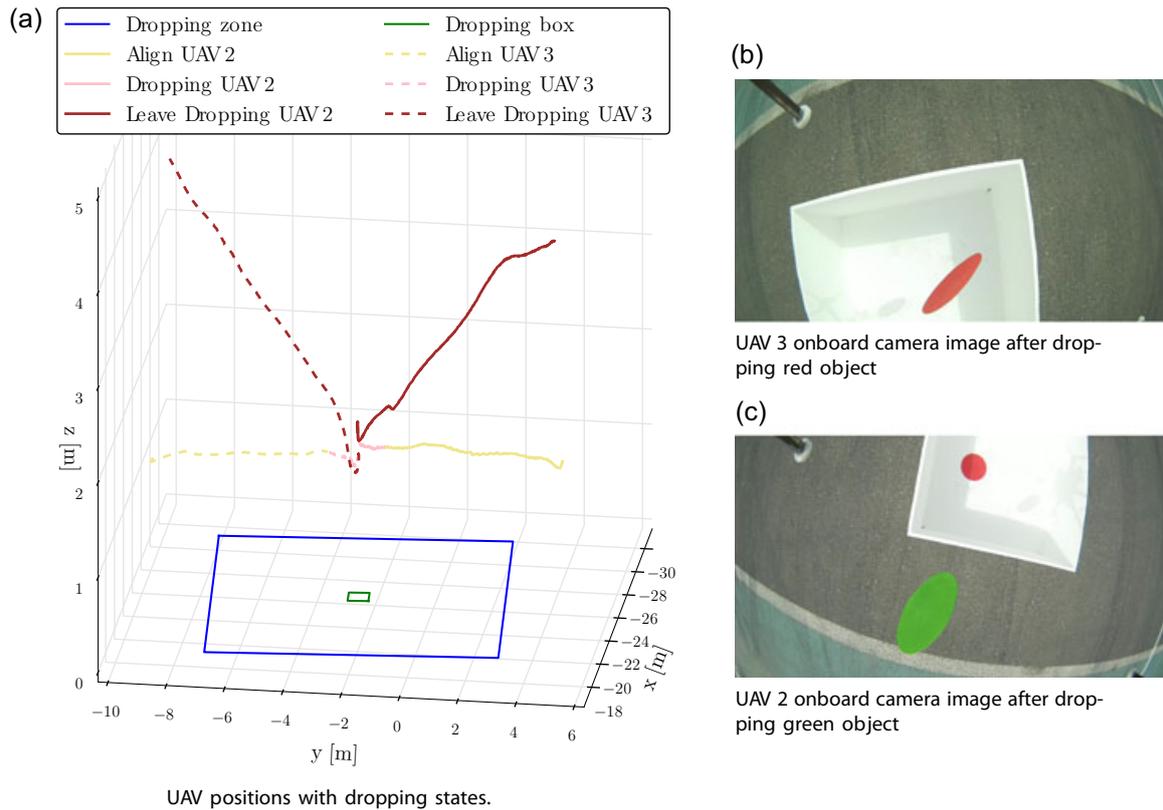
Another important part of described system is the grasping capability, where the UAV has to grasp a ferrous object. The overall grasping approach has been presented in Section 3.8, where the grasping state machine is depicted in Figure 12. Switching of the phases of the grasping state machine is shown in Figure 20, where an attempt at grasping was repeated after being aborted once. For a visualization of the transition

between these phases, the resolution of the graph in Fig. 20a is 0.05 m in the x-axis and in the y-axis. In addition, detections of the object in three parts, which are indicated by dotted arrows, are shown in Figure 20b–d.

The dropping approach for delivering the grasped objects into the dropping box has been described in Section 3.8. Switching the phases of the dropping state machine is shown in Figure 21a, where the dropping procedure was carried out by two UAVs. Objects were dropped by each UAV at a different time, and thus there was no collision between them. Figure 21b,c show snapshots from the onboard cameras on the UAVs during the dropping maneuver.



**FIGURE 20** Phases of the grasping state machine, UAV position estimation and object detection during grasping of a red static object, with a successful second grasping attempt. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

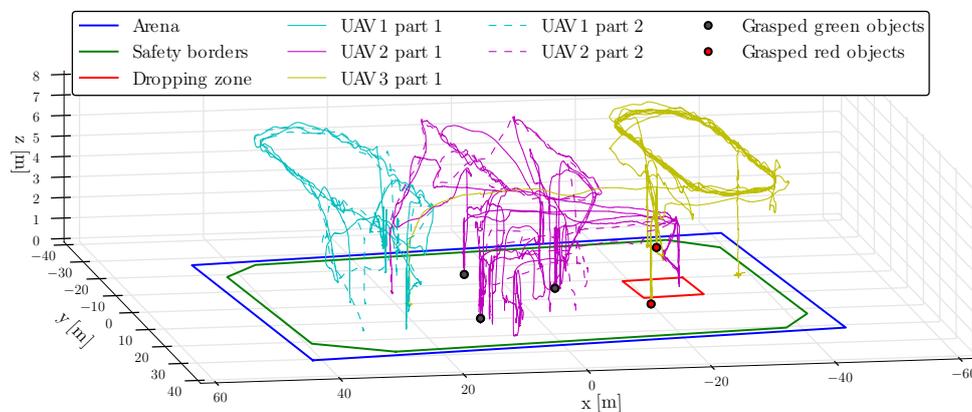


**FIGURE 21** Phases of the dropping state machine of two UAVs. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

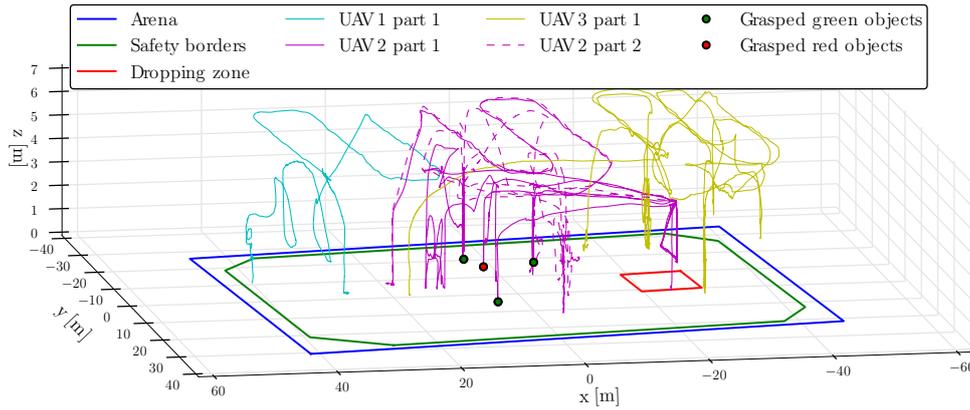
Photos from the competition are shown in Figure 24. The upper image shows the UAV following the static sweeping trajectory. The images in the middle row and the image on the left in the lower part of the figure capture moments when the UAVs were grasping objects. The remaining image shows an object being dropped into the dropping box. In addition, the paths traveled by the UAVs during the first trial of Challenge 3 are shown in Figure 22, and the paths traveled in the second trial of the same challenge are shown in Figure 23. Furthermore, in these graphs, the colored points denote the positions of the objects that were collected.

### 5 | LESSONS LEARNED

Although the competition results can be considered a major success, it was not without hurdles, mainly during implementation, testing and tuning of the proposed system. From the implementation part, it was convenient to develop the system compatible with the ROS, which allows to divide the system into independent components that were implemented separately by different research groups. Furthermore, their testing were significantly simplified by employing the Gazebo robotic simulator together



**FIGURE 22** Paths traveled by individual UAVs during the first trial of Challenge 3. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

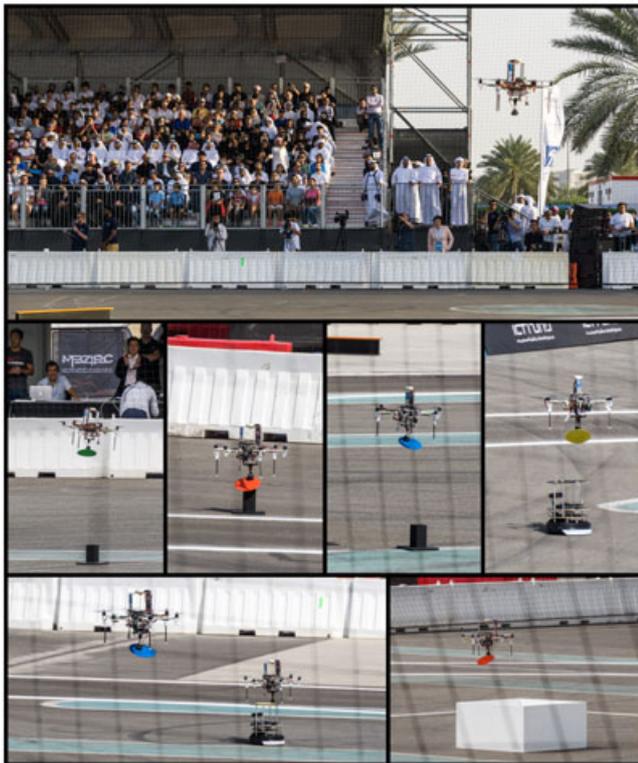


**FIGURE 23** Paths traveled by individual UAVs during the second trial of Challenge 3. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

with the firmware from PixHawk, which speeds up the overall progress of the development.

The required usage of more vehicles simultaneously even increased the complexity of the task. Every UAV is equipped with several sensors

that could be a source of unreliability. By testing the behavior of the proposed system in desert near Abu Dhabi in the United Arab Emirates, our system was well prepared for the environment at the Yas Marina Circuit in Abu Dhabi, where the competition was held. The system was tuned to properly react to strong wind, decreased visibility due to sand, and to problems occurring by intensive light from sun. Hence we stress the significance of the real-world outdoor experiments above simulation, to obtain real sensor data.



**FIGURE 24** Photos from the MBZIRC competition. The upper image shows the UAV while following the static sweeping trajectory. The images in the middle row and the left on the lower part of the figure capture the moments when the UAVs were grasping objects. The remaining image shows an object being dropped into the dropping box. During four trials within the competition (two for Challenge 3 and two for the Grand Challenge), 25 objects overall were placed into the dropping zone (Table 3). <http://mrs.felk.cvut.cz/jfr2018treasurehunt-video4>. MBZIRC: Mohamed Bin Zayed International Robotics Challenge; UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

### 5.1 | Toward a more general solution

Despite our best effort to develop a general solution capable of autonomous searching, picking, and placing objects, several sub-systems have been tailored specifically to the competition scenario. The vision system was designed to locate the objects with colors and shapes specified by the rules of the competition. In the case of an object of more difficult shape and color patterns, a different approach for its localization would be required, for example, based



**FIGURE 25** Team members that were involved in the MBZIRC competition in Abu Dhabi, United Arab Emirates. MBZIRC: Mohamed Bin Zayed International Robotics Challenge [Color figure can be viewed at wileyonlinelibrary.com]

on artificial neural networks. Further, estimation and prediction of the object movement using a car-like model provide a framework suitable for tracking the most common ground vehicles. A more precise model could be used to better estimate state of a specific vehicle (e.g. with differential drive model, or if capable of 3D motion). The presented proactive collision-free planning, using different flying heights and the visibility graph method, has been selected due to simple scenario with three UAVs. A requirement of a higher number of independent flying heights would occur with higher number of deployed UAVs. Then, a different splitting of the arena would be required since it is not efficient to often ascend and descend for the UAV. In this case, each individual UAV will be resolving a possible collision only with other UAVs, that are assigned to arena parts through which the UAV will need to fly. Taking these observations, the presented approach can be applied to various outdoor multirobot scenarios, as shown in our consequent research after the competition listed in Section 1.2.

## 6 | CONCLUSIONS

A system designed for Challenge 3 of the MBZIRC competition has been described in this paper. The paper has focused on the properties of the design that in our opinion were the most important factors leading to the best performance of the system in all trials in the 7 Treasure Hunt challenge. The system is able to solve object manipulation tasks in demanding outdoor environments, and to do so cooperatively in a team of three UAVs.

While many of the methods described here do not represent the bleeding edge of robotics research, they were designed to be versatile and substitutable. This allowed their easy integration into a complex modular system, which enabled efficient testing of the individual modules, making us aware of these modules deficiencies and possible faults during their deployment in real conditions. Our knowledge of the faults encountered during the field tests was reflected in the design of the core module of our system, the FSM. This module ensured that occasional faulty behaviour of the individual modules did not result in a critical failure or system deadlock. Still, the development of this complex system led to numerous significant contributions beyond the state-of-the-art in robotics, which could facilitate the deployment of multi-UAV platforms in challenging scenarios motivated by current needs of the industry. This was the main motivation for our paper and also for the MBZIRC competition itself.

The results shown in numerous realistic simulations in Gazebo and in experiments in a demanding desert environment have been presented in this paper following by analysis of necessary improvements of the system towards more general applications, which go beyond the MBZIRC 2017 competition. However, the most meaningful and credible verification of the performance and the reliability of the system was achieved in the MBZIRC competition, where our approach won the first place Challenge 3, on the basis of achieving the best score among 17 finalists selected from 142 registered teams.

## ACKNOWLEDGMENTS

The outstanding results of this project could not have been achieved without the full cooperation of each member of our team, comprising people from the Czech Technical University in Prague, the University of Pennsylvania, and the University of Lincoln, UK (Figure 25). The work has been supported by CTU grant no. SGS17/187/OHK3/3T/13, Research Center for Informatics project no. CZ.02.1.01/0.0/0.0/16\_019/0000765, the Grant Agency of the Czech Republic under grant no. 17-16900Y, ARL grant W911NF-17-2-0181, ONR grants N00014-07-1-0829, N00014-14-1-0510, and by Khalifa University.

## ORCID

Vojtěch Spurný  <http://orcid.org/0000-0002-9019-1634>

Tomáš Báča  <http://orcid.org/0000-0001-9649-8277>

Robert Pěnička  <http://orcid.org/0000-0001-8549-4932>

Tomáš Krajník  <http://orcid.org/0000-0002-4408-7916>

Justin Thomas  <http://orcid.org/0000-0002-4253-417X>

Dinesh Thakur  <http://orcid.org/0000-0001-5046-8160>

Giuseppe Loianno  <http://orcid.org/0000-0002-3263-5401>

## REFERENCES

- Afolabi, D., Man, K. L., Liang, H.-N., Guan, S.-U., & Krilavičius, T. (2015). 1543. monocular line tracking for the reduction of vibration induced during image acquisition. *Journal of Vibroengineering*, 17, 655–661.
- Baca, T., Loianno, G., & Saska, M. (2016, August). Embedded model predictive control of unmanned micro aerial vehicles. 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 992–997. Miedzyzdroje, Poland.
- Bähmann, R., Schindler, D., Kamel, M., Siegart, R., & Nieto, J. (2017). A decentralized multi-agent unmanned aerial system to search, pick up, and relocate objects. Proceedings of 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), 123–128. Shanghai, China.
- Bohren, J. (2017). SMACH a task-level python execution framework for rapidly composing complex robot behaviors. Robot Operating System (ROS). etrieved from <http://wiki.ros.org/smach>. (cited on July 17, 2018).
- Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11), 120–123.
- Choset, H., & Pignon, P. (1998). *Coverage path planning: The Boustrophedon cellular decomposition* (203–209). London: Springer London.
- Digi International, I. (2017). XBee-PRO RF Modules. <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>. (cited on July 17, 2018).
- Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3), 497–516.
- Fumagalli, M., Stramigioli, S., & Carloni, R. (2016). Mechatronic design of a robotic manipulator for unmanned aerial vehicles. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4843–4848. Daejeon, South Korea.
- Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12), 1258–1276.
- Ghadiok, V., Goldin, J., & Ren, W. (2012). On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor. *Autonomous Robots*, 33(1), 41–68.
- Gioioso, G., Franchi, A., Salvietti, G., Scheggi, S., & Prattichizzo, D. (2014). The flying hand: A formation of uavs for cooperative aerial

- tele-manipulation. 2014 IEEE International Conference on Robotics and Automation (ICRA), 4335–4341. Hong Kong, China.
- Heredia, G., Jimenez-Cano, A. E., Sanchez, I., Llorente, D., Vega, V., Braga, J., ... Ollero, A. (2014). Control of a multirotor outdoor aerial manipulator. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3417–3422. Chicago, IL.
- Herissé, B., Hamel, T., Mahony, R., & Russotto, F. X. (2012). Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics*, 28(1), 77–89.
- Kamel, M., Comari, S., & Siegwart, R. (2016). Full-body multi-objective controller for aerial manipulation. In 2016 24th Mediterranean Conference on Control and Automation (MED), 659–664. Athens, Greece.
- Kannan, S., Quintanar-Guzman, S., Dentler, J., Olivares-Mendez, M. A., & Voos, H. (2016). Control of aerial manipulation vehicle in operational space. In 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 1–4. Ploiesti, Romania.
- Kessens, C. C., Thomas, J., Desai, J. P., & Kumar, V. (2016). Versatile aerial grasping using self-sealing suction. In 2016 IEEE International Conference on Robotics and Automation (ICRA), 3249–3254. Stockholm, Sweden.
- Kim, S., Seo, H., Choi, S., & Kim, H. J. (2016). Vision-guided aerial manipulation using a multirotor with a robotic arm. *IEEE/ASME Transactions on Mechatronics*, 21(4), 1912–1923.
- Korpela, C. M., Danko, T. W., & Oh, P. Y. (2012). Mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent and Robotic Systems*, 65(1), 93–101.
- Krajník, T., Nitsche, M., Faigl, J., Vaněk, P., Saska, M., Přeucil, L., & Mejail, M. (2014). A practical multirobot localization system. *Journal of Intelligent and Robotic Systems*, 76(3-4), 539–562.
- Lee, T., Leok, M., & McClamroch, N. H. (2013). Nonlinear robust tracking control of a quadrotor uav on se(3). *Asian Journal of Control*, 15(2), 391–408.
- Lippiello, V., Cacace, J., Santamaria-Navarro, A., Andrade-Cetto, J., Trujillo, M., Esteves, Y. R., & Viguria, A. (2016). Hybrid visual servoing with hierarchical task composition for aerial manipulation. *IEEE Robotics and Automation Letters*, 1(1), 259–266.
- Lozano-Pérez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 560–570.
- Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10), 761–767.
- Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., & Pollefeys, M. (2012). Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1), 21–39.
- Mellinger, D., & Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In 2011 IEEE International Conference on Robotics and Automation (ICRA), 2520–2525. Shanghai, China.
- Mellinger, D., Lindsey, Q., Shomin, M., & Kumar, V. (2011). Design, modeling, estimation and control for aerial grasping and manipulation. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2668–2673. San Francisco, CA.
- Mellinger, D., Shomin, M., Michael, N., & Kumar, V. (2013). *Cooperative grasping and transport using multiple quadrotors* (545–558). Berlin, Heidelberg: Springer.
- Mobius ActionCam (2018). <http://www.mobius-actioncam.com/mobius/wp-content/uploads/2015/01/Mobius-Manual-23jan15a.pdf>. (cited on September 20, 2018).
- Morton, K., & Toro, L. F. G. (2016). Development of a robust framework for an outdoor mobile manipulation uav. 2016 IEEE Aerospace Conference, 1–8. Big Sky, MT.
- NicaDrone (2017). OpenGrab EPM v3 electropermanent magnet. <http://nicadrone.com>. (cited on July 17, 2018).
- Nieuwenhuisen, M., Beul, M., Rosu, R. A., Quenzel, J., Pavlichenko, D., Houben, S., & Behnke, S. (2017). Collaborative object picking and delivery with a team of micro aerial vehicles at mbzirc. In 2017 European Conference on Mobile Robotics (ECMR), 1–6. Paris, France.
- Orsag, M., Korpela, C., Pekala, M., & Oh, P. (2013). Stability control in aerial manipulation. In 2013 American Control Conference, 5581–5586. Washington, DC.
- Parra-Vega, V., Sanchez, A., Izaguirre, C., Garcia, O., & Ruiz-Sanchez, F. (2013). Toward aerial grasping and manipulation with multiple uavs. *Journal of Intelligent and Robotic Systems*, 70(1), 575–593.
- Pěnička, R., Faigl, J., Váňa, P., & Saska, M. (2017). Dubinsorienting problem. *IEEE Robotics and Automation Letters*, 2(2), 1210–1217.
- Pěnička, R., Saska, M., Reymann, C., & Lacroix, S. (2017). Reactive dubins traveling salesman problem for replanning of information gathering by uavs. In European Conference of Mobile Robotics (ECMR), 1–6. Paris, France.
- Pounds, P. E. I., Bersak, D. R., & Dollar, A. M. (2011a). Grasping from the air: Hovering capture and load stability. In 2011 IEEE International Conference on Robotics and Automation, 2491–2498. Shanghai, China.
- Pounds, P. E. I., Bersak, D. R., & Dollar, A. M. (2011b). Practical aerial grasping of unstructured objects. In 2011 IEEE Conference on Technologies for Practical Robot Applications, 99–104. Woburn, MA.
- RamonSoria, P., Arrue, B. C., & Ollero, A. (2017). Detection, location and grasping objects using a stereo sensor on uav in outdoor environments. *Sensors*, 17(1), 103.
- RamonSoria, P., Bevec, R., Arrue, B. C., Ude, A., & Ollero, A. (2016). Extracting objects for aerial manipulation on uavs using low cost stereo sensors. *Sensors*, 16(5), 700.
- Santamaria-Navarro, A., Grosch, P., Lippiello, V., Sola, J., & Andrade-Cetto, J. (2017). Uncalibrated visual servo for unmanned aerial manipulation. *IEEE/ASME Transactions on Mechatronics*, 22(4), 1610–1621.
- Saska, M. (2017). Large sensors with adaptive shape realised by selfstabilised compact groups of micro aerial vehicles. In International Symposium on Robotic Research. Puerto Varas, Chile.
- Saska, M., Kratky, V., Spurny, V., & Baca, T. (2017). Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control. In 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 1–8. Limassol, Cyprus.
- Saska, M., Kulich, M., & Preucil, L. (2006). Elliptic net-a path planning algorithm for dynamic environments. In *ICINCO-RA*, 372–377. Setúbal, Portugal.
- Spica, R., Franchi, A., Oriolo, G., Bühlhoff, H. H., & Giordano, P. R. (2012). Aerial grasping of a moving target with a quadrotor uav. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 4985–4992. Vilamoura, Portugal.
- Tersus-GNSS (2017). PRECIS-BX305 GNSS RTK Board. <https://www.tersus-gnss.com>. (cited on July 17, 2018).
- Thomas, J., Loianno, G., Polin, J., Sreenath, K., & Kumar, V. (2014). Toward autonomous avian-inspired grasping for microaerial vehicles. *Bioinspiration and Biomimetics*, 9(2), 025010.
- Tiderko, A. (2017). multimaster\_fkcie - Multi-master ROS Package, Robot Operating System (ROS). <http://wiki.ros.org/smach>. (cited on July 17, 2018).
- Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6), 854–874.

**How to cite this article:** Spurný V, Báča T, Saska M, et al. Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *J Field Robotics*. 2018;1–24. <https://doi.org/10.1002/rob.21816>