Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems

Cooperative Sensing by a Group of Unmanned Aerial Vehicles

Doctoral Thesis

Ing. Tomáš Báča

Prague, January 2021

Ph.D. programme: Electrical Engineering and Information Technology
Branch of study: Artificial Intelligence and Biocybernetics

Supervisor: Ing. Martin Saska, Dr. rer. nat.
Supervisor-Specialist: Ing. Michal Platkevic, Ph.D.

Acknowledgments

Firstly, I would like to express my gratitude to my family for providing me with material and mental support during my studies. I am grateful that they allowed me to pursue a student's and a researcher's path, a career that is not known for its immediate benefits and securities. Secondly, my thanks go to Martin Saska, my supervisor, colleague, and friend. Thank you for your guidance, leadership, and care that you provide to all your students. I am grateful for trusting in me, and I also do not take for granted the creative freedom I was given during my studies and work within the group. Furthermore, I thank Michal Platkevic, my co-supervisor, for his guidance in the radiation dosimetry field. I am also grateful to Tomas Krajnik for motivating me to apply to the CTU in Prague and later supervising me during my first steps in aerial robotics. Finally, my thanks go to all the present and past members of the MRS group. The past six years have been a *bumpy ride*, and I am grateful that I could make it with you. Specifically, I would like to thank my colleagues Vojtech Spurny, Daniel Hert, and Robert Penicka, who often *shared the front seats* with me. My path would have probably been different without you. I am grateful to all other members of FEE, CTU in Prague, for creating a flourishing research environment where collaboration is encouraged.

My following thanks go to everyone who allowed me to work on the projects related to radiation dosimetry for space applications. In my opinion, successful orchestration of research in space instrumentation is comparably more difficult than research in mobile robotics. Even though our results might be small in the grand scheme of things, the path towards them was no less difficult given the tight funding and limited know-how. Among others, I am grateful to Vladimir Daniel, and Petr Svoboda (Czech Aerospace Research Institute), Adolf Inneman, and Veronika Marsikova (Rigaku Innovative Technologies, s.r.o.), Jan Jakubek, and Michal Platkevic (at the time at the IEAP, CTU in Prague). Without you, the VZLUSAT-1 nanosatellite would not have seen the light of day. Moreover, I am grateful to our colleagues at the University of Iowa and Pennsylvania State University. Thank you for the opportunity to collaborate, despite the probably asymmetrical gains from the collaboration. I want to thank, among others, Randall Mc'Entaffer, Ted Schultz, and James Tutt for their hospitality during my visits to their laboratories.

During my Ph.D. studies, my work had been supported by the taxpayers of the Czech Republic through a Ph.D. scholarship. My work had also been supported by the Czech Technical University grants SGS15/157/OHK3/2T/13 and SGS17/187/OHK3/3T/13. The Ministry of education of the Czech Republic supported the work by grant no. 7AMB16FR017, and no. LH11053, and by OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". The Czech Science Foundation supported this work through projects no. 17-16900Y, no. 18-10088Y, and no. 20-10280S. The Technology Agency of the Czech Republic supported this work through project no. FW01010317. The European Union's Horizon 2020 research and innovation program supported this work under grant no. 871479. The National Grid Infrastructure MetaCentrum provided access to computing resources under CESNET 569/2015 and LM2015042. The Khalifa University of Science funded our MBZIRC 2017 and MBZIRC 2020 participation that also motivated this work. The work on the outer space radiation dosimetry was supported by the Technology Agency of the Czech Republic projects no. TA03011329, no. TA04011295, the Czech Science Foundation projects no. GA13-33324S, GJ18-10088Y, and the project MSMT LH14039 of the Ministry of Education of the Czech Republic. The work has been done on behalf of Medipix2 and Medipix3 collaborations.

Copyright

This thesis is a compilation of several articles published and submitted during my Ph.D. studies. The included publications are presented under the copyrights of IEEE, Springer, Elsevier, IOP Publishing, and Wiley for posting the works for internal institutional uses. The works are protected by the copyrights of respective publishers and can not be further reprinted without the publishers' permission.

Abstract

The study of autonomous Unmanned Aerial Vehicles (UAVs) has become a prominent sub-field of mobile robotics. Multirotor unmanned helicopters are systems useful on many levels of research. Multicopters in research serve as a plant for testing new techniques for feedback control of dynamical systems, as sensor carriers for remote sensing applications, and as units of multi-robot systems. A considerable amount of work is also being invested into research of principal sub-systems of multirotor UAVs. Real-time localization, state estimation, modeling, feedback control, planning, and navigation are well-established and active research fields, each contributing to making UAVs autonomous, robust, and safe. This thesis focuses on remote sensing with UAV systems. The first part of the thesis is dedicated to development of a novel UAV control system, designed for real-world testing and evaluation of new methods. The control and estimation system supports replicable research by allowing realistic simulations and real-world experiments. The second part of the thesis is motivated by the challenges of the MBZIRC 2017 and 2020 robotic competitions. Specifically, we study the applications of groups of UAVs to fulfill the autonomous mission of collaborative localization and gathering of objects and their delivery to desired locations. Initially, only small metal disc-shaped objects are gathered by the UAVs into a large box. Later, the challenge was elevated into a task of autonomous brick wall construction by a group of UAVs. Furthermore, we study the problem of concluding a UAV mission by autonomously landing a UAV on a moving ground vehicle. All the challenges were tackled by developing a complete UAV system and successfully performing an extensive experimental evaluation, that was finished by winning the international competitions. The third part of the thesis focuses on a specific remote sensing field, the localization and mapping of ionizing radiation by UAVs. The ongoing research investigates the use of hybrid pixel radiation sensors from the Timepix family onboard UAVs for directional and spatial localization of compact radiation sources.

Keywords Unmanned Aerial Vehicles, Mobile Robotics, Remote Sensing, Ionizing Radiation, Dosimetry

Abstrakt

Výzkum na poli autonomních bezpilotních prostředků (UAV) se stal významným oborem mobilní robotiky. Vícerotorové bezpilotní helikoptéry jsou užitečné v mnoha úrovních výzkumu. Vícerotorové UAV slouží jako systémy pro testování nových technik v oboru zpětnovazebního řízení dynamických systémů, jako nosiče senzorů pro vzdálené měření a také jako součásti výzkumu multi-robotických systémů. Značné úsilí věnujeme výzkumu klíčových sub-systémů fungování vícerotorových UAV. Systémy lokalizace, estimace stavu, modelování, zpětnovazební řízení, plánování pohybu, a autonomní navigace jsou již zavedené a aktivními poli výzkumu. Každé z nich přispívá k bezpečné a robustní autonomii bezpilotních prostředků. Tato práce se zabývá vzdáleným měřením pomocí autonomních bezpilotních systémů. První část práce je věnovaná vývoji nové řídicí platformy pro vícerotorové UAV, která byla navržena za účelem testování a vyhodnocování nových metod pro UAV v reálném prostředí. Tento systém pro řízení a odhadování stavů UAV umožňuje replikovatelný výzkum a poskytuje možnost realistických simulací a testování na UAV v reálném prostředí mimo laboratoř. Druhá část této práce je motivována výzvami mezinárodních robotických soutěží MBZIRC 2017 a 2020. Konkrétně zde představujeme aplikaci skupin autonomních UAV misí pro kolaborativní sběr objektů a jejich dopravení na zadaná místa. Nejprve jsme se zabývali sběrem malých kovových disků, které byly pomocí UAV autonomně dopraveny do sběrného boxu. Poté byl tento úkol rozšířen na automatické stavění cihlové zdi pomocí skupiny UAV. Dále jsme se zabývali problémem zakončování autonomních misí, konkrétně autonomním přistáním UAV na jedoucím vozidle. Všechny tyto výzvy byly vyřešeny spolu s vývojem kompletních bezpilotních systémů a jejich experimentální evaluace, která byla završena výhrami v obou mezinárodních soutěžích. Třetí část této práce se zaměřuje na specifický podobor vzdáleného měření, a to na lokalizaci a mapování zdrojů ionizujícího záření pomocí UAV. Probíhající výzkum se zabývá využitím hybridních pixelových senzorů radiace z rodiny detektorů Timepix. V této práci zkoumáme využití detektorů Timepix pro určování směrové a prostorové informace o kompaktních zdrojích ionizující radiace.

Klíčová slova Bezpilotní Prostředky, Mobilní Robotika, Vzdálené Měření, Ionizující Radiace, Dozimetrie

Abbreviations

API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CTU	Czech Technical University
CZT	Cadmium Zinc Telluride
CdTe	Cadmium Telluride
DARPA	Defense Advanced Research Projects Agency
FDNPP	Fukushima Daiichi Nuclear Power Plant
FOV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
ISS	International Space Station
LEO	Low-Earth Orbit
LiDAR	Light Detection and Ranging
MAV	Micro Aerial Vehicle
MBZIRC	Mohamed Bin Zayed International Robotics Challenge
MPC	Model Predictive Control
MRS	Multi-robot Systems Group
RAS	Robotics and Automation Society
ROS	Robot Operating System
SLAM	Simultaneous Localization And Mapping
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UKF	Unscented Kalman Filter
VZLU	Czech Aerospace Research Center

Contents

1 Introduction **1**

2 Contributions and Related Work **4**

2.1 Author's publications and contributions 4

2.2 Multirotor UAV control system 4

2.3 Remote sensing and data collection by UAVs 7

2.4 Measuring ionizing radiation, mapping and localization 9

2.4.1 Outer space radiation dosimetry and mapping 9

2.4.2 Localization and mapping of ionizing radiation sources by UAVs 10

3 Research-focused UAV Platform **12**

4 Remote Sensing by UAVs **49**

5 Radiation Dosimetry by UAVs **114**

6 Discussion and Results **141**

6.1 Research-focused UAV System 141

6.2 Advances in remote sensing by UAVs 142

6.3 Ionizing radiation sources localization 143

6.4 Summary 146

7 Conclusion **147**

A References **148**

A.1 Thesis core publications 148

A.2 Thesis-related author's publications 149

A.3 Partially-related author's publications 151

A.4 Unrelated author's publications 152

A.5 Cited references 152

B Citations of Author's Publications **162**

Chapter 1

Introduction

The emergence of small UAVs has created a new active field of mobile robotics. The rise of multirotor helicopters spawned revolutionary possibilities of remote sensing and data gathering. Unlike traditional ground robots, multirotor helicopters combine potentially fast and agile movement through a 3D environment with the ability to hover in place. Both traits require overcoming complex technical challenges as well as offering significant advantages over ground robots. The challenges arise from the inherently unstable dynamics of multirotor helicopters [146], [158]. Uninterrupted feedback control actions are required to maintain the machine in the desired state [160]. Moreover, feedback control of agile UAVs is vitally dependent on a smooth and feasible state estimate. Both the UAV state estimation [65], [125], [128], [175] and feedback control [103], [127], [168] have been intensively studied during the past decade and are still very active fields of research. Multirotor helicopters are a versatile platform for carrying out remote sensing [132], [140], environmental sampling, and providing technical support and aid in natural disaster rescue operations [121], [137].

Remote sensing, e.g., of an active volcano, can be traditionally performed by a stationary sensory system [81], an airplane [57], a satellite [50], a remotely controlled robot [120], and of course, a UAV [47], [89]. In recent years, autonomous robotic remote sensing became available with the emergence of autonomous systems. Onboard autonomy is used to control a robotics system when employing a human operator might not be possible, e.g., deep underground [73], [86] or in a vicinity of a damaged nuclear power plant [76]. However, multi-robotic distributed sensing is still in its infancy. Although distributing the process offers robustness through redundancy and a potential increase in information yield, it also poses new challenges in multi-robotic coordination and sensor fusion. The same applies to the use of multiple multirotor UAVs with even more challenges. On its own, control of multi-UAV swarms and formations is a challenging subfield that has direct implications to the field of distributed remote sensing. One of the most iconic subfields of remote sensing is the remote sensing of ionizing radiation due to the unusual nature of how it is measured [94], but also due to the danger ionizing radiation poses to living organisms. Recent advances in semiconductor technologies allow the fabrication of small semiconductor radiation detectors, which opened up the possibility to measure ionizing radiation onboard small UAVs. Thanks to onboard autonomy, small UAVs can be deployed to map ionizing radiation and even localize radioactive sources autonomously.

This thesis focuses on advances in replicable research with autonomous multirotor helicopters and their use for remote sensing and the remote sensing of ionizing radiation. The thesis's objectives were significantly shaped by the active participation of the Multi-robot Systems Group (MRS) research group in the 2017 and 2020 rounds of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC). MBZIRC proposed a set of robotics challenges that pushed the boundaries of UAV autonomy. The challenges ranged from an autonomous landing on a moving car by an unmanned helicopter, the collaborative gathering of small objects by a fleet of UAVs, and to the autonomous brick wall-building by a fleet of UAVs.

Those tasks provided opportunities and conditions to take state-of-the-art techniques outside of a laboratory and compare the advances directly with the best university teams from around the world.

Furthermore, interdisciplinary research is being pursued on the localization of ionizing radiation sources by UAVs. The research closely follows advances in imaging pixel radiation detectors (initially developed for medical imaging) for remote sensing applications of space dosimetry and X-ray imaging. The thesis author has made his initial contributions in the field by developing a radiation remote sensing module for the first Czech CubeSat[1], the VZLUSAT-1. The module houses the Timepix [172] hybrid CMOS detector developed at CERN in order to measure solar X-rays and background radiation in the Low-Earth Orbit (LEO). One of the thesis's goals is to transfer this know-how to the field of small Unmanned Aerial Vehicles.

Experimental verification of novel methods is crucial to provide objective evaluation and to support new publications. Although the accompanying subfields of cybernetics — machine learning and computer vision — benefit significantly from evaluating and comparing new methods on datasets, the same approach can be rarely used in mobile robotics and field robotics. Even realistic simulation do not fully substitute the testing and verification using a real UAV equipped with real sensors, most importantly, outside of laboratory conditions (see Fig. 1.1). However, conducting real-world experiments requires an onboard control platform that can satisfy the tested method's needs. Despite the plethora of existing solutions [58], [92], [111], [122], [124], [154], none provide all the features needed to support the work presented in the thesis. Not all the platforms support experiments both indoors and outdoors. The UAV state estimation is often limited to a single localization approach. Mid-air switching of controllers and control reference generators is also not standard. Finally, the low-level control output using a desired UAV attitude rate is rarely present.

The objectives of the thesis are summarized as follows:

(1) Development of a control system for real-world deployment of UAVs, verification of new methods for control, remote sensing, and deployment in indoor and outdoor environments. Despite many platforms for multirotor UAVs control and deployment being available, they lack features necessary for real-world testing and deployment of the methods within the focus of this thesis and the focus of the Multi-robot Systems Group group at Czech Technical University (CTU) in Prague. Therefore, the author's first objective and a long-term effort are to develop a modular UAV control system. The control system should allow safe indoor and outdoor deployment of multirotor helicopters. The system should also allow verification of high-level methods for motion planning, multi-UAV swarming, and formation flying. Finally, the system should also allow basic research on low-level control and stabilization of the multirotor UAV dynamics.

(2) Research of methods of collaborative remote sensing by a group of Unmanned Aerial Vehicles in real-world non-laboratory conditions. Collaboratively executing UAV missions poses challenges on many onboard autonomy levels, e.g., task allocation, estimation, motion planning, and control. Furthermore, mutual communication between the UAVs might be unreliable or completely unavailable. Therefore, sharing real-time sensor readings to pursue a common goal might not be possible in all circumstances. Moreover, mutual collisions between the UAVs can be expected if the UAVs are guided by common goals. The objective is to explore and push the field of collaborative remote sensing and deployment of UAV in complex robotics tasks forwards.

[1] CubeSat is a nanosatellite made up of $1\,\mathrm{dm}^3$ cubic modules.

Figure 1.1: Illustration of outcomes of the thesis author's contributions: deployments of multirotor UAVs as well as radiation measurements for space applications.

(3) Advancing the field of ionizing radiation dosimetry, mapping, and localization of compact sources by Unmanned Aerial Vehicles. Ionizing radiation has been traditionally measured onboard UAVs using dosimeters [114], [133], [155], [161] — sensors measuring the intensity of incoming radiation. Often, the intensity is utilized only to estimate the scalar field of radiation intensity. Rarely, direction measurement can be obtained with an additional device, e.g., the optical collimator or a coded aperture. However, those solutions are not well-suited for small UAVs due to the heavy weight of the sensor equipment. This work aims to push the state of the art by utilizing miniature semiconductor pixel detectors [172], and novel event-based radiation detectors [147] onboard Micro Aerial Vehicles (MAVs).

The rest of the thesis is organized as follows. This thesis is a compilation of 8 included core publications, referenced as 1c – 8c. Furthermore, the thesis is supported by additional authored publications, referenced as 9a – 44a. Firstly, the state of the art is summarized in Chapter 2. Chapter 3 introduces the publications related to the developed UAV platform. Chapter 4 covers the publications related to the multi-UAV sensing and deployment. Finally, Chapter 5 presents publications on radiation measurement, localization, and mapping.

Chapter 2

Contributions and Related Work

2.1 Author's publications and contributions

Figure 2.1 shows a publication graph composed of accepted peer-reviewed publications and publications that have been submitted in the time of writing this thesis. The publications are split into four main categories. The first category, a *Pre-Ph.D. research*, consists of publications based upon the research done before the author started the pursue of Ph.D. ($\leq$ 2015). Although, some of these were written and submitted during the author's Ph.D. studies. The second category is the research stream marked as *UAV platform for research* validation and reliable deployment of novel methods in control, navigation, formation flying, and swarming. Thirdly, the largest group of the author's work is from the field of *multi-UAV remote sensing*, swarming, and other deployment. Lastly, the fourth category is the research of *ionizing radiation dosimetry, imaging*, localization and mapping. The author conducts an interdisciplinary transfer from the space-oriented physics field to the field of Unmanned Aerial Vehicles.

2.2 Multirotor UAV control system

Commercial UAV systems are often closed-source and provide features tailored for photographers, video makers, and hobby pilots. Autonomous operation of commercial UAVs is typically limited to a single aerial vehicle flying outdoors under a Global Navigation Satellite System (GNSS) localization while following a set of waypoints. Therefore, commercial platforms are rarely used for research. If so, then in a field where the UAV is only considered a *sensor carrier*, without added onboard autonomy.

Research-focused UAVs are most commonly equipped with a low-level embedded flight controller. Available flight controllers [82] range from feature-packed open-source systems, such as Pixhawk, to proprietary commercial units manufactured by DJI. Table 2.1 shows a comparison of often used solutions. Pixhawk is often used in research projects (including ours), typically running either of the two open-source firmware: PX4 [131] and ArduPilot[1]. Although all of these flight stacks provide sophisticated features up to waypoint tracking and mission execution, the features are rarely used within real-world applications. Instead, researchers use other onboard computers to execute a custom localization system, state estimators, and flight controllers, and only low-level control commands are provided for the embedded flight controller.

Several comparable UAV systems have been published and released. Table 2.2 compares existing solution with the system proposed in this publication.

The RotorS [111] simulator is an initial release for the Aeroworks EU project[2]. It pro-

[1] `http://ardupilot.org`

[2] Aeroworks EU project, `http://www.aeroworks2020.eu`.

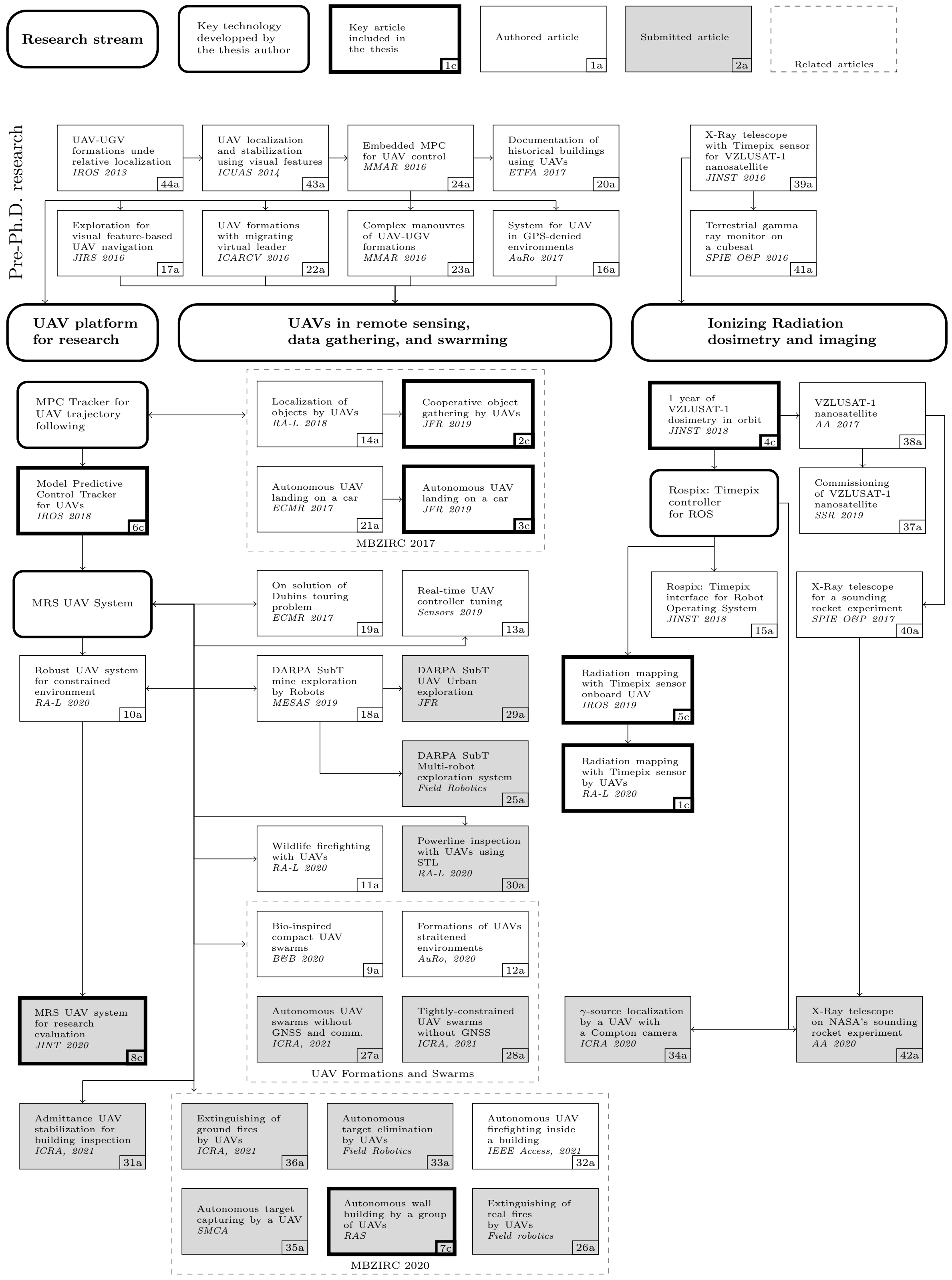

Research stream
Key technology developped by the thesis author
Key article included in the thesis 1c
Authored article 1a
Submitted article 2a
Related articles
Pre-Ph.D. research
UAV-UGV formations unde relative localization IROS 2013 44a
UAV localization and stabilization using visual features ICUAS 2014 43a
Embedded MPC for UAV control MMAR 2016 24a
Documentation of historical buildings using UAVs ETFA 2017 20a
X-Ray telescope with Timepix sensor for VZLUSAT-1 nanosatellite JINST 2016 39a
Exploration for visual feature-based UAV navigation JIRS 2016 17a
UAV formations with migrating virtual leader ICARCV 2016 22a
Complex manouvres of UAV-UGV formations MMAR 2016 23a
System for UAV in GPS-denied environments AuRo 2017 16a
Terrestrial gamma ray monitor on a cubesat SPIE O&P 2016 41a
UAV platform for research
UAVs in remote sensing, data gathering, and swarming
Ionizing Radiation dosimetry and imaging
MPC Tracker for UAV trajectory following
Localization of objects by UAVs RA-L 2018 14a
Cooperative object gathering by UAVs JFR 2019 2c
1 year of VZLUSAT-1 dosimetry in orbit JINST 2018 4c
VZLUSAT-1 nanosatellite AA 2017 38a
Model Predictive Control Tracker for UAVs IROS 2018 6c
Autonomous UAV landing on a car ECMR 2017 21a
Autonomous UAV landing on a car JFR 2019 3c
MBZIRC 2017
Rospix: Timepix controller for ROS
Commissioning of VZLUSAT-1 nanosatellite SSR 2019 37a
MRS UAV System
On solution of Dubins touring problem ECMR 2017 19a
Real-time UAV controller tuning Sensors 2019 13a
Rospix: Timepix interface for Robot Operating System JINST 2018 15a
X-Ray telescope for a sounding rocket experiment SPIE O&P 2017 40a
Robust UAV system for constrained environment RA-L 2020 10a
DARPA SubT mine exploration by Robots MESAS 2019 18a
DARPA SubT UAV Urban exploration JFR 29a
Radiation mapping with Timepix sensor onboard UAV IROS 2019 5c
DARPA SubT Multi-robot exploration system Field Robotics 25a
Radiation mapping with Timepix sensor by UAVs RA-L 2020 1c
Wildlife firefighting with UAVs RA-L 2020 11a
Powerline inspection with UAVs using STL RA-L 2020 30a
Bio-inspired compact UAV swarms B&B 2020 9a
Formations of UAVs straitened environments AuRo, 2020 12a
MRS UAV system for research evaluation JINT 2020 8c
Autonomous UAV swarms without GNSS and comm. ICRA, 2021 27a
Tightly-constrained UAV swarms without GNSS ICRA, 2021 28a
γ-source localization by a UAV with a Compton camera ICRA 2020 34a
X-Ray telescope on NASA's sounding rocket experiment AA 2020 42a
UAV Formations and Swarms
Admittance UAV stabilization for building inspection ICRA, 2021 31a
Extinguishing of ground fires by UAVs ICRA, 2021 36a
Autonomous target elimination by UAVs Field Robotics 33a
Autonomous UAV firefighting inside a building IEEE Access, 2021 32a
Autonomous target capturing by a UAV SMCA 35a
Autonomous wall building by a group of UAVs RAS 7c
Extinguishing of real fires by UAVs Field robotics 26a
MBZIRC 2020

Figure 2.1: Diagram of research performed by the thesis author from 2013 to 2020 in the fields of UAV control, remote sensing and its applications, and the field of ionizing radiation imaging, dosimetry, mapping and localization. The article numbering reflects the works in the reference section of this thesis.

vides Gazebo-based simulation of the now discontinued *Ascending Technologies* UAV system. The control pipeline features are basic, with little potential for transfer to real-world conditions. The system does not appear to be kept up-to-date, which gradually diminishes its usability and applicability. Moreover, the latest supported version of ROS is *ROS Kinetic*, which potentially provides lower compatibility with newer hardware and software.

OpenUAV[3] [92] is a UAV swarm simulation testbed. The system does not appear to allow transfer to a real-world setting, and is designed only to support prototyping of basic research in swarming. The UAVs are assumed to be controlled and localized solely using an embedded flight controller with PX4 firmware. This is comparable hardly with the numerous sensors and localization systems that our system allows to simulate and to be used in a real-world scenario.

ReCOPTER[4] [124] proposes an open-source multirotor system for research. The available materials were released as supporting material for the published paper. However, no software was attached, and the materials have not been updated since. Similarly, a framework for drone control using the Vicon localization system named MAVwork[5] [154] was published in 2011, but has not been updated since. Although sources were made available, they offered only basic features that would be difficult to transfer into a real-world scenario.

The XTDrone[6] [58] simulation testbed offers many complex functionalities that are comparable with our proposed system, including simulation of onboard sensors and complex localization systems. However, the control pipeline relies entirely on the PX4 embedded control software. This significantly limits any transfer to a custom hardware platform, or even the ability to simulate realistic conditions using onboard localization systems. Thus, the use of XTDrone outside laboratory conditions is mostly limited to Global Positioning System (GPS)-localized flight in a non-cluttered outdoor environment.

The full-stack Aerostack system[7] [122], [123] was designed for deployment of multirotor UAVs. The system is continuously being updated, and it offers an option to transfer to a real-world platform. According to the preprint [56] where the authors used Aerostack during the MBZIRC 2020 competition, the system's real-world deployment is possible. However, with the used DJI-based flight controller, the control command supplied to the underlying embedded control layer are limited to desired orientation and thrust. This level of control limits the potential precision and control authority comparing to our system. Furthermore, the system lacks the feature of switching between multiple frames of reference, which is one of our system's contributions. As it happens, the team of authors of Aerostack did not compete in the wall-building challenge of MBZIRC 2020, in which we found the feature to be crucial to precisely collect bricks by a group of UAVs.

Besides the Aerostack system, no other existing platform provides a full-stack system for a multirotor UAV that is actively being supported and updated. Many publications provide accompanying software sources that are released without being further updated. By contrast, we have decided to publish and release our working system with all its components to allow members of the research community, research teams, and students to engage in UAV research as effortlessly as possible. We aim to provide a thoroughly-documented open-source system to allow researchers and students to shorten their initial learning curve and to focus on their

[3] OpenUAV, `http://github.com/Open-UAV`

[4] ReCOPTER, `http://github.com/thedinuka/ReCOPTER`

[5] MAVwork, `http://github.com/uavster/mavwork`

[6] XTDrone, `http://github.com/robin-shaun/XTDrone`

[7] Aerostack, `http://github.com/Vision4UAV/Aerostack`

research instead of developing yet another control pipeline.

platform	open source	modular	SITL/HITL	outside lab	rate input
Pixhawk	**SW & HW**	+	+	+	+
DJI	-	-	- (proprietary)	+	-
Ardupilot	SW	+	+	+	-
Parrot	SW	-	+	-	-

Table 2.1: Comparison of commonly-used embedded flight controllers and low-level control systems. The Pixhawk flight controller was chosen due to several factors: both the hardware and software is open-source, the controller is modular enough to be used on a variety of custom multirotor platforms, Pixhawk supports both hardware- and software-in-the-loop simulation, can be used outside of laboratory conditions, and supports attitude rate input.

platform	modular	simulation	outside lab.	multi-frame localization	rate output	last updated	reference
MRS UAV system	+	+	+	+	+	**2020**	[8c]
Aerostack	+	+	+	-	-	2020	[122]
XTDrone	+	+	-	-	-	2020	[58]
RotorS	+	+	-	-	+	2020	[111]
OpenUAV	-	+	-	-	-	2020	[92]
ReCOPTER	-	-	-	-	-	2015	[124]
MAVwork	+	-	-	-	-	2013	[154]

Table 2.2: Comparison of high-level open-source UAV systems. The proposed system is extensible and modular, comes with an extensive simulation environment, is designed to be used outside of laboratory conditions, provides the novel multi-frame localization estimator, and supplies the attitude rate command to the underlying embedded flight controller.

2.3 Remote sensing and data collection by UAVs

Remote sensing by UAVs has been common since the first aerial platforms become available [132], [140]. Foliage and vegetation monitoring were among the first applications [162]. In earlier attempts, thermal cameras provided information that was later used to make educating actions during farming in the scanned field [170]. Also, water status monitoring within a vineyard [157] can be conducted by multispectral cameras onboard a UAV. Nowadays, precise agriculture is a promising agricultural technique that utilizes multispectral cameras to decide upon adjustments within the agriculture processes [49], [51], [59]. However, it is often conducted by a single UAV dedicated to monitoring the field from a birds-eye perspective or dedicated to delivering chemicals to precise locations within the field.

Distributing the remote sensing task provides measurements in different places at once. Multiple UAVs equipped with gimballed cameras can track a ground target more reliable than a single UAV [148]. The redundancy increases the robustness of the system. Missing information due to, e.g., image dropouts or target occlusions, can be substituted in real time by communication with neighboring UAVs [46], [126]. Moreover, distributed state estimation can be applied in real time [171], [173], [177]. When the states of an aerial target are estimated in real time, a group of UAVs can pursue the target and intercept its path [109].

A likely application of distributed sensing is during rescue operations and environmental monitoring of natural disasters [87]. Distributed monitoring of wildfires can provide crucial real-time information to firefighters [55], [137]. Similarly, monitoring the current state of floods can aid during rescue operations [52], [121].

Simultaneously deploying multiple UAVs poses challenges on various levels of the onboard autonomy architecture: automatic control, localization, planning, communication, and collision avoidance. Multi-UAV path planning and coordination algorithms provide the aircraft with feasible plans to fulfill the given task. The coordinate principles are often split into two categories: *formation* and *swarm* control. UAV formation control focuses on explicit coordination given either a centralized localization of all the agents or a mutual communication between the agents [104], [136], [144], [156]. On the other hand, UAV swarms is a bio-inspired decentralized technique that attempts to overcome the global localization or communication by applying behavioral rules [70], [108], [163].

We aim at the specific subfield of remote sensing related to autonomous experimental deployment of aerial robots. In particular, two scenarios were studied in the context of the MBZIRC 2017 challenge [63]. The first scenario focuses on autonomous localization, transportation, and delivery of objects by a team of UAVs. Furthermore, we investigate the task of autonomously landing a UAV on a moving vehicle, potentially ending an automated sensing mission. Before our achievements [2c], [14a], distributed object gathering UAVs has only been attempted in laboratory conditions, mostly concerning the automatic assembly of structures [118], [138], [139], [153]. The state-of-the-art work often focused on a subset of tasks needed for full autonomy, e.g., the autonomous grasping [100], [149] and motion planning [138]. Other sub-tasks such as onboard detections of the objects, UAV localization, or the full mission autonomy are often omitted. The MBZIRC 2017 challenge forced the research groups to leave ideal lab conditions, show a fully-integrated robotics system, and provide repeatable results. Only a handful of the 147 research groups who applied [60]–[62], [68] were able to complete the task of automatically grasping an object and delivering it to the desired location, with us winning the challenge [2c]. The second challenge of autonomous landing on a moving car was also successfully tackled by a handful of groups [66], [69], [95], [96], [98] with only the team of Beijing Institute of Technology team [66], and us [3c] completing the challenge. Before the competition, a plethora of research was published on the topic, as described be the review in [115]. However, as was the case with the first MBZIRC challenge, authors often relax on essential aspects of the problem and focus on its subproblems. Subproblems such as target detection and state estimation [105], [110], simulation [101], [116], [129], or use an external laboratory UAV localization system [112], [151], [159]. Those who attempted to solve the complete automatic landing with onboard detection, estimation, and control either conducted the maneuver at slow speed [102] or flying along a straight line [97].

Since the 2017 MBZIRC competition, the state of the art in the automatic grasping has further improved [48], [90]. Moreover, the subsequent MBZIRC 2020 challenge proposed a new problem with even higher requirements. In the 2020 challenge, a group of UAVs was tasked to automatically building a wall from bricks, again, using only onboard sensors and computation resources. The task can be seen as an extension of the 2017 object-gathering challenges with higher requirements on grasping precision, the object delivery, and overall system robustness. The 2020 challenge again caught the interest of approx. two hundred research teams from all over the world. Even though many tried [45], [54], only a few groups managed to automatically deliver even a single brick to the desired location [53], [54c], [7c]. Our solution [7c], [8c] again managed to win first place among many prestigious university teams in the world.

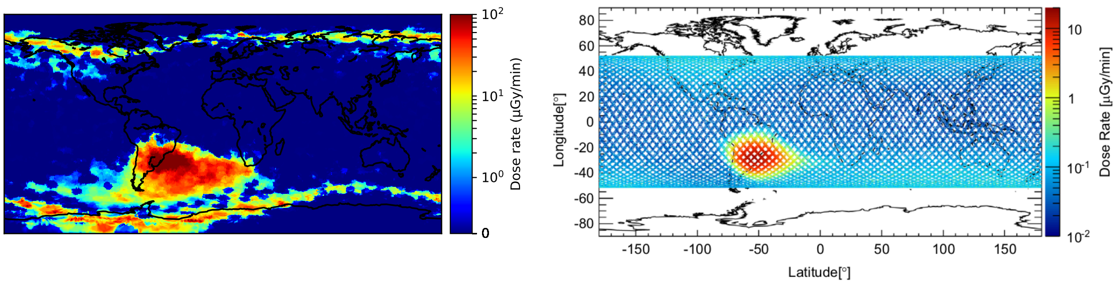
2.4 Measuring ionizing radiation, mapping and localization

This section focuses on interdisciplinary research of ionizing radiation dosimetry and localization of ionizing radiation sources onboard Unmanned Aerial Vehicles. First, we describe the author's first initial involvement and contributions in space radiation dosimetry and mapping. Then, we cross over to mobile aerial robotics and present the state of the art and contributions.

2.4.1 Outer space radiation dosimetry and mapping

Ionizing radiation is commonly measured for two reasons in the constrained environment of space applications. The first one is to assess the radiation dose deposited to crew members on board and the onboard space electronics. Traditional dosimeters on NASA's International Space Station (ISS) are being lately replaced with *smart* pixel detectors [75], [134], [167], namely, the Timepix sensor. The Timepix sensor [147], [172] is an hybrid Application-Specific Integrated Circuit (ASIC) Complementary Metal Oxide Semiconductor (CMOS) chip that can be bonded to a variety of semiconductor detection materials (Silicon, CdTe, CZT). Despite it being initially developed for medical imaging [80] and laboratory measurements, the Timepix has found applications even outside laboratory environments. Timepix pixel detectors are unique for their capability to measure traces of incoming ionizing particles with the detector. With those traces, machine learning algorithms can deduce the particle type, the energy of each particle [5c], [84]. The long-term and large-scale radiation dosimetry capability with Timepix detectors has also been tested in space outside of the ISS. Several satellites included the Timepix as their payloads: ESA's Proba-V [113], [141], British TechDemoSat-2 [83], and the Japanese RISESat [64].

The second use of radiation detectors in space is dedicated to capturing X-ray/Gamma-ray photons through a telescope's focusing device. X-ray observatories can not be deployed on the Earth's surface due to the presence of the atmosphere. The state-of-the-art observatories such as Chandra [178], Swift [176], and Fermi [169] typically use scintillating detectors to measure the incoming light, or CCD detectors for low-energy X-rays. CMOS detectors are rarely used in space applications, mainly due to undesired noise characteristics. However, Timepix detectors operate with an internal mechanism that filters out the dark current through the detector diodes, and therefore the resulting images are free to the traditional image noise.



(a) VZLUSAT-1

(b) ISS, courtesy of [134]

Figure 2.2: Ionizing radiation dose measured by the Timepix sensor onboard (a) VZLUSAT-1 (≈ 500 km Sun-synchronous orbit), and (b) the International Space Station (≈ 350 km orbit).

We add to the list an embedded and lite-weight design of a Timepix payload for the first Czech CubeSat satellite, the VZLUSAT-1 [37a], [38a]. The payload was dedicated to radiation mapping of Earth's LEO [4c] and capturing images using the onboard X-Ray telescope [39a]. The satellite has been operational for more than 3 years. New radiation data are being processed on a regular basis while being added to an open-source dataset[8] that is potentially useful for the community of future CubeSat designers. Further research led to a design of a one-time sub-orbital rocket experiment [42a], [40a]. A dedicated payload with two Timepix sensors were designed and developed based upon the Robot Operating System [15a]. Embedded hardware with Robot Operating System (ROS) automatically managed the recording of measured data in real time. This crossover of technologies later allowed continuation on the development of ROS-based Timepix technologies for the use onboard Unmanned Aerial Vehicles.

2.4.2 Localization and mapping of ionizing radiation sources by UAVs

In radiation sensing, unmanned robotic vehicles offer several advantages over conventional handheld detectors or piloted aircraft. These advantages can be exploited in a wide variety of applications.

Following the 2011 disaster at the Fukushima Daiichi Nuclear Power Plant (FDNPP), considerable amounts of radioactive material have been released into the plant area. Several Unmanned Ground Vehicles (UGVs) have been deployed directly inside the damaged reactor buildings of FDNPP under remote control. Various radiation detection methods have been tested inside the power plant, including a coded aperture scintillator [165], a semiconductor digital dosimeter [155], a Compton event camera composed of two scintillators [76], and a time-of-flight gamma camera [142]. Ground-based robots offer higher payload capacity and the ability to carry heavier sensory equipment than most aerial vehicles. On the other hand, these robots tend to be relatively bulky, struggle to navigate the cluttered corridors and staircases inside the damaged buildings, and generally move slower than a multirotor aircraft.

UAVs have been utilized to map the spread of the radioactive material outside the power plant. These range from large aircraft weighing more than 90 kg equipped with heavy scintillation detectors [114], [133], [161] to compact multirotors suitable for flying along a pre-defined trajectory close to the ground [99], [117], [145]. Outside of Japan, several projects have employed UAVs for radiation intensity mapping around uranium ore mines [85], [91], [130].

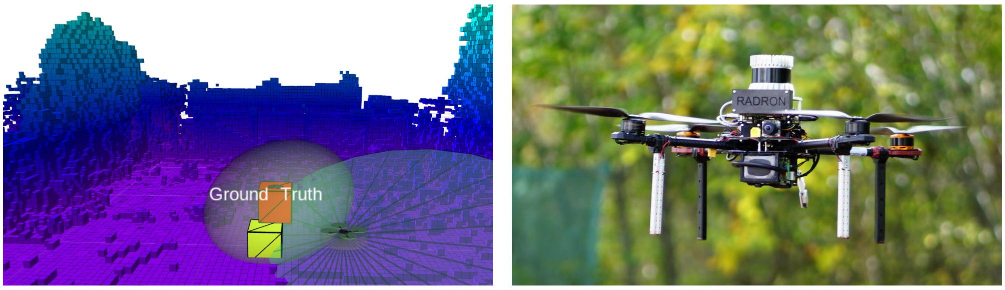
In [150], multiple fixed-wing UAVs equipped with miniature scintillators are used for contour analysis of an irradiated area. Trajectory planning and data processing are performed offline, contrary to our approach, which estimates the source's position in real time during the flight. In [119], the contour analysis is tackled using a single multirotor UAV. The contour analysis uses a Gaussian mixture model to estimate multiple radiation sources' positions with overlapping intensity fields. The projects mentioned above utilize unmanned vehicles to deliver a radiation sensor into a hazardous environment. However, the approaches do not respond to measured data in real time and thus do not exploit the mobility of UAVs to improve the measurement.

Active path-planning driven by the onboard measurements has been shown in [161] for an outdoor environment and in [88] for a GPS-denied indoor environment. Both works rely

[8] `https://github.com/vzlusat`

on a scintillator sensor to estimate the radiation intensity in the UAV's current position. As a result, the employed aerial platforms have to be large with a payload capacity exceeding 2 kg. As in [161], the aerial platform is a 90 kg unmanned aerial helicopter, which significantly limits its deployment conditions due to personal safety and considerable minimum distance to obstacles in the environment.

The lack of lightweight radiation detectors with immediate readout capability severely limits the application potential of aerial dosimetry. However, the Timepix pixel detectors are ideal for the use onboard micro UAVs thanks to their low weight, small size, and the absence of any active cooling mechanism. We propose a UAV system for outdoor and indoor environments while utilizing the know-how obtained with the embedded space applications' work. The ROS API for Timepix [15a], initially developed for a suborbital rocket experiment [42a], allows the technology to transfer to the robotics field. However, a new robotic methodology for motion planning and exploration needed to be developed to accommodate and utilize the proposed measurement system's specifics [1c]. Moreover, the Compton camera mechanism [93], is being utilized to provide the smallest real-time single-sensor Compton camera ever used on a UAVs [34a].



(a) Visualization of the localization

(b) UAV equipped with the Timepix3 sensor

Figure 2.3: Showcase from a radiation localization experiment with the MiniPIX Timepix3 Compton camera onboard a UAV. The visualization shows the ground truth position of the radiation source (yellow) and the estimated position (red).

Chapter 3

Research-focused UAV Platform

Our initial work on relatively-localized UAV formations [43a], [44a] led us to create a custom platform for experimental evaluation. This control system [24a] used an embedded Model Predictive Control (MPC) controller executed in real time onboard in custom hardware. This embedded system, in conjunction with the novel relative localization system [143], allowed us to conduct our first multi-UAV research that was backed up by real-world experiments [16a], [17a], [22a], [23a].

As the demands for the capabilities of the platform increased, we abandoned the embedded platform and adopted a modular approach of the Robot Operating System. The first core publication in this thesis, presented at the IEEE IROS 2018 conference, proposes a model predictive tracking mechanism for fast generation of feasible control references for UAVs.

> [6c] **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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8

Obtaining a feasible control reference that satisfies state constraints of the UAV is an essential requirement for precise feedback control. Moreover, the presented approach utilizes the 8 s MPC prediction horizon for a mutual UAV collision avoidance, which is especially useful for the verification of new multi-UAV methods. The presented control technique was partially responsible for our success in the MBZIRC 2017 robotic competition [2c], [3a], [14a].

The *MPC Tracker* [6c] is, however, just a small part of complex system for multirotor UAV experimental evaluation and deployment. With the second core publication in this thesis, we propose a modular control system for UAV state estimation, trajectory tracking, feedback control, and flight management. This open-source control system sets up novel paradigms, such as the multi-frame UAV state estimation, and the heading-based control pipeline, devoid of Euler and Tait-Bryan angles. Testing new feedback controllers, reference generators, state estimators, and other parts of the control pipeline are possible thanks to the modular design and the extensive simulation environment. The *MRS UAV System* was submitted to the Journal of Intelligent & Robotic Systems and has been made available as open source[1].

> [8c] **T. Baca**, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *submitted to JINT, after the 1st revision*, Aug. 2020. eprint: `arXiv:2008.08050`

[1] `https://github.com/ctu-mrs/mrs_uav_system`

2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles

Tomas Baca[1], Daniel Hert[1], Giuseppe Loianno[2], Martin Saska[1], and Vijay Kumar[3]

***Abstract*— We propose a novel approach for optimal trajectory tracking for unmanned aerial vehicles (UAV), using a linear model predictive controller (MPC) in combination with non-linear state feedback. The solution relies on fast onboard simulation of the translational dynamics of the UAV, which is guided by a linear MPC. By sampling the states of the virtual UAV, we create a control command for fast non-linear feedback, which is capable of performing agile maneuvers with high precision. In addition, the proposed pipeline provides an interface for a decentralized collision avoidance system for multi-UAV scenarios. Our solution makes use of the long prediction horizon of the linear MPC and allows safe outdoors execution of multi-UAV experiments without the need for in-advance collision-free planning. The practicality of the tracking mechanism is shown in combination with priority-based collision resolution strategy, which performs sufficiently in experiments with up to 5 UAVs. We present a statistical and experimental evaluation of the platform in both simulation and real-world examples, demonstrating the usability of the approach.**

I. INTRODUCTION

Robotic aerial systems have been extensively studied by the scientific community over the past ten years. Nowadays, the capabilities of a single unmanned aerial vehicle (UAV) extend from fully autonomous indoor and outdoor operation [1], [2], via onboard sensor processing and motion planning [3], to environment scanning and mapping [4]. Coordinated flight of multiple UAVs is currently under investigation by many research groups [5], [6]. Autonomous flight with a group of UAVs promises of faster task execution and even lower requirements for onboard sensors. A benefit of the added redundancy is that the task can continue to be executed even after the failure of one of the units. Numerous applications cannot be realized using a single UAV [7], [8], [9], and cooperative flight with small mutual distances is required. This even increases the demands on the precision and reliability of the UAV control mechanism (see figure 1 for numerous examples of the deployment of the proposed system in multi-UAV scenarios).

Research and development of autonomous multi-UAV systems involves a high risk of a collision between the vehicles. The mechanism for preventing collisions is often built into a mission control system, which takes care of safe trajectory planning in advance. However, high-level mission control systems that are a subject of research and development should be considered unreliable during the first stages of their development. An independent underlying system for preventing collisions in real time should therefore be in place during real-world testing and experiments. According to the "minimalist modular software development" philosophy[1], it is safer to develop a single transparent collision-preventing mechanism, which can be used by a high-level layer regardless of its purpose or way of operation.

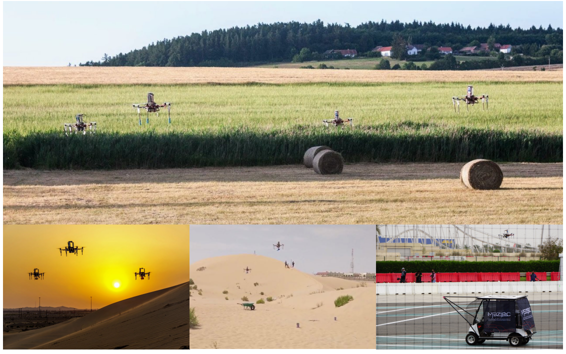


Fig. 1: The versatility of the proposed system is shown in various scenarios including multi-robotic remote sensing, compact formation flying, collaborative collecting of objects in a challenging outdoor environment, and autonomous landing on a moving car.

GNSS naturally localizes all vehicles within the same frame of reference, so the position information can be directly used for collaboration among UAVs. Global navigation satellite systems (GNSS) such as GPS are standard for outdoor use, and are the foremost source of position information for UAVs when flying autonomously. To successfully deploy multiple collaborating vehicles, we assume that they are localized within the same world coordinate system. Nevertheless, the proposed system can be used with any positioning system (even onboard) that allows synchronization of the reference frames of individual vehicles within the team, e.g., [10], [11].

A. State of the art

Several concepts have been proposed for avoiding mutual collisions when the trajectories of the vehicles are known in advance. Mutual collision avoidance between robots was historically studied by [12] within the context of a non-holonomic ground robotic system. The solution presented here relies on control law with explicitly incorporated control

[1]Authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague 6, {tomas.baca,hertdani,martin.saska}@fel.cvut.cz.

[2]The author is with the Tandon School of Engineering, New York University New York, 6 MetroTech Center, 11201 Brooklyn NY, USA, {kumar}@seas.upenn.edu.

[3]The author is with the GRASP lab, University of Pennsylvania, USA, {loiannog}@nyu.edu.

[1] Unix philosophy, Douglas McIlroy, 1994

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

action to avoid other robots in the team, when they appear in a proximity region. This type of technique provides a reliable solution for slow-moving robots; however, for fast UAVs, the proximity region would have to be larger than a typical problem in collaborative multi-UAV would allow.

In [13], the authors propose a mechanism which considers knowing the future trajectories of the robots in a team. The strategy avoids collisions by transforming the time parameter of all the UAV trajectories. By planning in so-called coordination time-space, the robots find a collision-free passage that can be converted to a warp of the sampling rate of the original trajectories. However, this method omits the dynamical constraints of the vehicles and might result in an unfeasible reference that would require additional processing. Moreover, in the case when one of the vehicles is static, simply resampling the original trajectories in the time domain requires additional mechanics, e.g., similar to the mechanics proposed in this manuscript.

A similar approach to [13], which also uses the configuration space, is described in [14]. However, the application involves ground robots with a centralized mechanism for planning and for distributing trajectories to the robots. This kind of solution is not applicable for decentralized reaction-based avoidance when each vehicle is controlled by onboard (decentralized) autonomy to fulfill a particular task, which is typically based on local sensor data. As in [13], this mechanism also cannot react in situations when one of the UAVs is stationary.

Mutual collision avoidance for aerial vehicles is tackled in [3]. A centralized formation planning approach is proposed, which utilizes a cylindrical proximity condition and a potential field repulsion-based avoidance strategy with an evasive vertical maneuver. However, the approach has been validated only in simulations, while real-world experiments are described by the authors as a challenge for the future.

A decentralized, non-linear model predictive control approach for the control of multiple UAVs is presented in [15]. The authors present trajectory tracking system, which reacts to potential future collisions by modifying the user reference. Both simulation and laboratory experiments show the real applicability of the solution. However, the solution uses only a 2 s prediction horizon, which limits the maximum speed and acceleration of the UAVs at which it can intervene. The non-linear MPC problem, in which the collision avoidance is directly integrated in the control problem, introduces several computational challenges compared to the linear MPC utilized in this paper. Moreover, the described optimization problem scales poorly with increasing number of UAVs in the group due to having avoidance constraints for each vehicle, which is not the case with the MPC formulation utilized here. It therefore allows only slow flight speeds (approx. $10.0\ \mathrm{km/h}$ at $2\ \mathrm{m/s^2}$ acceleration) in order to avoid potential collisions. In comparison, the system presented here has been tested with a prediction horizon of 8 s, which allows flight at a speed of over $30\ \mathrm{km/h}$, thanks to the long prediction horizon, as presented in the Experiments section.

State-of-the-art contributions to the avoidance of mutual collisions between fast unmanned aerial vehicles in real-time are scarce. Moreover, a truly decentralized and transparent system, which could be deployed in real-world outdoor experiments to serve as a collision safety mechanism, is virtually non-existent for conventional UAV speeds above $10\ \mathrm{km/h}$. We are therefore aiming beyond the state-of-the-art, while using current technologies, such as linear model predictive control and non-linear control [16].

B. Contributions

In this paper, we address a practical approach for optimal trajectory tracking allowing distributed collision avoidance for multiple unmanned aerial vehicles sharing the same workspace. The proposed mechanism can be implemented in the form of a transparent layer for a UAV control pipeline, and can be used by an application layer as a trajectory tracker. By solving linear model predictive control (MPC) for steering a simulated UAV in real-time onboard the vehicle, we manage to bring the otherwise impractically difficult task with very large MPC problem down to MPC with only the length of the prediction horizon that is necessary to enable evasive maneuvers. Moreover, by sampling the states of the simulated UAV, we create a reference for non-linear state feedback, which is capable of agile maneuvers and disturbance rejection, both of which are outside the operating point of linear MPC. If a sufficient reserve in dynamics is left to the non-linear state feedback, the combination exhibits the properties of non-linear predictive controller with a large enough prediction horizon to allow possible collisions between multiple vehicles flying within the same workspace to be detected and avoided. The proposed MPC trajectory tracker provides optimal tracking of arbitrarily long and potentially unfeasible user trajectories, which might not be designed given the actual dynamical constraints of a particular UAV. The proposed system therefore enables a substantially simpler design and safe testing of high-level planning systems. It should be noted that a simple command to fly to certain coordinates or to fly a trajectory containing step functions is a typical and unfeasible high-level command. Thanks to the predictive nature of the tracker, the vehicles conduct adequate feedforward maneuvers. Our method is suitable for real-world experiments, where novel multi-robotic algorithms can be safely tested without risking a mutual collision, either due to a program malfunction or due to a user error (see [7], [2], [8] for a list of multi-robot scenarios tested with the proposed system). The implementation favors both computational and communication resources, so it is applicable onboard in non-laboratory conditions, as presented towards the end of this paper.

C. Outline

This paper is structured as follows. First, in section II, we introduce a collision detection and avoidance mechanism, which operates with known future trajectories of the vehicles in the group. Section III describes the structure of the proposed control architecture. The non-linear state feedback (section III-B), together with linear model predictive control (section III-C), form the components of a novel model predictive tracker, which is further introduced in section (III-D). Section IV presents an experimental evaluation of the system in real-world scenarios.

II. Collision Avoidance

Deploying multiple UAVs in the same area introduces potential collisions, not only between the vehicles and the environment but also between the vehicles themselves. In this section, we focus on mutual collisions between UAVs in an open outdoor environment. There are two approaches to solve the avoidance problem, as we have discussed in the introduction. One is to incorporate collision-free path planning into a high-level mission planner. Mission planners may be designed to conduct remote sensing, environment exploration or more complex tasks, such as collaborative object localization and delivery. However, our experience in the field suggests that is is impractical to include the avoidance strategy, since it gives the responsibility for safety to a high-level system, which is often the subject of development and testing. It is reasonable to assume that if the mission planner is unsound and requires an avoidance mechanism to intervene, then the built-in avoidance may also not be reliable. Moreover, collision avoidance might also be required when switching between multiple high-level planners, or in situations when the UAVs are directly under control by field operators. Our approach therefore incorporates the avoidance mechanism as a transparent layer in the control pipeline (see figure 3).

In situations where the future trajectories of the vehicles are known in advance, techniques such as rubber banding [14] or time resampling [13] can be used to alter the trajectories before they are executed. We will focus on the specific situation of outdoor flight, where increasing the height of the trajectory to avoid mutual collision is a viable and straightforward option. In this section, we assume that the future trajectories are known and later, in section III-D, we explain how to obtain them in real-time during flight with properties such as the length of the planning horizon and the update rate, which are required by the collision avoidance technique presented in this section.

A. Shared information

The proposed mechanism relies on mutual communication between the UAVs and on the exchange of their future trajectories. Assuming that the trajectories resemble the actual future movement of the vehicle, potential collisions are detected by each vehicle individually in a decentralized way. Along with the predictions, each vehicle broadcasts a binary flag, which is raised when the vehicle collision avoidance mechanism is active. The data are shared at 2 Hz, which was chosen on the basis of the safety margins and the dynamics of the vehicles used in the experimental parts of this paper. During the experiments, communication between UAVs operates without confirmation and repetitions, which minimizes the communication overhead.

Algorithm 1 The avoidance algorithm reshapes the reference trajectory $\mathbf{r}_{\mathrm{D}}$ based on potential collisions found in the predictions made by the MPC tracker.

```
procedure RESHAPE
    Input:
        $\mathbf{r}_D$                                  ▷ the original reference trajectory
        $p_{\mathrm{current}}$                          ▷ the priority of this UAV
        $\Delta\mathrm{z}$                              ▷ height offset for avoiding a collision
    Output:
        $z_{\mathrm{offset}} \leftarrow 0$              ▷ accumulation of the height offset
    while find_collision () do
        limit_velocities()                              ▷ set velocity constraints to $\frac{1}{2}$ of the nominial
        if $p_{\mathrm{collision}} > p_{\mathrm{current}}$ then    ▷ this UAV has lower priority
            $z_{\mathrm{offset}} \leftarrow z_{\mathrm{offset}} + \Delta\mathrm{z}$
    return $\mathbf{r}_{\mathrm{new}}$
procedure FIND_COLLISION
    Input:
        $\mathbf{x} \leftarrow$ prediction for this UAV, offset by $\mathbf{z}_{\mathrm{offset}}$
        $\mathbf{x}_{\mathrm{n}} \leftarrow$ other UAV predictions
        $\mathrm{m} \leftarrow$ length of the prediction
    Output:
        $t_{\mathrm{collision}} \leftarrow$ -1          ▷ the time step when the collision would occur
        $p_{\mathrm{collision}}$                        ▷ priority of the other UAV we collided with
    for $\underline{\mathbf{x}}$ in $\mathbf{x}_n$ do   ▷ for all predictions of other UAVs
        for $i$ from $1$ to $m$ do
            if check_proximity($\mathbf{x}_{[\mathrm{i}]}, \underline{\mathbf{x}}_{[\mathbf{i}]}$) then    ▷ Eq. (1)
                $t_{\mathrm{collision}} \leftarrow \mathrm{i}$
                $p_{\mathrm{collision}} \leftarrow$ priority of $\underline{\mathbf{x}}$
                return True                             ▷ collision found
    return False                                        ▷ collision not found
```

B. Conflict resolution strategy

Given the shared information about the future trajectories of the UAV, an avoidance strategy can be used to alter the movement of the UAVs in the group to avoid collisions. Our aim is not to present a novel conflict resolution strategy, but rather to show that the proposed pipeline performs sufficiently, even with the following trivial method. The proposed framework is easily extensible and readers aiming to replicate the system can freely introduce a different resolution strategy (e.g. [17], [18]) with properties appropriate to their particular application. The showcased method, which is described in detail in algorithm 1, relies on iteratively modifying the flight height of the UAVs according to a hierarchical priority system. A unique number is given to each aircraft representing its priority in the group (a total ordering). A collision is detected by a UAV when another UAV violates the proximity condition at any point along the prediction horizon. Each UAV performs an avoidance maneuver when a potential collision with another UAV with higher priority is detected.

1) Proximity condition: Collisions between any two UAVs in the future are found by checking the proximity in space-time within their future trajectories. Since the future trajectories are considered as lists of points sampled at a known rate, we can find the violations in proximity by evaluating the space conditions between all corresponding points in the future trajectories. Let us consider a cylindrical area with radius R_p and height H_p around each point $\mathbf{x}^a[i]$ in 3D space in the future trajectory of UAV a. A collision is detected with UAV b, when a corresponding point $\mathbf{x}^b[i]$ lies in the cylindrical area. Formally, let $\mathbf{x}^n \in \mathbb{R}^{m\times 3}$ be predicted trajectory of the n_{th} UAV containing m positions in 3D space. Then a potential collision between two UAVs,

a and b, is detected when the following conditions are met

$$\begin{aligned} \left|\left|\mathbf{x}^a_{1:2}\left[i\right] - \mathbf{x}^b_{1:2}\left[i\right]\right|\right|^2_2 \leq R_p, \forall i \in 1, \dots, m \\ \left|\left|\mathbf{x}^a_{3}\left[i\right] - \mathbf{x}^b_{3}\left[i\right]\right|\right|^2_2 \leq \frac{H_p}{2}, \forall i \in 1, \dots, m \end{aligned} \tag{1}$$

where $\mathbf{x}^a_{1:2}\left[i\right]$ represents the x and y coordinates of the point, and $\mathbf{x}^a_{3}\left[i\right]$ represents the z coordinate of the point. Parameters R_p, H_p and dynamical constraints on the motion of the vehicle introduced later in sestion III-C determine safety of the avoidance system. Particular values showed in table I were obtained using simulations of the system (see section IV).

2) Avoidance maneuver: The maneuver consists of altering the reference trajectory of the vehicle by adding an offset in height. Moreover, by lifting the acceleration constraints in the vehicle's tracker, we allow a rate faster than the nominal rate of change of height, which helps to avoid ascending UAVs, whose constraints are active. Vehicles with higher priority, which are being avoided, do not alter the reference trajectory. However, any vehicle that detects a potential collision tightens its velocity constraints to mitigate potentially aggressive maneuvers when flying near other vehicles, as described in algorithm 1.

III. CONTROL PIPELINE

The architecture of the proposed control system, which also provides a sufficiently long prediction horizon and update rate for the collision avoidance mechanism follows a common multi-layer structure. The schematic in figure 3 depicts the components to which we will refer in this chapter. The UAV attitude control loop is closed by an embedded stabilizer, which receives a command in the form of the desired attitude $\mathbf{R}_D$ and thrust T_D. The surrounding layer is a non-linear $SO(3)$ state feedback controller, which follows a control command in the desired position $\mathbf{r}_D$, speed $\dot{\mathbf{r}}_D$, and acceleration $\ddot{\mathbf{r}}_D$, expressed in the world coordinate frame.

A. Embedded attitude control

An attitude controller is an onboard unit responsible for maintaining the desired attitude $\mathbf{R}\left(\phi, \theta, \psi\right) \in SO(3)$, where ϕ, θ, ψ are the Euler angles corresponding to the yaw, pitch and roll motions of the UAV. The controller block outputs the desired motor speeds and accepts the desired orientation $\mathbf{R}_D$ and total thrust T_D. The UAVs used here are equipped with the PX4 stack [19] on a PixHawk flight controller. However, the proposed system is not dependent on the choice of a particular attitude controller.

B. Non-linear SO(3) controller

The next block in the pipeline is the non-linear $SO(3)$ state feedback controller, which uses the model

$$\begin{aligned} \dot{\mathbf{r}} = \mathbf{v}, \qquad m\dot{\mathbf{v}} = f\mathbf{R}\mathbf{e}_z + mg\mathbf{e}_z, \\ \dot{\mathbf{R}} = \mathbf{R}\hat{\mathbf{\Omega}}, \quad \mathbf{J}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} = \mathbf{M}, \end{aligned} \tag{2}$$

where $\mathbf{r} = \left[x, y, z\right]^T$ is the position, $\mathbf{R}\left(\phi, \theta, \psi\right)$ is the orientation of the UAV in the world coordinate frame with the basis $\{\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_3\}$, $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame and the *hat map* $\hat{\cdot} : \mathbb{R}^3 \rightarrow SO(3)$ is defined by the condition $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$. Two forces act on the vehicle. The magnitude of the gravitational pull is denoted by $g \in \mathbb{R}$, whereas the magnitude of the total thrust force created by the propellers is denoted by $f \in \mathbb{R}$. The UAV mass is denoted by $m \in \mathbb{R}$. The angular velocity of the UAV in the body frame $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ is denoted by $\mathbf{\Omega} \in \mathbb{R}^3$ and $\mathbf{J} \in \mathbb{R}^{3\times3}$, the inertia matrix. The total moment exerted by the propellers onto the UAV is $\mathbf{M} = \left[M_1, M_2, M_3\right]^T$.

The control is built on the work presented in [16], [20], with the control inputs $f \in \mathbb{R}$ and $\mathbf{M} \in \mathbb{R}^3$ chosen as

$$\begin{aligned} \mathbf{M} = &- k_R\mathbf{e}_R - k_\Omega\mathbf{e}_\Omega + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} - \dots \\ &\dots - \mathbf{J}\left(\hat{\mathbf{\Omega}}\mathbf{R}^T\mathbf{R}_c\mathbf{\Omega}_c - \mathbf{R}^T\mathbf{R}_c\dot{\mathbf{\Omega}}_c\right) \end{aligned} \tag{3}$$

$$\begin{aligned} f = &- (-k_x\mathbf{e}_r - k_{ib}\mathbf{R}\int\limits_0^t R(\tau)^T\mathbf{e}_r d\tau - \dots \\ &\dots - k_{iw}\int\limits_0^t \mathbf{e}_r d\tau - k_v\mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d) \cdot \mathbf{R}\mathbf{e}_3, \end{aligned} \tag{4}$$

where $\ddot{\mathbf{x}}_d$ is the desired acceleration. The control errors in rotation, position and velocity are denoted by $\mathbf{e}_R$, $\mathbf{e}_r$ and $\mathbf{e}_v$, respectively and $\mathbf{e}_\Omega = \mathbf{R}^T\mathbf{R}_d\mathbf{\Omega}_d$. The resulting control action consists of f and $\mathbf{R}_C$, which is the orientation command. A more detailed description of the original controller extended with additional integral gains can be found in our recent publication [2].

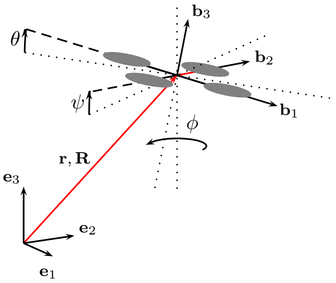


Fig. 2: The world coordinate frame w in which the position and orientation of the UAV is expressed by translation $\mathbf{r} = \left[x, y, z\right]^T$ and rotation $\mathbf{R}\left(\phi, \theta, \psi\right)$.

C. Linear model predictive control

Model Predictive Control is increasingly popular for controlling the fast dynamics of UAVs, and it is especially appealing for the collision avoidance scheme proposed here. Our previous work [21] featured a real-time embedded MPC on a micro aerial vehicle with a 2.2 s prediction horizon. In contrast with the previous work, the MPC here is not directly involved to control of the UAV, as it would traditionally be. It is part of the proposed trajectory tracker, which serves as a reference generator for an underlying state feedback. A linear MPC uses an LTI model with n states and k inputs, defined as

$$\begin{aligned} \mathbf{x}_{[t+1]} = \mathbf{A}\mathbf{x}_{[t]} + \mathbf{B}\mathbf{u}_{[t]} \\ \mathbf{y}_{[t]} = \mathbf{C}\mathbf{x}_{[t]} + \mathbf{D}\mathbf{u}_{[t]}, \end{aligned} \tag{5}$$

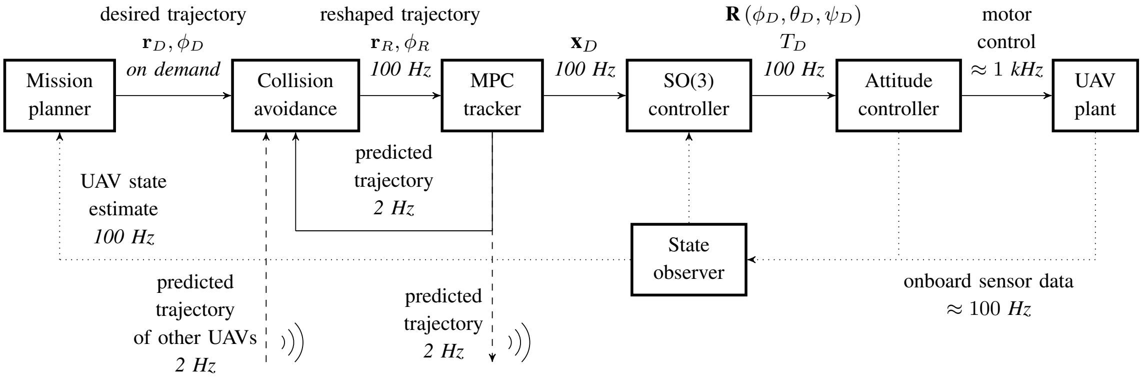


Fig. 3: Diagram of the control pipeline, including the proposed MPC tracker and the collision avoidance mechanism. The original reference trajectory $\mathbf{r}_D, \phi_D$ is reshaped by the collision avoidance mechanism to $\mathbf{r}_R, \phi_R$. The reshaped reference serves as a setpoint for the MPC in the MPC tracker, which outputs a command $\mathbf{x}_D, \dot{\mathbf{x}}_D, \ddot{\mathbf{x}}_D$, $\phi_D, \dot{\phi}_D, \ddot{\phi}_D$ for the non-linear $SO(3)$ controller. The non-linear controller produces the orientation and thrust reference for the embedded attitude controller.

where $\mathbf{x} \in \mathbb{R}^n$ denotes the state vector and $\mathbf{u} \in \mathbb{R}^k$ denotes the input vector. Matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$ are the main system matrix and the input matrix, respectively. In contrast with the traditional full description of the LTI system, we assume $\mathbf{C} = \mathbf{I}$, and $\mathbf{D} = \mathbf{0}$. Therefore $\mathbf{y}_{[t]} = \mathbf{x}_{[t]}$ holds.

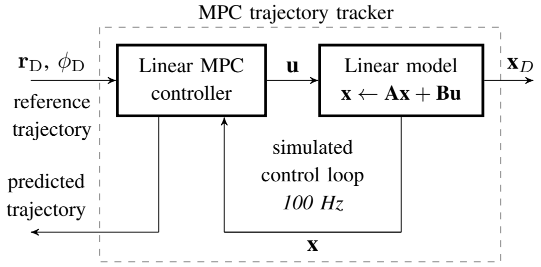


Fig. 4: Diagram of the MPC tracker loop. The reference trajectory $\mathbf{r}_\mathrm{D} \in \mathbb{R}^{3 \times N}$ of length N is supplied to the MPC controller. The UAV dynamics is simulated in a closed loop with MPC at 100 Hz. The state $\mathbf{x} \in \mathbb{R}^9$ of the simulated system serves as the command for the $SO(3)$ controller.

The definition of MPC relies on the notion of a control error over a future prediction horizon. The control error is defined as $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the trajectory reference. To find the control signals by MPC, it is necessary to solve the optimization problem

$$\min_{\mathbf{u}_{[t]},\mathbf{x}_{[t]}} \quad \mathrm{V}(\mathbf{x},\mathbf{u}) = \frac{1}{2}\sum_{i=1}^{m-1}\left(\mathbf{e}_{[i]}^T\mathbf{Q}\mathbf{e}_{[i]} + \mathbf{u}_{[i]}^T\mathbf{P}\mathbf{u}_{[i]}\right) \tag{6}$$

$$\text{s.t. } \mathbf{x}_{[t+1]} = \mathbf{A}\mathbf{x}_{[t]} + \mathbf{B}\mathbf{u}_{[t]}, \quad \forall t \in \{0, \dots, m-1\} \tag{7}$$

$$\mathbf{x}_{[t]} \leq \mathbf{x\_max}_{[t]}, \quad \forall t \in \{1, \dots, m\} \tag{8}$$

$$\mathbf{x}_{[t]} \geq \mathbf{x\_min}_{[t]}, \quad \forall t \in \{1, \dots, m\} \tag{9}$$

repeatedly, where the quadratic cost function in (6) penalizes the control error and the input action over a horizon $m \in \mathbb{Z}^+$ in length. Penalization matrices $\mathbf{Q}$ and $\mathbf{P}$ are positive semi-definite. Constraint (7) forces the states to follow model (5), while constraints (8) and (9) bound the states to a box to limit the maximum acceleration and velocity.

D. MPC Trajectory tracker

Figure 4 presents a diagram of the MPC tracker shown as a single block in the pipeline in figure 3. A trajectory tracker supplies a control command $\mathbf{x}$ to the underlying state feedback. The command is required to be updated at ≈ 100 Hz. It should also be smooth and feasible concerning the translational dynamics of the UAV. A state space representation of the translational dynamics in the world frame can be captured by state vector $\mathbf{x}$ as

$$\begin{aligned}
&\mathbf{x}[1] = \mathbf{r}[1], \quad \mathbf{x}[4] = \mathbf{r}[2], \quad \mathbf{x}[7] = \mathbf{r}[3], \quad \mathbf{x}[10] = \phi,\\
&\mathbf{x}[2] = \dot{\mathbf{r}}[1], \quad \mathbf{x}[5] = \dot{\mathbf{r}}[2], \quad \mathbf{x}[8] = \dot{\mathbf{r}}[3], \quad \mathbf{x}[11] = \dot{\phi},\\
&\mathbf{x}[3] = \ddot{\mathbf{r}}[1], \quad \mathbf{x}[6] = \ddot{\mathbf{r}}[2], \quad \mathbf{x}[9] = \ddot{\mathbf{r}}[3], \quad \mathbf{x}[12] = \ddot{\phi}.
\end{aligned} \tag{10}$$

The 4 independently controllable states are then expressed as a differentially flat system with matrices $\mathbf{A}$ and $\mathbf{B}$ defined as

$$\begin{aligned}
&\mathbf{A}(\Delta t, a_r, a_\phi) = \\
&= \begin{bmatrix} \mathbf{A}_s(\Delta t, a_r) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_s(\Delta t, a_r) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_s(\Delta t, a_r) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_s(\Delta t, a_\phi) \end{bmatrix}
\end{aligned} \tag{11}$$

$$\mathbf{B}(\Delta t, b_r, b_\phi) = \begin{bmatrix} \mathbf{B}_s(\Delta t, b_r) \\ \mathbf{B}_s(\Delta t, b_r) \\ \mathbf{B}_s(\Delta t, b_r) \\ \mathbf{B}_s(\Delta t, b_\phi) \end{bmatrix}, \tag{12}$$

where sub-system matrices $\mathbf{A}_s$ and $\mathbf{B}_s$ are defined as

$$\mathbf{A_s}(\Delta t, a) = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & a \end{bmatrix}, \ \mathbf{B_s}(\Delta t, b) = \begin{bmatrix} \frac{b\Delta t^2}{2} \\ b\Delta t \\ b \end{bmatrix}, \tag{13}$$

with Δt being the sampling step, and a_r, a_ϕ, b_r, b_ϕ are constants for the transfer function between the input and the acceleration states in the system. For the transfer to be a 1st order system, $a_r + b_r = 1$ must hold for all the a constants and b constants.

The control command for $SO(3)$ could theoretically be found solely with MPC. However, due to the potentially long reference trajectory and the required density of sampled states, that option is not feasible, because the number of independent parameters of the Hessian of the quadratic program exhibits approx. quadric growth in the length of the prediction horizon. For example, with the reference sampled at 100 Hz, a 60 s user trajectory would require the solution of MPC with 6000 steps of the prediction horizon. However, this is unrealistic, given the current technology and the onboard computational power of today's UAVs. On the other hand, the proposed solution requires to repeatedly solve MPC with only 40 sparsely distributed steps thanks to not being constrained by the actual dynamics (5) of the UAV. The number of steps is a design parameter, setting of which depends on the speed and agility of the flight. Given this can be solved at 100 Hz in real time onboard the vehicle, it supports onboard autonomy, which may be changing the reference trajectory on demand.

We tackle the complexity issue by having a shorter prediction horizon, which does not cover the full length of the reference, however in control of a real-time simulation of the transitional dynamics, which is evaluated at 100 Hz in real-time onboard the vehicle. The MPC is purposefully tuned to guarantee safe and sound tracking on the known simulated model. States of the simulated model are then sampled at 100 Hz, and are given as a control command to the $SO(3)$ controller, which mirrors the states of the real vehicle according to the state of the simulated model. This concept utilizes the best of each of the component. Thanks to the predictive nature of MPC, the resulting control command can proactively track even an unfeasible user trajectory, which violates velocity and acceleration constraints. On the other hand, the non-linear controller provides tilt and thrust commands to conduct aggressive maneuvers that would be outside the operating point for linear MPC. However, the dynamical constraints of MPC are set to leave a reserve for SO(3) to handle disturbances. As a result, by having a known prediction of the movement of the vehicle, we can efficiently detect potential collisions and avoid them before they happen. The predictions are transmitted to the other UAVs in the group. Additionally, a connection is established between the collision avoidance block and the MPC tracker block to allow the collision avoidance algorithm modify the MPC velocity and acceleration limits when collision is detected.

IV. EVALUATION

Multimedia material related to this work is available on the website `http://mrs.felk.cvut.cz/iros2018mpc`.

A. Platform-specific implementation

The experimental platform is built around the DJI F550 frame, which is equipped with in Intel NUC onboard computer (5th generation Core-i7, 8 GB of RAM). The Ubuntu 16.04 operating system houses all software except the attitude controller in PixHawk. PixHawk is an off-the-shelf and open-source control board, which fuses onboard inertial sensors with GPS and provides an estimate of the states of the UAV to the onboard computer. In addition to the sensors integrated with PixHawk, our basic UAV setup utilizes (optionally, according to the application) a precise RTK GPS and a laser rangefinder to measure the height of the UAV. Both of these are fused with the PixHawk's state estimate. Mutual communication between the vehicles is covered by a standard Wi-Fi connection. Custom software is built with the Robot Operating System (ROS), which provides a painless transition from simulation to the real hardware. The Gazebo robotic simulator is used in conjunction with PixHawk's firmware, which provides a life-like simulation of the UAV dynamics and the low-level control. The current implementation of the MPC tracker relies on the CVXGEN [22] to solve the quadratic program (6–8). A single iteration of the MPC tracker takes between 2 to 10 ms using only a single core of the Core-i7 processor, with a maximum of 30 iterations of the CVXGEN solver. The MPC tracker and collision avoidance have been used for more than a year with the following parameters (table I, which may vary according to the application demands:

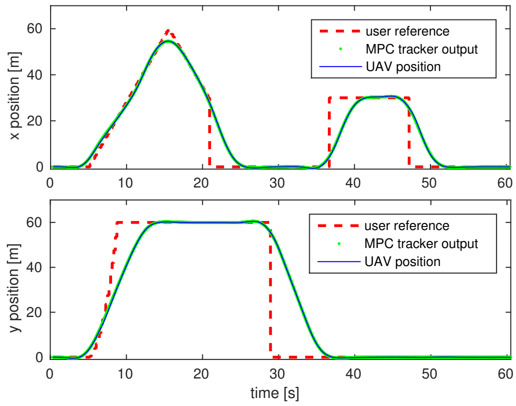


Fig. 5: Graphs of the x and y coordinates of MPC tracking a user reference. The same tracking with predictive capabilities is present in the remaining z and yaw axes.

TABLE I: Parameters for the DJI F550 hexacopter for outdoors.

MPC rate	100 Hz
Computational time	2 to 10 ms
prediction horizon	8.0 s (40 steps)
max. speed	8.3 m/s
max. acceleration	2.5 m/s^2
cylinder radius R_p	5 m
cylinder height H_p	5.99 m
height correction Δz	3 m

B. Statistical evaluation in simulation

Simulations conducted in the Gazebo simulator reflect the real-world platform in many important aspects. All software components running in the simulation are identical to the set used on the real hardware, including the PixHawk's embedded firmware. We modeled the F550 frame to match the physical properties of the vehicle. To test the avoidance strategy, we first established the baseline without the avoidance. The test scenario consists of 5 UAVs flying in the

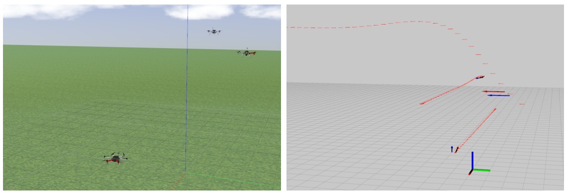


Fig. 6: These two images from the Gazebo simulator show the avoidance maneuver for three UAVs which would collide above the coordinate center without the collision avoidance system. The red paths mark the predicted trajectories that are already corrected with the avoidance mechanism.

same height level in a square area of 100×100 m. Each UAV conducts a random walk over a sequence of randomly generated waypoints sampled from an even distribution. A collision between two UAVs is considered to take place when their geometric centers come closer together than 1 m. Table II shows the percentiles of the duration of a scenario before the occurrence of a collision, after which the experiment was stopped. Over one half of the 495 simulation trials ended within 2 minutes after takeoff, which indicates the importance of the proposed system. Simulations with the collision avoidance system showed that the UAVs flew without any collisions for a period of 24 h, with 1.21 m being the minimal measured distance between any of the vehicles.

TABLE II: Percentiles of the period of time before the first collision occurred. The results come from 24 h of simulated flights with 5 UAVs, conducting a 2D random walk on an area of 100×100 m. A total of 495 collisions were recorded.

percentile	0.5	0.75	0.95	0.99
without avoidance	104 s	152 s	264 s	431 s
with avoidance	no collisions			

C. Experimental evaluation of system performance

A series of experiments were performed to validate the system in real-world conditions. Unlike simulations, hardware deployment can reveal design flaws, e.g., due to communication delays and bottlenecks. However, since randomized and long-term tests are impractical in the field, the system was tested on a series of handcrafted trajectories, which were designed to exploit various forms of mutual collisions between vehicles in the group. Figure 7 shows snapshots of one such scenario, in which two UAVs were ordered to swap their locations through the place where a third UAV would remain static. Other scenarios had various priorities of the vehicles, the paths and the initial positions of the UAVs, in the horizontal axis and also in the vertical axis. In all cases, the system prevented a collision in advance, and the vehicles reached their goal positions undamaged.

D. Real-World deployment of the system

The MPC tracker and its ability to follow reference trajectories optimally with respect to the dynamical model of the UAV was utilized for verifying of the methods on the Dubins traveling salesman and orienteering problems [23].

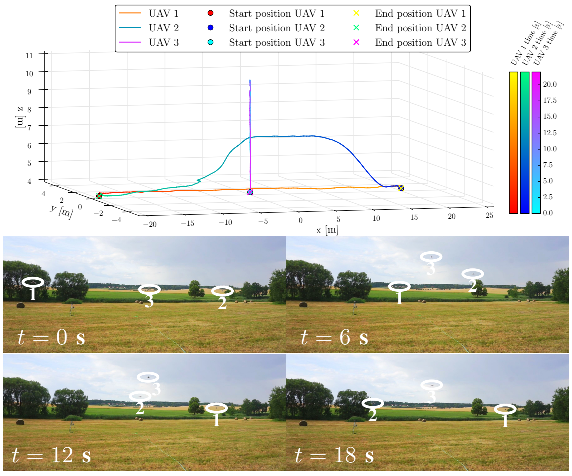


Fig. 7: Three UAVs follow a collision path, in which vehicles #1 and #2 exchanged their positions while flying through the location of vehicle #3, which should remain stationary. A vehicle with a lower number has higher priority than a vehicle with a higher number.

In addition, the proposed system with collision avoidance facilitated an experimental verification of novel research achievements in the field of multi-UAV deployment [7], [24]. Formation flight with 4 UAVs (figure 8d) utilizes migrating leader planning [24]. The MPC tracker enabled safe execution of the experiments, in which the UAVs operated in close proximity. The use of a large-scale sensor network was investigated in [9], in which multiple UAVs localize radio and radiation sources using onboard sensors. The UAV was driven using onboard autonomy, potential malfunctions of which were covered by the collision avoidance system.

The system, as proposed, was successfully deployed in the MBZIRC 2017 robotic competition. The authors were part of the CTU-UPENN-UOL team, which competed in both of the aerial challenges – autonomous landing on a moving car, and collaborative collection of objects by a group of UAVs. Autonomous landing on the moving car, which was traveling at a speed of 15 km/h, was possible thanks to optimal tracking of the predictions of the car movement. The MPC tracker conducted the fastest landing (25.1 s) among all competitors on the site [25]. The robustness of the solution was shown throughout the competition, with successful landings in each of the trials.

The team also scored the highest number of points and won the challenge for autonomous collection of objects by 3 UAVs. As described in [2], the onboard autonomy that is necessary to achieve the goals in real time relied on the MPC tracker and on the collision avoidance system proposed here. During the competition scenario, which took place on an area of 60×90 m, the MPC tracker used an 8.0 s prediction horizon. The UAVs were allowed to fly at up to 8.3 m/s (competition rules), with acceleration of 2.5 m/s^2. Thanks to collision avoidance, we encountered no UAV incidents while developing the system for tests conducted in over one year period with hundreds of experimental flights.

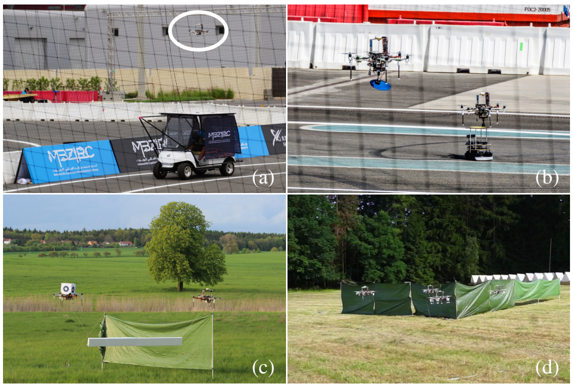


Fig. 8: Photos from real-world use of the system: (a) autonomous landing on a car [25] and (b) multi-robotic manipulation during MBZIRC 2017 [2], (c) collaborative carrying of a large object and (d) 4 UAVs in the formation [24].

V. CONCLUSIONS

This paper presented a novel approach to predictive trajectory tracking with mutual collision avoidance for unmanned aerial vehicles. We have presented a nested control pipeline with a non-linear state controller and a novel MPC trajectory tracker, where the tracker creates an optimal feedforward reference for the controller. The ability to formulate the tracking problem using an MPC approach allows to alleviate the computational problem arising from a classic MPC approach. Additionally, the system incorporates a priority-based collision avoidance technique, which utilizes a prediction of the future trajectories of the vehicles in the group to alter the course of the flight to prevent mutual collisions between vehicles. The proposed system has been used extensively as an underlying platform for experimental verification of novel research methods, and in 2017 it outperformed other teams in the MBZIRC robotic competition.

ACKNOWLEDGMENT

The presented work has been supported by projects: GACR 17-16900Y, CESNET LM2015042, SGS17/187/OHK3/3T/13, ARL grant W911NF-17-2-0181, ONR grants N00014-07-1-0829, N00014-14-1-0510, ARO grant W911NF-13-1- 0350, NSF grants IIS-1426840, IIS-1138847, DARPA grants HR001151626, HR0011516850

REFERENCES

[1] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in gps-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[2] 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*, 2018.

[3] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *IEEE American Control Conference (ACC)*. IEEE, 2014, pp. 2077–2082.

[4] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of mavs using rgb-d sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[5] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1039–1046, 2017.

[6] F. Balampanis, I. Maza, and A. Ollero, "Coastal areas division and coverage with multiple uavs for remote sensing," *Sensors*, vol. 17, no. 4, p. 808, 2017.

[7] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho *et al.*, "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.

[8] 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," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2017.

[9] M. Saska, "Large sensors with adaptive shape realised by selfstabilised compact groups of micro aerial vehicles," in *IEEE International Symposium on Robotic Research (ISRR)*. IEEE, 2017.

[10] 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.

[11] M. Robotics. (2017) Precise (2 cm) indoor gps: For autonomous robots, copters and vr. [Online]. Available: https://marvelmind.com

[12] S. Mastellone, D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[13] M. Cap, J. Gregoire, and E. Frazzoli, "Provably safe and deadlock-free execution of multi-robot plans under delaying disturbances," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5113–5118.

[14] F. G. Lopez, J. Abbenseth, C. Henkel, and S. Dorr, "A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications," in *IEEE European Conference on Mobile Robots (ECMR)*. IEEE, 2017.

[15] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Nonlinear model predictive control for multi-micro aerial vehicle robust collision avoidance," *arXiv preprint: 1703.01164*, 2017.

[16] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5420–5425.

[17] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, 2007.

[18] F. Belkhouche, "Reactive optimal uav motion planning in a dynamic world," *Robotics and Autonomous Systems*, vol. 96, pp. 114 – 123, 2017.

[19] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 6235–6240.

[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2520–2525.

[21] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2016, pp. 992–997.

[22] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[23] R. Penicka, J. Faigl, P. Vana, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, 2017.

[24] M. Saska, T. Baca, and D. Hert, "Formations of unmanned micro aerial vehicles led by migrating virtual leader," in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2016.

[25] T. Baca, P. Stepan, and M. Saska, "Autonomous landing on a moving car with unmanned aerial vehicle," in *IEEE European Conference on Mobile Robotics (ECMR)*. IEEE, 2017.

The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles

Tomas Baca* · **Matej Petrlik** · **Matous Vrba** · **Vojtech Spurny** ·
Robert Penicka · **Daniel Hert** · **Martin Saska**

Received: date / Accepted: date

Abstract We present a multirotor Unmanned Aerial Vehicle (UAV) control and estimation system for supporting replicable research through realistic simulations and real-world experiments. We propose a unique multi-frame localization paradigm for estimating the states of a UAV in various frames of reference using multiple sensors simultaneously. The system enables complex missions in GNSS and GNSS-denied environments, including outdoor-indoor transitions and the execution of redundant estimators for backing up unreliable localization sources. Two feedback control designs are presented: one for precise and aggressive maneuvers, and the other for stable and smooth flight with a noisy state estimate. The proposed control and estimation pipeline are constructed without using the Euler/Tait-Bryan angle representation of orientation in 3D. Instead, we rely on rotation matrices and a novel heading-based convention to represent the one free rotational degree-of-freedom in 3D of a standard multirotor helicopter. We provide an actively maintained and well-documented open-source implementation, including realistic simulation of UAVs, sensors, and localization systems. The proposed system is the product of years of applied research on multi-robot systems, aerial swarms, aerial manipulation, motion planning, and remote sensing. All our results have been supported by real-world system deployment that subsequently shaped the system into the form presented here. In addition, the system was utilized during the participation of our team from the Czech Technical University in Prague in the prestigious MBZIRC 2017 and 2020 robotics competitions, and also in the DARPA Subterranean challenge. Each time, our team was able to secure top places among the best competitors from all over the world.

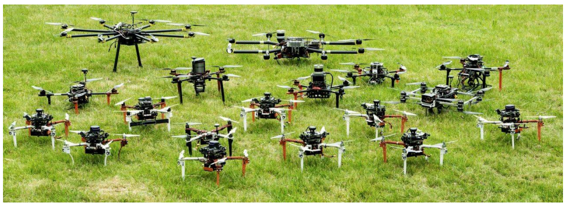


Fig. 1: Multirotor UAV platforms equipped for various scenarios carried out by the system presented here.

Keywords Unmanned Aerial Systems · Multirotor Helicopters · Control Systems Engineering · Educational Robotics

1 INTRODUCTION

The field of mobile robotics is steadily advancing towards smart, small and intelligent mobile agents, capable of autonomously solving complex tasks. Existing Unmanned Ground Vehicle (UGV) platforms already offer researchers complex functions (Clearpath Robotics[1], ETH ANYmal[2]). Ground robotics research tends to focus on high-level systems such as mission autonomy, robot localization, environment mapping, and remote sensing. Platforms such as the Boston Dynamics Spot[3] are out of the box equipped with automatic localization, mapping, path tracking, and navigation in

All authors are with
Multi-Robot Systems group, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague, Czech Republic

* corresponding author, E-mail: tomas.baca@fel.cvut.cz

[1] http://clearpathrobotics.com

[2] http://rsl.ethz.ch/robots-media/anymal

[3] http://www.bostondynamics.com/spot

an environment. However, UAVs, specifically multirotor helicopters, are still under intensive investigation in a wide range of research on all levels of their technological tree. Research in UAVs is still being carried in underlying fields of dynamic system modeling [13], automatic feedback control [26], and trajectory optimization [61]. These fields are vital for understanding and for realizing autonomous flying machines capable of supporting research in higher-level sub-systems for autonomous navigation through an environment, for remote sensing, and for multi-agent systems. Only a handful of UAV platforms are suited for research out of the box, and the researchers are most often tasked with developing full-stack UAV control and guidance to support the needs of these platforms.

Multirotor UAVs are capable of traversing 3D space and are often chosen for exploration and remote sensing in cluttered environments [14], especially when ground robots might fail. Their interesting dynamics makes them still a common choice for demonstrating novel techniques in control theory [54]. In addition, their flight properties, most of all the ability to hover, make them excellent for carrying sensors and for aiding research in distributed and remote sensing. Due to the inherent instability of multirotor dynamics, a continuously-updated feedback control loop is necessary to maintain stable flight. This emphasizes the importance of the onboard localization, state estimation, and control software. Failure of this software threatens the vehicle itself and its surroundings.

Experimental verification of novel methods for UAV systems is nowadays becoming a standard in application-oriented research. However, this comes at the cost of obtaining and maintaining an experimental platform, preferably with a realistic simulation environment. This task is especially demanding if the platform is intended for use outside laboratory conditions. Replicating and validating existing research and comparing it to novel proposed approaches is a necessary part of the research process. We argue that novel methods in applied robotics should be published together with materials necessary for replicating the results. Although the amount of cutting-edge research published with enclosed sources is increasing, the situation is not yet ideal. A positive trend is most prominent in machine learning, in computer vision, and in Simultaneous Localization And Mapping systems (SLAMs). Sadly, research groups rarely release all parts of their experimental and testing systems, making their results difficult to validate and replicate. Furthermore, research in the field of UAV control is typically limited to non-realistic simulations and, in many cases no implementation in a real UAV exists. To solve this, the robotics community has been collaborating on developing application frameworks [23, 17, 73] that unify the way for algorithms of different origin to interact and form complex robotics systems. The emergence of such frameworks helps to create systems that are reusable across research groups.

The Robot Operating System (ROS) [53] is one of many middleware robotics frameworks [73, 17, 23, 25, 12, 38]. ROS has several features that have raised it to the most prominent framework among UAV researchers. Renowned sensor manufactures such as *Velodyne*, *Ouster*, *Terabee*, *Garmin*, *MatrixVision* and *Intel* are making a significant effort to provide ROS drivers for their products. State-of-the-art research in computer vision and in SLAM algorithms is often accompanied by functioning ROS implementations of the published methods [52, 83, 28]. Finally, open-source robotic simulators such Gazebo[4] and CoppeliaSim[5] (previously V-REP) provide integration with ROS. These features allow researchers to focus their research more narrowly rather than on implementation aspects of a whole robotics pipeline. However, even with such advances, many tasks remain unsolved on the way to a real-world UAV platform for research, especially to platforms that can perform outside laboratory conditions.

Through this publication, we intend to share our full-stack UAV platform with all essential capabilities for research, development, and testing of novel methods. Our system is a product of many years of development in various robotic projects. The proposed platform has provided support for state-of-the-art research and has resulted in dozens of high-quality publications in cooperation with several research groups. These works have focused on particular applications and on relevant research, but the underlying system will be thoroughly described and published for the first time in this manuscript. We offer a modular and extensible open-source platform, together with a complex simulation environment. The platform is suited for both indoor and outdoor use, with an emphasis on onboard multi-sensor fusion to allow safe execution of experiments outside laboratory conditions (see Fig. 1 for a showcase of our hardware platforms). We propose a pair of feedback controllers that satisfy the needs of a wide range of applications, ranging from fast and aggressive flight to stable flight using unreliable sensors producing noisy data.

[4] Gazebo simulator, `http://gazebosim.org`

[5] `http://www.coppeliarobotics.com`

1.1 State of the art

Research-focused UAVs are most commonly equipped with a low-level embedded flight controller. Available flight controllers [16] range from feature-packed open-source systems, such as Pixhawk, to proprietary commercial units manufactured by DJI. Table 1 shows a comparison of often used solutions. Pixhawk is often used in research projects (including our project), typically running either of the two open-source firmwares: PX4 [35] and ArduPilot[6]. Although all of these flight stacks provide sophisticated features up to waypoint tracking and mission execution, the features are rarely used within real-world applications and for real-world verification of research. As supported by the existence of many high-level control systems [59, 81, 20, 66, 1, 36], researchers instead use other onboard companion computers to execute a custom localization system, state estimators, and flight controllers, and only low-level control commands are provided for the embedded flight controller [27, 52, 21, 39, 61].

Several comparable UAV systems have been published and released. Table 2 compares existing solution with the system proposed in this publication.

The RotorS [20] simulator is an initial release for the Aeroworks EU project[7]. It provides Gazebo-based simulation of the now discontinued *Ascending Technologies* UAV system. The control pipeline features are basic, with little potential for transfer to real-world conditions. The system does not appear to be kept up-to-date, which gradually diminishes its usability and applicability. Moreover, the latest supported version of ROS is *ROS Kinetic*, which potentially provides lower compatibility with newer hardware and software.

OpenUAV[8] [66] is a UAV swarm simulation testbed. The system does not appear to allow transfer to a real-world setting, and is designed only to support prototyping of basic research in swarming. The UAVs are assumed to be controlled and localized solely using an embedded flight controller with PX4 firmware. This is comparable hardly with the numerous sensors and localization systems that our system allows to simulate and to be used in a real-world scenario.

ReCOPTER[9] [1] proposes an open-source multirotor system for research. The available materials were released as supporting material for the published paper. However, no software was attached, and the materials have not been updated since. Similarly, a framework for drone control using the Vicon localization system named MAVwork[10] [36] was published in 2011, but has not been updated since. Although sources were made available, they offered only basic features that would be difficult to transfer into a real-world scenario.

The XTDrone[11] [81] simulation testbed offers many complex functionalities that are comparable with our proposed system, including simulation of onboard sensors and complex localization systems. However, the control pipeline relies entirely on the PX4 embedded control software. This significantly limits any transfer to a custom hardware platform, or even the ability to simulate realistic conditions using onboard localization systems. Thus, the use of XTDrone outside laboratory conditions is mostly limited to Global Positioning System (GPS)-localized flight in a non-cluttered outdoor environment.

The full-stack Aerostack system[12] [59, 60] was designed for deployment of multirotor UAVs. The system is continuously being updated, and it offers an option to transfer to a real-world platform. According to the preprint[72] where the authors used Aerostack during the MBZIRC 2020 competition, the system's real-world deployment is possible. However, with the used DJI-based flight controller, the control command supplied to the underlying embedded control layer are limited to desired orientation and thrust. This level of control limits the potential precision and control authority comparing to our system. Furthermore, the system lacks the feature of switching between multiple frames of reference, which is one of our system's contributions. As it happens, the team of authors of Aerostack did not compete in the wall-building challenge of MBZIRC 2020, in which we found the feature to be crucial to precisely collect bricks by a group of UAVs.

Besides the Aerostack system, no other existing platform provides a full-stack system for a multirotor UAV that is actively being supported and updated. Many publications provide accompanying software sources that are released without being further updated. By contrast, we have decided to publish and release our working system with all its components to allow members of the research community, research teams, and students to engage in UAV research as effortlessly as possible. We aim to provide a thoroughly-documented open-source system to allow researchers and students to shorten their initial learning curve and to focus on their research instead of developing yet another control pipeline. In our case, the future continuity of our system is supported for use in the next 5+ years

[6] `http://ardupilot.org`

[7] Aeroworks EU project, `http://www.aeroworks2020.eu`.

[8] OpenUAV, `http://github.com/Open-UAV`

[9] ReCOPTER, `http://github.com/thedinuka/ReCOPTER`

[10] MAVwork, `http://github.com/uavster/mavwork`

[11] XTDrone, `http://github.com/robin-shaun/XTDrone`

[12] Aerostack, `http://github.com/Vision4UAV/Aerostack`

through our numerous activities in projects supported by European grants[13] and by national grants[14].

The proposed platform is provided with two control designs — extended *SE(3) geometric tracking* [32] for agile and aggressive flight, and the novel *MPC controller* for stable flight using a potentially unreliable state estimate. However, we highlight the modularity of our platform, which can easily be extended with new control approaches as needed. The survey of UAV controllers provides a rich list of potentially useful control techniques [42]. For example, a novel adaptive backstepping controller [82, 30] may provide better performance during aggressive maneuvers, thanks to the included rotor drag compensation. The proposed extension to *geometric tracking on SE(3)* [32] can be further improved with remarks from [33] to provide robust control to bounded uncertainties. Furthermore, nonlinear Model Predictive Control (MPC) controllers are becoming popular [41, 47, 27], thanks to their inherent ability to deal with obstacle avoidance. However, when dealing not just with theoretical work but also with the deployment of UAVs in real-world conditions, we favor practicality over complexity. We therefore propose the use of relatively simple controllers (described further in Sec. 5), with well tractable performance.

Although the vast majority of existing control systems (all those mentioned above) and control designs for UAVs [42] rely on the separation of the attitude and translational dynamics, methods for full-dynamics control also exist. Those end-to-end control designs [51, 31] treat the full dynamics of the multirotor UAV as one model. Despite the design being cutting-edge, we prefer splitting the dynamics for many practical reasons. We prefer the split design aspects for the ability to introspect and limit the inner states if needed. For example, limit the maximum acceleration or tilt produced by the controller, regardless of the outer translational control loop. Also, the large variance on the input (estimator noise) of an end-to-end controller can produce unbounded control outputs (moments) and states (acceleration, tilt, tilt rate) unless the control design natively supports constraining them. Finally, as will be discusses in this manuscript, we employ on attitude controller (*geometric force tracking on SO(3)*) and several *outer loop* controllers (*MPC*, *SE(3)*, *Speed tracker*) depending on the particular experimental scenario requirements. However, the proposed platform allows for the use of an end-to-end design, if needed.

platform	open source	modular	SITL/HITL	outside lab	rate input
Pixhawk	**SW & HW**	+	+	+	+
DJI	-	-	- (proprietary)	+	-
Ardupilot	SW	+	+	+	-
Parrot	SW	-	+	-	-

Table 1: Comparison of commonly-used embedded flight controllers and low-level control systems. The Pixhawk flight controller was chosen due to several factors: both the hardware and software is open-source, the controller is modular enough to be used on a variety of custom multirotor platforms, Pixhawk supports both hardware- and software-in-the-loop simulation, can be used outside of laboratory conditions, and supports attitude rate input.

1.2 Contributions

The proposed system goes beyond existing systems with

- a novel bank-of-filters estimator design that overcomes challenges with diverse sensory equipment,
- a heading-oriented control design, devoid of ambiguous use of Euler/Tait-Bryan angles,
- a body/world disturbance estimation approach that does not rely on a specific state estimator design,
- a reliable MPC-based controller with the benefits of the nonlinear *SO(3)* force feedback,
- a system that can be employed with a variety of onboard localization systems and sensors,
- an ability to supply references in coordinate frames, which differ from the feedback loop reference frame,
- extended trajectory generation mechanism [40] for generating fast trajectories through waypoints,
- an open-source implementation [15] suitable for large variety of applications and scenarios,
- a modular design suitable for testing new control methods allowing them to be hot-swapped in mid-flight with the provided ones.

The system is not only innovative, but also provides practical contributions to the community. The open-source implementation of the proposed platform has been tested extensively in real-world settings and in conditions of outdoor fields, in a forest, indoors, in a factory, in mines, caves and tunnels, during object manipulation, during fast and aggressive flights, and in autonomous landing on a moving platform. The system includes a simulation environment based on the Gazebo 3D simulator with realistic sensors and models that can be run in real time. The released platform is fully compatible with multiple releases of ROS (Melodic, Noetic), and is being actively used and maintained. The system is scalable for multiple UAVs and is well suited for research in swarming.

[13] `https://aerial-core.eu`, `http://rci.cvut.cz`

[14] `http://mrs.felk.cvut.cz`

[15] `http://github.com/ctu-mrs/mrs_uav_system`

platform	open source	modular	simulation	outside laboratory	multi-frame localization	rate output	software last updated	reference
MRS UAV system	+	+	+	+	+	+	**2020**	—
Aerostack	+	+	+	+	-	-	2020	[59]
XTDrone	+	+	+	-	-	-	2020	[81]
RotorS	+	+	+	-	-	+	2020	[20]
OpenUAV	+	-	+	-	-	-	2020	[66]
ReCOPTER	+	-	-	-	-	-	2015	[1]
MAVwork	+	+	-	-	-	-	2013	[36]

Table 2: Comparison of high-level open-source UAV systems. The proposed system is extensible and modular, comes with an extensive simulation environment, is designed to be used outside of laboratory conditions, provides the novel multi-frame localization estimator, and supplies the attitude rate command to the underlying embedded flight controller.

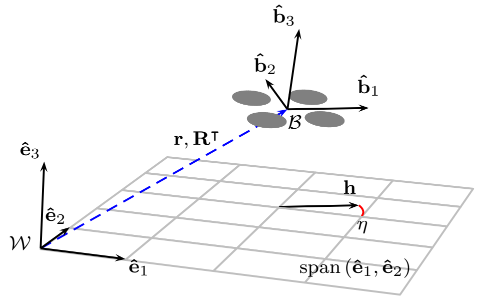


Fig. 2: The image depicts the world frame $\mathcal{W} = \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ in which the 3D position and the orientation of the UAV body is expressed. The body frame $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ relates to $\mathcal{W}$ by the translation $\mathbf{r} = [x, y, z]^\intercal$ and by rotation $\mathbf{R}^\intercal$. The UAV heading vector $\mathbf{h}$, which is a projection of $\hat{\mathbf{b}}_1$ to the plane $\mathit{span}\left(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\right)$, forms the heading angle $\eta = \mathrm{atan2}\left(\hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_1\right) = \mathrm{atan2}\left(\mathbf{h}_{(2)}, \mathbf{h}_{(1)}\right)$.

1.3 System Architecture & Outline

We start with a description of the multirotor UAV dynamics model (Sec. 2), which is the foundation for further control design. The proposed platform consists of several interconnected subsystems, as depicted in Fig. 3. *Mission & navigation* software supplies the desired trajectory, i.e., a time-parametrized sequence of the desired position and heading. The module is specific to any particular application of the platform (autonomous exploration, swarming, remote sensing) and is conveyed via the publications presented in Sec. 9. We will therefore not focus on the module here, as numerous examples in other papers have shown where the proposed system may be applied. Onboard sensor data (e.g., position measurements from GPS, velocity measurements from visual odometry) are processed by the *State estimator*, which provides the *Reference tracker* and the *Reference controller* with hypotheses of UAV states in all available frames of reference (Sec. 3). The block generates estimated states of the translational dynamics with the UAV orientation, for all considered world frames of reference. One of the world frames is always selected as the *main* frame, in which a feedback loop is closed by the *Feedback controller* block. The desired trajectory is processed by a *Reference tracker* (see Sec. 4), and is then converted into a feasible, smooth, and evenly-sampled full-state control reference. The reference contains the desired position, its derivatives up to the jerk, the heading[16], and the heading rate, supplied at 100 Hz. The reference is used by a *Reference controller* (see Sec. 5) to provide feedback control of the translational dynamics and the orientation of the UAV. This block creates an attitude rate $\boldsymbol{\omega}_d$ and a thrust command T_d, which are sent to an embedded flight controller[17]. We consider the underlying hardware platform already pre-configured with motors, motor speed controllers, and a basic embedded flight controller providing *Attitude rate control.* We rely on the embedded flight controller for a backup control using a remote controller in case of a malfunction of the high-level computer. The flight controller encapsulates the underlying physical UAV system with motors and motor Electronic Speed Controllers (ESCs) and creates 4 new controllable degrees-of-freedom (DOFs): the desired angular speed around $\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3$ and the desired thrust $T_d \in [0, 1]$ of all propellers. This encapsulation provides an abstraction that allows us to control any standard multirotor helicopter regardless of the number of propellers and the geometry of its fuselage. Section 6 briefly describes a trajectory generation approach used within our system. Section 7 contains remarks on the implementation aspects of our system. Section 8 provides the results of an experimental evaluation of the control system, with emphasis on a comparison between

[16] The heading, as defined later in our paper, removes ambiguities caused by numerous conventions of the widely-used Tait-Bryan and Euler angles [15], and provides a user-friendly representation of the 4th controllable degree of freedom of a multirotor UAV.

[17] The proposed system is compatible with the Pixhawk flight controller, installed with PX4 firmware.

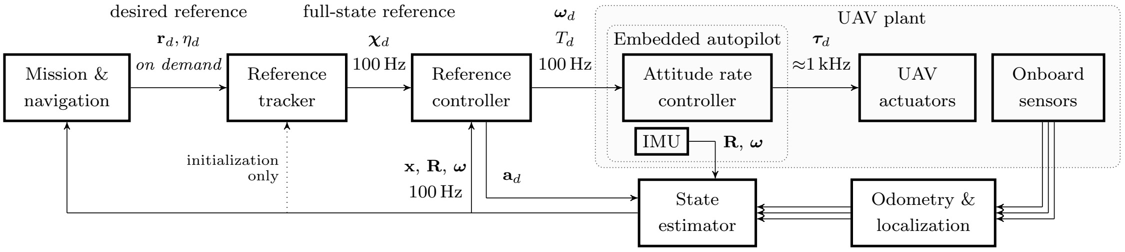


Fig. 3: A diagram of the system architecture: *Mission & navigation* software supplies the position and heading reference ($\mathbf{r}_d$, η_d) to a reference tracker. *Reference tracker* creates a smooth and feasible reference $\boldsymbol{\chi}$ for the reference feedback controller. The feedback *Reference controller* produces the desired thrust and angular velocities (T_d, $\boldsymbol{\omega}_d$) for the Pixhawk embedded flight controller. The *State estimator* fuses data from *Onboard sensors* and *Odometry & localization* methods to create an estimate of the UAV translation and rotation ($\mathbf{x}$, $\mathbf{R}$).

the simulation environment and a real-world counterpart. Finally, examples of real-world use and application of the system for validating research, for education, and for competing in robotics competitions are presented in Sec. 9.

2 Multirotor aerial vehicle dynamics model

The design of a high-performance attitude and position controller often requires an accurate model of the system. Here, we recall the widely-used dynamical model of a multirotor aerial vehicle [32]. Figure 2 illustrates the coordinate frames used in this manuscript. For the sake of brevity, we do not explicitly annotate variables with their respective coordinate frames, since all variables with the exception of the angular velocities $\boldsymbol{\omega}$ are expressed in the world coordinate frame. The UAV feedback control relies on state variables defined as:

$\mathbf{r} = [x, y, z]^\intercal$ the position of the center of the mass of a UAV in the world frame,

$\dot{\mathbf{r}} \in \mathbb{R}^3$ the velocity of the center of the mass of a UAV in the world frame,

$\ddot{\mathbf{r}} \in \mathbb{R}^3$ the acceleration of the center of a mass of a UAV in the world frame,

$\mathbf{R} \in SO(3) \subseteq \mathbb{R}^{3\times3}$ the rotation matrix from the body frame of a UAV to the world frame, $\det \mathbf{R} = 1$, $\mathbf{R}^\intercal = \mathbf{R}^{-1}$,

$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\intercal$ the angular velocity in the body frame of a UAV.

These states are linked by a nonlinear model, which has a translation part:

$$m\ddot{\mathbf{r}} = f\mathbf{R}\hat{\mathbf{e}}_3 - mg\hat{\mathbf{e}}_3, \tag{1}$$

and a rotational part

$$\dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\Omega}, \tag{2}$$

where $\boldsymbol{\Omega}$ is the tensor of angular velocity, under the condition $\boldsymbol{\Omega}\,\mathbf{v} = \boldsymbol{\omega} \times \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^3$. The vehicle experiences downwards gravitational acceleration with magnitude $g \in \mathbb{R}$ together with the thrust force f created collectively by the propellers in the direction of $\hat{\mathbf{b}}_3$. However, as we are focused on non-aerobatic flight, we separately consider and estimate the azimuth of the $\hat{\mathbf{b}}_1$ axis in the world as the UAV *heading*. Under the condition of $|\hat{\mathbf{e}}_3^\intercal \hat{\mathbf{b}}_1| > 0$, we define the heading as

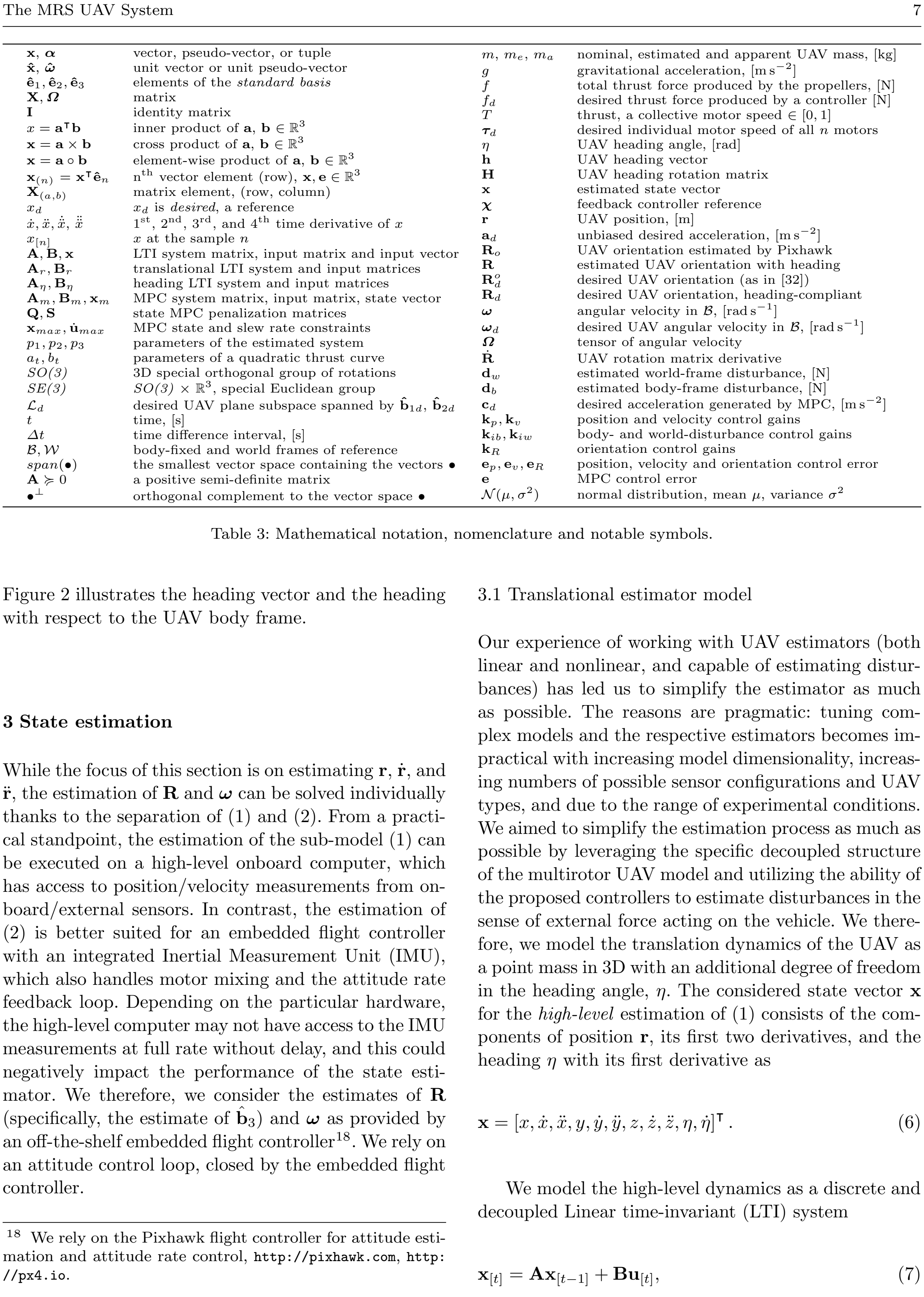
$$\eta = \operatorname{atan2}\left(\hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_1\right). \tag{3}$$

The heading is a more intuitive alternative to the widely-used *yaw* angle as one of the 4 controllable DOFs. It is possible to use the yaw, but with the assumption that the tilt of the UAV ($\cos^{-1} \hat{\mathbf{b}}_3^\intercal \hat{\mathbf{e}}_3$) is low, near horizontal. We advice against the use of the *Tait-Bryan angles* (commonly mistaken for *Euler angles*), due to the overwhelming number of conventions, which often lead to misunderstanding. Generally, the widely-used *yaw* angle (as in *Euler angles*, *Tait-Bryan angles* [15]) has no direct meaning with respect to the particular orientation of any of the body axes in any of the conventions, since the final orientation also depends on the remaining two rotations (pitch, roll). A user would need to take the remaining part of the desired orientation (produced by the controllers) into account in the *Mission & navigation* software to properly design the desired *yaw*, which leads to a *chicken or egg* problem. We therefore, we define the heading vector by the $\hat{\mathbf{b}}_1$ axis as

$$\mathbf{h} = \left[\mathbf{R}_{(1,1)}, \mathbf{R}_{(2,1)}, 0\right]^\intercal = [\mathbf{b}_1^\intercal \hat{\mathbf{e}}_1, \mathbf{b}_1^\intercal \hat{\mathbf{e}}_2, 0]^\intercal \tag{4}$$

and its normalized form

$$\hat{\mathbf{h}} = \frac{\mathbf{h}}{||\mathbf{h}||} = [\cos\eta, \sin\eta, 0]^\intercal. \tag{5}$$



The MRS UAV System 7

$\mathbf{x}$, $\boldsymbol{\alpha}$	vector, pseudo-vector, or tuple	m, m_e, m_a	nominal, estimated and apparent UAV mass, [kg]
$\hat{\mathbf{x}}$, $\hat{\boldsymbol{\omega}}$	unit vector or unit pseudo-vector	g	gravitational acceleration, [$\mathrm{m\,s^{-2}}$]
$\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$	elements of the *standard basis*	f	total thrust force produced by the propellers, [N]
$\mathbf{X}, \boldsymbol{\Omega}$	matrix	f_d	desired thrust force produced by a controller [N]
$\mathbf{I}$	identity matrix	T	thrust, a collective motor speed $\in [0,1]$
$x = \mathbf{a}^\intercal\mathbf{b}$	inner product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$	$\boldsymbol{\tau}_d$	desired individual motor speed of all n motors
$\mathbf{x} = \mathbf{a} \times \mathbf{b}$	cross product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$	η	UAV heading angle, [rad]
$\mathbf{x} = \mathbf{a} \circ \mathbf{b}$	element-wise product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$	$\mathbf{h}$	UAV heading vector
$\mathbf{x}_{(n)} = \mathbf{x}^\intercal\hat{\mathbf{e}}_n$	n$^{\text{th}}$ vector element (row), $\mathbf{x}, \mathbf{e} \in \mathbb{R}^3$	$\mathbf{H}$	UAV heading rotation matrix
$\mathbf{X}_{(a,b)}$	matrix element, (row, column)	$\mathbf{x}$	estimated state vector
x_d	x_d is *desired*, a reference	$\boldsymbol{\chi}$	feedback controller reference
$\dot{x}, \ddot{x}, \dddot{x}, \ddddot{x}$	1$^{\text{st}}$, 2$^{\text{nd}}$, 3$^{\text{rd}}$, and 4$^{\text{th}}$ time derivative of x	$\mathbf{r}$	UAV position, [m]
$x_{[n]}$	x at the sample n	$\mathbf{a}_d$	unbiased desired acceleration, [$\mathrm{m\,s^{-2}}$]
$\mathbf{A}, \mathbf{B}, \mathbf{x}$	LTI system matrix, input matrix and input vector	$\mathbf{R}_o$	UAV orientation estimated by Pixhawk
$\mathbf{A}_r, \mathbf{B}_r$	translational LTI system and input matrices	$\mathbf{R}$	estimated UAV orientation with heading
$\mathbf{A}_\eta, \mathbf{B}_\eta$	heading LTI system and input matrices	$\mathbf{R}_d^o$	desired UAV orientation (as in [32])
$\mathbf{A}_m, \mathbf{B}_m, \mathbf{x}_m$	MPC system matrix, input matrix, state vector	$\mathbf{R}_d$	desired UAV orientation, heading-compliant
$\mathbf{Q}, \mathbf{S}$	state MPC penalization matrices	$\boldsymbol{\omega}$	angular velocity in $\mathcal{B}$, [$\mathrm{rad\,s^{-1}}$]
$\mathbf{x}_{max}, \dot{\mathbf{u}}_{max}$	MPC state and slew rate constraints	$\boldsymbol{\omega}_d$	desired UAV angular velocity in $\mathcal{B}$, [$\mathrm{rad\,s^{-1}}$]
p_1, p_2, p_3	parameters of the estimated system	$\boldsymbol{\Omega}$	tensor of angular velocity
a_t, b_t	parameters of a quadratic thrust curve	$\dot{\mathbf{R}}$	UAV rotation matrix derivative
SO(3)	3D special orthogonal group of rotations	$\mathbf{d}_w$	estimated world-frame disturbance, [N]
SE(3)	*SO(3)* $\times \mathbb{R}^3$, special Euclidean group	$\mathbf{d}_b$	estimated body-frame disturbance, [N]
$\mathcal{L}_d$	desired UAV plane subspace spanned by $\hat{\mathbf{b}}_{1d}$, $\hat{\mathbf{b}}_{2d}$	$\mathbf{c}_d$	desired acceleration generated by MPC, [$\mathrm{m\,s^{-2}}$]
t	time, [s]	$\mathbf{k}_p, \mathbf{k}_v$	position and velocity control gains
Δt	time difference interval, [s]	$\mathbf{k}_{ib}, \mathbf{k}_{iw}$	body- and world-disturbance control gains
$\mathcal{B}, \mathcal{W}$	body-fixed and world frames of reference	$\mathbf{k}_R$	orientation control gains
$span(\bullet)$	the smallest vector space containing the vectors $\bullet$	$\mathbf{e}_p, \mathbf{e}_v, \mathbf{e}_R$	position, velocity and orientation control error
$\mathbf{A} \succcurlyeq 0$	a positive semi-definite matrix	$\mathbf{e}$	MPC control error
$\bullet^\perp$	orthogonal complement to the vector space $\bullet$	$\mathcal{N}(\mu, \sigma^2)$	normal distribution, mean μ, variance σ^2

Table 3: Mathematical notation, nomenclature and notable symbols.

Figure 2 illustrates the heading vector and the heading with respect to the UAV body frame.

3 State estimation

While the focus of this section is on estimating $\mathbf{r}$, $\dot{\mathbf{r}}$, and $\ddot{\mathbf{r}}$, the estimation of $\mathbf{R}$ and $\boldsymbol{\omega}$ can be solved individually thanks to the separation of (1) and (2). From a practical standpoint, the estimation of the sub-model (1) can be executed on a high-level onboard computer, which has access to position/velocity measurements from onboard/external sensors. In contrast, the estimation of (2) is better suited for an embedded flight controller with an integrated Inertial Measurement Unit (IMU), which also handles motor mixing and the attitude rate feedback loop. Depending on the particular hardware, the high-level computer may not have access to the IMU measurements at full rate without delay, and this could negatively impact the performance of the state estimator. We therefore, we consider the estimates of $\mathbf{R}$ (specifically, the estimate of $\hat{\mathbf{b}}_3$) and $\boldsymbol{\omega}$ as provided by an off-the-shelf embedded flight controller[18]. We rely on an attitude control loop, closed by the embedded flight controller.

3.1 Translational estimator model

Our experience of working with UAV estimators (both linear and nonlinear, and capable of estimating disturbances) has led us to simplify the estimator as much as possible. The reasons are pragmatic: tuning complex models and the respective estimators becomes impractical with increasing model dimensionality, increasing numbers of possible sensor configurations and UAV types, and due to the range of experimental conditions. We aimed to simplify the estimation process as much as possible by leveraging the specific decoupled structure of the multirotor UAV model and utilizing the ability of the proposed controllers to estimate disturbances in the sense of external force acting on the vehicle. We therefore, we model the translation dynamics of the UAV as a point mass in 3D with an additional degree of freedom in the heading angle, η. The considered state vector $\mathbf{x}$ for the *high-level* estimation of (1) consists of the components of position $\mathbf{r}$, its first two derivatives, and the heading η with its first derivative as

$$\mathbf{x} = [x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}, \eta, \dot{\eta}]^\intercal \,. \tag{6}$$

We model the high-level dynamics as a discrete and decoupled Linear time-invariant (LTI) system

$$\mathbf{x}_{[t]} = \mathbf{A}\mathbf{x}_{[t-1]} + \mathbf{B}\mathbf{u}_{[t]}, \tag{7}$$

[18] We rely on the Pixhawk flight controller for attitude estimation and attitude rate control, `http://pixhawk.com`, `http://px4.io`.

with 4 independently estimated subsystems expressed together by matrices **A** and **B** as

$$\mathbf{A}(\mathit{\Delta}t{,}p_1{,}p_2{,}p_3) = \\ = \begin{bmatrix} \mathbf{A}_r(\mathit{\Delta}t{,}p_1) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_r(\mathit{\Delta}t{,}p_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_r(\mathit{\Delta}t{,}p_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_\eta(\mathit{\Delta}t{,}p_3) \end{bmatrix}, \quad (8)$$

$$\mathbf{B}(\mathit{\Delta}t{,}p_1{,}p_2{,}p_3) = \begin{bmatrix} \mathbf{B}_r(\mathit{\Delta}t{,}1{-}p_1) \\ \mathbf{B}_r(\mathit{\Delta}t{,}1{-}p_1) \\ \mathbf{B}_r(\mathit{\Delta}t{,}1{-}p_2) \\ \mathbf{B}_\eta(\mathit{\Delta}t{,}1{-}p_3) \end{bmatrix}. \quad (9)$$

Matrices $\mathbf{A}_r$ and $\mathbf{B}_r$ are the sub-system matrices for the translation part of the model:

$$\mathbf{A}_r(\mathit{\Delta}t{,}a) = \begin{bmatrix} 1 & \mathit{\Delta}t & \frac{\mathit{\Delta}t^2}{2} \\ 0 & 1 & \mathit{\Delta}t \\ 0 & 0 & a \end{bmatrix}, \ \mathbf{B}_r(\mathit{\Delta}t{,}b) = \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix}, \quad (10)$$

and $\mathbf{A}_\eta$ and $\mathbf{B}_\eta$ are the sub-system matrices for the heading part of the model:

$$\mathbf{A}_\eta(\mathit{\Delta}t{,}a) = \begin{bmatrix} 1 & \mathit{\Delta}t \\ 0 & a \end{bmatrix}, \ \mathbf{B}_\eta(\mathit{\Delta}t{,}b) = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad (11)$$

with $\mathit{\Delta}t$ being the sampling step of the estimator, and with p_1, p_2, p_3 being the 1st order transfer parameters for the horizontal, vertical and heading subsystems respectively. This model has only three free parameters (assuming that both horizontal axes behave identically), which which simplifies its tuning and allows it to be reused between various UAV platforms without changes. The decoupling of the system to (10) and (11) is used during implementation to speed up the computations thanks to operations with smaller matrices.

3.1.1 System input

System input **u** consists of the *unbiased desired acceleration* $\mathbf{a}_d$. As discussed later in Sec. 5.8, the controllers report on the desired acceleration caused by their control output. However, the controllers are required to supply the desired unbiased acceleration, i.e., without compensation for gravity acceleration, integrated *body* and *world* force disturbances, and the estimated UAV mass difference. All the biases compensated by our controllers are subtracted from the desired acceleration, thanks to their physical dimension being convertible into acceleration. The heading subsystem is left without an input, since measurement corrections from embedded gyroscopes (see Sec. 3.4) are more than sufficient to maintain a stable and quickly-converging estimate.

3.1.2 Sources of measurement

We often work with a very diverse set of onboard sensors and localization systems. Some systems directly provide us with 3D UAV position and heading, e.g., 3D visual and laser SLAMs [52, 83], which can be directly fused into the position and heading state of our filters. When a sensory system provides only a 2D (horizontal) position measurement, e.g., the Global Navigation Satellite System (GNSS) system or a 2D laser SLAM [28], we use an additional measurement of UAV height above the ground provided by down-facing rangefinder. Some systems may provide us only with a velocity measurement, e.g., an optic flow algorithm[19]. Optic flow measurements can be used for an odometry estimate of the position and heading when coupled with a height sensor. 3D Light Detection and Ranging (LiDAR) SLAM might also provide us with odometry velocity measurements (e.g., from a scan-matching algorithm), and absolute position measurements. Heading estimation fuses measured angular velocity $\boldsymbol{\omega}$ supplied by an IMU. The magnetometer is fused when flying with the use of a GNSS localization system.

3.2 Linear Kalman Filter

The dynamic model is estimated by a recursive discrete Linear Kalman Filter (LKF). This estimator, coupled with the model (7), exhibits stable and fast tracking of the states of the translational dynamics, under the condition that the reference controller is capable of calculating and compensating for biases such as external force disturbance or internal input offset (see Sec. 5). Under these conditions, the use of more complex nonlinear filters, such as the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF), would not provide us with the desired computational-cost benefit.

In our experience, it is simpler and more practical to utilize controllers with this property (potentially using any source of the UAV state) rather than to build complex (nonlinear) estimators, which would estimate the biases themselves. With this approach, any source of the UAV state can be used, even substituting the proposed estimator. The sources of measurements and the estimators that are used may change from platform to platform within a laboratory, even to the extent of not using an onboard estimator at all (e.g., when using the external motion capture system). For such situations, force disturbance estimation is provided by our control pipeline. If the proposed platform were to have relied on

[19] `http://github.com/ctu-mrs/mrs_optic_flow`

an estimator capable of estimating external force disturbances, this substitution would not have been possible. Thus, this choice keeps the platform more universal.

3.3 Updating the UAV orientation with the estimated heading

Since we estimate the UAV heading separately, the original UAV rotation $\mathbf{R}_o = \left[\hat{\mathbf{b}}_{1o}, \hat{\mathbf{b}}_{2o}, \hat{\mathbf{b}}_{3o}\right]$ supplied by the embedded flight controller is modified by generating new body frame vectors $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, and $\hat{\mathbf{b}}_3$. For that, the original heading η_o is firstly calculated as

$$\eta_o = \mathrm{atan2}\left(\hat{\mathbf{b}}_{1o}^{\intercal}\hat{\mathbf{e}}_2, \hat{\mathbf{b}}_{1o}^{\intercal}\hat{\mathbf{e}}_1\right). \tag{12}$$

Then the difference between the original heading and the estimated heading is calculated:

$$\Delta\eta = \eta - \eta_o, \tag{13}$$

and the original orientation is rotated around $\hat{\mathbf{e}}_3$ by the heading difference:

$$\mathbf{R} = \begin{bmatrix} \cos\Delta\eta & -\sin\Delta\eta & 0 \\ \sin\Delta\eta & \cos\Delta\eta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_o. \tag{14}$$

3.4 Fusing angular rates into the heading rate state

Although ω_3 (the yaw rate) is often treated as the heading rate, it is applicable only as an approximation under small tilts: $\angle(\hat{\mathbf{b}}_3, \hat{\mathbf{e}}_3) \lessapprox 10°$. In general, all the components of an arbitrary angular speed $\boldsymbol{\omega}$ contribute to the resulting heading rate. To obtain the heading rate, we first apply (2) to obtain $\dot{\mathbf{R}}$, the first time derivative of the rotational matrix. Components $\dot{\mathbf{R}}_{(1,1)}$, $\dot{\mathbf{R}}_{(2,1)}$, which represent the rate of change of $\hat{\mathbf{b}}_1$ along $\hat{\mathbf{e}}_1$, $\hat{\mathbf{e}}_2$, respectively, are extracted and are used to evaluate the total differential of atan2() in the current orientation $\mathbf{R}$ to obtain the heading rate:

$$\dot{\eta} = \frac{-\mathbf{R}_{(2,1)}}{\mathbf{R}_{(1,1)}^2 + \mathbf{R}_{(2,1)}^2}\dot{\mathbf{R}}_{(1,1)} + \frac{\mathbf{R}_{(1,1)}}{\mathbf{R}_{(1,1)}^2 + \mathbf{R}_{(2,1)}^2}\dot{\mathbf{R}}_{(2,1)}. \tag{15}$$

As with most heading-related operations, this operation is only feasible if $|\hat{\mathbf{e}}_3^{\intercal}\hat{\mathbf{b}}_1| > 0$.

3.5 Bank of filters for multiple hypotheses

With the individual filter structure defined, we now establish a bank of Kalman filters $\mathcal{K} = \{K_1, K_2, ..., K_n\}$. The bank of filters allows for simultaneous estimation of the UAV state from various combinations of onboard sensors, without necessarily combining all the measurements ($\mathbf{z} \in \mathbb{R}^m$, where m is the number of measured states) into a single hypothesis. This type of separation is beneficial for many applications, e.g., for transitions from one sensory system to another (e.g., GNSS → indoor SLAM), for running multiple instances of one filter with different parameters, or for maintaining a backup estimator to facilitate emergency landing. Each filter maintains its hypothesis $\mathbf{x}_n$, covariance $\mathbf{\Sigma}_n$, and is corrected by a different set of measurements $\mathbf{z}_n \subseteq \mathbf{z}$.

Multiple hypotheses $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ with covariances $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, ..., \mathbf{\Sigma}_n$ are estimated by the respective filters $K_1, K_2, ..., K_n$ as depicted in Fig. 4. An *arbiter* chooses one of the available hypotheses that is being outputted as the current state estimate. The *arbiter* selects/changes the current filter and its corresponding hypothesis $\mathbf{x}_*$ with covariance Σ_* by one of several criteria:

- a request for a particular filter by the *Mission & navigation* part of the pipeline,
- the current filter becomes unreliable,
- $\mathbf{x}_* = \mathbf{x}_k = \mathrm{argmin}_{\mathbf{x}}\, \mathrm{trace}\left(\boldsymbol{\Sigma}_k\right)$ otherwise.

Whenever the *arbiter* switches the output, the coordinates of the UAV in the world frame change, although the physical manifestation of the UAV has not moved. This switch is treated by the rest of the control pipeline as a sudden change between frames of reference; the change in numerical values can be arbitrary. Any internal states of trackers and controllers are recalculated to the new frame of reference. Therefore, the switch is not apparent to an outside observer as the transition is perfectly smooth.

The use of multiple hypotheses instead of fusing all measurements in a single filter provided the motivation for the bank of filters approach. Let us explain the problem with a practical example: fusing two GNSS signal sources — a classical GPS, and a differential Real-time Kinematics (RTK) GPS. Both sources of data localize the UAV within the same coordinate system. However, each source has a different level of accuracy (the measurements can differ by several meters), and the RTK system may not be available all the time due to the physical limitations of the system. Fusing both systems into a single hypothesis creates a problem. For example, when the precise RTK data starts to be fused (possibly after some time of inaccessibility, or because the RTK system has just been activated during the flight), the hypothesis starts getting corrected. The correction step may introduce an innovation in the order of several meters, which produces state changes within the hypothesis that do not follow the model and do not respect any state constraints. More importantly, the motion of the hypothesis does not correspond to any real motion of the UAVs. This state convergence towards newly-fused measurements subsequently creates motion of the

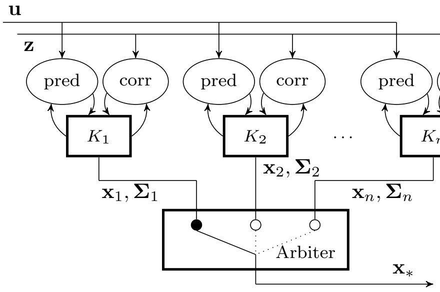

u
z
pred
corr
pred
corr
pred
corr
K_1
K_2
…
K_n
$\mathbf{x}_1, \mathbf{\Sigma}_1$
$\mathbf{x}_2, \mathbf{\Sigma}_2$
$\mathbf{x}_n, \mathbf{\Sigma}_n$
Arbiter
$\mathbf{x}_*$

Fig. 4: The bank of filters $\mathcal{K} = \{K_1, K_2, ..., K_n\}$. The filters simultaneously estimate $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$. The output hypothesis is chosen by the *arbiter*.

UAV due to the increased control error. However, this UAV motion is unplanned (and undesired), since it is not governed by feedforward action. In extreme cases, the sudden control error may even saturate the feedback controller and could endanger the UAV, as shown experimentally in Fig. 13. Any physical motion of the UAV should be produced by a desired and planned action, not by a state estimator suddenly shifting a hypothesis.

In contrast, the same situation is handled here by a bank of filters. We would consider two separate estimators, one fusing GPS and the other fusing both GPS and RTK GPS. The control pipeline can be switched on demand to use any independent hypothesis, to recalculate all its inner states from one to another, and to treat both hypotheses as independent coordinate systems. Thus when the *active* estimator is switched, the UAV does not move in a physical world, but its coordinates (and the coordinates of a control reference) will jump. It is then for the *Mission & navigation* part of the pipeline to decide whether the new coordinates within the new coordinate should be adjusted by generating a new control command. However, this multi-hypothesis structure requires the presence of an arbiter. The arbiter needs to switch the system automatically from the RTK GPS estimator to the GPS estimator when the RTK corrections become unavailable.

The multiple hypothesis system also handles scenarios where sensors do not appear within the same frame of reference, e.g., an onboard visual-based SLAM and a GPS localization system. Moreover, maintaining transformations between all the frames of reference allows us to close the feedback loop using the best estimator for control performance while generating references in other frames of reference.

4 Feedforward tracking and reference generation

A *Reference tracker* provides a feedforward control command and a reference to a *Feedback controller* within the pipeline (see Fig. 3). An input to the *Reference tracker* might be a 3D position and a heading reference $(\mathbf{r}_d, \eta_d)$, or a reference trajectory $\{(\mathbf{r}_d, \eta_d)_1, (\mathbf{r}_d, \eta_d)_2, \ldots, (\mathbf{r}_d, \eta_d)_k\}$ from the *Mission & navigation* block.

4.1 MPC Tracker for normal flight

Feedforward trajectory tracking serves a crucial role in supplying a smooth and feasible reference for feedback controllers. The control reference consists of states of the differentially-flat translational dynamics (position, velocity, acceleration, jerk) as well as the heading and the heading rate:

$$\boldsymbol{\chi} = \left[x, \dot{x}, \ddot{x}, \dddot{x}, y, \dot{y}, \ddot{y}, \dddot{y}, z, \dot{z}, \ddot{z}, \dddot{z}, \eta, \dot{\eta}\right]^{\intercal}. \tag{16}$$

Our trajectory tracking approach, originally published in [3], utilizes linear MPC for controlling a virtual UAV model in real time. The linear MPC controls the states of the virtual model (which behaves ideally). States of the virtual model are then sampled on demand, and are given to the *feedback controller* as a reference. The linear MPC produces optimal state transients in real time while satisfying state constraints. The *MPC tracker* creates a full-state reference $\boldsymbol{\chi}$ at 100 Hz either from a single 3D reference $(\mathbf{r}_d, \eta_d)$ or from a time-parametrized reference trajectory $\{(\mathbf{r}_d, \eta_d)_1, (\mathbf{r}_d, \eta_d)_2, \ldots, (\mathbf{r}_d, \eta_d)_k\}$ sampled at arbitrary sampling rate.

4.2 Take-off and landing

Take-off and landing can generally be solved by the same tracker as other situations during a routine flight. However, we separate the trajectory generation for take-off and for landing in order to increase safety. Safety concerns arise in the take-off phase, since the UAV can get entangled in ground foliage when taking off outdoors. When such a situation occurs, significant control errors can arise quickly, forcing the feedback controller into aggressive actions. We solve this with an admittance tracking mechanism, which saturates the movement of the control reference $\boldsymbol{\chi}$ beyond a set distance from the current UAV state $\mathbf{x}$. Automatic landing is performed by setting the altitude coordinate of the control reference below the estimated altitude of the UAV, which serves the same purpose and allows landing even when the altitude of the ground level is unknown.

4.3 Speed tracking for aerial swarming

Aerial swarming, which will be briefly introduced in Sec. 9.5, imposes special requirements on control reference generation. 2D swarming approaches often require classical tracking of the desired altitude and heading, but the horizontal motion may be dictated by the desired velocity or by the desired acceleration. We provide a specialized tracking approach that allows us to bypass desired states within the control reference $\boldsymbol{\chi}$, and to specify only the states that the swarming mechanism requires. The controllers, which will be described in the following section, use only the specified portion of the control reference to calculate the feedback.

5 Feedback Control

The *Feedback controller* is a crucial component within the pipeline (see Fig. 3) for controlling the flight dynamics around an unstable equilibrium point of the UAV system. The task of the controller is to minimize the control error around the desired control reference $\boldsymbol{\chi}$ (provided by the *Reference tracker* block) and to supply low-level control reference to the embedded *Attitude rate controller.* The low-level control reference produced by a controller within our pipeline consists of the desired intrinsic angular velocities of the UAV body $\boldsymbol{\omega}_d \in \mathbb{R}^3$ and the desired collective motor speed $T_d \in [0, 1]$. This section focuses on the development of two feedback control approaches. Each of the approaches serves a particular purpose in various of field experimentation scenarios. The first purpose is an extension of the *SE(3) geometric state feedback* [32]. This controller is well-suited for fast and agile maneuvers, as well as for precise control. However, both the UAV state estimate and the reference need to be continuous, smooth, and are assumed to follow the model. The second controller that we propose is a combination of a linear MPC with a nonlinear *SO(3)* force tracking feedback. This controller is designed to provide stable feedback even when the UAV state estimate is noisy or unreliable, or when state constraints need to be imposed on the control level.

As shown in Fig. 3, our architecture is a cascade-based control loop. Cascade-based control architectures are based on the singular perturbation theory [57], commonly known as the principle of time-scale separation. This approach assumes that the inner loop (in our case the attitude control) is exponentially stable and that the inner loop bandwidth is greater than the dynamics of the outer loop. So the controller of the outer loop can be designed without considering the dynamics of the inner loop. This assumption holds, since the attitude rate control loop within the PX4 firmware is executed at the utmost rate, with all new data from the embedded IMU.

5.1 *SO(3)* geometric force tracking

We base our work on the geometric tracking controller proposed in [32]. Specifically, we utilize the force tracking part of their approach. Given a desired force $\mathbf{f}_d$ to be acting on the UAV, and a desired heading vector

$$\hat{\mathbf{h}}_d = \left[\cos\eta_d, \sin\eta_d, 0\right]^\intercal, \tag{17}$$

we define a desired orientation matrix $\mathbf{R}_d$. The originally published way of constructing $\mathbf{R}_d$ is feasible; however, it does not maintain heading η during maneuvers. We therefore, we also propose a different approach explicitly designed to facilitate heading angle control.

5.1.1 Original structure of desired orientation

The matrix

$$\mathbf{R}_d^o = \left[\hat{\mathbf{p}}_{1d}, \hat{\mathbf{p}}_{2d}, \hat{\mathbf{b}}_{3d}\right], \tag{18}$$

which is composed of vectors

$$\hat{\mathbf{b}}_{3d} = \frac{\mathbf{f}_\mathbf{d}}{\|\mathbf{f}_\mathbf{d}\|}, \quad \hat{\mathbf{p}}_{2d} = \frac{\hat{\mathbf{b}}_{3d} \times \hat{\mathbf{h}}_d}{\|\hat{\mathbf{b}}_{3d} \times \hat{\mathbf{h}}_d\|}, \quad \hat{\mathbf{p}}_{1d} = \hat{\mathbf{p}}_{2d} \times \hat{\mathbf{b}}_{3d}, \tag{19}$$

maintains the desired force vector as the direction of the $\hat{\mathbf{b}}_{3d}$ axis, and finds $\hat{\mathbf{p}}_{1d}$ as the orthogonal projection of $\hat{\mathbf{h}}_d$ to the subspace

$$\mathcal{L}_d = span\left(\hat{\mathbf{b}}_{3d}\right)^\perp. \tag{20}$$

However, the heading is not preserved in this case (the azimuth of the $\hat{\mathbf{p}}_{1d}$ axis is not generally equal to the azimuth of $\hat{\mathbf{h}}_d$).

5.1.2 Heading-compliant desired orientation

We tackle the heading control by constructing the desired orientation matrix as

$$\mathbf{R}_d = \left[\hat{\mathbf{b}}_{1d}, \hat{\mathbf{b}}_{2d}, \hat{\mathbf{b}}_{3d}\right], \tag{21}$$

by finding $\hat{\mathbf{b}}_{1d}$ as an oblique projection of $\hat{\mathbf{h}}_d$ in the direction of $\hat{\mathbf{e}}_3$ to the subspace $\mathcal{L}_d$. This projection is constructed as

$$\mathbf{b}_{1d} = \mathbf{O}(\mathbf{P}^\intercal\mathbf{O})^{-1}\mathbf{P}^\intercal, \ \hat{\mathbf{b}}_{1d} = \frac{\mathbf{b}_{1d}}{\|\mathbf{b}_{1d}\|}, \tag{22}$$

where $\mathbf{O} \in \mathbb{R}^{3\times2}$ is the orthogonal projector to $\mathcal{L}_d$ (constructed, e.g., as the first two columns of $\mathbf{I} - \hat{\mathbf{b}}_{3d}\hat{\mathbf{b}}_{3d}^\intercal$), and $\mathbf{P} = [\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2]$ is the orthogonal basis of the world xy-plane. The $\hat{\mathbf{b}}_{2d}$ axis, is at last constructed as

$$\hat{\mathbf{b}}_{2d} = \hat{\mathbf{b}}_{3d} \times \hat{\mathbf{b}}_{1d}. \tag{23}$$

Both $\mathbf{R}_d$ and $\mathbf{R}_d^o$ can be obtained under the assumption that $\hat{\mathbf{b}}_{3d} \nparallel \hat{\mathbf{h}}_d$. Both options are valid, and their selection should be carefully considered. We prefer the $\mathbf{R}_d^o$ option, due to the consistency of the resulting reference with the heading feedforward control described further in Sec. 5.2, and 5.3.

Given $\mathbf{R}_d$, we express the rotation error according to [32] as

$$\mathbf{e}_R = \frac{1}{2}\left(\mathbf{R}_d^\intercal \mathbf{R} - \mathbf{R}^\intercal \mathbf{R}_d\right). \tag{24}$$

Finally, the desired angular rate is obtained as

$$\boldsymbol{\omega}_d = -\mathbf{k}_R \circ \mathbf{e}_R + \boldsymbol{\omega}_j - \boldsymbol{\omega}_c, \tag{25}$$

where $\mathbf{k}_R$ are the rotation control gains, $\boldsymbol{\omega}_j$ is the feedforward attitude rate caused by the desired jerk $\dddot{\mathbf{r}}_d$, and $\boldsymbol{\omega}_c$ is the *parasitic* heading rate compensation described further in Sec. 5.3. The feedforward attitude rate is constructed as

$$\boldsymbol{\omega}_j = \frac{\|\mathbf{f}_d\|}{m_e} \left[\hat{\mathbf{e}}_3\right]_\times^\intercal \mathbf{R}_d^\intercal \dddot{\mathbf{r}}_d, \tag{26}$$

where $\|\mathbf{f}_d\|/m_e \left[\mathrm{m\,s^{-2}}\right]$ is the effective thrust, m_e [kg] is the estimated mass of the vehicle, $\left[\hat{\mathbf{e}}_3\right]_\times$ is the cross-product matrix satisfying the condition $\left[\hat{\mathbf{e}}_3\right]_\times \mathbf{v} = \hat{\mathbf{e}}_3 \times \mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^3$. The final control output is the desired attitude rate $\boldsymbol{\omega}_d$ and the desired thrust force $f_d = \mathbf{f}_d^\intercal \hat{\mathbf{b}}_3$.

5.2 Applying the reference heading rate as the feedforward yaw rate

As mentioned in Sec. 4, the reference trackers output the heading η and its derivative $\dot{\eta}$ as a reference. Using the heading rate for feedforward in (25) requires converting it to the desired yaw rate ω_{3d} (the yaw rate is the 4th independently-controllable intrinsic degree-of-freedom (DOF), which does not influence the translational dynamics). First, we define the derivative of heading vector $\mathbf{h}$ as

$$\dot{\mathbf{h}} = [0, 0, \dot{\eta}]^\intercal \times \mathbf{h}. \tag{27}$$

Then we define the orthogonal projector $\mathbf{P}$ on the linear subspace spanned by $\dot{\mathbf{h}}$. However, it is vital to define $\mathbf{P}$ even when $\dot{\mathbf{h}} = \mathbf{0}$. One option is:

$$\mathbf{P} = \left(\hat{\mathbf{e}}_3 \times \hat{\mathbf{h}}\right)\left(\hat{\mathbf{e}}_3 \times \hat{\mathbf{h}}\right)^\intercal. \tag{28}$$

Then we project the orthogonal basis of the subspace spanned by the derivative of $\hat{\mathbf{b}}_1$, which is consequently $\hat{\mathbf{b}}_2$, on the subspace spanned by $\dot{\mathbf{h}}$:

$$\mathbf{p} = \mathbf{P}\hat{\mathbf{b}}_2. \tag{29}$$

Now we find a scaling factor k between $\dot{\mathbf{h}}$ and $\mathbf{p}$

$$k = \operatorname{sign}\left(\dot{\mathbf{h}}^\intercal \mathbf{p}\right) \frac{\|\dot{\mathbf{h}}\|}{\|\mathbf{p}\|}, \text{ for } \|\mathbf{p}\| \neq 0, \tag{30}$$

which is applied to recreate the desired derivative of $\hat{\mathbf{b}}_1$ as $k\hat{\mathbf{b}}_2$. Thus, the angular velocity around $\hat{\mathbf{b}}_3$ is

$$\omega_{3d} = k. \tag{31}$$

5.3 Compensating for the parasitic heading rate

The desired angular rate $\boldsymbol{\omega}_d$ obtained from the force tracking approach, can influence the resulting heading rate $\dot{\eta}$. This can easily be observed while flying a dynamic trajectory with a constant desired heading. The control law (25) inevitably creates angular velocities around $\hat{\mathbf{b}}_1$ and $\hat{\mathbf{b}}_2$ that are being reflected in $\dot{\eta}$. These *disturbances* will be counteracted by the feedback. However, feedback corrections are made after a control error has occurred, and this makes them appear too late during aggressive maneuvers. We compensate for them in advance by calculating the *parasitic* heading rate created by the $\hat{\mathbf{b}}_1$ and $\hat{\mathbf{b}}_2$ rotations, similarly as in Sec. 3.4. In addition, as in Sec. 5.2, the heading rate is converted to the intrinsic yaw rate, ω_{3c}, (the angular velocity around $\hat{\mathbf{b}}_3$), which is then added back to (25) as $\boldsymbol{\omega}_c = [0, 0, \omega_{3c}]^\intercal$.

5.4 Converting the desired thrust force to thrust

The motor speed is often controlled by dedicated modules, i.e. by ESCs. The input to an ESC is typically a desired motor speed scaled linearly between $[0, 1]$, which represents the range from the minimum speed to the maximum speed. The desired thrust force $f_d = \|\mathbf{f}_d\|$ therefore needs to be converted to the output collective thrust $T_d \in [0, 1]$. The simplest but still effective thrust model relies on the approximate relationship between the produced force f and the angular rate ω of a motor: $f \propto \omega^2$. We therefore model the thrust as

$$T_d = a_t \sqrt{f_d} + b_t, \tag{32}$$

where a_t and b_t are parameters of a quadratic thrust curve. The parameters are obtained empirically by the least-square fit on experimentally obtained data — tuples of the thrust and the mass (T_h, m) — using equation (32) as a hover thrust curve

$$T_h = a_t\sqrt{mg} + b_t. \tag{33}$$

The accuracy of the thrust model is important for the correct calculation of the applied thrust, which has an influence on the overall control performance. The inversion of (33)

$$m_a = \frac{1}{g}\left(\frac{T_d - b_t}{a_t}\right)^2 \tag{34}$$

is also used to deduce the current apparent mass m_a based on the currently-used thrust output T_d, namely during landing for automatic touchdown detection.

5.5 Disturbance estimation

Various effects can cause a steady state control error in the position of the UAV: $\mathbf{e}_p = \mathbf{r} - \mathbf{r}_d$. External disturbances that appear to be fixed within the world frame (e.g. wind) occur together with disturbances that are tied to the body frame of the UAV, e.g. air drag. Also, a miscalibrated artificial horizon (accelerometer bias) will generally cause control errors, which can be observed and estimated as a steady-state body disturbance. UAV disturbance estimation is a well-established field with a prominent use of non-linear state estimators (often EKF or UKF) [24, 11, 9, 2] with a linear estimator being also suitable, as shown in our prior work [5]. It is common to account for the disturbances as dedicated observable states within the UAV model and use an observer to estimate them. In this work, we strafe from the typical disturbance estimation process within the main state estimator of the UAV. Estimating both the world and the body disturbances simultaneously would require a non-linear state estimator (to separate them within the model [2]). Since we utilize the simpler LKF, we propose the following disturbance estimation approach based on the body- and world-frame integrators. The disturbance estimator within our system is part of the controllers, rather than the UAV state estimator, which opens the possibility to use any UAV state estimator that does not support disturbance estimation or even to fly without a state estimator completely, e.g., using a precise motion capture system. Although our disturbance estimator is arguably simple, it does not require the UAV model. The estimator does not need to know the UAV mass; even more, it estimates the mass disturbance. It can be executed regardless of the type of used UAV estimator.

We continually estimate the world force disturbance $\mathbf{d}_w$ [N] and the body force disturbance $\mathbf{d}_b$ [N] simultaneously during the flight as

$$\begin{aligned} \mathbf{d}_w &= \sum_{n=0}^{N} \mathbf{k}_{iw} \circ \mathbf{e}_{p[n]}\, \mathit{\Delta} t_{[n]}, \\ \mathbf{d}_b &= \mathbf{H}_{[N]} \sum_{n=0}^{N} \mathbf{k}_{ib} \circ (\mathbf{H}_{[n]}^{\intercal}\, \mathbf{e}_{p[n]})\, \mathit{\Delta} t_{[n]}, \end{aligned} \tag{35}$$

where

$$\mathbf{H}_{[n]} = \begin{bmatrix} \cos\ \eta_{[n]} & -\sin\ \eta_{[n]} & 0 \\ \sin\ \eta_{[n]} & \cos\ \eta_{[n]} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{36}$$

is the heading rotation matrix at sample n, $\mathbf{k}_{iw}$ is the world integral gain, and $\mathbf{k}_{ib}$ is the body integral gain. Until the UAV changes its heading, all estimated disturbances are equally split in both $\mathbf{d}_w$ and $\mathbf{d}_b$. The physical interpretation of the x-axis and y-axis components is the force [N] after we compensate for them by the feedback in the desired force $\mathbf{f}_d$, as described in Sec. 5.6 and Sec. 5.7.

Another undesired effect is the apparent change in the mass of the UAV that can be deduced from the applied thrust. This can indeed be a change in the mass of the UAV, e.g., due to deploying the payload or gathering objects, or it can be an apparent change caused by a discharge of the battery, contact of a horizontal surface, and real-time changes in the efficiency of the propulsion system. Either way, we estimated the apparent mass change by using the z-axis disturbance as a part of a total estimated mass of the UAV

$$m_e = m + (\mathbf{d}_w + \mathbf{H}\mathbf{d}_b)^{\intercal}\, \hat{\mathbf{e}}_3, \tag{37}$$

where m stands for the nominal mass obtained by weighting the UAV. The physical interpretation of $(\mathbf{d}_w + \mathbf{H}\mathbf{d}_b)^{\intercal}\, \hat{\mathbf{e}}_3$ is the mass difference [kg] from the nominal take-off mass, thanks, again, to the total estimated mass m_e being used in the control feedback loop.

5.6 *SE(3)* state feedback

The first of our controller variants is the *agile* controller option. It is based upon the *SE(3) geometric tracking feedback* [32] with the addition of disturbance compensation. To supplement the force tracking from section 5.1, we define the desired force as

$$\mathbf{f}_d = \overbrace{-m_e\mathbf{k}_p \circ \mathbf{e}_p}^{\substack{\text{position}\\ \text{feedback}}} + \overbrace{-m_e\mathbf{k}_v \circ \mathbf{e}_v}^{\substack{\text{velocity}\\ \text{feedback}}} + \overbrace{m_e\ddot{\mathbf{r}}_d}^{\substack{\text{reference}\\ \text{feedforward}}} + \underbrace{-m_e g\hat{\mathbf{e}}_3}_{\substack{\text{gravity}\\ \text{compensation}}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\substack{\text{world disturbance}\\ \text{compensation}}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\substack{\text{body disturbance}\\ \text{compensation}}}, \tag{38}$$

where $\mathbf{d}_w$, $\mathbf{d}_b$ $\left[\mathrm{m\,s^{-2}}\right]$ are the world and body disturbance force terms, m_e [kg] is the estimated UAV mass, $\ddot{\mathbf{r}}_d$ $\left[\mathrm{m\,s^{-2}}\right]$ is the desired acceleration, g $\left[\mathrm{m\,s^{-2}}\right]$ is the magnitude of the gravitational acceleration, $\mathbf{k}_p$ are the position gains, $\mathbf{k}_v$ are the velocity gains, and $\mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_d$ is the velocity control error. The z-axis component of the disturbances is eliminated by the element-wise product, as it is already compensated for in the form of the estimated mass m_e.

5.7 Model Predictive Control Force Feedback

This controller uses a linear MPC approach to generate a desired acceleration $\mathbf{c}_d \in \mathbb{R}^3$. The acceleration is used while calculating the desired force, similarly to the previous case:

$$\mathbf{f}_d = \overbrace{m_e\ddot{\mathbf{r}}_d}^{\substack{\text{reference}\\ \text{feedforward}}} + \overbrace{m_e\mathbf{c}_d}^{\substack{\text{MPC}\\ \text{feedforward}}} + \overbrace{m_e g\hat{\mathbf{e}}_3}^{\substack{\text{gravity}\\ \text{compensation}}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\substack{\text{world disturbance}\\ \text{compensation}}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\substack{\text{body disturbance}\\ \text{compensation}}}. \tag{39}$$

Linear MPC is a robust feedback method for a system with a known model. In this case, the MPC controller is formulated such that the control input of its model is the acceleration of the point-mass translation dynamics. Thus, the control input is directly used as $\mathbf{c}_d$. Moreover, the MPC approach naturally solves the control problem optimally subject to given state and input constraints. This ensures the feasibility and the smoothness of the acceleration command, bound to satisfy maximum velocity, acceleration, and jerk.

5.7.1 MPC Model

The MPC controller operates with an LTI model, similar to the model used for estimation. However, the heading is still controlled via the *SO(3)* feedback, so there is no need to include it here. For MPC we consider the following state vector:

$$\mathbf{x}_m = \left[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}\right]^{\intercal}. \tag{40}$$

The model matrices are defined as

$$\mathbf{A}_m(\Delta t) = \begin{bmatrix} \mathbf{A}_r(\Delta t,0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_r(\Delta t,0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_r(\Delta t,0) \end{bmatrix}, \tag{41}$$

$$\mathbf{B}_m(\Delta t) = \begin{bmatrix} \mathbf{B}_r(\Delta t,1) \\ \mathbf{B}_r(\Delta t,1) \\ \mathbf{B}_r(\Delta t,1) \end{bmatrix}, \tag{42}$$

where $\mathbf{A}_r$ and $\mathbf{B}_r$ are the same subsystem matrices (10) as in the estimator model, with $\Delta t := 0.05\,\mathrm{s}$. However, this time we use the free parameters of the model to apply the system input directly without delay to the acceleration state ($p_1 = p_2 = 0$).

5.7.2 MPC controller

An MPC control error is defined along a prediction horizon of length n as

$$\mathbf{e}_{[i]} = \mathbf{x}_{m[i]} - \mathbf{x}_{md[i]}, \forall i \in \{1, \dots, n\}, \tag{43}$$

where $\mathbf{x}_{m[i]}$ is a state vector and $\mathbf{x}_{md[i]}$ is a reference at sample i of the prediction. The reference state takes the form of

$$\mathbf{x}_{md[i]} = \left[x_d, 0, 0, y_d, 0, 0, z_d, 0, 0\right]^{\intercal}, \forall i \in \{1, \dots n\}. \tag{44}$$

The initial condition $\mathbf{x}_{m[0]}$ is commonly set to values of the current state estimate. However, to make the system more stable even when estimated states violate dynamic constraints, position derivatives can be substituted with states of reference vector $\boldsymbol{\chi}$ from a feedforward tracker:

$$\mathbf{x}_{m[0]} = \begin{cases} \left[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}\right]^{\intercal}, \text{if constraints satisfied,} \\ \left[x, \dot{x}_d, \ddot{x}_d, y, \dot{y}_d, \ddot{y}_d, z, \dot{z}_d, \ddot{z}_d\right]^{\intercal}, \text{if violated.} \end{cases} \tag{45}$$

The MPC is formulated as a quadratic programming problem

$$\min_{\mathbf{u}_{[1:n]}} \quad \frac{1}{2}\sum_{i=1}^{n-1}\left(\mathbf{e}_{[i]}^{\intercal}\mathbf{Q}\mathbf{e}_{[i]}\right) + \mathbf{e}_{[n]}^{\intercal}\mathbf{S}\mathbf{e}_{[n]} \tag{46}$$

$$\text{s.t. } \mathbf{x}_{m[i]} = \mathbf{A}_m \mathbf{x}_{m[i-1]} + \mathbf{B}_m \mathbf{u}_{[i]}, \quad \forall i \in \{1, \ldots, n\} \tag{47}$$

$$\mathbf{x}_{m[i]} \leq \mathbf{x}_{\max}, \quad \forall i \in \{1, \ldots, n\} \tag{48}$$

$$\mathbf{x}_{m[i]} \geq -\mathbf{x}_{\max}, \quad \forall i \in \{1, \ldots, n\} \tag{49}$$

$$\mathbf{u}_{[i]} - \mathbf{u}_{[i-1]} \leq \dot{\mathbf{u}}_{\max} \Delta t, \quad \forall i \in \{2, \ldots, n\} \tag{50}$$

$$\mathbf{u}_{[i]} - \mathbf{u}_{[i-1]} \geq -\dot{\mathbf{u}}_{\max} \Delta t, \quad \forall i \in \{2, \ldots, n\} \tag{51}$$

where the minimized quadratic cost is the sum of the squares of the control errors over the prediction horizon of length $n \in \mathbb{Z}^+$. $\mathbf{Q} \succcurlyeq 0$ is the state error penalization matrix and $\mathbf{S} \succcurlyeq 0$ is the last state error penalization matrix. Constraint (47) forces the states to follow model (41)–(42), while (48)–(49) bound the states of the dynamical system, and (50)–(51) limit the input slew rate, i.e., the system jerk. Note that we do not penalize the input action within the cost function. No penalty is necessary, because the slew rate directly limits the jerk. With the MPC problem solved in every iteration of the control loop (at 100 Hz), the acceleration reference $\mathbf{c}_d$ is extracted directly from $\mathbf{u}_{[1]}$, i.e., the first control input of the MPC.

The particular values of $\mathbf{Q}$ and $\mathbf{S}$ were found empirically as

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}(500, 100, 100, 500, 100, 100, 100, 10, 10), \\ \mathbf{S} &= \mathrm{diag}(1000, 300, 300, 1000, 300, 300, 100, 10, 10). \end{aligned} \tag{52}$$

These values were extensively tested on a variety of platforms, ranging from 1.5 kg, ≈ 0.5 m DJI f450, to 15 kg, ≈ 1.2 m Tarot t18. We rely on this controller even for solving *emergency* situations when the *SE(3)* controller fails, since the MPC is designed to be more stable with respect to sensor noise and is designed to intrinsically satisfy state constraints. The choice of the constraints $\mathbf{x}_{max}$ and $\mathbf{u}_{max}$ depends on the particular application scenario. Most of the time, we allow the controller to reach speeds up to $2\,\mathrm{m\,s^{-1}}$ with acceleration of $2\,\mathrm{m\,s^{-2}}$ and jerk $5\,\mathrm{m\,s^{-3}}$. However, to make the flight safe, the controller also overrides state constraints of feedforward trackers, to ensure that they are at most half the value for the controller.

5.8 Unbiased desired acceleration

The unbiased desired acceleration is created by subtracting the estimated disturbances from the desired force created by controllers, applied to the current body orientation:

$$\mathbf{a}_d = \frac{f_d \hat{\mathbf{b}}_3 - g\hat{\mathbf{e}}_3 + \mathbf{d}_w \circ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \mathbf{d}_b \circ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}{m_e}. \tag{53}$$

The acceleration $\mathbf{a}_d$ then has a zero DC component, although nonzero tilt is produced, e.g., in order to compensate for wind and for a mass disturbance. Both can be achieved by dividing the compensated force by the estimated mass m_e in the denominator.

5.9 Take-off and landing

Take-off and landing can be executed using both of the proposed controllers without special modification, as in the case with the reference generation (see Sec. 4.2). The *SE(3) controller* is preferred when high control accuracy is required, but only if the localization system provides a smooth enough state estimate. As later shown experimentally in Sec. 8.3, the *MPC controller* provides much better estimator noise suppression, which is desirable during take-off and landing. In general, the *MPC controller* is the default choice within our pipeline.

5.10 Feedforward failsafe controller

Position feedback control cannot be executed when a localization system is lost in mid-flight. If velocity odometry is present, e.g., in the form of an optical flow system, the active state estimator can be switched, and an emergency landing can be executed. However, the system cannot continue with flight without any position and velocity state estimate. In such a situation, we employ a feedforward failsafe landing, which relies on the attitude controller within the installed embedded flight controller (Pixhawk). The *failsafe controller* outputs the desired orientation to keep the UAV leveled and to maintain the desired thrust to cause moderate uncontrolled descent. The initial thrust is calculated using the hover-thrust curve (33) with the last known estimated mass m_e. Then the thrust is decreased by a fixed rate to cause the UAV to descend. This procedure stops the UAV from accelerating in any direction. When such an emergency occurs during aggressive maneuvers, it is up to a safety pilot to recognize that there is a problem and to regain control using a remote controller. However, if this type of situation occurs during a slow indoor flight at low altitude, the UAV typically safely reaches the ground before any damage can occur.

6 Trajectory generation

Generating a reference trajectory from a set of waypoints in real time is an essential step for some UAV applications. A commonly-adopted approach relies on optimizing a parametrized polynomial function that is later sampled [37] to obtain a trajectory $\{(\mathbf{r}_d, \eta_d)_1, (\mathbf{r}_d, \eta_d)_2, \ldots, (\mathbf{r}_d, \eta_d)_k\}$. The approach in [37] was later extended in [55, 10] by solving a non-linear optimization problem to solve both for the geometry of the path and time sampling within one problem. We extend the approach in [55] even further. Although the approach generates trajectories that do not violate state constraints, they also do not minimize the total flight time, sometimes resulting in prolonged flight. We updated the constraint satisfaction mechanism in [55] by rescaling only the generated trajectory segments that directly violate the constraints instead of rescaling the whole trajectory. We also provide lower-bound initial segment time estimates for each segment. Again, this modification improves the overall flight time of the path since the constraint satisfaction mechanism only prolongs the segment times if necessary but does not shorten them if possible. Furthermore, we provide an iterative sub-sectioning mechanism that automatically satisfies the given maximum distance of the generated trajectory from the provided waypoint path. Our other improvements to the method and software provided by [55] can be found at our GitHub page[20] with the rest of our software.

7 Implementation

Implementation aspects has not often been an integral part of published reports on control-oriented research. However, we argue that there is a need for a system-oriented manuscript that includes software and sources. In this section, we will discuss implementation and software design considerations of our system[21] that have been shaped by the requirements of real-world deployment. Real-world deployment and verification of novel UAV methods often require a specific platform configuration for the method being verified. The proposed system is designed to be extensively modular, allowing *hot-swapping* of feedback controllers, trajectory trackers, state estimators, controller gains, and dynamic constraints. These can be changed in mid-flight so that new methods can benefit from existing and tested systems, for safely managing the initial take-off and landing, or for regaining control in the event of unwanted behavior of the tested methods. It is very useful to have the option to switch to a reliable backup system is when testing new real-time software. The proposed system is built on ROS[22], and is available as open-source with all the components described within this section. We have striven to provide a well-documented system to allow researchers and students to flatten the initial learning curve and to focus on their particular research instead of developing yet another control pipeline. This approach has been shown to be effective, as demonstrated in Sec. 9 on various examples of real-world use and deployment of the platform. Figure 5 is an implementation diagram of the system with its modules, which will be presented in the following sections.

7.1 State Estimator

The state estimator (see Sec. 3) was designed to provide multi-frame localization. Unlike a generally accepted approach to fuse all available sensory inputs into a single hypothesis, we execute a bank of estimators, each for a subset of inputs. If, for example, a UAV is provided with data from a GPS receiver (with a magnetometer), a 2D SLAM and an optic flow algorithm (velocity relative to the ground plane), we may consider executing the following estimators simultaneously: GPS, 2D SLAM, 2D SLAM & optic flow, optic flow. At any time, all hypotheses are available, and the UAV is simultaneously localized within multiple independent world coordinate systems. One estimator (the coordinate system) is always selected as the *primary estimator*, which is used for feedback control at the moment. The primary estimator can be switched in mid-flight on-demand or automatically when its hypothesis is deemed *unreliable*. Transformations between the coordinate systems are maintained (using the ROS Transformation library[23]), which allows a seamless definition of references in any of the existing frames of reference. This significantly increases the overall stability of the system. For example, the feedback loop can be closed using an optic flow odometry estimator when the GPS signal is too inaccurate for feedback control. However, control references can still be given in the GPS coordinate frame, without the need to change the *mission & navigation* software. Frequent switching of frames of references can occur, e.g., when manipulating with the environment using local sensor information. In the 2020 MBZIRC competition, we employed frequent switching between onboard visual servoing and global GNSS localization. The UAV was attempting to grasp a brick autonomously while

[20] github.com/ctu-mrs/mrs_uav_trajectory_generation
[21] github.com/ctu-mrs/mrs_uav_system
[22] Robot Operating System, `http://ros.org`
[23] ROS tf2, `http://wiki.ros.org/tf2`

being localized relative to the object of interest and transitioning between pickup and place locations using GNSS localization.

7.2 Control Manager

As demonstrated by the system architecture diagram in Fig. 3, the two most important parts of the control system are the feedforward *reference tracker* and the feedback *reference controllers*; for brevity *trackers* and *controllers*. Implementation-wise, various methods are used for both components, in addition to the methods presented in Sec. 4 and in Sec. 5. We employ multiple trackers to fulfill different roles during the flight, and being able to switch between each of these roles is a major software design factor within the system. Trackers and controllers are implemented as ROS plugins (using the *ROS Pluginlib library*), which makes them follow an interface pre-defined by a central *plugin manager*, called the *Control Manager*. The controller and tracker interfaces were designed to keep tracker and controller implementation minimalistic, while the *Control manager* is responsible for

- loading a defined set of trackers and controllers,
- gathering estimator data,
- synchronizing the active tracker and controller,
- providing all trackers and controllers with current dynamic constraints,
- providing a unified interface for setting desired trajectories and references,
- providing an Application Programming Interface (API) to the common libraries used throughout the plugins,
- outputting the desired attitude rate and thrust command.

Moreover, all the incoming references and desired trajectories are transformed into the current control frame before being given to the active tracker. When the current control frame changes (when the active estimator is switched), all the loaded controllers and trackers are synchronously prompted to transform their internal state from the old frame to the new frame. When a controller or a tracker is switched, the newly activated plugin is given the last state and result of the previously-active plugin, making the transitions safe and imperceptible. Additionally, the *Control manager* facilitates routines for

- handling excessive control errors using emergency and feedforward failsafe landing,
- the virtual allowed safety area with no-fly zones,
- the virtual reactive obstacle bumper,
- the control bindings to an RC controller (via Mavros[24]).

The system is designed with emphasis on simplifying the development and testing of new trackers/controllers and on developing new trackers and controllers for use in particular specialized applications. Thanks to the plugin architecture, a custom tracker and a controller can be deployed with no software changes to the proposed platform (except for customization of the ROS *launch* and *config* files). This helps users to keep the core unchanged and therefore updated and it simplifies customization for a particular project and application, even when a single UAV is shared by multiple users and projects.

7.3 Reference controllers

The feedback controllers, which are described in Sec. 5, form part of a bank of controllers loaded by the *Control manager*. The *SE(3) controller* (Sec. 5.6) takes on the role of an agile and *fast* controller that is capable of executing aggressive maneuvers with accelerations approaching $10\,\mathrm{m\,s^{-2}}$. The *MPC controller* (Sec. 5.7) is almost immune against estimation noise and disturbances, and also against reference infeasibilities. Furthermore, we utilize a *Failsafe controller*, which provides feed-forward action in situations when feedback is not computable.

7.4 Reference trackers

The trackers are the reference generators for the controllers. Although we use the *MPC tracker* [3] most of the time, there are scenarios where different approaches are required. We intentionally separated landing and take-off reference generation to another tracker, called the *Landoff tracker*. Landing and take-off do not usually require fast maneuvers, agility, or tracking of complex trajectories. In contrast, admittance tracking is used to mitigate excessive control errors due to the UAV being trapped on the ground by an unwanted mechanical attachment during take-off. In addition, research on UAV swarming [63, 65] often requires more direct access to the desired states of the UAV. For those situations, we provide the *Speed tracker*, which allows direct control of the desired speed and/or acceleration of the UAV, while maintaining the desired height and heading. In the *Speed tracker*, we only constrain the first derivative of given references by a low-pass filter, which gives

[24] Mavros, a ROS interface to the Mavlink protocol and thus to the Pixhawk flight controller github.com/mavlink/mavros.

users more hands-on control over the hardware while still maintaining safety.

7.5 Gain & Constraint Management

Dynamic constraints are supplied and managed globally for all trackers and controllers by the *Constraint manager*. Pre-defined groups of constraints are loaded during each software startup, allowing users to switch between them in mid-flight. The following dynamic constraints are considered within one group: speed, acceleration, jerk, and snap for horizontal translation, and for vertical ascending and for vertical descending translation. For rotations, we consider heading speed, acceleration, jerk and snap, and the intrinsic roll, pitch, and yaw rates. The group can be designated with a name (e.g., *slow*, *medium*, *fast*) and can be assigned to a matrix of allowed constraints for each type of estimator. A fallback option (a default constraint group) is also defined for each state estimator type. When the estimator type is switched during a flight, the fallback constraint group is switched automatically, if the group is missing within the allowed constraints. The *Mission & navigation* software can switch the constraint groups on demand, but only if they are within the list of allowed constraints. The *Constraint manager* transfers the particular constraint values to the *Control manager*, which distributes the values to all loaded trackers and controllers.

A similar mechanism is employed to manage the *SE(3) controller* gains, since the gains depend on the particular application and on the type of sensor fusion that is used. Again, groups of gains are defined (e.g., *soft*, *medium*, *tight*) and are assigned to the estimator types. This mechanism is necessary, especially when the estimators that are used vary significantly in their noise parameters and therefore require different gains to make a flight possible.

7.6 UAV Manager

The *UAV manager* implements essential high-level state machines for take-off and landing. Both take-off and landing routines can use a specified tracker and controller. The selected tracker and controller are also automatically activated after take-off. The user or the *Mission & navigation* software may issue an instruction to land immediately, or after returning to the last take-off location, or after flying to particular coordinates.

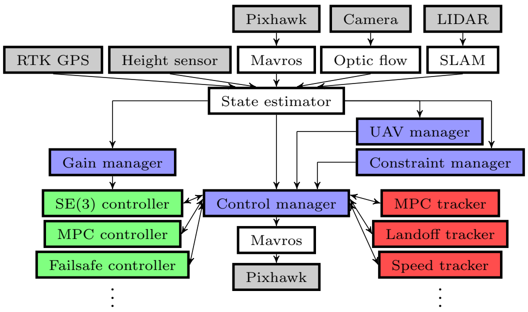


Fig. 5: An illustration of the implementation diagram of the proposed UAV system. Onboard sensors and actuator modules are depicted as grey blocks. The sensor combination varies depending on the particular UAV task. White blocks represent ROS components responsible for managing sensors or for interacting with the actuators (Mavros). Green blocks stand for feedback controllers (see Sec. 7.3) and red blocks stand for reference trackers (see Sec. 7.4). Purple blocks represent high-level components that provide the controllers and trackers with up-to-date data and maintain the synchronicity of the events. These include controller, tracker and estimator switching, gain and constraint scheduling, and take-off and landing.

7.7 Mission & navigation software

In a typical scenario, the UAV control pipeline is commanded by a user directly, using a remote terminal, or by onboard mission control software. Typically, both scenarios include supplying the control pipeline with desired references, trajectories, switching between constraints, trackers, and controllers. Although this element of the system is essential in practical applications, it is highly application-specific and it is independent of the core control pipeline. For examples of practical applications of the proposed control pipeline, including references to relevant perception, planning, and mission control algorithms, see Sec. 9.

7.8 Simulation environment

The simulation software is a crucial tool for robotic research. For this purpose, we have developed our simulation environment, which is also made publicly available[25]. It makes use of the open-source Gazebo simulator and it is set up for multiple different variants of our hardware UAV platforms (DJI f450, DJI f550, Tarot 650 sport). It can also easily be extended to a new hardware setup. All UAV hardware elements, including the Pixhawk flight controller, the actuators, and various sensors are simulated with high fidelity, so there is

[25] Simulation, `http://github.com/ctu-mrs/simulation`

only a minimal difference between simulated flight and real-world flight when using the proposed UAV system. This ensures a smooth transition between simulation and reality, which significantly accelerates the deployment of new robotic methods and algorithms.

8 Experimental evaluation

We performed a series of experiments to demonstrate the control and tracking performance of the proposed system in various conditions. All experiments were carried out in the real world as and also in the proposed simulator environment using the Tarot 650 platform (see Fig. 6). Figure 11 shows photos from the experiments, as described in the following sections. As shown in the comparative figures within this section, the dynamics system behaves almost identically in simulation as well as in real world. Importantly, the conducted maneuvers were near to the physical limits of the tested UAV, particularly its maximum thrust.

8.1 Agile tracking of step position references

A step reference with increasing size in the desired position was supplied to the reference tracker. Figure 7 shows the position, the velocity and the acceleration of the UAV under a series of step references in a single axis. Figure 8 shows the position of the UAV when commanded with a 3D step reference. Both situations demonstrate precise and agile control near the limits of the physical capabilities of the UAV given the state constraints: $\dot{x}_{max} = 9\,\mathrm{m\,s^{-1}}$, $\ddot{x}_{max} = 12\,\mathrm{m\,s^{-2}}$, $\dddot{x}_{max} = 50\,\mathrm{m\,s^{-3}}$, $\ddddot{x}_{max} = 50\,\mathrm{m\,s^{-4}}$.

8.2 Circular trajectory

Tracking a circular trajectory is a challenging task due to the ever-changing acceleration of the vehicle. Figures 9 and 10 show the x, y position, and the heading η of the UAV while tracking a horizontal trajectory with a radius of 5 m and a speed of $7\,\mathrm{m\,s^{-1}}$. The UAV produced centripetal acceleration close to $10\,\mathrm{m\,s^{-2}}$ to maintain the circular motion. Figure 9 shows a trajectory with the heading pointing towards the center of the circle. This is the simplest scenario, for several reasons. The air drag acts on the vehicle from the same direction throughout the flight, enabling an estimate to be made using the proposed body disturbance estimator. In addition, this situation does not create any parasitic heading rate, and the desired heading rate is completely satisfied with just the ω_1 and ω_2 angular velocities. On the other hand, Fig. 10 shows a circular trajectory with a constant heading in the world. This is a challenging trajectory to follow, since the air drag cannot be estimated using the proposed pipeline, and the motion requires continuous action using the angular velocity ω_3 to produce the feedforward heading rate motion and to compensate the parasitic heading rate. However, despite these difficulties, the *SE(3) controller* is able to track trajectories of this type with an average position error of 0.5 m, and 0.1 m for the first case. As with the step references, these circular trajectories are near the physical limits of the tested UAV.

8.3 Estimator noise suppression

The proposed *MPC controller* provides stabilization and control even with a noisy state estimate. It is vital to deploy this type of control scheme in scenarios where the localization system may produce noisy measurements. Tuning a state estimator to smooth out the noise in the estimated states is not always an option, as it can increase the transfer delay of the estimator to such an extent that the estimator can make the closed loop unstable. We therefore, we prefer to use a controller that is resistant to excessive noise in the estimated states. Figure 12 shows a simulation of the stabilization properties of the *MPC controller* and the *SE(3) controller*, when the estimated position and velocity are increasingly noisy. The performance of the *MPC controller* allows the flight to continue even after a significant noise is present, whereas the *SE(3) controller* would possibly lead to a premature uncontrolled landing due to excessive control actions leading to a loss of onboard localization systems.

8.4 Estimator position jump handling

As in the case of high estimator noise, the *MPC controller* outperforms the *SE(3) controller* in terms of resistance to state estimator infeasibilities. Jumps in the estimated positions are common problem with onboard SLAMs. Similarly, jumps in the control reference may occur when developing and testing new trajectory tracking approaches. Figure 13 shows a feedback reaction of both controllers to a 5 m jump in the estimated position. The *MPC controller* minimizes the control error smoothly while satisfying its internal state constraints ($2\,\mathrm{m\,s^{-1}}$ speed, $2\,\mathrm{m\,s^{-2}}$ acceleration) and producing mild control actions. The *SE(3) controller* also stabilizes the UAV. However, the controlled states are unbounded, leading to excessive tilts and again possibly to the loss of onboard localization systems.

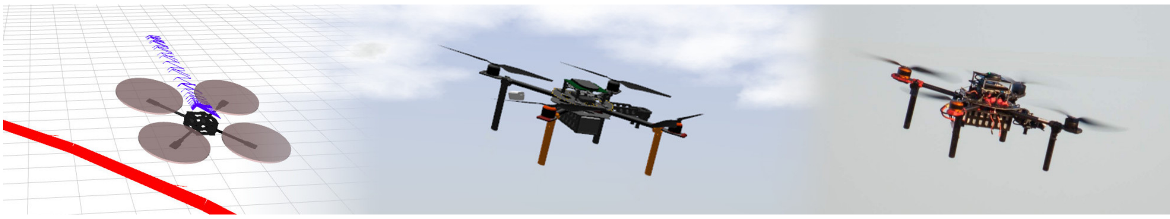


Fig. 6: The Tarot 650 UAV platform is modeled with high fidelity within the simulation platform provided here.

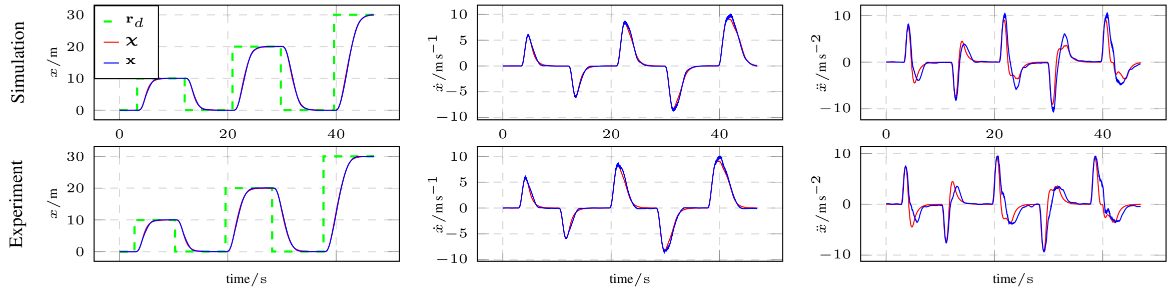


Fig. 7: Comparison of the simulated response and the real response to a step position reference $\mathbf{r}_d$ in a single axis. The UAV was controlled using the *SE(3) controller*. The graphs show the position, the velocity and the acceleration of the system, in terms of both the control reference $\boldsymbol{\chi}$ and the estimated state $\mathbf{x}$. The *MPC tracker* operated with the following constraints: $\dot{x}_{max} = 9\,\mathrm{m\,s^{-1}}$, $\ddot{x}_{max} = 12\,\mathrm{m\,s^{-2}}$, $\dddot{x}_{max} = 50\,\mathrm{m\,s^{-3}}$, $\ddddot{x}_{max} = 50\,\mathrm{m\,s^{-4}}$.

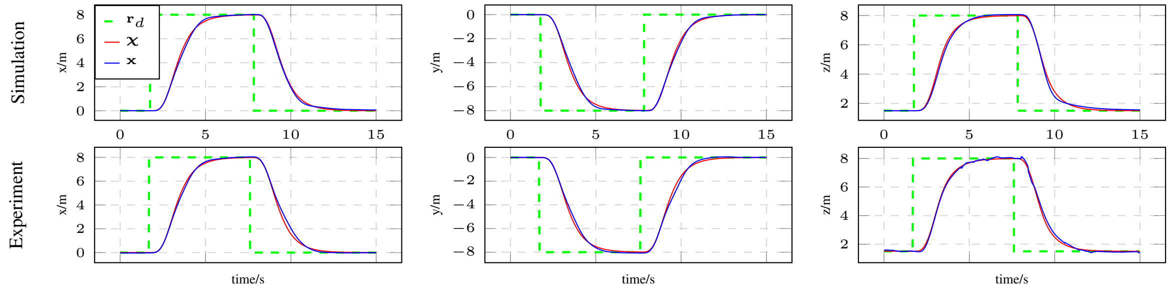


Fig. 8: Comparison of the simulated response and the real response to a step position reference $\mathbf{r}_d$ in all three translation axes. The UAV was controlled using the *SE(3) controller*. The graphs show the x, y and z position of the UAV, in terms of both the control reference $\boldsymbol{\chi}$ and the estimated state $\mathbf{x}$. The *MPC tracker* operated with the following constraints: $\dot{x}_{max} = 9\,\mathrm{m\,s^{-1}}$, $\ddot{x}_{max} = 12\,\mathrm{m\,s^{-2}}$, $\dddot{x}_{max} = 50\,\mathrm{m\,s^{-3}}$, $\ddddot{x}_{max} = 50\,\mathrm{m\,s^{-4}}$.

9 Pushing the frontiers of UAV research

The proposed UAV system has been used extensively for evaluating of basic research outside laboratory conditions, in applied research, and during real-world verification of novel approaches within robotic challenges and competitions. The system has been evolving continuously over the years as we have faced the challenges of various scenarios described in this section. None of the previously published papers contains a complete and up-to-date description of the system, mainly due to their focus on high-level robotics tasks. This publication therefore focuses solely on the underlying UAV system, which has been shaped by the vast number of application scenarios that have required different onboard sensor configurations. One of the main contributions of this publication resulting from the diverse application requirements is the creation of a universal system. The following subsections will briefly discuss the major results achieved using the proposed architecture.

9.1 UAV mutual detection and localization

The system played an integral role in the ongoing research on relatively-localized UAV swarms and forma-

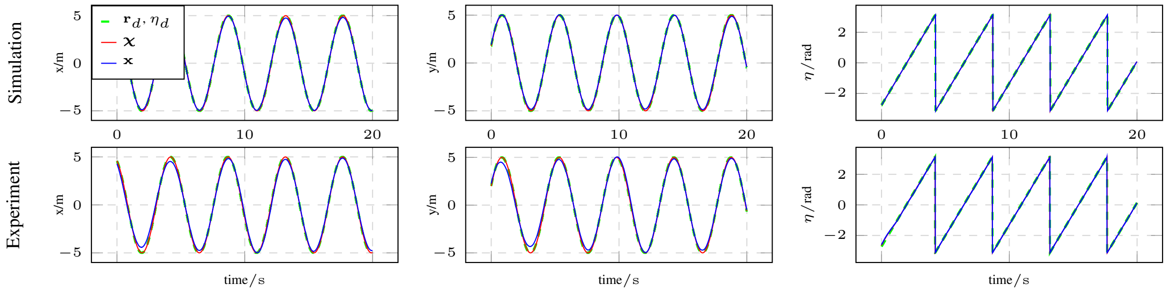


Fig. 9: Comparison of the simulated tracking and the real tracking of a horizontal circular reference $\mathbf{r}_d$ with a 5 m radius, 7 m s^{-1} speed, constant height, and with the heading pointing towards the center of the circle. The UAV was controlled using the *SE(3) controller*. The graphs show the x, y position of the UAV and heading η in terms of both the control reference $\boldsymbol{\chi}$ and the estimated state $\mathbf{x}$.

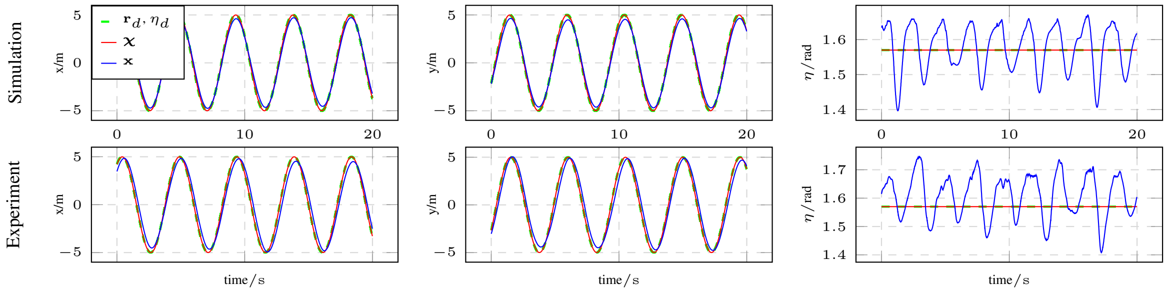


Fig. 10: Comparison of the simulated tracking and the real tracking of a horizontal circular reference $\mathbf{r}_d$ with a 5 m radius, 7 m s^{-1} speed, constant height, and a constant heading. The UAV was controlled using the *SE(3) controller*. The graphs show the x, y position of the UAV and heading η in terms of both the control reference $\boldsymbol{\chi}$ and the estimated state $\mathbf{x}$.

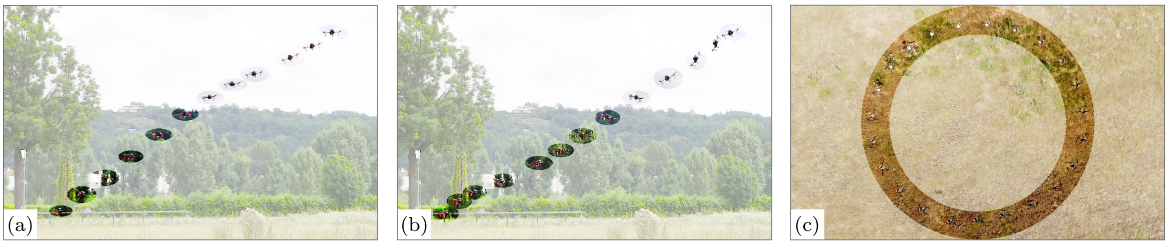


Fig. 11: Photo collage of a UAV performing aggressive testing maneuvers. Figures (a) and (b) depict the UAV performing the 3D step reference, as showcased in Fig. 8. Figure (c) shows a top-down view of the circular trajectory, showcased in Fig. 10 and in Fig. 9. Go to `http://mrs.felk.cvut.cz/mrs-uav-system` for video material from the experiments.

tions. Onboard marker-less UAV detection and localization were studied in [76, 74]. Two approaches to UAV detection were proposed, and were experimentally verified with the proposed system: a Convolutional Neural Network-based method, and a system for processing depth-camera images. Mutual localization of UAVs within swarms and formations was presented in [77, 78, 79, 80]. The system relies on modulated Ultra-Violet (UV) Light-emitting Diode (LED) blinkers, which are detected using specialized onboard cameras (see Fig. 14). This Ultra-Violet Direction And Ranging (UVDAR) system is also available as open-source[26].

9.2 UAV motion planning

Basic research on optimal planning for data collection with UAVs was studied in [43, 44, 45, 46, 19]. The plat-

[26] UVDAR, `http://github.com/ctu-mrs/uvdar`

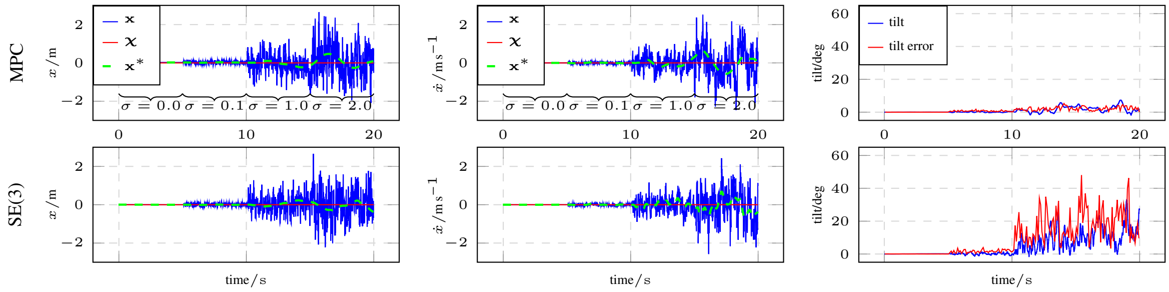


Fig. 12: Comparison of simulated control performance under noisy UAV state estimate $\mathbf{x}$. A simulated ground truth is denoted by $\mathbf{x}^*$. The UAV state estimate is based upon noisy position and velocity measurements (e.g., visual odometry) with artificial noise drawn from the distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0.0$ and $\sigma \in \{0.0, 0.1, 1.0, 2.0\}$. The *MPC controller* (first row) successfully stabilizes the UAV without producing excessive tilts $\angle(\hat{\mathbf{b}}_3, \hat{\mathbf{e}}_3)$ and tilt control errors. The *SE(3) controller* (second row) handles the situation with more difficulty while producing excessive tilts and tilt control errors. The control action of the *MPC controller* is thus fit for noisy localization systems. The *SE(3) controller* may destabilize the UAV and disturb the onboard localization systems with excessive control actions.

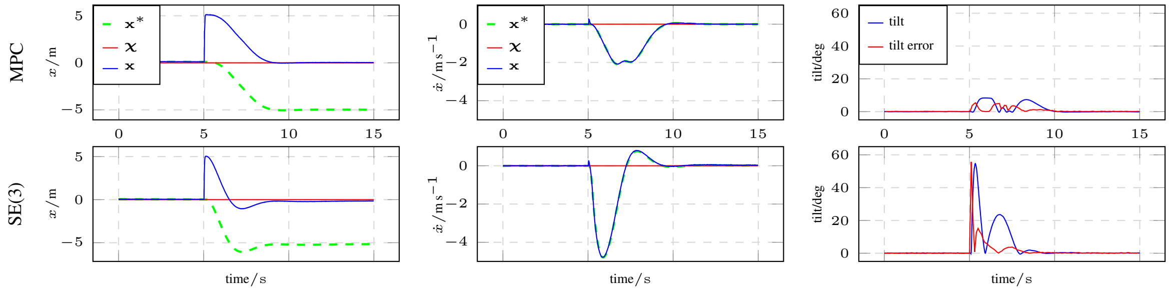


Fig. 13: Simulated comparison of control reactions to a 5 m jump in the estimated position of the UAV $\mathbf{x}$ (e.g., due to a malfunction of a localization system), $\mathbf{x}^*$ stands for the ground truth. The *MPC controller* (first row) successfully minimizes the control error while satisfying its dynamics constraints (2 m s^{-1} speed, 2 m s^{-2} acceleration) and thus produces reasonably small tilts $\angle(\hat{\mathbf{b}}_3, \hat{\mathbf{e}}_3)$ and tilt control errors. The *SE(3) controller* (second row) handles the situation with more difficulty while producing unbounded speed, acceleration, and therefore excessive tilts and tilt control errors, which may further disturb the onboard localization systems.

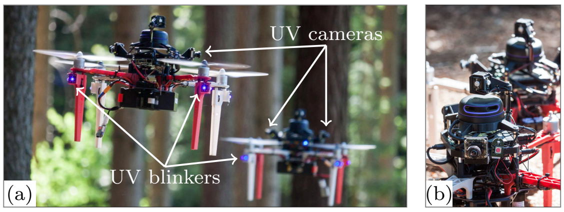


Fig. 14: Mutual localization of UAVs by the UVDAR system is provided by (a) UV blinkers on the UAV arms and top. The blinkers are observed by onboard cameras (b) equipped with UV band pass filters.

form provided real-world verification and showed the feasibility of the proposed approaches. Coverage optimization for multi-UAV cooperative surveillance was tackled in [50, 18]. Complex maneuvers and cooperative load-carrying by multiple UAVs were reported on in [69, 67].

9.3 Automatic control

A system for automatic gain tuning for the *SE(3) controller* (see Sec. 5.6) was published in [22]. A novel optimal control design approach for automatic fire extinguishing is showcased in [58]. The properties of the *SE(3)* geometric feedback proved crucial for verifying the feasibility of the almost-free-fall trajectories designed to dispatch water during extreme maneuvers (see Fig. 15).

9.4 Data gathering

The system is being used actively in a project working on indoor aerial inspection of historical buildings and

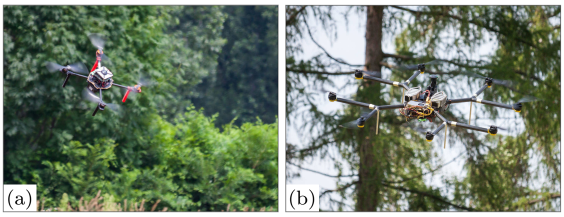


Fig. 15: Novel control approaches can be tested on a real hardware. Off-the-shelf platforms such as (a) Tarot 650, and also (b) custom-built airframes, can be equipped with the proposed system.

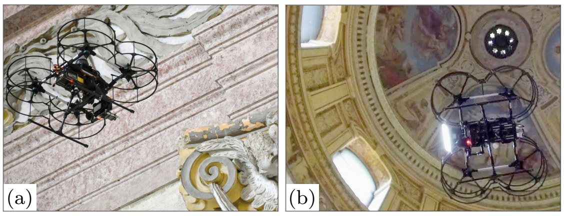


Fig. 16: An inspection of an indoor historical building is conducted (a) to monitor the state of frescoes, and (b) to assess the state of wall paintings.

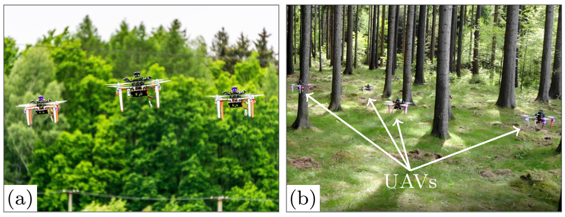


Fig. 17: Swarms of multirotor UAVs testing novel flocking algorithms while localized (a) by a GNSS system, and (b) by onboard sensors only within a forest environment.

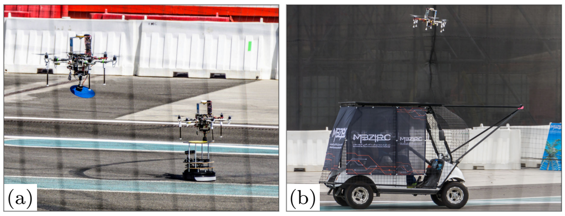


Fig. 18: The CTU-UPENN-UoL team during the MBZIRC 2017 competition. The photos show (a) two UAVs while delivering ferrous objects, and (b) a UAV during autonomous landing on a moving car.

monuments [48, 64, 29]. Within this scenario, a UAV is equipped with a 3D LiDAR sensor and is automatically guided through an indoor environment, where it captures detailed imagery of hard-to-reach points of interest (see Fig. 16). In another project[27], ionizing radiation mapping and localization is studied in [8, 4, 71]. Similarly, transmission radio sources were automatically localized in [75].

9.5 UAV swarms and formations

Basic research in the area of UAV swarming and formation flying was studied in [63, 65, 62]. UAV swarm control is a relatively new field of research, and its applications are yet to be explored. One of many possibilities being explored by the authors is the use of UAVs for inspecting hard-to-access locations such as power line towers without putting personnel at risk[28]. This type of application requires the swarm coordination to be flexible, and to move, while minimizing the observed object estimation error. Flocking capabilities are being explored within the framework of ongoing projects with real-world experiments in a field, and also within a forest environment (see Fig. 17). Interactions between UAVs are studied in order to overcome challenging situations such as GNSS-denied environment navigation.

9.6 MBZIRC 2017 competition

The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017[29] aimed at pushing the frontiers of field robotics. Two tasks out of the three challenges within the competition were focused solely on aerial manipulation and UAV control. The competition imposed real-world constraints in its tasks that forced the participating teams to show the current state of the art in robotics and to perform the tasks within a short time window and within specified time slots. The first task — autonomous gathering of colored ferrous objects by a group of UAVs — was successfully tackled by the CTU-UPENN-UoL[30] team, using the proposed system [68, 18, 34] (see Fig. 18). We won 1st place among the best teams from all over the world. The second task of autonomous landing on a moving car was also tackled by the proposed system. We achieved the fastest autonomous landing among all the teams, and we took the 2nd place overall in the competition [7, 70].

[27] `http://mrs.felk.cvut.cz/radron`

[28] `https://aerial-core.eu`

[29] MBZIRC 2017, `http://mbzirc.com/challenge/2017`

[30] Collaboration of Czech Technical University in Prague, University of Pennsylvania, and the University of Lincoln.

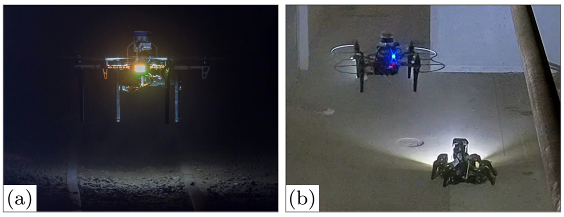


Fig. 19: Unmanned Aerial Vehicles during the DARPA SubT challenge. The photos depict (a) a UAV exploring an underground mine, and (b) mapping an unfinished nuclear power plant.

9.7 The DARPA Subterranean (SubT) challenge

The Defense Advanced Research Projects Agency (DARPA), an agency of the United States Department of Defense, organizes series of challenges focused on automatic search & rescue in an underground environment — the DARPA Subterranean challenge. In the DARPA Tunnel Circuit, the first round of the challenge, we deployed autonomous UAVs and semi-autonomous ground robots to explore underground mine shafts [49, 56]. Our team deployed autonomous UAVs with the proposed system (see Fig. 19), which navigated the underground tunnels and returned safely to the entrance while autonomously localizing objects of interest. We won the 1st prize among the self-funded teams and the 3rd prize overall. To the best of our knowledge, our UAVs managed to explore a greater distance into the tunnels than any of the other teams.

In the DARPA Urban Circuit, the second round of the challenge, we deployed autonomous UAVs and semi-autonomous ground robots to explore the infrastructure of an unfinished nuclear power plant. Our UAVs managed to explore 2867 m^3 of one floor of the reactor building while automatically navigating up to 100 m in just 200 s in a completely unknown environment. We again took 1st place among the self-funded teams, and 3rd place overall. Scientific publications on tasks within the Urban Circuit are under preparation.

9.8 MBZIRC 2020 competition

The second round of the MBZIRC competition was organized in 2020. It pushed the current state of the art in aerial robotics to its limits, with tasks such as organizing a group of UAVs and a UGV to build a brick wall autonomously, autonomous indoor and outdoor firefighting with UAVs, and autonomously catching a ball carried by a UAV, performed simultaneously with balloon popping by a group of UAVs (see Fig. 20). All of the tasks were solved using the proposed UAV system, and our participation in the competition helped to consolidate many of the platform's functionalities. The CTU-UPENN-NYU[31] team achieved the highest score of all the teams for building the brick wall autonomously. We also took 2nd in the autonomous balloon popping and ball-catching task. We won the gold medal in the *grand challenge* in which all the tasks were tested simultaneously. Scientific publications reporting on MBZIRC 2020 are under preparation [6].

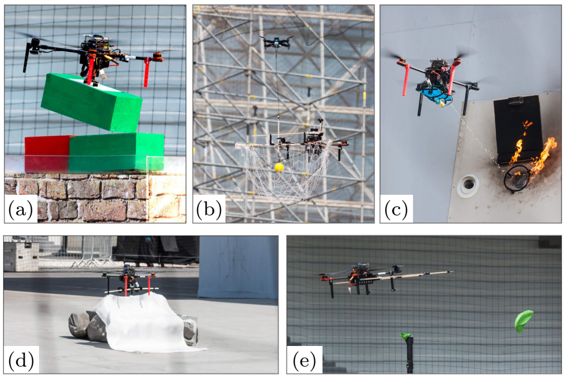


Fig. 20: The CTU-UPENN-NYU team during the MBZIRC 2020 competition. The photos depict (a) autonomous wall building, (b) autonomous ball catching, (c) autonomous fire extinguishing, (d) autonomous fire blanket deployment, and (e) autonomous balloon popping.

9.9 IEEE RAS Summer School on Multi-robot Systems

The proposed system was used as an educational tool during the 2019 Institute of Electrical and Electronics Engineers (IEEE) Robotics and Automation Society (RAS) summer school on multirobot systems[32]. More than 70 international students were challenged to solve a multi-UAV Dubins traveling salesman problem with neighborhoods during the summer school exercises. Student solutions were put to test during an outdoor experimental session.

10 CONCLUSIONS

We have presented a multirotor UAV control and estimation system created with emphasis on realistic sim-

[31] Collaboration between the Czech Technical University in Prague, the University of Pennsylvania, and the New York University.

[32] `http://mrs.felk.cvut.cz/summer-school-2019`

ulations and real-world experiments. The system is a product of years of cutting-edge research on aerial systems and their use in various branches of autonomous robotics. The proposed architecture allows reliable deployment of UAVs outside laboratory conditions using only onboard sensors. The proposed control pipeline supports fast and agile maneuvers as well as safe flight even with noisy and unreliable sensors. We have provided a well-documented and open-source implementation, which is being actively used by many researchers in the field. The Multi-robot Systems Group (MRS) team at CTU in Prague has achieved outstanding results in robotics challenges and competitions using this system. The experience gained from the challenges helped to shape the proposed system into the presented form.

Acknowledgements This work was supported by CTU grant no SGS20/174/OHK3/3T/13, by the Czech Science Foundation under research project No. 20-10280S, by Ministry of Education of the Czech Republic project CZ.02.1.01/0.0/0.0/16 019/0000765 “Research Center for Informatics”, and by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 871479.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Abeywardena, D., Pounds, P., Hunt, D., Dissanayake, G.: Design and development of recopter: An open source ros-based multi-rotor platform for research. In: ACRA (2015)
2. Aboudonia, A., Rashad, R., El-Badawy, A.: Composite hierarchical anti-disturbance control of a quadrotor UAV in the presence of matched and mismatched disturbances. Journal of Intelligent & Robotic Systems **90**(1-2), 201–216 (2018)
3. Baca, T., Hert, D., Loianno, G., Saska, M., Kumar, V.: 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, pp. 1–8. IEEE (2018)
4. Baca, T., Jilek, M., et al.: Timepix Radiation Detector for Autonomous Radiation Localization and Mapping by Micro Unmanned Vehicles. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1–8. IEEE (2019)
5. Baca, T., Loianno, G., Saska, M.: Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles. In: IEEE MMAR, pp. 992–997. IEEE (2016)
6. Baca, T., Penicka Robert Stepan, P., Petrlik, M., Spurny, V., Hert, D., Saska, M.: Autonomous Cooperative Wall Building by a Team of Unmanned Aerial Vehicles in the MBZIRC 2020 Competition. submitted to Robotics and Autonomous System (2020)
7. Baca, T., Stepan, P., Spurny, B., Hert, D., Penicka, R., Saska, M., Thomas, J., Loianno, G., Kumar, V.: Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle. Journal of Field Robotics **36**, 874–891 (2019)
8. Baca, T., Turecek, D., McEntaffer, R., Filgas, R.: Rospix: modular software tool for automated data acquisitions of Timepix detectors on Robot Operating System. Journal of Instrumentation **13**(11), C11008 (2018)
9. Benallegue, A., Mokhtari, A., Fridman, L.: High-order sliding-mode observer for a quadrotor UAV. International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal **18**(4-5), 427–440 (2008)
10. Burri, M., Datwiler, M., Achtelik, M.W., Siegwart, R.: Robust state estimation for micro aerial vehicles based on system dynamics. In: 2015 IEEE international conference on robotics and automation, pp. 5278–5283. IEEE (2015)
11. Cayero, J., Rotondo, D., Morcego, B., Puig, V.: Optimal state observation using quadratic boundedness: Application to UAV disturbance estimation. International Journal of Applied Mathematics and Computer Science **29**(1), 99–109 (2019)
12. Ceballos, A., De Silva, L., Herrb, M., Ingrand, F., Mallet, A., Medina, A., Prieto, M.: GenoM as a robotics framework for planetary rover surface operations. In: ASTRA, pp. 12–14 (2011)
13. Rodriguez de Cos, C., Acosta, J.A.: Explicit Aerodynamic Model Characterization of a Multirotor Unmanned Aerial Vehicle in Quasi-Steady Flight. Journal of Computational and Nonlinear Dynamics **15**(8) (2020)
14. Delmerico, J., et al.: The current state and future outlook of rescue robotics. Journal of Field Robotics **36**(7), 1171–1191 (2019)
15. Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors. Matrix **58**(15-16), 1–35 (2006)
16. Ebeid, E., Skriver, M., Terkildsen, K.H., Jensen, K., Schultz, U.P.: A survey of open-source uav flight controllers and flight simulators. Microprocessors and Microsystems **61**, 11–20 (2018)

17. Elkady, A., et al.: Robotics middleware: A comprehensive literature survey and attribute-based bibliography. Journal of Robotics **2012** (2012)
18. Faigl, J., Vana, P., Penicka, R., Saska, M.: Unsupervised Learning based Flexible Framework for Surveillance Planning with Aerial Vehicles. Journal of Field Robotics **36**(1), 270–301 (2019)
19. Faigl, J., Vana, P., Saska, M., Baca, T., Spurny, V.: On solution of the Dubins touring problem. In: IEEE ECMR, pp. 1–6 (2017)
20. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: RotorS — A modular gazebo MAV simulator framework. In: Robot Operating System (ROS), pp. 595–625. Springer (2016)
21. Gao, F., Wu, W., Lin, Y., Shen, S.: Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In: 2018 IEEE International Conference on Robotics and Automation, pp. 344–351. IEEE (2018)
22. Giernacki, W., Horla, D., Baca, T., Saska, M.: Real-time model-free minimum-seeking autotuning method for unmanned aerial vehicle controllers based on fibonacci-search algorithm. Sensors **19**(2), 312 (2019)
23. Hentout, A., Maoudj, A., Bouzouia, B.: A survey of development frameworks for robotics. In: IEEE ICMIC, pp. 67–72. IEEE (2016)
24. Hentzen, D., Stastny, T., Siegwart, R., Brockers, R.: Disturbance Estimation and Rejection for High-Precision Multirotor Position Control. In: IEEE/RSJ IROS, pp. 2797–2804. IEEE (2019)
25. Inigo-Blasco, P., et al.: Robotics software frameworks for multi-agent robotic systems development. Robotics and Autonomous Systems **60**(6), 803–821 (2012)
26. Jasim, O.A., Veres, S.M.: A robust controller for multi rotor uavs. Aerospace Science and Technology **105**, 106010 (2020)
27. Kamel, M., Alonso-Mora, J., Siegwart, R., Nieto, J.: Robust Collision Avoidance for Multiple Micro Aerial Vehicles Using Nonlinear Model Predictive Control. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 236–243. IEEE (2017)
28. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In: IEEE SSRR. IEEE (2011)
29. Kratky, V., Petracek, P., Spurny, V., Saska, M.: Autonomous reflectance transformation imaging by a team of unmanned aerial vehicles. IEEE Robotics and Automation Letters **5**(2), 2302–2309 (2020)
30. Labbadi, M., Cherkaoui, M.: Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV. Aerospace Science and Technology **93**, 105306 (2019)
31. L'Afflitto, A., Anderson, R.B., Mohammadi, K.: An Introduction to Nonlinear Robust Control for Unmanned Quadrotor Aircraft: How to Design Control Algorithms for Quadrotors Using Sliding Mode Control and Adaptive Control Techniques [Focus on Education]. IEEE Control Systems Magazine **38**(3), 102–121 (2018)
32. Lee, T., et al.: Geometric tracking control of a quadrotor UAV on SE(3). In: 2010 IEEE Conference on Decision and Control, pp. 5420–5425. IEEE (2010)
33. Lee, T., et al.: Nonlinear robust tracking control of a quadrotor UAV on SE(3). Asian Journal of Control **15**(2), 391–408 (2013)
34. Loianno, G., Spurny, V., Thomas, J., Baca, T., Thakur, D., et al.: Localization, Grasping, and Transportation of Magnetic Objects by a team of MAVs in Challenging Desert like Environments. IEEE Robotics and Automation Letters **3**(3), 1576–1583 (2018)
35. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation, pp. 6235–6240. IEEE (2015)
36. Mellado-Bataller, I., Pestana, J., Olivares-Mendez, M.A., Campoy, P., Mejias, L.: MAVwork: a framework for unified interfacing between micro aerial vehicles and visual controllers. In: Frontiers of Intelligent Autonomous Systems, pp. 165–179. Springer (2013)
37. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2520–2525. IEEE (2011)
38. Metta, G., Fitzpatrick, P., Natale, L.: YARP: yet another robot platform. International Journal of Advanced Robotic Systems **3**(1), 8 (2006)
39. Mohta, K., Watterson, M., Mulgaonkar, Y., Liu, S., Qu, C., Makineni, A., Saulnier, K., Sun, K., Zhu, A., Delmerico, J., et al.: Fast, autonomous flight in GPS-denied and cluttered environments. Journal of Field Robotics **35**(1), 101–120 (2018)
40. Mueller, M.W., Hehn, M., D'Andrea, R.: A computationally efficient motion primitive for quadrocopter trajectory generation. IEEE Transactions on Robotics **31**(6), 1294–1310 (2015)
41. Nascimento, I.B., Ferramosca, A., Piment, L.C., Raffo, G.V.: NMPC Strategy for a Quadrotor UAV

in a 3D Unknown Environment. In: ICAR, pp. 179–184. IEEE (2019)

42. Nascimento, T.P., Saska, M.: Position and attitude control of multi-rotor aerial vehicles: A survey. Annual Reviews in Control **48**, 129–146 (2019)
43. Penicka, R., Faigl, J., Saska, M., Vana, P.: Data collection planning with non-zero sensing distance for a budget and curvature constrained unmanned aerial vehicle. Autonomous Robots (2019)
44. Penicka, R., Faigl, J., Vana, P., Saska, M.: Dubins orienteering problem. IEEE Robotics and Automation Letters **2**(2), 1210–1217 (2017)
45. Penicka, R., Faigl, J., Vana, P., Saska, M.: Dubins orienteering problem with neighborhoods. In: 2017 IEEE International Conference on Unmanned Aircraft Systems, pp. 1555–1562 (2017)
46. Penicka, R., Saska, M., Reymann, C., Lacroix, S.: Reactive dubins traveling salesman problem for replanning of information gathering by UAVs. In: IEEE ECMR. IEEE (2017)
47. Pereira, J.C., Leite, V.J., Raffo, G.V.: Nonlinear Model Predictive Control on SE(3) for Quadrotor Trajectory Tracking and Obstacle Avoidance. In: ICAR, pp. 155–160. IEEE (2019)
48. Petracek, P., Kratky, V., Saska, M.: Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments. IEEE Robotics and Automation Letters **5**(2), 2078–2085 (2020)
49. Petrlik, M., Baca, T., Hert, D., Vrba, M., Krajnik, T., Saska, M.: A Robust UAV System for Operations in a Constrained Environment. IEEE Robotics and Automation Letters **5** (2020)
50. Petrlik, M., Vonasek, V., Saska, M.: Coverage optimization in the cooperative surveillance task using multiple micro aerial vehicles. In: IEEE SMC (2019)
51. Poultney, A., Kennedy, C., Clayton, G., Ashrafiuon, H.: Robust tracking control of quadrotors based on differential flatness: Simulations and experiments. IEEE/ASME Transactions on Mechatronics **23**(3), 1126–1137 (2018)
52. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018)
53. Quigley, M., et al.: Ros: an open-source robot operating system. In: IEEE ICRA workshop on open source software, vol. 3, p. 5. Kobe, Japan (2009)
54. Ramirez-Mendoza, A., Covarrubias-Fabela, J., Amezquita-Brooks, L., Garcia Salazar, O., Ramirez-Mendoza, W.: Fuzzy Adaptive Neurons Applied to the Identification of Parameters and Trajectory Tracking Control of a Multi-Rotor Unmanned Aerial Vehicle Based on Experimental Aerodynamic Data. Journal of Intelligent & Robotic Systems pp. 380–393 (2020)
55. Richter, C., Bry, A., Roy, N.: Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: Robotics Research, pp. 649–666. Springer (2016)
56. Roucek, T., et al.: DARPA Subterranean Challenge: Multi-robotic exploration of underground environments. In: IEEE MESAS, vol. 11995, pp. 274–290 (2019)
57. Roussel, M.R.: Nonlinear Dynamics: A hands-on introductory survey. Morgan & Claypool Publishers (2019)
58. Saikin, D.A., Baca, T., Gurtner, M., Saska, M.: Wildfire Fighting by Unmanned Aerial System Exploiting Its Time-Varying Mass. IEEE Robotics and Automation Letters **5**(2), 2674–2681 (2020)
59. Sanchez-Lopez, J.L., Suarez Fernandez, R.A., Bavle, H., Sampedro, C., Molina, M., Pestana, J., Campoy, P.: AEROSTACK: An architecture and open-source software framework for aerial robotics. In: 2016 IEEE International Conference on Unmanned Aircraft Systems, pp. 332–341 (2016)
60. Sanchez-Lopez, J.L., et al.: A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-UAV systems: Simulation and experimentation. Journal of Intelligent & Robotic Systems **84**(1-4), 779–797 (2016)
61. Sanchez-Lopez, J.L., et al.: Trajectory Tracking for Aerial Robots: an Optimization-Based Planning and Control Approach. Journal of Intelligent & Robotic Systems pp. 1–44 (2020)
62. Saska, M.: Large Sensors with Adaptive Shape Realised by Self-stabilised Compact Groups of Micro Aerial Vehicles. In: Robotics Research, pp. 101–107. Springer International Publishing (2020)
63. Saska, M., Hert, D., Baca, T., Kratky, V., Nascimento, T.: Formation Control of Unmanned Micro Aerial Vehicles for Straitened Environments. Autonomous Robots pp. 1573–7527 (2020)
64. Saska, M., Kratky, V., Spurny, V., Baca, T.: Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control. In: IEEE ETFA. IEEE (2017)
65. Saska, M., et al.: Formations of unmanned micro aerial vehicles led by migrating virtual leader. In: IEEE ICARCV. IEEE (2016)
66. Schmittle, M., et al.: OpenUAV: a UAV testbed for the CPS and robotics community. In: ACM/IEEE ICCPS, pp. 130–139. IEEE (2018)

67. Spurny, V., Baca, T., Saska, M.: Complex manoeuvres of heterogeneous MAV-UGV formations using a model predictive control. In: IEEE MMAR, pp. 998–1003 (2016)
68. Spurny, V., Baca, T., Saska, M., Penicka, R., Krajnik, T., et al.: Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs. Journal of Field Robotics **36**(1), 125–148 (2019)
69. Spurny, V., Petrlik, M., Vonasek, V., Saska, M.: Cooperative transport of large objects by a pair of unmanned aerial systems using sampling-based motion planning. In: IEEE ETFA, pp. 955–962 (2019)
70. Stepan, P., Krajnik, T., Petrlik, M., Saska, M.: Vision techniques for on-board detection, following and mapping of moving targets. Journal of Field Robotics **36**(1), 252–269 (2019)
71. Stibinger, P., Baca, T., Saska, M.: Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles With Pixel Detectors in Real-Time. IEEE Robotics and Automation Letters **5**, 3634–3641 (2020)
72. Suarez Fernandez, R., Rodriguez Ramos, A., Alvarez, A., et al.: The Skyeye Team Participation in the 2020 Mohamed Bin Zayed International Robotics Challenge (2020). Online (accessed on 2020/09/05): `https://www.researchgate.net/publication/339725858_The_Skyeye_Team_Participation_in_the_2020_Mohamed_Bin_Zayed_International_Robotics_Challenge`
73. Tsardoulias, E., Mitkas, P.: Robotic frameworks, architectures and middleware comparison. arXiv preprint arXiv:1711.06842 (2017)
74. Vrba, M., Hert, D., Saska, M.: Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system. IEEE Robotics and Automation Letters **4**(4), 3402–3409 (2019)
75. Vrba, M., Pogran, J., Pritzl, V., Spurny, V., Saska, M.: Real-time localization of transmission sources using a formation of micro aerial vehicles. In: IEEE RCAR, pp. 203–208 (2019)
76. Vrba, M., Saska, M.: Marker-less micro aerial vehicle detection and localization using convolutional neural networks. IEEE Robotics and Automation Letters **5**(2), 2459–2466 (2020)
77. Walter, V., Novak, T., Saska, M.: Self-localization of unmanned aerial vehicles using onboard sensors, with focus on optical flow from camera image. In: MESAS (2017)
78. Walter, V., Saska, M., Franchi, A.: Fast mutual relative localization of UAVs using ultraviolet LED markers. In: 2018 IEEE International Conference on Unmanned Aircraft Systems (2018)
79. Walter, V., Staub, N., Saska, M., Franchi, A.: Mutual Localization of UAVs based on Blinking Ultraviolet Markers and 3D Time-Position Hough Transform. In: IEEE CASE (2018)
80. Walter, W., Staub, N., Franchi, A., Saska, M.: UVDAR System for Visual Relative Localization With Application to Leader-Follower Formations of Multirotor UAVs. IEEE Robotics and Automation Letters **4**(3), 2637–2644 (2019)
81. Xiao, K., Tan, S., Wang, G., An, X., Wang, X., Wang, X.: XTDrone: A Customizable Multi-Rotor UAVs Simulation Platform. arXiv preprint arXiv:2003.09700 (2020)
82. Zhang, J., Gu, D., Deng, C., Wen, B.: Robust and adaptive backstepping control for hexacopter uavs. IEEE Access **7**, 163502–163514 (2019)
83. Zhang, J., Singh, S.: LOAM: Lidar Odometry and Mapping in Real-time. In: Robotics: Science and Systems, vol. 2 (2014)

Chapter 4

Remote Sensing by UAVs

The author of this thesis has contributed to various subfields of multi-UAV research, including multi-UAV swarming [27a], [28a], [9a], [12a], search and rescue [25a], [29a], [10a], [18a], planning and control [11a], [30a], [13a], [107], [23a], and sensing [29a], [1c], [15a]. However, most of the author's effort during his studies had been invested in the research scenarios proposed by the Khalifa University as the 2017 and 2020 MBZIRC robotic challenges[1].

The following core publication [3c] in this thesis proposes a UAV system for autonomous landing on a moving car. This complete system was able to perform the fastest autonomous landing (25 s) among all the teams competing in the challenge, using onboard sensors and computational power. Onboard computer vision [77] was employed to detect the car in wide Field of View (FOV) monocular camera images and nonlinear state estimation based on Unscented Kalman Filter (UKF) was performed to predict the car's future motion. The car was driving at the speed of $15\,\mathrm{km\,h^{-1}}$ in a designated area, following an 8-shaped path. The system was evaluated extensively in simulations and various real-world conditions, and the final experiments were successfully performed in the constrained environment of the competition venue.

> [3c] **T. Baca**, P. Stepan, B. Spurny, D. Hert, R. Penicka, M. Saska, *et al.*, "Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle," *Journal of Field Robotics*, vol. 36, pp. 874–891, 5 2019

The second challenge in MBZIRC 2017 competition focused on the collaborative gathering of metal objects by a group of UAVs. A team of 3 UAVs was tasked with gathering colored metal discs that were both stationary and moving (attached to ground mobile robots). The challenge posed many subproblems, including onboard computer vision [77], sensor fusion, state estimation, feedback control, motion planning, robot coordination, and mechatronics. All the subproblems needed to be addressed competently and efficiently to obtain any score within the competition. The proposed system was tested extensively in simulations and various outdoor conditions [14a]. It performed consistently as the best solution to the challenge during the competition trials in Abu Dhabi. The MRS team won the 1st place among prestigious university teams from all over the world. The following core publication presents the system, its core features, and the experimental results.

> [2c] V. Spurny, **T. Baca**, M. Saska, R. Penicka, T. Krajnik, J. Thomas, *et al.*, "Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs," *Journal of Field Robotics*, vol. 36, no. 1, 125–148, 2019

The MBZIRC 2020 challenge inspired the third core publication in this research stream (submitted to *Robotics & Autonomous Systems*, after the first revision). The 2nd round of this

[1] `https://mbzirc.com`

competition focused again on Unmanned Aerial Vehicles and extended the 1st 2017 challenge. The article reports on the system for multi-robotic brick wall construction by a team of UAVs. Although this task is similar to the 2017 object-gathering challenge, significant advances in UAV control and state estimation were required to address the problem successfully. Moreover, a novel multi-frame state estimation framework was proposed for seamless switching between globally-localized flight and egomotion visual servoing [8c]. This system allowed the use of true visual servoing to achieve pinpoint accuracy during the autonomous grasping maneuver. The MRS team again won the 1st place in the competition and showed, by a significant margin, the best approach for solving the multi-robotic task.

[7c] **T. Baca**, P. Penicka Robert Stepan, M. Petrlik, V. Spurny, D. Hert, and M. Saska, "Autonomous Cooperative Wall Building by a Team of Unmanned Aerial Vehicles in the MBZIRC 2020 Competition," *submitted to Robotics and Autonomous System*, Dec. 2020. eprint: `arXiv:2012.05946`

Received: 15 October 2017 | Revised: 13 August 2018 | Accepted: 9 December 2018
DOI: 10.1002/rob.21858

REGULAR ARTICLE

WILEY

Autonomous landing on a moving vehicle with an unmanned aerial vehicle

Báča Tomáš[1] | Petr Stepan[1] | Spurný Vojtěch[1] | Daniel Hert[1] | Robert Penicka[1] | Martin Saska[1] | Justin Thomas[2] | Giuseppe Loianno[3] | Vijay Kumar[2]

[1]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

[2]GRASP Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania

[3]Department of ECE and MAE, Tandon School of Engineering, New York University, New York City, New York

Correspondence
Báča Tomáš, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 166 27, Czech Republic.
Email: tomas.baca@fel.cvut.cz

Funding information
České Vysoké Učení Technické v Praze, Grant/Award Number: SGS17/187/OHK3/3T/13; Army Research Laboratory, Grant/Award Number: W911NF-17-2-0181; Office of Naval Research Global, Grant/Award Numbers: N00014-14-1-0510, N00014-07-1-0829; Grantová Agentura České Republiky, Grant/Award Number: 17-16900Y; Khalifa University of Science, Technology and Research, Grant/Award Number: MBZIRC 2017

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/hr. 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 Mohamed Bin Zayed International Robotics Challenge (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.

KEYWORDS
aerial robotics, control, planning, position estimation

1 | INTRODUCTION

Autonomous takeoff 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 multirotor 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 takeoff 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/hr 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 (hardware [HW] and software [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 (MPC)-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 nonstraight path.

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/hr

Team	Trial 1	Trial 2	Grand 1	Grand 2
Beijing Institute of Technology	63.4	63.4	NQ	NQ
CTU in Prague, UPENN and UoL	143.2	84.6	25.1	M
University of Bonn	110.5	Not landed	58.6	42.3
University of Catania	134.5	Falling off	NQ	NQ

Note. Falling off: UAV touched the mobile landing platform but was not able to fix there; Grand 1 and 2: trials of the MBZIRC Grand Challenge; M: manual mode applied due to other robots in the arena; MBZIRC: Mohamed Bin Zayed International Robotics Challenge; not landed: UAV not landed due to a crash or time limit of 15 min exceeding; NQ: not qualified for the final challenge; Trial 1 and 2: trials of the MBZIRC Challenge 1; UAV: unmanned aerial vehicle.

1.1 | State of the art

The academic community has identified great interest in the task of autonomous landing of unmanned aerial vehicles (UAVs) on ground or marine vehicles. The survey in Jin, Zhang, Shen, and Li (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, Zhou, Zhang, and Zhang (2014).

Vision-based estimation of a ground vehicle states using an only UAV onboard sensors is proposed in Benini, Rutherford, and Valavanis (2016). Robust marker detection using onboard graphics processing unit provides precise pose information even in occluded and cluttered environments. Lin, Garratt, and Lambert (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.

Fu, Zhang, Yi, and Shi (2016), Ghommam and Saad (2017), W. Jung, Kim, and Bang (2016), and Y. Jung, Lee, and Bang (2015) 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, Wang, Paw, and Liao (2016).

Many of indoor experiments on autonomous landing on a slow-moving target are presented in works by Araar, Aouf, and Vitanov (2017), Bi and Duan (2013), Ghamry, Dong, Kamel, and Zhang (2016), and D. Lee, Ryan, and Kim (2012). 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 D. 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, Yousheng, Xiaokun, and Shing (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 Masselli, Yang, Wenzel, and Zell (2014), Yang, Scherer, and Zell (2013), and Yang, Ying, Lu, and Lu (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, Jung, Lee, and Shim (2014) and in H. Lee, Jung, and Shim (2016). 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, Bayasgalan, Wang, Tsechpenakis, and Panagou (2017). Borowczyk et al. (2017) propose a system that utilizes a vision-based approach combined with inertial and global positioning system (GPS) measurements from a cell phone placed on the ground vehicle. Experiments show landings at speeds up to 50 km/hr. However, it is unclear whether the system is capable of landing during nonlinear 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, Houben, Nieuwenhuisen, & Behnke, 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 UAV. 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 nonlinear motion model by an UKF. The proposed control architecture is tailored specifically to the challenge of following fast dynamic objects with a UAV by leveraging the nonlinear state controller in conjunction with a novel MPC 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 nonlinear 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/hr 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 UAV, a multirotor helicopter. The competition takes place in an arena with dimensions of 90 × 60 m, 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 × 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/hr. It reduces the speed to 10 km/hr after 6 min and to 5 km/hr after 12 min 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 important, we intended to reuse the

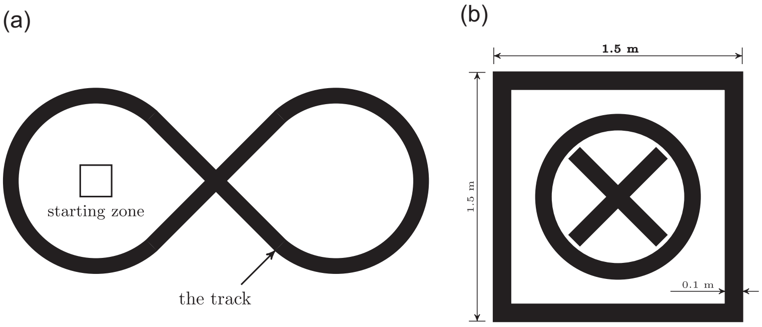


FIGURE 1 (a) A schematic image of the arena, showing the track for the ground vehicle. (b) The visual marker attached to the helipad of the ground vehicle, as described in the rules of the competition

platform for future research activities, which introduced a need for simple potential modifications to the system.

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 Linux Ubuntu 16.04 and the robot operating system (ROS) in the kinetic version. The ROS 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, and so forth. Sensor drivers are often found with the ROS interface already integrated, which makes them simpler to integrate.

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 real-time kinematics (RTK) 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.

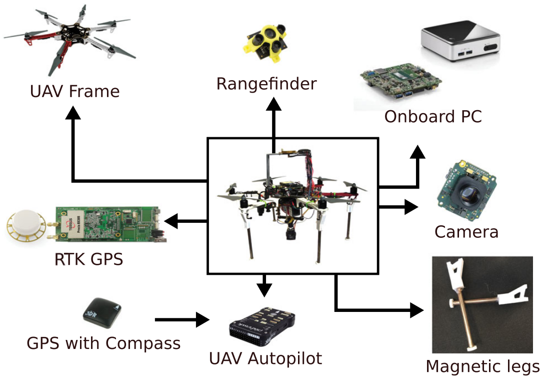


FIGURE 2 Schematic of individual hardware modules on the UAV. GPS: global positioning system; UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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 4–9.

The first component is the *landing pattern detector* (presented in Section 4), which provides measurements of car position in the world frame coordinate system. Position measurements are processed by the *car state estimator* (Section 6), using an UKF. 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 nonholonomic 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

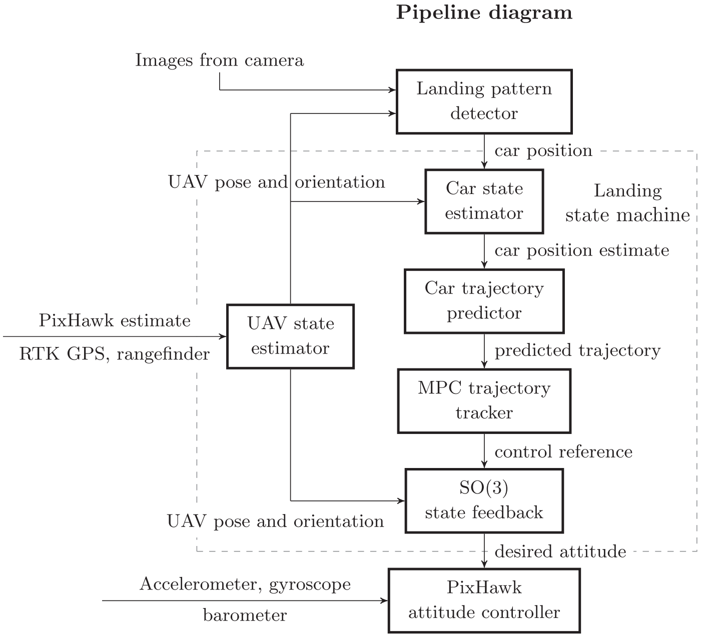


FIGURE 3 Scheme of the software pipeline for landing on a moving vehicle. See Section 4 for *landing pattern detector*, Section 6 for *car state estimator* and *car state predictor*, Section 7 for *UAV state estimator*, Section 8 for *MPC trajectory tracker*, and Section 9 for *SO(3) state feedback*. The dashed line surrounds the parts which are controlled by a landing state machine, later discussed in Section 5. MPC: model predictive controller; UAV: unmanned aerial vehicle

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, and 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 MPC 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, Loianno, and Saska (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/s, with resolution 752 × 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°. 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 reused 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, Krajnik, Petrlik, and Saska (2018), where all vision approaches used 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, for example, OpenCV, to provide simple implementation and reproducibility. Mainly the very fast response (50 frames/s 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.

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 (e.g., with box size 5 and 11 pixels; see Figure 4b,c, 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 used approach, which is described in detail 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.

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).

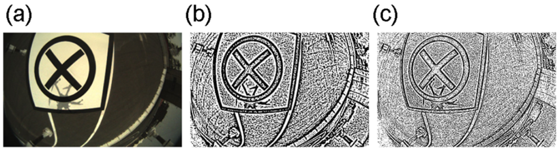


FIGURE 4 (a) Image taken by the camera, (b) result of the adaptive threshold with box size of 11 pixels, and (c) adaptive threshold with box size 5 pixels. The box size is variable and depends on the thickness of the expected segments in the landing pattern and decreases with the height of the unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

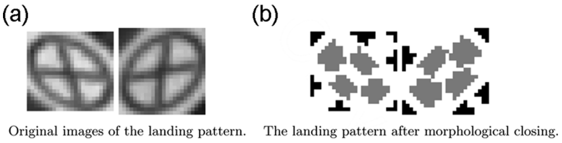

(a)
(b)
Original images of the landing pattern.
The landing pattern after morphological closing.

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 × 24 pixels. (a) Original images of the landing pattern. (b) The landing pattern after morphological closing

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 & 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 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, that is, 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 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

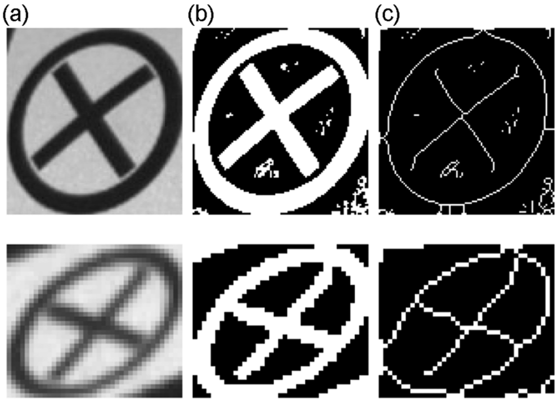


FIGURE 6 (a) Image taken by the camera, (b) application of the adaptive threshold with box size of 11 pixels, and (c) results of Guo Hall thinning

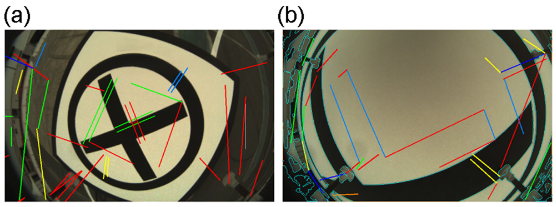


FIGURE 7 Line detection (a) if the cross is inside the circle and (b) if only a part of the landing pattern is detected [Color figure can be viewed at wileyonlinelibrary.com]

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, for example, 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.

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 to 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

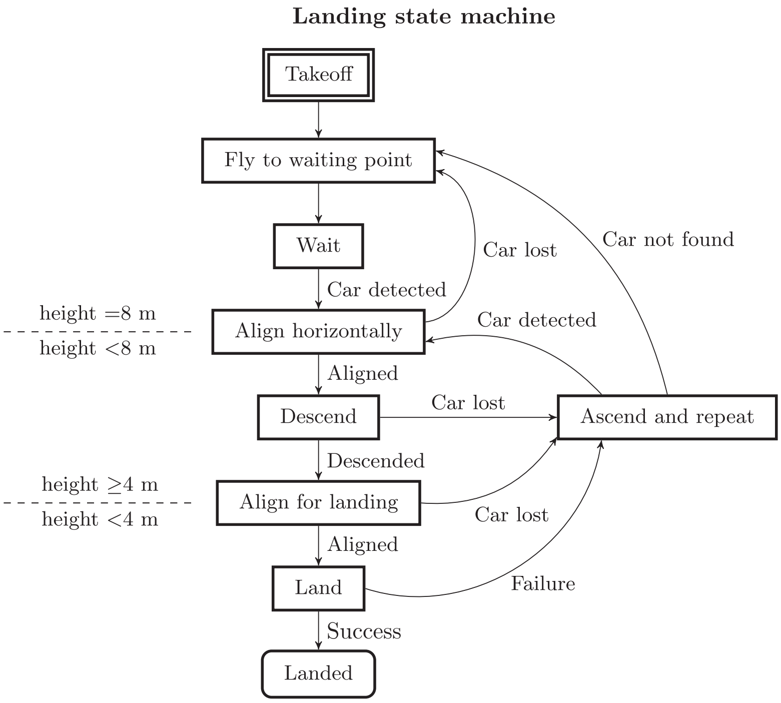


FIGURE 8 The autonomous flight, from takeoff to landing, is controlled by the *landing state machine*. The linear passage from the *takeoff* state to the *landed* state can be divided to three groups of states, based on the current height above ground −8 m or higher, between 4 and 8 m, and below 4 m. The state *ascend and repeat* handles situations when the car was lost from the field of view of the camera

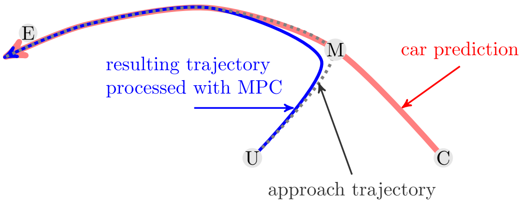


FIGURE 9 Illustration of the approach strategy. Point **M** represents the common meeting point of the UAV and the car, **U** is the current position of the UAV, **C** is the current position of the car, and **E** marks the final point in the car prediction. The car predicted trajectory is shown in red, the approach trajectory is marked as dotted, and the resulting feasible trajectory, optimized by MPC, is shown in blue. MPC: model predictive control; UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

the MPC tracker. This situation may occur if the car is first spotted while driving away from the UAV Figure 9.

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 rangefinder, 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 machines 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 subproblems 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 second-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

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{2}$$

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 nonlinear motion, a different model is required, for example, the nonholonomic car-like model:

$$\begin{aligned} \mathbf{x}^o_{[n+1]} &= \mathbf{x}^o_{[n]} + \dot{\mathbf{x}}^o_{[n]}\Delta t, \\ \dot{\mathbf{x}}^o_{[n+1]} &= \begin{pmatrix} \cos\phi_{[n+1]} \\ \sin\phi_{[n+1]} \end{pmatrix} v_{[n+1]}, \\ \phi_{[n+1]} &= \phi_{[n]} + \dot{\phi}_{[n]}\Delta t, \\ \dot{\phi}_{[n+1]} &= K_{[n]}v, \\ v_{[n+1]} &= v_{[n]} + a_{[n]}\Delta t, \\ K_{[n+1]} &= K_{[n]} + \dot{K}_{[n]}\Delta t, \end{aligned} \tag{3}$$

where $\mathbf{x}^o_{[n]} = (x, y)^T_{[n]}$ 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 Δt is the time difference.

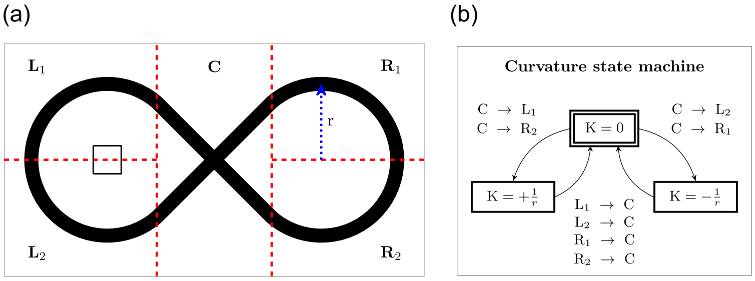


FIGURE 10 (a) The competition arena is divided into five areas. The center area (C) contains track with zero curvature. The four corner areas ($\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{R}_1$, $\mathbf{R}_2$) contain track with curvature $|K| = 1/r$. (b) State machine producing the bias for the curvature of the car. Inputs to the state machine are transitions of the car between the different parts of the arena [Color figure can be viewed at wileyonlinelibrary.com]

Car-like model better reflects the physics of the car motion, thanks to adding nonholonomic 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 UKF (Wann & 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

$$\mathbf{q}_{[n+1]} = f(\mathbf{q}_{[n]}),\ \forall n = 0,\ldots,n-1, \tag{4}$$

where $f()$ is the model function according to Equation (3), $\mathbf{q}_{[n]}$ is a complete state vector defined as

$$\mathbf{q}_{[n]} = [{\mathbf{x}^{o}_{[n+0]}}^{T}, \phi_{[n+1]}, K_{[n+1]}, v_{[n+1]}, a_{[n+1]}]^{T}, \tag{5}$$

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.

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 predefined 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 $\mathbf{L}_1$, $\mathbf{L}_2$, $\mathbf{R}_1$, $\mathbf{R}_2$, where the curvature of the turn is $|K| = 1/r$, and the center area $\mathbf{C}$, 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.

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.

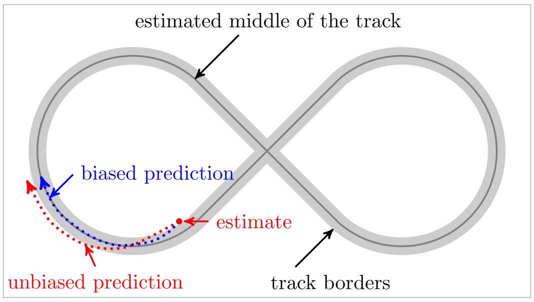


FIGURE 11 Scheme of the real track, on which the car was supposed to drive, with the path denoted in the middle of the track, to which the car prediction was biased. An example of a predicted (red) and a biased trajectory (blue) is shown [Color figure can be viewed at wileyonlinelibrary.com]

The main source of data for both the vertical axis and the horizontal axis in the proposed system is the PixHawk flight controller. Its EKF 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, 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

$$\mathbf{A} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \mathbf{B} = \begin{pmatrix}\Delta t\\ \Delta t\end{pmatrix}, \tag{6}$$

where $\mathbf{x}^e_{[n+1]} = \mathbf{A}\mathbf{x}^e_{[n]} + \mathbf{B}\mathbf{u}_{[n]}$ is the linear system equation, $\mathbf{x}^e_{[n]} = (x, y)^T_{[n]}$ is the state vector finally used for control, and $\mathbf{u}_{[n]}$ is the system input. According to our experience,

$$\mathbf{x}^p_{[0]} + \sum_{n=0}^{k} \dot{\mathbf{x}}^p_{[n]} \Delta t_{[n]} = \mathbf{x}^p_{[k]}, \forall k \in \mathbb{N} \tag{7}$$

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

$$\mathbf{u}_{[k]} = \left(\mathbf{x}^p_{[k]} - \mathbf{x}^p_{[k-1]}\right)/\Delta t_{[k]}, \tag{8}$$

which results in position update

$$\mathbf{x}^e_{[k+1]} = \mathbf{x}^e_{[k]} + \left(\mathbf{x}^p_{[k+1]} - \mathbf{x}^p_{[k]}\right), \tag{9}$$

that follows the PixHawk estimate when left uncorrected. However, when the RTK GPS is available, the estimate $\mathbf{x}^e_{[k+1]}$ 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 used 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, quadratic programming (QP) optimization can be performed to find a polynomial, given the initial and final state conditions (Mellinger & Kumar, 2011), which can then be derived and sampled to create the reference. In our case, we chose to generate the reference using a MPC approach. The following text describes a simpler variant of the original approach designed for multiple vehicles, which was proposed in Baca, Hert, Loianno, Saska, and Kumar (2018).

The 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 $\mathbf{x}_{[n]} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z})^T_{[n]}$ by minimizing the function

$$\begin{aligned} V(\mathbf{x}_{[0,\ldots,m-1]}, \mathbf{u}_{[0,\ldots,m-1]}) \;=\; & \frac{1}{2}\sum_{i=1}^{m-1}\left(\mathbf{e}^T_{[i]}\mathbf{Q}\mathbf{e}_{[i]} + \mathbf{u}^T_{[i]}\mathbf{P}\mathbf{u}_{[i]}\right), \\ \text{s.t.} \quad & \mathbf{x}_{[0,\ldots,m-1]} \geq \mathbf{x}_L, \\ & \mathbf{x}_{[0,\ldots,m-1]} \leq \mathbf{x}_U, \end{aligned} \tag{10}$$

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 $\mathbf{x}_\mathrm{L}$ and $\mathbf{x}_\mathrm{U}$ 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 third-order translational dynamics of the UAV with the system matrices

$$\mathbf{A} = \begin{pmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta t \end{pmatrix}, \tag{11}$$

where $\Delta t = 0.01$ s. The same matrices are used to formulate the MPC prediction. In our MPC formulation, Δt 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 Δt 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

$$\mathbf{Q} = \begin{pmatrix} 5000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 800 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 800 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 800 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 800 & 0 \end{pmatrix},$$

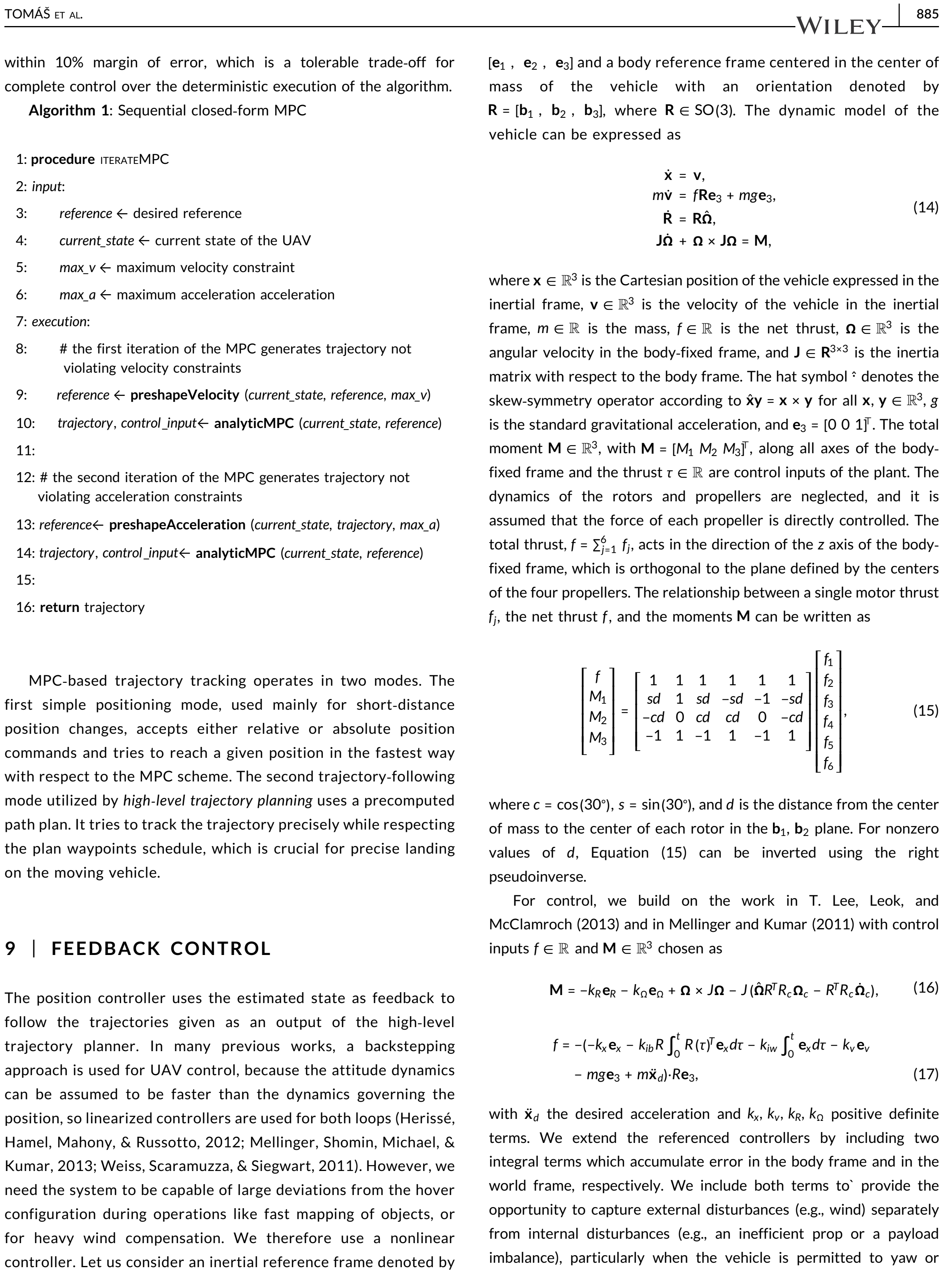
$$\mathbf{P} = \begin{pmatrix} 500 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 500 \end{pmatrix}. \tag{12}$$

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

$$\mathbf{U} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 5 \quad 5 \quad 5 \quad 5 \quad 5 \quad 10), \tag{13}$$

which results in 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 preshaping rather than more complicated control-oriented solutions. Thus in every iteration, a reference trajectory is first preshaped 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 preshaped 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

```
procedure ITERATEMPC
input:
    reference ← desired reference
    current_state ← current state of the UAV
    max_v ← maximum velocity constraint
    max_a ← maximum acceleration acceleration
execution:
    # the first iteration of the MPC generates trajectory not
      violating velocity constraints
    reference ← preshapeVelocity (current_state, reference, max_v)
    trajectory, control_input← analyticMPC (current_state, reference)

# the second iteration of the MPC generates trajectory not
  violating acceleration constraints
reference← preshapeAcceleration (current_state, trajectory, max_a)
trajectory, control_input← analyticMPC (current_state, reference)

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 (Herissé, Hamel, Mahony, & Russotto, 2012; Mellinger, Shomin, Michael, & Kumar, 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 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 $\mathbf{R} \in \mathrm{SO}(3)$. The dynamic model of the vehicle can be expressed as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + mg\mathbf{e}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\mathbf{\Omega}}, \\ \mathbf{J}\dot{\mathbf{\Omega}} &+ \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} = \mathbf{M}, \end{aligned} \tag{14}$$

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, $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame, and $\mathbf{J} \in \mathbf{R}^{3\times 3}$ 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 $\mathbf{e}_3 = [0\ 0\ 1]^T$. 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 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 $c = \cos(30°)$, $s = \sin(30°)$, 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 (15) can be inverted using the right pseudoinverse.

For control, we build on the work in T. 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

$$\mathbf{M} = -k_R\mathbf{e}_R - k_\Omega\mathbf{e}_\Omega + \mathbf{\Omega} \times J\mathbf{\Omega} - J(\hat{\mathbf{\Omega}}R^T R_c\mathbf{\Omega}_c - R^T R_c\dot{\mathbf{\Omega}}_c), \tag{16}$$

$$\begin{aligned} f = -(&-k_x\mathbf{e}_x - k_{ib}R\int_0^t R(\tau)^T\mathbf{e}_x d\tau - k_{iw}\int_0^t \mathbf{e}_x d\tau - k_v\mathbf{e}_v \\ &- mg\mathbf{e}_3 + m\ddot{\mathbf{x}}_d)\cdot R\mathbf{e}_3, \end{aligned} \tag{17}$$

with $\ddot{\mathbf{x}}_d$ the desired acceleration and k_x, k_v, k_R, k_Ω 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

$$\begin{aligned}
\mathbf{b}_{2,des} &= [-\sin\psi_{des}, \cos\psi_{des}, 0]^\top, \mathbf{b}_{3,C} = \frac{\mathbf{f}}{||\mathbf{f}||}, \mathbf{b}_{1,C} = \frac{\mathbf{b}_{2,des} \times \mathbf{b}_3}{||\mathbf{b}_{2,des} \times \mathbf{b}_3||}, \\
\mathbf{b}_{2,C} &= \mathbf{b}_3 \times \mathbf{b}_1, \dot{\mathbf{b}}_{2,des} = [-\cos\psi_{des}\dot{\psi}_{des}, -\sin\psi_{des}\dot{\psi}_{des}, 0]^\top, \\
\dot{\mathbf{b}}_{3,C} &= \mathbf{b}_{3,C} \times \frac{\dot{\mathbf{f}}}{||\mathbf{f}||} \times \mathbf{b}_{3,C}, \dot{\mathbf{b}}_{1,C} = \mathbf{b}_{1,C} \times \frac{\dot{\mathbf{b}}_{2,des} \times \mathbf{b}_{3,C} + \mathbf{b}_{2,des} \times \dot{\mathbf{b}}_{3,C}}{||\mathbf{b}_{2,des} \times \mathbf{b}_{3,C}||} \times \mathbf{b}_1, \\
\dot{\mathbf{b}}_{2,des} &= \dot{\mathbf{b}}_{3,C} \times \mathbf{b}_{1,C} + \mathbf{b}_{3,C} \times \dot{\mathbf{b}}_{1,C}, \hat{\mathbf{\Omega}}_C = R_C^\top \dot{R}_C.
\end{aligned} \tag{18}$$

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 ROS 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.

10.1 | Experiments before 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 nonlinear motion requires a more descriptive car-like model. Figure 12a,b shows 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.

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 × 30 min), two competition trials (2 × 15 min), and two rounds of the Ground challenge (2 × 25 min). 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

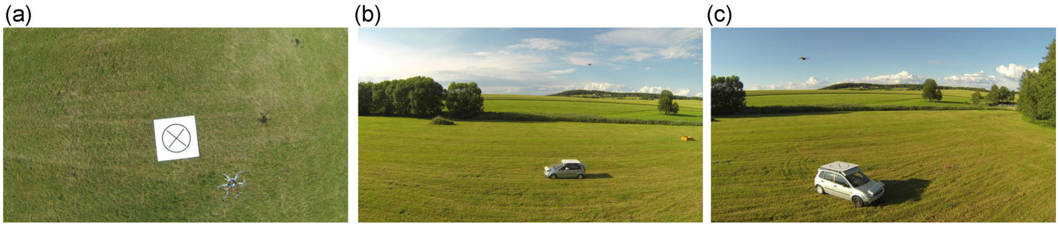


FIGURE 12 Photos from the first experiments in the field. (a) a UAV hovering above a static target, (b) a UAV tracking a vehicle moving along a linear trajectory, and (c) first attempts with tracking a ground vehicle moving in circles. The linear Kalman filter was used to estimate the states of the ground vehicle during this stage of development. Additional video material documenting these experiments can be found at http://mrs.felk.cvut.cz/ jfr2018landing-video1. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

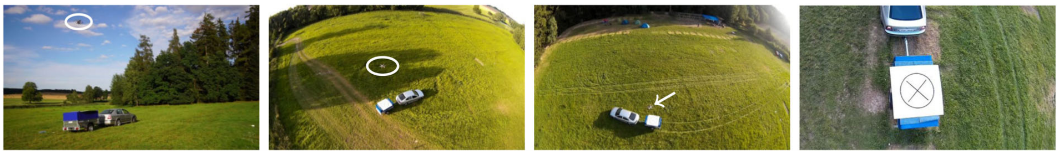


FIGURE 13 An unmanned aerial vehicle following the ground vehicle at a speed of 10 km/hr. The motion of the car was estimated by the linear Kalman filter. The system was thoroughly tested on general trajectories of the ground vehicle. Video material for this experiment can be found at http://mrs.felk.cvut. cz/jfr2018landing-video2 [Color figure can be viewed at wileyonlinelibrary.com]

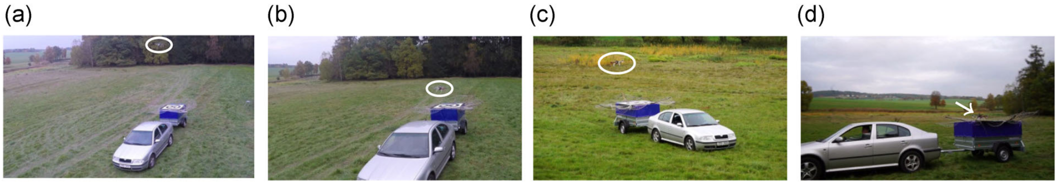


FIGURE 14 Experiments aimed at the autonomous following and landing on the ground vehicle using car-like motion model. Various stages of the landing state machine (see Section 5) are shown. (a) The UAV tracking the car as it aligns horizontally, (b) the descending phase, (c) the UAV aligns for the second 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. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

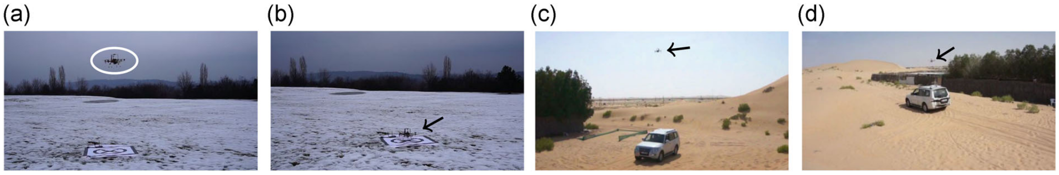


FIGURE 15 The performance of the system was tested in all weather conditions. Experiments in snowy and desert environments helped to fine-tune the computer vision for various lighting conditions and ground textures. A video showing a landing in a desert car be seen at http:// mrs.felk.cvut.cz/ jfr2018landing-video4 [Color figure can be viewed at wileyonlinelibrary.com]

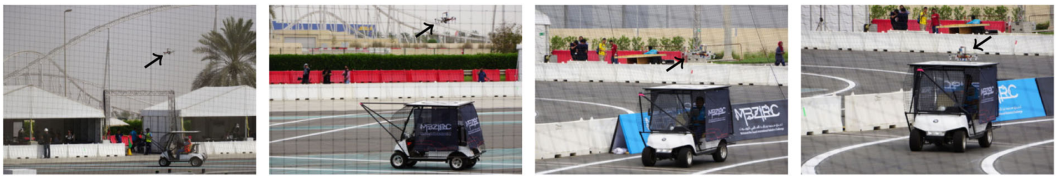


FIGURE 16 A sequence of images from the second trial during the competition. The whole videos, as well as additional material including onboard footage, can be found at http://mrs.felk.cvut.cz/jfr2018landing [Color figure can be viewed at wileyonlinelibrary.com]

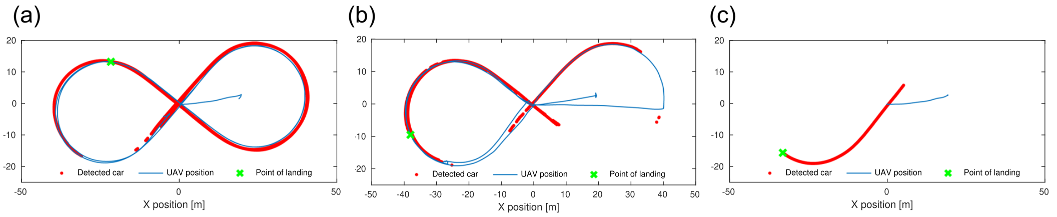


FIGURE 17 Top-down view of all three successful trials: (a) 1:44 min flight, (b) 1:28 min flight, and (c) 0:25 min flight [Color figure can be viewed at wileyonlinelibrary.com]

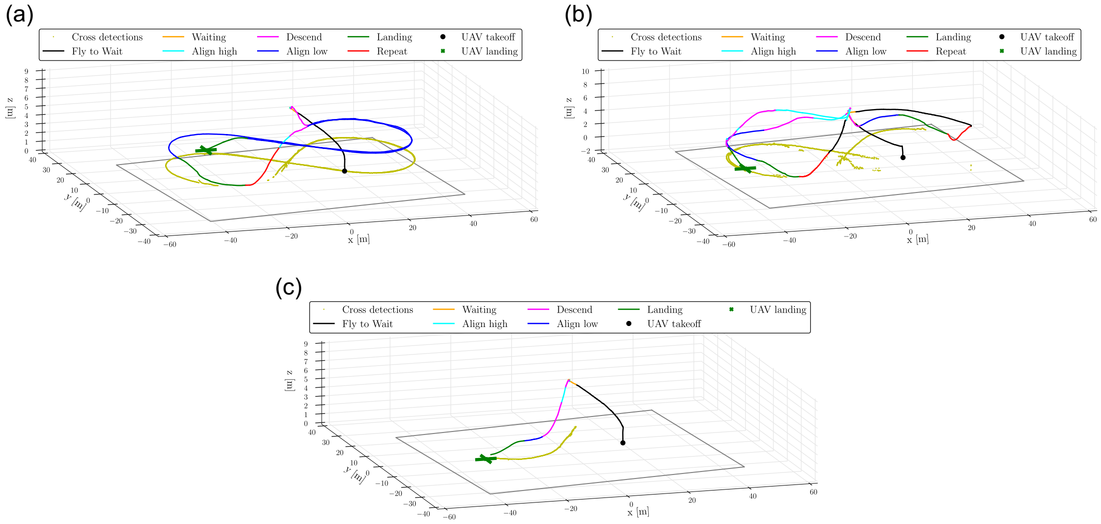


FIGURE 18 Three-dimensional plot of positions of the ground vehicle, as detected by the UAV, and the UAV during the (a) first competition trial, (b) second competition trial, and (c) first Grand Challenge trial. The trajectory of the UAV is color-coded according to the states of the *landing state machine*. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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 Figure 18a–c, with the states of the landing state machine (Section 5) color-coded in the trajectory of the UAV (Figure 19).

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

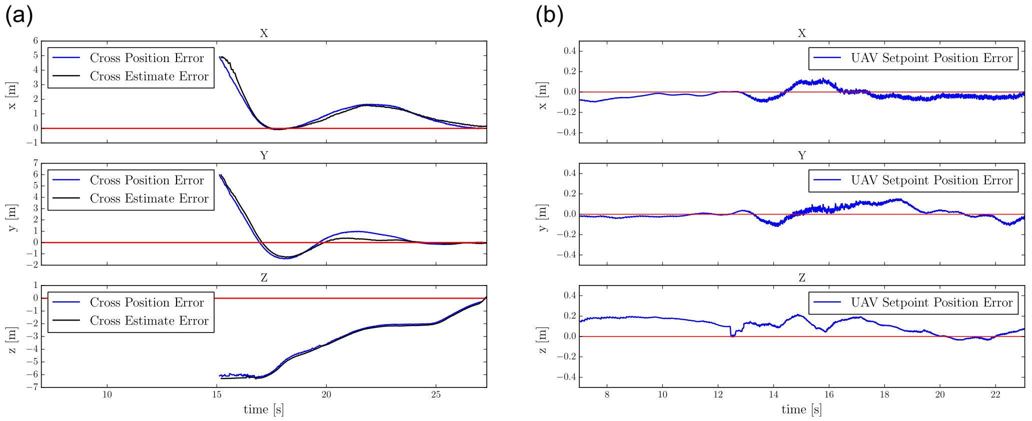


FIGURE 19 Plots of (a) the relative position of the detected and estimated ground vehicle, and the position of the UAV during the third (the fastest) landing and (b) the position control error during the flight. (a) Relative position of the car to the UAV and (b) position control error. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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 subsystems 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, for example, based on nowadays popular artificial neural networks. Estimation and prediction of the car movement using a nonlinear 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 the 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 UKF, 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 MPC tracker creates an optimal feed-forward reference in third-order dynamics based on the predicted trajectory. Nonlinear 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 1 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/hr. 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.

FIGURE 20 Team members after the award ceremony of the Mohamed Bin Zayed International Robotics Challenge competition in Abu Dhabi, United Arab Emirates [Color figure can be viewed at wileyonlinelibrary.com]

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 no. N000140710829, N000141410510, Army Research Laboratory grant no. W911NF1720181, and Khalifa University of Science, Technology and Research grant no. MBZIRC 2017.

ORCID

Báča Tomáš http://orcid.org/0000-0001-9649-8277
Petr Stepan http://orcid.org/0000-0002-7444-3264
Spurný Vojtěch http://orcid.org/0000-0002-9019-1634
Daniel Hert http://orcid.org/0000-0003-1637-6806
Robert Penicka http://orcid.org/0000-0001-8549-4932
Martin Saska http://orcid.org/0000-0001-7106-3816
Justin Thomas http://orcid.org/0000-0002-4253-417X
Giuseppe Loianno http://orcid.org/0000-0002-3263-5401

REFERENCES

Araar, O., Aouf, N., & Vitanov, I. (2017). Vision based autonomous landing of multirotor uav on moving platform. *Journal of Intelligent and Robotic Systems*, *85*(2), 369–384.

Baca, T., Hert, D., Loianno, G., Saska, M., & Kumar, V. (2018, October). *Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles*. IIEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid.

Baca, T., Loianno, G., & Saska, M. (2016, August-September). *Embedded model predictive control of unmanned micro aerial vehicles*. International Conference on Methods and Models in Automation and Robotics (MMAR), Midzyzdroje, Poland.

Benini, A., Rutherford, M. J., & Valavanis, K. P. (2016, May). *Real-time, gpu-based pose estimation of a uav for autonomous takeoff and landing*. International Conference on Robotics and Automation (ICRA), Stockholm, Sweden: IEEE, pp. 3463–3470.

Beul, M., Houben, S., Nieuwenhuisen, M., & Behnke, S. (2017, September). *Fast autonomous landing on a moving target at mbzirc*. The European Conference on Mobile Robotics (ECMR), Paris, France: IEEE.

Bi, Y., & 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-VanNguyen, A., Nguyen, D. Q., Saussié, D., & LeNy, 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., & Shi, C. (2016, August). *Autonomous landing of a quadrotor on an ugv*. International Conference on Mechatronics and Automation (ICMA), Harbin, China: IEEE, pp. 988–993.

Ghamry, K. A., Dong, Y., Kamel, M. A., & Zhang, Y. (2016, June). *Real-time autonomous take-off, tracking and landing of uav on a moving ugv platform*. 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece: IEEE, pp. 1236–1241.

Ghommam, J., & Saad, M. (2017). Autonomous landing of a quadrotor on a moving platform. *Transactions on Aerospace and Electronic Systems*, *53*, 1504–1519.

Guo, Z., & Hall, R. W. (1989). Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, *32*(3), 359–373.

Herissé, B., Hamel, T., Mahony, R., & 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., & Panagou, D. (2017, May). *Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle*. American Control Conference (ACC), IEEE, Seattle, WA, pp. 5580–5585.

Hui, C., Yousheng, C., Xiaokun, L., & Shing, W. W. (2013, July). *Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision*. Chinese Control Conference (CCC), Xi'an, China: IEEE, pp. 5895–5901.

Jin, S., Zhang, J., Shen, L., & Li, T. (2016, July). *On-board vision autonomous landing techniques for quadrotor: A survey*. 35th Chinese Control Conference (CCC), Chengdu, China: IEEE, pp. 10284–10289.

Jung, W., Kim, Y., & Bang, H. (2016, June). *Target state estimation for vision-based landing on a moving ground target*. International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA: IEEE, pp. 657–663.

Jung, Y., Lee, D., & Bang, H. (2015, August). *Close-range vision navigation and guidance for rotary uav autonomous landing*. International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden: IEEE, pp. 342-347.

Kim, J., Jung, Y., Lee, D., & Shim, D. H. (2014, May). *Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera*. International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL: IEEE, pp. 1243–1252.

Kong, W., Zhou, D., Zhang, D., & Zhang, J. (2014, September). *Vision-based autonomous landing system for unmanned aerial vehicle: A survey*. International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), San Diego, CA: IEEE, pp. 1–8.

Lee, D., Ryan, T., & Kim, H. J. (2012, May). *Autonomous landing of a vtol uav on a moving platform using image-based visual servoing*. International Conference on Robotics and Automation (ICRA), St. Paul, MN: IEEE, pp. 971–976.

Lee, H., Jung, S., & Shim, D. H. (2016, June). *Vision-based uav landing on the moving vehicle*. International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA: IEEE, pp. 1–7.

Lee, T., Leok, M., & 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., & 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., & Zell, A. (2014). A cross-platform comparison of visual marker based approaches for autonomous flight of quadrocopters. *Journal of Intelligent and Robotic Systems*, *73*(1-4), 349–359.

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, May). *Minimum snap trajectory generation and control for quadrotors*. International Conference on Robotics and Automation (ICRA), Shanghai, China: IEEE, pp. 2520–2525.

Mellinger, D., Shomin, M., Michael, N., & Kumar, V. (2013). *Cooperative grasping and transport using multiple quadrotors*. The 10th International Symposium Distributed Autonomous Robotic Systems, Springer tracts in advanced robotics, Springer, Berlin Heidelberg, Vol. 83, pp. 545–558.

Rossiter, J. A. (2003). *Model-based predictive control: A practical approach*. Boca Raton, FL: CRC Press.

Ruffo, M., DiCastro, M., Molinari, L., Losito, R., Masi, A., Kovermann, J., & Rodrigues, L. (2014, September). *New infrared time-of-flight measurement sensor for robotic platforms*. 18th International Workshop on ADC Modelling and Testing, Benevento, Italy: IMEKO.

Spurny, V., Baca, T., Saska, M., Penicka, R., Krajnik, T., Thomas, J., & 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*, *36*, 125–148.

Stepan, P., Krajnik, T., Petrlik, M., & Saska, M. (2018). Vision techniques for on-board detection, following, and mapping of moving targets. *Journal of Field Robotics*, *36* 252–269.

Tan, C. K., Wang, J., Paw, Y. C., & 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. Retrieved from https://www.tersus-gnss.com. Accessed September 15, 2017.

Wann, E. A., & van der Merwe, R. (2000, October). *The unscented kalman filter for nonlinear estimation*. Adaptive Systems for Signal Processing, Communications, and Control Symposium, Alberta, Canada: IEEE, pp. 153–158.

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.

Xu, L., & Luo, H. (2016, June). *Towards autonomous tracking and landing on moving target*. International Conference on Real-time Computing and Robotics (RCAR), Angkor Wat, Cambodia: IEEE, pp. 620–628.

Yang, S., Scherer, S. A., & Zell, A. (2013). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent and Robotic Systems*, *69*, 499–515.

Yang, S., Ying, J., Lu, Y., & Li, Z. (2015, May). *Precise quadrotor autonomous landing with srukf vision perception*. International

Conference on Robotics and Automation (ICRA), Seattle, WA: IEEE, pp. 2196–2201.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Báča T, Stepan P, Spurný V, et al. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *J Field Robotics.* 2019;36:874–891. https://doi.org/10.1002/rob.21858

Received: 15 October 2017 | Revised: 10 April 2018 | Accepted: 12 July 2018

DOI: 10.1002/rob.21816

SYSTEMS ARTICLE

WILEY

Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles

Vojtěch Spurný[1] | Tomáš Báča[1] | Martin Saska[1] | Robert Pěnička[1] | Tomáš Krajník[2] | Justin Thomas[3] | Dinesh Thakur[3] | Giuseppe Loianno[4] | Vijay Kumar[3]

[1]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic

[2]Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic

[3]GRASP Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania

[4]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 multirotor 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[1]). In the competition, the

[1] 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[2]. 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 rangefinder 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

[2] 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, Bülthoff, 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[3] 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 Bähnemann, 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 electro-permanent 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 Bähnemann 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[4] 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

[3] http://www.vicon.com/ - Accessed: July 17, 2018

[4] 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.

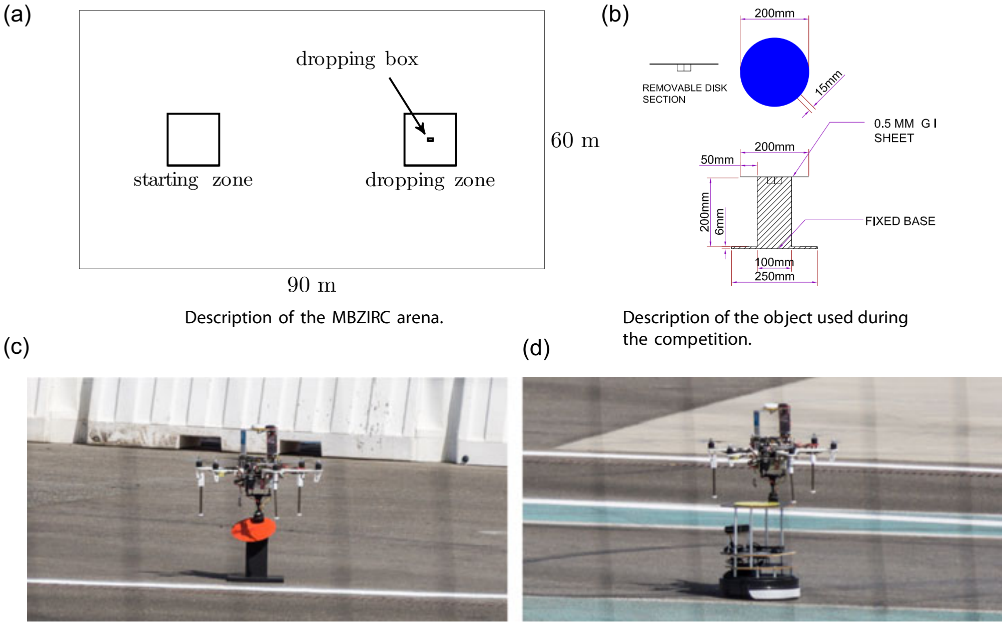
1.3 | Problem statement

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



Description of the MBZIRC arena.

Description of the object used during the competition.

Photo of a UAV grasping a static object from the stand.

Photo of a UAV grasping a dynamic object from a TurtleBot2 mobile robot.

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]

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

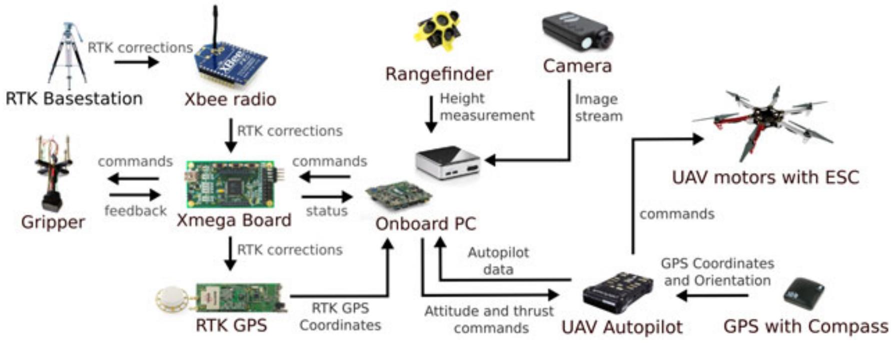


FIGURE 2 Description of components in our UAV platform. ESC: electronic speed controller; GPS: global positioning system; RTK: real-time kinematic; UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

GNSS RTK BOARD (GPS L1L2/GLONASS G1/BEIDOU B1B3) (Tersus-GNSS, 2017), with accuracy of 10 mm ± 1 parts per million (ppm) horizontally and 15 mm ± 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,

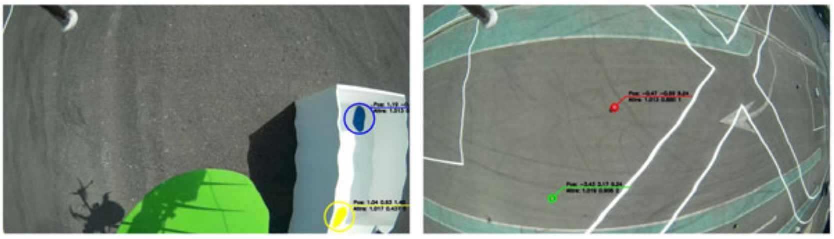


FIGURE 3 Object detection in onboard camera images affected by the 'jelly' or 'wobble' effect, which deforms lines (left image), as well as circular and square objects (right image). The detection results indicate the 3D relative position (top line) and attributes like roundness, eccentricity and type (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]

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 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

$$\begin{aligned}
\mathrm{x}^o_{[n+1]} &= \mathrm{x}^o_{[n]} + \dot{\mathrm{x}}^o_{[n]}\Delta t,\\
\dot{\mathrm{x}}^o_{[n+1]} &= \begin{pmatrix}\cos\phi_{[n+1]}\\ \sin\phi_{[n+1]}\end{pmatrix} v_{[n+1]},\\
\phi_{[n+1]} &= \phi_{[n]} + \dot{\phi}_{[n]}\Delta t,\\
\dot{\phi}_{[n+1]} &= K_{[n]}v_{[n]},\\
v_{[n+1]} &= v_{[n]} + a_{[n]}\Delta t,\\
K_{[n+1]} &= K_{[n]} + \dot{K}_{[n]}\Delta t,
\end{aligned} \tag{1}$$

where $\mathbf{x}^o_{[n]} = (x, y)^T_{[n]}$ is the position of the object 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 Δt 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

$$\mathrm{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathrm{B} = \begin{pmatrix} \Delta t \\ \Delta t \end{pmatrix}, \tag{2}$$

where $\mathbf{x}^e_{[n+1]} = \mathbf{A}\mathbf{x}^e_{[n]} + \mathbf{B}\mathbf{u}_{[n]}$ is the linear system equation, $\mathbf{x}^e_{[n]} = (x, y)^T_{[n]}$ is the state vector finally used for control, and $\mathbf{u}_{[n]}$ is the system input. According to our experience,

$$\mathrm{x}^p_{[0]} + \sum_{n=0}^{k} \dot{\mathrm{x}}^p_{[n]} \Delta t_{[n]} = \mathrm{x}^p_{[k]}, \quad \forall\, k \in \mathbb{N} \tag{3}$$

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 **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 $\mathbf{Q} = \mathrm{diag}(1, 1)$ and measurement covariance $\mathbf{R} = \mathrm{diag}(10e3, 10e3)$. 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 $[\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3]$ and a body reference frame centered in the center of mass of the vehicle with an orientation denoted by $\mathbf{R} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3]$, where $\mathbf{R} \in SO(3)$. The dynamic model of the vehicle can be expressed as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + mg\mathbf{e}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\mathbf{\Omega}}, \\ \mathbf{J}\dot{\mathbf{\Omega}} &+ \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} = \mathbf{M}, \end{aligned} \tag{4}$$

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, $\mathbf{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame, and $\mathbf{J} \in \mathbf{R}^{3\times3}$ 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 $\mathbf{e}_3 = [0 \quad 0 \quad 1]^\top$. The total moment $\mathbf{M} \in \mathbb{R}^3$, with $\mathbf{M} = [M_1 \quad M_2 \quad M_3]^\top$, 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 six propellers. The relationship between a single motor thrust f_j, the net thrust f, and the moments $\mathbf{M}$ can be written as

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ sd & 1 & sd & -sd & -1 & -sd \\ -cd & 0 & cd & cd & 0 & -cd \\ -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \tag{5}$$

where $c = \cos(30°)$, $s = \sin(30°)$, 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

$$\mathbf{M} = -k_R\mathbf{e}_R - k_\Omega\mathbf{e}_\Omega + \mathbf{\Omega} \times J\mathbf{\Omega} - J(\hat{\mathbf{\Omega}}\mathbf{R}^T\mathbf{R}_c\mathbf{\Omega}_c - \mathbf{R}^T\mathbf{R}_c\dot{\mathbf{\Omega}}_c), \tag{6}$$

$$\begin{aligned} f = -\Big(&-k_x\mathbf{e}_x - k_{ib}\mathbf{R}\int_0^t \mathbf{R}(\tau)^T\mathbf{e}_x d\tau - k_{iw}\int_0^t \mathbf{e}_x d\tau - k_v\mathbf{e}_v - mg\mathbf{e}_3 \\ &+ m\ddot{\mathbf{x}}_d\Big) \cdot \mathbf{R}\mathbf{e}_3 = \mathbf{f} \cdot \mathbf{R}\mathbf{e}_3, \end{aligned} \tag{7}$$

with $\ddot{\mathbf{x}}_d$ the desired acceleration, and k_{iw}, k_{ib}, k_x, k_v, k_R, k_Ω 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 $R_C = [\mathbf{b}_{1,C} \quad \mathbf{b}_{2,C} \quad \mathbf{b}_{3,C}]$ is calculated as

$$\begin{aligned} &\mathbf{b}_{2,des} = [-\sin\psi_{des}, \quad \cos\psi_{des}, \quad 0]^\top, \qquad \mathbf{b}_{3,C} = \frac{\mathbf{f}}{||\mathbf{f}||}, \\ &\mathbf{b}_{1,C} = \frac{\mathbf{b}_{2,des} \times \mathbf{b}_3}{||\mathbf{b}_{2,des} \times \mathbf{b}_3||}, \qquad \mathbf{b}_{2,C} = \mathbf{b}_3 \times \mathbf{b}_1. \end{aligned} \tag{8}$$

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

$$\begin{aligned} \mathbf{e}_R = \tfrac{1}{2}(\mathbf{R}_C{}^\top\mathbf{R} - \mathbf{R}^\top\mathbf{R}_C)^\vee, \quad \mathbf{e}_\Omega &= \mathbf{\Omega} - \mathbf{R}^\top\mathbf{R}_C\mathbf{\Omega}_C, \\ \mathbf{e}_x = \mathbf{x} - \mathbf{x}_d, \quad \mathbf{e}_v &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_d, \end{aligned} \tag{9}$$

represent the orientation, the angular rate errors, and the translation errors, respectively. The symbol $.^\vee$ represents the *vee* map $\mathrm{so}(3) \rightarrow \mathbb{R}^3$. If the initial attitude error is less than 90°, the zero equilibrium of the tracking error is exponentially stable, that is, $[\mathbf{e}_x^\top \quad \mathbf{e}_v^\top \quad \mathbf{e}_\Omega^\top \quad \mathbf{e}_R^\top]^\top \equiv [\mathbf{0}^\top \quad \mathbf{0}^\top \quad \mathbf{0}^\top \quad \mathbf{0}^\top]^\top$.

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 $\mathbf{x}_{[n]} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z})^T_{[n]}$ by minimizing the function

$$\begin{aligned} V(\mathbf{x}_{[0,\ldots,m-1]}, \mathbf{u}_{[0,\ldots,m-1]}) \quad &= \quad \frac{1}{2}\sum_{i=1}^{m-1}\left(\mathbf{e}_{[i]}^T\mathbf{Q}\mathbf{e}_{[i]} + \mathbf{u}_{[i]}^T\mathbf{P}\mathbf{u}_{[i]}\right), \\ \text{s.t.} \quad & \mathbf{x}_{[0,\ldots,m-1]} \geq \mathbf{x}_L, \\ & \mathbf{x}_{[0,\ldots,m-1]} \leq \mathbf{x}_U, \end{aligned} \tag{10}$$

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 $\mathbf{x}_K$ 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 $\tilde{\mathbf{x}}$ and the reference, which is generated by the MPC tracker. The trajectory setpoint $\tilde{\mathbf{x}}$ is provided by an operator or a program. No requirements are imposed on $\tilde{\mathbf{x}}$. 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, Δt 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 Δt 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

$$\mathbf{U} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 5 \quad 5 \quad 5 \quad 5 \quad 5 \quad 10), \tag{11}$$

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.

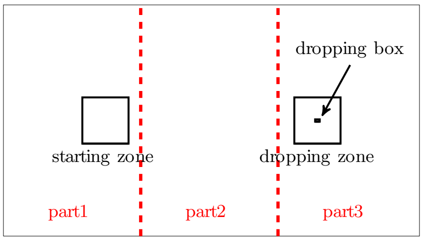


FIGURE 4 Decomposition of the Mohamed Bin Zayed International Robotics Challenge arena into three equally large zones [Color figure can be viewed at wileyonlinelibrary.com]

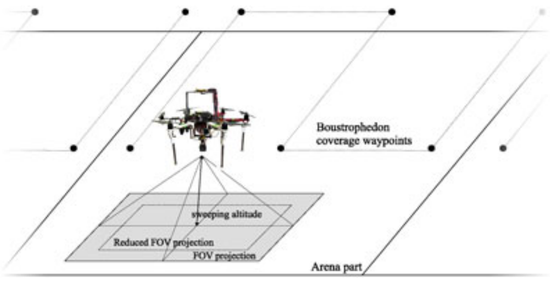


FIGURE 5 Boustrophedon coverage of the decomposed competition arena. FOV: field of view [Color figure can be viewed at wileyonlinelibrary.com]

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 (~ 3 ms^{-1}) and the maximal acceleration of the UAV a_{max} (~ 2 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 (~ 7 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

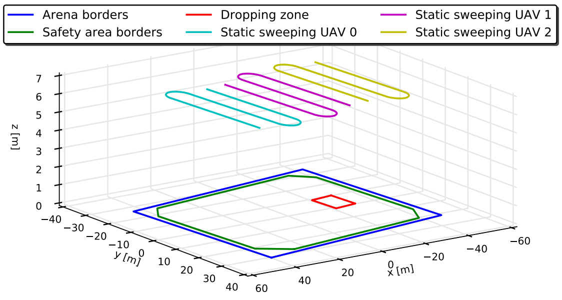


FIGURE 6 Sweeping trajectories based on Boustrophedon coverage using the Dubins vehicle and decomposition of the arena into three distinct parts, one for each UAV. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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).

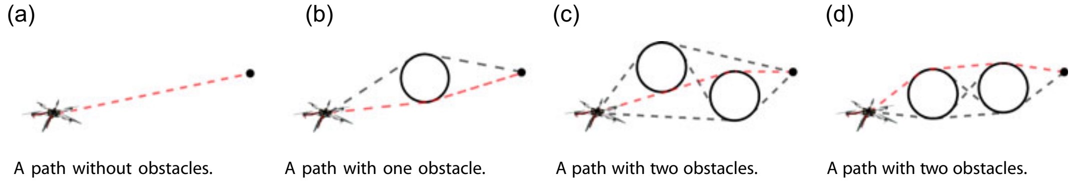
Thanks to these assumptions, the method for very rapidly computing a collision-free trajectory can be simplified to finding a collision-free path in 2D (at the dedicated height) between two points, where only two obstacles can occur. These obstacles are circles centered on the x and y coordinates of 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



A path without obstacles. A path with one obstacle. A path with two obstacles. A path with two obstacles.

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]

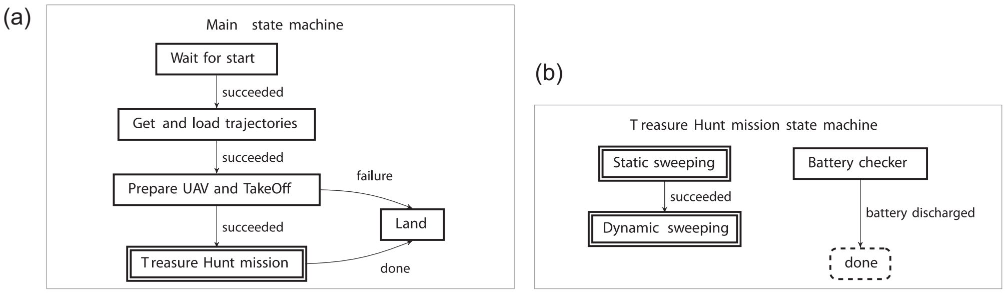


Diagram of the main state machine.

Diagram of the Treasure Hunt mission state machine.

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

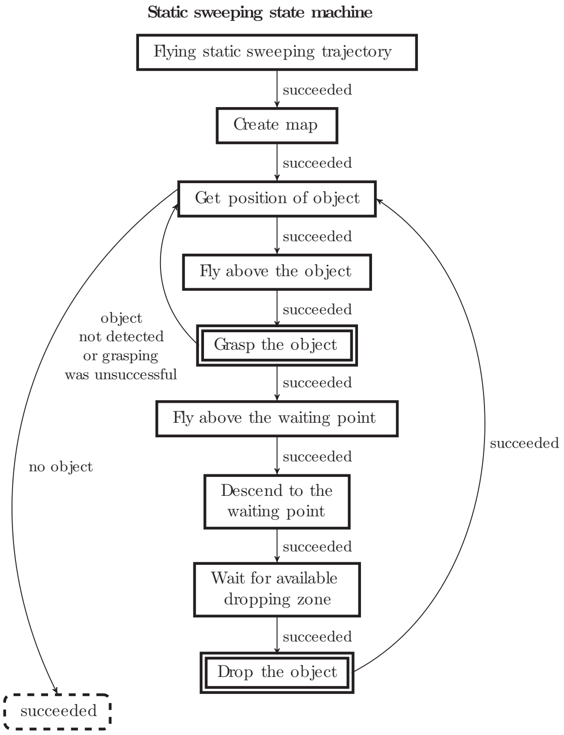

Static sweeping state machine
Flying static sweeping trajectory
succeeded
Create map
succeeded
Get position of object
succeeded
Fly above the object
succeeded
object not detected or grasping was unsuccessful
Grasp the object
succeeded
Fly above the waiting point
succeeded
no object
Descend to the waiting point
succeeded
Wait for available dropping zone
succeeded
Drop the object
succeeded
succeeded

FIGURE 9 Diagram of the static sweeping state machine

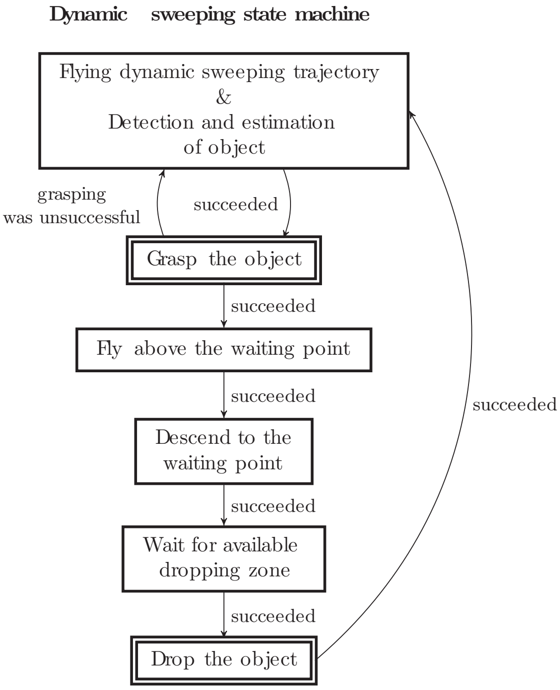
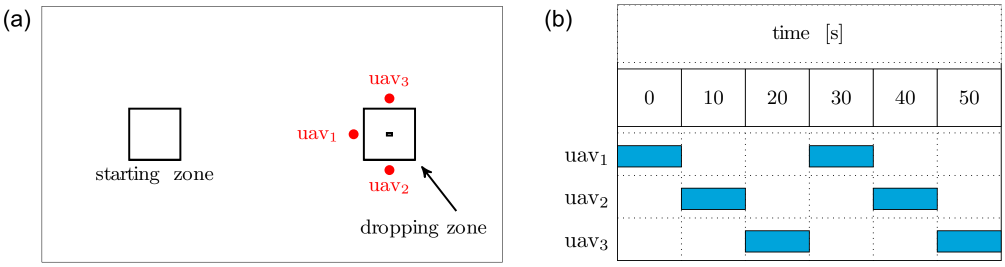

Dynamic sweeping state machine
Flying dynamic sweeping trajectory & Detection and estimation of object
grasping was unsuccessful
succeeded
Grasp the object
succeeded
Fly above the waiting point
succeeded
Descend to the waiting point
succeeded
Wait for available dropping zone
succeeded
Drop the object
succeeded

FIGURE 10 Diagram of the dynamic sweeping state machine



Position of the waiting point in the MBZIRC arena.

Gantt diagram of the time window for particular UAVs.

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]

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.

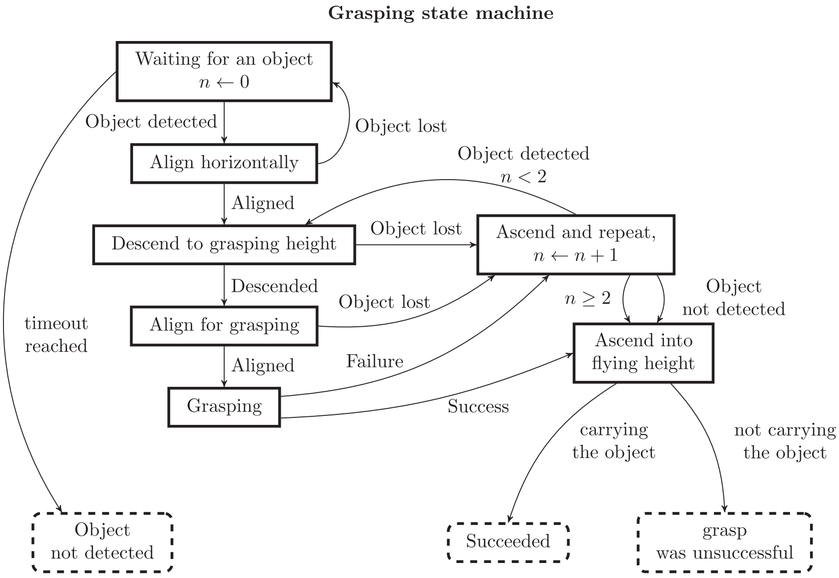


FIGURE 12 Diagram of the grasping state machine

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 × 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 × 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 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.

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.

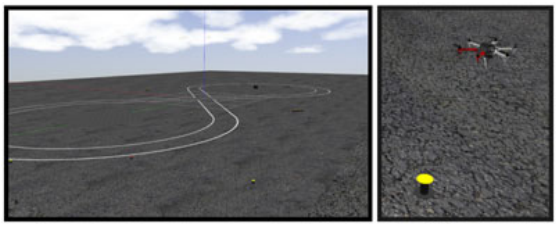
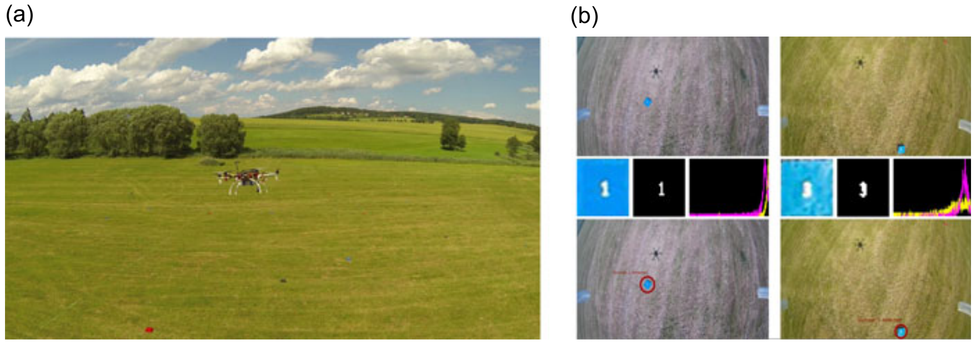


FIGURE 13 Snapshots from the simulation developed for the Mohamed Bin Zayed International Robotics Challenge competition [Color figure can be viewed at wileyonlinelibrary.com]

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



MPC-based trajectory tracking with low-level UAV control during the mapping of the colored objects spread throughout the experimental field.

Object detection and number (reward) recognition of square-shaped colored objects.

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]

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 (~ 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

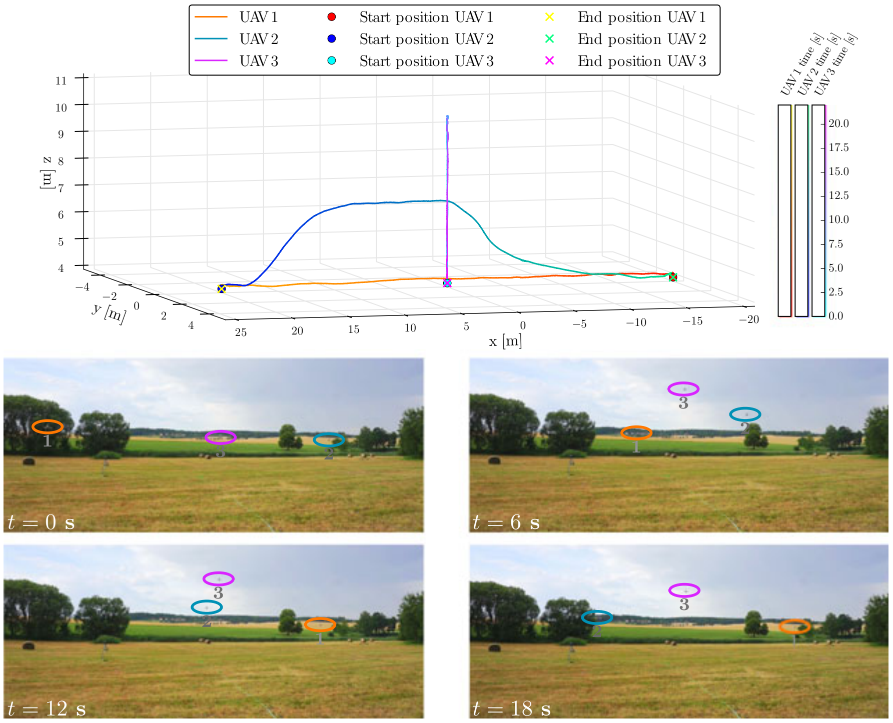


FIGURE 15 MPC-based collision avoidance between three drones. Two drones (UAV 1 and UAV 2) exchange their positions, while the third UAV 3 hovers in a position colliding with their trajectories. Using MPC future trajectory prediction, the UAVs avoid a collision by changing their trajectory height. http://mrs.felk.cvut.cz/jfr2018treasurehunt-video2. 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

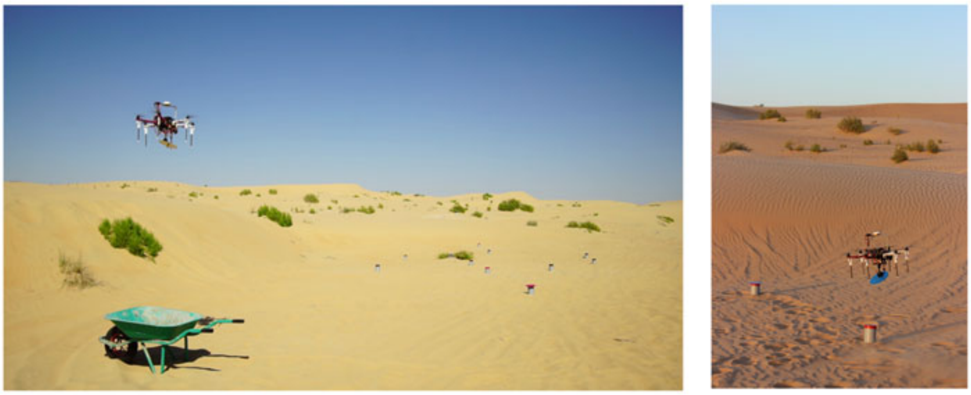


FIGURE 16 Photos from the tests of the proposed system in the desert near Abu Dhabi, United Arab Emirates. The grasping procedure is captured in the image on the right, and the dropping maneuver is shown in the image on the left. http://mrs.felk.cvut.cz/jfr2018treasurehunt-video3 [Color figure can be viewed at wileyonlinelibrary.com]

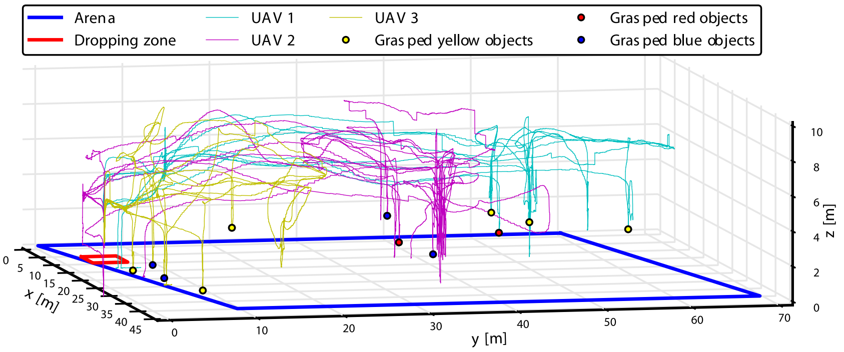


FIGURE 17 The paths traveled by individual UAVs during the desert experiment. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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

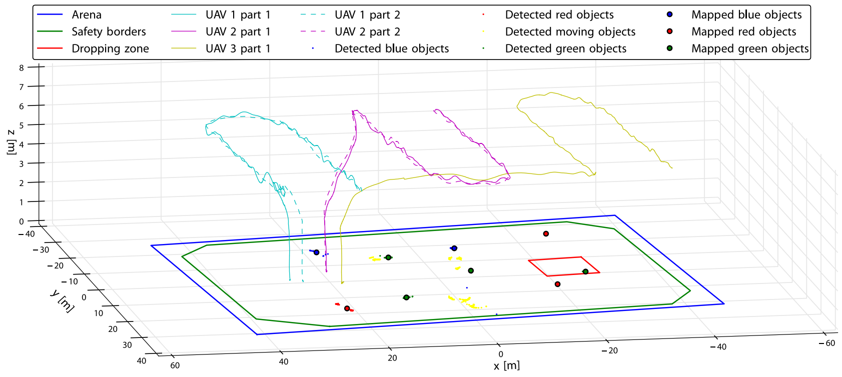


FIGURE 18 Mapping sweep during the first trial of Challenge 3. The colored points denote the detections of the objects that were observed, and the larger circles denote the estimated positions of these objects. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

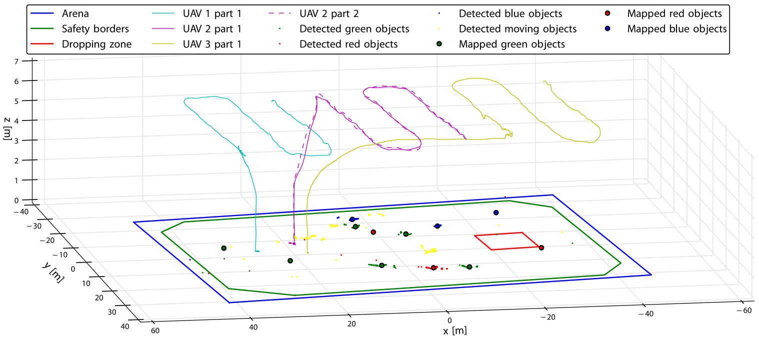
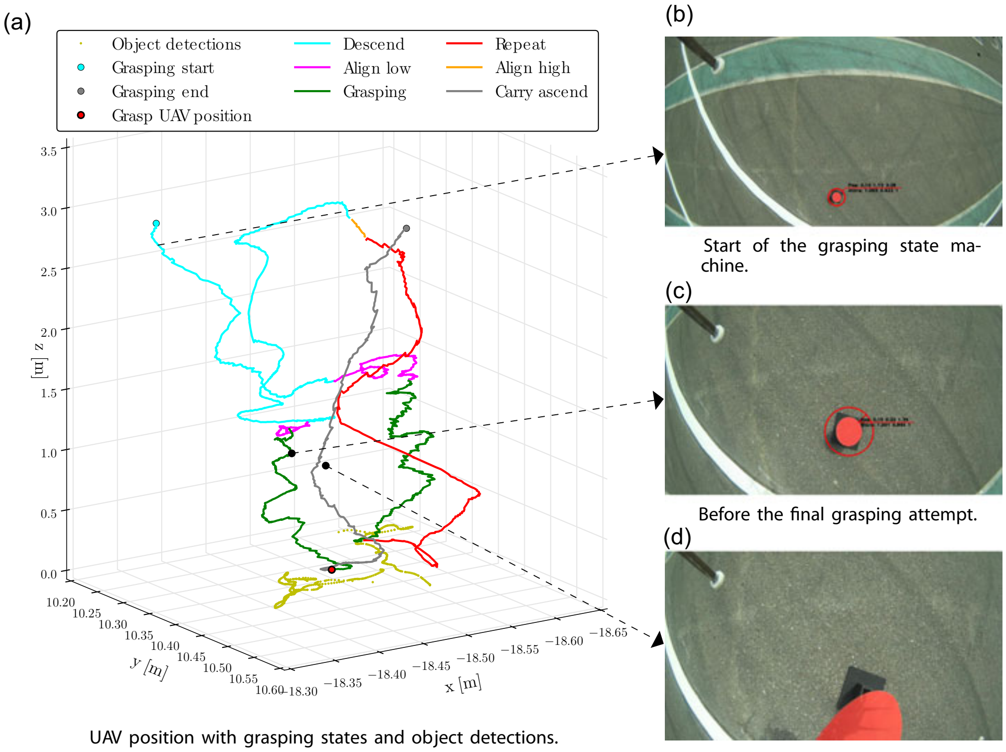


FIGURE 19 Mapping sweep during the second trial of Challenge 3. The colored points denote the detections of the objects that were observed, and the larger circles denote the estimated positions of these objects. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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.



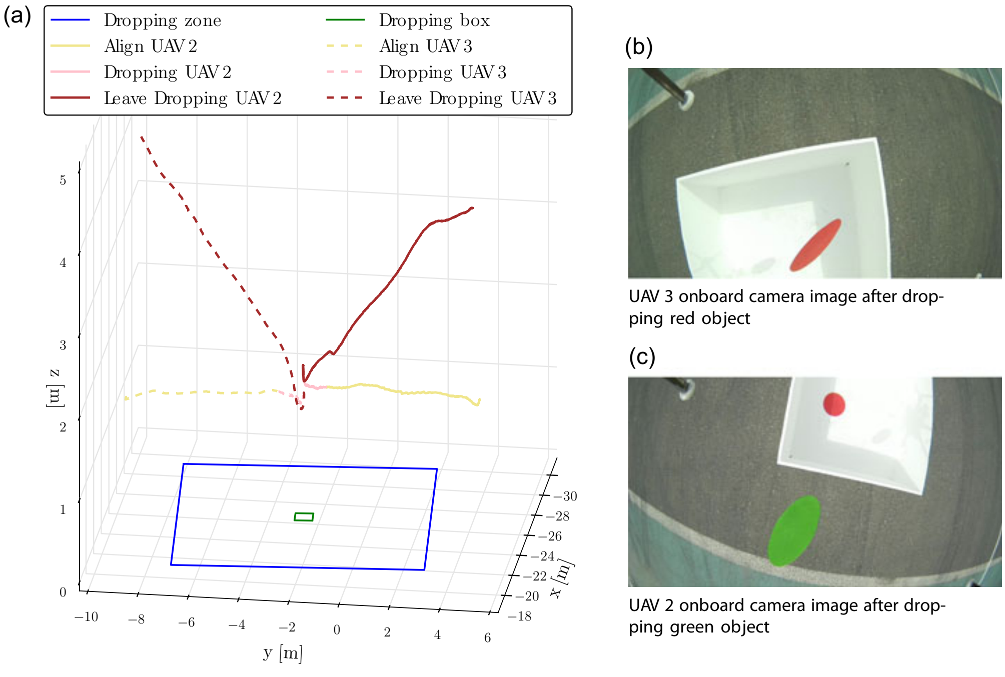
UAV position with grasping states and object detections.

Start of the grasping state machine.

Before the final grasping attempt.

Carrying a grasped object.

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]



UAV 3 onboard camera image after dropping red object

UAV 2 onboard camera image after dropping green object

UAV positions with dropping states.

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

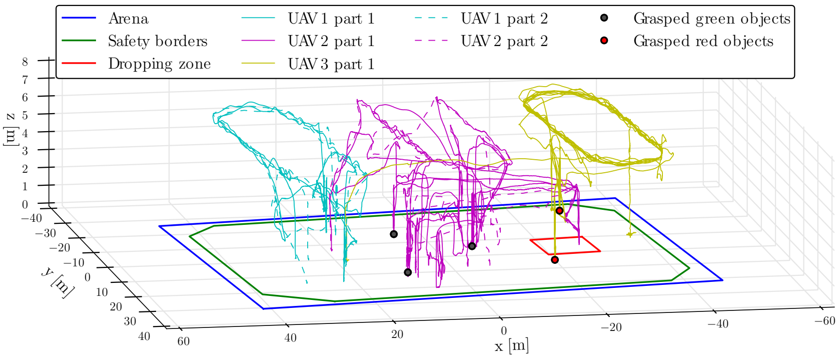


FIGURE 22 Paths traveled by individual UAVs during the first trial of Challenge 3. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

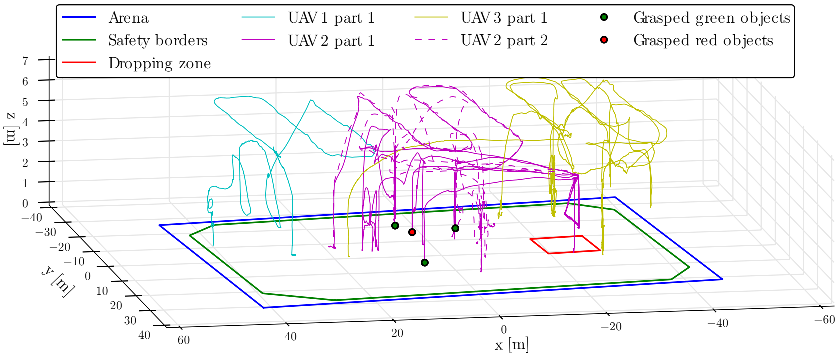


FIGURE 23 Paths traveled by individual UAVs during the second trial of Challenge 3. The colored points denotes the positions of the objects that were collected. UAV: unmanned aerial vehicle [Color figure can be viewed at wileyonlinelibrary.com]

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.

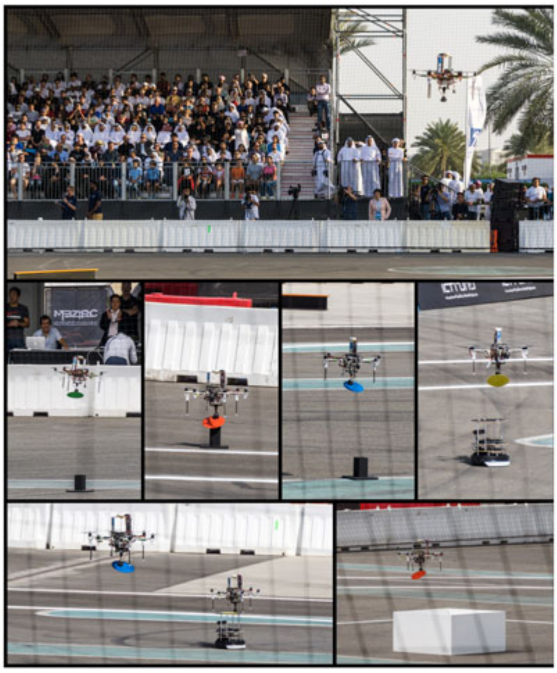


FIGURE 24 Photos from the MBZIRC competition. The upper image shows the UAV while following the static sweeping trajectory. The images in the middle row and the left on the lower part of the figure capture the moments when the UAVs were grasping objects. The remaining image shows an object being dropped 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 subsystems 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.

Bähnemann, R., Schindler, D., Kamel, M., Siegwart, 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., Přeučil, 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). Dubinsorienteering 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., Bülthoff, 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_fkie - 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*. 2019;36:125–148. https://doi.org/10.1002/rob.21816

Autonomous Cooperative Wall Building by a Team of Unmanned Aerial Vehicles in the MBZIRC 2020 Competition

Tomas Baca[a,1,c], Robert Penicka[a,c], Petr Stepan[a,c], Matej Petrlik[a], Vojtech Spurny[a], Daniel Hert[a], Martin Saska[a]

[a]*Authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague 160 00.*
[b]*Corresponding author, e-mail: tomas.baca@fel.cvut.cz.*
[c]*Authors contributed equally.*

Abstract

This paper presents a system for autonomous cooperative wall building with a team of Unmanned Aerial Vehicles (UAVs). The system was developed for Challenge 2 of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020. The wall building scenario of Challenge 2 featured an initial stack of bricks and wall structure where the individual bricks had to be placed by a team of three UAVs. The objective of the task was to maximize collected points for placing the bricks within the restricted construction time while following the prescribed wall pattern. The proposed approach uses initial scanning to find a priori unknown locations of the bricks and the wall structure. Each UAV is then assigned to individual bricks and wall placing locations and further perform grasping and placement using onboard resources only. The developed system consists of methods for scanning a given area, RGB-D detection of bricks and wall placement locations, precise grasping and placing of bricks, and coordination of multiple UAVs. The paper describes the overall system, individual components, experimental verification in demanding outdoor conditions, the achieved results in the competition, and lessons learned. The presented CTU-UPenn-NYU approach achieved the overall best performance among all participants to won the MBZIRC competition by collecting the highest number of points by correct placement of a high number of bricks.

1. Introduction

UAVs belong to one of the most studied topics in the field of robotics due to the numerous possible applications. One of the possible areas of the UAV deployment is in construction [1] where the UAVs can, for example, visually inspect existing construction sites, survey areas before construction starts, or monitor security and safety of the sites [2, 3]. This paper goes beyond these works to present a fully autonomous system enabling physical interaction and not only inspection. The UAVs directly take part in the construction and are used for building walls. The proposed multi-robot system is designed in order to autonomously build walls with only a little a priori knowledge of the construction site. The system uses onboard detection of bricks and wall structure locations using carried camera and depth sensors. An initial scan of the construction area is conducted using one UAV to find the locations of the brick stack and of the wall building site. Afterwards, each UAV is assigned to a particular stack part and wall segment to then cooperatively build the wall according to a given wall pattern. During the building process, each UAV repeatedly attempts to grasp a brick from the assigned stack, delivers the brick above the designated segment of the wall, and then precisely places the brick on the wall. Figure 1 illustrates the scanning, grasping, and placing subtasks of the wall building.

The proposed system was developed by the joint CTU-UPenn-NYU[1] team for the participation in Challenge 2 of the MBZIRC 2020 [4]. Challenge 2 consisted of the wall building task where three UAVs and one Unmanned Ground Vehicle (UGV) were assigned to autonomously build two walls — one by the UGV and one by the UAVs. This paper presents details of the system used for the UAV part of the challenge. The challenge featured brick stacking for the UAVs containing 46 bricks, each with ferromagnetic plate on top to facilitate grasping. Four types of bricks were present, each with different color, weight, length, and earned points for placement. The future wall structure for the UAVs consisted of four segments arranged in a 'W' letter shape located 1.7 m above ground, capable of containing all the bricks from the stack in only two layers. The goal of the challenge was to maximize collecting points by autonomously placing the bricks on the wall according to a given wall pattern in a given time limit.

The wall building approach by the CTU-UPenn-NYU team exhibited the best performance among all participants of the MBZIRC 2020 Challenge 2. During the two competition trials, each with a duration of 25 minutes, the UAVs were able to grasp a total number of 17 bricks and successfully place ten of them. The UGV helped by placing one brick to fulfill the requirements for winning the challenge. The CTU-UPenn-NYU team was thus able to place the most bricks among the participants to achieve a score of 8.24- far higher than the second best team with a score of 1.33 points.

The solution proposed for the wall building task consists of three main autonomous capabilities of the UAVs. The first is

[1] Collaboration of the Czech Technical University in Prague, University of Pennsylvania, and New York University.

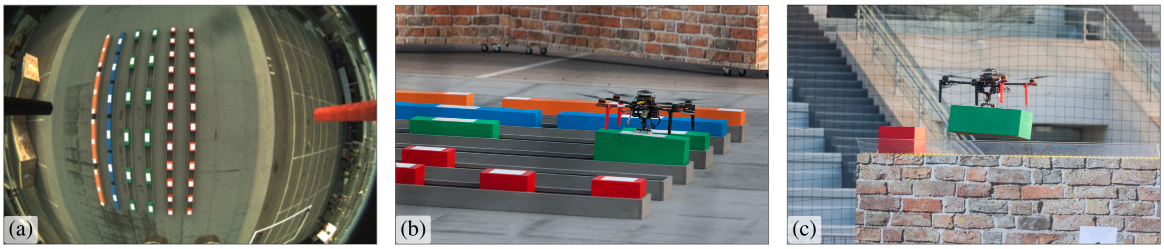


Figure 1: Illustration of the proposed UAV system for the wall building task showing images captured from the onboard camera during scanning (a) and photos of UAV during brick grasping (b) and placing (c).

the scanning of the arena to find locations of the brick stack and the wall. All detections from the one UAV performing the scanning are used to create a topological map of the arena which is created by employing statistical analysis of the detections using known sizes and shapes of the brick stack and the wall segments. Distribution of the wall building task is then based on sharing the topological map among the UAVs using Wi-Fi communication and deterministic assignment of the brick stack and wall parts to individual UAVs. Each UAV then creates a plan to grasp and place according to a given wall pattern and assigned part, and afterward repeats grasping and placing until battery depletion or plan fulfillment.

The second primary capability is for brick grasping, which requires precise navigation of the UAV to the center of the brick marked by a white ferromagnetic plate. The most essential part of ensuring precise grasping is robust and fast brick detection. The color and Red-Green-Blue-Depth (RGB-D) cameras provide sufficient information about brick position from altitudes above the bricks and their fusion improves the robustness of detection. The duration of brick detection for grasping was no longer than 7 ms and thus allowed the use of a visual servoing technique during the final approach to increase grasping precision. A grasping state machine is used to govern various stages of approaching the brick, e.g., decides when to switch from Global Positioning System (GPS) localization to visual servoing. Finally, both the UAV estimated mass and attitude are checked during grasping by the UAV control system to abort grasping in close brick interactions when, e.g., the brick is grasped far from the center of mass or the UAV mass is transferred to the ground by its landing gear.

The last main autonomous capability is the placement of a brick to a desired position on the wall structure. This task is challenging as the grasped brick may influence wall detection due to the sensors possibly being obscured by the brick. The brick may further influence the UAV control system as additional brick mass could generate torque to the UAV if not grasped exactly above the center of mass. The brick concealing a significant amount both sensors' views is compensated immediately after successful grasping by removing such parts of sensory data during consequent placing. As only the dimensions of the wall structure were known a priori and not its position or orientation, the RGB-D camera alone is used for wall detection. The wall detection and computation of the placing position on the wall takes up to 10 ms. The brick placing uses its own dedicated state machine to manage various placement stages and considers that, e.g, only a part of the wall segment can be visible and the placement is planned to be on the leftmost free position on the wall.

The visual detection for autonomous wall building with drones has to be robust, fast, and with minimal computation demands. Detection during the scanning of the arena (i.e. looking for both bricks and walls) takes up to 15 ms, detection of brick with known color takes up to 5 ms, and the computation of brick placement on the wall takes up to 10 ms. The wall detection pipeline first detects the ground in the RGB-D data by creating a histogram of measured distances to ground plane transformed from RGB-D data using measurements from the Inertial Measurement Unit (IMU). A number of highest distance values in the histogram are used as altitude measured by the RGB-D sensor. Thresholding of the distances to ground plane using the measured altitude is then used to a create a binary image with possible wall detections. Finally, the wall segments are verified by examining contour lines of the possible detections to be parallel and in distance approximately equal to wall width. Visual recognition of the bricks is mainly based on white plate detection using color segmentation applied to Hue, Saturation, Value (HSV) image from the color camera. The contours of white segments are transformed to a plane parallel to the ground plane in altitude equal to brick height. Such transformed contours are then checked for the size of the white plates. Finally, additional color thresholding of the HSV image is used to identify different types of the bricks. All detection functions take only one thread on the onboard computer, and therefore allow enough computation power for the rest of the system, e.g., control algorithms.

Automatic control of the UAV motion is vital for the precise grasping and placing of bricks. We build upon our success from the first MBZIRC 2017 challenge for which we developed a hybrid Model Predictive Control (MPC) tracking controller [5]. An MPC feedforward tracker is coupled with the geometric tracking controller [6] to minimize a control error around the pre-planned differentially flat dynamics and to provide us with attitude tracking. The tracking controller is part of the provided open-source UAV system [7]. The UAV system allows the use a visual servoing technique to estimate the states of the UAV directly using observations of an object, i.e., the brick. With the visual servoing, the control feedback loop is closed using only the camera-based data and the onboard IMU. The visual servoing removes the inaccurate GPS localization from the loop for the duration of the grasping manoeuvre, significantly increasing the accuracy of the grasping manoeuvre to an order of centimeters. We empirically verified that relying on a traditional GPS introduces a significant localization position drift. The potential GPS drift impacts the UAV control performance to the extent of making a precise grasping manoeuvre an unfavorable probabilistic event. Furthermore, we employed a real-time scheduling of controller gains and dynamic constraints to satisfy the varying conditions during the various stages of the mission. This was especially important during the transitions between the GPS and visual servoing stages of the flight where the UAV feedback loop exhibited different properties, mainly due to changing noise and delay characteristics of the UAV state estimate.

In this paper, we present the overall approach and system that won, by a significant margin, Challenge 2 of the MBZIRC 2020 in autonomously placing the most bricks on the wall. The vision techniques for brick and wall detection that allowed for precise vision-based grasping and placing have been detailed in this paper. Description of the UAV control used in a closed-loop with the visual detection of the bricks and the wall is given. The multi-UAV cooperative wall building approach is described as well, including the state machine of individual UAVs, creation of a topological map of the arena, and the deterministic distribution of the multi-robot task. The whole system is open-sourced[2] to allow the community to further build on our successful system. Finally, the results from the competition along the lessons learned are described.

The remainder of this work is organized as follows — the rest of this section begins with an overview of related literature works and then details the MBZIRC Challenge 2. Section 2 introduces the hardware platform used for the wall building task. The UAV control system is presented in Sec. 3.1. The overall approach used for the task is then described in Sec. 4. Sections 5, 6, and 7 describe, in this order, the three most crucial parts of the wall building system: the arena scanning, brick grasping, and brick placing. Results achieved during the competition are discussed in Sec. 8 and conclusions are drawn in Section 9.

1.1. Related work

UAVs can be deployed in various scenarios in the field of construction [1]. Visual inspection of construction sites, area surveying prior to construction, and security and safety monitoring are examples of such tasks [3]. Inspection of existing structures, such as bridges [8], can also be considered among these scenarios. Nowadays, each of these tasks can be performed by considerably small UAVs that are manually piloted or semi-autonomous. However, UAVs participating directly in physical construction and operating autonomously are still being considered, mainly in lab-controlled environments.

Authors of [9] proposed a system for building cubic truss-like structures from simple nodes by a team of UAVs. The system relies on a motion tracking system. Additional work on the assembly of truss structures has been explored by the authors in [10]. The main focus is on a distributed construction algorithm to build a truss according to a given blueprint using a team of UAVs. An approach for building tensile structures, such as structures from ropes, using UAVs is presented in [11]. The paper focuses on creating trajectories for UAVs with respect to a built structure and on the UAV control required for building elements with tension forces. Building bridges with cooperating UAVs using the tensile ropes is further described in [12]. Trajectory planning for UAVs for assembly and structure construction is proposed in [13]. The authors focus on collision-free planning for multiple UAVs performing the construction task. In [14], a group of four UAVs are used to build a tower from foam bricks. The paper describes the indoor application where the positions of bricks for grasping are predefined and UAVs rely solely on a motion capture system. The system is thus very informed about its environment and serves as proof of the concept of building structure from bricks by UAVs.

Research of UAVs for assembly and construction with a main focus being on multi-robot cooperative aspects was part of the ARCAS project [15]. An important capability of the UAVs for direct participation in construction is the aerial manipulation and physical interaction with structures being built. We refer to a thorough survey on the aerial manipulation [16]. In [17], control of aerial robots interacting with other objects is examined for cases such as UAVs equipped with an arm manipulator which could perhaps be used for building more complex structures. In [18], an autonomous aerial helicopter is also equipped with an industrial manipulator. A controller with kinematic coupling is proposed to improve operation with the manipulator onboard the UAV. Fully-actuated UAVs [19] can also be considered for construction tasks due to having higher stability during physical interaction from various tilt angles. Authors of [20] propose a planning approach for structure construction with multiple UAVs equipped with a robotic arm. The approach is addressed by consecutive assembly planning, task allocation planning, and action planning. In contrast with the approach proposed in this paper, none of the state-of-the art publications solve all the sub-problems required for fully-autonomous operation, i.e., visual brick detection and localization, autonomous detection of the pickup and placement locations, mission scheduling for multiple UAVs, control, state estimation, and motion planning.

The herein presented grasping approach uses the visual servoing technique that was previously mentioned for gasping in [21]. The approach in [21] simplified the task to a one dimensional problem with an external motion capture system controlling other dimensions. Such simplification is not possible for a real outdoor experiment. The detection of an object for manipulation with a robotic arm is discussed in [22]. In this work, a stereo camera system is used for object detection in an outdoor environment without a motion capture system. The speed of

[2] `https://github.com/ctu-mrs/mbzirc_2020_wall_building`

object detection is slow, taking up to 1 s and unusable for UAV control and visual servoing. The presented work does not deal with object placing and presents only preliminary results.

In [23], an autonomous aerial pickup and delivery is approached by using a magnetic gripper. The paper focuses on a grasping device employing Electro Permanent Magnet (EPM) and on visual servoing for precise object grasping. The work is motivated by MBZIRC 2017. Similarly, the work [24] focuses on object pickup and delivery. However, both the pickup location and the delivery location are known and marked. Therefore, this task is similar to the gathering of ferrous objects in the MBZIRC 2017. The approach presented in this paper covers full visual servoing in all three dimensions. Furthermore, we do not rely on Global Navigation Satellite System (GNSS) Realtime Kinematics (RTK) thanks to our robust picking mechanism that can compensate for real-world phenomenons.

Related to the previously discussed Challenge 2 of the MBZIRC 2020 is Challenge 3 (Ch3) of the MBZIRC 2017 which featured a treasure hunt scenario where metallic disk-shaped objects were searched for in an arena by three UAVs and collected to a common box. In contrast to the treasure hunt scenario, the wall building task requires additional precise placement on a wall and also features a UGV within the challenge. However, grasping with a magnetic gripper [25] and required cooperation of the UAV team are the same for both challenges. The team lead by Czech Technical University (CTU) won the treasure hunt scenario of the MBZIRC 2017 [26]. The approach [26] also contains initial scanning of the arena. However, the grasping in [26] does not use the herein employed visual servoing and instead uses a more precise RTK GPS. Most importantly, the collection box in the Ch3 of the MBZIRC 2017 was in an a priori known location and of decent size.

The proposed system for the MBZIRC 2017 Ch3 by University of Seville [27] uses a search phase where the arena is divided and cooperatively scanned. A centralized Ground Control Station (GCS) is used for object detection stochastic filtering and further heuristic cooperative planning is used to assign individual UAVs to collect particular detected objects. The GCS also resolves potential conflicts and minimizes probability of collision. The object detection uses color segmentation and clustering, while the grasping employs a visual-based controller to precisely hit the target. The drop is done using the priori known position of the dropping box. The employed UAV platform uses standard GPS and IMU localization while the pickup mechanism uses EPM [28].

Team from ETH Zurich [29] for Ch3 of the MBZIRC 2017 used an approach with repeated switching between exploration and greedy pickup of the closest detected object with consequent delivery. The exploration uses a predefined zig-zag path during the scanning of an assigned arena part and switches to pickup/delivery mode once a valid target is detected. The system is decentralized with minimal data sharing of odometry for collision avoidance and drop box semaphore for dropping synchronization. The object detection is based on color thresholding and a blob detector with consequent classification of blob geometrical shape features for filtering. The detected objects are further tracked and used for pose-based visual servoing. The Nonlinear Model Predictive Controller (NMPC) [30] is used for trajectory control. The localization is based on a combination of RTK GPS and visual-inertial odometry. The grasping employs EPM [28] gripper with Hall effect sensors grasp feedback.

Approach of the University of Bonn [31] for the Ch3 of the MBZIRC 2017 divided the arena into sectors with one for each drone. Each UAV broadcasts its position, navigation target, flight state, and detected objects outside of its own sector. Exploration of each sector is done with one UAV using a spiral pattern with random start. Object detection defines the likelihood of pixels belonging to colored object to be used further in the blob detector. The detected blobs are filtered based on blob shape and color parameters. The approach uses visual detection of the drop box in contrast to other teams. A variant of MPC based on precise trajectory generation [32] is used for controlling the UAVs. The UAV platform uses standard GPS and IMU, and the grasping device uses an electromagnet on a telescopic rod with a ball joint.

The above presented systems addressing the MBZIRC 2017 challenge [29, 31, 27] including the winning solution [26] do not provide sufficient mechanisms for solving the 2020 challenge, despite being state-of-the-art in the field. The aforementioned solutions require delivery of much larger objects that pose more difficult requirements on the precision of grasping and control. The placement of the objects is a key factor, which did not need to be solved in the previous installment of the challenge. The 2020 challenge requires precise placement of the bricks in 3D environment, which is even more challenging due to implied higher risk for the UAV since the UAV is required to fly nearby a complex 3D structure.

Overall, the presented wall building task of MBZIRC 2020 featured a very challenging scenario that required both autonomous outdoor grasping and placing using onboard sensors only. So far, such construction tasks were restricted mostly to controlled lab environments with motion capture systems or were not entirely autonomous. Furthermore, Challenge 2 of the MBZIRC 2020 is more complex than the former Ch3 of the MBZIRC 2017. The wall in MBZIRC 2020 has to be localized automatically due to its arbitrary position in each round and the brick placement has to be very precise for a brick to stay on the wall after placement.

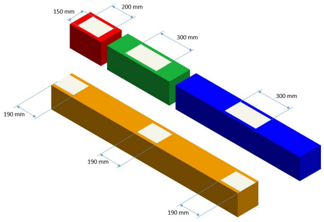
1.2. Problem overview

Challenge 2 of the MBZIRC 2020 featured a wall building task carried out by three UAVs and one UGV. Different wall placement areas and stacks of bricks with which to build were assigned for the UGV and for the UAVs. The layout of the wall building arena, with size of 40 × 50 × 20 m, can be seen in Fig. 7. This paper concerns the UAV part of the Challenge 2, therefore we will further focus on the challenge details concerning this part. A total of four colored brick types — RED, GREEN, BLUE, and ORANGE — were available for possible placement on a wall, each with different weight, shape, and points for placing. Table 2 summarizes length, weight, and scoring of each brick type.

The ORANGE brick could be carried and place by a single UAV or by a group of UAVs. However, the UAV size was

Brick color	Length/m	Weight/kg	Score
RED	0.3	1.0	6
GREEN	0.6	1.0	8
BLUE	1.2	1.5	10
ORANGE	1.8	2.0	20

(a) Brick sizes, weights, and score points.



(b) Illustration of the bricks' visuals.

Figure 2: Parameters of the bricks present in Challenge 2 of the MBZIRC competition [4].

penalized if it exceeded a dimension limit, so a collaborative approach was encouraged. Each brick was also equipped with a ferromagnetic white plate in the middle (and additionally to the sides of the orange brick) to be grasped by a magnetic gripper allowing for multi-robot grasping. Initial layout of the brick stack for the UAVs was in 8 × 4 m area with six rows of bricks where two were reserved for 24 RED bricks, two for 12 GREEN bricks, one for six BLUE bricks, and one for four ORANGE bricks. The UGV had a different brick stack area that was distinguishable from the UAV stack by its properties, as later discussed in Sec. 5.

The wall for UAVs had a shape of the letter 'W' and consisted of four segments. Each segment was 4 m long and placed on a 1.7 m high base. Convex U-shaped channels with transparent sides were attached to the top of the segments to simplify placement and to support the already placed bricks in case of wind. The order in which the bricks were supposed to be placed on the wall was given just before the trials in order to build a wall in a given pattern. The given wall pattern consisted of randomly ordered 4 RED, 2 GREEN and 1 BLUE brick for each layer of the first three segments. The last channel was reserved for ORANGE bricks and can fit two such bricks per layer. Each UAV channel could contain two layers. The final score was based on reward of placed bricks and was further decreased based on number of mistakes in the given wall pattern using a rather complicated formula not relevant to the approach description. Therefore, the goal of Challenge 2 was to build as many bricks as possible according to the wall pattern within 25 minutes of the challenge trial.

2. Hardware platform

This section describes the UAV platform shown in Fig. 3 which was used for all UAVs deployed by the CTU-UPenn-NYU team in Challenge 2 of the MBZIRC 2020.

The utilized UAV quadrotor platform is composed of only commercial off-the-shelf (COTS) parts and rather inexpensive components and sensors. The brick and wall detection relies on one fish-eye color camera and one RGB-D camera. The global localization of the UAVs in the arena is based on a standard GPS receiver accompanied by Light Detection and Ranging (LiDAR) sensor for measuring altitude. The grasping was done using an in-house designed electromagnetic gripper with grasp feedback sensors. Finally, the basic stability of the platform was controlled by COTS flight controller governed by an onboard miniature computer that was used for all computations and autonomy during the wall building task.

The platform is based on the *Tarot 650 Sport* quadrotor frame with four *Tarot 4114 320Kv* motors, each connected to *BLheli32 51A* electronic speed controller and equipped with a 15-inch carbon fiber propeller. The thrust of individual motors, and thus the lowest-level control of the platform, is governed by the *PixHawk 4* flight controller which receives angular rate and total thrust commands from the control pipeline running on the onboard computer. The primary localization system is based on the *ublox Neo-M8N* GPS receiver connected to the flight controller. *Intel NUC Kit NUC8i7BEH* with *Intel i7-8559U* processor and 8 GB of RAM are used for onboard high-level computations including calculation of control commands, high-level planning, brick and wall detection, and others. *Ubuntu 18.04 LTS* operating system is installed along with Robot Operating System (ROS) [33] Melodic flavor which integrates the whole UAV software system.

Apart from the GPS-based localization, the *Garmin LIDAR Lite v3* distance sensor is used to measure the UAV altitude above ground. Brick detection, primarily during grasping, uses the RGB *mvBlueFOX-200w* camera with global shutter, 752 × 480 px resolution, and up to 93 frames per second (fps). The fish-eye camera lens *Sunex DSL215* is used to significantly enlarge the footprint of the camera on the ground. The camera is set to 20 fps, which is a sufficient value for visual servoing during grasping. To avoid obstruction of both the LiDAR and the mvBlueFOX camera by the grasped brick, both sensors are placed on the left side of the platform using a custom holder. Figure 3 shows the placement of the individual sensors on the platform. Down-facing *Intel RealSense D435* RGB-D camera, with depth Field-of-View (FOV) of ≈ $90° \times 58°$ and range of up to 10 m, is primarily used for the wall detection. Depth resolution of the RealSense is up to 1280 × 720 px with a frame rate up to 90 fps. For this challenge, the resolution 848 × 480 px is used with 30 fps. The RealSense camera is mounted under one of the motors and rotated towards the geometric center of the UAV. The mounting points of both RealSense and mvBlueFOX cameras enable navigation close above the walls and bricks.

Grasping of the bricks is done using two *YJ-40/20* electromagnets, each with up to a 25 kg equivalent of holding force. The magnets are connected to a common rod equipped with a

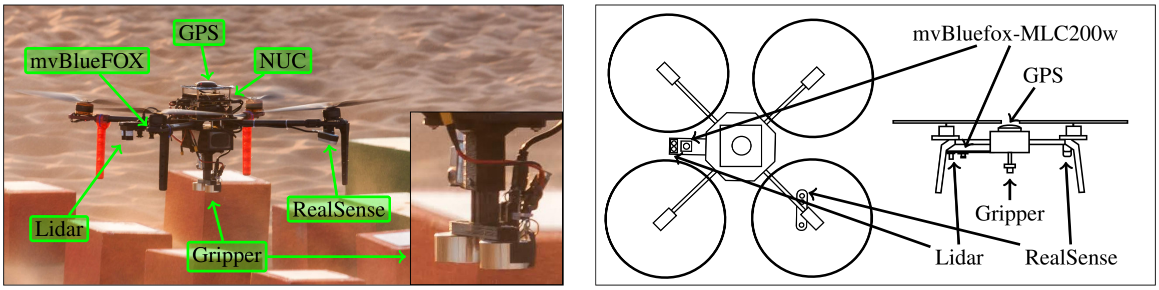


Figure 3: UAV platform for the brick challenge.

spring mechanism along the z-axis for dampening the shocks when landing on a brick gripper-first. Each magnet is equipped with an integrated Hall effect sensor to verify proper attachment of the ferromagnetic part of the brick to the magnet. The electromagnets are rated to operate at 12 V, however, operation at 24 V is selected instead to further increase the grasping force at the cost of higher power consumption and heating.

3. Preliminaries

3.1. UAV control system

Multirotor UAVs are notable for their inherently unstable dynamics. Continual corrections to their flight need to be supplied by a feedback controller at a rate of approximately 100 Hz to maintain stable flight. Moreover, automatic feedback control requires an accurate estimate of the UAV dynamical system states. The tasks of state estimation and feedback control are complemented by several others, such as automatic feedback reference generation, trajectory following, take-off, landing, and more. All these vital subsystems are encapsulated in the *MRS UAV System* [7], an open-source[3] standalone and general control pipeline (see Fig. 4). The *MRS UAV System* was used by the CTU-UPenn-NYU team in all the challenges of the MBZIRC 2020 competition. The provided framework aids deployment of autonomous UAVs, allowing focus mainly on the diverse scenarios of the competition. It relies on the PixHawk embedded flight controller to control the UAV attitude rate $\boldsymbol{\omega}$ and thrust T, while the rest of the pipeline is executed on an onboard high-level computer. The *Mission & navigation* block, which is the core topic of this manuscript, provides the *MRS UAV System* with desired trajectory references to fulfill the objectives of the challenge.

3.2. UAV state estimation

The state estimation part of the *MRS UAV System* fuses data from onboard sensors into multiple independent hypotheses of the UAV state. In context of this particular challenge, the UAV state is estimated using three individual estimators: a GPS-based localization, an optic-flow odometry, and visual servoing relative to an observed brick. These three sources of localization can be used independently depending on the particular situation. Transitions between the stack of bricks and the wall area is made using the GPS-based estimation and the grasping of a brick is achieved via the visual servoing. The optic-flow estimator is used as a backup in case the visual servoing fails. The *MRS UAV System* provides a state estimate consisting of the UAV *body frame* ($\mathcal{B}$) position $\mathbf{r}^{\mathcal{B},\mathcal{W}}$ and orientation $\mathbf{R}^{\mathcal{B},\mathcal{W}}$ within a *world frame* ($\mathcal{W}$). Figure 5 depicts the coordinate frames used within the control pipeline. The absolute position of the *world frame* depends on the actively used state estimator. When the state estimator is changed, the control pipeline synchronizes a virtual jump between the old and new coordinate frame, such that it is not noticeable to an outside observer.

3.3. UAV feedback control and tracking

The *tracking controller*, as depicted in Fig. 4, encapsulates a MPC feed-forward tracking approach [5] for generating a smooth control reference and a *geometric tracking controller on SE(3)* for tracking the control reference [6]. We also utilize an alternative MPC-based feedback controller [7], when the state estimate provided by the onboard estimator might be unreliable (e.g., while grasping a brick). The input to the control pipeline, supplied from the *Mission & navigation* block, can be a 3D position and heading reference $(\mathbf{r}_d, \eta_d)$ or a time-parametrized reference trajectory

$$\{(\mathbf{r}_d, \eta_d)_1, (\mathbf{r}_d, \eta_d)_2, \ldots, (\mathbf{r}_d, \eta_d)_k\}. \tag{1}$$

4. Autonomous multi-UAV wall building

This section describes the proposed high level approach for the wall building task, the state machine of individual UAVs, and the approaches used for the multi-robot coordination of the task. The proposed approach for wall building is designed to distribute the task among the three UAVs as much as possible while mitigating possible mutual collisions. Furthermore, we refer to the individual UAVs as UAV1, UAV2, and UAV3. The task starts with UAV1 scanning the arena in order to find the positions of the wall and the stack of bricks for the UAVs.

[3] http://github.com/ctu-mrs/mrs_uav_system

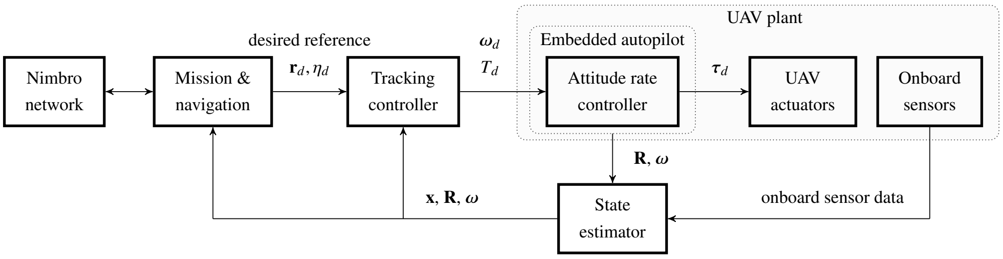


Figure 4: A diagram of the system architecture. The *Mission & navigation* part supplies position and heading reference ($\mathbf{r}_d$, η_d) to a tracking controller. The *Tracking controller* encapsulates feed-forward tracking and feedback control techniques to produce desired thrust and angular velocities (T_d, ω_d) for the Pixhawk embedded flight controller. The *State estimator* fuses data from onboard sensors to create an estimate of the UAV translation and rotation ($\mathbf{x}$, $\mathbf{R}$). The *Nimbro network* manages communication between the UAVs and allows execution of a coordinated multi-robot scenario.

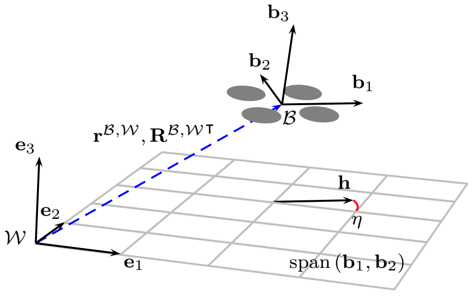


Figure 5: The image depicts the world frame $\mathcal{W} = \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ in which the 3D position and orientation of the UAV body is expressed. The body frame $\mathcal{B} = \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ relates to $\mathcal{W}$ by translation $\mathbf{r}^{\mathcal{B},\mathcal{W}} = [x, y, z]^\intercal$ and rotation $\mathbf{R}^{\mathcal{B},\mathcal{W}\intercal}$, respectively. The UAV heading vector $\mathbf{h}$, which is a projection of $\hat{\mathbf{b}}_1$ to the plane span$(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2)$, forms the heading angle $\eta = \text{atan2}\left(\hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_2, \hat{\mathbf{b}}_1^\intercal \hat{\mathbf{e}}_1\right) = \text{atan2}\left(\mathbf{h}_{(2)}, \mathbf{h}_{(1)}\right)$.

The scanning process is described in detail Sec. 5. After scanning, the positions of mapped wall channels and the individual brick type stacks are shared to the other UAVs (UAV2 and UAV3). Each UAV is then assigned to one of the three first (non-ORANGE) channels to build the bricks sequentially from one side according to the given pattern for individual channels. The brick stack area is also divided along the longer side, such that UAV1 and UAV3 are grasping from the sides and gradually progressing to the middle of the area with each grasped brick, while UAV2 is grasping from the middle part of the area. However, due to the UAV stack area being a size of 8 × 4 m, only UAV1 and UAV3 were flown simultaneously in the competition while UAV2 waited till the others have finished their mission to increase safeness. The task then proceeds with grasping and placing according to the wall pattern assigned to each individual UAVs. The sensor connection and battery state are checked before each grasping begins and the UAV lands after task completion or in the case of battery depletion. The used state machine is further described in the next section while the multi-robot aspects are detailed in Sec. 4.2.

4.1. UAV state machine

The state machine used onboard each UAV to solve the wall building task is depicted in Fig. 6. It is implemented in FlexBE Behavior Engine [34] based on the state machine framework SMACH [35]. The whole system is integrated in the ROS.

The state machine starts by the *Prepare UAV and wait for start* procedure [S1] that initializes all UAV system parts, arms the UAVs, and awaits trigger from remote control to start wall building task. The scanning UAV1 then preforms *Take-off* [S2] immediately after the task starts. Meanwhile, both UAV2 and the UAV3 are in the *Wait for map* state [S3] where they wait for arena map shared from UAV1. UAV1 scans the arena (*Scanning* procedure [S5]) detailed in Sec. 5 then shares the four mapped wall channels and brick stack designed for the UAVs. When the map is received by UAV3, it continues with *Take-off* [S2] as it is used with UAV1 for simultaneous wall building. UAV2 waits for the other two drones to land (*Wait for finish* state [S4]) before continuing with the *Take-off* and further building of the wall. The wall building then continues with *Assign wall and bricks plan* state [S6] that contains the deterministic method that assigns different wall channels, flight altitude, and grasping positions above the brick stack to each UAV. This state creates a plan of individual grasping and placement attempts according to assigned channel pattern. It is discussed more in the following Multi-robot coordination section. The UAVs then proceed with *Assign next brick* state [S7] that selects the next brick according to the plan (assuming there are still bricks in the plan to be placed). If not, the UAV switches to *Land* procedure [S8] and lands at the UAV take off position. Before attempting the actual grasping, the *Check UAV* state [S9] makes sure that the sensors necessary for the grasping and placing tasks are connected. Furthermore, the battery state is checked and if both the sensors and the battery are in ready-to-fly conditions, the UAV proceeds with grasping. Otherwise, the UAV switches to the *Land* procedure. In the *Grasping* procedure [S10], the UAV initially flies to the mapped grasping position for the current brick type in the stack that is assigned to the particular UAV. Afterwards, the procedure continues to the lower-level grasping state machine that includes, e.g., visual brick servoing for precise relative positioning of the UAV above the brick, as is

7

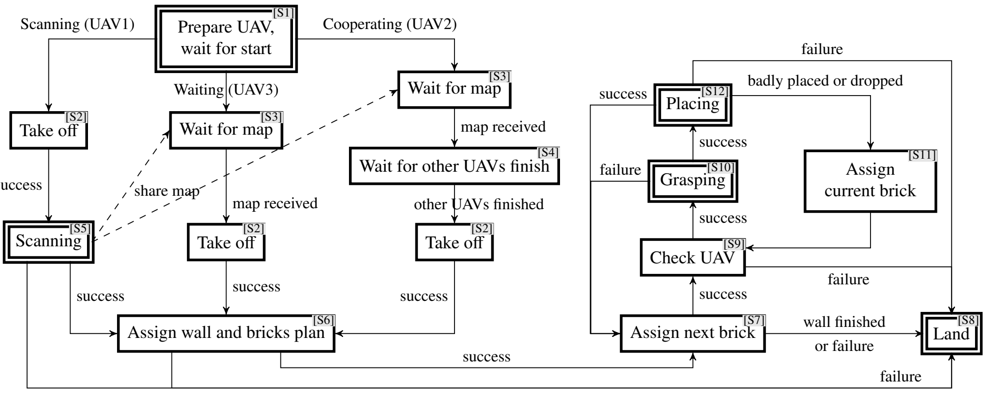


Figure 6: UAV state machine for the wall building task.

described in Sec. 6. In case of grasping failure, the UAV continues with assigning the next brick from the plan in order to try, e.g., a different brick type or brick closer to the middle stack area in case of side UAVs. When the grasping is successful, the *Placing* procedure [S12] starts with flying (with a heading minimizing the brick air resistance) to a designated wait position next to wall using the assigned flight altitude. It then flies to the fixed altitude above the mapped position of the assigned wall channel where a lower-level placing state machine begins. During the placing procedure, the brick is checked for having been dropped or being badly placed. In both cases, the same brick is assigned in *Assign current brick* state [S11] and the UAV proceed with check and grasping. In case of successful placement, the wall building continues to the next brick in the plan [S7]. Notice that all states [S5], [S6], [S7], [S9], and [S12] can result in failure that would switch the UAV to landing at take-off position. However, almost all states contain an additional hard failure mode in which the UAV performs an emergency landing at its current place.

4.2. Multi-robot coordination

The multi-robot coordination with three UAVs for the wall building task was proposed on two levels: the first being based on communication and the second utilizing known arena properties and thus mitigating possible UAV collisions.

The communication between UAVs is based on 5 GHz Wi-Fi network together with the *NimbroNetwork* [36] ROS package that handles sharing certain messages over Wi-Fi. The continuously shared messages are the predicted trajectories and diagnostics of the used onboard MPC [5]. The predicted trajectories are mainly used for collision avoidance purposes while the diagnostics is used as a “heartbeat” of the flying UAVs to be used, e.g., for triggering UAV2 take-off after other UAVs finish or stop responding. Furthermore, the current drone positions and the arena map are shared among the UAV team. The arena map is shared from the scanning UAV1 once the scanning is finished and is used to proceed from the *Wait for map state* [S3]. The map itself contains position and rotation $(x, y, heading)$ of all four wall channels and line segments along the individual brick types.

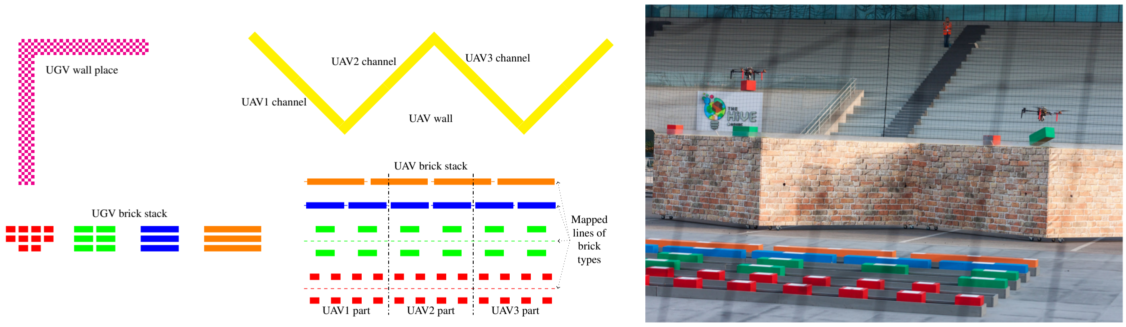
The arena properties that are used to mitigate collisions are the possible partitioning of the UAV wall into four channels and the division of the brick stack area to three parts along the longer side. Only the first three wall channels (non-ORANGE) are handled, each by different UAV and filled sequentially from one side according to the given wall pattern (i.e., from the left in the case of Fig. 7). Each UAV has its own brick stack part where the grasping maneuver of selected brick type starts at the mapped line of that particular brick type (see Fig. 7a). UAV1 and UAV3 begin grasping initially on the outside of their brick stack parts and gradually progress to the middle. UAV2 starts the grasping maneuver in the middle of its part. Furthermore, the stack part of UAV1 and UAV3 are optionally swapped, minimizing the distance between a particular channel and stack part. Finally, each UAV has its own flight altitude ($z \in \{3.0, 4.0, 5.0\}$ m) that is used between the assigned stack part and wall channel. See Fig. 7a for the arena layout and partitioning of the individual wall channels and brick stack among the UAVs.

5. Scanning for bricks and wall placement

The proposed approach begins with scanning the arena as the brick stack and wall channel locations are initially unknown. The scanning task as described throughout this section includes planning of the scanning path, detection of both the bricks and wall channels, filtering and tracking of the detections, and, finally, in creation of the topological map of the brick stack and wall channels.

5.1. Path planning for scanning the arena

The path planning for scanning of the arena is a task of coverage path planning [37] where the entire arena has to be be covered with both the RGB BlueFOX camera and the RGB-D RealSense sensors. The whole task is handled by one UAV as topological map creation from multiple UAVs would require



(a) Illustration of the wall building challenge arena layout

(b) Photo of the UAV arena

Figure 7: Wall building challenge arena layout (a) and photo from the competition (b).

a high-bandwidth network and synchronization of all detections. The path is composed of classical zig-zag primitives with smooth curvature constrained by turning radius ρ computed as $\rho = v_t^2/a_{max}$. The used turning velocity v_t and maximal acceleration a_{max} are $1.5\,\mathrm{m\,s^{-1}}$ and $2\,\mathrm{m\,s^{-2}}$, respectively. Scanning speed is however $3\,\mathrm{m\,s^{-1}}$, meaning that the scanning UAV is accelerating and decelerating after and before turns. The final zig-zag path, as shown in Fig. 21 in the Sec. 8, is then calculated with respect to the ≈90° RealSense horizontal FOV (which is smaller than the BlueFOX camera lens), and also with respect to the set scanning altitude of 4.5 m (due to limitations of the RealSense distance measurements quality).

5.2. *Wall detection*

The wall detection method uses the onboard RealSense D435 sensor. As mentioned in [38], the accuracy of this sensor depends on the selected resolution and parameters of the sensor. The RealSense was dedicated for the improvement of brick detection primarily from short distances. Therefore, the image resolution was set to 848×480 px with a minimal detection distance of 0.175 m. The selected resolution has a further influence on depth accuracy. Therefore, while scanning from altitude 4.5 m, the root mean square (RMS) error of the distance measurements is ≈ 0.6 m. Such measurement error is for the worst case scenario in an outdoor environment where it also depends highly on the target's texture.

The first step of the wall detection method is to find the ground plane. As the UAV orientation is known, it only seeks to find the UAV height above the ground. The arena for Challenge 2 had an almost flat surface with the brick height at 0.2 m, the pillar of bricks for UGV with 0.6–0.8 m height, and a wall height of 1.7 m. The UAV is equipped with down-pointing Garmin LiDAR lite v3 sensor to measure UAV height, but the sensor can also point to an obstacle. The LiDAR height measurement was not used during the grasping or placing of a brick. Instead, we used a ground plane distance estimated from the RealSense stereo camera. The orientation of the UAV is estimated using the onboard IMU (tilt) and magnetometer (heading).

To speed up the detection, the input depth data with resolution 848×480 px are reduced to resolution 106×60 px by selecting minimal valid value (greater than zero) from each 8×8 px sub-image. The minimal value filter is used to reduce the size of the input data and to simultaneously remove outliers with invalid data of zero or measurements that are higher than actually possible.

The measured sensor data are then rotated to the world coordinate system. The depth measurement $d = d(x, y)$ represents a 3D point

$$\mathbf{p} = \left[d \cdot \frac{x - c_x}{f_x}, d \cdot \frac{y - c_y}{f_y}, d\right]^\intercal, \tag{2}$$

where c_x, c_y, f_x, f_y are parameters of the RealSense camera received from factory calibration. Firstly, we transform the measurements $\mathbf{p}^{\mathcal{S}}$ from the sensor frame $\mathcal{S}$ to the UAV body frame $\mathcal{B}$ as

$$\mathbf{p}^{\mathcal{B}} = \mathbf{R}^{\mathcal{S},\mathcal{B}}\mathbf{p}^{\mathcal{S}} + \mathbf{r}^{\mathcal{S},\mathcal{B}}, \tag{3}$$

where $\mathbf{R}^{\mathcal{S},\mathcal{B}}$ is the rotation from $\mathcal{S}$ to $\mathcal{B}$ and $\mathbf{r}^{\mathcal{S},\mathcal{B}}$ is the translation from $\mathcal{S}$ to $\mathcal{B}$. We then similarly transform the measurement from the body frame $\mathcal{B}$ to the world frame $\mathcal{W}$:

$$\mathbf{p}^{\mathcal{W}} = \mathbf{R}^{\mathcal{B},\mathcal{W}}\mathbf{p}^{\mathcal{B}} + \mathbf{r}^{\mathcal{B},\mathcal{W}}. \tag{4}$$

Therefore, the final transformation is written as

$$\begin{aligned}\mathbf{p}^{\mathcal{W}} &= \mathbf{R}^{\mathcal{B},\mathcal{W}}\left(\mathbf{R}^{\mathcal{S},\mathcal{B}}\mathbf{p}^{\mathcal{S}} + \mathbf{r}^{\mathcal{S},\mathcal{B}}\right) + \mathbf{r}^{\mathcal{B},\mathcal{W}}\\ &= \mathbf{R}^{\mathcal{B},\mathcal{W}}\mathbf{R}^{\mathcal{S},\mathcal{B}}\mathbf{p}^{\mathcal{S}} + \mathbf{R}^{\mathcal{B},\mathcal{W}}\mathbf{r}^{\mathcal{S},\mathcal{B}} + \mathbf{r}^{\mathcal{B},\mathcal{W}},\end{aligned}$$

to obtain $\mathbf{p}^{\mathcal{W}} = \left[p_x^{\mathcal{W}}, p_y^{\mathcal{W}}, p_z^{\mathcal{W}}\right]^\intercal$. However, for object detection only the z component $p_z^{\mathcal{W}}$ is important and $\mathbf{R}^{\mathcal{B},\mathcal{W}}\mathbf{r}^{\mathcal{S},\mathcal{B}} + \mathbf{r}^{\mathcal{B},\mathcal{W}}$ is constant for one measurement. The altitude of the point representing depth measurement $d = d(x, y)$ at pixel coordinates $[x, y]$ is defined as

$$p_z^{\mathcal{W}} = d \cdot \mathbf{R}_z \left[\frac{x - c_x}{f_x}, \frac{y - c_y}{f_y}, 1\right]^\intercal + \mathbf{r}, \tag{5}$$

where $\mathbf{R}_z \in \mathbb{R}^3$ is the last row of the rotation matrix $\mathbf{R} = \mathbf{R}^{\mathcal{B},\mathcal{W}}\mathbf{R}^{\mathcal{S},\mathcal{B}}$ and $\mathbf{r}$ is a constant value for one measurement. This simplification speeds up computation by 3 times.

The altitude of the UAV is then computed from a histogram of altitudes of all points in the reduced 106×60 px image. Based on the experimental evaluations in desert environment prior to the competition, altitude of the UAV is estimated as a value for which more than 1000 px have a larger or equal measurement. If the detected distance is bigger than 3.5 m, the accuracy of the RealSense sensor decreases and the UAV altitude combines the Garmin LiDAR measurement and RealSense measurement.

The next step is to create a binary image I_{thr} by thresholding all pixels with an altitude higher than 1 m above the ground (the height of the wall is 1.7 m and the accuracy of the RealSense sensor from an altitude of 5 m is 0.6 m). The following steps in the Alg. 1 use the OpenCV functions [39].

Algorithm 1: Wall detection

```
Input: $I_{thr}$ – $thresholdedimage$
Output: $(x, y, \alpha)$ — center of the wall segment with $\alpha$ wall
        rotation
$I_{clo} = morphology\_close(I_{thr})$        // OpenCV functions
  erode, dilate
$Contours = findContours(I_{clo})$        // OpenCV functions
  findContours
$Contours_{transform} = \emptyset$
for $p \in Contours$ do
    if $p$ is not at border then
        Add $R \cdot p$ into list $Contours_{transform}$
$Lines_{approx} = approxPolyDP(Contours_{transform})$ // OpenCV
  functions approxPolyDP
for $l_1, l_2 \in Lines_{approx}$ do
    if $|l_1 \times l_2| < thr_1$ then        // $l_1$ is parralel to $l_2$
        if $|l_1 - l_2| \approx wall\_width$ then   // distance $l_1$ to
          $l_2$ is correct
            output $(x, y, \alpha)$ — $(x, y)$ is center between $l_1$ and
              $l_2$, $\alpha$ — is $l_1$, $l_2$ orientation
```

The functions $morphology\_close, findContours$, and $approxPolyDP$ are from OpenCV library. The last step of the Alg. 1 is testing for whether the distance between two lines is correct. This test compares the distance of endpoints of one line from the second line and vice versa, but this test does not recognize whether the lines are parallel and opposite to each other. To test this feature, the minimal rectangle that contains all endpoints of both lines is created. The length of this rectangle has to be less than a 0.8 sum of both lines and the width of this rectangle should be approximately equal to wall width, 0.25–0.40 m. Positions of the lines with respect to minimal rectangle is shown in Fig. 8.

The results and various stages of the wall detection method are depicted in Fig. 9.

5.3. Brick detection

Detection of the bricks using the mvBlueFOX color camera is based on the white plate detection in the center of each brick.

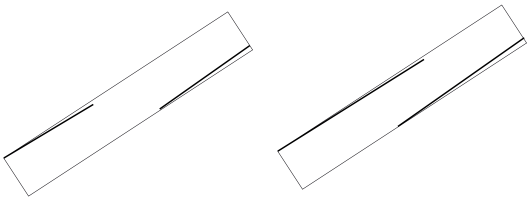


Figure 8: Positions of the wall border lines in a minimal containing rectangle. Wrong position of the parallel lines is show on the left and the correct position on the right.

Table 1: HSV ranges for color segmentation.

	Hue		Saturation		Value	
	min	max	min	max	min	max
White	0	180	0	60	180	255
Red1	0	8	70	255	80	255
Red2	160	180	70	255	80	255
Green	44	80	60	255	60	255
Blue	80	130	60	255	60	255

The position of a brick in global world is based on known altitude and orientation of UAV and the known brick height. The white detection is based on simple color segmentation using OpenCV function *inRange()* applied to a HSV image. The HSV image is created by function *cvtColor* (for color space conversion) from the original color camera data. The HSV image is further used for red, green, and blue detection. The parameters for white, red, green, and blue segmentation are listed in Table 1, where hue is from interval $\langle 0, 180\rangle$, with saturation and value from interval $\langle 0, 255\rangle$.

The method that finds a white plate in the segmented image is described in Alg. 2. After segmentation, the image is processed by morphological *closing* operation and the contours are computed by OpenCV *findContours()* function with chain simple approximation. All points on contours are then undistorted and transformed into a plane parallel to the ground plane with height equal to the brick height of 0.2 m. The lenses used for the color camera are very wide with a horizontal FOV of 185°; therefore we use the *Ocam toolbox* [40] for omnidirectional cameras. The undistortion operation is done simultaneously with the rotational transformation to the global coordinate system in order to speed up the computation. Finally, a convex hull of points in the global coordinate system is found and used for brick classification.

If the entire white segment is inside the camera image, then the correctness of detection depends only on the size of the minimal rectangle area that contains the border of white segment transformed into the world coordinate frame. If the UAV is close to the brick, the white segment can cross the border of the image, so that the entire white plate is not in the camera image. If the white segment forms a U shape (i.e. shape from two parallel lines and one perpendicular line) then the center of the brick can be calculated not as the center of the transformed

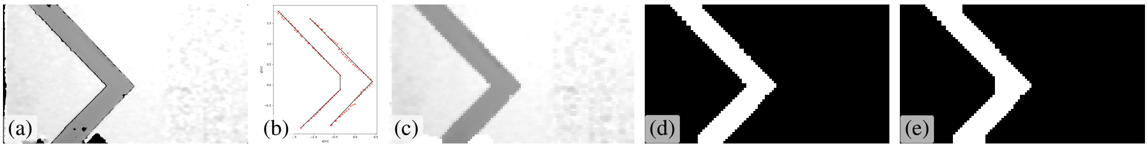


Figure 9: Original depth data (a) used for computing the result of the wall detection (b). The depth data are first filtered to lower resolution (c), then thresholded using wall height (d), and applied with the morphological *closing* operation (e).

area, but as a point with correct distance from the transformed border.

The results of the brick detection algorithm from the scanning altitude 4.5 m are depicted in Fig. 10. Details of transformed points to a global coordinate system with illustrated brick detection is shown in Fig. 11. The better and more accurate detection from a lower altitude is depicted in Fig. 12. In results from brick grasping (see Fig. 13), it can be noticed that only part of the white plate is visible and the correct brick position has to be calculated from the border shape.

5.4. *Brick and wall filtering*

Each detected brick and wall segment is filtered and placed in a map. The map consists of a bank of Linear Kalman Filters (LKFs) that maintains a smooth hypotheses of each object and provides stable references during grasping and placing attempts. Upon detection, each object is first checked for a series of preconditions to be later fused in the map:

- the object is excluded, if its coordinates are within 5 m from other UAV target,
- objects situated outside of the designated challenge area are excluded,
- bricks whose attempt to grasp was previously unsuccessful is excluded,
- wall segments outside the expected height range $[1.0, 2.3]$ m are excluded.

The objects which pass the preconditions are matched with their nearest neighbour in the map. In the case of a brick, a standard correction to the LKF is formed, containing the brick's x, y, z world coordinates and heading. The wall segments also contain their length, which is an important factor for matching the measurements to the map. The wall segment detections are projected orthogonally to a candidate hypothesis to obtain a measure for evaluating the similarity of the segments. When no match is found for the detected object, a new instance of LKFs is created and placed in the map. Each hypothesis in the map maintains a counter for the number of corrections that were applied to the instance of the LKF.

Post-processing of the map is applied periodically during flight to merge nearby hypotheses by combining their states in the ratio of the number of corrections in each hypothesis. This is required due to the drift of the GPS localization system which causes the objects to drift even in the time span of a single flight. The post-processed detection map is later used to obtain a topological estimate of important sectors in the map (e.g. UAV wall area, UGV wall area, UAV brick area, and UGV brick area).

5.5. *Topological map creation*

The wall and brick detections are saved and filtered in the detection map during the entire scanning flight in order to create the topological map of the arena. It is essential to precisely map positions of the wall channels and the individual brick type stacks (as shown in Fig. 7a), to determine wall building plans for individual UAVs. Map creation had to manage possible wrong detections, filter out the UGV bricks present in the arena, and correctly decide the order of wall channels to follow for the prescribed wall pattern. Figure 22 in the results section shows an example of the mapped wall channels and bricks.

The brick and wall detections received from the detection map are handled separately as they were placed independently in the arena. Initially, all bricks with a low number of corrections (empirically set to 6 corrections) are filtered out and considered detection noise. Next, a Gaussian Mixture Model (GMM) [41] for two clusters is estimated using the detections' (x, y) position in order to separate UAV and UGV brick stacks. Only the detections which are close (within 6 m) to one of the two cluster means are kept and a Principal Component Analysis (PCA) [42] with two components (due to the data being two-dimensional in (x, y)) is applied to both clusters. The PCA returns two variances for each cluster effectively proportional to width and height of the arbitrarily rotated UAV and UGV brick stacks. The UAV stack is then selected as the one with the larger width. A median filter together with outlier removal (bricks farther than 8 m from the median) is then iteratively used for the UAV stack until the median converges, or for a limited number of iterations. The line segments along the brick types (see Fig. 7a and Fig. 22b) are formed from the brick (x, y) mean position of individual brick types and from the median heading of all remaining bricks.

The walls from the detection map are also first filtered out in case of no corrections of the particular wall. The iterative median filter with outlier removal (of wall centers farther than 10 m from the median), similar to the one for the brick detections, is then used to find the most perspective location of the 'W' letter-shaped wall segments (see Fig. 7a). After the median filtering, the remaining wall segments are clustered together such that each detection is assigned to a cluster with horizontal distance within 3 m and heading distance within 0.5 rad. The average position and rotation of the clusters are then assumed to

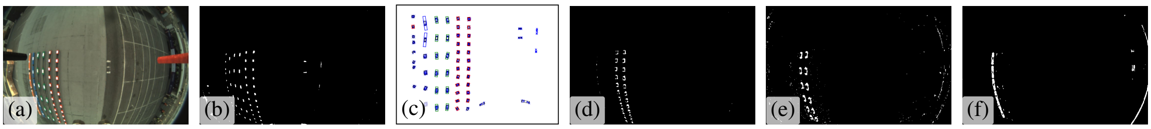


Figure 10: Brick detection based on an original color camera image (a) with consequent white color segmentation (b) and contour detection (c). The segmentation results are shown for red (d), green (e), and blue (f) colors.

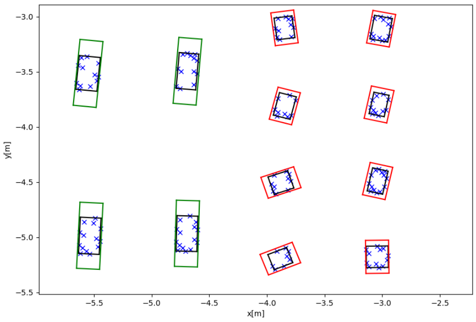


Figure 11: Detail of contours and brick detection results from Fig. 10 including the incorrect orientation of several bricks from scanning altitude.

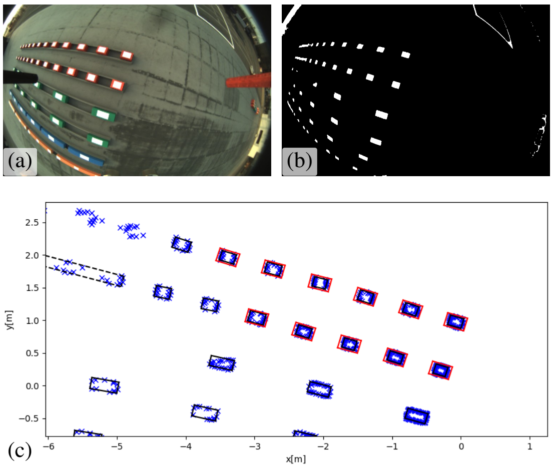


Figure 12: An original image from the color camera, white color segmentation result, and the contours and brick detection results.

be the individual wall channels. Line intersections of such wall channels are used to decide the order of wall channels in the 'W' letter shape while only intersections within short distance (≤ 6 m) from channel centers and with rather perpendicular mutual heading angle (≥ 0.7 rad) are further used. A channel with only one intersection is then used as the first in the string of the 'W' shape. Alternatively, an intersection with the highest sum of distances from wall centers is iteratively removed until a wall with one intersection exists. Afterwards, the other wall channels are added to the selected first channel according to the remaining intersections, and if more options exist, by selecting the intersection with shortest distances from wall centers. Finally, the channel centers, lengths, and headings are determined (see Fig. 22b) based on the intersections of the formed 'W' shape wall structure.

6. Brick grasping

Brick grasping is the second primary capability for the wall building task. The proposed approach for grasping uses a fusion of color and depth camera sensors for brick detection and visual servoing for precise control during grasping attempts. The grasping state machine is used to govern various stages of grasping with UAV mass and attitude being checked during grasping to abort in case of, e.g., a grasp far from the brick center of mass.

6.1. Brick detection and localization

Brick grasping is based on fast and robust brick detection. A fusion of detections from the color camera and from the depth RealSense sensor is used to improve the robustness. The method of brick detection from the color camera is the same as is used during the scanning (described in Sec. 5.3). Brick detection from the RealSense sensor is similar to the wall detection from the RealSense as described in Sec. 5.2. The altitude of measured points is computed for each pixel of the depth reduced image. Similarly to the wall detection, the brick detection uses altitude thresholding with the threshold value of 0.15 m. Figure 14 shows the result of the thresholding (14c), boundary pixels transformed to UAV coordinate system, and their line approximations (14d - points and dashed lines). The final brick detection is the same as for the color brick detection from Sec. 5.3.

The final data fusion uses a weighted average of the color and depth detections where the weights depend on the quality of the detections. The best quality detection weight 1 is in case of the whole brick contour being visible by the sensor. Supposing

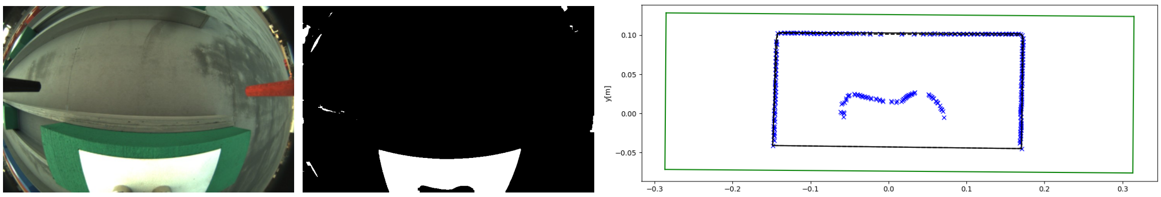


Figure 13: An original image from the color camera, white color segmentation result, and the contours and brick detection results.

three perpendiculars lines represent the contour then the weight 0.5 is used. In case of only two perpendicular lines, the weight 0.25 is used. Example of data fusion is depicted in Fig. 14.

6.2. UAV visual servoing

For precise grasping, we employ a visual servoing technique where the position of the UAV is computed in the coordinate system of the brick that is being grasped. The main challenge of such visual servoing is the ambiguity of the brick's coordinate system (see Fig. 15 with two possible axis placements). The z axis of the brick frame is parallel to the world frame z axis, and therefore the z coordinate is only a shift of the z coordinate in world frame by the brick's height. The brick coordinate system is defined by brick position in world frame $\mathbf{b}^{\mathcal{W}}$ and brick orientation η_b in x, y the axis plane of world frame as seen in Fig. 15. The computation of UAV position within the brick coordinate system must remember the last brick orientation, and thus the choice of initial axis placement. The ambiguity of the brick coordinate system is caused by inaccuracy of brick orientation that can change as the UAV moves closer to the brick.

The detection algorithm finds bricks from actual data in a temporarily created map for data fusion. The last brick orientation is used to select a new brick orientation. The selection is based on the angle difference between orientation of the last brick η_b^{map} and the newly detected brick η_b^{new}. The selection can be expressed as

$$\eta_b = \begin{cases} \eta_b^{new}, & \text{if } \langle \eta_b^{map} - \eta_b^{new} \rangle < \langle \eta_b^{map} - \eta_b^{new} - \Pi \rangle, \\ \eta_b^{new} + \Pi, & \text{otherwise.} \end{cases} \tag{6}$$

The equation uses angle difference $\langle a - b \rangle$ which is the absolute value difference between angle a and angle b, with result in interval $< 0, \Pi >$.

The position of the UAV in world frame system is denoted $\mathbf{r}^{\mathcal{W}}$. The UAV position $\mathbf{r}^{\mathcal{O}}$ within the coordinate frame $\mathcal{O}$ of the brick is expressed as

$$\mathbf{r}^{\mathcal{O}} = \begin{bmatrix} \cos \eta_b & -\sin \eta_b & 0 \\ \sin \eta_b & \cos \eta_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{r}_x^{\mathcal{W}} - \mathbf{b}_x^{\mathcal{W}} \\ \mathbf{r}_y^{\mathcal{W}} - \mathbf{b}_y^{\mathcal{W}} \\ \mathbf{r}_z^{\mathcal{W}} - 0.2 \end{bmatrix}, \tag{7}$$

where 0.2 represents height of the brick that is used as a shift in the z axis.

6.3. UAV-brick interaction and control

Interaction of a multirotor UAV with the environment is a complex challenge. Small objects, such as the ones being collected during the MBZIRC 2017 challenge [26], posed little to no challenge for common UAVs to carry. However, the much larger and heavier bricks impose torque on the UAV if not grasped in line with the center of mass of the object. Moreover, the grasping event poses a threat to the UAV by possibly limiting the controllable degrees of freedom (DOFs) of the UAV due to mechanical contact.

The first challenge of carrying a sizeable elongated object was solved by designing the underlying UAV control architecture. The control pipeline executes a real-time weight estimator that allows the UAV to not only detect an increase of its weight if an object was grasped, but also to detect a decrease when the UAV rests upon the brick during the grasping maneuver. The estimated mass is used throughout the control pipeline to provide adequate feed-forward control terms and scale the control gains of the employed *SE(3) geometric feedback controller* [6]. With such measures, our UAVs were able to repeatedly carry all the brick types while performing moderately aggressive maneuvers. It is worth noting our team's UAVs were the smallest vehicles of all teams, which conducted the task autonomously with an approximate 3:1 ratio of UAV mass to brick mass.

The task of grasping a brick requires automatic safety measures to abort the action when the UAV becomes uncontrollable. Such a situation often occurs if the UAV transfers its weight unevenly through its landing gear to the ground during the last moments of the grasping maneuver. This state needs to be detected automatically by measuring the attitude control error and applying acceleration upwards to mitigate the effect quickly. On the other hand, a false positive grasping event can occur when the magnetic gripper fails to attach. This situation is detected as a significant decrease in the estimated mass due to the transfer of the UAV weight thought the gripper to the brick. In both cases, the maneuver is aborted and repeated before a collision can occur.

6.4. Brick grasping state machine

The action of grasping a brick by the UAV was governed by a state machine closely resembling the prize-winning variant from our last success during the MBZIRC 2017 challenge [26]. Figure 16 depicts the states of the grasping state machine. The UAV is expected to be located at the vicinity of the desired brick (such that the brick is visible in the camera) when

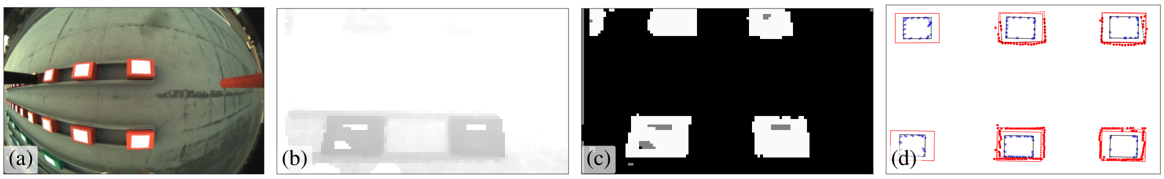


Figure 14: Fusion of data for brick detection using color-based (a) and depth-based (b) detection. Thresholded depth data are shown in (c), while the fused data are in (d) showing detections from the color camera as solid line, and from depth camera as dashed line.

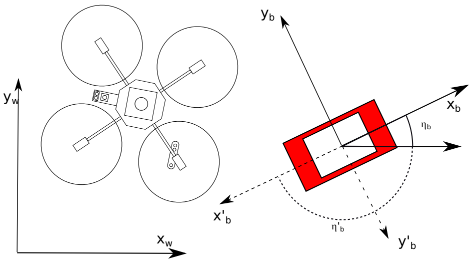


Figure 15: Brick coordinate with two possible axis placements.

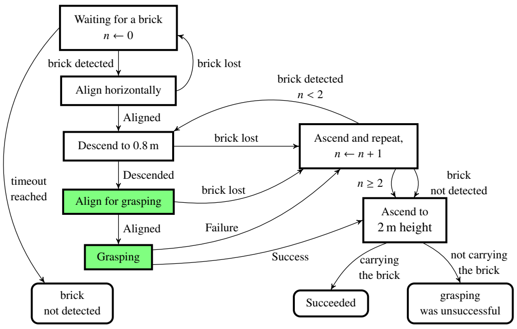


Figure 16: UAV state machine for grasping a particular brick. This whole state machine corresponds to state [S10] within Fig. 6. The white state represents situations when the UAV was localized by the GPS, while in the green state, the states of the UAV were estimated using only the visual detection of the brick for increased precision.

the first state is activated. When a brick is detected, the UAV first aligns itself horizontally with the closest brick of the desired color. After the horizontal alignment distance is lower than 0.2 m, the UAV slowly descends to the height of 0.8 m while actively maintaining the alignment. If the alignment is broken, the UAV ascends and attempts to realign with the brick to repeat the process. The process is repeated a maximum of only twice after which the brick is abandoned and its location is temporarily banned to prevent deadlocks. Conversely, when the UAV successfully descends to the height of 0.8 m, it switches its localization system to the direct brick visual servoing (the green states within Fig. 16). The UAV then realigns itself again using only the detected brick as a source of the state estimate. This second alignment starts with 3 cm alignment criterion and relaxes the distance with time. This ensures that the UAV eventually attempts to grasp the brick even if the control accuracy is low. The final grasping maneuver is also performed using the visual localization of the UAV relative to the brick. The process of adaptive and repeated switching from GNSS-based control into visual servoing and back is the main contribution of this part for general object manipulation in demanding outdoor conditions.

7. Brick placing

7.1. Placing location detection

Place detection uses a similar approach as wall detection (see Sec. 5.2). However, there are three main challenges for detection of locations for where to place the next brick. The top surface of the wall is tiled with a repetitive pattern (see Fig. 17) that makes the stereo camera detection difficult. This problem is solved mainly by filtering minimal distances by reduction of the depth image size. Additionally, the second solution uses a morphological *closing* operation after thresholding. The second challenging aspect is that the view of onboard camera to the wall is partially blocked by the brick attached to the gripper. This problem is solved by applying an automatically created mask after successfully grasping the brick. An example of such a mask is depicted in Fig. 17.

The last challenge is the transparent channel wings that were used to facilitate placement of bricks. These transparent acrylic borders are rather randomly visible on depth measurements and may occasionally appear as a place free for placing a brick. The proposed solution is to detect the free end of the

```
Algorithm 2: White_plate_detection
Input: $I_{thr}$ – $thresholdedimage$
Output: $(x, y, \alpha)$ — center of the white plate with $\alpha$ plate
    rotation
$I_{clo} = morphology\_close(I_{thr})$    // OpenCV functions
 erode, dilate
$Contours = findContours(I_{clo})$    // OpenCV functions
 findContours
for $Contour \in Contours$ do
    $Points = \emptyset$
    for $p \in Contour$ do
        $v = undistort(p)$  // $v$ is vector pointing in
         undistorted direction
        $v' = R \cdot v$  // $R$ is the rotation matrix from
         sensor to world frame
        $koef = UAV_{pos}(2) - brick\_height$    // $UAV_{pos}(2)$
         is altitude of the UAV
        $p' = UAV_{pos} + v' \cdot koef$
        Add $p'$ into list $Points$
    $Convex = convexHull(Points)$  // OpenCV functions
     convexHull
    if $Contour$ is not at border then
        $box = minAreaRect(Convex)$          // OpenCV
         functions minAreaRect
        if box has correct size then
            output $(x, y, \alpha)$ – $(x, y)$ is box center, $\alpha$ — is box
             orientation
    else
        $Lines = approxPolyDP(Convex)$    // only for
         detection from discance less than 1.5 m
        if Lines forms U shape and size of U shape is
         correct then
            find middle parallel line inside U shape and
             found expected brick center $(x, y)$
            output $(x, y, \alpha)$ — $\alpha$ — is middle line
             orientation
```

wall by detecting the leftmost border point of the thresholded image (see Fig. 18).

The method for detecting the placing spot on the wall assumes the wall segment is already aligned with the wider axis of the camera image. The alignment is initially governed by the global planner which operates with necessary information obtained during the initial sweep. The size of the brick being placed is known since the grasping procedure, and thus the method detects a place on the free wall at a correct distance from the leftmost border of the detected wall segment within the image. As two layers of bricks can be built on the wall, the free area on the wall depends on the currently active layer. In many cases, such free space contains a transparent acrylic border of the wall. This border is removed from the detected wall by morphological *erosion*.

The leftmost place on the wall is selected and if the left border of the wall is not visible, the UAV moves to the leftmost part of a visible wall to find the correct edge. The results of this algorithm are depicted in Fig. 18.

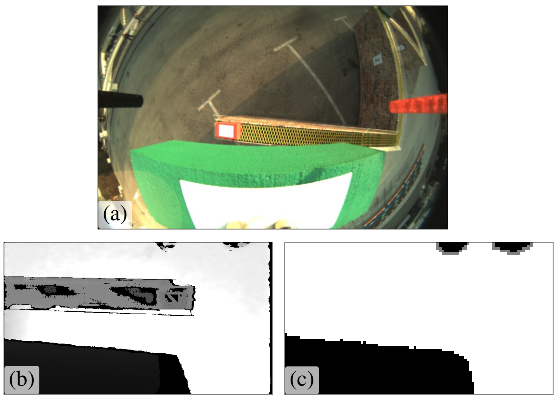


Figure 17: Original data from the color camera (a) and original depth data (b). The mask (c) for the depth data is used to remove the carried brick and the UAV leg in the top part of the mask.

7.2. Placing state machine

The action of placing a brick on the wall is governed by the state machine depicted in Fig. 19. This lower-level state machine is responsible for guiding the UAV above the spot designated for placement and controlling the descent to a desired height above the wall. Since the outcome of placing a brick can rarely be influenced after releasing the brick from the magnetic gripper, we do not consider any actions in case of failure. Moreover, numerous bricks are available in the grasping area, so grasping a misplaced brick or even repairing the wall is forfeit over continuing for a fresh brick. Therefore, the placing state machine sequentially follows the actions of aligning horizontally with the wall and descending while aiming for the designated spot. If anything fails, the held brick is dropped and the UAV continues above the brick area to obtain a new brick. This was chosen so as to not counteract any potential failure states, such as sudden misalignment (e.g., caused by localization drift). A simpler yet capable approach was chosen due to the added complexity and less-deterministic execution of a more failure-proof solution.

8. Experimental results

This section describes the results achieved during the MBZIRC Challenge 2 competition trials using the proposed system. The herein system for wall building by UAVs, developed as detailed above by the CTU-UPenn-NYU team, was able to win Challenge 2 by placing the far most number of bricks, mostly with the UAVs. However, the UGV deployed in the challenge also contributed by autonomously placing a brick during the second competition trial as described in [43]. The CTU-UPenn-NYU team won by scoring 8.24 points, while the second Nimbro Team (University of Bonn) scored 1.33 points, and the third (Technical University of Denmark) 0.89 points. Figure 26 depicts the team at the winner stand.

Prior to the competition, the team dedicated over a month for preparation and experimental evaluation in a desert near

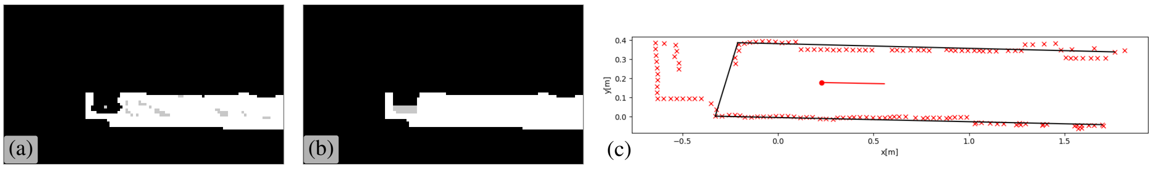


Figure 18: Thresholded depth data (a) and thresholded data after applying the morphological *closing* operation (b). The result of place detection (c) where the red line demonstrates shift from the center of the wall to the leftmost position of the brick.

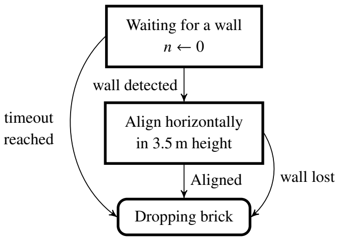


Figure 19: UAV state machine for placing a brick on the wall. This state machine corresponds to state [S12] within Fig. 6.

Table 2: Grasped bricks during the two competition trials of Challenge 2.

Trial	Restart	UAV1	UAV3
one	1	R^b, R, G, G^b	G
	2	R, R, G^b	R, G^b
	3	R^b, R^b	R, R
	4	G	R^s, R^b
two	1	G, G, R^b, R^b, R	–

Abu Dhabi. Real world experiments could not be conducted in the Czech Republic in the final months of preparations due to the winter weather conditions. Therefore, the team decided to conduct final preparation near the competition venue, despite the unforgiving high temperatures, sand, and wind conditions of the coastal United Arab Emirates. This preparation phase proved to be crucial in securing first place in the competition just as it was in 2017. Figure 20 depicts photos from the desert experiments. Videos from the experiments are available at `http://mrs.felk.cvut.cz/mbzirc-2020-uav-wall`.

Table 2 shows the overall performance of the system during the two competition trials, each lasting 25 minutes. It shows the types of bricks, denoted as 'R' for RED and 'G' for GREEN, in the order they were grasped during the individual trials. The grasping of BLUE bricks was not attempted during the competition trials, although the UAVs were capable of carrying them, mainly due a significantly higher detachment probability of a grasped brick caused by the pendulum effect of the longer and heavier BLUE brick. The grasping of ORANGE bricks was also not attempted for similar reasons and difficulty of cooperative carrying. The competition did not require delivering any ORANGE bricks to qualify for obtaining points. The grasped bricks are further displayed per individual restarts within the trials, where the restart had the possibility of keeping the already placed bricks and running the system again with all robots in their initial positions.

Table 2 shows all grasped bricks, although not all were successfully placed on a wall. The bricks denoted with 'b' index did not have successful placement, which in most cases was due to the brick bouncing off the wall after release. The single case of a placed RED brick during the fourth restart of the first trial, denoted with '^{s}' index, is a placement into the second layer of a particular channel which was also not achieved by any other team.

Notice that only UAV1 and UAV3 were used for wall building during the competition trials. The strategy of having UAV2 wait for a collision or battery depletion of another UAV was not required during the competition. However, this reliable multi-UAV strategy was successfully tested during the pre-competition trials.

Ten bricks were successfully placed on the wall during the first trial consisting of seven RED and three GREEN bricks. Seven bricks bounced off the wall during the same trial. The main focus of the CTU-UPenn-NYU team during the second trial was to autonomously place at least one brick using the UGV as was required for winning the challenge. Therefore, many restarts were done to ensure this goal and only before the first restart were the UAVs used to grasp and place bricks as shown in Table 2. In the rest of this section, we focus on the individual (i.e. scanning, grasping, placing) wall building subtasks and the overall performance achieved during the first trial, in which UAVs were used throughout the entire trial.

8.1. Scanning for bricks and wall placement

Scanning of the arena was the first subtask of the proposed UAV wall building approach performed in order to find the location of the brick stack and wall position designed for the UAVs. The arena scanning was planned using a zig-zag path within the predefined arena space (defined by arena corners and safety area) as described in Sec. 5.1.

Figure 21 shows all brick detections (RED, GREEN and BLUE- ORANGE bricks were not considered) as well as the detections of the wall channels. The zig-zag scanning path is shown within the arena boundaries as recorded by the onboard GPS. The employed brick detection using the onboard RGB camera was able to detect bricks in the range of ≈ 10 m × 5 m

Figure 20: Photos from preparation in the Abu Dhabi desert. Videos are available at `http://mrs.felk.cvut.cz/mbzirc-2020-uav-wall`.

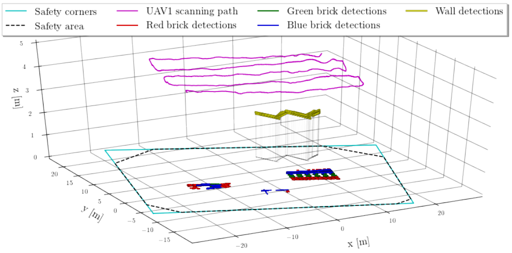


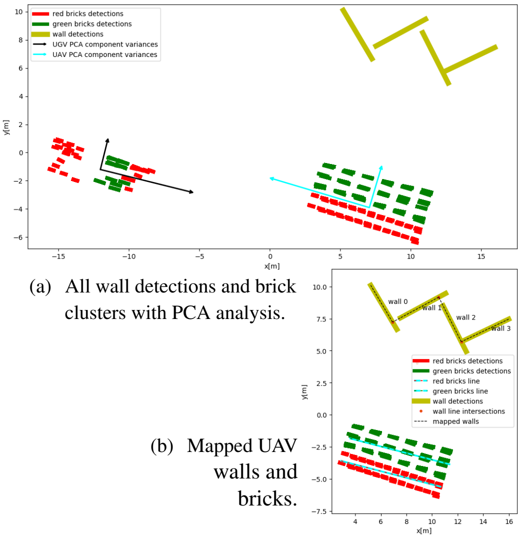
Figure 21: Path of UAV1 while scanning the arena and detections of the bricks and UAV wall channels.

in x × y coordinates of the camera when flying at the scanning 4.5 m height. However, robust detections of bricks were obtained in range of ≈ 5 m × 3 m of x × y coordinates. This required an overlapping camera field of view while following the zig-zag pattern path.

Figure 21 clearly shows the 'W' letter shape of the four wall channels created by the wall detections. The brick detections form two large clusters being the UGV and UAV brick stacks. The UAV stack being wider was further used for the PCA analysis. Figure 21 also features a set of false positive brick detections between the two brick stacks corresponding to the position of waiting UAV2 and the starting takeoff area marked as a white rectangle on the ground.

After completion of the scanning path, the wall and brick detections are processed to create a topological map of the arena. The map contains positions of the individual wall channels ordered in a 'W' letter chain as well as lines along the particular brick types that can be deterministically divided among the three UAVs. Figure 22 shows the brick and wall detections already filtered out by the number of reoccurrences during the scanning. Figure Fig. 22a shows all the wall detections and the two bricks GMM clusters with corresponding PCA component variances proportional to width and height of the two brick stacks. Figure Fig. 22b features only the UAV stack together with the topological map consisting of 'wall 0'–'wall 3' and the red and green brick lines. The positions of the mapped walls and brick lines are then shared among UAVs and used to distribute the wall building task, as was detailed before in Sec. 4.2.

By comparing Fig. 21 and Fig. 22, it can been seen that the initial reoccurrences-based filter removes the false positive detections present in the takeoff area. However, the blue bricks present in the Fig. 21 are also filtered out and are instead incorrectly labeled as green bricks in most detections. The PCA analysis indeed selects the correct brick stack for UAVs having the larger width, with the smaller variance component being larger for the UAV stack. Figure 22b shows that the wall detections are correctly recognized as 'wall 0'–'wall 3' based on the intersections of line approximations of the wall detections. Moreover, the wall detection and mapping shows a great performance by creating the individual channels of almost the same size based only on the detections and analysis of the intersections. Finally, by comparing the raw wall detections and the mapped walls, we can see a *shadow effect* of the wall detection in 1.7 m when projected to the ground plane due to the left-to-right scanning trajectory above the wall.



(a) All wall detections and brick clusters with PCA analysis.

(b) Mapped UAV walls and bricks.

Figure 22: Topological map based on scanning of the arena.

8.2. Brick grasping

The brick grasping is another key capability required for competing in the wall building task. The grasping procedure [S10] consists of the lower-level grasping state machine described in Sec. 6.4. Figure 23 shows the evolution of the grasping states with respect to the UAV position for the first success-

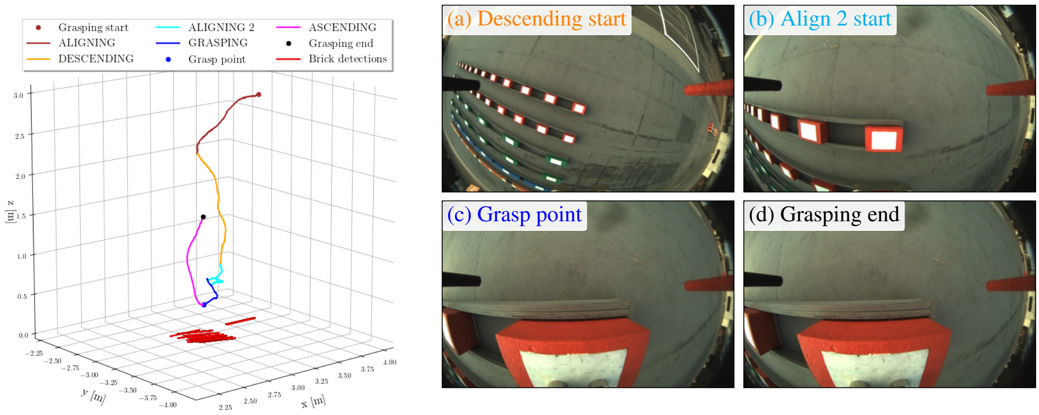


Figure 23: Grasping of a red brick with color indicated stages of the grasping procedure.

ful grasp of the red brick with UAV1 during the second restart of first trial.

Figure 23 also features the images taken by the mvBlueFOX camera that used for RGB detection of the bricks during various stages of the grasping state machine. The brick detections are shown as measured during the grasping. The rest of the detections have a mean position for x, y, heading being 3.022 m, −3.215 m, 2.825 rad, respectively, and a corresponding standard deviation of 0.029 m, 0.077 m , 0.077 rad, respectively. However, the absolute localization of the grasped brick is only relevant within the first two stages of the grasping manoeuvre (the first alignment and descent), where the UAV is guided using these estimated GPS coordinates of the brick. The later stages use direct visual servoing to estimate the UAV states using the brick detections which outperforms the accuracy of standard GPS by an order of magnitude. Our UAVs were able to target the magnetic plates on the bricks reliably within centimeter precision.

8.3. *Brick placement*

Brick placement together with grasping is one of the most important wall building capabilities. Figure 24 depicts the successful placement of the same brick being grasped in Fig. 23. After successful grasping, the UAV flies above its assigned wall channel and switches to the lower-level placement state machine using place detection (both described in Sec. 7) to guide the UAV above the release point on the appropriate location of the wall. Note that the pattern (i.e. sequence of bricks in both layers) forming individual channels was given for each trial. The brick building sequence was planed as a consecutive placement of a brick into the next unoccupied position in the wall segment — either the leftmost position in the completely-free segment or neighbouring position to the rightmost brick on the segment. Figure 24 shows the stages of the lower-level placement state machine together with the UAV positions and detections of the wall channel during alignment and placing states. Additionally, the images from both the RGB mvBlueFOX camera and the depth images from the RealSense camera are show in various stages of placement.

In Fig. 24, it can be seen that placement starts at approximately the middle of the assigned wall channel. During the alignment state, the UAV moves along the channel to the leftmost position on the empty wall. The wall detections are clearly shown to be of various lengths as the smaller portion of the wall is visible once descending during the placing state. However, the left corner of the wall is measured during the detections with a mean position for x, y, heading being 5.374 m, 9.790 m, −1.110 rad, respectively, with corresponding standard deviation of 0.139 m, 0.093 m, 0.113 rad, respectively.

The brick placing state machine is comparably simpler than the grasping state machine, since the grasping state machine needs to cover various failure stages during the grasping process. This was required to deal with failures in the grasping stage, because the UAV adds no value to the mission outcome if it does not succeed with grasping. Moreover, the grasping manoeuvre is sensitive to control accuracy and timing while also being more dangerous for the UAV. On the other hand, placing allows for a significant slack in the control of the UAV thanks to the width of the wall channel and the possibility of dropping the brick from a higher height without any physical interaction with the wall. The wall detection also worked more reliably thanks to more prominent features in sensory input as the UAV rarely lost the wall from its field of vision during testing. Lastly, we did not consider any possible correcting action for instances of improperly placed bricks.

8.4. *Wall building performance*

Finally, the performance of the wall building stage using the proposed system is depicted in Fig. 25 showing the recorded positions of both the UAV1 and UAV3 throughout the first trial with four performed restarts. All 17 grasped bricks are shown with their respective release positions, however only ten bricks stayed on the wall without bouncing off as detailed in Table 2. The recorded positions of the UAVs shown do not include the initial scanning of UAV1 already discussed in the Sec. 8.1. Instead, the positions of the UAVs once carrying bricks is highlighted. The mapped wall and brick locations are based on the

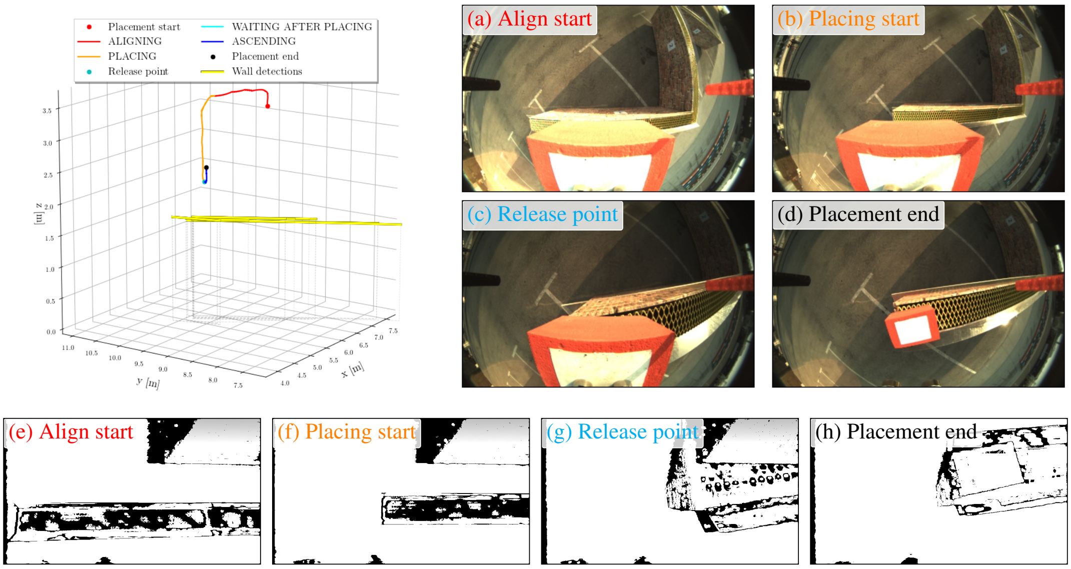


Figure 24: Placement of a red brick on the beginning of the first wall channel shown with placement stages and wall detections, together with images from the RGB camera (a)–(d) and depth camera (e)–(h). Videos from experimental testing and from the competition are available at `http://mrs.felk.cvut.cz/mbzirc-2020-uav-wall`.

scanning in the second restart as already shown in Fig. 21 and Fig. 22.

See video of the CTU-UPenn-NYU team `https://youtu.be/1-aRtSarYz4` with summary of the preparations in the Abu Dhabi desert, of competition rehearsals, and of the actual Challenge 2 MBZIRC 2020 competition trials.

8.5. Lessons learned

The most important factor driving development was the need for safety and reliability of the multi-UAV system. Interaction with the environment during grasping and placing is potentially dangerous and can easily damage the UAV. The use of real-time weight and force estimation for the detection of potentially dangerous situations was of significant benefit. A common approach of relying solely on UAV position estimation to drive the decision-making process would not be sufficient.

One of the tunable parameters of the grasping/placing maneuvers was the speed of the descent. Speed too slow increases the overall duration of the UAV being in a potentially dangerous location and allows the ground effect to build up (aerodynamic effects caused by the rotor downwash close to a ground). Alternatively, too fast of speed increases the risk of damaging the UAV due to the sudden bump caused by interaction with the brick, the wall, or the ground. On several occasions during our preparations, we experienced a complete stall of motors due to the sudden impact at higher speed, which subsequently caused a loss of onboard power and an uncontrollable, unstoppable tumbling of the UAV. Therefore, we advise caution when working with UAVs if sharp acceleration spikes may be transmitted to the UAV body. Finally, we settled on the descending speed of $0.25\,\mathrm{m\,s^{-1}}$ which showed to be the most reliable and rewarding.

This proposed wall building system depends on successful arena scanning and creation of a topological map used for planning further brick pickup and placement. As such, the brick and wall detections need to be robust and without excessive false positives that could influence the topological map creation based on statistical analysis of detections. The task was further challenging due to the stack of UGV bricks present that could not be used safely by UAVs and thus had to be recognized among the detections. During the preparations, a rather high number of false detections forced implementing significant detection filtering during scanning, requiring a minimal number of corrections in the detection map and used the iterative median filtering of detections to remove outliers. However, during the competition rehearsals, the creation of the topological map had to be further fine-tuned to, e.g., filter out already placed bricks from previous restarts.

Finally, the GPS drift significantly influenced the entire system deployment as arena borders were defined in GPS coordinates, with either the brick stack or the wall channels possibly placed too close to the borders. This prevented UAVs from flying too close to the net-protected borders and suggests that an additional LiDAR or camera-based detection of the border (or even sensor-based fix of the GPS drift) would significantly improve deployment robustness in similar competitions. An RTK-based localization is technically a possible solution to this problem, but it was not used by the team due to the penalization of RTK in scoring. However, after discussion with potential industrial partners, a solution using both the RTK GPS and the

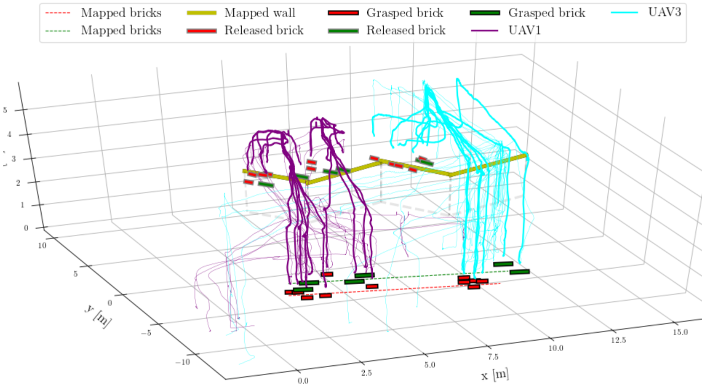


Figure 25: Visualization of wall building performance during the first trial with shown grasping/placing positions and highlighted UAV trajectories by the thicker lines when carrying a brick.

onboard local sensors would be preferred. Important construction locations could be pre-measured using the RTK (as it is nowadays common on construction sites). Combining this with the proposed onboard sensor capabilities would yield a robust cognitive system capable of reacting to changes in the environment.

The execution time of brick detection for visual servoing is critical for smooth real-time UAV control and navigation. The Intel NUC computer, by its design similar to laptop hardware, had a power-saving mode that caused irregular execution times of our methods. This problem was detected before the competition and resolving it improved the robustness of the whole system significantly.

9. Conclusions

In this paper, an autonomous system developed by CTU-UPenn-NYU team for wall building with a team of UAVs was introduced. The examined task was part of Challenge 2 of the MBZIRC 2020 where three UAVs were assigned to find, pickup, and place color-typed bricks on a prepared wall structure. The goal of the task was to maximize collected points for placing the bricks while following the prescribed wall pattern. This paper presents the key parts of the UAV system developed for the competition, including the UAV control system, the algorithms for brick and wall detection, the single robot state machine, and the multi-robot distributed approach for the task. The core autonomous capabilities of scanning the arena for wall/brick locations, autonomous grasping using visual servoing technique, and precise placement of bricks on the wall structure are described in detail. We further report the experimental results achieved during the competition trials showing the performance of the core autonomous capabilities and of the entire system. The proposed approach performed the best among all participants with 22 successfully grasped bricks in which 13 of these bricks were successfully placed during the two trials of the Challenge 2 MBZIRC competition. The entire system for the wall building task is open-sourced for the community to be used for possible deployment and future development. It can also serve as a useful reference for future robotic challenges, such as the MBZIRC that indeed serves as a great verification of robotic research.

Figure 26: CTU-UPenn-NYU team after winning the wall building Challenge 2 of MBZIRC 2020.

Acknowledgements

We thank the University of Pennsylvania and the New York University for the collaboration as the members of the CTU-UPenn-NYU team (see Fig. 26). This work would not have been possible without the tireless dedication of all the team members and support and patience from their families. The presented work has been supported by Khalifa University and by CTU grant no. SGS20/174/OHK3/3T/13.

References

[1] M. C. Tatum, J. Liu, Unmanned aircraft system applications in construction, Procedia Engineering 196 (2017) 167–175.

[2] Y. Ham, K. K. Han, J. J. Lin, M. Golparvar-Fard, Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works 4 (1) (2016) 1–8.

[3] J. Howard, V. Murashov, C. M. Branche, Unmanned aerial vehicles in construction and worker safety, American Journal of Industrial Medicine 61 (1) (2018) 3–10. doi:10.1002/ajim.22782.

[4] Khalifa University, Mohamed bin zayed international robotics challenge 2020, available online, https://www.mbzirc.com/challenge/2020, accessed on, citied on 2020/11/16.

[5] T. Baca, D. Hert, et al., Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles, in: 2018 IEEE/RSJ IROS, IEEE, 2018, pp. 1–8.

[6] T. Lee, M. Leoky, et al., Geometric tracking control of a quadrotor UAV on SE(3), in: 2010 IEEE CDC, IEEE, 2010, pp. 5420–5425.

[7] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, M. Saska, The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles, submitted to JINT (8 2020). arXiv:2008.08050v2.

[8] J. Zink, B. Lovelace, Unmanned aerial vehicle bridge inspection demonstration project, Tech. rep. (2015).

[9] Q. Lindsey, D. Mellinger, V. Kumar, Construction of cubic structures with quadrotor teams, Proc. Robotics: Science & Systems VII (2011).

[10] Q. Lindsey, et al., Distributed construction of truss structures, in: Algorithmic Foundations of Robotics X, Springer, 2013, pp. 209–225.

[11] F. Augugliaro, A. Mirjan, et al., Building tensile structures with flying machines, in: 2013 IEEE/RSJ IROS, IEEE, 2013, pp. 3487–3492.

[12] A. Mirjan, F. Augugliaro, R. D'Andrea, F. Gramazio, M. Kohler, Building a bridge with flying robots, in: Robotic Fabrication in Architecture, Art and Design 2016, Springer, 2016, pp. 34–47.

[13] D. Alejo, J. A. Cobano, et al., Collision-Free 4D Trajectory Planning in Unmanned Aerial Vehicles for Assembly and Structure Construction, Journal of Intelligent & Robotic Systems 73 (1) (2014) 783–795.

[14] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, et al., The flight assembled architecture installation: Cooperative construction with flying machines, IEEE Control Systems Magazine 34 (4) (2014) 46–64.

[15] Aerial Robotics Cooperative Assembly system, retrieved from http://www.arcas-project.eu, 2020/08/28 (2020).

[16] F. Ruggiero, V. Lippiello, A. Ollero, Aerial manipulation: A literature review, IEEE Robotics and Automation Letters 3 (3) (2018) 1957–1964.

[17] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, et al., Closed-Loop Behavior of an Autonomous Helicopter Equipped with a Robotic Arm for Aerial Manipulation Tasks, International Journal of Advanced Robotic Systems 10 (2) (2013) 145–154.

[18] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, et al., Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator, in: 2014 IEEE ICRA, IEEE, 2014, pp. 2107–2112.

[19] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, A. Franchi, 6D physical interaction with a fully actuated aerial robot, in: 2017 IEEE ICRA, IEEE, 2017, pp. 5190–5195.

[20] J. Munoz-Morera, I. Maza, C. J. Fernandez-Aguera, F. Caballero, A. Ollero, Assembly planning for the construction of structures with multiple UAS equipped with robotic arms, in: 2015 IEEE ICUAS, IEEE, 2015, pp. 1049–1058.

[21] J. Thomas, G. Loianno, et al., Toward image based visual servoing for aerial grasping and perching, in: 2014 IEEE ICRA, IEEE, 2014, pp. 2113–2118.

[22] P. Ramon Soria, B. C. Arrue, A. Ollero, Detection, location and grasping objects using a stereo sensor on UAV in outdoor environments, Sensors 17 (1) (2017) 103.

[23] A. Gawel, M. Kamel, T. Novkovic, J. Widauer, D. Schindler, B. P. Von Altishofen, R. Siegwart, J. Nieto, Aerial picking and delivery of magnetic objects with MAVs, in: 2017 IEEE ICRA, IEEE, 2017, pp. 5746–5752.

[24] K. Feng, W. Li, S. Ge, F. Pan, Packages delivery based on marker detection for UAVs, in: 2020 IEEE CCDC, IEEE, 2020, pp. 2094–2099.

[25] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, et al., Localization, Grasping, and Transportation of Magnetic Objects by a team of MAVs in Challenging Desert like Environments, IEEE Robotics and Automation Letters 3 (3) (2018) 1576–1583.

[26] V. Spurny, T. Baca, M. Saska, R. Penicka, T. Krajnik, J. Thomas, D. Thakur, G. Loianno, et al., Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs, Journal of Field Robotics 36 (1) (2019) 125–148.

[27] A. R. Castano, F. Real, P. Ramon-Soria, J. Capitan, V. Vega, B. C. Arrue, et al., Al-Robotics team: A cooperative multi-unmanned aerial vehicle approach for the Mohamed Bin Zayed International Robotic Challenge, Journal of Field Robotics 36 (1) (2019) 104–124.

[28] NicaDrone, Electro Permanent Magnet OpenGrab EPM V3, retrieved August 22, 2020, from https://nicadrone.com/products/epm-v3 (2020).

[29] R. Bahnemann, M. Pantic, M. Popovic, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, J. Nieto, The ETH-MAV Team in the MBZ International Robotics Challenge, Journal of Field Robotics 36 (1) (2019) 78–103.

[30] M. Kamel, J. Alonso-Mora, R. Siegwart, J. Nieto, Robust Collision Avoidance for Multiple Micro Aerial Vehicles Using Nonlinear Model Predictive Control, in: 2017 IEEE/RSJ IROS, IEEE, 2017, pp. 236–243.

[31] M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, S. Behnke, Team NimbRo at MBZIRC 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles, Journal of Field Robotics 36 (1) (2019) 204–229.

[32] M. Beul, S. Behnke, Fast full state trajectory generation for multirotors, in: 2017 IEEE ICUAS, IEEE, 2017, pp. 408–416.

[33] Open Robotics, Robotic Operating System, retrieved July 22, 2020, from https://www.ros.org (2020).

[34] P. Schillinger, S. Kohlbrecher, O. von Stryk, Human-robot collaborative high-level control with application to rescue robotics, in: 2016 IEEE ICRA, IEEE, 2016, pp. 2796–2802.

[35] J. Bohren, S. Cousins, The smach high-level executive [ros news], IEEE Robotics Automation Magazine 17 (4) (2010) 18–20.

[36] M. Schwarz, nimbro_network - ros transport for high-latency, low-quality networks, available online, https://github.com/AIS-Bonn/nimbro_network, accessed on 2020/11/16.

[37] E. Galceran, M. Carreras, A survey on coverage path planning for robotics, Robotics and Autonomous Systems 61 (12) (2013) 1258–1276.

[38] A. Grunnet-Jepsen, J. N. Sweetser, J. Woodfill, Best-Known-Methods for Tuning Intel® RealSense™ D400 Depth Cameras for Best Performance, Tech. rep., Intel Corporation: Satan Clara, CA, USA (2018).

[39] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools (2000).

[40] D. Scaramuzza, A. Martinelli, R. Siegwart, A toolbox for easily calibrating omnidirectional cameras, in: 2006 IEEE/RSJ IROS, IEEE, 2006, pp. 5695–5701.

[41] D. Reynolds, Gaussian mixture models, Encyclopedia of Biometrics 741 (2009) 659–663.

[42] I. Jolliffe, Principal Component Analysis, 2011, pp. 1094–1096.

[43] P. Stibinger, G. Broughton, F. Majer, Z. Rozsypalek, A. Wang, K. Jindal, A. Zhou, D. Thakur, G. Loianno, T. Krajnik, M. Saska, Mobile Manipulator for Autonomous Localization, Grasping and Precise Placement of Construction Material in a Semi-structured Environment, Submitted to Robotics and Automation Letters (2020). arXiv:2011.07972.

Chapter 5

Radiation Dosimetry by UAVs

The first core publication related to ionizing radiation dosimetry presents the results from the first year of the VZLUSAT-1 nanosatellite operation. The satellite [38a] was launched in June 2017 [37a]. It carried, among other scientific instruments, a miniaturized X-ray telescope with the Timepix particle detector [39a]. The author carried out the design, development, and commissioning of the satellite's payload and manages ongoing data acquisition and processing. This research represents the author's first step into the particle imaging and radiation dosimetry community. Without these contributions and gained know-how, the following work on radiation localization and mapping by UAVs would have likely not been possible. Furthermore, one of the tangents of this outer space research is the author's involvement in one of NASA's suborbital sounding rocket flights [78], [40a], [106]. The thesis author designed a hardware readout interface and software, based on the Robot Operating System [15a], for two Timepix detectors onboard a sub-orbital rocket [42a]. The following article shows results of one year of orbital measurements onboard the VZLUSAT-1 satellite using the Timepix sensor.

> [4c] **T. Baca**, M. Jilek, I. Vertat, M. Urban, O. Nentvich, R. Filgas, *et al.*, "Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements," *Journal of Instrumentation*, vol. 13, no. 11, p. C11010, 2018

The second core publication in this field provides an overview of the potential use of Timepix particle imaging detectors onboard UAVs. The article discusses the detector's critical properties and their relation to various methods of obtaining a directional measurement of a radiation source, such as pinhole apertures, X-ray optics, and collimators. A method for real-time classification of the measured particle tracks is proposed and developed. Existing particle track classifiers are either not available (NASA [164], ESA [113], [141]), built into laboratory software [135], [166], [174], or not suited for real-time applications [83]. Therefore, an open-source classifier and ROS drivers for the Timepix detector is proposed. Labeled datasets[1] are also provided that can be used for training of new classifiers. Furthermore, it presents a real-time Monte-Carlo simulation model for the Compton camera and concludes that the Compton camera principle shows promising properties.

> [5c] **T. Baca**, M. Jilek, P. Manek, P. Stibinger, V. Linhart, J. Jakubek, *et al.*, "Timepix Radiation Detector for Autonomous Radiation Localization and Mapping by Micro Unmanned Vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2019, pp. 1–8

[1] `https://github.com/vzlusat`

The next core publication in this field focuses on distributed localization of ionizing radiation sources using multiple UAVs equipped with the Timepix sensor. This work presents real-time radiation source localization and estimation methods for UAVs equipped with the Timepix sensors. A path planning approach that exploits the asymmetrical intensity measurement response of the Timepix detector was proposed. The approach uses multiple UAVs and their ability to vary the orientation of the sensor in space to estimate the direction to the radiation source. Moreover, the work presents verification of real-world measurements onboard a UAV against the proposed real-time Monte-Carlo ray-tracing simulation model of the Timepix sensor. Realistic simulations in the Gazebo/ROS robotic simulator demonstrate the feasibility of the method.

[1c] P. Stibinger, **T. Baca**, and M. Saska, "Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles With Pixel Detectors in Real-Time," *IEEE Robotics and Automation Letters*, vol. 5, pp. 3634–3641, 2 Apr. 2020, ISSN: 2377-3766

Jinst

Published by IOP Publishing for Sissa Medialab

Received: *October 14, 2018*
Accepted: *November 5, 2018*
Published: *November 16, 2018*

20TH International Workshop on Radiation Imaging Detectors
24–28 June 2018
Sundsvall, Sweden

Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements

T. Baca,[a,1] M. Jilek,[a] I. Vertat,[b] M. Urban,[a] O. Nentvich,[a] R. Filgas,[d] C. Granja,[e] A. Inneman[a,c] and V. Daniel[e]

[a] *Czech Technical University in Prague, Faculty of Electrical Engineering,
Technicka 2, Prague, Czech Republic*
[b] *University of west Bohemia, Faculty of Electrical Engineering, Univerzitni 8, Pilsen, Czech Republic*
[c] *Rigaku Innovative Technologies, Za Radnici 868, Dolni Brezany, Czech Republic*
[d] *Czech Technical University in Prague, Institute of Experimental and Applied Physics,
Horska 3, Prague, Czech Republic*
[e] *Czech Aerospace Research Centre, Beranovych 130, Prague, Czech Republic*

E-mail: tomas.baca@fel.cvut.cz

Abstract: The VZLUSAT-1 satellite, the first Czech CubeSat, was successfully launched on June 23, 2017, to a 510 km Sun-synchronous low-Earth orbit. It carries several scientific payloads including a Timepix detector as focal plane imager for the X-Ray telescope onboard. The Timepix detector contributes significantly to the satellite data collection, with more than 25 000 sampling acquisitions in the first year of deployment. Despite limitations of the satellite attitude control system, necessary for capturing X-Ray images of the Sun, the Timepix detector allows measuring the space radiation environment along the satellite orbit. As of September 2018, we conducted 33 whole-Earth mappings, recording radiation doses around the planet. Further, we show data from scans of the South Atlantic Anomaly and polar radiation horns, where the location and acquisition time were tailored to minimize event pile-up and particle track overlap. Since October 2017, the optics segment of the onboard X-Ray telescope was deployed, which exposed the Timepix detector unshielded to free open space. This change produced entirely new observations namely of low energy charged particles and a significant increase of measured particle flux. We also registered the effects of exposing the sensor to direct intense sunlight. We will summarize on the actual performance of the custom readout interface, which exceeds expectations in the constrained environment of the low-cost and low-powered CubeSat nanosatellite.

Keywords: On-board space electronics; Particle detectors; Space instrumentation; X-ray detectors and telescopes

[1] Corresponding author.

https://doi.org/10.1088/1748-0221/13/11/C11010

Contents

1 Introduction 1
1.1 VZLUSAT-1 nanosatellite 1
2 Timepix payload, commissioning and constraints 2
3 Operation modes 3
4 Overview of orbital data 4
5 Conclusions 7

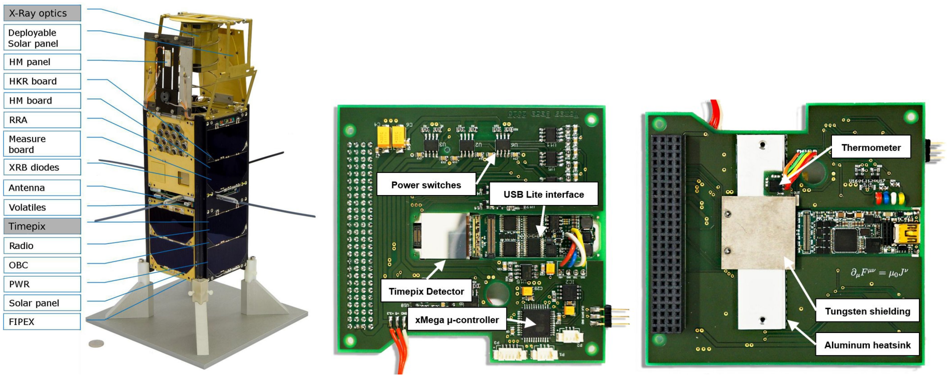
1 Introduction

We present a successful utilization of the Timepix sensor onboard VZLUSAT-1, a low-powered and low-cost CubeSat satellite. We provide information and know-how regarding the operation of the sensor in the constrained environment of the probe. A novel data were gathered allowing analysis of the density of various ionizing particle types in the low-Earth orbit (LEO).

The Timepix detector has been previously launched to LEO orbit onboard the International Space Station (ISS) [1] and onboard satellites [2, 3]. The crew of the ISS uses Timepix detectors with *USB Lite* interfaces as online quantum imaging dosimeters to precisely measure the dose rate in wide dynamic range. In all cases, Timepix detectors were deployed either in significantly larger satellites than VZLUSAT-1 or in the atmosphere onboard the ISS. The ISS occupies the LEO with altitude in 405 km and inclination 51.6°. The ESA's Probe-V satellite carries the SATRAM experiment [2] in 820 km Sun-synchronous orbit. SATRAM contains the Timepix detectors as a spacecraft payload for minisatellites conducting a pre-programmed loop of acquisition with 0.2 s, 2 s and 20 s acquisition times. The LUCID experiment utilizes five Timepix sensors onboard the British satellite TechDemoSat-1. The satellite orbited Earth for over two years continuously collecting data. Results from LUCID were published showing data comparable to those presented in this manuscript [3].

1.1 VZLUSAT-1 nanosatellite

The VZLUSAT-1 is a 2U CubeSat nanosatellite developed by the Czech Aerospace Research Centre (VZLU) as a technology platform demonstrator. The two-unit CubeSat with dimensions $10 \times 10 \times 34\,\mathrm{cm}^3$ (after deploying external modules) was launched to Sun-synchronous low-Earth orbit (LEO) on June 23$^{\mathrm{rd}}$, 2017. The LEO orbit of 510 km altitude and inclination of 97.4° allows the satellite to interact with the South-Atlantic Anomaly (SAA) as well as the Southern and Northern radiation belts. The satellite serves as a technological demonstrator for novel sensors and materials which were designed specifically for space radiation applications. The full description of the satellite and its payloads can be found in [4]. The main payload, the miniaturized lobster-eye X-Ray telescope, incorporates the Timepix detector as a focal plane imager [5].



(a) The VZLUSAT-1 satellite. **(b)** The Timepix payload board in CubeSat standard with the USB Lite interface.

Figure 1. The flight model of VZLUSAT-1 (a) with all its onboard modules deployed. One of the payloads is the Timepix sensor board (b) which is responsible for readout and pre-processing of measured data [5].

2 Timepix payload, commissioning and constraints

The satellite payload [5], is a standalone module, capable of image acquisition, readout, pre-processing, and compression directly onboard the satellite. It is crucial to have a complete processing pipeline onboard, due to minimal communication capabilities of the satellite [6]. Having a complete onboard processing pipeline is unique when comparing to the related projects [2, 3], which do not introduce onboard processing. Table 1 summarizes the parameters and capabilities of the Timepix payload.

Table 1. List of the Timepix payload parameters and capabilities.

Detector	Timepix
Sensor	300 µm silicon, aluminum-coated
Cooling	passive convection
Shielding	backside 0.5 mm tungsten
Readout	Customized *USB Lite* interface
Processing unit	ATxMega127a4u + 256kB FRAM
Min. acquisition time	0.001 s
Max. acquisition time	not limited
Acquisition time overhead	15 s including equalization upload and readout
Frame processing time	3 to 60 s depending on the pixel occupancy
onboard processing	Binning, energy histogram, track filtering
Max. readout rate	once per 90 s
VZLUSAT-1 storage size	4 MB shared between payloads
VZLUSAT-1 down link	approx. 30 kB/day
Acquisition control	remote (manual) and onboard (script-driven)

Commissioning: was conducted after communication with the satellite was established. Systems and payloads were tested individually, and their state was examined for potential damage. The Timepix detector responded correctly to all inputs. Exposures were acquired to test the noise edge given the sensor equalization loaded in the satellite read-only memory. The obtained data in the new environmental conditions required adjustment of the detector settings by increasing per-pixel signal threshold to 5 keV. After the first year of operation, the detector threshold was shifted once more, to approx. 6 keV, due to new noise near the bonding edge of the sensor. Close to the end of the first year of operation, just 3 pixels appear to be damaged, requiring masking.

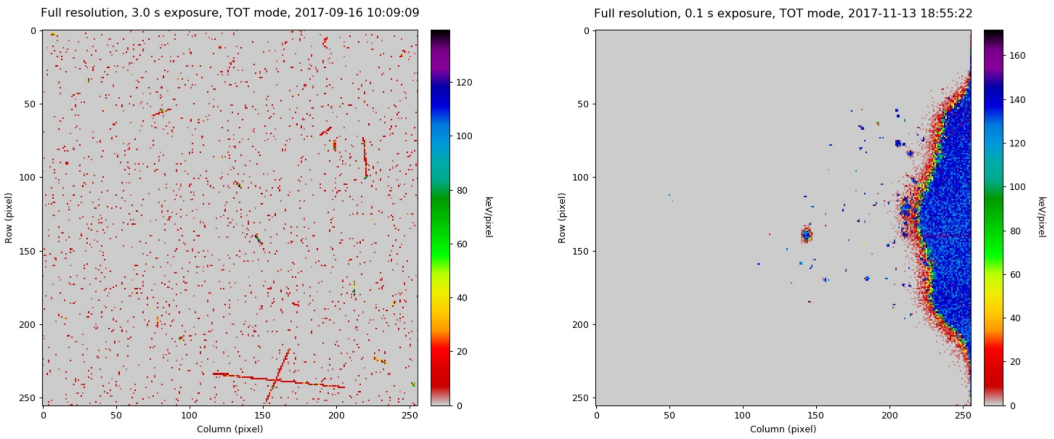
Up till now, the satellite's attitude determination and control system (ADCS) could not be fully operated. Due to a malfunction, the 3D orientation of the satellite along its orbit is uncontrollable. Thus the optical axis of the telescope cannot be pointed in the desired direction. This condition limits the capability of the onboard X-Ray telescope to perform long exposures while being pointed at astronomical objects. Nevertheless, short automated exposures of the Sun can be still performed using the onboard ultra-violet (UV) trigger [7].

3 Operation modes

Since its commissioning, the Timepix payload has been operated in 3 distinct modes of operation.

UV-triggered Solar imaging: uses an autonomous UV trigger to expose the Sun using Timepix and the X-Ray optics. Without the functioning ADCS system, the opportunity of pointing directly to Sun is low. Although the system gathered over 100 images, most of them carry no direct observation of the Solar disc. Some data suggest that the Solar disc went across the field-of-view of the optics during the acquisition. However, due to long exposures (3.0 sec, later 0.1 s) the observation of the focused solar disc cannot be confirmed. Moreover, images taken after the optics deployment (refer to [4, 5] for detail information) show intense direct visible light illuminating the Timepix sensor. The illumination is likely to happen due to the lack of baffle in the telescope. Exposures of direct sunlight suggest that most of the observations were taken with the Solar disc being approx. 6 deg outside of the field-of-view of the X-Ray telescope. Images in figure 2 show a sample of typical data taken with the UV-trigger.

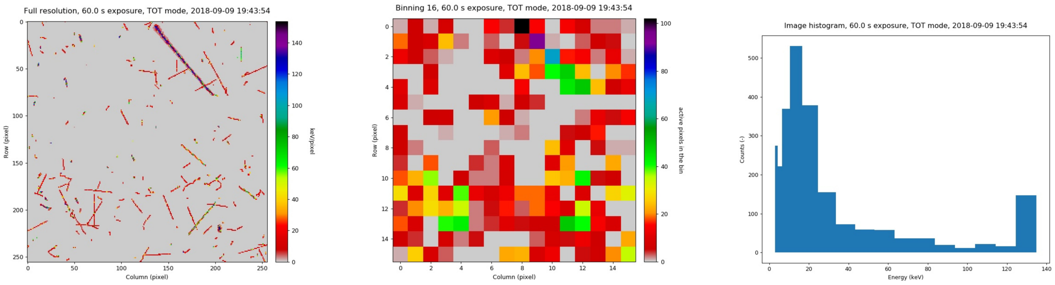
Binned sampling, frequent scanning: is conducted approx. once per fortnight. It gathers post-processed data during a 24 h session, taking an image every 90 s. The data contain original images (figure 3a) binned to 16 × 16 px resolution (figure 3b) and a histogram of pixel values across the whole image (figure 3c). The binning was designed for the X-Ray telescope, where the background radiation would be filtered out, leaving only the tracks of X-Ray photons. The use of such methods of compression for data containing background radiation is not optimal. However, even with such compression, information about the track types can be partially reconstructed, as shown in the section 4. Parameters of the acquisition are fixed for every image during the high-frequency scanning. Notably, the acquisition time is set to 0.05 s, which provides images free of saturated pixels. Figure 4a shows an example of an interpolated radiation intensity map with highlighted measurement locations.



(a) 3 s acquisition of Solar X-Ray radiation.

(b) 0.1 s acquisition of direct sunlight.

Figure 2. Sample of data acquired by Timepix using the automatic Solar UV-trigger of the X-Ray telescope.



(a) Full-resolution image

(b) Binning of the active pixels 16 × 16.

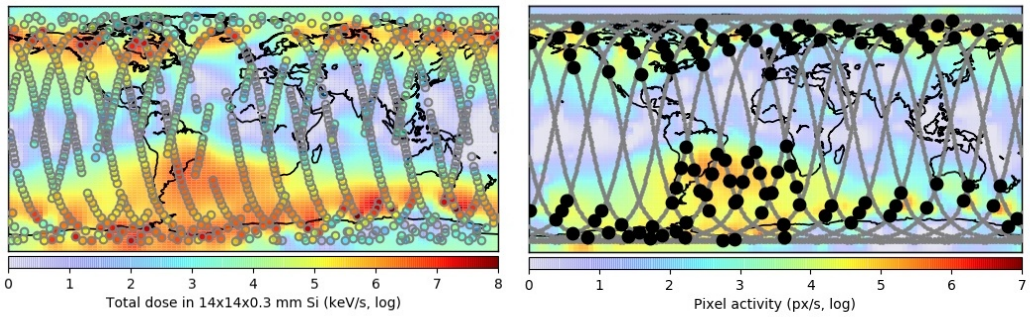
(c) Energy histogram.

Figure 3. Full resolution images acquired by Timepix are converted from the original full-resolution (a) to binned images (b) and computed energy histograms (c). For high-frequency planetary scanning, only (b) and (c) are stored due to limited onboard data storage and low downlink data transfer capacity.

Full-resolution sampling: of polar belts and SAA utilizes a scripting mechanism of the VZLUSAT-1, which allows iterating over a predefined list of commands. The scripts are planned on Earth using the historical data gathered by the satellite and are uploaded regularly from the ground segment. See figure 4b for an example of a scanning plan. The measurement position on the orbit and the parameters of the acquisition are optimized to enable registration of full resolution images (figure 3a) and to minimize event pileup and allow particle track classification. Such measures are necessary to allow further processing since the intensity of radiation varies by up to 8 orders of magnitude across the entire orbit. Processed results of the full-resolution scanning are presented in section 4.

4 Overview of orbital data

The satellite conducted a total of 26 212 image acquisitions, from which 1 608 were retrieved in full-resolution. A total of 33 full-day, high-frequency scanning sessions and eight full-resolution scanning of polar belts and South-Atlantic anomaly produced valuable data, which help us to


0
1
2
3
4
5
6
7
8
Total dose in 14x14x0.3 mm Si (keV/s, log)
0
1
2
3
4
5
6
7
Pixel activity (px/s, log)

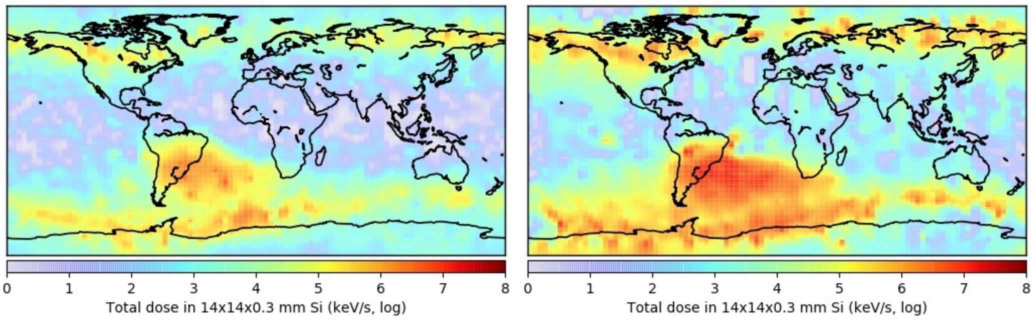
(a) Radiation intensity map obtained from post-processed binned data in a single day. Circles mark the actual points of measurement.

(b) Plan of full-resolution scanning using historical data. Black dots mark the pre-planed points of measurement within a predicted orbit.

Figure 4. Background radiation in LEO is being scanned in two modes: (a) frequent scanning producing post-processed binned data, (b) full-resolution sparse scanning using planned scripts.

understand the space radiation environment in LEO. The ground segment downloaded 34 MB of data, which contain all the acquisitions mentioned above.

In the first months of operation, the Timepix payload focused on capturing an image of the Sun using the X-Ray telescope. The initial phase of observations was conducted with the telescope's optics retracted inside the satellite, where the optics functions as a collimator. Nineteen images were taken, before the optics were deployed on October 4th, 2017. After deploying the optics an additional 88 images were taken. However, thanks to a free motion of the satellite due to the malfunction of the ADCS system, no sharp image of the Sun was captured. We can only estimate that the telescope observed solar X-Ray photons which were evenly distributed across the sensor due to the motion of the Sun across the field-of-view during several of the 3 s exposures (see figure 2).

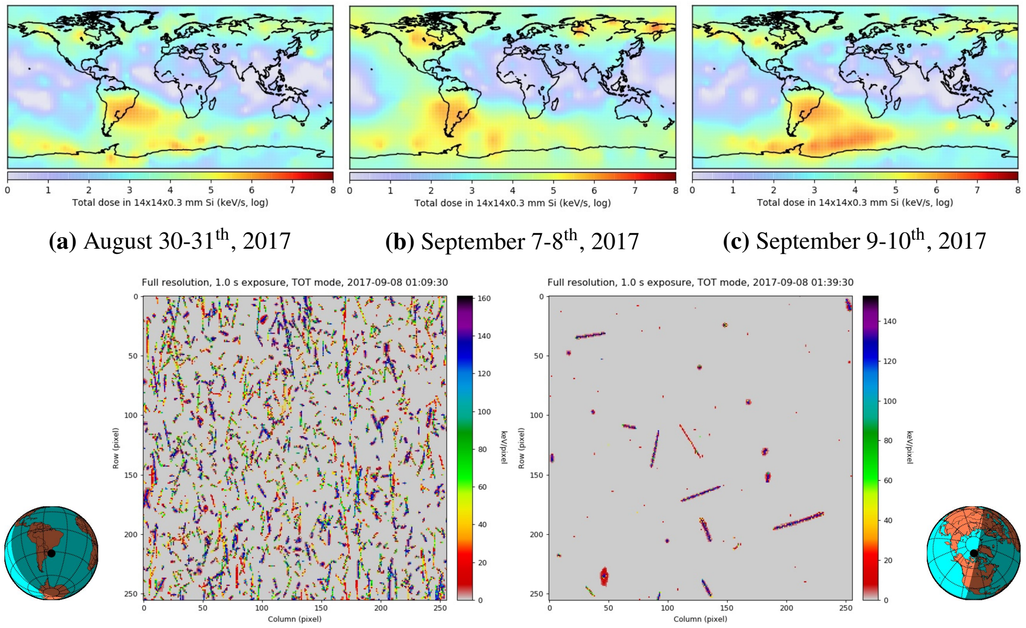

0
1
2
3
4
5
6
7
8
Total dose in 14x14x0.3 mm Si (keV/s, log)
0
1
2
3
4
5
6
7
8
Total dose in 14x14x0.3 mm Si (keV/s, log)

(a) Radiation intensity map before optics deployment.

(b) Radiation intensity map after optics deployment.

Figure 5. Earth maps of radiation intensity registered by Timepix before and after deployment of the X-Ray telescope optics. After deployment, the Timepix detector is partially exposed to free open space, which leads to an increase of the recorded radiation intensity by up to 2 orders of magnitude.

Meanwhile, the deployment of the optics partially exposed the Timepix detector to open free space. Since then the perceived level of background radiation significantly increased by up to 2



(a) August 30-31th, 2017 **(b)** September 7-8th, 2017 **(c)** September 9-10th, 2017

(d) Effects of solar flare in SAA. **(e)** Effects of solar flare above North pole.

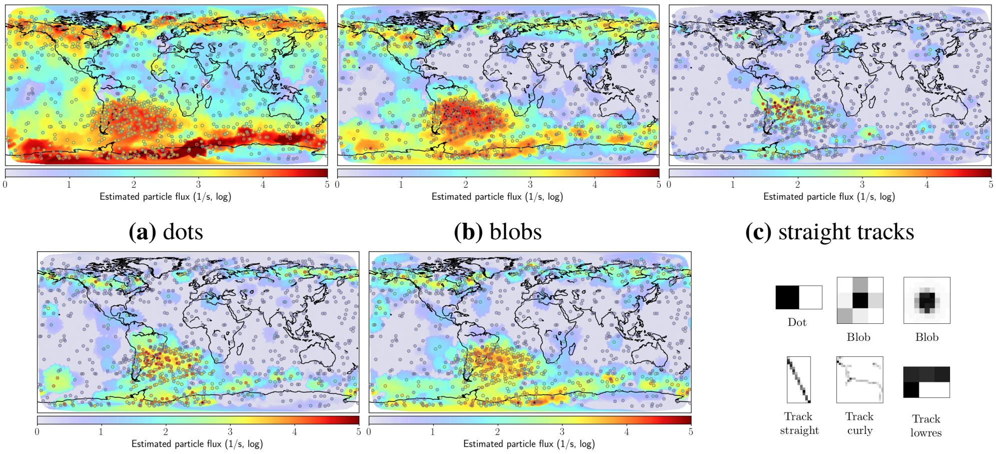
Figure 6. Top row: radiation maps captured (a) before, (b) during and (c) after effects of the X9.3 Solar flare on September 6th, 2017. The effects were observable up to several days after the event. Bottom row: Timepix frames characteristic for specific locations along the satellite orbit.

orders of magnitude. The additional signal is caused by low-energy charged particles, which are previously blocked by the satellite thin housing. Figure 5 shows the levels of perceived radiation before and after optics deployment. Additionally, the effects of direct sunlight can be seen as well (figure 2b) despite the aluminum coating of the detector. We do not observe any degradation of the detector due to sunlight exposure.

The X9.2 Solar flare: on September 6th, 2017 was the strongest coronal mass ejection event in a decade. Additional measurement time was dedicated to examining the effects of the flare on the in orbit radiation. As depicted in figure 6, the effect consists of increased radiation intensity farther from the polar regions and the SAA, as well as above Central America. Moreover, the composition of the added radiation shows increased intensity of heavy charged particles (protons) (figures 6d and 6e). Equivalent data have not been captured ever since the event.

Full-resolution images: allow to perform detailed single particle track analysis. Particle tracks were classified into morphological classes [8] using a custom random forest classifier [9]. Blobs, which are generally taken for tracks of protons and ions, can be seen mostly in the area of the so-called south-Atlantic anomaly. Figure 7 shows the map for each of the classes.

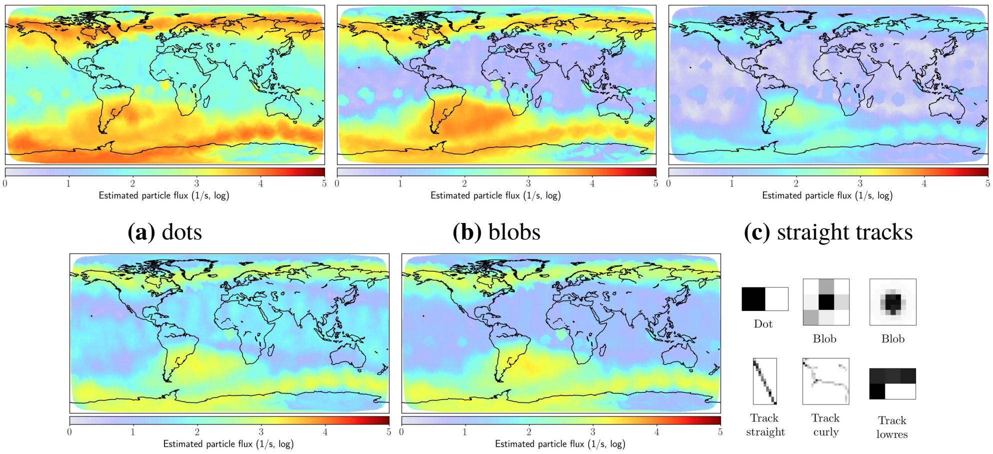
Particle track reconstruction: was performed on the post-processed binned data (figures 3b and 3c), this was conducted using a random forest regression model [9], developed for the reconstruction of compressed data. The results, shown in figure 8, suggest that machine learning methods



(a) dots **(b)** blobs **(c)** straight tracks

(d) curly tracks **(e)** low resolution tracks **(f)** track examples

Figure 7. Measurements and interpolation of particle track type flux for (a) *dots*, (b) *blobs*, (c) *straight tracks*, (d) *curly tracks* and (e) *low resolution tracks*. Examples of track shapes are shown in (f). The maps were constructed out of 1030 full-resolution images captured between June 23rd, 2017 and November 4th, 2018. The positions of the measurements are represented by dots. Data were interpolated using k-NN.



(a) dots **(b)** blobs **(c)** straight tracks

(d) curly tracks **(e)** low resolution tracks **(f)** track examples

Figure 8. Regression of particle track type flux for (a) *dots*, (b) *blobs*, (c) *straight tracks*, (d) *curly tracks* and (e) *low resolution tracks*. Examples of track shapes are shown in (f). The maps were reconstructed using 16 877 post-processed images recorded between June 23rd, 2017 and September 17th, 2018.

allow estimating track types in lossy-compressed data, which is likely with resources constrained missions as VZLUSAT-1.

5 Conclusions

The VZLUSAT-1 CubeSat nanosatellite has successfully passed its first year of operation. Using the Timepix detector, it gathered data of in-orbit radiation at 510 km Sun-synchronous low-Earth orbit.

After deploying the X-Ray optics, the onboard configuration becomes the first partial exposure of Timepix to free open space as well as the first to utilize Timepix by a CubeSat satellite. Machine learning methods were used to analyze compressed and binned data to obtain estimates radiation intensity distributions across satellite orbit. Future work includes detailed physics-based data analysis according to particle-type classification [10]. Earth-wide radiation maps are produced including track-type classification. The Timepix payload brings direct in orbit data valuable for environmental and radiation effects studies as well as for potential future missions aiming at LEO X-Ray astronomical imaging using pixel detectors.

Acknowledgments

The work has been on behalf of Medipix2 collaboration and was supported by the Grant Agency of the Czech Republic under grant no. 18-10088Y and by the Czech Technical University grant no. SGS17/187/OHK3/3T/13 and SGS18/186/OHK3/3T/13.

References

[1] X. Llopart, R. Ballabriga, M. Campbell, L. Tlustos and W. Wong, *Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements*, Nucl. Instrum. Meth. **A 581** (2007) 485 [*Erratum ibid.* **A 585** (2008) 106-108].

[2] C. Granja, S. Polansky, Z. Vykydal, S. Pospisil, A. Owens, Z. Kozacek et al., *The SATRAM timepix spacecraft payload in open space on board the proba-v satellite for wide range radiation monitoring in LEO orbit*, Planet. Space Sci. **125** (2016) 114.

[3] W. Furnell, A. Shenoy, E. Fox and P. Hatfield, *First results from the LUCID-Timepix spacecraft payload onboard the TechDemoSat-1 satellite in Low Earth Orbit*, arXiv:1810.12876.

[4] M. Urban, O. Nentvich, V. Stehlikova, T. Baca, V. Daniel and R. Hudec, *VZLUSAT-1: Nanosatellite with miniature lobster eye x-ray telescope and qualification of the radiation shielding composite for space application*, Acta Astronaut. **140** (2017) 96.

[5] T. Baca, M. Platkevic, J. Jakubek, A. Inneman, V. Stehlikova, M. Urban et al.,, *Miniaturized X-Ray telescope for VZLUSAT-1 nanosatellite with Timepix detector*, 2016 *JINST* **11** C10007.

[6] I. Vertat, R. LInhart, M. Pokorny, J. Masopust, P. Fiala and J. Mraz, *Small satellite ground station in Pilsen – experiences with VZLUSAT-1 commanding and future modifications toward open reference ground station solution*, in proceedings of the *28th International Conference Radioelektronika*, Prague, Czech Republic, 2018, pp. 1–6.

[7] I. Vertat, R. Linhart, J. Masopust, A. Vobornik and L. Dudacek, *Earth's thermal radiation sensors for attitude determination systems of small satellites*, *Contrib. Astron. Obs. Skalnate Pleso* **47** (2017) 157.

[8] T. Holy, E. Heijne, J. Jakubek, S. Pospisil, J. Uher and Z. Vykydal, *Pattern recognition of tracks induced by individual quanta of ionizing radiation in Medipix2 silicon detector*, Nucl. Instrum. Meth. **A 591** (2008) 287.

[9] M. Jilek, *Processing of radiation data from the Timepix sensor on the VZLUSAT-1 satellite*, MSc thesis, Czech Technical University in Prague, Faculty of Electrical Engineering (2018).

[10] C. Granja, J. Jakubek, S. Polansky, V. Zach, P. Krist, D. Chvatil et al., *Resolving power of pixel detector timepix for wide-range electron, proton and ion detection*, Nucl. Instrum. Meth. **A 908** (2018) 60.

2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Timepix Radiation Detector for Autonomous Radiation Localization and Mapping by Micro Unmanned Vehicles

Tomas Baca[1], Martin Jilek[1], Petr Manek[2], Petr Stibinger[1], Vladimir Linhart[3], Jan Jakubek[4] and Martin Saska[1]

***Abstract*— A system for measuring radiation intensity and for radiation mapping by a micro unmanned robot using the Timepix detector is presented in this paper. Timepix detectors are extremely small, but powerful 14×14 mm, 256×256 px CMOS hybrid pixel detectors, capable of measuring ionizing alpha, beta, gamma radiation, and heaving ions. The detectors, developed at CERN, produce an image free of any digital noise thanks to per-pixel calibration and signal digitization. Traces of individual ionizing particles passing through the sensors can be resolved in the detector images. Particle type and energy estimates can be extracted automatically using machine learning algorithms. This opens unique possibilities in the task of flexible radiation detection by very small unmanned robotic platforms. The detectors are well suited for the use of mobile robots thanks to their small size, lightweight, and minimal power consumption. This sensor is especially appealing for micro aerial vehicles due to their high maneuverability, which can increase the range and resolution of such novel sensory system. We present a ROS-based readout software and real-time image processing pipeline and review options for 3-D localization of radiation sources using pixel detectors. The provided software supports off-the-shelf FITPix, USB Lite readout electronics with Timepix detectors.**

I. INTRODUCTION

Before remotely controlled and autonomous mobile robots become available, human presence was required for any in situ measurement. Be it at the bottom of a sea or outside of the Earth's atmosphere, sensing in unreachable places was impossible. Nowadays, terrestrial sensing is not short on utilizing robotic platforms in hazardous environments. Since the widespread of Unmanned Aerial Vehicles (UAVs), often called *drones*, much effort was directed towards creating flying platforms capable of sensing the environment [1]. Aircraft can traverse ground obstacles and move quickly to the desired location when compared to ground robots. Although, the presented system is designed for use with any mobile platform, including even satellites, where it was successfully deployed [2], [3], let us put this paper into the context of MAVs, with their highest applicability,

Micro Aerial Vehicles (MAVs) are small UAVs which can be handled by a single person. Multirotor helicopters are popular MAVs due to their simple construction and low maintenance. As the technology of small and intelligent aircraft became available, many fields started to utilize new options for carrying sensor equipment. Security and rescue forces utilize camera equipment and often thermal imaging cameras to assist ground forces during environmental disasters like earthquakes and floods [4], [5]. In research and science, unmanned aircraft have various roles, from testbeds for control algorithms to autonomous sensor carriers. This paper will focus on a novel sensor setups for autonomous ionizing radiation mapping and localization by UAVs.

Demand for UAV platforms capable of localizing an unknown radiation source is increasing. Early attempts to design a radiation detection module for a UAV was presented in [6]. Fixed-wing aircraft, equipped with Kromek 1 cm^3 gamma-ray spectrometer, was used to scan legacy mines in England [7]. Area of the size of $300\ \mathrm{m} \times 400\ \mathrm{m}$ was covered during the total of 5 hours of flight, and a radiation map was created ex-post. A solution suited for a search for compact sources of radiation was presented in [8], [9]. The system utilizes 5 cm^3 scintillator with an air sampler, which takes measurement of *counts per second* in 1 s intervals. Authors tested that the aircraft can detect presence of $^{137}\mathrm{Cs}$ (activity $2.3\ \mathrm{GBq}$) and $^{60}\mathrm{Co}$ (activity $1.1\ \mathrm{GBq}$) from a distance up to 50 m, while flying at 70 km/h. The plane followed a path designed by a human operator. Another remotely controlled aircraft system is proposed in [10].

Since Fukushima Daiichi nuclear power plant (FDNPP) incident, research groups aim to prepare aircraft, that would remotely scan the affected area without endangering human workers [11], [12], [7], [13], [14], [15], [16]. Authors of [14] present an aerial solution equipped with two gamma-ray spectrometers. Data would be transmitted to a ground station over a data link during the flight. In [16] a multi-UAV approach was taken to enhance the potential yield of information gain from onboard sensors. Three formation types and trajectory schemes for three fixed-wing aircraft are proposed in a scenario with simulated sensor scanning. A multirotor MAV was used in [11] to carry lightweight CdZnTe *Kromek* spectrometer with 1 cm^3 of detection material. Real-world experiments, situated in 20 m^2 area, showed a process of creating a radiation map where several uranium samples were located. During the flight, the unmanned vehicle followed a flight plan with a set of waypoints. Scanning of 1 km^2 area was also conducted using the *Kromek* detector, near FDNPP [12], [7]. The fixed-wing plane carried a laser rangefinder, which was later used to create a 3-D map of the radiation above the affected area. The plane was controlled remotely or flew according to a pre-fabricated flight plan.

A multitude of published work relies on Kromek spec-

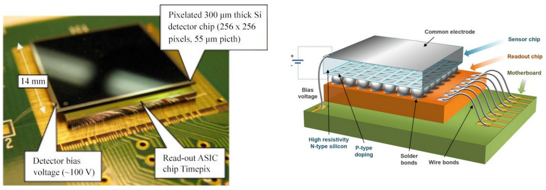
[1] Authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, Prague 6, tomas.baca@fel.cvut.cz.

[2] Author is with the Institute of Experimental and Applied Physics, Czech Technical University in Prague, Husova 240/5, Prague 1.

[3] Author is with the Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Prague 1.

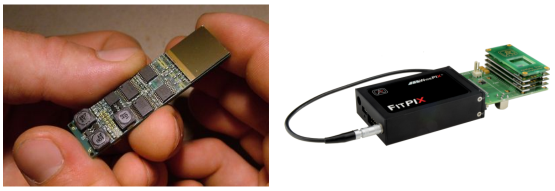
[4] Author is with the Advacam s.r.o., Prague, Czech Republic, U Pergamenky 12, Prague 7.

978-1-7281-4004-9/19/$31.00 ©2019 IEEE


Pixelated 300 μm thick Si detector chip (256 x 256 pixels, 55 μm picth)
14 mm
Detector bias voltage (~100 V)
Read-out ASIC chip Timepix
Common electrode
Sensor chip
Readout chip
Motherboard
Bias voltage
High resistivity N-type silicon
P-type doping
Solder bonds
Wire bonds

(a) Timepix chip [17] (b) Structure of the Timepix chip

Fig. 1: The Timepix (a) wire-bonded on a chipboard. The detector (b) is a combination of a sensor material, which is bump-bonded to the CMOS ASIC chip.



(a) USB Lite interface [18]. (b) FITPix interface [19].

Fig. 2: USB interfaces supported by the proposed Rospix software. The USB Lite interface (a) has a very small form factor while the FITPix (b) supports multiple Timepix chips in a stack.

trometer, which measures the *counts per second*, i.e., the number of ionizing particles that interacted with the detector [16], [11], [12], [7]. Others use scintillating detectors [13], [15]. Scintillators are large and heavy, compared to other sensors, and require a large aircraft. A 94 kg UAV conducted flights around FDNPP [13] and a radiation map of the area was estimated. The UAV was flying a preprogrammed path through waypoints. The only exhibit of an algorithmic approach to locating an unknown radiation source in real-time was presented in [15].

A. Detection of ionizing radiation

Ionizing radiation is imperceivable by the human senses. However, the effects of interaction of the radiation with matter can be measured. Processing of electrical and optical signals is common since the results can be obtained in real time in contrast to measuring the chemical effects of radiation.

1) Scintillating detectors: Scintillation is an effect of visible light production (luminescence) in transparent materials by the passage of an ionizing particle. The radiation excites electrons in the material that releases light in the visible/UV line spectrum. The light is gathered and measured using a photo-multiplier. Scintillating detectors were used on UAVs [8], [9], [13], [15]. However, due to their size and weight, their use is limited to large vehicles, e.g., unmanned airplanes.

2) Semiconductor pixel detectors: Ionizing radiation interacts in many ways with materials. Depending on the energy, X-Ray photons can cause photoelectric effect, Compton scattering and electron-positron pair and triplet production. Charged particles interact directly by the Coulomb force. A piece of semiconductor material (a diode) is used directly as a sensor by convert newly produced electron-hole pairs to electric current. A single *pin-diode* is often made of Si, CdTe and GaAs and can be as large as $1\ \mathrm{cm}^3$.

A special class of a semiconductor detector is a pixel detector which is composed of a matrix of detectors. The Timepix detectors (figure 1), developed at CERN (European Organization for Nuclear Research) by the Medipix collaboration [17], [20] are low-powered pixel detectors, commonly used for medical imaging and radiography. Timepix uses a single piece of a semiconductor sensor material, which is bump-bonded to an ASIC (Application Specific Integrated Circuit) CMOS (Complementary Metal-Oxide Semiconductor) readout electronics. Each pixel is individually configured and calibrated to produce a signal only when the measured energy exceeds a pre-set threshold, which can be as low as $3\ \mathrm{keV}$. The detector does not require cooling. Each pixel of the Timepix detector is an individual dosimeter, capable of measuring in one of three modes. In *Time-over-Threshold* (ToT) mode, the pixel integrates energy deposited in the pixel during the acquisition time. *Time-of-Arrival* (ToA) mode measures the time of the first over-the-threshold event to the end of the acquisition. The *Medipix* mode counts the number of events during the acquisition. Timepix detector has a resolution of 256×256 px with each pixel being $55\ \mu\mathrm{m}$ large. Timepix has an external trigger mechanism that can control the electronic shutter to achieve arbitrary acquisition time. Acquisitions with the length of minutes are possible if the detector is calibrated correctly. On the other hand, a continuous stream of short exposures can deliver, e.g., up to 80 frames per second in the case of FITPix USB interface.

Recently, a newer version of the Timepix detector was introduced, the Timepix3 [20]. In contrast with Timepix, which reads out complete frames regardless of the recorded information, Timepix3 is an event-driven camera. Event-driven cameras output a continuous stream of data that is generated by the active (hit) pixels. A similar trend emerged in the visible-light camera field [21]. Similarly, as in the case of event-based cameras that indicated a new research stream mainly related to micro aerial vehicles, Timepix3 promises a similar impact in the field of aerial radiation detection, as its key properties are perfectly suited for dynamically flying MAVs.

The power of Pixel detectors resides in the capability to show the type and direction of the incoming radiation. Similarly to a bubble chamber, particles leave different traces in the image depending o their type, energy, and direction. The information can be obtained onboard and in real time from the recorded data by methods of computer vision and machine learning. This requires completely new approaches to radiation mapping and source of radiation detection. The ability to estimate position and matter of the source of radiation in real-time enables to control MAVs and even groups of MAVs based on the obtained information. Agile movement towards the estimated source increases the gain of the sensors and enables to exploit the potential of Timepix sensor fully.

B. Contributions

We present a novel system for acquisition flexible online radiation mapping and dynamic localization of a radiation source using Timepix radiation detectors on Robot Operating System. We present a ROS-based interface to the detectors, which allows direct integration to various robotic platforms. We provide an open source package for interfacing common Timepix electronic boards. Moreover, we include a particle track classification pipeline, for estimating the observed type of radiation in real time. The proposed system is novel in its applicability onboard mobile robotics, which could not be done so far due to the lab-focused nature of the current control software for Timepix pixel detectors. The proposed unique combination of Timepix detector and mobile robots, especially flexible MAVs, significantly improves measuring capabilities of this novel sensor and enlarges application potential on mobile robots in radiation detection scenarios. The main objective of the paper is to offer this unique and fully functional tool for real-time onboard radiation measurement to the robotics community and to motivate consequent research in the fields of homeland security and nuclear disaster mitigation. To facilitate the initial steps of the research, we provide a model of the Timepix sensor for Gazebo simulator under ROS, which enables verification of robotic algorithms.

II. LOCALIZATION OF IONIZING RADIATION SOURCES

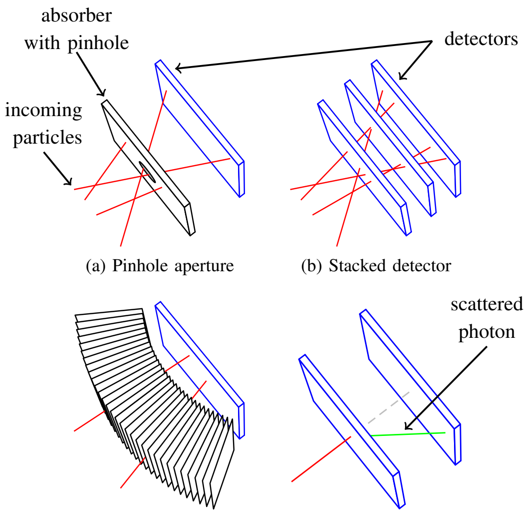
Localizing a radiation source can utilize several physical principles, that can yield information on the direction to the source. Unlike with visible light, ionizing radiation cannot be redirected by an optical lens. Heavy electrons and ions do not change their heading in matter as visible light does in glass. X-Ray photons exhibit reflective properties, which can be exploited in X-Ray reflective optics. However, the use of optics in the atmosphere is limited; the reflectance decreases with the increase in photon energy; however, only high energy photons can penetrate large portions of the atmosphere. Following paragraphs will provide an overview of different options of estimating direction to the radiation source with the use of Timepix detectors.

A. Intensity mapping and estimation

As with the event-counting detectors, Timepix can also be used to estimate the intensity of a source by measuring the particle flux. When the spectrum of the source is known or estimated on the fly, the event originating from the natural radiation background can be filtered by an image processing algorithm, e.g., the one presented in section III-A.

B. Pinhole camera aperture

As with the visible light camera, a pinhole aperture (figure 3a) is used to restrict the direction of an incoming particle to a specific point on the detector. This method works for both photons and charged particles and is especially useful for an environment with significant particle flux since the aperture shield blocks most of the incoming radiation [22]. An extension of the pinhole aperture is a coded mask. A deconvolution is then used to re-project the image using the mask's point-spread function.



(a) Pinhole aperture (b) Stacked detector

(c) Particle colimator (d) Compton camera

Fig. 3: Showcase of the different detector configurations (detectors are showed in blue color). The pinhole aperture (a) is the simplest solution to estimate the direction. The stacked detector (b) provides a feasible solution for heavy charged particles. The particle collimator (c) is feasible when sensor mass is not a constraint. Compton camera (d) uses the Compton effect to estimate the direction to a gamma radiation source.

C. Multi-detector stack

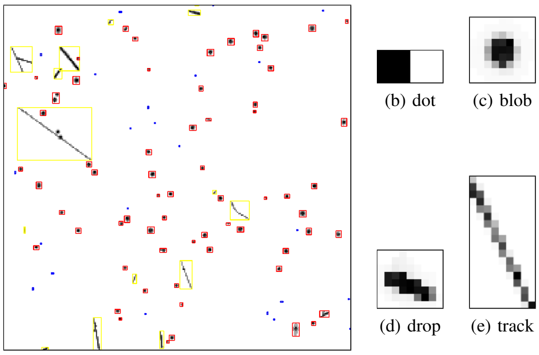
Multiple Timepix detectors are combined in multi-detector setup by stacking them on top of each other (figure 3b) [23]. Coincidences in all detectors are extracted by synchronizing the exposures and utilizing the *ToA* mode of the detector. Charged particles that intersect multiple detectors will leave tracks, which are matched and the path of the particle is reconstructed. Stacked detectors are useful for observing heavy ions and electrons that give a portion of their energy to each of the detectors while maintaining the original path.

D. X-Ray collimator

A collimator (figure 3c) performs similarly to the pinhole camera. Each place on a detector is dedicated to capturing information on particles in a particular direction. Collimators can have better angular resolution than pinhole apertures; however, their use on flying platforms is limited due to their significant mass. When the collimator has reflective surfaces in its inner tubes, an increase in gathering gain can be observed. Such collimators are called X-Ray optics, and they are often used in laboratory or space cameras [24], [25].

E. Compton effect camera

Compton camera [26] utilizes the Compton scattering effect, during which a photon is scattered in a *scattering detector* while producing a new electron. Both the scattered photon and the new electron are measured using two



(a) Image from the Timepix detector. (b) dot (c) blob (d) drop (e) track

Fig. 4: A showcase of a single image (a) from the Timepix detector with a variety of particle tracks. Tracks corresponding to photons are typically small dots (blue – (b)), ions and protons leave larger blobs and drops (red – (c),(d)) and electrons leave straight and curly tracks (yellow – (e)).

synchronized detectors (figure 3d). The angle of scattering is calculated from the measured energy signatures of the particles according to the Compton ratio formula

$$E_r\left(\theta, E_0\right) = \frac{E_s\left(\theta, E_0\right)}{E_0} = \frac{1}{1 + \frac{E_0}{m_e c^2}\left(1 - \cos\theta\right)}, \tag{1}$$

where E_s and E_0 [J] are the energies of the scattered and incoming photon, $\theta \in [-\pi, \pi]$ is the scattering angle, $m_e \approx 9.10 \cdot 10^{-31}$ kg is the invariant mass of the electron, $c \approx 2.99 \cdot 10^8$ m s^{-1} is the speed of light in vacuum. A cone of possible directions to the source is constructed using the known coordinates of the events in the two detectors.

The intensity mapping approach using a single Timepix detector was simulated in Gazebo simulator as well as deployed given the currently available detectors and interfaces, see section V. The Compton effect camera is part of the prepared simulation plugin for the Gazebo simulator. However, the experimental hardware, which requires two Timepix3 detectors and powerful onboard processing of the data-driven detectors, is still in development. The approaches relying on detection of heavy ions were not simulated, due to their low applicability due to the high particle attenuation by the atmosphere. Nevertheless, the simulation model can be extended to include those types of radiation.

III. DETECTOR READOUT AND PROCESSING PIPELINE

Reading out data from Timepix requires providing power and low-level communication to the detector. Readout interfaces are electronics boards, which allow connecting the detector to a computer via USB or Ethernet. Interfaces such as the *USB Lite* [18] and *FITPix* [19] were developed for laboratory use, however, they can be utilized on a mobile robot as well. See figure 2 for showcase of the interfaces. The software Pixelman [27], which is licensed with the interfaces, is a graphical program that does not support work on a mobile robot or an MAV. For that reason, we proposed the *Rospix*[1] software, which allows connecting *USB Lite*, *FITPix* and in the future *Katherine* interfaces to Robot Operating System.

The complete pipeline (figure 5) consists of the Timepix detector, which is mounted on or connected to a readout interface. The readout interface communicates with the onboard computer of the MAV via the Rospix [3] software. Unprocessed image frames are published to the ROS ecosystem, where they are picked up by the track classifier and acquisition time controller. Lastly, a radiation source state estimator uses the processed data and provides feedback for the MAV controller.

A. Particle track segmentation and classification

Connecting a detector interface into the ROS ecosystem is facilitated by the proposed Rospix software. Rospix configures the Timepix detectors and handles the measurement parameters such as exposure time, bias voltage, pixel equalization, and low-level analog signals. Controls of the connected detectors are provided using ROS services, and the measured data are presented via ROS topics. Rospix is designed to provide a robust connection to the hardware, even when the communication may be corrupted, which is a common theme on mobile robots. The commercial programs such as Pixelman or Pixet tend to work poorly in such conditions. The robustness was well tested during a suborbital rocket flight, where two FITPix devices were continuously measuring while connected to Odroid XU4 microcomputer [3].

Ionizing particles create characteristic tracks in the detector images. By classifying the tracks into geometric classes, we can estimate the type of radiation that caused them. Particle tracks classification can be done using the proprietary Pixelman software [27], however, only offline. Also, the solution [28] is too slow for dynamical and real-time use, and it requires a dedicated CUDA capable graphics card since it uses a neural network. Thus we propose and provide a particle track classification pipeline, which is capable of running in real time onboard a mobile robot.

The tracks are classified into the following basic classes (see figure 4), which are based on track geometry and morphology:

- **dot**: 1–2 active pixels (*photons* or *fast electrons*),
- **blob** small-to-large clusters of pixels with higher energy pixels in the center (*ions*)
- **straight track**: straight or curly lines (*electrons, high-energy ions under shallow angle*)
- **drop**: elongated blobs with deltoid (drop-like) shape (*low energy ions*, *protons*),

The shape and size of the track is closely related to the form of interaction of the ionizing particle with the matter of the detector is influenced by the particle energy, the material of the detector and the measurement parameters, such as the voltage applied to the detection diode. The proposed classifier and classes were learned on data from Low-Earth

[1] http://github.com/rospix/rospix

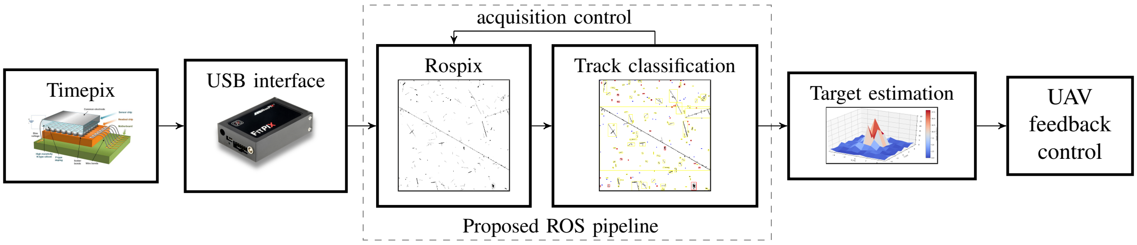


Fig. 5: The diagram of the data pipeline with the FITPix USB interface, which can be used onboard a mobile robot.

orbit [2], recorded with 300 μm Si sensor, which provides very rich family of particle tracks. Thus the full list of classes contains also minor class variants: *blob branched*, *blob small*, *blob big*, *track straight*, *track curly* and *track lowres*. Further labeling into the actual physical particle types is very challenging and often requires additional information, e.g., prior knowledge of the radiation spectrum or the angle of incidence with the detector [29], [30].

First, the image is segmented into the individual clusters (particle tracks). This process is straightforward if the amount of particles is low. The lack of any digital noise in the Timepix images implies that the tracks can be separated cleanly. However, the particle tracks can overlap during high event pileup. Rather than to solve this issue algorithmically which is challenging, we set the acquisition time in real-time to minimize this effect. Following features are extracted for the track classification:

1) Area and occupancy of a convex hull: occupancy is $o_{hull} = n_{active}/n_{hull}$, where n_{active} is the number of nonzero pixels of the track and n_{hull} is the number of pixels forming its convex hull.

2) Linearity: given the eigenvalues e_1, e_2 of the point cloud formed by the active pixels in the track, the *linearity* is defined as $linearity = max\,(e_1, e_2)\,/\,(e_1 + e_2)$.

3) Number of crossings: after skeletonization of the binarized particle cluster, the skeleton is converted to a graph, where each node represents a pixel, edges are created between 8-connected pixels. Cycles in the graph are removed, its edges are weighted (diagonal ones are penalized) and a minimum spanning tree (MST) is found. The sum of node degrees of the MST larger than 2 is the number of crossing in the trace.

4) Skeleton to hull area ratio (SKHR): The feature is calculated as $SKHR = n_{skeleton}/n_{hull}$, where $n_{skeleton}$ is the number of pixels of the cluster skeleton and n_{hull} is the number of pixels in its convex hull.

5) Tortuosity: the ratio $t = l_{curve}/l_{ends}$ is a measure of a *curliness* of a curve, where l_{curve} is the length of the curve and l_{ends} is the distance between its start and end. In our case, the longest path in the skeleton of the cluster is used to calculate tortuosity.

6) Distance transform measures: the mean and std. deviation of distances from each pixel to the edge of the cluster was used as features.

7) Boxiness: blob branched class often contains isolated clusters of pixels, which are connected with a thin line. Morphological erosion is applied to disconnect the clusters and the number of connected components in the pixel graph is the *boxines* feature.

8) Foreground connectivity with background: three features, which represent the number of pixels that are facing the background with 1, 2 and three edges.

9) Diagonality and straightness: an image I_1 is created from the original image I_0 as $I_1 = (I_0 - (I_0 \circ s_1)) \circ s_2$. s_1 is structuring element for morphological opening, s_2 is structuring element with two horizontally and two diagonally placed pixels for *straightness* and *diagonality* respectively. s_2 is applied two times with different orientation to obtain features in arbitrary direction. The features are calculated by summing the pixels in the resulting images for both operations.

10) Basic features: area: number of nonzero pixels in the cluster, *energy*: sum of values of pixels in the cluster, *energy per pixel*: ratio of area and energy, *energetic quartiles*: lower decile, upper decile and median of energies of pixels in the cluster, *width and height* of the rectangular hull of the cluster.

Support Vector Machines (SVM), *logistic regression* and *random forest* classifiers were trained on human-labeled data using the previously defined features. Implementation of the classifiers relied on the Python scikit-learn library. After hyperparameter search (see table I) and evaluation, the random forest classifier was chosen after having the best Mathews Correlation Coefficient (MCC) on testing data set. The random forest performed with test $MCC = 0.7905$. It was followed by the support vector machine with $MCC = 0.7689$ and the logistic regression with $MCC = 0.7438$. The confusion matrix in figure 6 shows that the classifier performance is very good for particles of types *Dot* (0), *Track small* (1) and *Blob big* (2). The classifier often confuses classes *Blob branched* (3) and *Track curly* (5). However, *Blob branched* is not very common in terrestrial measurements, since it is caused by very high energy ions such as cosmic rays. The branching is caused by a complex interaction of the ion with the sensor while producing additional particles – delta electrons. The classes *Drop* and *Track lowres* are problematic since the training data set did not contain enough data. The particle classification pipeline is available in ROS as a Python scikit implementation[2].

[2] http://github.com/rospix/classification

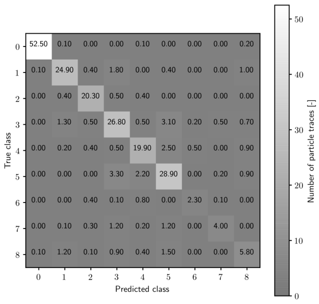
Part of the pipeline	Variants	Parameters
scaling	standardization to $\mathcal{N}(0,1)$	-
multiclass extension	not needed	-
hyperparameters	number of estimators	240
	maximum tree depth	∞
	min. samples per split	5
	min. samples per leaf	1

TABLE I: Optimized parameters of random forest.

IV. Detector Simulation Model

Together with the Rospix driver, we provide a simulation package for the Gazebo-ROS simulator to facilitate research of MAV control and planning algorithms with this new sensory system in feedback. Ray tracing method was developed to simulate gamma rays and their interaction with simulated Pixel detectors. Our focus is in localizing weak radiation sources, located in greater distance (>50 m). It has an important implication on the expected incoming radiation since most of any potential ions and beta radiation would be blocked by the atmosphere. High energy gamma photons (>500 keV) still have a high chance to penetrate even hundreds of meters of air. Such radiation can originate from materials that are commonly used in proton therapy: ^{60}Co, ^{137}Cs and ^{90}Sr, which can be possible lost and misused. Search for those sources is the main interest of homeland security departments and therefore motivates our research the most. We aim to develop the Compton camera specifically for MAVs. Photo-electric effect and the Compton scattering are the key effects in the Compton camera. Following sections present the simulation model used to model these effects in the detectors.

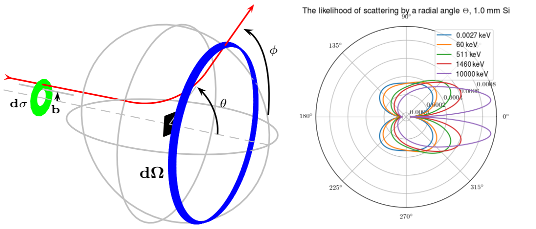
1) Differential cross section: The properties of non-elastic scattering from scattering center and particle collisions are described by a differential cross section. A total cross section characterizes an effective area of an event (collision, scattering). Let us have a particle on an incident trajectory with the scattering object. The impact parameter b is the displacement of the particle from the path to the scattering center; the radial angle of scattering is denoted by θ. The total area of the impact parameter is the impact cross section σ, which is obtained by integrating the impact parameter b over all possible azimuthal angles ϕ. In a case where the scattering is not a function of ϕ (axially symmetrical case), the impact cross section takes the form



(a) confusion matrix

Class	
0. Dot	705
1. Blob small	383
2. Blob big	288
3. Blob branched	448
4. Track straight	332
5. Track curly	473
6. Drop	49
7. Track lowres	134
8. Other	93

Fig. 6: Confusion matrix of the random forest classifier, generated with the use of 10-fold cross validation from testing part (25%) of the dataset.



(a) Differential cross section.

(b) Compton scattering likelihood.

Fig. 7: (a) Showcase of solid angle $d\Omega$ and the differential size of the impact plane $d\sigma$ for $\theta = 60$ deg and (b) the plots of likelihood $\mathrm{P}(\theta \mid E_0)$, integrated over azimuthal angle ϕ for various energies.

$$\sigma(b) = \int_{\Phi} b \, d\phi = \frac{b^2}{2} 2\pi = \pi b^2. \tag{2}$$

Such relaxation is viable for Compton scattering, since the scattering bodies (electrons) are spherically symmetrical objects. The differential of the impact cross section is

$$d\sigma(b) = \frac{\partial \sigma}{\partial b} db = 2\pi \, b \, db. \tag{3}$$

The solid angle on a unit sphere under the angle $\theta < \Theta$ is obtained by the integration:

$$\Omega(r, \Theta) = \int_0^{\Theta} 2\pi r \, \cos\theta \, r \, d\theta = 2\pi r^2 - 2\pi r^2 \cos\Theta. \tag{4}$$

The differential of the solid angle is

$$d\Omega(\theta) = -4\pi r \, \cos\theta \, dr + 4\pi r \, dr + 2\pi r^2 \sin\theta \, d\theta. \tag{5}$$

In the case of a unit sphere, the differential of the area is simplified to

$$d\Omega(\theta) = 2\pi r^2 \sin\theta \, d\theta. \tag{6}$$

The total cross section σ is obtained by integrating the differential cross section over the area of a unit sphere:

$$\sigma = \oint_{4\pi} \frac{d\sigma}{d\Omega} d\Omega = \int_0^{2\pi} \int_0^{\pi} \frac{d\sigma}{d\Omega} \sin\theta \, d\theta \, d\phi. \tag{7}$$

The decrease of the intensity $d\Phi$ of an incident beam with original flux $\Phi \left[\mathrm{s}^{-1}\right]$ is described as

$$\frac{d\Phi}{dz} = -n\sigma\Phi, \tag{8}$$

where dz [m] is the thickness of the material, $n \left[\mathrm{m}^{-3}\right]$ is the particle density of the material and $\sigma \left[\mathrm{m}^2\right]$ is the total cross section of the interaction. By solving the differential equation, we obtain a relationship between the initial flux Φ and the remaining flux Φ_{out} behind the object with the thickness z:

$$\Phi_{out} = \Phi e^{-n\sigma z}. \tag{9}$$

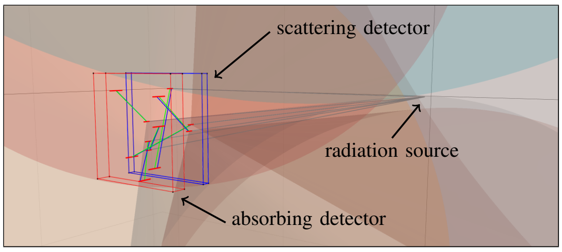


Fig. 8: Simulation of Compton camera which consists of two Timepix detectors. Possible directions to the radiation source are showed by semi-transparent cones originating from the pixels in the scattering detector.

The probability of an *event* (E) is modeled as

$$\mathrm{P}\left(E\right) = 1 - e^{-n\sigma z}. \tag{10}$$

2) Photoelectric effect: Photoelectric effect describes a total absorption of a photon by an electron. A portion of the energy is responsible for releasing the electron from the atomic orbital; the rest is converted to kinetic energy of the electron. Photon energy can be expressed using its wavelength λ [m] or frequency ν [Hz] as

$$E_\gamma = \frac{hc}{\lambda} = h\nu, \tag{11}$$

where $h \approx 6.62 \cdot 10^{-34}\ \mathrm{m^2\,kg\,s^{-1}}$ is the Planck constant and $c \approx 2.99 \cdot 10^8\ \mathrm{m\,s^{-1}}$ is the speed of light in a vacuum. $k = E_\gamma / E_e$ is the ratio between the photon energy $E_\gamma = h\nu\,[\mathrm{eV}]$ and the electron rest mass energy $E_e = m_e c^2 \approx 5.11 \cdot 10^5$ eV. According to [31], the simplified Gavrila-Pratt [32] cross section for the photoelectric effect is

$$\sigma_{ph} = \frac{16}{3}\sqrt{2}\pi r_e^2 \alpha^4 \frac{Z^5}{k^{3.5}}, \tag{12}$$

where $r_e \approx 2.81 \cdot 10^{-15}$ m is the classical electron radius, $\alpha \approx 1/137.04$ is the fine structure constant and Z is the atomic number of the element. The accuracy of (12) is relatively low even in the energy range, where it should be valid ($\approx$ 1 to 1000 keV), since (12) expresses the cross section for a free electron, not an electron bound in an orbital. Attenuation coefficients from NASA NIST[3] can be interpolated and resampled to achieve better accuracy.

3) Compton scattering: Compton scattering occurs when a photon transfers a portion of its energy to an electron. During this interaction, the photon is deflected from its original path by the radial angle θ and azimuthal angle ϕ. The Klein-Nishina formula [33] describes the differential cross section $d\sigma/d\Omega$ $[\mathrm{m^2/sr}]$ for the incident and scattered beam:

$$\frac{d\sigma}{d\Omega} = \frac{1}{2} r_e^2 E_r^2 \left(E_r + \frac{1}{E_r} - \sin^2\theta\right), \tag{13}$$

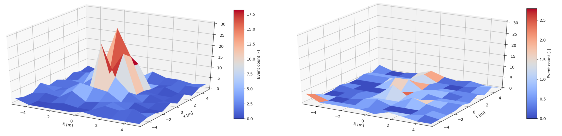
where E_r is the Compton ratio (1), $r_e \approx 2.81 \cdot 10^{-15}$ m is the classical electron radius. The prior probability of the scattering in a material with thickness z is computed as:

$$\mathrm{P}\left(E_{cs}\right) = 1 - e^{-n\sigma_{cs} z}. \tag{14}$$

However, as well as with the photoelectric effect, the accuracy of the prior is low, since the Klein-Nishina formula describes the cross section for a free electron. Empirical-based probability distributions should be used to obtain more precise results. The value of likelihood probability density for the event E_{cs} of a single photon with initial energy E_o [eV] being scattered by the angle θ is calculated as

$$\mathrm{P}\left(\theta \mid E_{cs}\right) = \frac{\int_0^{2\pi} \frac{d\sigma}{d\Omega} \sin\theta \, d\phi}{\sigma_{cs}}, \tag{15}$$

where σ_{cs} is the total cross section for Compton scattering obtained from (13). Figure 7 depicts an illustration of $d\sigma/d\Omega$ and the likelihood of scattering by the angle θ calculated for 1 mm of silicon for various energies.



(a) 2-D map of X-Ray photon event count measured with the MAV.

(b) 2-D map of natural background radiation event count.

Fig. 9: Maps of the radiation intensity during a flight with the Timepix detector (300 μm Si) above a ^{241}Am radiation source ($\approx$ 500 MBq). Photon events (a) can be distinguished from the background radiation (b) by analyzing the particle tracks in the gathered images.

V. EXPERIMENTS

Proof of concept experiments that showcase the Timepix detector on an MAV was conducted with ^{241}Am source with activity $\approx$ 500 MBq. This radiation source produces gamma radiation with energy 59.5 keV. A small MAV was equipped with FITPix interface and Timepix detector with 300 μm Si sensor. The MAV was built upon the DJI F450 frame and equipped with Intel NUC i7 computer, and the FITPix was connected via USB. Robot Operating System control pipeline [34] was used to guide the MAV along a pre-planned trajectory while being localized by GPS and optic flow. Figure 9 shows the resulting radiation maps for photon events and for the radiation background, which was separated using the classifier. Data gathered during the experiment and the classification pipeline can be accessed at `http://mrs.felk.cvut.cz/iros2019timepix`.

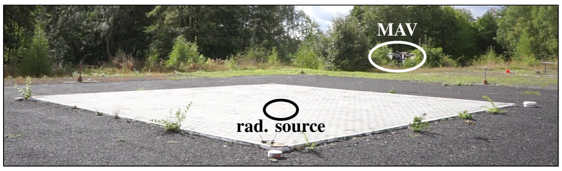


Fig. 10: Photo of the autonomous MAV while mapping the radiation environment using the Timepix detector.

[3] `http://physics.nist.gov/PhysRefData/FFast/html/form.html`

VI. Conclusions and Future Work

We present a system based on Timepix detectors for real-time localization of radiation sources by a mobile robot. The proposed solution consists of the Rospix software interface for Robot Operating System. The solution includes a fast particle track classifier, which is a feature that was up to now part of proprietary laboratory software only. Mobile robots can use our system to perceive and interpret the measured data in real time including estimating the position of a source and the type of incoming radiation. We demonstrate the system in an experiment with a Micro Aerial Vehicle and $^{241}\mathrm{Am}$ radiation source. The Timepix sensor and the Compton effect camera assembly is provided in the form of a plugin for Gazebo-ROS simulator. Thanks to the onboard radiation data processing, the proposed pipeline will enable automatic control of MAVs with the radiation sensor in control feedback. This will allow fully autonomous localization of radiation sources thanks to the high mobility of the MAVs. Currently, our group focuses on the Compton camera principle for localizing weak gamma-ray sources using Timepix3 detectors. With this physical principle the concept should be sufficient for localization of faint compact gamma sources (tens of GBq) at a distance of approx $100\ \mathrm{m}$.

Acknowledgment

The presented work was done on behalf of Medipix2 collaboration and was supported by Czech Grant Agency project No. 17-16900Y CTU, grant SGS17/187/13 and OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics".

References

[1] G. Pajares, "Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–329, 2015.

[2] T. Baca *et al.*, "Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements," *Journal of Instrumentation*, vol. 13, no. 11, p. C11010, 2018.

[3] T. Baca, D. Turecek *et al.*, "Rospix: modular software tool for automated data acquisitions of Timepix detectors on Robot Operating System," *Journal of Instrumentation*, vol. 13, no. 11, p. C11008, 2018.

[4] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.

[5] M. T. Perks, A. J. Russell, and A. R. Large, "Advances in flash flood monitoring using unmanned aerial vehicles (UAVs)," *Hydrology and Earth System Sciences*, vol. 20, no. 10, pp. 4005–4015, 2016.

[6] K. Kurvinen, P. Smolander, R. Pöllänen *et al.*, "Design of a radiation surveillance unit for an unmanned aerial vehicle," *Journal of environmental radioactivity*, vol. 81, no. 1, pp. 1–10, 2005.

[7] P. Martin, O. Payton, J. Fardoulis *et al.*, "The use of unmanned aerial systems for the mapping of legacy uranium mines," *Journal of environmental radioactivity*, vol. 143, pp. 135–140, 2015.

[8] R. Pöllänen, H. Toivonen, K. Peräjärvi *et al.*, "Radiation surveillance using an unmanned aerial vehicle," *Applied radiation and isotopes*, vol. 67, no. 2, pp. 340–344, 2009.

[9] R. Pillanen *et al.*, "Performance of an air sampler and a gamma-ray detector in a small unmanned aerial vehicle," *Journal of radioanalytical and nuclear chemistry*, vol. 282, no. 2, p. 433, 2009.

[10] J. Aleotti *et al.*, "Unmanned aerial vehicle equipped with spectroscopic CdZnTe detector for detection and identification of radiological and nuclear material," in *NSS/MIC*. IEEE, 2015, pp. 1–5.

[11] J. MacFarlane, O. Payton *et al.*, "Lightweight aerial vehicles for monitoring, assessment and mapping of radiation anomalies," *Journal of environmental radioactivity*, vol. 136, pp. 127–130, 2014.

[12] P. G. Martin, S. Kwong, N. Smith *et al.*, "3D unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 12–19, 2016.

[13] Y. Sanada *et al.*, "Aerial radiation monitoring around the Fukushima Dai-ichi nuclear power plant using an unmanned helicopter," *Journal of environmental radioactivity*, vol. 139, pp. 294–299, 2015.

[14] X.-B. Tang, J. Meng, P. Wang, Y. Cao *et al.*, "Efficiency calibration and minimum detectable activity concentration of a real-time UAV airborne sensor system with two gamma spectrometers," *Applied Radiation and Isotopes*, vol. 110, pp. 100–108, 2016.

[15] J. Towler, B. Krawiec, and K. Kochersberger, "Radiation mapping in post-disaster environments using an autonomous helicopter," *Remote Sensing*, vol. 4, no. 7, pp. 1995–2015, 2012.

[16] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field," *Journal of Intelligent & Robotic Systems*, pp. 1–10, 2013.

[17] X. Llopart, R. Ballabriga, M. Campbell *et al.*, "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 581, no. 1-2, pp. 485–494, 2007.

[18] Z. Vykydal and J. Jakubek, "USB LiteMiniaturized readout interface for Medipix2 detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 633, pp. S48–S49, 2011.

[19] V. Kraus, M. Holik *et al.*, "FITPixfast interface for Timepix pixel detectors," *Journal of Instrumentation*, vol. 6, no. 01, p. C01079, 2011.

[20] T. Poikela, J. Plosila *et al.*, "Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout," *Journal of instrumentation*, vol. 9, no. 05, p. C05013, 2014.

[21] B. Kueng *et al.*, "Low-latency visual odometry using event-based feature tracks," in *IEEE/RSJ IROS*. IEEE, 2016.

[22] M. Gmar, M. Agelou, F. Carrel *et al.*, "GAMPIX: A new generation of gamma camera," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 652, no. 1, pp. 638–640, 2011.

[23] P. Soukup, J. Jakubek, and Z. Vykydal, "3D sensitive voxel detector of ionizing radiation based on Timepix device," *Journal of Instrumentation*, vol. 6, no. 01, p. C01060, 2011.

[24] T. Baca, M. Platkevic, J. Jakubek *et al.*, "Miniaturized X-ray telescope for VZLUSAT-1 nanosatellite with Timepix detector," *Journal of Instrumentation*, vol. 11, no. 10, p. C10007, 2016.

[25] V. Stehlikova, M. Urban *et al.*, "Study of Lobster eye optics with iridium coated X-ray mirrors for a rocket experiment," in *EUV and X-ray Optics: Synergy between Laboratory and Space V*, vol. 10235. International Society for Optics and Photonics, 2017, p. 1023505.

[26] D. Turecek *et al.*, "Compton camera based on Timepix3 technology," *Journal of Instrumentation*, vol. 13, no. 11, p. C11022, 2018.

[27] D. Turecek, T. Holy *et al.*, "Pixelman: a multi-platform data acquisition and processing software package for Medipix2, Timepix and Medipix3 detectors," *Journal of Instrumentation*, vol. 6, no. 01, p. C01046, 2011.

[28] W. Furnell *et al.*, "First results from the LUCID-Timepix spacecraft payload onboard the TechDemoSat-1 satellite in Low Earth Orbit," *Advances in Space Research*, vol. 63, no. 5, pp. 1523–1540, 2019.

[29] T. Holy, E. Heijne, J. Jakubek *et al.*, "Pattern recognition of tracks induced by individual quanta of ionizing radiation in Medipix2 silicon detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 591, no. 1, pp. 287–290, 2008.

[30] S. M. Hoang, "A pattern recognition approach to learning tracks of heavy-ion particles in timepix detectors," Ph.D. dissertation, 2013.

[31] K. W. Fornalski, "Simple empirical correction functions to cross sections of the photoelectric effect, Compton scattering, pair and triplet production for carbon radiation shields for intermediate and high photon energies," *Journal of Physics Communications*, vol. 2, no. 3, p. 035038, 2018.

[32] D. C., "Interaction of gamma-radiation with matter," *Alpha-, Beta-, Gamma-ray Spectroscopy*, vol. 1, no. 1, 1980.

[33] W. R. Leo, *Techniques for nuclear and particle physics experiments: a how-to approach*. Springer Science & Business Media, 2012.

[34] T. Baca *et al.*, "Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles," in *IEEE/RSJ IROS*. IEEE, 2018.

IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED FEBRUARY, 2020

Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles with Pixel Detectors in Real-Time

Petr Štibinger[1], Tomáš Báča and Martin Saska

***Abstract*—We provide a complex software package allowing the user to deploy multiple ionizing radiation sources and detectors modeled after the Timepix miniature pixel detector. The software is provided to the community as open-source, and allows preliminary testing and method development even without a pixel detector or radiation sources. Our simulation model utilizes ray-tracing and Monte Carlo methods to resolve interactions of ionizing radiation with the detector, obstacles and the atmosphere. An open-source implementation is provided as a plugin for Gazebo, a simulator popular within the robotics community. The plugin is capable of simulating radiation sources with activities in the order of GBq[2] in real-time with a conventional PC. We also provide a ROS interface, which allows full integration of the Timepix pixel detector into a robotic system. The credibility and the precision of the simulator plugin were confirmed via a real-world experiment with a micro aerial vehicle (MAV) equipped with a Timepix detector mapping the radiation intensity of an Am-241 sample. Finally, we present a method for cooperative localization of a source of ionizing radiation by a group of autonomous MAVs in an environment with obstacles.**

Aerial Systems: Applications, Environment Monitoring and Management, Computational Geometry

I. Introduction

WITH recent advances in mobile robotics, exploration of hazardous environments no longer requires direct human presence. An unmanned vehicle equipped with radiation detectors can safely perform measurements for longer periods of time and closer to the radiation source than a human worker. This provides a way to collect more data and to localize sources of radiation more precisely. Unmanned vehicles can assist in surveying sites of nuclear accidents, uranium ore mines or spent fuel storage facilities. Another major research topic related to this paper is a rapid response to the public safety threat, that can occur as a result of mishandled radiotherapy equipment or an imminent act of terrorism. Currently, a heavy pickup truck equipped with powerful radiation sensors and protective shielding for the crew has to be dispatched to the threatened area.

Several research projects have been dealing with localization of ionizing radiation sources by unmanned robotic vehicles. Most of these projects consider a ground robot, which will be controlled remotely by a human operator, or an aerial robot following a predefined trajectory. In some scenarios, however, remote control is rendered impossible due to strong wireless signal absorption or electromagnetic interference. Moreover, in an environment with an unknown obstacle layout, a trajectory cannot be prepared in advance. For these reasons, the community has recently been focusing on developing autonomous systems, that utilize additional onboard sensors and decision-making algorithms to complete the task without external guidance in real-time.

Developing of autonomous systems is a complex process, and a proper simulation environment is often used in all stages of development. Highly realistic simulators of ionizing radiation and its interactions with matter, such as *GEANT4* [1], *CASINO* [2] or *FLUKA* [3], have been developed for the purposes of nuclear science and medicine. However, it may be far too difficult to interface these programs with the simulation software commonly used in robotics, such as *V-REP* [4], *Gazebo* [5], *Actin* [6] or *Webots* [7], and to perform the simulation in real-time. Simulated radiation using custom-built software is presented in [8], [9], while most projects are evaluated through real experiments instead [10]–[15].

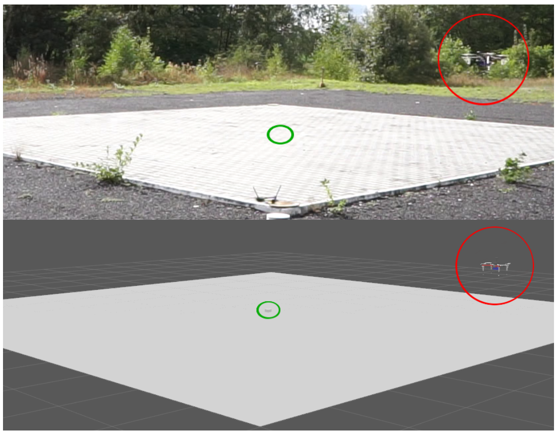


Fig. 1: An experiment with a MAV (red) equipped with a Timepix detector creating a map of radiation intensity. The radioactive source (green) was a sample of ^{241}Am with intensity of 0.5 GBq. The same setup as during the experiment was simulated in Gazebo using the novel radiation plugin.

Manuscript received: September, 10, 2019; Revised January, 3, 2020; Accepted February, 9, 2020.
This paper was recommended for publication by Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments.
This work was supported by CTU grant no. SGS17/187/OHK3/3T/13 and the OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics". The work of Martin Saska was supported by the CSF grant no. 17-16900Y.

The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
[1] stibipet@fel.cvut.cz

Digital Object Identifier (DOI): see top of this page.

[2] Bq (Becquerel) = 1 particle emission per second

With this paper, we aim to present our radiation simulator to the community as open-source[3], and to motivate further development in autonomous localization of radioactive sources for nuclear and homeland security applications. For the community, we offer a complete software package, which enables seamless transition from a realistic simulator into a real deployment of robots equipped with Timepix sensors. Special interest is taken in the use of Timepix in a direct feedback for autonomous methods for localizing radiation sources by micro aerial vehicles (MAVs). Semiconductor pixel detectors, such as Timepix, are well suited for a setup with small and agile MAVs, due to their low weight, low power consumption and fast readout rate.

II. RELATED WORK AND PRELIMINARIES

Several contributions have been made in the area of radiation mapping and source localization over the past ten years. Since the Fukushima Daiichi nuclear power plant accident in 2011, the site has been under surveillance by multiple robotic missions. Unmanned aerial vehicles (UAV) have been used to monitor radiation levels from the air while following a predefined trajectory in [16]–[20]. Unmanned ground vehicles (UGV) have been used under remote control in [13], [14].

The system presented in [21] employs multiple UAVs for contour analysis of an irradiated area. The UAVs move along a predefined series of waypoints, and measurements are transmitted to a ground station for processing. In [9], a contour analysis is presented for localization of multiple sources with overlapping radiation fields. A simulation in [8] demonstrates a control algorithm for multiple robots to cooperatively explore an area containing multiple radiation sources. In [22], a 90 kg UAV is shown actively searching for unknown sources of radiation while making real-time adjustments to its trajectory based on onboard measurements by a NaI scintillator. To the best of our knowledge, this is so far the only project to employ an autonomous active search for a radiation source with an aerial platform.

Related projects utilize heavy robotic platforms, either ground vehicles or multirotor helicopters with a payload capacity of more than 1 kg, capable of carrying radiation detectors which are bulky or require a cooling mechanism. Non-use of radiation detectors as sensors in a feedback loop for active source detection and precise localization may be caused by unavailability of devices with a real-time response and the possibility of direct onboard processing.

A. RADIATION DETECTION

Before mobile robots became available, handheld detectors had to be used, since human bodies lack the ability to sense ionizing radiation on their own. This chapter summarizes the commonly used types of radiation sensors.

Ionization chambers consist of a gas-filled tube with an electric field introduced by a voltage source. Radiation passing through the tube causes the gas to ionize, and creates measurable disruptions in the electric field. The main advantage of these detectors is their low cost. However, small sized sensors of this type often lack the sensitivity necessary for detecting comparatively weak sources, due to low density of the atoms in the gases. Use of ionization chamber detectors in robotics has been shown in [23], [24].

Scintillation detectors utilize the emission of visible or UV light, generated by radiation passing through certain materials. Since the wavelength of the emitted light depends on the radiation energy, these detectors can also determine the spectral properties of the radiation. Mobile robots equipped with scintillators have been presented in [10], [25]. Due to indirect conversion of radiation energy to an electrical signal, and the need to employ additional electronics for visible light processing, these detectors are often heavier than other types of detectors. A miniature detector *Kromek* using a 1 cm^3 CdZnTe scintillator weighing 0.5 kg was used onboard UAVs in [16]–[18].

Semiconductor detectors rely on the presence of charge carriers in the form of electrons and holes in their atomic lattice. Similarly to ionization chambers, radiation passing through a semiconductor can cause the generation of new charge carriers. Using a P-N junction, this event can be directly transformed into an electrical signal. The main advantage of semiconductor detectors is low ionization energy – radiation energy necessary for creating new charge carriers. This property makes semiconductor detectors very sensitive, even capable of detecting individual particles. A small lightweight semiconductor detector is used in [13], mounted to a remotely controlled ground vehicle. The most recent trend is to stack two semiconductor pixel detectors to form a Compton camera [26], which utilizes the kinematics of Compton scattering to provide additional information about direction of incoming radiation. Compton cameras mounted onto mobile robots have been used to create 3D radiation maps in [12], [14], [15]. The additional spatial information provided by the Compton camera offers great volumetric mapping potential. However no real-time onboard processing of the output data has been demonstrated, as the output data complexity and volume is much higher than with other detectors.

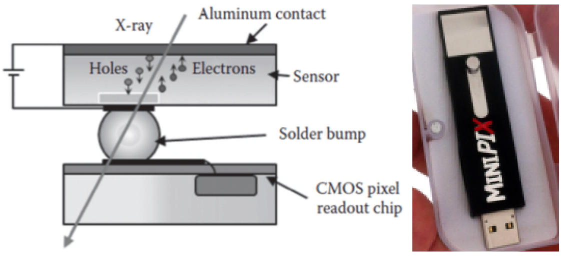
Timepix belongs to a group of semiconductor detectors called pixel detectors. It consists of a matrix of 256×256 pixels, which are tiled onto a single block of semiconductor material, such as Si, GeAs or CdTe. The inner structure of one pixel is shown in Fig. 2a. The area of the chip is only 14.08×14.08 mm and the thickness can be selected from 100 µm to 1 mm. In the smallest commercially available configuration, called MiniPIX [27] (Fig. 2b), the device only weighs 50 g and it is comparable in size with a USB flash drive. The device is connected to a computer through a USB 2.0 interface, which serves both as a data channel and as a power source. Compared to other types of detectors, Timepix is extremely lightweight, compact and capable of a high data readout rate (up to 100 frames per second). It is also incredibly sensitive, even capable of counting individual high-energy photons. Although these properties offer great application potential in mobile robotics, especially for robots with limited payload capacity, such as small-sized MAVs, Timepix has never been used onboard aerial robots, to the best of our knowledge.

[3] Source code repository: https://github.com/rospix
Multimedia: http://mrs.felk.cvut.cz/radiation2020ral

B. *SIMULATED RADIOACTIVITY*

Most robotic systems designed for operation in radioactive environment are tested via real world experiments. A radiation model, implemented in MATLAB, is presented in [8]. It is based on the decrease of radiation intensity with a square of distance. The model represents a coarse approximation, and does not consider environment losses and obstacles. The simulator presented in [9] uses a Gaussian mixture model (GMM) to approximate the radiation field from multiple sources. The intensity map is constructed by overlaying individual Gaussian distributions centered around point sources. This approach allows for simulation of a more widespread pollution, and the variance of the Gaussian curve can be used to model different strengths of radiation penetration in the atmosphere. Obstacles are not considered in [9].

In this paper, we focus solely on gamma radiation, as alpha and beta particles are usually absorbed by the atmosphere after only a few decimeters.



(a) Inner pixel structure [28] (b) MiniPIX [27]

Fig. 2: The Timepix chip consists of 256 × 256 individual pixels (**a**), which all act as miniature radiation detectors. A very compact Timepix-based radiation camera is commercially available as MiniPIX (**b**). The device is comparable in size with a USB flash drive. The metallic Timepix chip is visible in the upper section of the sensor.

III. CONTRIBUTION

1) We present a novel method for realistic simulation of X-ray/gamma sources. The model utilizes the principles of high energy photon physics to achieve realistic results. Ray-tracing algorithms are used to simulate the ionizing radiation on the level of individual photons. Given the substantial amount of particles emitted by real radioactive materials, we also present methods for reduction of necessary calculations, which significantly speed up the process. We provide our implementation of the model as an open-source plugin for Gazebo. The realistic properties of the simulation were experimentally evaluated by a comparison with real-world measurements taken by a micro aerial platform shown in Fig. 1. We also provide a ROS[1] interface for the Timepix detector, which allows a seamless integration of the sensor into existing robotic platforms.

2) A new approach to the use of multiple MAVs for fast detection and localization of ionizing radiation sources was designed to demonstrate the abilities of the system. This first attempt at using cooperating MAVs for active real-time localization of radiation sources has clearly shown the advantage of such a lightweight system in comparison with conventional radiation detection methods.

This system aspires to be an enabling technique for using micro-sized aerial vehicles in tasks of radiation monitoring and localization of radiation sources. Exploiting the mobility and maneuverability of MAVs together with the unique properties of the Timepix pixel detector, which has not been used with MAVs so far, opens new perspectives and application domains in the fields of nuclear safety and civil protection.

IV. SIMULATION PIPELINE

In this section, we outline the core principles and algorithms used for the simulation of ionizing radiation. The mathematical model is derived from well known interactions between matter and high energy photons [30].

A. *MATHEMATICAL MODEL*

The direction of emission cannot be predicted in advance for any photon released by a real radiation source. On a large scale, however, the spatial distribution of photon directions is considered uniform. The output of a radiation source is defined by *activity*, which is calculated as follows:

$$A = \frac{m}{m_a} N_A \frac{\ln(2)}{t_{1/2}} , \tag{1}$$

where m is mass of the source in grams, m_a is mass of one atom of the isotope in grams, N_A is the Avogadro constant, and $t_{1/2}$ is the half-life of the isotope in seconds. Activity defines the *average* number of decay events (in our case photon emissions) per second. For our model, the activity of a source serves as a constant frequency.

Two assumptions were made. Firstly, the radioactive material is considered to be an omnidirectional point source. Secondly, the velocity of emitted photons is assumed to be infinite. The intended world scale of use for the model is under a hundred meters in diameter, and the highest readout frequency of the sensor is 100 Hz. Therefore, the time of flight can be safely omitted, and each photon is considered to be detectable at the time of its emission.

In a lossless environment, the total photon output of a radiation source is equal to the sum of photons intersecting the surface of a sphere centered around the source. Let us denote the solid angle on a sphere Ω, and the probability density function $f(\Omega)$. The surface area S of a unit sphere is equal to 4π, therefore a uniform distribution of points on its surface is given as:

$$f(\Omega) = \begin{cases} \frac{1}{4\pi}, & \text{for } 0 \leq \Omega \leq 4\pi, \\ 0, & \text{otherwise ,} \end{cases} \qquad \iint_S f(\Omega)\, d\Omega \quad = 1 \, . \tag{2}$$

Let us now denote the probability $P(\Omega)$, that a point lies on a part of spherical surface $S_\Omega < S$ corresponding to the solid angle Ω:

$$P(\Omega) = \iint_{S_\Omega} f(\Omega)\, d\Omega = \frac{\Omega}{4\pi} \, . \tag{3}$$

We define an *apparent activity* A_{app} as the total amount of particles hitting the specific portion of the spherical surface:

$$A_{app}(\Omega) = AP(\Omega) = A\frac{\Omega}{4\pi} \, . \tag{4}$$

[1] Robot Operating System [29]

This allows us to compute the amount of detectable photons, by projecting individual sides of the detector on the surface of the unit sphere. Let us consider a "triangle" drawn onto the surface of a sphere. It consists of three vertices, which are connected by the shortest possible paths on the surface, the *orthodromes*. The solid angle for the surface of this triangle is equal to a spherical excess *E*, which is calculated as follows:

$$E = A + B + C - \pi, \tag{5}$$

where A, B, C are the angles between the orthodromes. The angles can be obtained from Equation 6, which is derived from the *law of haversines*:

$$C = \operatorname{archav}\left(\frac{\operatorname{hav}(c) - \operatorname{hav}(a-b)}{\sin(a)\sin(b)}\right). \tag{6}$$

In this equation, central angles a, b, c for the corresponding orthodromes are used. The trigonometric function *haversine* used in this equation is defined as:

$$\operatorname{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2}\ . \tag{7}$$

The central angle θ between any two points on a spherical surface is calculated using the known coordinates of the spherical center (world position of the radiation source) and the world coordinates of the two points. The two points are projections of the detector vertices onto the sphere. However, the central angle remains the same, if we use the coordinates of the vertices directly, as the angular calculations are invariant to the radius of the sphere. Therefore, the central angle θ for coordinate vectors $\vec{u}, \vec{v}$ is calculated as:

$$\theta = \arccos\left(\frac{\vec{u}\cdot\vec{v}}{|\vec{u}|\cdot|\vec{v}|}\right)\ . \tag{8}$$

By far the most common type of interaction of a photon with other matter (detector, obstacles, air) is absorption. Photon absorption may be caused by one of three processes: the photoelectric effect, the Compton scattering or the pair production. For a single photon, it is nearly impossible to predict the exact outcome of the interaction. On a large scale, however, absorption percentages can be measured.

In the proposed model, the detection of a photon only occurs as a result of the photoelectric effect, as this is the predominant interaction for photons at the energy levels produced by natural radioactive decay. According to [30], the photoelectric absorption percentage is calculated as:

$$P_e = 1 - e^{-(\mu_{en}/\rho)\rho l}\ , \tag{9}$$

where ρ is the density of the absorber material and l is the length of a photon track through the absorber. The mass-energy attenuation coefficient μ_{en}/ρ depends on the energy of the photon, and the corresponding value is obtained by a cubic spline interpolation of the data provided by the NIST[2] Standard Reference Database 126 [30]. The absorption in obstacles is analogous, only the absorption coefficient is replaced by the total attenuation coefficient μ/ρ, which encompasses all three types of absorption.

Algorithm 1 Radiation raytracer

```
$PhotonCounter \leftarrow 0$
$N \leftarrow A_{app} \cdot T_{exp} \cdot L_{env}$ # ray count to be simulated

# simulate the rays hitting one SensorSide
procedure SimulationStep($SensorSide$, $N$)
    for $N$ do
        $p1 \leftarrow SensorSide.\mathrm{RandomPoint}$
        $r \leftarrow \mathrm{Raycast}(RadSource, p1)$
        $p2 \leftarrow \mathrm{Intersect}(r, OtherSides)$
        $track \leftarrow \mathrm{Distance}(p2, p1)$
        $P_e \leftarrow \mathrm{PhotoelAbsorptionProb}$          ▷ Equation (9)
        if random(0,1) $< P_e$ then                        ▷ hit detected
            $PhotonCounter \leftarrow PhotonCounter + 1$
        end if
    end for
    $TimepixMessage \leftarrow PhotonCounter, T_{exp}$
    ROS.Publish($TimepixMessage$)
end procedure
```

B. IMPLEMENTATION DETAILS

The implementation utilizes a common rendering technique, where rays are cast from the camera towards the scene, rather than the other way around. In our model, the rays are cast from the detector towards the radiation source. This significantly reduces the computational complexity, as only detectable photons are simulated. Since the detector has a cuboid shape, up to three sides can be exposed to the radiation source every simulation step. As mentioned before, the directional distribution of photons emitted by the source is uniform, thus the origin of each ray is selected by uniformly sampling the exposed detector sides. The number of samples is given by the apparent activity for each of the sides.

The rays tend to form a compact bundle, as the detector is extremely small. Therefore, it is not necessary to detect collisions with all obstacles along the line of each ray. Instead, the environment loss L_{env} is computed once per simulation step by casting a ray from the center of the detector towards the radiation source:

$$L_{env} = L_{air}(d_s - \sum_{i=1}^{N} d_i)\prod_{i=1}^{N} L_i(d_i)\ , \tag{10}$$

where $L_{air} = 1 - P_{air}$ is the loss induced by air absorption, d_s is the Euclidean distance between the source and the detector, d_i is the distance a photon has to travel through i-th obstacle, L_i is the loss caused by absorption in the obstacle and N is the number of obstacles intersected by the ray. The result is used to decrease the apparent activity of a source, and all subsequently simulated photons are treated as if they were in a lossless environment.

For each photon, a pseudorandom number in the range $[0, 1]$ is generated. If the number is lower than P_e, the photon is detected. This step preserves the stochastic nature of radiation sources. The process is visualized in Fig. 3, which shows an outline of the detector and individual rays originating from two different sources. The complete simulation loop is presented in Algorithm 1. The frequency of the loop is

[2] National Institute of Standards and Technology

determined by the readout rate of the simulated sensor, and the number of particles to be simulated in one step is derived from the apparent activity of the source and the environment loss.

The proposed model also supports simultaneous use of multiple sensors and radiation sources. The unique environment loss and apparent activity have to be computed separately for each source-detector pair. Multiple sources may be used to counteract the point source approximation in order to model a widespread contamination.

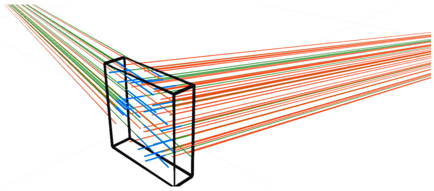


Fig. 3: Visualization of one simulation step. The Timepix semiconductor block is shown as a black box. The gamma radiation from two sources is being simulated. Red rays represent photons, which pass through the detector undetected. Green rays are absorbed and detected. The tracks of absorbed photons inside the detector are highlighted in blue color. The probability of detection is affected by the length of the track, the detector material and the photon energy.

C. ROSPIX

To allow seamless transition from simulations into real deployment of robots, a ROS interface for Timepix sensors has been developed [31]. This powerful tool enables direct streaming of the Timepix output in the form of ROS messages, a dynamic configuration of sensor parameters (such as acquisition time or bias voltage) via ROS services, automatic communication recovery, and also processing of measured data in real-time, directly onboard the robotic platform. This step is crucial for active localization of radiation sources. The Rospix tool also includes a built-in image processing pipeline, which is capable of distinguishing different particle types in real-time based on the shape of the track created along the detector surface [32]. This allows the system to filter out natural radiation background. Reliability of the Rospix driver has already been proven during a sub-orbital rocket launch as a part of the Rocket EXperiment (REX), which carried two Timepix sensors as a part of its payload [33].

V. SIMULATION RESULTS

A simple scenario with one MAV and one radiation source is presented for a baseline performance evaluation of the simulator. The MAV follows a desired path at a fixed height of 1.5 m with a fixed orientation, so that the Timepix is always pointed in the direction of the map Y-axis. The MAV continually collects measurements from the onboard Timepix detector. A map of radiation intensity, with resolution of 0.5 m is built in real-time from the measurements. The path shown in Fig. 6 was generated so that each cell of the map is explored by the MAV.

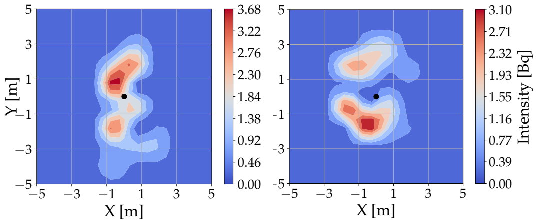
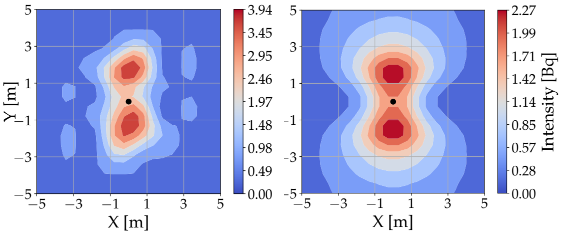


Fig. 4: Examples of radiation maps created in the simulated environment. The parameters of the simulation were identical to the parameters of the real experiment. The MAV moved at a fixed height of 1.5 m with a fixed orientation. A sample of ^{241}Am with activity of 0.5 GBq was placed in the origin of the coordinate system. The differences in the maps are introduced by the randomized photon detection events.



(a) Simulation result

(b) Theoretical expectation

Fig. 5: Mean average radiation intensity map (a) generated as a result of 100 consecutive simulations. The measurements were obtained at a fixed height of 1.5 m with a fixed orientation of the sensor pointed in the direction of Y-axis. A sample of ^{241}Am with the activity of 0.5 GBq was placed at the origin of the coordinate system for all iterations. The map shows strong agreement with a theoretical expectation (b) generated from the mathematical model after removing the stochastic element.

Fig. 4 shows examples of maps, which were created by the MAV in the simulated environment of Gazebo. The stochastic nature of the radiation source causes the results to look different in each iteration. A quantitative analysis was performed by launching the simulation 100 times with the same parameters. The mean average distribution of the measured radiation intensity across all 100 mapping iterations is shown in Fig. 5a. On a large scale, the distribution of the radiation intensity resembles an overlay of two multivariate Gaussian distributions. Replacing the final stochastic step of the simulation by the probability of absorption in the detector, an ideal map of radiation intensity can be plotted as a function of position. The map shown in Fig. 5b represents the theoretical expectation based on the previously described physical principles of our model. The measured intensity decreases around the true position of the source due to the orientation of the detector. Directly above the source, the largest side of the detector is not exposed to any photons.

VI. COOPERATIVE LOCALIZATION

For large areas, the mapping approach may be too time-consuming, and a different approach is necessary. However, it may not be possible to determine the position of a radiation source from the measured intensity, without fully covering the area. In an obstacle-free area, a gradient-following approach may be used to navigate a MAV towards a radiation source. However, in an environment with obstacles that attenuate the radiation intensity, only a local maximum will be reached.

The uneven shape of the detector can be exploited to make a crude estimate of the direction of incoming radiation even

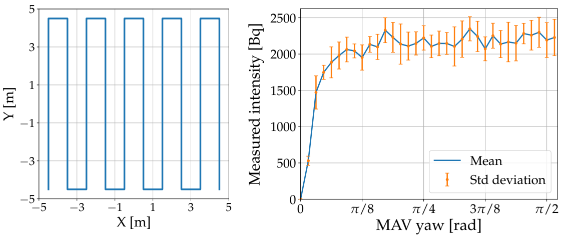


Fig. 6: (Left) A space-filling path generated for the MAV to follow in order to obtain measurements in all cells of the map.
(Right) Simulated changes of radiation intensity in relation to the orientation of the detector. The MAV and the radiation source were placed 5 meters apart and this distance did not change during the simulation. The yaw of the MAV was incremented by $\frac{\pi}{20}$ rad after each 10000 measurements. The prominent minimum in the measured radiation intensity can be exploited for an estimate of the radiation direction.

with a single-chip device. Rotating the detector around its axis causes measurable changes in particle flux, as it affects the size of the detector surface exposed to the radiation. This is illustrated in Fig. 6, where a significant decrease in measured radiation intensity occurs. This is a result of the largest detector side being parallel to the direction of the incoming photons.

Mounting the sensor as in Fig. 7 allows for changes of sensor orientation by increasing the yaw of the MAV. The position of the radiation source can be estimated as the intersection of the directional estimates taken at multiple positions in the area. The measurements can be taken by a single MAV. However, employing multiple MAVs significantly decreases the time required to find the radiation source, as multiple measurements are taken simultaneously. This also greatly increases the localization precision and provides the option to track moving radiation sources.

The MAV remains in one position while changing its yaw. This allows the method to be environment-invariant, as the obstacle attenuation does not change while performing the estimation. The Timepix can therefore be considered a crude sensor of radiation direction, and the source position can be estimated as an intersection of multiple direction measurements taken in different positions. Simulations have shown, that the accuracy of the direction estimate with this method is approximately ±9 degrees. To improve the precision, we can formulate the localization problem as a superresolution task. This approach is typically used for detail improvement and noise suppression in imaging techniques, by combining multiple lower resolution images [34]. Related projects employ multiple signal classification (MUSIC) algorithm for localization of radio transmitters [35], or targets in radar measurements [36]. Variants of the Kalman filter have also been used to improve resolution of images [37], [38] and depth images [39]. In this work, we employ a Linear Kalman filter to fuse direction estimates taken by the MAVs to improve the precision of radiation source localization.

In the following section, operator $\hat{}$ is used to denote an estimate rather than the actual value. The state vector $\mathbf{x} = (s_x, s_y)^T$ is the position of the radiation source in the world coordinate system, $\mathbf{P}$ is the covariance of the position estimate, $\mathbf{Q}$ is the measurement covariance and k is the iteration index.

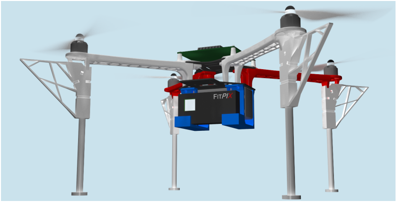


Fig. 7: The DJI F450 quadrotor with a FITPix device mounted to the front of the frame. The 14×14 mm Silicon surface of the Timepix detector is visible as a silver square.

For a single, static radiation source, the prediction step of the filter is reduced to:

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k} \,, \tag{11}$$

$$\mathbf{P}_{k+1|k} = \mathbf{P}_{k|k} + \mathbf{Q} \,. \tag{12}$$

The direction estimate is introduced as a new state observation $\mathbf{y}$ with a known observation covariance $\mathbf{R}$, which is calculated in the camera coordinate frame (X – forward, Z – up), with Timepix representing the camera. The covariance ellipsoid is extremely enlarged in the forward direction, as the distance between the source and the detector cannot be determined without knowing the activity of the source in advance. The ellipsoid is then transformed into the world coordinate frame, assuming the MAV pose in the world frame is known. This provides an estimate of the position of the radiation source in the world coordinates. Then, a correction step is performed by computing the Kalman gain $\mathbf{K}$ and updating the estimate:

$$\mathbf{K} = \mathbf{P}_{k|k-1}\left(\mathbf{P}_{k|k-1} + \mathbf{R}\right)^{-1} \,, \tag{13}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}\left(\mathbf{y}_k - \hat{\mathbf{x}}_k\right) \,, \tag{14}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}\right)\mathbf{P}_{k|k-1} \,. \tag{15}$$

To obtain measurements in multiple locations, the entire area of interest is uniformly sampled, and the MAVs are assigned a predefined number of waypoints, in which the direction estimate is performed. The sensor is turned off while moving to a new waypoint.

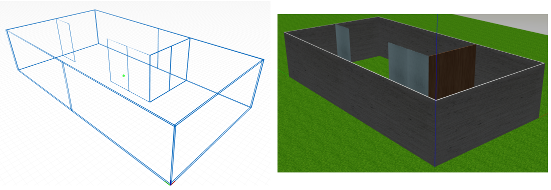


Fig. 8: The simulated environment with obstacles shown in ROS visualization tool (top) with transparent obstacles, and in Gazebo (bottom) with obstacles textured according to the simulated materials (concrete, wood).

A series of simulations was performed to evaluate the efficiency of the active localization approach. Two MAVs were deployed in an area with obstacles of different materials (concrete, wood) shown in Fig. 8. The layout of all obstacles was known by the MAVs in advance, and a radiation source was placed in an unknown position within the area. Each MAV followed a cyclic routine shown in Fig. 9. The localization process is illustrated in Fig. 10 with four snapshots from simulation showing iterative improvements with additional direction estimations by the MAVs.

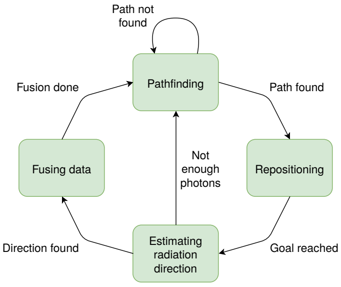


Fig. 9: A diagram illustrating the routine of individual MAVs during the cooperative localization process. The measurements from all MAVs are transmitted via WiFi to a central node (either one of the MAVs or a ground station), which fuses the data and estimates the position of the source.

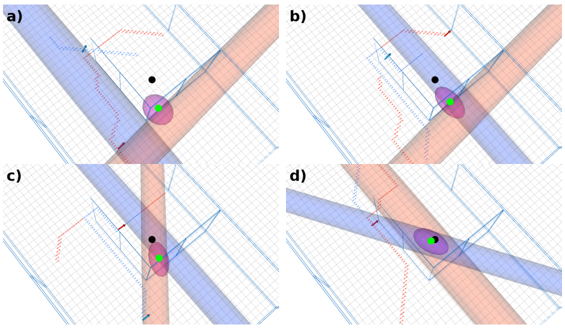


Fig. 10: A sequence of images showing the iterative improvement in precision of source localization with additional measurements. Orange and blue ellipsoids represent the latest directional estimates obtained by the MAVs. Actual position of the radiation source is represented by a black marker, while the estimated position of the source is represented by a green marker. Covariance of the estimate is represented by a purple ellipsoid.

VII. Experiments

A series of experiments was performed with one MAV in an outdoor environment. In order to validate the credibility of the simulator, the conditions of the real experiment were identical to the simulation with one MAV presented earlier.

A single source of gamma photons was placed in an area without obstacles. The source was a sample of Americium-241 with activity of 0.5 GBq. The energy of the photons produced by this isotope is 59.5 keV. A quadrotor built on the DJI F450 frame, with a Pixhawk flight controller and an Intel NUCi7 onboard computer, was used as the robotic platform. Timepix with a 300 µm Silicon chip was used as a radiation detector. The detector was connected to the onboard PC via a FITPix USB 2.0 interface [40] and the Rospix driver [31] was used to stream measurements into ROS as messages. The detector was mounted to the front of the MAV, as shown in Fig. 7. The position of the MAV was determined by fusing data from a uBlox GPS module with a compass. The optic flow was provided by a downward-facing mvBlueFOX camera, and the distance from ground was provided by a Garmin LIDAR-Lite v3 laser rangefinder.

The MAV followed the path shown in Fig. 6 at a fixed height above ground and maintained a fixed heading. The horizontal velocity of the MAV was limited to 1 m/s, and the exposure time of Timepix was set to 0.1 s. The images were processed by an image classification pipeline [32], to filter out natural background radiation other than the gamma photons. The resulting gamma radiation intensity was stored in a grid with resolution of 0.5 m^2. Maps created during the experiments are shown in Fig. 11 for flights at 1.5 m above ground and in Fig. 12 for flights at 3 m above ground.

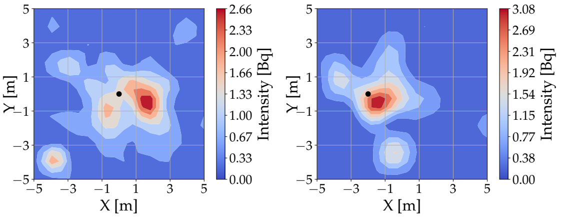


Fig. 11: Radiation maps created by the MAV moving at a height of 1.5 m above ground. The true position of the sample of ^{241}Am with activity 0.5 GBq is marked by a black dot.

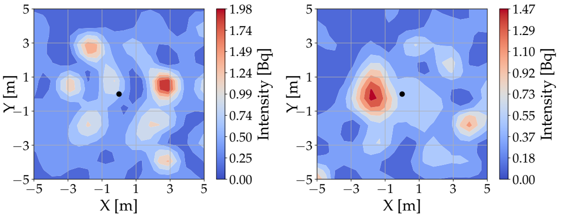


Fig. 12: Radiation maps created by the MAV moving at a height of 3 m above ground. Despite a comparatively low activity of the radiation source (0.5 GBq), Timepix was still capable of detecting the emitted gamma photons. The photon flux was, however, too low for the passive mapping to determine the position of the source accurately. The need to localize the radiation source with more precision motivates the development of more advanced active localization methods.

The experiments have proven, that a single MAV equipped with one Timepix detector is capable of estimating the position of a relatively weak radiation source while flying at 1.5 m above ground. Increasing the MAV height resulted in insufficient amount of photons captured in each grid cell, and the resulting is not guaranteed to show a clear peak in the radiation intensity. The results may be improved by lowering the horizontal velocity of the MAV to allow the detector to collect more photons, but this would result in significantly longer flight duration. Using an active localization approach with multiple MAVs would significantly reduce the time requirements.

VIII. Conclusion and Future Work

We have presented a complete open-source software package for the development of autonomous robotic systems dealing with detection and localization of radioactive sources. A novel ray-tracing based simulator, which allows users to create realistic simulations with multiple radioactive sources and Timepix detectors has been developed, and our implementation in the form of a plugin for Gazebo is available as open-source. We have also presented a ROS interface for the Timepix detector, which allows full integration of the powerful and compact sensor into an existing robotic system[1]. The properties of the simulated environment have been statistically evaluated and the credibility and the accuracy of the simulation were evaluated by a series of real-world experiments. During the experiments, a MAV equipped with a Timepix radiation detector onboard created a map of the gamma radiation intensity released by a sample of Americium-241 in real-time.

Our efforts are further focused on developing an autonomous system for fast and precise localization of radiation sources by cooperating micro aerial vehicles. We especially focus on exploring GPS-denied environments with obstacles, e.g. underground parking areas or former Uranium ore mines. With this system, we aim to provide a significant contribution to routine nuclear safety surveying as well as to rapid homeland security response systems.

REFERENCES

[1] GEANT Collaboration, S. Agostinelli, *et al.*, "Geant4, a simulation toolkit," *Nucl. Instrum. Meth. A*, vol. 506, no. 25, 2003.

[2] D. Drouin. (2016) CASINO, simulation of electron trajectory in solids. [Online]. Available: https://www.gel.usherbrooke.ca/casino/

[3] T. Böhlen, F. Cerutti, M. Chin, A. Fassò, A. Ferrari, *et al.*, "The fluka code: developments and challenges for high energy and medical applications," *Nuclear data sheets*, vol. 120, pp. 211–214, 2014.

[4] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ IROS*.

[5] O. S. R. Foundation. (2014) Gazebo robot simulation made easy. [Online]. Available: https://gazebosim.org/

[6] Energid Technologies. (2019) Actin: Industrial-grade software for advanced robotic solutions. [Online]. Available: https://www.energid.com/actin

[7] O. Michel, "Cyberbotics Ltd. Webots: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.

[8] N. Pinkam *et al.*, "Exploration of a group of mobile robots for multiple radiation sources estimation," in *IEEE IRIS*, 2016.

[9] A. A. R. Newaz, S. Jeong, H. Lee, H. Ryu, and N. Y. Chong, "Uav-based multiple source localization and contour mapping of radiation fields," *Robotics and Autonomous Systems*, vol. 85, pp. 12–25, 2016.

[10] F. Mascarich, T. Wilson, C. Papachristos, and K. Alexis, "Radiation source localization in gps-denied environments using aerial robots," in *IEEE ICRA*, 2018.

[11] A. Miller, R. Machrafi, and A. Mohany, "Development of a semi-autonomous directional and spectroscopic radiation detection mobile platform," *Radiation Measurements*, vol. 72, pp. 53–59, 2015.

[12] M. S. Lee, D. Shy, W. R. Whittaker, and N. Michael, "Active range and bearing-based radiation source localization," in *IEEE/RSJ IROS*.

[13] K. Nagatani *et al.*, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[14] Y. Sato *et al.*, "Radiation imaging using a compact compton camera mounted on a crawler robot inside reactor buildings of fukushima daiichi nuclear power station," *Journal of Nuclear Science and Technology*, pp. 1–8, 2019.

[15] D. Kim, H. Woo, Y. Ji, Y. Tamura, A. Yamashita, and H. Asama, "3d radiation imaging using mobile robot equipped with radiation detector," in *IEEE/SICE SII*, 2017.

[16] P. G. Martin, S. Kwong, N. Smith, Y. Yamashiki, *et al.*, "3d unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 12–19, 2016.

[17] Y. Sanada and T. Torii, "Aerial radiation monitoring around the fukushima dai-ichi nuclear power plant using an unmanned helicopter," *Journal of environmental radioactivity*, vol. 139, pp. 294–299, 2015.

[18] J. MacFarlane, O. Payton, *et al.*, "Lightweight aerial vehicles for monitoring, assessment and mapping of radiation anomalies," *Journal of environmental radioactivity*, vol. 136, pp. 127–130, 2014.

[19] J. Jiang *et al.*, "A prototype of aerial radiation monitoring system using an unmanned helicopter mounting a gagg scintillator compton camera," *Journal of Nuclear Science and Technology*, vol. 53, no. 7, pp. 1067–1075, 2016.

[20] G. Christie, A. Shoemaker, K. Kochersberger, P. Tokekar, L. McLean, and A. Leonessa, "Radiation search operations using scene understanding with autonomous uav and ugv," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1450–1468, 2017.

[21] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-uav technologies for contour mapping of nuclear radiation field," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 401–410, 2013.

[22] J. Towler, B. Krawiec, and K. Kochersberger, "Radiation mapping in post-disaster environments using an autonomous helicopter," *Remote Sensing*, vol. 4, no. 7, pp. 1995–2015, 2012.

[23] H.-I. Lin *et al.*, "Search strategy of a mobile robot for radiation sources in an unknown environment," in *IEEE ARIS*, 2014.

[24] S. Chaiyasoonthorn, N. Hongyim, and S. Mitatha, "Building automatic packet report system to report position and radiation data for autonomous robot in the disaster area," in *IEEE ICCAS*, 2015.

[25] R. Pavlovsky *et al.*, "3-d radiation mapping in real-time with the localization and mapping platform lamp from unmanned aerial systems and man-portable configurations," *arXiv preprint arXiv:1901.05038*, 2018.

[26] Y. Du *et al.*, "Evaluation of a compton scattering camera using 3-d position sensitive cdznte detectors," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 457, no. 1-2, pp. 203–211, 2001.

[27] ADVACAM. (2019) MiniPIX, portable small usb camera. [Online]. Available: http://advacam.com/camera/minipix

[28] S. Bheesette *et al.*, "Medipix3rx neutron camera for ambient radiation measurements," in *IEEE NSS/MIC*, 2017.

[29] T. Foote. (2018) Robot Operating System documentation. [Online]. Available: http://wiki.ros.org/

[30] J. Hubbell and S. Seltzer, "Nist standard reference database 126," *Gaithersburg, MD: National Institute of Standards and Technology*, 2004. [Online]. Available: https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients

[31] T. Baca, D. Turecek, R. McEntaffer, and R. Filgas, "Rospix: modular software tool for automated data acquisitions of timepix detectors on robot operating system," *Journal of Instrumentation*, vol. 13, no. 11, p. C11008, 2018.

[32] T. Baca, M. Jilek, P. Manek, P. Stibinger, V. Linhart, J. Jakubek, and M. Saska, "Timepix radiation detector for autonomous radiation localization and mapping by micro unmanned vehicles," in *IEEE/RSJ IROS*, 2019.

[33] V. Dániel *et al.*, "Rex le x-ray telescope experiment overview," in *EUV and X-ray Optics: Synergy between Laboratory and Space VI*, vol. 11032. International Society for Optics and Photonics, 2019, p. 1103206.

[34] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.

[35] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[36] J. Odendaal, E. Barnard, and C. Pistorius, "Two-dimensional superresolution radar imaging using the music algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 10, pp. 1386–1391, 1994.

[37] C. Newland, D. Gray, and D. Gibbins, "Modified kalman filtering for image super-resolution," in *Image and Vision Computing New Zealand*, vol. 2006, 2006, pp. 79–84.

[38] G. H. Costa and J. C. M. Bermudez, "Statistical analysis of the lms algorithm applied to super-resolution image reconstruction," *IEEE Trans. on Signal Processing*, vol. 55, no. 5, pp. 2084–2095, 2007.

[39] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten, "Real-time non-rigid multi-frame depth video super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 8–16.

[40] V. Kraus, M. Holik, J. Jakubek, M. Kroupa, P. Soukup, and Z. Vykydal, "Fitpix, fast interface for timepix pixel detectors," *Journal of Instrumentation*, vol. 6, no. 01, p. C01079, 2011.

[1] Source code repository: https://github.com/rospix
Multimedia: http://mrs.felk.cvut.cz/radiation2020ral

Chapter 6

Discussion and Results

In this chapter, the contributions presented in the core articles are summarized. Furthermore, a discussion in context with the remainder of the author's work is provided. Lastly, future work in the context of the three research streams is suggested.

6.1 Research-focused UAV System

The proposed UAV system [8c], [6c] provides students and researchers the means to develop and test new methods in the field of feedback control, tracking, estimation, and planning. The system is modular and allows its users to add new components that can be safely tested in simulations, and even during a real-world flight. Moreover, the system can be used without changes for testing of high-level localization methods, mapping, planning, and multi-robot coordination.

The first of these contributions is the Model Predictive Control tracker [6c]. Even now, more than four years after it was first developed, a viable substitute for the tracking mechanism has not yet been found. The MPC tracker provides a mechanism for real-time generation of feasible and smooth feedforward control reference from any time-parametrized input trajectory. The tracker's output satisfies UAV constraints even when the input trajectory is not feasible. The real-time tracking shows near-optimal results of minimizing the square error from the input trajectory due to the linear MPC problem formulation. Moreover, the integrated mutual collision avoidance mechanism allows the safe deployment of multiple Unmanned Aerial Vehicle in outdoor conditions.

The MRS UAV system [8c] comes with two controller designs — extended *SE(3) geometric tracking* [168] for agile and aggressive flight and the novel *MPC controller* for stable flight in case of potentially unreliable UAV state estimate. However, the system can be easily extended with new control approaches as needed thanks to its modularity. UAV controllers' survey provides a rich list of potentially useful control techniques [72]. For example, a novel adaptive backstepping controller [67], [79] may provide better performance during aggressive maneuvers due to the included rotor drag compensation. The proposed extension to *geometric tracking on SE(3)* [168] can be further improved with remarks from [152] to provide robust control to bounded uncertainties. Furthermore, nonlinear MPC controllers are becoming popular [71], [74], [103] thanks to their inherent ability to cope with complex constraints. However, when dealing with theoretical work and experimental deployment in real-world conditions, practicality over complexity is favored. Therefore, relatively simple controllers [8c] with well-tractable inner mechanisms are utilized. Moreover, we still find even relatively old concepts [168] to be worth improving by incremental and implementation ways, as described in [8c].

The proposed control system has the following key properties:

- bank-of-filters estimator that overcomes challenges with diverse sensory equipment,

- heading-oriented control design, devoid of ambiguous use of Euler/Tait-Bryan angles,
- body/world disturbance estimation that does not rely on specific state estimator design,
- reliable MPC-based controller with the benefits of the nonlinear *SO(3)* force feedback,
- system for utilizing a variety of onboard localization systems and sensors,
- input in any coordinate frame, which may differ from the feedback loop reference frame,
- built upon the MPC tracker for control reference generation [6c].

The system has been used not only for research testing and evaluation, but also for education and teaching. The system was utilized for practical exercises during the 2019 and 2020 IEEE RAS Summer School on Multi-Robot Systems[1]. Over a hundred students used the system while solving the tasks of the multi-UAV traveling salesman problem with neighborhoods (2019) and relatively-localized leader-follower flight with two UAVs (2020). Both tasks were finished with field experiments where the student's code was tested (see Fig. 6.1a).

The future work on the MRS UAV system is currently foreseen as mostly implementation improvements, including the transition to ROS2 and continuous improvement of the system's documentation. However, some research topics remain. The heading-based control approach should be generalized to allow the user to specify a vector in the UAV's body frame that represents the UAV *front*, other than the currently-used principle x-axis. The azimuth of this vector should be considered as the heading. This would allow precise tracking and control of the heading with respect to, e.g., an arbitrary optical axis of a camera mounted on the UAV. Without such a mechanism, the heading control relative to other than the UAV body's x-axis can be achieved only approximately and with notable errors, especially during aggressive maneuvers. However, this mechanism needs to be incorporated in both control and reference generation. The mechanism must also guarantee smooth control references and control actions even when the definition of the UAV's *front* changes in mid-flight.

6.2 Advances in remote sensing by UAVs

The UAV tasks tackled within the core publications in this thesis were on the edge of field robotics at their time. We successfully showed a multi-robotic system for autonomous object gathering [2c] and autonomous wall construction [7c] by UAVs. Both systems were put to the test during the MBZIRC 2017 and MBZIRC 2020 robotics competitions where both achieved first place among respected university teams from all over the world (see Figures 6.1b, and 6.1c). The task of autonomous landing with a UAV on top of a moving car [3c] was also investigated, contributing to the second place in the competition and the fastest autonomous landing time among all the competing teams. In the future, this robotics problem may be part of automatic remote sensing applications with UAVs being deployed and recovered by a car. Each of the three problems brought us further in understanding the caveats of onboard state estimation, computer vision, control, and planning. The important lesson learned about robotics from these endeavors is the danger of overly focusing on one or a few specializations. UAV autonomy is a complex cybernetics field that encompasses many interconnected subfields. We found that researchers often focus too narrowly on one of the subfields while ignoring the influence between other components of autonomous systems. This can make the approaches narrow-minded, detached from reality, and potentially not applicable to a real-world robotics system.

Although the particular problems solved in [7c], [2c], [3c] are probably not going to be

[1] http://mrs.felk.cvut.cz/summer-school-2020

(a) 2019 IEEE RAS Summer School

(b) MBZIRC 2017 Award Ceremony.

(c) Team members during the MBZIRC 2020 Award Ceremony.

pursued in the future by our team, the acquired know-how and knowledge are transferable to other applications. For example, we also pursue the Defense Advanced Research Projects Agency (DARPA) Subterranean competition on multi-robotic autonomous localization of objects and people in underground environments [29a], [10a], [18a]. Furthermore, the primary research on relatively-localized multi-UAV swarms [27a], [28a], [9a], [12a] benefits significantly from the systems and expertise we obtained in experimental field robotics. Finally, the following section discusses our current and future work in the remote sensing subfield of ionizing radiation sources.

6.3 Ionizing radiation sources localization

We have come to the field of ionizing radiation detection from a different direction and using a different perspective than the rest of the UAV community, specifically from the constrained environment of miniature space technologies [4c], [39a]. Where the rest of the community often attempts to mount as large radiation detectors as possible to a UAV (to maximize measured counts-per-second), we strive towards miniaturizing both the sensor and the UAV to deploy the UAV in tightly constraint environments. The distance to the radiation source even in a cluttered environment can be closed, and therefore obtain even more information due to the inverse square law of radiation intensity over distance. The concept of obtaining as much information as possible using large detectors is prominent throughout the state of the art. However, multiple factors should be considered when building an autonomous aerial system for radiation source localization. We identified the following four factors:

- detector size,
- aircraft size,
- environmental constraints,
- localization strategy.

As depicted in Figure 6.1, these four factors influence each other in a loop. With a larger sensor comes a higher event count rate, which is understandably preferred by physicists. However, large and heavy sensors require larger aircraft with higher payload capacity. The aircraft size dictates its motion constraints through an environment; larger aircraft fly higher above the terrain and need increased safety distance from obstacles. Those constraints determine the search strategy that, in turn, typically favors sensors with higher sensitivity, and therefore sensors with a larger mass. The current state of the art does not break from this cycle in more than a couple of factors. On the contrary, the state of the art mostly utilizes UAVs just as a sensor carrier and there is little incentive for real-time onboard autonomy. However, we conjecture that all four factors need to be adjusted simultaneously to show a breakthrough in the field. In other words, the system needs to be developed while focusing on the specifics of sensors and the UAVs autonomy simultaneously.

Making the sensor smaller will lower its weight as well as the raw count-per-second information yield. However, miniature sensors can be mounted on an MAV that can safely fly much closer to environmental obstacles. Flying close to obstacles requires more intelligent onboard autonomy, that can sense the obstacles and plan the UAV motion with respect to the environment. Proximity to obstacles opens new possibilities for search strategies which can utilize the abilities of today's autonomous systems and the potential to close the distance to the radiation sources autonomously. We aim to make such a breakthrough in both research fields.

We have made contributions that allow us to employ the innovative Timepix sensors and Timepix3 event cameras onboard UAVs, and proposed the Rospix[2] interface [15a] for the Robot Operating System together with a particle classification pipeline [5c]. The traditional software and tools available for the Timepix family of detectors are dedicated solely for laboratory use and often expect a human input while conducting measurements. Also, the data processing is commonly done only after all the data were measured. Any specific application outside of the intended use requires specific hardware and software to be developed in collaboration with the sensor interface manufacturer, as it was, e.g., for the VZLUSAT-1 nanosatellite [24a]. The ROS tools provided through Rospix allow full automation using standard tools within the robotics community. Moreover, the particle classification pipeline is the first available to be executable during the data acquisition and is available as open source. The classification pipeline was trained using diverse radiation data[3] gathered by the VZLUSAT-1 nanosatellite mission [4c].

Our first contributions related to the radiation dosimetry onboard UAVs consist of the review of radiation localization options for Unmanned Aerial Vehicle, the particle classification pipeline, and a real-time Monte Carlo ray-tracing model for a dual-sensor Compton

[2] `https://github.com/rospix`

[3] `https://github.com/vzlusat`

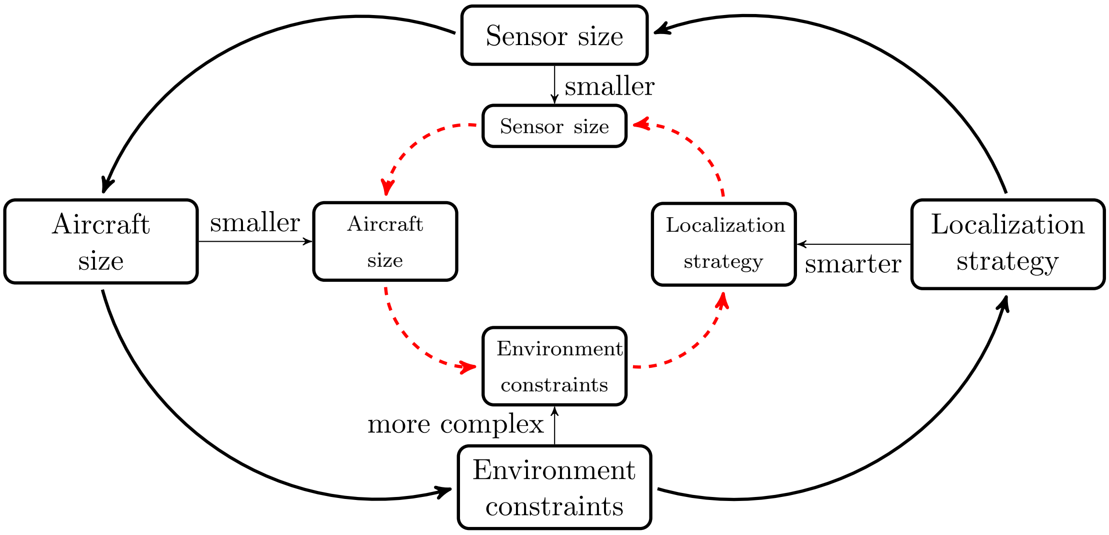


Figure 6.1: The four interconnected factors of an autonomous aircraft system for localization of ionizing radiation sources. All four factors need to be adjusted simultaneously to introduce a breakthrough in the field (the smaller red cycle).

camera [5c]. Furthermore, the use of the bare Timepix sensor for multi-robotic localization of compact radiation sources [1c] without a specialized mechanism to deduce a direction to the source is investigated. We show the approach's potential using realistic robotic simulations and a real-time Monte Carlo ray-tracing model of gamma radiation. The previously mentioned contributions were motivation and a precursor for the project proposal for the Technology Agency of the Czech Republic. Such interdisciplinary tasks as the localization of ionizing radiation sources can be challenging to accomplish by an institution specializing in either field. Therefore, the thesis author co-wrote a project proposal, together with *Advacam, s.r.o.* and the *Czech Metrology Institute*, that would allow focusing on the topic and dedicate more resources to the research. Project no. FW01010317 (2020 – 2022, 8th of 396 submitted for the call[4]) is called "*RADRON: Localization of ionizing radiation sources using small unmanned helicopters equipped with a Compton camera detector*". The project focuses on utilizing the unique properties of the Timepix3 [147] event cameras for real-time localization and tracking of compact radiation sources. The first project results, submitted to *IEEE ICRA* [34a], show the Timepix3 event camera used as a single-detector Compton camera onboard a UAV. Experiments using an autonomous UAV being localized by LiDAR SLAM are showed first. The system was able to localize a relatively weak gamma source using only onboard sensors.

> [34a] **T. Baca**, P. Stibinger, D. Doubravova, D. Turecek, J. Solc, J. Rusnak, *et al.*, "Gamma Radiation Source Localization for Micro Aerial Vehicles with a Miniature Single-Detector Compton Event Camera," submitted to IEEE ICRA, 2020. eprint: `arXiv:2011.03356`

The future work in this research field is co-aligned with the project goals. We focus on developing an MAV equipped with a Compton camera based on the Timepix3 technology and aim to combine the Compton camera principle [34a] for localization of compact gamma

[4] https://www.tacr.cz/soutez/program-trend/prvni-verejna-soutez-trend

sources, together with the count-based system [1c] that can be utilized for ionizing charged particles.

6.4 Summary

The contributions and achievements of the submitted doctoral thesis and the related publications are summarized as:

- An MPC trajectory tracking approach was proposed which allows fast and precise tracking of desired trajectories. The tracker also provides distributed collision avoidance for multiple UAVs sharing the same outdoor workspace.
- A control architecture that integrates the MPC tracking approach in conjunction with a modified non-linear geometric tracking controller was proposed.
- The proposed control architecture is integrated into the provided open-source ROS control system. The system has been utilized during research, education, and experimental activities of the MRS group at CTU in Prague and has been made available to the public in further support of open research and experimental validation.
- The problems of multi-UAV collaborative missions for autonomous object manipulation and delivery were studied in the context of the MBZIRC 2017 and 2020 missions. During the objective experimental evaluation of the competition, the proposed systems focusing on visual servoing demonstrated the best results among all other teams.
- The problem of autonomous UAV visual servoing of a moving car and subsequently landing on the car was also studied in the context of the MBZIRC 2017 competition. The proposed solution showed excellent performance during the evaluation, while also achieving the fastest landing among competing teams.
- Remote sensing of the 500 km low Earth orbit's radiation background had been conducted using the novel pixel detector Timepix since 2017. The data (shared openly) provide valuable information on radiation doses by the various present particle types.
- A system for the radiation dosimetry using Timepix sensors onboard UAVs was proposed. The system consists of a real-time Monte-Carlo ray-tracing simulator, software for integrating Timepix sensors onboard a UAV, and a pipeline for real-time track classification. Simulations were conducted showing the potential of the approach.
- A Compton camera sensor, based on the Timepix3 event camera, on a UAV was first proposed and tested onboard as the first step in an ongoing project regarding localization of compact sources of ionizing radiation by UAVs.

Chapter 7

Conclusion

This thesis addressed collaborative remote sensing by a group of Unmanned Aerial Vehicles. First, we focused on developing a UAV control system for experimental evaluation of new methods in realistic real-world conditions. The proposed system utilizes an MPC tracking approach for generating feedforward control reference provides a modular control pipeline. A novel feedback controller design combining an MPC with geometric tracking controller on *SO(3)* was proposed, together with a heading-based orientation system that is devoid of the convention of Euler and Tait-Bryan angles. The control system has been extensively tested throughout the years and provided the means for other research to be conducted. One of the many uses of the system was during the 2017 and 2020 MBZIRC robotics competitions. We tackled several problems on the edge of the state of the art in field robotics at the time. Our solutions to the problems of multi-UAV object gathering and multi-UAV brick wall construction proved to be the best in the world, with our solution to the autonomous landing of a UAV on a moving car performing the fastest landing among the competitors. This thesis's collection of core publications reports on complex cybernetics problems of feedback control, estimation, sensor fusion, computer vision, planning, and mechatronics subfields of aerial robotics. Finally, contributions in the field of aerial localization of ionizing radiation sources were presented. We have made the first steps into adopting the Timepix family of CMOS hybrid semiconductor sensors to be used onboard a mobile robot. The sensors' properties have been shown, providing essential tools for their use onboard a robot, and a methodology for their application has been suggested. First experiments were conducted with the Timepix and Timepix3 sensors onboard a UAV. Our latest results — the first results of an ongoing project — showed successful use of the Timepix3 event camera in the form of Compton camera onboard a UAV for localization of a compact gamma radiation source.

Chapter A

References

References to the author's work are listed first, followed by other references cited within this work. The authored references contain his contribution and the number of citations based on Web of Science (WoS), Scopus, and Google Scholar (GS). The author has reached h-index 7. The citation counts were gathered on January 1st, 2021.

A.1 Thesis core publications

Core articles in peer-reviewed journals

[1c] P. Stibinger, **T. Baca**, and M. Saska, "Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles With Pixel Detectors in Real-Time," *IEEE Robotics and Automation Letters*, vol. 5, pp. 3634–3641, 2 Apr. 2020, ISSN: 2377-3766, **Contributions: PS: 75%, TB: 12.5%, MS: 12.5%, IF 3.608 (Q1 in Robotics), citations: 0 in WoS, 0 in Scopus, 0 in GS**.

[2c] 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, 125–148, 2019, **Contributions: VS: 50%, TB: 22%, MS: 4%, RP: 4%, TK: 4%, JT: 4%, DT: 4%, GL: 4%, VK: 4%, IF 4.345 (Q1 in Robotics), citations: 19 in WoS, 22 in Scopus, 62 in GS**.

[3c] **T. Baca**, P. Stepan, B. 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*, vol. 36, pp. 874–891, 5 2019, **Contributions: TB: 50%, PS: 20 %, VS: 5%, DH: 5%, RP: 5%, MS: 5%, JT: 4%, GL: 4%, VK: 4%, IF 4.345 (Q1 in Robotics), citations: 11 in WoS, 11 in Scopus, 46 in GS**.

[4c] **T. Baca**, M. Jilek, I. Vertat, M. Urban, O. Nentvich, R. Filgas, C. Granja, A. Inneman, and V. Daniel, "Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements," *Journal of Instrumentation*, vol. 13, no. 11, p. C11010, 2018, **Contributions: TB: 60%, MJ: 5%, IV: 5%, MU: 5%, ON: 5%, RF: 5%, CG: 5%, AN: 5%, VD: 5%, IF 1.366 (Q3 in Instruments & Instrumentation), citations: 8 in WoS, 10 in Scopus, 14 in GS**.

Core conference articles

[5c] **T. Baca**, M. Jilek, P. Manek, P. Stibinger, V. Linhart, J. Jakubek, and M. Saska, "Timepix Radiation Detector for Autonomous Radiation Localization and Mapping by Micro Unmanned Vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2019, pp. 1–8, **rank A (CORE), Contributions: TB: 50%, MJ: 15%, PM: 10%, PS: 10%, VL: 5%, JJ: 5%, MS: 5%, citations: 0 in WoS, 1 in Scopus, 1 in GS**.

[6c] **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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8, **rank A (CORE), Contributions: TB: 60%, DH: 16%, GL: 8%, MS: 8%, VK: 8%, citations: 19 in WoS, 23 in Scopus, 65 in GS**.

Core articles — submitted

[7c] **T. Baca**, P. Penicka Robert Stepan, M. Petrlik, V. Spurny, D. Hert, and M. Saska, "Autonomous Cooperative Wall Building by a Team of Unmanned Aerial Vehicles in the MBZIRC 2020 Competition," *submitted to Robotics and Autonomous System*, Dec. 2020. eprint: `arXiv:2012.05946`, **Contributions: TB: 25%, RP: 25%, PS: 25%, MP: 6%, VS: 6%, DH: 6%, MS: 6%**.

[8c] **T. Baca**, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *submitted to JINT, after the 1st revision*, Aug. 2020. eprint: `arXiv:2008.08050`, **Contributions: TB: 60%, MP: 10%, VM: 6%, VS: 6%, RP: 6%, DH: 6%, MS: 6%**.

A.2 Thesis-related author's publications

Thesis-related articles in peer-reviewed journals

[9a] P. Petracek, V. Walter, **T. Baca**, and M. Saska, "Bio-Inspired Compact Swarms of Micro Aerial Vehicles without Communication and External Localization," *Bioinspiration & Biomimetics*, 2020, Accepted, available online `http://mrs.felk.cvut.cz/data/papers/BB20swarm.pdf`, **25% contribution, citations: 0 in WoS, 0 in Scopus, 0 in GS**.

[10a] M. Petrlik, **T. Baca**, D. Hert, M. Vrba, T. Krajnik, and M. Saska, "A Robust UAV System for Operations in a Constrained Environment," *IEEE Robotics and Automation Letters*, vol. 5, 2 Apr. 2020, ISSN: 2169-2176, **30% contribution, IF 3.608 (Q1 in Robotics), citations: 1 in WoS, 1 in Scopus, 9 in GS**.

[11a] D. A. Saikin, **T. Baca**, M. Gurtner, and M. Saska, "Wildfire Fighting by Unmanned Aerial System Exploiting Its Time-Varying Mass," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2674–2681, 2020, **25% contribution, IF 3.608 (Q1 in Robotics), citations: 1 in WoS, 1 in Scopus, 4 in GS**.

[12a] M. Saska, D. Hert, **T. Baca**, V. Kratky, and T. Nascimento, "Formation Control of Unmanned Micro Aerial Vehicles for Straitened Environments," *Autonomous Robots*, pp. 1573–7527, 2020, **20% contribution, IF 3.634 (Q1 in Robotics), citations: 0 in WoS, 0 in Scopus, 0 in GS**.

[13a] W. Giernacki, D. Horla, **T. Baca**, and M. Saska, "Real-time model-free minimum-seeking autotuning method for unmanned aerial vehicle controllers based on fibonacci-search algorithm," *Sensors*, vol. 19, no. 2, p. 312, 2019, **25% contribution, IF 3.031 (Q1 in Robotics), citations: 5 in WoS, 6 in Scopus, 24 in GS**.

[14a] G. Loianno, V. Spurny, J. Thomas, **T. Baca**, D. Thakur, D. Hert, R. Penicka, T. Krajnik, 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, **10% contribution (the first 7 authors contributed equally), IF 3.608 (Q1 in Robotics), citations: 26 in WoS, 30 in Scopus, 66 in GS**.

[15a] **T. Baca**, D. Turecek, R. McEntaffer, and R. Filgas, "Rospix: modular software tool for automated data acquisitions of Timepix detectors on Robot Operating System," *Journal of Instrumentation*, vol. 13, no. 11, p. C11008, 2018, **70% contribution, IF 1.366 (Q3 in Instruments & Instrumentation), citations: 3 in WoS, 3 in Scopus, 5 in GS**.

[16a] 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, **22% contribution, IF 3.634 (Q1 in Robotics), citations: 50 in WoS, 66 in Scopus, 142 in GS**.

[17a] J. Chudoba, M. Kulich, M. Saska, **T. Baca**, and L. Preucil, "Exploration and Mapping Technique Suited for Visual-features Based Localization of MAVs," English, *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2016, **15% contribution, IF 2.02 (Q3 in Robotics), citations: 3 in WoS, 5 in Scopus, 35 in GS**.

Thesis-related conference articles

[18a] T. Roucek, M. Pecka, P. Cizek, T. Petricek, J. Bayer, V. Salansky, D. Hert, M. Petrlik, **T. Baca**, V. Spurny, F. Pomerleau, V. Kubelka, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajnik, "DARPA Subterranean Challenge: Multi-robotic exploration of underground environments," in *IEEE Modelling and Simulation for Autonomous Systems (MESAS)*, vol. 11995, 2019, pp. 274–290, **5% contribution, citations: 2 in WoS, 0 in Scopus, 12 in GS**.

[19a] J. Faigl, P. Vana, M. Saska, **T. Baca**, and V. Spurny, "On solution of the Dubins touring problem," in *2017 IEEE European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–6, **10% contribution, citations: 3 in WoS, 7 in Scopus, 19 in GS**.

[20a] 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," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2017, **25% contribution, citations: 11 in WoS, 11 in Scopus, 46 in GS**.

[21a] **T. Baca**, P. Stepan, and M. Saska, "Autonomous Landing On A Moving Car With Unmanned Aerial Vehicle," in *IEEE European Conference on Mobile Robotics (ECMR)*, IEEE, 2017, **33% contribution, citations: 11 in WoS, 16 in Scopus, 52 in GS**.

[22a] M. Saska, **T. Baca**, and D. Hert, "Formations of unmanned micro aerial vehicles led by migrating virtual leader," in *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, IEEE, 2016, **rank A (CORE), 30% contribution, citations: 3 in WoS, 3 in Scopus, 29 in GS**.

[23a] V. Spurny, **T. Baca**, and M. Saska, "Complex manoeuvres of heterogeneous MAV-UGV formations using a model predictive control," in *2016 IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016, pp. 998–1003, **10% contribution, citations: 7 in WoS, 8 in Scopus, 51 in GS**.

[24a] **T. Baca**, G. Loianno, and M. Saska, "Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles," in *IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, IEEE, 2016, pp. 992–997, **80% contribution, citations: 25 in WoS, 29 in Scopus, 88 in GS**.

Thesis-related publications — submitted

[25a] T. Roucek, M. Pecka, P. Cizek, T. Petricek, J. Bayer, V. Salansky, D. Hert, M. Petrlik, **T. Baca**, V. Spurny, V. Kratky, P. Petracek, D. Baril, M. Vaidis, V. Kubelka, F. Pomerleau, J. Faigl, K. Zimmerman, M. Saska, T. Svoboda, and T. Krajnik, "System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA Subterranean Challenge," 2021, submitted to Field Robotics.

[26a] V. Walter, M. Petrlik, **T. Baca**, D. Zaitlik, L. Demkiv, and M. Saska, "Extinguishing of Real Fires by Fully Autonomous MAVs in the MBZIRC 2020 Competition," 2021, submitted to Field Robotics.

[27a] A. Ahmad, V. Walter, P. Petracek, M. Petrlik, **T. Baca**, D. Zaitlik, and M. Saska, "Autonomous Aerial Swarm in a Complex Environment without GNSS and without Communication," 2020, submitted to IEEE ICRA, **15% contribution**.

[28a] A. Dmytruk, T. Nascimento, A. Ahmad, **T. Baca**, and M. Saska, "Safe Tightly-Constrained UAV Swarming in GNSS-denied Environments," 2020, submitted to IEEE ICRA, **20% contribution**.

[29a] V. Kratky, P. Petracek, **T. Baca**, and M. Saska, "An Autonomous Unmanned Aerial Vehicle System for Fast Exploration of Large Complex Indoor Environments," *Submitted to Journal of Field Robotics*, 2020, Available online: `http://mrs.felk.cvut.cz/data/papers/darpa_urban.pdf`, **25% contribution**.

[30a] G. Silano, **T. Baca**, R. Penicka, D. Liuzza, and M. Saska, "Power Line Inspection Tasks with Multi-Aerial Robot Systems via Signal Temporal Logic Specifications," 2020, submitted to IEEE Robotics and Automation Letters, **20% contribution**.

[31a] D. Smrcka, **T. Baca**, T. Nascimento, and M. Saska, "Admittance Force-Based UAV-Wall Stabilization and Press Exertion for Documentation and Inspection of Historical Buildings," 2020, submitted to IEEE ICRA, **25% contribution**.

[32a] V. Spurny, V. Pritzl, V. Walter, M. Petrlik, **T. Baca**, P. Stepan, D. Zaitlik, and M. Saska, "Autonomous Firefighting Inside Buildings by an Unmanned Aerial Vehicle," *accepted to IEEE Access*, 2020, **15% contribution**.

[33a] Y. Stasinchuk, M. Vrba, **T. Baca**, V. Spurny, M. Petrlik, D. Hert, D. Zaitlik, and M. Saska, "A Multi-MAV System For Autonomous Multiple Targets Elimination in the MBZIRC 2020 competition," 2020, submitted to Field Robotics, **20% contribution**.

[34a] **T. Baca**, P. Stibinger, D. Doubravova, D. Turecek, J. Solc, J. Rusnak, M. Saska, and J. Jakubek, "Gamma Radiation Source Localization for Micro Aerial Vehicles with a Miniature Single-Detector Compton Event Camera," submitted to IEEE ICRA, 2020. eprint: `arXiv:2011.03356`, **70% contribution**.

[35a] M. Vrba, S. Yurii, **T. Baca**, V. Spurny, M. Petrlik, D. Hert, D. Zaitlik, and M. Saska, "Autonomous Capturing of Agile Flying Objects using MAVs: The MBZIRC 2020 Challenge," *submitted to IEEE Transactions on Systems, Man, and Cybernetics*, 2020, **20% contribution**.

[36a] V. Walter, V. Spurny, M. Petrlik, **T. Baca**, D. Zaitlik, and M. Saska, "Extinguishing of Ground Fires by Fully Autonomous UAVs motivated by the MBZIRC 2020 Competition," 2020, submitted to IEEE ICRA, **20% contribution**.

A.3 Partially-related author's publications

Partially-related articles in peer-reviewed journals

[37a] V. Daniel, A. Inneman, I. Vertat, **T. Baca**, O. Nentvich, M. Urban, V. Stehlikova, L. Sieger, P. Skala, R. Filgas, V. Zadrazil, R. Linhart, J. Masopust, T. Jamroz, L. Pina, V. Marsikova, L. Mikulickova, E. Belas, S. Pospisil, Z. Vykydal, Y. Mora, and R. Pavlica, "In-Orbit Commissioning of Czech Nanosatellite VZLUSAT-1 for the QB50 Mission with a Demonstrator of a Miniaturised Lobster-Eye X-Ray Telescope and Radiation Shielding Composite Materials," *Space Science Reviews*, vol. 215, no. 5, p. 40, Jul. 2019, ISSN: 1572-9672, **5% contribution, IF 8.142 (Q1 in Astronomy & Astrophysics), citations: 5 in WoS, 5 in Scopus, 5 in GS**.

[38a] M. Urban, O. Nentvich, V. Stehlikova, **T. Baca**, V. Daniel, and R. Hudec, "VZLUSAT-1: Nanosatellite with miniature lobster eye X-ray telescope and qualification of the radiation shielding composite for space application," *Acta Astronautica*, vol. 140, pp. 96–104, 2017, **20% contribution, IF 2.482 (Q1 in Engineering, Aerospace), citations: 26 in WoS, 29 in Scopus, 35 in GS**.

[39a] **T. Baca**, M. Platkevic, J. Jakubek, A. Inneman, V. Stehlikova, M. Urban, O. Nentvich, M. Blazek, R. McEntaffer, and V. Daniel, "Miniaturized X-ray telescope for VZLUSAT-1 nanosatellite with Timepix detector," *Journal of Instrumentation*, vol. 11, no. 10, p. C10007, 2016, **52% contribution, IF 1.454 (Q3 in Instruments & Instrumentation), citations: 39 in WoS, 40 in Scopus, 45 in GS**.

Partially-related conference articles

[40a] V. Daniel, A. Inneman, L. Pina, V. Zadrazil, **T. Baca**, V. Stehlikova, O. Nentvich, M. Urban, V. Marsikova, R. McEntaffer, J. Tutt, and T. Schulz, "X-ray Lobster Eye All-sky Monitor for Rocket Experiment," in *EUV and X-Ray Optics: Synergy Between Laboratory And Space V*, SPIE, vol. 10235, 2017, **10% contribution, citations: 11 in WoS, 16 in Scopus, 16 in GS**.

[41a] V. Daniel, L. Pina, A. Inneman, V. Zadrazil, **T. Baca**, M. Platkevic, V. Stehlikova, O. Nentvich, and M. Urban, "Terrestrial gamma-ray flashes monitor demonstrator on CubeSat," English, in *SPIE: CubeSats and NanoSats for Remote Sensing*, San Diego, California, United States, 2016, **11% contribution, citations: 17 in WoS, 19 in Scopus, 15 in GS**.

Partially-related publications — submitted

[42a] M. Urban, O. Nentvich, **T. Baca**, I. Vertat, V. Marsikova, D. Doubravova, V. Daniel, A. Inneman, J. Tutt, R. McEntaffer, and L. Pina, "REX: X-ray experiment on the Water Recovery Rocket," 2020, submitted to Acta Astronautica. eprint: `2011.10072`, **20% contribution**.

A.4 Unrelated author's publications

[43a] J. Chudoba, M. Saska, **T. Baca**, and L. Preucil, "Localization and stabilization of micro aerial vehicles based on visual features tracking," English, in *2014 IEEE International Conference on Unmanned Aircraft Systems*, vol. 1, IEEE, 2014, pp. 611–616, ISBN: 978-1-4799-2376-2, **16% contribution, citations: 2 in WoS, 7 in Scopus, 33 in GS**.

[44a] M. Saska, V. Vonasek, **T. Baca**, and L. Preucil, "Ad-hoc Heterogeneous (MAV-UGV) Formations Stabilized Under a Top-View Relative Localization," English, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Piscataway: IEEE, 2013, **25% contribution, citations: 0 in WoS, 0 in Scopus, 2 in GS**.

A.5 Cited references

[45] K. Ankit, L. A. Tony, S. Jana, and D. Ghose, "Multi-Agent Collaboration for Building Construction," *arXiv preprint arXiv:2009.03584*, 2020.

[46] S Baek and G York, "Optimal Sensor Management for Multiple Target Tracking Using Cooperative Unmanned Aerial Vehicles," in *2020 International Conference on Unmanned Aircraft Systems*, 2020, pp. 1294–1300.

[47] J. Ericksen, A. Aggarwal, G. M. Fricke, and M. E. Moses, "LOCUS: A multi-robot loss-tolerant algorithm for surveying volcanic plumes," in *2020 IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2020, pp. 113–120.

[48] K. Feng, W. Li, S. Ge, and F. Pan, "Packages delivery based on marker detection for UAVs," in *2020 IEEE Chinese Control And Decision Conference*, IEEE, 2020, pp. 2094–2099.

[49] Z. Fu, J. Jiang, Y. Gao, B. Krienke, M. Wang, K. Zhong, Q. Cao, Y. Tian, Y. Zhu, and W. Cao, "Wheat growth monitoring and yield estimation based on multi-rotor unmanned aerial vehicle," *Remote Sensing*, vol. 12, no. 3, p. 508, 2020.

[50] G. Ganci, A. Cappello, G. Bilotta, and C. Del Negro, "How the variety of satellite remote sensing data over volcanoes can assist hazard monitoring efforts: The 2011 eruption of Nabro volcano," *Remote Sensing of Environment*, vol. 236, p. 111 426, 2020.

[51] G. Jang, J. Kim, J.-K. Yu, H.-J. Kim, Y. Kim, D.-W. Kim, K.-H. Kim, C. W. Lee, and Y. S. Chung, "Cost-Effective Unmanned Aerial Vehicle (UAV) Platform for Field Plant Breeding Application," *Remote Sensing*, vol. 12, no. 6, p. 998, 2020.

[52] E. Karamuz, R. J. Romanowicz, and J. Doroszkiewicz, "The use of unmanned aerial vehicles in flood hazard assessment," *Journal of Flood Risk Management*, vol. 13, no. 4, e12622, 2020.

[53] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, "Cooperative Aerial-Ground Multi-Robot System for Automated Construction Tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 798–805, 2020.

[54] C. Lenz, M. Schwarz, A. Rochow, J. Razlaw, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Autonomous Wall Building with a UGV-UAV Team at MBZIRC 2020," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov. 2020.

[55] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A Distributed Control Framework of Multiple Unmanned Aerial Vehicles for Dynamic Wildfire Tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1537–1548, 2020.

[56] R. Suarez Fernandez, A. Rodriguez Ramos, A. Alvarez, J. Rodriguez-Vázquez, H. Bavle, L. Lu, M. Fernandez, A. Rodelgo, A. Cobano, D. Alejo, D. Acedo, R. Rey, S. Martinez-Rozas, M. Molina, L. Merino, F. Caballero, and P. Campoy, *The Skyeye Team Participation in the 2020 Mohamed Bin Zayed International Robotics Challenge*, misc, Online (accessed on 2020/09/05): https://www.researchgate.net/publication/339725858_The_Skyeye_Team_Participation_in_the_2020_Mohamed_Bin_Zayed_International_Robotics_Challenge, Feb. 2020.

[57] T. R. Walter, A. Belousov, M. Belousova, T. Kotenko, and A. Auer, "The 2019 Eruption Dynamics and Morphology at Ebeko Volcano Monitored by Unoccupied Aircraft Systems (UAS) and Field Stations," *Remote Sensing*, vol. 12, no. 12, p. 1961, 2020.

[58] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "XTDrone: A Customizable Multi-Rotor UAVs Simulation Platform," *arXiv preprint arXiv:2003.09700*, 2020.

[59] H. Zha, Y. Miao, T. Wang, Y. Li, J. Zhang, W. Sun, Z. Feng, and K. Kusnierek, "Improving unmanned aerial vehicle remote sensing-based rice nitrogen nutrition index prediction with machine learning," *Remote Sensing*, vol. 12, no. 2, p. 215, 2020.

[60] R. Bahnemann, M. Pantic, M. Popovic, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, and J. Nieto, "The ETH-MAV Team in the MBZ International Robotics Challenge," *Journal of Field Robotics*, vol. 36, no. 1, pp. 78–103, 2019.

[61] M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, and S. Behnke, "Team NimbRo at MBZIRC 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles," *Journal of Field Robotics*, vol. 36, no. 1, pp. 204–229, 2019.

[62] A. R. Castano, F. Real, P. Ramon-Soria, J. Capitan, V. Vega, B. C. Arrue, A. Torres-Gonzalez, and A. Ollero, "Al-Robotics team: A cooperative multi-unmanned aerial vehicle approach for the Mohamed Bin Zayed International Robotic Challenge," *Journal of Field Robotics*, vol. 36, no. 1, pp. 104–124, 2019.

[63] J. Dias, P. U. Lima, L. Seneviratne, O. Khatib, S. Tadokoro, and P. Dario, "Journal of Field Robotics special issue on MBZIRC 2017 Challenges in Autonomous Field Robotics," *Journal of Field Robotics*, vol. 36, no. 1, pp. 3–5, 2019.

[64] R. Filgas, M. Malich, T. Kuwahara, J. Broulím, M. Holík, M. Sakal, Y. Murata, H. Tomio, S. Gohl, and J. M. Pineda T., "RISEPix – A Timepix-based radiation monitor telescope onboard the RISESAT satellite," *Astronomische Nachrichten*, vol. 340, no. 7, pp. 674–680, 2019.

[65] D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, "Disturbance Estimation and Rejection for High-Precision Multirotor Position Control," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 2797–2804.

[66] R. Jin, H. M. Owais, D. Lin, T. Song, and Y. Yuan, "Ellipse proposal and convolutional neural network discriminant for autonomous landing marker detection," *Journal of Field Robotics*, vol. 36, no. 1, pp. 6–16, 2019.

[67] M. Labbadi and M. Cherkaoui, "Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV," *Aerospace Science and Technology*, vol. 93, p. 105 306, 2019.

[68] J. Lee, D. H. Shim, S. Cho, H. Shin, S. Jung, D. Lee, and J. Kang, "A mission management system for complex aerial logistics by multiple unmanned aerial vehicles in MBZIRC 2017," *Journal of Field Robotics*, vol. 36, no. 5, pp. 919–939, 2019.

[69] Z. Li, C. Meng, F. Zhou, X. Ding, X. Wang, H. Zhang, P. Guo, and X. Meng, "Fast vision-based autonomous detection of moving cooperative target for unmanned aerial vehicle landing," *Journal of Field Robotics*, vol. 36, no. 1, pp. 34–48, 2019. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21815`.

[70] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, 2019.

[71] I. B. Nascimento, A. Ferramosca, L. C. Piment, and G. V. Raffo, "NMPC Strategy for a Quadrotor UAV in a 3D Unknown Environment," in *2019 19th International Conference on Advanced Robotics (ICAR)*, IEEE, 2019, pp. 179–184.

[72] T. P. Nascimento and M. Saska, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Annual Reviews in Control*, vol. 48, pp. 129–146, 2019.

[73] C Papachristos, S Khattak, F Mascarich, and K Alexis, "Autonomous Navigation and Mapping in Underground Mines Using Aerial Robots," in *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–8.

[74] J. C. Pereira, V. J. Leite, and G. V. Raffo, "Nonlinear Model Predictive Control on SE(3) for Quadrotor Trajectory Tracking and Obstacle Avoidance," in *2019 19th International Conference on Advanced Robotics (ICAR)*, IEEE, 2019, pp. 155–160.

[75] L. S. Pinsky and S. Pospisil, "Timepix-based detectors in mixed-field charged-particle radiation dosimetry applications," *Radiation Measurements*, p. 106 229, 2019.

[76] Y. Sato, Y. Terasaka, W. Utsugi, H. Kikuchi, H. Kiyooka, and T. Torii, "Radiation imaging using a compact Compton camera mounted on a crawler robot inside reactor buildings of Fukushima Daiichi Nuclear Power Station," *Journal of Nuclear Science and Technology*, vol. 56, no. 9-10, pp. 801–808, 2019.

[77] P. Stepan, T. Krajnik, 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.

[78] M. Wages, S. V. Hull, A. D. Falcone, T. B. Anderson, M. McQuaide, E. Bray, T. Chattopadhyay, D. N. Burrows, L. Buntic, R. L. McEntaffer, D. M. Miles, J. H. Tutt, T. B. Schultz, B. D. Donovan, C. R. Hillman, and D. Yastishock, "Flight camera package design, calibration, and performance for the Water Recovery X-ray Rocket mission," in *UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XXI*, O. H. Siegmund, Ed., International Society for Optics and Photonics, vol. 11118, SPIE, 2019, pp. 72–88.

[79] J. Zhang, D. Gu, C. Deng, and B. Wen, "Robust and adaptive backstepping control for hexacopter uavs," *IEEE Access*, vol. 7, pp. 163 502–163 514, 2019.

[80] R. Ballabriga, M. Campbell, and X. Llopart, "Asic developments for radiation imaging applications: The medipix and timepix family," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 878, pp. 10–23, 2018.

[81] F. Di Traglia, S. Calvari, L. D'Auria, T. Nolesini, A. Bonaccorso, A. Fornaciai, A. Esposito, A. Cristaldi, M. Favalli, and N. Casagli, "The 2014 effusive eruption at stromboli: New insights from in situ and remote-sensing measurements," *Remote Sensing*, vol. 10, no. 12, p. 2035, 2018.

[82] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, "A survey of open-source uav flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, pp. 11–20, 2018, ISSN: 0141-9331.

[83] W. Furnell, A. Shenoy, E. Fox, *et al.*, "First results from the LUCID-Timepix spacecraft payload onboard the TechDemoSat-1 satellite in Low Earth Orbit," *Advances in Space Research*, 2018. eprint: `arXiv:1810.12876`.

[84] C. Granja, J. Jakubek, S. Polansky, V. Zach, P. Krist, D. Chvatil, J. Stursa, M. Sommer, O. Ploc, S. Kodaira, *et al.*, "Resolving power of pixel detector Timepix for wide-range electron, proton and ion detection," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 908, no. 1, pp. 60–71, 2018.

[85] A. Keatley, P. Martin, K. Hallam, O. Payton, R Awbery, F. Carvalho, J. Oliveira, L Silva, M Malta, and T. Scott, "Source identification of uranium-containing materials at mine legacy sites in Portugal," *Journal of Environmental Radioactivity*, vol. 183, pp. 102–111, 2018.

[86] R. Losch, S. Grehl, M. Donner, C. Buhl, and B. Jung, "Design of an autonomous robot for mapping, navigation, and manipulation in underground mines," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1407–1412.

[87] S. Manfreda, M. F. McCabe, P. E. Miller, R. Lucas, V. Pajuelo Madrigal, G. Mallinis, E. Ben Dor, D. Helman, L. Estes, G. Ciraolo, *et al.*, "On the use of unmanned aerial systems for environmental monitoring," *Remote sensing*, vol. 10, no. 4, p. 641, 2018.

[88] F. Mascarich, T. Wilson, C. Papachristos, and K. Alexis, "Radiation source localization in GPS-denied environments using aerial robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 6537–6544.

[89] K. Nagatani, S. Kiribayashi, R. Yajima, Y. Hada, T. Izu, A. Zeniya, H. Kanai, H. Kanasaki, J. Minagawa, and Y. Moriyama, "Micro-Unmanned Aerial Vehicle-based Volcano Observation System for Debris Flow Evacuation Warning," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1222–1241, 2018.

[90] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[91] O. Salek, M. Matolin, and L. Gryc, "Mapping of radiation anomalies using UAV mini-airborne gamma-ray spectrometry," *Journal of environmental radioactivity*, vol. 182, pp. 101–107, 2018.

[92] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "OpenUAV: a UAV testbed for the CPS and robotics community," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 130–139.

[93] D Turecek, J Jakubek, E Trojanova, and L Sefc, "Compton camera based on Timepix3 technology," *Journal of Instrumentation*, vol. 13, no. 11, p. C11022, 2018.

[94] P. Andreo, D. T. Burns, A. E. Nahum, J. Seuntjens, and F. H. Attix, *Fundamentals of ionizing radiation dosimetry.* John Wiley & Sons, 2017.

[95] S. Battiato, L. Cantelli, F. D'Urso, G. M. Farinella, L. Guarnera, D. Guastella, C. D. Melita, G. Muscato, A. Ortis, F. Ragusa, *et al.*, "A System for Autonomous Landing of a UAV on a Moving Vehicle," in *International Conference on Image Analysis and Processing*, Springer, 2017, pp. 129–139.

[96] M. Beul and S. Behnke, "Fast full state trajectory generation for multirotors," in *2017 IEEE International Conference on Unmanned Aircraft Systems*, IEEE, 2017, pp. 408–416.

[97] A. Borowczyk, D.-T. Nguyen, A. Phu-Van Nguyen, D. Q. Nguyen, D. Saussie, and J. 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.

[98] L Cantelli, D Guastella, C. Melita, G Muscato, S Battiato, F D'urso, G. Farinella, A Ortis, and C Santoro, "Autonomous landing of a UAV on a moving vehicle for the MBZIRC," in *20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, World Scientific, 2017.

[99] G. Christie, A. Shoemaker, K. Kochersberger, P. Tokekar, L. McLean, and A. Leonessa, "Radiation search operations using scene understanding with autonomous UAV and UGV," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1450–1468, 2017.

[100] A. Gawel, M. Kamel, T. Novkovic, J. Widauer, D. Schindler, B. P. Von Altishofen, R. Siegwart, and J. Nieto, "Aerial picking and delivery of magnetic objects with MAVs," in *2017 IEEE International Conference on Robotics and Automation*, IEEE, 2017, pp. 5746–5752.

[101] J. Ghommam and M. Saad, "Autonomous landing of a quadrotor on a moving platform," *Transactions on Aerospace and Electronic Systems*, vol. 53, pp. 1504–1519, 3 2017.

[102] T. Hoang, E. Bayasgalan, Z. Wang, G. Tsechpenakis, and D. Panagou, "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," in *American Control Conference (ACC)*, IEEE, 2017, pp. 5580–5585.

[103] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust Collision Avoidance for Multiple Micro Aerial Vehicles Using Nonlinear Model Predictive Control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2017, pp. 236–243.

[104] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System," in *Robot Operating System (ROS)*, Springer, 2017, pp. 3–39.

[105] S. Lin, M. A. Garratt, and A. 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.

[106] D. M. Miles, R. L. McEntaffer, T. B. Schultz, B. D. Donovan, J. H. Tutt, D. Yastishock, T. Steiner, C. R. Hillman, J. A. McCoy, M. Wages, S. Hull, A. Falcone, D. N. Burrows, T. Anderson, M. McQuaide, and T. Chattopadhyay, "An introduction to the water recovery x-ray rocket," in *UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XX*, O. H. Siegmund, Ed., SPIE, Aug. 2017, p. 28, ISBN: 9781510612518.

[107] R. Penicka, J. Faigl, P. Vana, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210–1217, 2017.

[108] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1039–1046, 2017.

[109] B. Zhu, A. H. B. Zaini, and L. Xie, "Distributed Guidance for Interception by Using Multiple Rotary-Wing Unmanned Aerial Vehicles," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5648–5656, 2017.

[110] A. Benini, M. J. Rutherford, and K. P. Valavanis, “Real-time, GPU-based pose estimation of a UAV for autonomous takeoff and landing,” in *2016 IEEE International Conference on Robotics and Automation*, IEEE, 2016, pp. 3463–3470.

[111] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, “RotorS — A modular gazebo MAV simulator framework,” in *Robot Operating System (ROS)*, Springer, 2016, pp. 595–625.

[112] K. A. Ghamry, Y. Dong, M. A. Kamel, and Y. Zhang, “Real-time autonomous take-off, tracking and landing of UAV on a moving UGV platform,” in *24th Mediterranean Conference on Control and Automation*, IEEE, 2016, pp. 1236–1241.

[113] C. Granja, S. Polansky, Z. Vykydal, *et al.*, “The SATRAM Timepix spacecraft payload in open space on board the Proba-V satellite for wide range radiation monitoring in LEO orbit,” *Planetary and Space Science*, vol. 125, pp. 114–129, 2016.

[114] J. Jiang, K. Shimazoe, Y. Nakamura, H. Takahashi, Y. Shikaze, Y. Nishizawa, M. Yoshida, Y. Sanada, T. Torii, M. Yoshino, *et al.*, “A prototype of aerial radiation monitoring system using an unmanned helicopter mounting a GAGG scintillator Compton camera,” *Journal of Nuclear Science and Technology*, vol. 53, no. 7, pp. 1067–1075, 2016.

[115] S. Jin, J. Zhang, L. Shen, and T. Li, “On-board vision autonomous landing techniques for quadrotor: A survey,” in *35th Chinese Control Conference (CCC)*, IEEE, 2016, pp. 10 284–10 289.

[116] W. Jung, Y. Kim, and H. Bang, “Target state estimation for vision-based landing on a moving ground target,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2016, pp. 657–663.

[117] P. G. Martin, S Kwong, N. Smith, Y Yamashiki, O. D. Payton, F. Russell-Pavier, J. S. Fardoulis, D. Richards, and T. B. Scott, “3D unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 12–19, 2016.

[118] A. Mirjan, F. Augugliaro, R. D’Andrea, F. Gramazio, and M. Kohler, “Building a bridge with flying robots,” in *Robotic Fabrication in Architecture, Art and Design 2016*, Springer, 2016, pp. 34–47.

[119] A. A. R. Newaz, S. Jeong, H. Lee, H. Ryu, and N. Y. Chong, “UAV-based multiple source localization and contour mapping of radiation fields,” *Robotics and Autonomous Systems*, vol. 85, pp. 12–25, 2016.

[120] C. E. Parcheta, C. A. Pavlov, N. Wiltsie, K. C. Carpenter, J. Nash, A. Parness, and K. L. Mitchell, “A robotic approach to mapping post-eruptive volcanic fissure conduits,” *Journal of Volcanology and Geothermal Research*, vol. 320, pp. 19–28, 2016.

[121] M. T. Perks, A. J. Russell, and A. R. Large, “Advances in flash flood monitoring using unmanned aerial vehicles (UAVs),” *Hydrology and Earth System Sciences*, vol. 20, no. 10, pp. 4005–4015, 2016.

[122] J. L. Sanchez-Lopez, R. A. Suarez Fernandez, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy, “AEROSTACK: An architecture and open-source software framework for aerial robotics,” in *2016 IEEE International Conference on Unmanned Aircraft Systems*, 2016, pp. 332–341.

[123] J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, and P. Campoy, “A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-UAV systems: Simulation and experimentation,” *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 779–797, 2016.

[124] D. Abeywardena, P. Pounds, D. Hunt, and G. Dissanayake, “Design and development of recopter: An open source ros-based multi-rotor platform for research,” in *Australasian Conference on Robotics and Automation (ACRA)*, 2015.

[125] M. Burri, M. Datwiler, M. W. Achtelik, and R. Siegwart, "Robust state estimation for micro aerial vehicles based on system dynamics," in *2015 IEEE international conference on robotics and automation*, IEEE, 2015, pp. 5278–5283.

[126] N. Farmani, L. Sun, and D. Pack, "Tracking multiple mobile targets using cooperative unmanned aerial vehicles," in *2015 IEEE International Conference on Unmanned Aircraft Systems*, IEEE, 2015, pp. 395–400.

[127] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 9, 2015.

[128] V. Grabe, H. H. Bulthoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.

[129] Y. Jung, D. Lee, and H. Bang, "Close-range vision navigation and guidance for rotary UAV autonomous landing," in *IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, 2015, pp. 342–347.

[130] P. Martin, O. Payton, J. Fardoulis, D. Richards, and T. Scott, "The use of unmanned aerial systems for the mapping of legacy uranium mines," *Journal of environmental radioactivity*, vol. 143, pp. 135–140, 2015.

[131] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation*, IEEE, 2015, pp. 6235–6240.

[132] G. Pajares, "Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–329, 2015.

[133] Y. Sanada and T. Torii, "Aerial Radiation Monitoring Around the Fukushima Dai-ichi Nuclear Power Plant Using an Unmanned Helicopter," *Journal of environmental radioactivity*, vol. 139, pp. 294–299, 2015.

[134] N. Stoffle, L. Pinsky, M. Kroupa, S. Hoang, J. Idarraga, C. Amberboy, R. Rios, J. Hauss, J. Keller, A. Bahadori, *et al.*, "Timepix-based radiation environment monitor measurements aboard the International Space Station," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 782, pp. 143–148, 2015.

[135] D Turecek and J Jakubek, *PIXET Software package tool for control, readout and online display of pixel detectors Medipix/Timepix*, misc, 2015.

[136] C. Wang, F. Ma, J. Yan, D. De, and S. K. Das, "Efficient aerial data collection with uav in large-scale wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 286 080, 2015.

[137] C. Yuan, Y. Zhang, and Z. Liu, "A survey on Technologies for Automatic Forest Fire Monitoring, Detection, and Fighting Using Unmanned Aerial Vehicles and Remote Sensing Techniques," *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.

[138] D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero, "Collision-Free 4D Trajectory Planning in Unmanned Aerial Vehicles for Assembly and Structure Construction," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 783–795, 2014.

[139] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.

[140] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, 2014.

[141] C. Granja, S. Polansky, Z Vykydal, *et al.*, "Directional visualization of space radiation quanta with Timepix/SATRAM spacecraft payload on board ESA Proba-V satellite," in *Proceedings X Latin American Symposium on Nuclear Physics and Applications. Trieste: SISSA/ISAS*, 2014, pp. 1–10.

[142] H. Kinoshita, R. Tayama, E. Y. Kometani, T. Asano, and Y Kani, "Development of new technology for Fukushima Daiichi nuclear power station reconstruction," *Hitachi Review*, vol. 63, no. 4, pp. 183–190, 2014.

[143] 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*, 2014, ISSN: 0921-0296.

[144] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-UAV system," in *IEEE American Control Conference (ACC)*, IEEE, 2014, pp. 2077–2082.

[145] J. MacFarlane, O. Payton, A. Keatley, G. Scott, H Pullin, R. Crane, M Smilion, I Popescu, V Curlea, and T. Scott, "Lightweight aerial vehicles for monitoring, assessment and mapping of radiation anomalies," *Journal of environmental radioactivity*, vol. 136, pp. 127–130, 2014.

[146] M. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," 2014.

[147] T Poikela, J Plosila, T Westerlund, M Campbell, M De Gaspari, X Llopart, V Gromov, R Kluit, M van Beuzekom, F Zappon, *et al.*, "Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout," *Journal of instrumentation*, vol. 9, no. 05, p. C05013, 2014.

[148] L. Sun, S. Baek, and D. Pack, "Distributed probabilistic search and tracking of agile mobile ground targets using a network of unmanned aerial vehicles," in *Human Behavior Understanding in Networked Sensing*, Springer, 2014, pp. 301–319.

[149] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *2014 IEEE International Conference on Robotics and Automation*, IEEE, 2014, pp. 2113–2118.

[150] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 401–410, 2013.

[151] C. Hui, C. Yousheng, L. Xiaokun, and W. W. Shing, "Autonomous takeoff, tracking and landing of a UAV on a moving UGV using onboard monocular vision," in *Chinese Control Conference (CCC)*, IEEE, 2013, pp. 5895–5901.

[152] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.

[153] Q. Lindsey and V. Kumar, "Distributed construction of truss structures," in *Algorithmic Foundations of Robotics X*, Springer, 2013, pp. 209–225.

[154] I. Mellado-Bataller, J. Pestana, M. A. Olivares-Mendez, P. Campoy, and L. Mejias, "MAVwork: a framework for unified interfacing between micro aerial vehicles and visual controllers," in *Frontiers of Intelligent Autonomous Systems*, Springer, 2013, pp. 165–179.

[155] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[156] H. Rezaee and F. Abdollahi, "Motion synchronization in unmanned aircrafts formation control with communication delays," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 744–756, 2013.

[157] J. Baluja, M. P. Diago, P. Balda, R. Zorer, F. Meggio, F. Morales, and J. Tardaguila, "Assessment of vineyard water status variability by thermal and multispectral imagery using an unmanned aerial vehicle (UAV)," *Irrigation Science*, vol. 30, no. 6, pp. 511–522, 2012.

[158] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[159] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2012, pp. 971–976.

[160] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[161] J. Towler, B. Krawiec, and K. Kochersberger, "Radiation mapping in post-disaster environments using an autonomous helicopter," *Remote Sensing*, vol. 4, no. 7, pp. 1995–2015, 2012.

[162] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[163] A. Burkle, F. Segor, and M. Kollmann, "Towards autonomous micro UAV swarms," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 339–353, 2011.

[164] "Development of pattern recognition software for tracks of ionizing radiation in medipix2-based (Timepix) pixel detector devices," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 331, 2011, p. 032 052.

[165] K. Ohno, S. Kawatsuma, T. Okada, E. Takeuchi, K. Higashi, and S. Tadokoro, "Robotic control vehicle for measuring radiation in Fukushima Daiichi Nuclear Power Plant," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, IEEE, 2011, pp. 38–43.

[166] D Turecek, T Holy, J Jakubek, S Pospisil, and Z Vykydal, "Pixelman: a multi-platform data acquisition and processing software package for Medipix2, Timepix and Medipix3 detectors," *Journal of Instrumentation*, vol. 6, no. 01, p. C01046, 2011.

[167] D Turecek, L Pinsky, J Jakubek, Z Vykydal, N Stoffle, and S Pospisil, "Small dosimeter based on Timepix device for International Space Station," *Journal of Instrumentation*, vol. 6, no. 12, p. C12037, 2011.

[168] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *2010 IEEE Conference on Decision and Control*, IEEE, 2010, pp. 5420–5425.

[169] W. Atwood, A. A. Abdo, M. Ackermann, W Althouse, B Anderson, M Axelsson, L. Baldini, J Ballet, D. Band, G. Barbiellini, *et al.*, "The large area telescope on the Fermi gamma-ray space telescope mission," *The Astrophysical Journal*, vol. 697, no. 2, p. 1071, 2009.

[170] J. A. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres, "Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring from an Unmanned Aerial Vehicle," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722–738, 2009.

[171] J. Capitan, L. Merino, F. Caballero, and A. Ollero, "Delayed-state information filter for cooperative decentralized tracking," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 3865–3870.

[172] X. Llopart, R. Ballabriga, M. Campbell, *et al.*, "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements," *Nuclear Instruments and Methods*, vol. 581, no. 1–2, pp. 485–494, 2007.

[173] L. Merino, F. Caballero, J. Ferruz, J. Wiklund, P.-E. Forssen, and A. Ollero, "Multi-UAV cooperative perception techniques," in *Multiple Heterogeneous Unmanned Aerial Vehicles*, Springer, 2007, pp. 67–110.

[174] T. Holy, J. Jakubek, S. Pospisil, J. Uher, D. Vavrik, and Z. Vykydal, "Data acquisition and processing software package for Medipix2," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 563, no. 1, pp. 254–258, 2006.

[175] L. Merino, J. Wiklund, F. Caballero, A. Moe, J. R. M. De Dios, P.-E. Forssen, K. Nordberg, and A. Ollero, “Vision-based multi-UAV position estimation,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 53–62, 2006.

[176] N. Gehrels, G. Chincarini, P Giommi, K. Mason, J. A. Nousek, A. Wells, N. White, S. Barthelmy, D. N. Burrows, L. Cominsky, *et al.*, “The Swift gamma-ray burst mission,” *The Astrophysical Journal*, vol. 611, no. 2, p. 1005, 2004.

[177] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium, IEEE*, IEEE, 2000, pp. 153–158.

[178] M. C. Weisskopf, H. D. Tananbaum, L. P. Van Speybroeck, and S. L. O’Dell, “Chandra X-ray Observatory (CXO): overview,” in *X-Ray Optics, Instruments, and Missions III*, International Society for Optics and Photonics, vol. 4012, 2000, pp. 2–16.

Chapter B

Citations of Author's Publications

Citations of the author's work were extracted from the Web of Science. First- and second-order self-citations are excluded. The data were gathered on January 1$^{\text{st}}$, 2021.

[16a] M. Saska, **T. Baca**, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, *et al.*, "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

- J. Dunn and R. Tron, "Temporal Siamese Networks for Clutter Mitigation Applied To Vision-based Quadcopter Formation Control," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, 32–39, Jan. 2021.
- M. Amir and A. M. Bruckstein, "Fast Uniform Dispersion of A Crash-prone Swarm," in *Robotics: Science and Systems XVI*, 2020.
- C. Chen, Y. Tian, L. Lin, S. Chen, H. Li, Y. Wang, and K. Su, "Obtaining World Coordinate Information of UAV In GNSS Denied Environments," *Sensors*, vol. 20, no. 8, Jan. 2020.
- M. Chen, Z. Xiong, J. Liu, R. Wang, and J. Xiong, "Cooperative Navigation of Unmanned Aerial Vehicle Swarm Based On Cooperative Dilution of Precision," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, Jan. 2020.
- M. Coppola, K. N. Mcguire, C. De Wagter, and G. C. H. E. De Croon, "A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges And Constraints," *Frontiers In Robotics and Ai*, vol. 7, Jan. 2020.
- K. Guo, X. Li, and L. Xie, "Ultra-wideband and Odometry-based Cooperative Relative Localization With Application To Multi-UAV Formation Control," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, 2590–2603, Jan. 2020.
- H. Isakhani, N. Aouf, O. Kechagias-stamatis, and J. F. Whidborne, "A Furcated Visual Collision Avoidance System for An Autonomous Microrobot," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 1, 1–11, Jan. 2020.
- Y. Li, "Design of Path Tracking Control System for UAV Based on Adaptive Preview Method," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 14, no. 1, Si, 101–108, Jan. 2020.
- P. Moreno, S. Esteva, I. Mas, and J. Giribet I, "Multi-UAV Specification and Control With A Single Pilot-in-the-loop," *Unmanned Systems*, vol. 8, no. 4, Si, 269–277, Jan. 2020.
- X. Niu, X. Yuan, Y. Zhou, and H. Fan, "UAV Track Planning Based on Evolution Algorithm In Embedded System," *Microprocessors and Microsystems*, vol. 75, Jan. 2020.
- F. She, Y. Zhang, D. Shi, H. Zhou, X. Ren, and T. Xu, "Enhanced Relative Localization Based on Persistent Excitation For Multi-UAVs In GPS-denied Environments," *IEEE Access*, vol. 8, 148136–148148, 2020.
- J. Tao, Y. Jia, and Y. Gao, "Intelligent Remote Monitoring System for Minor Faults of Intelligent Unmanned Vehicle," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 14, no. 1, Si, 61–70, Jan. 2020.
- X. Tian, G. Wei, L. Wang, and J. Zhou, "Wireless -Sensor -Network -Based Target Localization: A Semidefinite Relaxation Approach With Adaptive Threshold Correction," *Neurocomputing*, vol. 405, 229–238, Jan. 2020.

- A. H. Zaini and L. Xie, "Distributed Drone Traffic Coordination Using Triggered Communication," *Unmanned Systems*, vol. 8, no. 1, 1–20, Jan. 2020.
- M. Amir and A. M. Bruckstein, "Minimizing Travel In the Uniform Dispersal Problem for Robotic Sensors," in *AAMAS '19: Proceedings of the 18th International Conference On Autonomous Agents and Multiagent Systems*, 2019, 113–121.
- M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, and S. Behnke, "Team Nimbro At MBZIRC 2017: Fast Landing on A Moving Target and Treasure Hunting With A Team of Micro Aerial Vehicles," *Journal of Field Robotics*, vol. 36, no. 1, Si, 204–229, Jan. 2019.
- Z. Chen, C. Jiang, and Y. Guo, "Distance-based Formation Control of A Three-robot System," in *Proceedings of the 2019 31st Chinese Control and Decision Conference (CCDC 2019)*, 2019, 5501–5507.
- D. Erez, S. Arogeti, and D. Zarrouk, "A Novel Simple Two-robot Precise Self-localization Method," *IEEE Access*, vol. 7, 154044–154055, 2019.
- C. Liu and W. Hu, "Real-time Geometric Fitting and Pose Estimation for Surface Of Revolution," *Pattern Recognition*, vol. 85, 90–108, Jan. 2019.
- A. Lopez-martinez and F. J. Cuevas, "Automatic Circle Detection on Images Using the Teaching Learning Based Optimization Algorithm and Gradient Analysis," *Applied Intelligence*, vol. 49, no. 5, 2001–2016, Jan. 2019.
- P. Martinez and M. Barczyk, "Implementation and Optimization of the Cascade Classifier Algorithm For UAV Detection and Tracking," *Journal of Unmanned Vehicle Systems*, vol. 7, no. 4, 296–311, Jan. 2019.
- S. Nilwong, D. Hossain, S.-i. Kaneko, and G. Capi, "Deep Learning-based Landmark Detection for Mobile Robot Outdoor Localization," *Machines*, vol. 7, no. 2, Jan. 2019.
- F. Schilling, J. Lecoeur, F. Schiano, and D. Floreano, "Learning Vision-based Flight In Drone Swarms By Imitation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, 4523–4530, Jan. 2019.
- D. Shen and Q. Lu, "Hierarchical Formation Control With Applications To Multi-quadrotor Systems," *IEEE Access*, vol. 7, 130599–130609, 2019.
- E. Soria, F. Schiano, and D. Floreano, "The Influence of Limited Visual Sensing on the Reynolds Flocking Algorithm," in *2019 Third IEEE International Conference on Robotic Computing (IRC 2019)*, 2019, 138–145.
- X. Tian, G. Wei, J. Wang, and D. Zhang, "A Localization and Tracking Approach In Nlos Environment Based On Distance and Angle Probability Model," *Sensors*, vol. 19, no. 20, Jan. 2019.
- Z. Wang and T. Liu, "Coordinated Formation Control of Wheeled Mobile Robots With Switching Communication Topologies," *Iet Control Theory and Applications*, vol. 13, no. 18, 3164–3173, Jan. 2019.
- A. Zhang, D. Zhou, M. Yang, and P. Yang, "Finite-time Formation Control for Unmanned Aerial Vehicle Swarm System With Time-delay and Input Saturation," *IEEE Access*, vol. 7, 5853–5864, 2019.
- J. I. Giribet, I. Mas, and P. Moreno, "Vision-based Integrated Navigation System and Optimal Allocation In Formation Flying," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, 52–61.
- A. Kohlbacher, J. Eliasson, K. Acres, H. Chung, and J. C. Barca, "A Low Cost Omnidirectional Relative Localization Sensor for Swarm Applications," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, 694–699.
- K. Maeda, Y. Funabora, S. Doki, and K. Doki, "Flight Path Planning of Multiple UAVs for Robust Localization Near Infrastructure Facilities," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, 2522–2527.
- F. Poiesi and A. Cavallaro, "A Distributed Vision-based Consensus Model for Aerial-robotic Teams," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, 169–176.
- S. Suzuki, "Recent Researches on Innovative Drone Technologies In Robotics Field," *Advanced Robotics*, vol. 32, no. 19, Si, 1008–1022, Jan. 2018.

- O. Shrit, S. Martin, K. Alagha, and G. Pujolle, "A New Approach To Realize Drone Swarm Using Ad-hoc Network," in *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2017.

[14a] G. Loianno, V. Spurny, J. Thomas, **T. Baca**, D. Thakur, D. Hert, *et al.*, "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

- S. Liu, W. Dong, Z. Ma, and X. Sheng, "Adaptive Aerial Grasping and Perching With Dual Elasticity Combined Suction Cup," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, 4766–4773, Jan. 2020.
- M. Lopez Garcia and J. Martinez Carranza, "A First CNN-based Approach Towards Autonomous Flight for Object Lifting," *Computacion Y Sistemas*, vol. 24, no. 3, 1219–1228, 2020.
- G. Michieletto, A. Cenedese, L. Zaccarian, and A. Franchi, "Hierarchical Nonlinear Control for Multi-rotor Asymptotic Stabilization Based on Zero-moment Direction," *Automatica*, vol. 117, Jan. 2020.
- S. Patel, A. Sarabakha, D. Kircali, and E. Kayacan, "An Intelligent Hybrid Artificial Neural Network-based Approach For Control of Aerial Robots," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 2, 387–398, Jan. 2020.
- R. C. Shit, "Precise Localization for Achieving Next-generation Autonomous Navigation: State-of-the-art, Taxonomy and Future Prospects," *Computer Communications*, vol. 160, 351–374, Jan. 2020.
- D. K. D. Villa, A. S. Brandao, and M. Sarcinelli-filho, "A Survey on Load Transportation Using Multirotor UAVs," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, 267–296, Jan. 2020.
- R. Baehnemann, M. Pantie, M. Popovic, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, and J. Nieto, "The ETH-MAV Team In the MBZ International Robotics Challenge," *Journal of Field Robotics*, vol. 36, no. 1, Si, 78–103, Jan. 2019.
- I. H. Beloti Pizetta, A. S. Brandao, and M. Sarcinelli-filho, "Avoiding Obstacles In Cooperative Load Transportation," *Isa Transactions*, vol. 91, 253–261, Jan. 2019.
- I. H. Beloti Pizetta, A. Santos Brandao, and M. Sarcinelli-filho, "Cooperative Load Transportation Using Three Quadrotors," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS' 19)*, 2019, 644–650.
- M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, and S. Behnke, "Team Nimbro At MBZIRC 2017: Fast Landing on A Moving Target and Treasure Hunting With A Team of Micro Aerial Vehicles," *Journal of Field Robotics*, vol. 36, no. 1, Si, 204–229, Jan. 2019.
- S. Hamaza, I. Georgilas, M. Fernandez, P. Sanchez, T. Richardson, G. Heredia, and A. Ollero, "Sensor Installation and Retrieval Operations Using An Unmanned Aerial Manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, 2793–2800, Jan. 2019.
- A. Sarabakha and E. Kayacan, "Online Deep Learning for Improved Trajectory Tracking of Unmanned Aerial Vehicles Using Expert Knowledge," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, 7727–7733.
- D. Sartori and W. Yu, "Experimental Characterization of A Propulsion System for Multi-rotor UAVs," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3-4, 529–540, Jan. 2019.
- H. Zhou, J. Lynch, and D. Zekkos, "Vision-based Precision Localization of UAVs for Sensor Payload Placement and Pickup for Field Monitoring Applications," in *Sensors and Smart Structures Technologies for Civil, Mechanical, And Aerospace Systems 2019*, vol. 10970, 2019.
- D. K. D. Villa, A. S. Brandao, and M. Sarcinelli-filho, "Load Transportation Using Quadrotors: A Survey of Experimental Results," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, 84–93.
- Y. Wu, J. Song, J. Sun, F. Zhu, and H. Chen, "Aerial Grasping Based on VR Perception and Haptic Control," in *Proceedings of 2018 IEEE International Conference on Real-time Computing and Robotics (IEEE RCAR)*, 2018, 556–562.

[39a] **T. Baca**, M. Platkevic, J. Jakubek, A. Inneman, V. Stehlikova, M. Urban, *et al.*, "Miniaturized X-ray telescope for VZLUSAT-1 nanosatellite with Timepix detector," *Journal of Instrumentation*, vol. 11, no. 10, p. C10007, 2016

- O. Urban, O. Vavroch, L. Polacek, V. Georgiev, P. Burian, P. Turjanica, P. Fiala, P. Broulim, and B. Bergmann, "Hodoscope With Timepix Detectors for Pilsencube2 Cubesat," *Nuclear Instruments & Methods In Physics Research Section Accelerators Spectrometers Detectors and Associated Equipment*, vol. 980, Jan. 2020.
- R. Filgas, M. Malich, T. Kuwahara, J. Broulim, M. Holik, M. Sakal, Y. Murata, H. Tomio, S. Gohl, and J. M. T. Pineda, "Risepix — A Timepix-based Radiation Monitor Telescope Onboard the Risesat Satellite," *Astronomische Nachrichten*, vol. 340, no. 7, Si, 674–680, Jan. 2019.
- W. Furnell, A. Shenoy, E. Fox, and P. Hatfield, "First Results From the Lucid-timepix Spacecraft Payload Onboard The Techdemosat-1 Satellite In Low Earth Orbit," *Advances In Space Research*, vol. 63, no. 5, 1523–1540, Jan. 2019.
- P. Hatfield, W. Furnell, A. Shenoy, E. Fox, B. Parker, L. Thomas, and E. A. C. Rushton, "Iris Opens Pupils' Eyes To Real Space Research," *Astronomy & Geophysics*, vol. 60, no. 1, 22–24, Jan. 2019.
- R. Hudec, "X/EUV and UV Optics for Miniature Cube Satellites Payloads," in *EUV and X-Ray Optics: Synergy Between Laboratory and Space VI*, vol. 11032, 2019.
- V. Tichy, M. Barbera, R. Hudec, and R. Willingale, "Effective Collecting Area of Lobster Eye Optics and Optimal Value Of Effective Angle," *Experimental Astronomy*, vol. 47, no. 1-2, 161–175, Jan. 2019.
- R. Filgas, "Space Radiation Monitoring With Timepix," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 386–390, Jan. 2018.
- P. Hatfield, W. Furnell, A. Shenoy, E. Fox, R. Parker, and L. Thomas, "The Lucid-timepix Spacecraft Payload and the CERN School Educational Programme," *Journal of Instrumentation*, vol. 13, Jan. 2018.
- R. Hudec, "Axro Introduction and Historical Background," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 48, no. 3, 396–404, 2018.
- ——, "Low-dispersion Spectroscopy With Cubesats and Photographic Plates," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 416–419, Jan. 2018.
- V. Tichy and R. Willingale, "Optimization of Mirror Spacing Or Pore Width of Lobster Eye Optics," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 363–366, Jan. 2018.
- R. Hudec, "Astrophysical Payloads for Picosatellites," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 143–150, 2017.
- R. Hudec and S. Michalova, "Fish Eye Optics," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 94–99, 2017.
- R. Hudec and K. Remisova, "Biomimetics and Astronomical X-Ray Optics," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 67–75, 2017.
- V. Simon, "Perspectives of the Lobster-eye Telescope: the Promising Types of Cosmic X-Ray Sources," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 184–191, 2017.
- I. Vertat, R. Linhart, J. Masopust, A. Vobornik, and L. Dudacek, "Earth's Thermal Radiation Sensors for Attitude Determination Systems Of Small Satellites," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 157–164, 2017.

[2c] V. Spurny, **T. Baca**, M. Saska, R. Penicka, T. Krajnik, J. Thomas, *et al.*, "Cooperative Autonomous Search, Grasping and Delivering in a Treasure Hunt Scenario by a Team of UAVs," *Journal of Field Robotics*, vol. 36, no. 1, 125–148, 2019

- M. Coppola, K. N. Mcguire, C. De Wagter, and G. C. H. E. De Croon, "A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges And Constraints," *Frontiers In Robotics and Ai*, vol. 7, Jan. 2020.

- B. E. Jackson, T. A. Howell, K. Shah, M. Schwager, and Z. Manchester, "Scalable Cooperative Transport of Cable-suspended Loads With UAVs Using Distributed Trajectory Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 3368–3374, Jan. 2020.
- M. Lopez Garcia and J. Martinez Carranza, "A First CNN-based Approach Towards Autonomous Flight for Object Lifting," *Computacion Y Sistemas*, vol. 24, no. 3, 1219–1228, 2020.
- G. Michieletto, A. Cenedese, L. Zaccarian, and A. Franchi, "Hierarchical Nonlinear Control for Multi-rotor Asymptotic Stabilization Based on Zero-moment Direction," *Automatica*, vol. 117, Jan. 2020.
- A. Mohiuddin, T. Tarek, Y. Zweiri, and D. Gan, "A Survey of Single and Multi-UAV Aerial Manipulation," *Unmanned Systems*, vol. 8, no. 2, 119–147, Jan. 2020.
- D. Sanalitro, H. J. Savino, M. Tognon, J. Cortes, and A. Franchi, "Full-pose Manipulation Control of A Cable-suspended Load With Multiple UAVs Under Uncertainties," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2185–2191, Jan. 2020.
- F. Guo, M. Wei, M. Ye, J. Li, O. Mechali, and Y. Cao, "An Unmanned Aerial Vehicles Collaborative Searching and Tracking Scheme In Three-dimension Space," in *2019 9th IEEE Annual International Conference on Cyber Technology In Automation, Control, and Intelligent Systems (IEEE-cyber 2019)*, 2019, 1262–1266.
- A. Mohiuddin, T. Taha, Y. Zweiri, and D. Gan, "UAV Payload Transportation Via RTDP Based Optimized Velocity Profiles," *Energies*, vol. 12, no. 16, Jan. 2019.
- L. Campos-macias, R. Aldana-lopez, R. De La Guardia, J. I. Parra-vilchis, and D. Gomez-gutierrez, "Autonomous Navigation of MAVs In Unknown Cluttered Environments," *Journal of Field Robotics*,

[41a] V. Daniel, L. Pina, A. Inneman, V. Zadrazil, **T. Baca**, M. Platkevic, *et al.*, "Terrestrial gamma-ray flashes monitor demonstrator on CubeSat," English, in *SPIE: CubeSats and NanoSats for Remote Sensing*, San Diego, California, United States, 2016

- A. Alanazi and J. Straub, "Engineering Methodology for Student-driven Cubesats," *Aerospace*, vol. 6, no. 5, Jan. 2019.
- W. Furnell, A. Shenoy, E. Fox, and P. Hatfield, "First Results From the Lucid-timepix Spacecraft Payload Onboard The Techdemosat-1 Satellite In Low Earth Orbit," *Advances In Space Research*, vol. 63, no. 5, 1523–1540, Jan. 2019.
- R. Hudec, "X/EUV and UV Optics for Miniature Cube Satellites Payloads," in *EUV and X-Ray Optics: Synergy Between Laboratory and Space VI*, vol. 11032, 2019.
- P. Hatfield, W. Furnell, A. Shenoy, E. Fox, R. Parker, and L. Thomas, "The Lucid-timepix Spacecraft Payload and the CERN School Educational Programme," *Journal of Instrumentation*, vol. 13, Jan. 2018.
- R. Hudec, "Axro Introduction and Historical Background," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 48, no. 3, 396–404, 2018.
- ——, "Astrophysical Payloads for Picosatellites," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 143–150, 2017.
- R. Hudec and S. Michalova, "Fish Eye Optics," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 94–99, 2017.
- R. Hudec and K. Remisova, "Biomimetics and Astronomical X-Ray Optics," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 67–75, 2017.

[6c] **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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2018, pp. 1–8

- B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV Assistance Paradigm: State-of-the-art In Applications and Challenges," *Journal of Network and Computer Applications*, vol. 166, Jan. 2020.
- S. H. Arul and D. Manocha, "DCAD: Decentralized Collision Avoidance With Dynamics Constraints For Agile Quadrotor Swarms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 1191–1198, Jan. 2020.
- J. Hu, H. Zhang, L. Liu, X. Zhu, C. Zhao, and Q. Pan, "Convergent Multiagent Formation Control With Collision Avoidance," *IEEE Transactions on Robotics*, vol. 36, no. 6, 1805–1818, Jan. 2020.
- S. S. Mansouri, C. Kanellakis, B. Lindqvist, F. Pourkamali-anaraki, A.-a. Agha-mohammadi, J. Burdick, and G. Nikolakopoulos, "A Unified NMPC Scheme for MAVs Navigation With 3d Collision Avoidance Under Position Uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, 5740–5747, Jan. 2020.
- J. Bayer and J. Faigl, "Localization Fusion for Aerial Vehicles In Partially GNSS Denied Environments," in *Modelling and Simulation for Autonomous Systems (MESAS 2018)*, vol. 11472, 2019, 251–262.
- J. Faigl, P. Vana, and R. Penicka, "Multi-vehicle Close Enough Orienteering Problem With Bezier Curves For Multi-rotor Aerial Vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, 3039–3044.
- Y. Li, X. Chen, and N. Li, "Online Optimal Control With Linear Dynamics and Predictions: Algorithms and Regret Analysis," in *Advances In Neural Information Processing Systems 32 (NIPS 2019)*, vol. 32, 2019.

[21a] **T. Baca**, P. Stepan, and M. Saska, "Autonomous Landing On A Moving Car With Unmanned Aerial Vehicle," in *IEEE European Conference on Mobile Robotics (ECMR)*, IEEE, 2017

- D. Cazzato, C. Cimarelli, J. L. Sanchez-Lopez, H. Voos, and M. Leo, "A Survey of Computer Vision Methods for 2d Object Detection From Unmanned Aerial Vehicles," *Journal of Imaging*, vol. 6, no. 8, Jan. 2020.
- M. Galimov, R. Fedorenko, and A. Klimchik, "UAV Positioning Mechanisms In Landing Stations: Classification And Engineering Design Review," *Sensors*, vol. 20, no. 13, Jan. 2020.
- R. Polvara, M. Patacchiola, M. Hanheide, and G. Neumann, "Sim-to-real Quadrotor Landing Via Sequential Deep Q-networks and Domain Randomization," *Robotics*, vol. 9, no. 1, Jan. 2020.
- R. Baehnemann, M. Pantie, M. Popovic, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, and J. Nieto, "The ETH-MAV Team In the MBZ International Robotics Challenge," *Journal of Field Robotics*, vol. 36, no. 1, Si, 78–103, Jan. 2019.
- M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, and S. Behnke, "Team Nimbro At MBZIRC 2017: Fast Landing on A Moving Target and Treasure Hunting With A Team of Micro Aerial Vehicles," *Journal of Field Robotics*, vol. 36, no. 1, Si, 204–229, Jan. 2019.
- M. Bhargavapuri, A. K. Shastry, H. Sinha, S. R. Sahoo, and M. Kothari, "Vision-based Autonomous Tracking and Landing of A Fully-actuated Rotorcraft," *Control Engineering Practice*, vol. 89, 113–129, Jan. 2019.
- S. Lee, T. Shim, S. Kim, J. Park, K. Hong, and H. Bang, "Vision-based Autonomous Landing of A Multi-copter Unmanned Aerial Vehicle Using Reinforcement Learning," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, 108–114.

[3c] **T. Baca**, P. Stepan, B. Spurny, D. Hert, R. Penicka, M. Saska, *et al.*, "Autonomous Landing on a Moving Vehicle with an Unmanned Aerial Vehicle," *Journal of Field Robotics*, vol. 36, pp. 874–891, 5 2019

- D. Horla and J. Cieslak, "On Obtaining Energy-optimal Trajectories for Landing of UAVs," *Energies*, vol. 13, no. 8, Jan. 2020.

- J. Xie, X. Peng, H. Wang, W. Niu, and X. Zheng, "UAV Autonomous Tracking and Landing Based on Deep Reinforcement Learning Strategy," *Sensors*, vol. 20, no. 19, Jan. 2020.
- J. Goslinski, W. Giernacki, and A. Krolikowski, "A Nonlinear Filter for Efficient Attitude Estimation of Unmanned Aerial Vehicle (UAV)," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3-4, 1079–1095, Jan. 2019.
- K. D. Nguyen and T. thang Nguyen, "Vision-based Software-in-the-loop-simulation for Unmanned Aerial Vehicles Using Gazebo and PX4 Open Source," in *Proceedings of 2019 International Conference on System Science And Engineering (ICSSE)*, 2019, 429–432.
- C. Potena, D. Nardi, and A. Pretto, "Joint Vision-based Navigation, Control and Obstacle Avoidance for UAVs In Dynamic Environments," in *2019 European Conference on Mobile Robots (ECMR)*, 2019.
- J. Wubben, F. Fabra, C. T. Calafate, T. Krzeszowski, J. M. Marquez-barja, J.-c. Cano, and P. Manzoni, "Accurate Landing of Unmanned Aerial Vehicles Using Ground Pattern Recognition," *Electronics*, vol. 8, no. 12, Jan. 2019.

[40a] V. Daniel, A. Inneman, L. Pina, V. Zadrazil, **T. Baca**, V. Stehlikova, *et al.*, "X-ray Lobster Eye All-sky Monitor for Rocket Experiment," in *EUV and X-Ray Optics: Synergy Between Laboratory And Space V*, SPIE, vol. 10235, 2017

- R. Filgas, M. Malich, T. Kuwahara, J. Broulim, M. Holik, M. Sakal, Y. Murata, H. Tomio, S. Gohl, and J. M. T. Pineda, "Risepix — A Timepix-based Radiation Monitor Telescope Onboard the Risesat Satellite," *Astronomische Nachrichten*, vol. 340, no. 7, Si, 674–680, Jan. 2019.
- M. Ouyang, X. Zhao, W. He, L. Yang, Y. Hu, Y. Han, S. Ma, and Y. Fu, "Structural Design Method of the Meridional Lobster-eye Lens With Optimal Efficiency," *Applied Optics*, vol. 58, no. 33, 9033–9038, Jan. 2019.
- R. Filgas, "Space Radiation Monitoring With Timepix," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 386–390, Jan. 2018.
- R. Hudec, "Axro Introduction and Historical Background," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 48, no. 3, 396–404, 2018.
- ——, "Low-dispersion Spectroscopy With Cubesats and Photographic Plates," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 416–419, Jan. 2018.
- ——, "Astrophysical Payloads for Picosatellites," *Contributions of the Astronomical Observatory Skalnate Pleso*, vol. 47, no. 2, 143–150, 2017.

[24a] **T. Baca**, G. Loianno, and M. Saska, "Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles," in *IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, IEEE, 2016, pp. 992–997

- D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear Model Predictive Control With Enhanced Actuator Model For Multi-rotor Aerial Vehicles With Generic Designs," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3-4, 1213–1247, Jan. 2020.
- B. Rubi, R. Perez, and B. Morcego, "A Survey of Path Following Control Strategies for UAVs Focused On Quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, 241–265, Jan. 2020.
- L. Ding and H. Wu, "Dynamical Modelling and Robust Control for An Unmanned Aerial Robot Using Hexarotor With 2-DOF Manipulator," *International Journal of Aerospace Engineering*, vol. 2019, Jan. 2019.
- A. Swief, A. El-zawawi, and M. El-habrouk, "A Survey of Model Predictive Control Development In Automotive Industries," in *2019 3rd International Conference on Applied Automation and Industrial Diagnostics (ICAAID 2019)*, 2019.
- J. Faigl and P. Vana, "Surveillance Planning With Bezier Curves," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, 750–757, Jan. 2018.
- ——, "Unsupervised Learning for Surveillance Planning With Team of Aerial Vehicles," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, 4340–4347.

[38a] M. Urban, O. Nentvich, V. Stehlikova, **T. Baca**, V. Daniel, and R. Hudec, "VZLUSAT-1: Nanosatellite with miniature lobster eye X-ray telescope and qualification of the radiation shielding composite for space application," *Acta Astronautica*, vol. 140, pp. 96–104, 2017

- J. J. Lopez-salamanca, L. O. Seman, M. D. Berejuck, and E. A. Bezerra, "Finite-state Markov Chains Channel Model for Cubesats Communication Uplink," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, 142–154, Jan. 2020.
- R. Filgas, M. Malich, T. Kuwahara, J. Broulim, M. Holik, M. Sakal, Y. Murata, H. Tomio, S. Gohl, and J. M. T. Pineda, "Risepix — A Timepix-based Radiation Monitor Telescope Onboard the Risesat Satellite," *Astronomische Nachrichten*, vol. 340, no. 7, Si, 674–680, Jan. 2019.
- W. Furnell, A. Shenoy, E. Fox, and P. Hatfield, "First Results From the Lucid-timepix Spacecraft Payload Onboard The Techdemosat-1 Satellite In Low Earth Orbit," *Advances In Space Research*, vol. 63, no. 5, 1523–1540, Jan. 2019.
- R. Filgas, "Space Radiation Monitoring With Timepix," *Astronomische Nachrichten*, vol. 339, no. 5, Si, 386–390, Jan. 2018.
- P. Hatfield, W. Furnell, A. Shenoy, E. Fox, R. Parker, and L. Thomas, "The Lucid-timepix Spacecraft Payload and the CERN School Educational Programme," *Journal of Instrumentation*, vol. 13, Jan. 2018.

[13a] W. Giernacki, D. Horla, **T. Baca**, and M. Saska, "Real-time model-free minimum-seeking autotuning method for unmanned aerial vehicle controllers based on fibonacci-search algorithm," *Sensors*, vol. 19, no. 2, p. 312, 2019

- A. Ayyad, M. Chehadeh, M. Awad I, and Y. Zweiri, "Real-time System Identification Using Deep Learning for Linear Processes With Application To Unmanned Aerial Vehicles," *IEEE Access*, vol. 8, 122539–122553, 2020.
- M. Gopalakrishnan and N. K. Mohanty, "Integration of CFA2FB Control Scheme With Modified Shunt Active Line Conditioner (MSALC) Connected Distribution System for Mitigation Of Harmonics," *International Transactions on Electrical Energy Systems*, vol. 30, no. 12, Jan. 2020.
- X. Niu, X. Yuan, Y. Zhou, and H. Fan, "UAV Track Planning Based on Evolution Algorithm In Embedded System," *Microprocessors and Microsystems*, vol. 75, Jan. 2020.
- O. Rodriguez-abreo, J. Manuel Garcia-guendulain, R. Hernandez-alvarado, A. Flores Rangel, and C. Fuentes-silva, "Genetic Algorithm-based Tuning of Backstepping Controller for A Quadrotor-type Unmanned Aerial Vehicle," *Electronics*, vol. 9, no. 10, Jan. 2020.

[20a] 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," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2017

- A. Wasik, P. U. Lima, and A. Martinoli, "A Robust Localization System for Multi-robot Formations Based on An Extension of A Gaussian Mixture Probability Hypothesis Density Filter," *Autonomous Robots*, vol. 44, no. 3-4, Si, 395–414, Jan. 2020.
- B. Sabetghadam, A. Alcantara, J. Capitan, R. Cunha, A. Ollero, and A. Pascoal, "Optimal Trajectory Planning for Autonomous Drone Cinematography," in *2019 European Conference on Mobile Robots (ECMR)*, 2019.
- A. V. Savkin and H. Huang, "Asymptotically Optimal Deployment of Drones for Surveillance And Monitoring," *Sensors*, vol. 19, no. 9, Jan. 2019.

[43a] J. Chudoba, M. Saska, **T. Baca**, and L. Preucil, "Localization and stabilization of micro aerial vehicles based on visual features tracking," English, in *2014 IEEE International Conference on Unmanned Aircraft Systems*, vol. 1, IEEE, 2014, pp. 611–616, ISBN: 978-1-4799-2376-2

- M. Rabah, A. Rohan, M. Talha, K.-H. Nam, and S. H. Kim, "Autonomous Vision-based Target Detection and Safe Landing for UAV," *International Journal of Control Automation and Systems*, vol. 16, no. 6, 3013–3025, Jan. 2018.
- H.-j. Jeong, J. D. Choi, and Y.-g. Ha, "Vision Based Displacement Detection for Stabilized UAV Control on Cloud Server," *Mobile Information Systems*, vol. 2016, 2016.

[11a] D. A. Saikin, **T. Baca**, M. Gurtner, and M. Saska, "Wildfire Fighting by Unmanned Aerial System Exploiting Its Time-Varying Mass," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2674–2681, 2020

- J. Gimenez, L. R. Salinas, D. C. Gandolfo, C. D. Rosales, and R. Carelli, "Control for Cooperative Transport of A Bar-shaped Payload With Rotorcraft UAVs Including A Landing Stage on Mobile Robots," *International Journal of Systems Science*,

[10a] M. Petrlik, **T. Baca**, D. Hert, M. Vrba, T. Krajnik, and M. Saska, "A Robust UAV System for Operations in a Constrained Environment," *IEEE Robotics and Automation Letters*, vol. 5, 2 Apr. 2020, ISSN: 2169-2176

- J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative Multi-robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," *IEEE Access*, vol. 8, 191617–191643, 2020.

[37a] V. Daniel, A. Inneman, I. Vertat, **T. Baca**, O. Nentvich, M. Urban, *et al.*, "In-Orbit Commissioning of Czech Nanosatellite VZLUSAT-1 for the QB50 Mission with a Demonstrator of a Miniaturised Lobster-Eye X-Ray Telescope and Radiation Shielding Composite Materials," *Space Science Reviews*, vol. 215, no. 5, p. 40, Jul. 2019, ISSN: 1572-9672

- A. Kear and S. L. Folkes, "A Solution To the Hyper Complex, Cross Domain Reality of Artificial Intelligence: the Hierarchy of AI," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, 49–59, Jan. 2020.

[4c] **T. Baca**, M. Jilek, I. Vertat, M. Urban, O. Nentvich, R. Filgas, *et al.*, "Timepix in LEO Orbit onboard the VZLUSAT-1 Nanosatellite: 1-year of Space Radiation Dosimetry Measurements," *Journal of Instrumentation*, vol. 13, no. 11, p. C11010, 2018

- R. Hudec, "X/EUV and UV Optics for Miniature Cube Satellites Payloads," in *EUV and X-Ray Optics: Synergy Between Laboratory and Space VI*, vol. 11032, 2019.

[15a] **T. Baca**, D. Turecek, R. McEntaffer, and R. Filgas, "Rospix: modular software tool for automated data acquisitions of Timepix detectors on Robot Operating System," *Journal of Instrumentation*, vol. 13, no. 11, p. C11008, 2018

- M. Urban, O. Nentvich, D. Doubravova, O. Petr, A. Inneman, R. Hudec, and L. Sieger, "Timepix: Influence of Temperature and Vacuum on Equalisation of X-Ray Detector and Its Verification," in *UV, X-Ray, and Gamma-ray Space Instrumentation for Astronomy XXI*, vol. 11118, 2019.

[19a] J. Faigl, P. Vana, M. Saska, **T. Baca**, and V. Spurny, "On solution of the Dubins touring problem," in *2017 IEEE European Conference on Mobile Robots (ECMR)*, 2017, pp. 1–6

- A. Wolek, J. McMahon, B. R. Dzikowicz, and B. H. Houston, "The Orbiting Dubins Traveling Salesman Problem: Planning Inspection Tours for A Minehunting AUV," *Autonomous Robots*,

[23a] V. Spurny, **T. Baca**, and M. Saska, "Complex manoeuvres of heterogeneous MAV-UGV formations using a model predictive control," in *2016 IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016, pp. 998–1003

- W. T. Botelho, M. D. G. Bruno Marietto, E. D. L. Mendes, D. R. De Sousa, E. P. Pimentel, V. L. Da Silva, and T. Dos Santos, "Toward An Interdisciplinary Integration Between Multi-agents Systems And Multi-robots Systems: A Case Study," *Knowledge Engineering Review*, vol. 35, 2020.

[17a] J. Chudoba, M. Kulich, M. Saska, **T. Baca**, and L. Preucil, "Exploration and Mapping Technique Suited for Visual-features Based Localization of MAVs," English, *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2016

- X. He, Q. Wang, and Y. Hao, "Finite-time Adaptive Formation Control for Multi-agent Systems With Uncertainties Under Collision Avoidance and Connectivity Maintenance," *Science China-technological Sciences*, vol. 63, no. 11, 2305–2314, Jan. 2020.