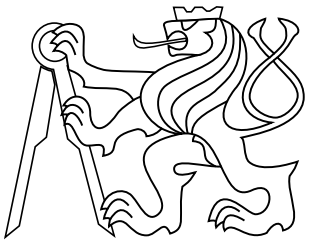
Master's thesis

Predictive Control and Planning Enabling Complex Maneuvers of Formations of Autonomous Helicopters

Bc. Pavel Jícha



May 2015

Thesis supervisor: Ing. Martin Saska, Dr. rer. nat.

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Pavel J í c h a

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Predictive Control and Planning Enabling Complex Maneuvers of Formations of Autonomous Helicopters

Guidelines:

The aim of this thesis is to extend the system designed for stabilization of formations of Micro Aerial Vehicles (MAV) to enable complex maneuvers that require sudden changes of velocity of the group.
Work plan:

- To implement a proper representation of the MAV formation by a convex hull and to design an algorithm that enables a movement of a virtual leader of the formation along the hull.
- To adapt the system of motion planning for the possibility of autonomous design of complex maneuvers with changes of formation velocity, that are allowed by the migration of the leader along the hull.
- To integrate the system of MAV control into the V-REP (virtual robot experimentation platform) simulator.
- To verify the implemented method in the V-REP simulation engine and to study the influence of mutual interactions between MAVs of the formation, which are caused by the air flow from their propellers.

Bibliography/Sources:

[1] M. Saska, Z. Kasl, L. Preucil: Motion planning and control of formations of micro aerial vehicles. Accepted for IFAC World Congress 2014.
[2] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, D. D. Siljak: Cooperative avoidance control for multi-agent systems. Journal of Dynamic Systems, Measurement, and Control, 129:699-707, 2007.
[3] T. Lee, M. Leok, N.H. McClamroch: Geometric tracking control of a quadrotor UAV on SE(3), IEEE Conference on Decision and Control (CDC), 2010.
[4] S.M. LaValle: Rapidly-exploring random trees: A new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State University, 1998.
[5] S.M. LaValle: Planning algorithms. Cambridge University Press, 2006.

Diploma Thesis Supervisor: Ing. Martin Saska, Dr. rer. nat.

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Pavel J í c h a

Studijní program: Kybernetika a robotika (magisterský)

Obor: Robotika

Název tématu: Prediktivní řízení a plánování umožňující komplexní manévry formací bezpilotních helikoptér

Pokyny pro vypracování:

Cílem práce je rozšířit systém vyvinutý pro plánování a stabilizaci formací bezpilotních letounů (MAV - Micro Aerial Vehicles) [1] o možnost komplexních manévrů vyžadující náhlou změnu směru pohybu formace. Plán prací:

- Implementovat reprezentaci MAV formace formou konvexní obálky a navrhnout algoritmus umožňující migraci virtuálního vedoucího formace po této obálce.
- Uzpůsobit systém plánování pohybu formace pro návrh komplexních manévrů se změnou směru pohybu, která je umožněna přesunem virtuálního vedoucího formace.
- Integrovat systém pro řízení MAV formací do simulačního prostředí V-REP (virtual robot experimentation platform).
- Ověřit implementovanou metodu pomocí simulací v prostředí V-REP a studovat vliv vzájemných interakcí mezi helikoptérami formace způsobených simulovaným proděním vzduchu od rotorů.

Seznam odborné literatury:

[1] M. Saska, Z. Kasl, L. Preucil: Motion planning and control of formations of micro aerial vehicles. Accepted for IFAC World Congress 2014.
[2] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, D. D. Siljak: Cooperative avoidance control for multi-agent systems. Journal of Dynamic Systems, Measurement, and Control, 129:699-707, 2007.
[3] T. Lee, M. Leok, N.H. McClamroch: Geometric tracking control of a quadrotor UAV on SE(3), IEEE Conference on Decision and Control (CDC), 2010.
[4] S.M. LaValle: Rapidly-exploring random trees: A new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State University, 1998.
[5] S.M. LaValle: Planning algorithms. Cambridge University Press, 2006.

Vedoucí diplomové práce: Ing. Martin Saska, Dr. rer. nat.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Acknowledgement

I would like to thank to the supervisor of this thesis, Dr. rer. nat. lng. Martin Saska, for his help with this thesis and for his patience about my work on this thesis. I would like to thank also to my family and to my girlfriend for their support.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Datum	Podpis autora práce

Abstract

Existující systém vyvinutý pro plánování trajektorie a stabilizaci formací bezpilotních helikoptér je v této práci rozšířen o možnost plánování komplexních manévrů vyžadujících náhlou změnu směru pohybu formace a také jednotlivých helikoptér ve formaci. Systém založený na metodě prediktivního řízení (MPC) je použit spolu s popisem formace pomocí flexibilní virtuální struktury pro umožnění složitých manévrů během plánování trajektorie. V kombinaci s použitím virtuálních struktur je navrženo řízení rychlosti kvadrokoptér, které je založeno na existujícím řízení pozice (geometric tracking controller). Za pomoci řízení rychlosti jednotlivých kvadrokoptér je naplno využito jejich manévrovacích schopností a jsou tím umožněny komplexní manévry celé formace. Celý systém je implementován a vyzkoušen v simulačním programu V-REP, který zároveň umožňuje studovat vzájemný vliv kvadrokoptér, způsobený prouděním vzduchu z jejich rotorů.

Klíčová slova

prediktivní řízení; MPC; plánování trajektorie; formace; kvadrokoptéra; V-REP; komplexní manévry; sekvenční kvadratické programování; SQP

Abstract

The trajectory planning system originally designed for stabilization of formations of Micro Aerial Vehicles (MAV) is extended to enable complex maneuvers that require sudden changes of velocity of the group as well as the individual formation members. The model predictive control based algorithm is combined with formation representation by flexible virtual structure. Furthermore, the velocity controller for the quadrocopters based on position geometric controller is presented to enable the movement of formation members and whole formation in any direction. The proposed system is implemented and verified in the V-REP (virtual robot experimentation platform) simulation software with studying the influence of mutual interactions between MAVs of the formation, which are caused by the air flow from their propellers.

Keywords

model predictive control; MPC; trajectory planning; formation; quadrocopter; V-REP; complex maneuvers, sequential quadratic programming; SQP

Contents

1 Introduction

There can be found a lot of applications using formations of autonomous robots nowadays, e.g. environment monitoring, searching missions, security monitoring, and there can be found even more. Some of these applications are more suitable for aerial robots, referred as UAV (Unmanned Aerial Vehicle) or MAV (Micro Aerial Vehicle). Furthermore, there can be found applications, where the UAVs with capability of vertical take off and landing (referred as VTOL) are more suitable, e.g. searching in buildings with a lot of obstacles and narrow corridors. These VTOL UAVs include helicopters and multicopters, while multi-copters are commonly used for their simpler mechanical structure, especially, the most common type with four rotors referred as a quadrocopter. The goal of this thesis is to extend the system designed for stabilization of formations of Micro Aerial Vehicles presented in [1], to enable complex maneuvers that require sudden changes of velocity of the group. The system presented in [1] uses a model predictive control based algorithm for keeping the formation of MAVs and also for the motion planning into a target region. This approach allows for integrating the obstacle avoidance system into motion planning of the individual members of a formation as well as into global planning of the whole formation. However, the capability of complex maneuvers is limited due to the use of the leader-follower approach and curvilinear coordinates used to determine the position of the followers. The position of the followers is then determined by the position of the leader in the past. This introduces complications during the complex maneuvers which involves sudden changes of the direction. The complex maneuvers for formations of unmanned mobile car-like vehicles are discussed and the solution introduced in [2].

The structure of this thesis is as follows. The introduction with the description of the state of the art of the formation driving methods and approaches is given in chapter 1, followed by chapter 2, describing the problem, discussing the possible approaches to the problem and introducing the methods used in this thesis. The approach using the flexible virtual structure for formation is further described in detail in chapter 3. In chapter 4, the global trajectory planning for whole formation is described and the local trajectory planning for the formation members is described in chapter 5. The velocity controller for the quadrocopters, needed for the proposed approach, is described in chapter 6 with experiments in simulation to verify the controller functionality. Then, the experiments with the whole formation trajectory planning system is verified by experiments in chapter 7. Finally, the last chapter 8 summarizes the results of the work in this thesis and describes its contribution.

1.1 State of the art

The task of trajectory planning for formations of autonomous robots is intensively studied topic in last years and it can be found many works on the topic, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and many more. There can be found different approaches of driving the formations. These are: virtual structures, leader-follower methods, behavior-based methods, potential field methods and neural networks. The

last two mentioned methods are combined with other formation driving methods to control the formation.

The methods using virtual structures define the formation using the virtual rigid body object, where the formation members are located on vertices of the virtual object. This approach is used in [8] for driving ground mobile robots with the possible use to push the real object along the path.

The leader-follower method defines the position of formation members relatively to leader. The leader can be virtual with no assigned member, or some more equipped robot can be the real leader of the formation. This approach is used in [5] for car-like robots using curvilinear coordinates, that enables specifying the trajectory for the non-holonomic robots. Moreover the multiple virtual leaders are used in [5] and [2] to enable complex maneuvers with change of direction, where it is switched between trajectories of the virtual leaders. The leader-follower method is also used in [1], where the method is extended for use with quadrotor UAVs. The work [3] is also based on leader-follower method using the UAV for top-view relative localization of the other robots and ensuring the direct visibility by using the convex hull of the formation projected along the path.

The behavior-based methods are inspired by the nature and how the animals in the flocks or schools of fishes behave. This method is more suitable for large number of robots, implementing relatively simple control schemas for individual robots in formation, but it can be also used for small formations, as it is presented in [9], where this method is used to drive formation of four robotic scout vehicles in different formation shape.

The methods using the potential fields can be also combined with other methods. The potential fields are generated based on the environment and the mission goals. The 3D potential field filed is for example used in [10] to plan the trajectory for formation of helicopter UAVs. In that paper, the potential field method is combined with leader-follower method to drive the formation. The potential field is generated for every follower based on actual position, desired position and obstacles size and positions. More details can be found in original paper [10].

The neural networks are similarly as the potential fields combined with other formation driving methods. In the paper [11], the neural networks are combined with leader-follower method to control the trajectory of the formation members. The neural networks are in that paper used to control the quadrotor UAVs with their complex dynamics and aerodynamics nonlinearities.

2 Preliminaries

The methods used for trajectory planning for formation of autonomous quadrotor helicopters are introduced in this chapter. First, the task of trajectory planning for formations in general with possible approaches is discussed in following section 2.1. Next, the model predictive control method, used as base for the formation driving, is introduced in section 2.2. Finally, the method for solving the optimization problems, needed to use the model predictive control method, the sequential quadratic programming is introduced in the section 2.3.

2.1 Problem statement

The task of trajectory planning for formation of the quadrotor helicopters is defined with requirement to enable complex maneuvers with sudden changes of direction of the formation motion. The assumption for the global environment is that the global map $\mathcal{M}$ is known with set of all static obstacles $\mathcal{O}$. There is set of quadrotor UAVs $\mathcal{Q}$ in the formation. The visual-based relative localization is used to determine the positions of formation members. The estimate of the global position in the map is also available.

The introduction to different methods for driving the formations of autonomous robots was given in the section 1.1. To revise, there were mentioned virtual structure based methods, leader-follower methods, behavior-based methods, potential field methods and neural networks. Two possible approaches to the task of extending the existing system for trajectory planning for formations of micro UAVs [1] by enabling complex maneuvers were selected to further analyze. The first approach is using the leader-follower method, originally used in [1], and the second is using the virtual structure based method for formation driving.

2.1.1 Leader-follower approach

The idea behind the leader-follower approach is to us the convex hull for formation representation with combination of virtual leader. The complex maneuvers are in this case enabled by migration of the virtual leader along the convex hull. This enables the sudden changes of the direction of the formation without complications that arise when the curvilinear coordinates are used to specify the positions of the formation members where the position is retrieved using the past position of the leader, therefore the leader has to be always in the front of the whole formation. The example of the convex hull of the formation with illustrated migration of the virtual leader is shown on the Fig. 1.

The migration of the virtual leader along the convex hull can be implemented by adding parameters to optimization vector for trajectory planning of the virtual leader. The convex hull representation can be reduced to smallest possible sphere around the formation members. The position of the virtual leader L_v on the sphere is then given by fixed radius r_{VL} and two spherical coordinates: the azimuthal θ and the polar ϕ. These two spherical coordinates would then be part of every element of the trajectory.

The virtual leader was used to overcome the complications related to complex maneuvers mentioned above in works studying the formations of the ground car-like robots

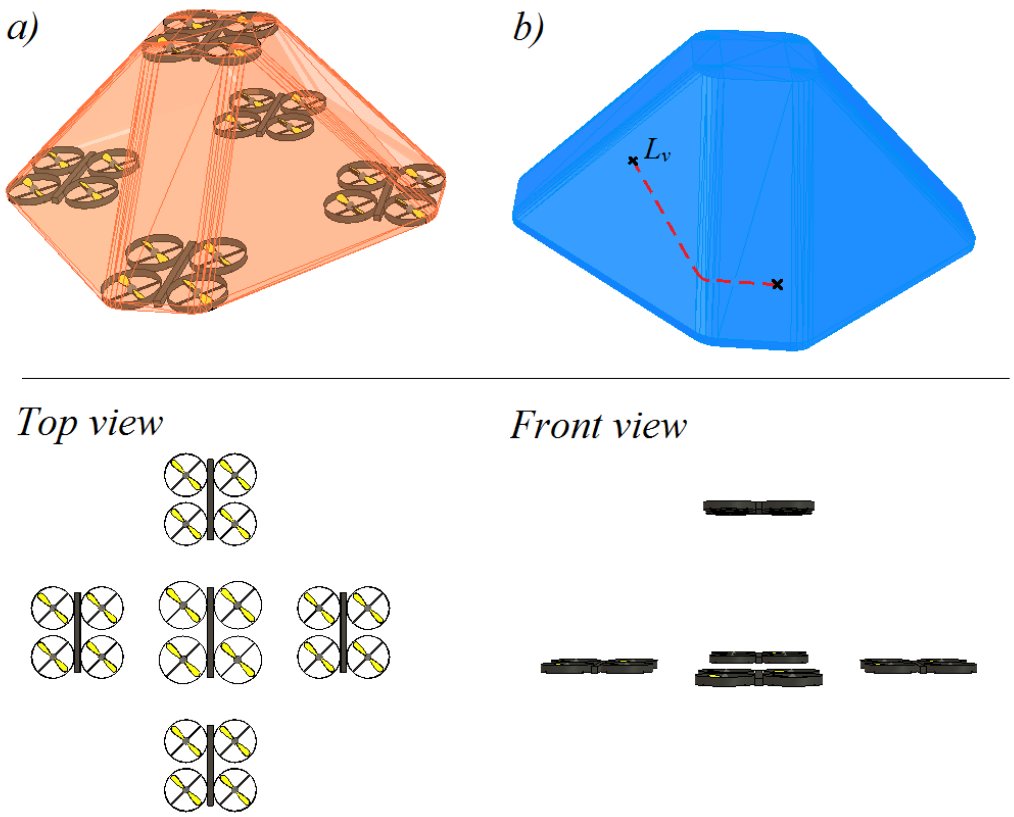


Figure 1 The example of the convex hull of the formation. a) The UAVs in formation with the convex hull shown in light red. b) The convex hull with the migration of the virtual leader position L_v. The formation is shown as viewed from top and from front.

[2] and [5]. The demonstration of difficulties that can occur during complex maneuvers of the formation is shown on the Fig. 2, where the formation of the four car-like robots tries to change the direction of the movement.

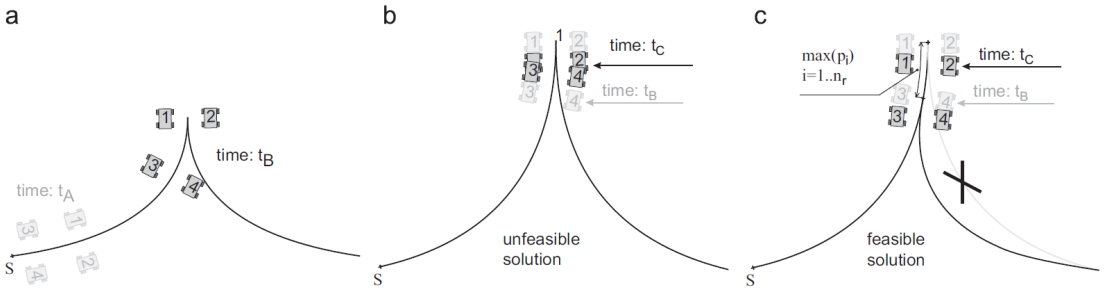


Figure 2 Demonstration of difficulties during complex maneuvers of the formation using the common leader-follower concept. Source: [2]

The snapshot a) on the Fig. 2 shows the initial position of the formation (shaded contours) and the position where the direction of the motion is changed. The snapshot b) shows the collision between the formation members if the common leader-follower method was used. On the snapshot c) the modified path is shown for backward motion of the formation with the robots numbered $3, 4$ are leading the formation. There is in-

troduced concept of leader-follower approach with two virtual leaders, and it is switched between the trajectories of these two leaders to overcome the difficulties shown on the Fig. 2.

2.1.2 Virtual flexible structure based approach

The virtual structure based approach is using the rigid-body like virtual structure to represent the formation. The center of the structure is reference point for the formation members to determine the position in the formation. The global trajectory is planned for the center of the virtual structure with size of the structure defined by sphere with radius r_{VS}.

To enable the virtual structure movement without the modification of the shape of the formation, and further, without the limitations of the virtual structure movement, such as minimal radius of the curve in the trajectory, the individual robots have to be controlled in a way, that allows motion in any direction. This can be done by using the velocity vector $\vec{v}$ as control input for the quadrotor UAV. The controller is described in chapter 6.

This approach allows the formation to hold the desired shape throughout the progress in the environment to the target and allows the formation to suddenly change the direction of the motion, e.g. to proceed with sideway motion or backwards motion. The only deformation of the shape of the formation can be caused by collision avoidance of the robots, if it is allowed by global trajectory planning, e.g. the global trajectory is planned closer to the obstacles than the size of the formation. Therefore the virtual structure is flexible.

2.1.3 Summary

There were described two approaches to enable complex maneuvers for formation of quadrotor UAVs in sections 2.1.1 and 2.1.2.

The first proposed approach enables the complex maneuvers by using the leader-follower method with virtual leader, that can migrate on the convex hull representation of the formation. This approach would involve extending the vector representing the trajectory by additional parameters, defining the position of the virtual leader on the convex hull. This would increase the dimension of the optimization solution depending on the number of elements in the vector.

The second proposed approach is using virtual flexible structure to represent the formation and keep the desired shape. The direct advantage of this approach is in non-constrained motion of the formation, because the whole formation can move in any direction in any moment. Further, when compared with the leader-follower approach, there are no additional parameter in the vector, therefore the solution of the optimization can be calculated faster.

After the comparison of the two proposed approaches, the second approach, using the virtual flexible structure for the formation representation, is selected for this thesis, mainly because of the keeping the low count of the parameters in optimization vector, therefore faster calculations. Another advantage is related to the visual relative localization used in several works dealing with the formation of the autonomous robots, where in the formation that keeps the predefined shape of the virtual structure, most of the time is the relative localization used only to stay in fixed relative position to other formation members, without any changes in relative position during the turns in contrast with the leader-follower approach using the curvilinear coordinates for the

followers, therefore changing the relative positions of the followers in global cartesian coordinate system.

2.2 Model predictive control

The model predictive control (MPC), also referred as the Receding Horizon Control (RHC), is the method of determining the optimal control sequence with given objective function as subject to minimization. The method is using the model of the system in the optimization. The optimization is done by minimization of the objective function, which is constructed according to application specific needs. The method is using the model to predict the state of the system for given sequence of the control inputs for given prediction horizon, which can be finite or infinite [12]. When the optimization of the current sequence is done, only the first control is applied, the state of the system is updated and the optimization for the new state is run. The more detailed view on MPC can be found in [13].

This origin of this technique can be dated to the 1960s [12] and the method was used for long time only for industrial processes with slow dynamics for which the processing power of computers was sufficient to solve the optimization of the model of the system. It is shown, that in last years, this technique can be also used for systems with fast dynamics, such as micro UAVs, and the MPC can be computed onboard of embedded controllers directly on the UAV [14].

The general idea of MPC can be further used for trajectory planning and formation stabilization in one optimization process. This approach has been introduced in [15] for the leader of the formation. The original MPC optimization problem is coded into an optimization vector with N elements as the predictive horizon. Each of the vector elements has constant duration Δt_N. The idea introduced in [15] is to extend the optimization vector by additional M vector elements with variable duration Δt_M. This part of the optimization vector is added to include the global trajectory planning to the optimization problem, therefore the whole trajectory is coded into the optimization vector. As the result of this extension, the formation leader can therefore react not only to local changes in environment, but also the changes in global environment and change the trajectory accordingly.

2.3 Sequential quadratic programming

The sequential quadratic programming (further referred as SQP) is method for solving nonlinearly constrained optimization problems. The SQP itself is not just a single algorithm but rather it is a base method from which special algorithms are derived [16]. The nonlinear programming problem can be described by following notation:

$$\begin{aligned} &\underset{x}{minimize}\ f(x) \\ &subject\ to: \quad \begin{aligned} h(x) &= 0 \\ g(x) &\leq 0 \end{aligned} \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ specifies the optimality of the solution x, and the constraint functions $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ describe the limitations of the system [16].

As the name of the method suggests, the method works sequentially or iteratively. The function f is modeled by a quadratic programming subproblem at a given approximate solution. The subproblem is solved to get a better approximation of the solution

to the original problem. The idea is to iteratively repeat the process to converge to a solution x^*. More details about SQP can be found in [16].

To solve the MPC based optimization problem for the trajectory planning of the quadrotor helicopters, the C library with implementation of the SQP is used. This library is called C code for Feasible Sequential Quadratic Programming (CFSQP) by Crag T. Lawrence, Jian L. Zhou and André L. Tits and more information about this library can be found in the library user-manual [17].

3 Formation representation - Flexible virtual structure

As stated in previous chapter, the model predictive control based algorithm is used for driving the formation of the quadrotor UAVs. The formation stabilization is done using the virtual structure method. The formation is given by the center of the virtual structure, the $\vec{x}_{VS}$, and the positions of the formation members, given by relative position to the center of the virtual structure, the $\vec{p_{f1}}, \vec{p_{f2}}, \ldots, \vec{p_{fn}}$, where the n is the number of formation members. The example of formation with the virtual structure and its center is shown on Fig. 3. The design allows to change the formation also during the flight in the middle of the trajectory.

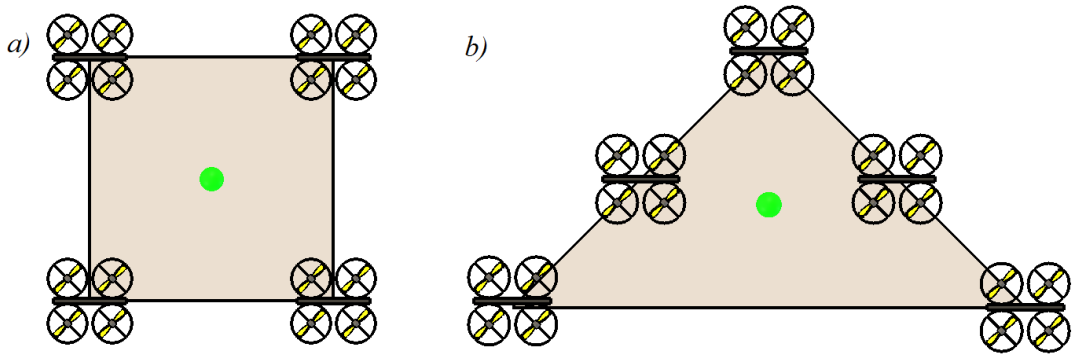


Figure 3 The examples of formations represented by virtual structure with the center shown as green circle. a) the rectangle formation, b) arrow formation

As the result of using the virtual structure for the formation representation and for the global planning, the whole formation behaves almost as the rigid body (the virtual structure is flexible to allow the collision avoidance) with the formation members as the vertices of the virtual rigid body. The movement of the formation is depicted on Fig. 4, where the formation executes a sharp turn. It is shown, that the formation shape is kept in global coordinate system. If the mission requires the rotation of the formation according to the direction of the movement with the proposed virtual structure method, the formation representation in global coordinate system would have to be recalculated to accommodate this change. Also for such a situation, the leader-follower approach is more suitable.

3.1 Trajectory parameterization

As it was stated in section 2.1, the trajectory has to be coded into a vector of constant control inputs. The trajectory is then represented by sequence of constant control inputs. The representation that allows the complex maneuvers with immediate changes of move direction is chosen with respect to capabilities of the quadrotor UAV to move

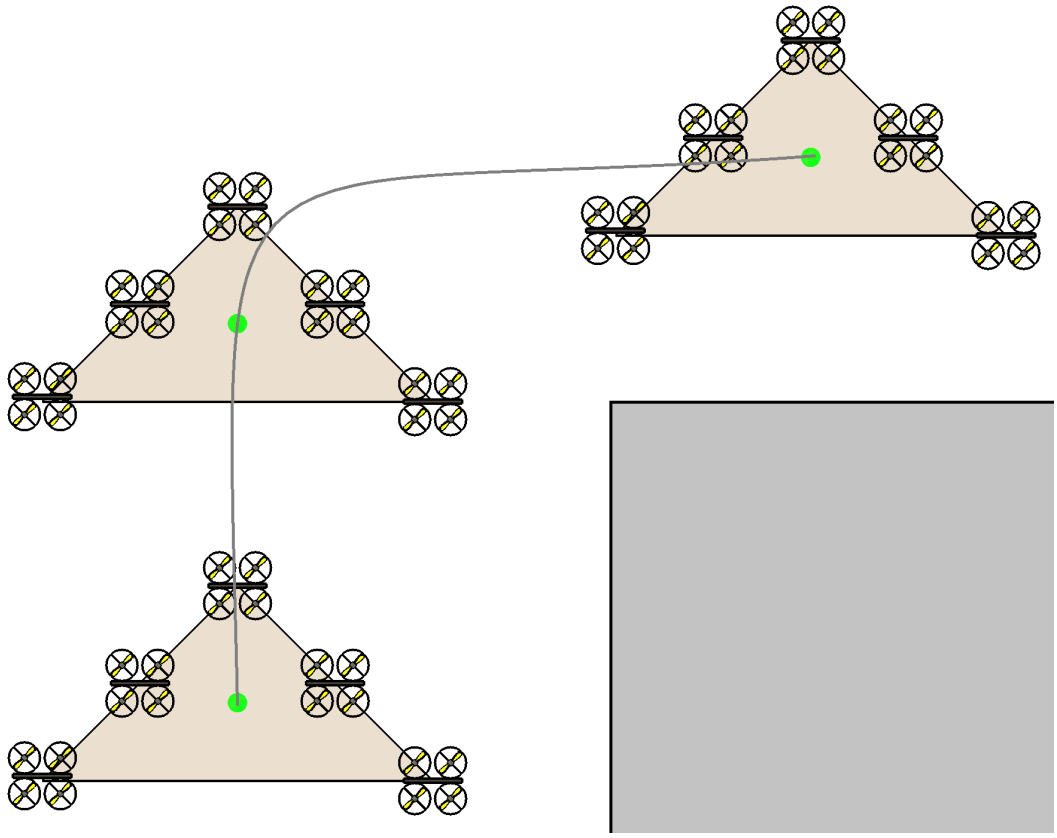


Figure 4 The example of formation movement, when represented by virtual structure with the center shown as green circle. The example trajectory to avoid the obstacle, shown as grey box, is shown as grey line.

in any direction immediately. The trajectory element is described by velocity vector $v = [v_x, v_y, v_z]^T$ and the time duration of the element Δt:

$v_x[m \cdot s^{-1}]$	the velocity in direction of global x-axis (i_1),
$v_y[m \cdot s^{-1}]$	the velocity in direction of global y-axis (i_2),
$v_z[m \cdot s^{-1}]$	the velocity in direction of global z-axis (i_3),
$\Delta t[s]$	the time duration of the trajectory element.

The vector that represents the trajectory is shown on the Fig. 5. The trajectory is represented by an array of the control inputs with time duration as elements. For the purpose of the implementation, the representation by an 1-D array is used, where a single element of the trajectory consists of four numbers: $v_x, v_y, v_z, \Delta t$. The number of elements is shown as N on the Fig. 5.

Figure 5 The trajectory representation by a vector.

The trajectory represented by the proposed vector creates poly-line with straight elements. The transition between two elements is done by the quadrotor controller,

described in chapter 6.2. The transitions will be rounded, as it is indicated on the Fig. 6 by green line, depending on the quadrotor dynamics and the controller. The "ideal" trajectory with constant velocity inputs, when no dynamic or kinematic model is used for trajectory estimation, is shown as blue dashed line. The simplified model, described in section 3.3, is used to estimate the predicted trajectory for MPC algorithm. The predicted trajectory is then evaluated by objective function as it is described in detail in chapters 4 and 5. To ensure that the inaccuracies of the simplified model of the UAV will not lead to collisions, the safety region around the obstacles has to be chosen carefully to be large enough.

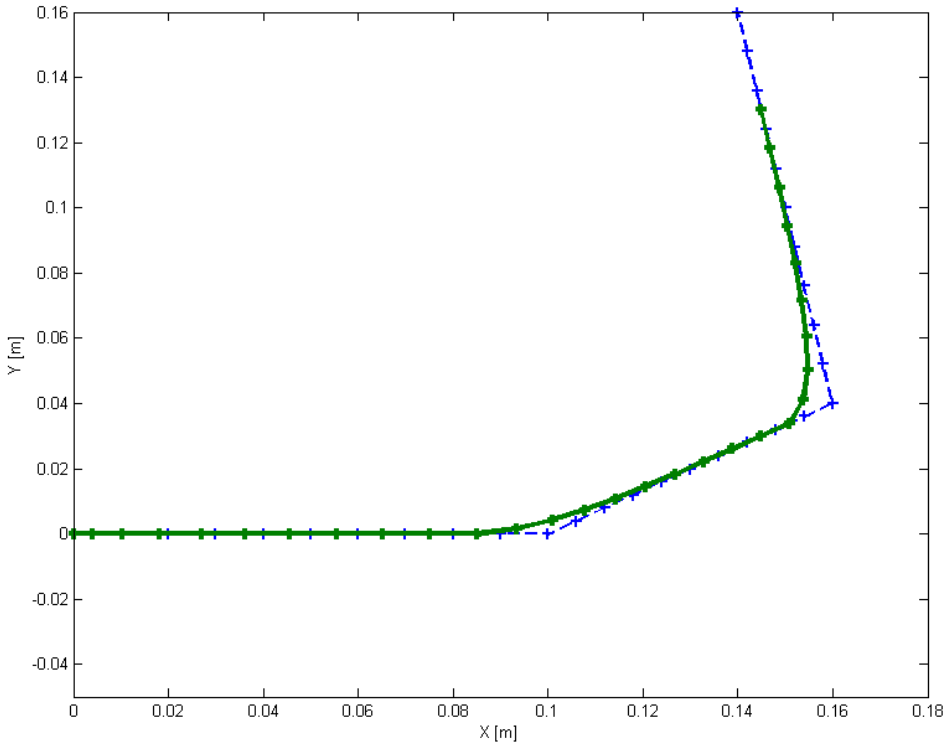


Figure 6 The trajectory example for constant velocity inputs. The inputs are: $v_x(t) = 0.5m \cdot s^{-1}, v_y(t) = 0m \cdot s^{-1}$ for $t \in < 0, 3)$, $v_x(t) = 0.3m \cdot s^{-1}, v_y(t) = 0.2m \cdot s^{-1}$ for $t \in < 3, 6)$ and $v_x(t) = -0.1m \cdot s^{-1}, v_y(t) = 0.6m \cdot s^{-1}$ for $t \in < 6, 9 >$.

3.2 Trajectory planning

The formation trajectory planning structure is shown on Fig. 7. It consists of the trajectory planning for the virtual structure and the trajectory planning for the formation members. The trajectory planning for the virtual structure is used to find global trajectory in the global environment to reach the target region and avoid the obstacles. The goal of the trajectory planning of each formation member is to track the desired position to stay in formation and avoid the collisions with obstacles and other robots in formation. The trajectory is represented by vector with elements $Tr(i), i = 1 \dots N + M$ for the global trajectory and $Tr(i), i = 1 \dots N$ for the trajectory of the individual robots.

Every trajectory element consists of four numbers as stated in the previous section 3.1, the $v_x, v_y, v_z, \Delta t$.

The approach for global trajectory planning discussed in the previous chapter and originally presented in [15] is used. The global trajectory consists of N constant time duration elements and M variable time duration elements. The M elements with variable Δt are added to trajectory vector for global planning to include the global trajectory planning in the MPC optimization [15].

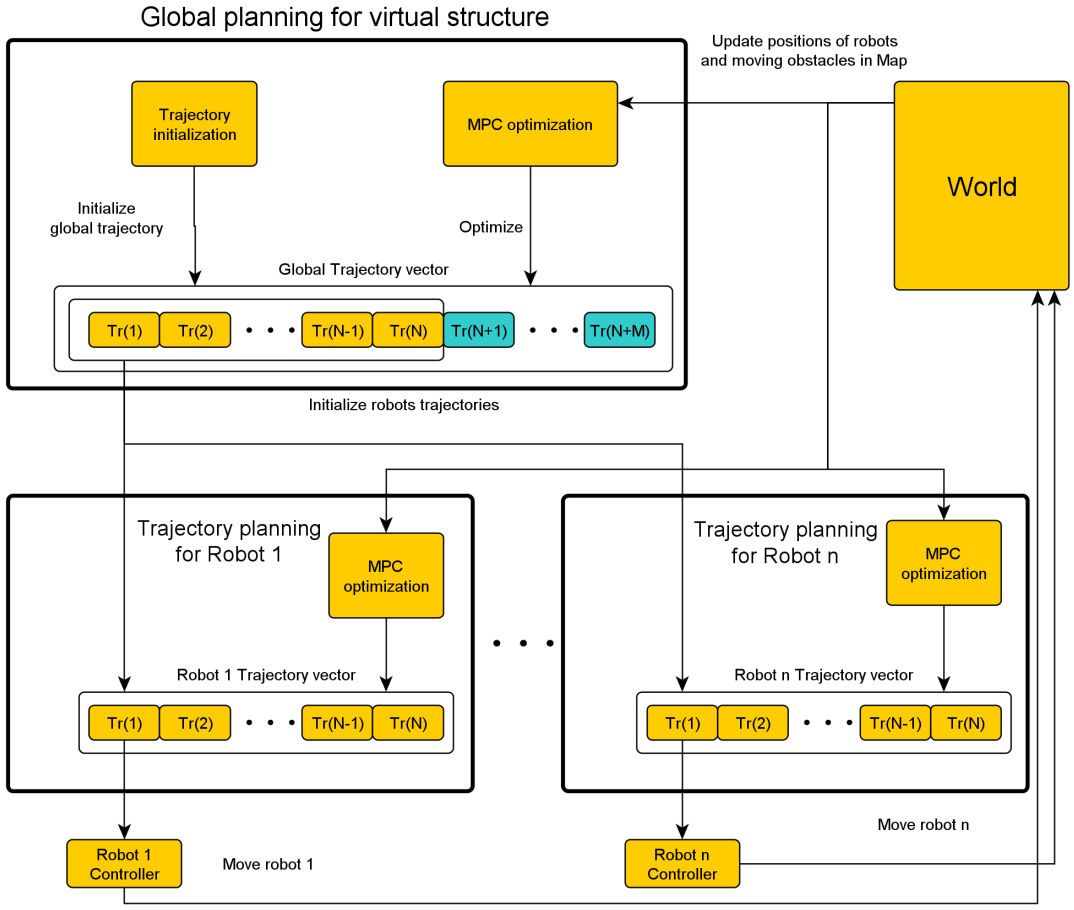


Figure 7 The structure of trajectory planning for virtual structure and formation members.

3.3 Simplified model for MPC calculations

The MPC-based trajectory planning used for the formation driving is using a model of the real system to predict future states of the system and optimize the control inputs accordingly. More details about the MPC method can be found in section 2.2.

In this thesis, the MPC is used for trajectory planning of the virtual structure and the individual robots. The model used in MPC has to be adequately complex to enable accurate prediction of the trajectory with negligible inaccuracies when compared with the real robot. On the other hand, the model has to be as simple as possible to enable fast calculations of the trajectory prediction and to enable the system to run online on embedded controllers onboard the UAVs. Therefore the controller and the complete

dynamics model of the quadrotor can not be used for trajectory prediction.

To satisfy the requirements for the model of the system of a quadrotor UAV with the velocity controller, a simplified kinematic model is used for trajectory evaluation. The model with the controller is described by following equations. The equations (2) and (3) describe the general kinematic model of the quadrotor UAV. The equations (4) and (5) describe the simple proportional controller.

Because the trajectory is represented by vector of constant velocity control inputs, the velocity output of the proposed model is shown on step function on Fig. 8. There is shown the effect of the proportional constant k_v on the speed of the controller on the Fig. 8.

The comparison between the proposed simplified model and the "real" quadrotor UAV with dynamics simulated in V-REP simulation software and with the geometric tracking controller implemented is done in section 6.3.2.

$$v = \dot{x} \tag{2}$$

$$a = \dot{v} \tag{3}$$

$$e_v = v - v_d \tag{4}$$

$$a = -k_v \cdot e_v \tag{5}$$

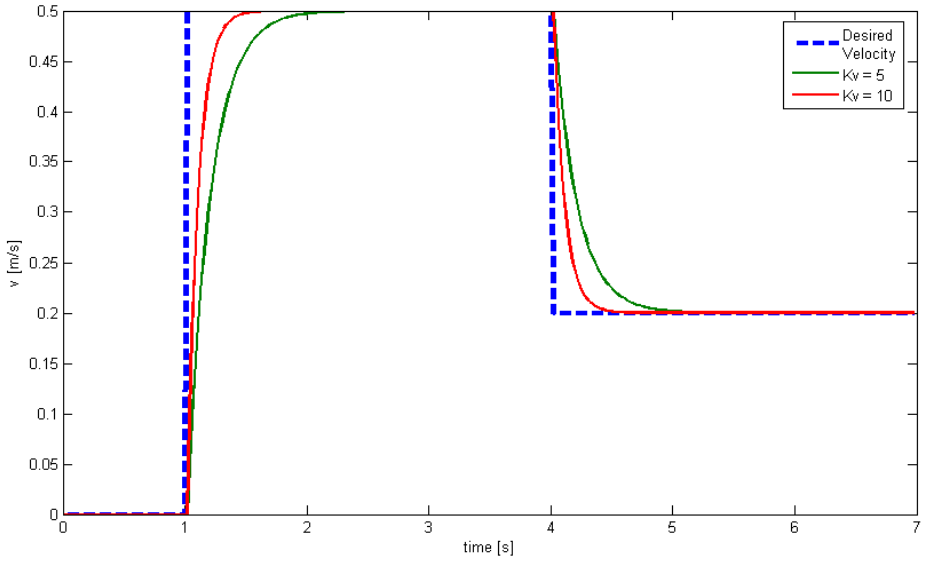


Figure 8 The step responses of the simplified model used for MPC with different k_v.

4 Trajectory planning for virtual structure

The virtual structure represents the whole formation, the center of the virtual structure serves as the reference point for the formation members to keep the desired formation. The global trajectory is planned for the center of the virtual structure and for the trajectory planning, it is represented by sphere with radius r_{VS} determining its size. The size of the virtual structure used for the trajectory planning can be different from the real formation size and it can be set according to specific demands of application. For the applications, where the direct visibility between the formation members is needed, e.g. for formations, where visual relative localization is used [18], the size of the virtual structure used for trajectory planning has to be large enough for trajectory to be feasible for the whole formation without changing its shape, or with minimal changes, and to ensure that no obstacles can be between the formation members. If the different localization is used, e.g. global visual localization [19], and there are no such constraint, then the size of the virtual structure can be smaller. This will allow the formation to split around smaller obstacles and therefore can lead to shorter trajectories.

Before the optimization of the trajectory begins, there has to be given the initial estimation of the feasible trajectory. The possible methods of finding this initial trajectory are discussed in following section 4.1.

4.1 Initialization of the trajectory

In order for the trajectory optimization to work properly, the initial trajectory estimate has to be given. The initial trajectory has to be feasible and has to go from start position to the target region. According to the CFSQP (the SQP is described in section 2.3) library manual [17], the algorithm is able to find the feasible trajectory (the optimization vector in general) even when the un-feasible trajectory is given as initial, but it is not guaranteed.

As the implementation of an algorithm for finding the initial estimate of the trajectory for the MPC method is not the objective of this thesis and some of the methods have been already implemented in other works [20], [21], the possible approaches to the initialization are only described and discussed in this section.

There can be found numerous methods and algorithms for path planning or motion planning in the literature. The path planning methods divided by the techniques of representation of the configuration space are shown on Fig. 9. There are geometric methods using the roadmap representation, such as visibility graph or Voronoi diagrams, the methods using a discrete grid-based space, or the cell decomposition and the graph searching algorithms such as breath-first search, depth-first search, A-star, Dijkstra, etc. [22]. Another commonly used path planning algorithms using the Roadmap representation and based on stochastic approaches are probabilistic roadmaps and Rapidly exploring dense trees (RDT), that are using incremental sampling and searching, and the special case where the sampling sequence is random, called Rapidly exploring random tree (RRT) [22], [23]. Another method for path planning is using the potential fields.

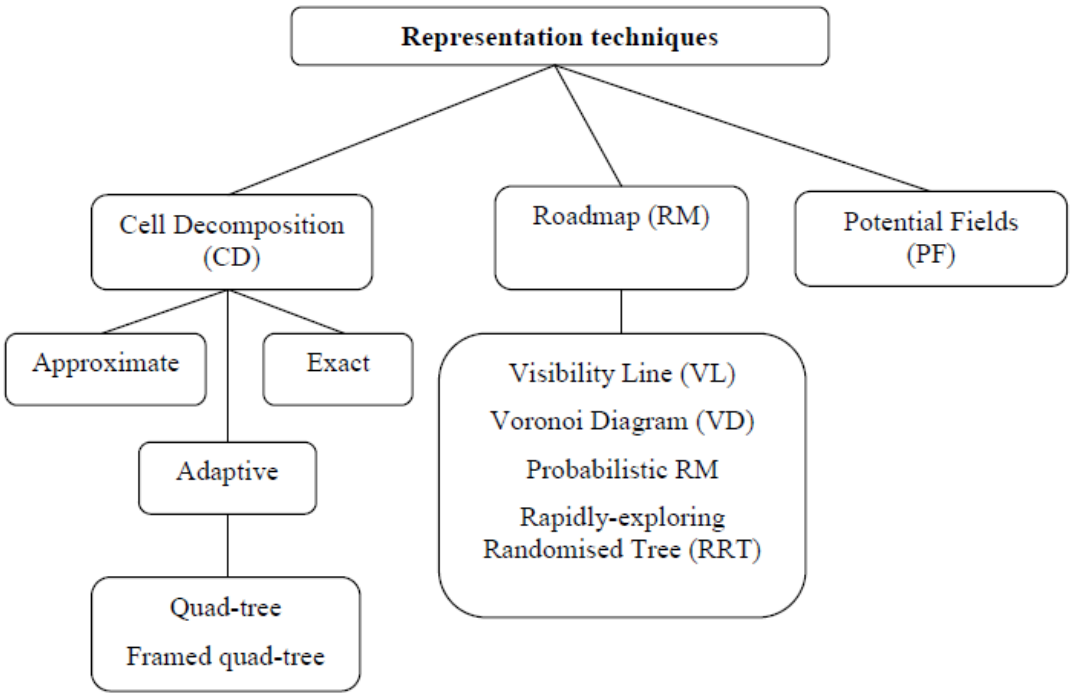


Figure 9 Path planning methods divided by $C - space$ representation. Source: [24]

4.1.1 Rapidly exploring random tree

The last mentioned method has been already used in works [20], [21] to find feasible trajectory for MPC technique based trajectory planning. In the work [20], it is shown, that the RRT can be used directly to find the feasible trajectory for the UAV robot in an known environment with obstacles and the trajectory is in the form of a vector suitable for the optimization in MPC method. Further the sequential processing for the trajectory search is used in [20], where piecewise processing of the trajectory, where the start and end states of each of the processed parts of the trajectory is fixed. The sequential processing helps to reduce the computation time of the implemented RRT algorithm. The example of the resulting tree generated by the RRT implemented in [20] is shown on Fig. 10, where the start point is represented by green cross, the target point is represented by red cross and the trajectory found by the algorithm is shown as bold blue line.

Both mentioned works dealing with the RRT method [20], [21], use the RRT for trajectories, that are represented by a vector of control inputs for a non-holonomic UAVs, the tangential velocity v_τ, normal velocity v_η and the curvature k. As it is stated in section 3.1, the trajectory is represented, in the approach presented in this thesis, by a vector of control inputs for a holonomic UAV, with control inputs v_x, v_y, v_z. Therefore the trajectory found by RRT algorithm modified to be used with approach presented in this thesis would look a bit different, where the resulting tree would be composed rather from the straight lines with rounded transitions, as it is illustrated in Fig. 6, instead of circular-like sections.

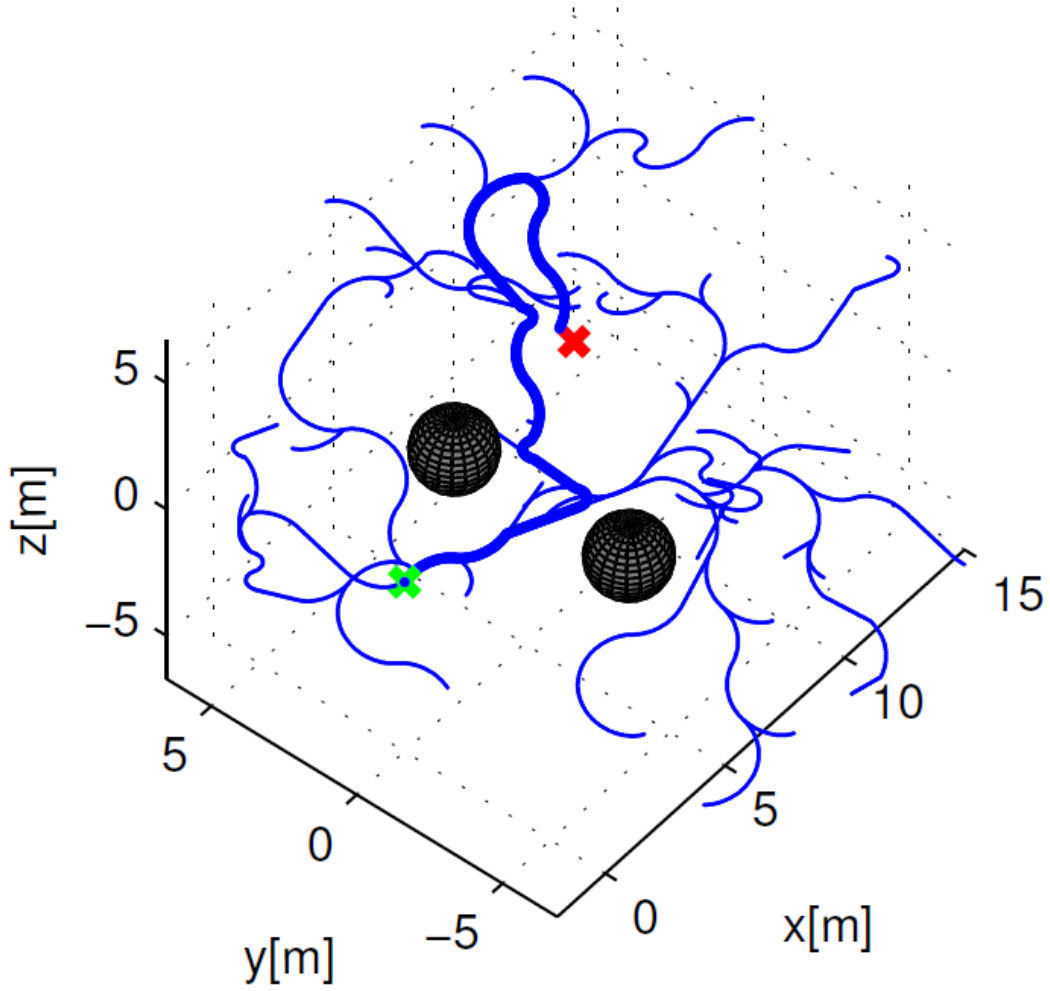


Figure 10 An example of the tree generated by the RRT algorithm. Source: [20]

4.1.2 Visibility graph

Another approach commonly used for path planning is the visibility graph. This method is designed to find the shortest path between the obstacles. The visibility graph is constructed by connecting the visible nodes, where the nodes are the path endpoints and the obstacle vertices [25]. When the visibility graph is constructed, some of the graph search method can be used to find the shortest path. An example of the path planned by visibility graph method and the Dijkstra's graph search algorithm is illustrated on Fig. 11, where the black numbered boxes represent the obstacles, the blue triangle represents the starting position, the purple circle represents the target position. The resulting trajectory is shown as purple line. The visibility graph method has been commonly used for 2-D environments for ground mobile robots, but several extension to use this method in 3-D environments has been introduced [25], [24].

The resulted paths found by the visibility graph methods are composed from the straight line segments and from the nature of the approach, the paths are generated close to the obstacles, which can be in some cases inappropriate, like in applications with non-holonomic robots. In this thesis, the holonomic quadrotor UAVs are used in

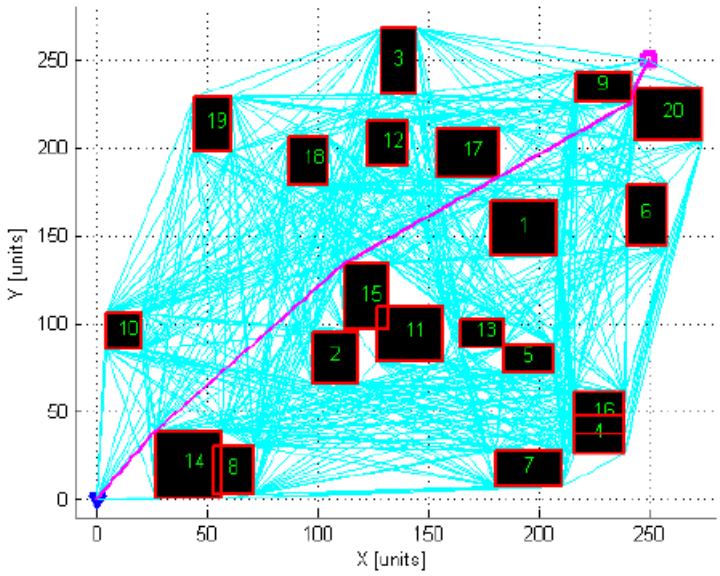


Figure 11 An example of the path planned using visibility graph and Dijkstra's graph search algorithm. Source: [24]

conjunction with the trajectory represented by the elements with constant UAV velocity, therefore, the visibility graph based methods for 3-D environments can be used for the initial path and then the initial trajectory estimation.

4.2 Trajectory optimization

When the initial trajectory is given as the vector described in previous sections, the vector is passed to the CFSQP solver. The trajectory vector is then optimized by minimizing the objective function and respecting the constraints given by constraint functions. Both these functions are user-defined and control the parameters of the optimal trajectory based on the applications needs. The objective function is described in detail in following section 4.2.1 and the constraints function is described in section 4.2.2, both based on originally presented objective and constraints from [1]. The possibility to avoid collisions with dynamically moving obstacles is added to the trajectory planning and the details about this functionality is described in section 4.2.3.

4.2.1 Objective function

The objective function is used to evaluate the optimality of the trajectory T_{VS}. The CFSQP solver tries to minimize the objective function to find the optimal trajectory. The objective function depends on the application and should be constructed to satisfy the specific needs for intended environment, e.g. there can be situations where the time of flight is not important, but the furthest possible distance from the obstacles should be held. For general use, the objective function J_{VS} is constructed as weighted sum of several components. Every component evaluate the trajectory according to different parameter. The components of objective function are listed below:

Obstacle avoidance component of the objective function penalizes the trajectories

according to the closest point of the trajectory to obstacles. The function from [1] is used:

$$J_{obst,VS} = \max_{i=1...n_o} \left(min \left\{ 0, \frac{dist(T_{VS}, o_i) - r_{s,VS}}{dist(T_{VS}, o_i) - r_{a,VS}} \right\} \right)^2, \tag{6}$$

where the n_o is the number of obstacles, including the dynamic obstacles described in section 4.2.3, the $dist(T_{VS}, o_i)$ function determines the minimal distance between every point in the trajectory T_{VS} and the obstacle o_i, the $r_{s,VS}$ is the safety distance from which the trajectory will be penalized and the $r_{a,VS}$ is the critical distance from the obstacle. This obstacle avoidance function satisfy the requirements, that the penalty will be 0, if the trajectory is far enough from the obstacles, specifically, every point of the trajectory is further than the safety distance. Another requirement for such a function is to rapidly increase, when the safety distance from obstacle is breached and increase to infinity when approaching the critical distance. This requirement is also fulfilled if the condition $r_{a,VS} < r_{s,VS}$. As it was stated in chapter 4, the size of virtual structure used for the trajectory planning determines, how close the global trajectory will be planned to the obstacles. The safety distance from the obstacle has to satisfy $r_{s,VS} > r_{VS}$, where the r_{VS} is the radius of the virtual structure. To ensure the direct visibility between the formation members the r_{VS} has to be large enough to enclose the formation members to the sphere defined by r_{VS}. The critical distance from the obstacle $r_{a,VS}$ determines the minimal allowed distance from the obstacle. With lower the value $r_{a,VS}$, the global trajectory can lead closer to the obstacles, therefore it can lead to shorter trajectories at the cost of the deformation of the formation shape.

Distance to target component of the objective function penalizes the trajectories, that do not end near the target region around the goal. This part of objective function, J_{target} is directly calculated as the distance of the last point in trajectory to the target. This component helps to stabilize the optimization by pushing the solution closer to the goal.

Trajectory length component of the objective function penalizes the longer trajectories, therefore the solver will prefer the shorter trajectories. The value of this function is calculated as:

$$J_{len,VS} = \sum_{i=1}^{N+M} \parallel \vec{v_i} \parallel \Delta t_i. \tag{7}$$

Time of flight component of the objective function penalizes the time needed to reach the target region. The time of flight is determined as sum of all segments of the trajectory:

$$J_{tof,VS} = \sum_{i=1}^{N+M} \Delta t_i. \tag{8}$$

4.2.2 Constraint function

The constraint function is used to add limitations of the controlled system to optimization process. The user can define several constraint functions in CFSQP. The constraint functions are then selected by an index in the optimization.

The constraints are defined using inequality equations. The constraint functions returns the value C for given trajectory. The trajectory is feasible, if the constraint function value $C < 0$. If the constraint function value $C \geq 0$, the trajectory is considered unfeasible.

For the virtual structure trajectory planning, the following constraint functions are applied to optimization process, ensuring the collision free trajectories and respecting the limitations of the robots in formation. These constraints are based on constraints presented in [1].

Obstacles avoidance constraint is similar to obstacle avoidance objective function. The closest distance to obstacles is determined for the whole trajectory T_{VS} and all obstacles $o_i, i \in \{1 \dots n_o\}$, including the dynamic obstacle described in section 4.2.3, by function $dist(T_{VS}, o_i)$, where n_o is the number of obstacles. The constraint ensures that the minimal distance of the trajectory to obstacles is more than critical radius $r_{a,VS}$. Therefore the obstacle avoidance constraint can be written as

$$C_{obst,VS} = r_{a,VS} - \min_{i=1\dots n_o} \left(dist(T_{VS}, o_i)\right). \tag{9}$$

The value of this function is < 0, therefore the trajectory is feasible, if minimal distance from obstacles is greater than critical radius $r_{a,VS}$.

Distance to target constraint is ensuring that the trajectory will end in target region defined by sphere with radius r_F and the center in the target position x_F. This constraint is important for the stability of the formation driving process as the trajectories that does not end in the target region as considered as infeasible. The constraint can be written as:

$$C_{target,VS} = dist(T_{VS}(N+M), x_F) - r_F, \tag{10}$$

where the $dist(T_{VS}(N+M), x_F)$ stands for distance of the last point in trajectory from the target position.

Control limits of the whole trajectory added as constraints to optimization process to limit the control inputs needed to follow the prescribed trajectory. The constraints for the whole trajectory are stricter than for formation members to allow them to recover the position in formation after maneuvering, e.g. after avoiding the collision. Also the limits for the variable time duration of the trajectory vector elements are defined. These limits can be written as set of constraint functions:

$$C_{lim,VS} = \begin{pmatrix} v_{xVS} - v_{xVS,max} \\ v_{yVS} - v_{xVS,max} \\ v_{zVS} - v_{xVS,max} \\ v_{xVS,min} - v_{xVS} \\ v_{yVS,min} - v_{yVS} \\ v_{zVS,min} - v_{zVS} \\ \Delta t_{VS} - \Delta t_{VS,max} \\ \Delta t_{VS,min} - \Delta t_{VS} \end{pmatrix} \tag{11}$$

4.2.3 Dynamic obstacles avoidance

The basic functionality is added to the trajectory planning to enable robots avoiding the collisions with dynamic obstacles. This part of obstacles avoidance is designed with respect to the limited possibilities of the robots onboard sensors to detect the moving obstacle.

It is assumed that the position of the moving obstacle is measured repeatedly and its velocity can be estimated based on consecutive position measurements. The estimated velocity is then used to predict the position of the obstacle in future to allow the MPC algorithm to optimize the trajectory with respect to the dynamic obstacles. Let's denote the two consecutive position measurements $x_{dyn}(t_{k-1}), x_{dyn}(t_k)$, then the estimated velocity is calculated as:

$$v_{dyn,est}(t_k) = \frac{x_{dyn}(t_k) - x_{dyn}(t_{k-1})}{t_k - t_{k-1}} \tag{12}$$

The dynamic obstacles avoidance is included to function $dist(Traj, o_i)$, which returns the minimal distance between every point of the trajectory $Traj$ and the obstacle o_i. In case of the dynamic obstacles, the distance is calculated for the predicted position based on estimated velocity $v_{dyn,est}$ for appropriate time of the point on the trajectory.

5 Trajectory planning for members of formation

The trajectory planning for the members of formation works similarly as the trajectory planning for the virtual structure of the whole formation with few differences. The trajectory is also coded into a vector, but the number of elements for the individual robots is only given by N. These N elements have fixed duration time Δt_N. There is no need for additional M elements, because the trajectory plan of individual robots is only local, and does not include the global trajectory to the goal. The vector for the trajectory is depicted on Fig. 12. It is shown, that the trajectory element constists of velocity vector components and the time duration of the element, which is fixed for the formation members.

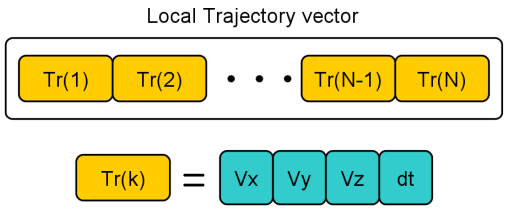


Figure 12 The illustration of the vector representing the trajectory of the formation members.

The robot is represented by sphere with radius r_q for use in calculations. The initialization of the trajectory is also different from the global trajectory initialization and it is described in section 5.1. The optimization of the trajectory uses modified objective and constraint functions, described in section 5.2.

5.1 Initialization of the trajectory

The initialization of the members trajectory does not include any path planning algorithm as the initialization for the virtual structure, described in previous chapter in section 4.1. After the initial trajectory estimate is known for the whole virtual structure, the optimization for it is run with resulting optimal trajectory for given optimization parameters, objective function and constraint functions. Part of this optimized trajectory with N constant duration time Δt_N is used directly as initial trajectory for the formation members, as it is shown on Fig. 13.

This initialization is done only for first time, before the optimization for the formation members is done. Throughout the flight of the formation, the starting trajectory for the optimization process is taken from the previous resulting trajectory without the first control input, that was used for robot control. The missing last element is added with velocity $v = [0, 0, 0]^T$ and constant duration time Δt_N.

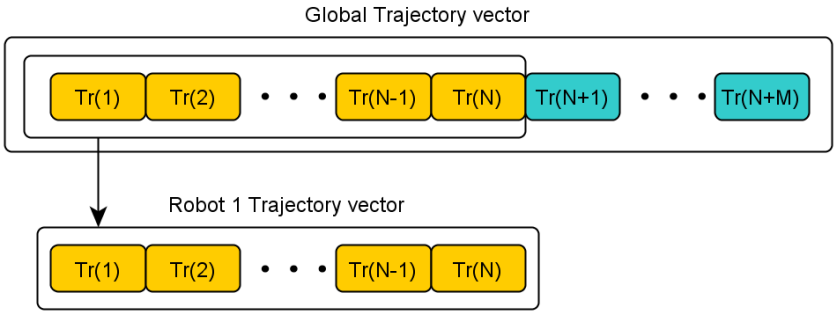


Figure 13 The illustration of the initialization of the formation members trajectory from the global trajectory.

5.2 Trajectory optimization

The optimization of the trajectory is done in the same way as for the whole virtual structure. There are few modifications in objective function and constraint functions, described in section 5.2.1 and 5.2.2 respectively.

5.2.1 Objective function

Objective function for the formation members is also combination of several components with different user-defined weights. This allows the user to adjust the optimization for deployment of the system in different environment with different conditions. The components of the objective function are:

Obstacle avoidance component of the objective function is designed similarly as for the virtual structure with addition of the collision avoidance between the formation members. The part of this component to avoid collisions with obstacles is the same as for the virtual structure with different safety and critical radiuses $r_{s,q}$ and $r_{a,q}$ respectively. These critical radius has to be chosen to satisfy condition $r_{a,q} > r_q$, where the r_q is radius of the sphere representing the quadrotor robot. The safety radius determines how close the trajectory can be to obstacles without any penalization. The safety radius has to be chosen with respect to possible inaccuracies of the predicted trajectory given by inaccuracies of the model and the quality of the quadrotor controller.

The part of this component that serves as collision avoidance between formation members is using the known trajectory of each member, finds the closest distance of current trajectory and the trajectories of other robots and use the same transformation function with critical and safety radiuses. The whole obstacle avoidance component of the objective function can be written as:

$$J_{obst,q} = \max \left[\max_{i=1\ldots n_o} \left(min \left\{ 0, \frac{dist(T_q, o_i) - r_{s,VS}}{dist(T_q, o_i) - r_{a,VS}} \right\} \right)^2, \max_{j=1\ldots n_q, j\neq q} \left(min \left\{ 0, \frac{dist_{traj}(T_q, T_j) - r_{s,VS}}{dist_{traj}(T_q, T_j) - r_{a,VS}} \right\} \right)^2 \right], \quad (13)$$

where the q is the index of the current robot and the $dist_{traj}(T_q, T_j)$ denote the function, that determines the distance of two trajectories for robot given by index j and current robot.

The dynamic obstacles collision avoidance is ensured by the function $dist(Traj, o_i)$, where also the distance from the moving obstacles is calculated, as it is described in section 4.2.3.

Formation keeping component of the objective function is used for keeping the prescribed position in formation. The position is given relative to the virtual structure center for every member of the formation. The formation keeping component is designed to penalize every deviation from the desired position. The total penalty for the virtual structure trajectory and current robot trajectory is given as the sum of the deviations along the trajectory. This can be written as:

$$J_{form,q} = dev(T_{VS}, T_q, \vec{p_{f}}_q) \tag{14}$$

where the $dev(T_{VS}, T_q, \vec{p_{f}}_q)$ function returns the total sum of deviations between the trajectory of virtual structure T_{VS}, trajectory of the current robot T_q and given relative position given by vector $\vec{p_{f}}_q$.

Control smoothness component is designed to prefer the smooth trajectories over the trajectories with a lot of immediate changes of direction. This component helps to create straighter trajectories that require less energy which is important especially for small quadrotor UAVs with very limited battery. The smoothness of the trajectory is defined as the sum of the control input changes along the trajectory. This can be written as:

$$J_{smooth,q} = \sum_{i=2}^{N} \parallel v_i - v_{i-1} \parallel \tag{15}$$

5.2.2 Constraint function

Several constraint functions are used for formation members trajectory optimization to add limitations for feasible trajectories. The trajectory is feasible if the condition $C < 0$ is satisfied. The constraint function for collision avoidance is used the same way as in case for the whole virtual structure with addition of other members collision avoidance.

Obstacles avoidance constraint uses the minimal distance from the robot's trajectory to a obstacle or the minimal distance from another robot in formation, whatever is smaller, and if minimal distance is less than the critical radius defined for the robot, the trajectory is unfeasible. The objective function is defined as:

$$C_{obst,q} = r_{a,q} - \min \left[\min_{i=1\ldots n_o} \left(dist(T_q, o_i) \right), \min_{j=1\ldots n_q, j\neq q} \left(dist(T_q, T_j) \right) \right]. \tag{16}$$

This constraint ensures that the robot will not collide with any object and any other robot in formation.

Control limits for the formation members are given by limits of the robots used and their dynamic and kinematic capabilities. The limit values of the velocities, used as control inputs for robots, have to be large enough for robots to be able to recover their position in formation in case they had to divert from the desired position, e.g.

due to collision avoidance maneuver. The limits are implemented as set of constraint functions as follows:

$$C_{lim,q} = \begin{pmatrix} v_{xq} - v_{xq,max} \\ v_{y_q} - v_{xq,max} \\ v_{zq} - v_{xq,max} \\ v_{xq,min} - v_{xq} \\ v_{y_{q,min}} - v_{y_q} \\ v_{zq,min} - v_{zq} \end{pmatrix}. \tag{17}$$

6 Quadrocopter control

As it is stated in chapter 2, the UAV's control has to be designed to take the desired velocity as input, because constant control inputs have to be used to encode the trajectory to vector of control inputs. The controller determines the velocities of the four rotors according to actual state of quadrocopter and the desired velocity.

The controller, that satisfies the requirements, is based on geometric tracking control of a quadrotor UAV introduced in [26]. In that paper, a nonlinear tracking controller is developed on the special Euclidean group SE(3) to enable the UAV to follow a predefined trajectory of the desired location of the center of mass and the direction of the first body-fixed axis, the heading. It is shown by simulations, that the complex and acrobatic maneuvers can be done with the geometric tracking control. The original controller is modified in order to track the velocity of the center of mass instead of position and the heading is fixed to keep the number of variable input parameters low for the optimization process. The controller is based on the quadrocopter dynamics model presented in [26], which is recapitulated in following chapter.

6.1 Quadrocopter dynamic model

The quadrotor UAV is depicted in Fig. 14 with the global inertial frame $\{i_1, i_2, i_3\}$ and the quadrocopter body-fixed frame $\{b_1, b_2, b_3\}$ shown, where the center of mass is the origin of the body-fixed frame.

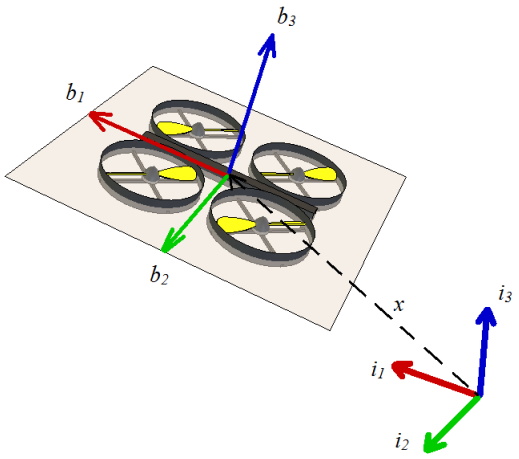


Figure 14 Global inertial frame and a quadrocopter body fixed frame

In this thesis, the direction of the third axes of both, the global inertial frame and body-fixed frame, are selected to point upwards to be consistent with the simulation software V-REP used for the controller verification and final experiments simulation. The first two axes of the body fixed frame b_1, b_2 lies in the plane defined by the centers

of the propellers shown on Fig. 14 as light grey plane. The third axis b_3 is normal to the plane and points the same direction as the total thrust created by all propellers.

The equations of motion for the quadrotor UAV [26]:

$$\dot{x} = v, \tag{18}$$
$$m\dot{v} = mge_3 - fRe_3, \tag{19}$$
$$\dot{R} = R\hat{\Omega}, \tag{20}$$
$$J\dot{\Omega} + \Omega \times J\Omega = M, \tag{21}$$

where

$m \in \mathbb{R}$	the total mass
$J \in \mathbb{R}^3$	the inertia matrix with respect to the body-fixed frame
$R \in SO(3)$	the rotation matrix from the body-fixed frame to the inertial frame
$\Omega \in \mathbb{R}^3$	the angular velocity in the body-fixed frame
$x \in \mathbb{R}^3$	the location of the center of mass in the inertial frame
$v \in \mathbb{R}^3$	the velocity of the center of mass in the inertial frame
$d \in \mathbb{R}$	the distance from the center of mass to the center of each rotor
$fi \in \mathbb{R}$	the thrust generated by the i-th propeller along the $\vec{b_3}$ axis
$\tau_i \in \mathbb{R}$	the torque generated by the i-th propeller about the b3 axis
$f \in \mathbb{R}$	the total thrust, i.e., $f = \sum_{i=1}^{4} f_i$
$M \in \mathbb{R}^3$	the total moment in the body-fixed frame

The total thrust f and the total moment M can be determined from thrusts of the individual propellers as

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -d & 0 & d \\ d & 0 & -d & 0 \\ -c_{\tau f} & c_{\tau f} & -c_{\tau f} & c_{\tau f} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{22}$$

This matrix equation can be also used to determine the thrust of the individual propellers based on total thrust f and the total moment M if the following constraints are satisfied: $d \neq 0$ and $c_{\tau f} \neq 0$. The details can be found in [26].

6.2 Quadrocopter controller

As already stated, the geometric tracking controller presented in [26] is modified to follow the velocity of the center of mass, the $v_d(t) = [v_{xd}, v_{y_d}, v_{zd}]^T$ and the desired heading, which is the direction of the first body-fixed axis $b_{1d}(t)$ is fixed to keep the number of input variables low, as the length of the optimization vector increases with the number of variables to optimize.

The quadrotor UAV is able to produce force only in the direction of its third body-fixed frame b_3 and this force determines the translational dynamics of the quadrocopter. This force is given as fRe_3, where the force of the total thrust f is directly controlled. The Re_3 is the third body-fixed axis b_3. The desired orientation of the third body-fixed axis b_{3d} is determined from the current velocity v and the desired velocity v_d. To specify the desired attitude R_d, the direction of one more axis has to be determined.

The desired heading b_{1d} is fixed vector perpendicular to global third axis i_3 and is fixed in the modified controller, i.e. $\vec{b_{1d}} = 1, 0, 0]^T$. The second axis of the desired attitude b_{2d} is found as the cross product of b_{3d} and b_{1d} normalized to size of 1. The first axis of the desired attitude is then found by the cross product of b_{2d} and b_{3d}. The desired attitude can be written as $R_d = [\vec{b_{2d}} \times \vec{b_{3d}}, \vec{b_{2d}}, \vec{b_{3d}}]$, where $\vec{b_{2d}} = (\vec{b_{3d}} \times \vec{b_{1d}}) / \parallel \vec{b_{3d}} \times \vec{b_{1d}} \parallel$. The total thrust f is selected according to desired vertical speed and current attitude. The controller scheme is depicted on Fig. 15.

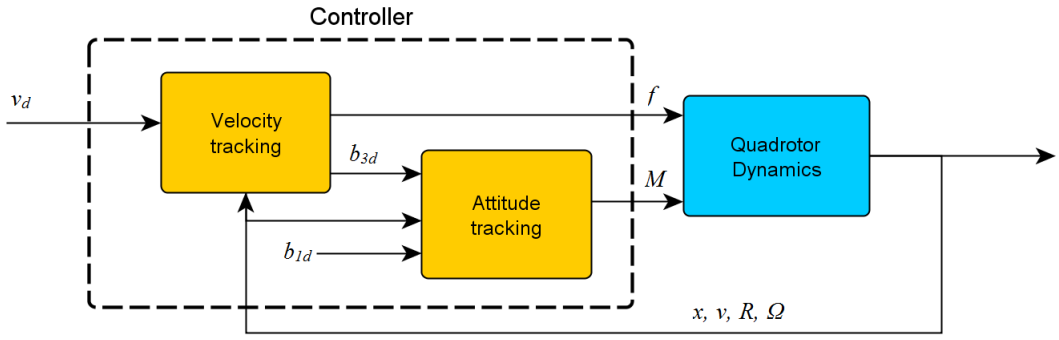


Figure 15 Controller scheme

The tracking errors for the controller are defined for v, R, Ω. These tracking errors are discussed and the process of choosing them is shown in detail in [26]. These tracking errors are used unmodified. The only change is that originally presented tracking error for position e_x is not used as the controller is re-designed to track only velocity. The tracking error for velocity is:

$$e_v = v - v_d \tag{23}$$

where the v is current velocity and the v_d is desired velocity. The tracking error for attitude is defined as:

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d) \tag{24}$$

where R_d is desired attitude and R is current attitude of the quadrocopter. The tracking error for the angular velocity is as follows:

$$e_\Omega = \Omega - R^T R_d \Omega_d \tag{25}$$

The desired direction of the third body-fixed axis b_{3d} is defined for velocity command $v_d(t)$ as follows:

$$\vec{b_{3d}} = \frac{-k_v e_v + mge_3}{\parallel -k_v e_v + mge_3 \parallel} \tag{26}$$

where it is assumed that

$$\parallel -k_v e_v + mge_3 \parallel \neq 0 \tag{27}$$

The desired velocity will be constant for the time interval of an optimization step. The desired acceleration will be 0 for this interval and therefore it is not used in the calculations.

The total thrust is determined to satisfy the desired vertical force, which is given as the third coordinate of the $\vec{b_{3d}}$, for the current attitude R. The desired vertical force is defined as follows:

$$F_{zd} = (-k_v e_{vz} + mg) \tag{28}$$

the total force in global coordinates is:

$$F = fRe_3 \tag{29}$$

therefore the total thrust is given by:

$$f = (-k_v e_{vz} + mg)/R_{3,3} \tag{30}$$

The moment M is determined using the attitude and angular velocity tracking errors as stated in [26]:

$$M = -k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) \tag{31}$$

The control moment M corresponds to the attitude tracking part of controller on Fig. 15 and the thrust f corresponds with the velocity tracking part.

6.3 Verification of the controller

The proposed modification of the geometric tracking controller for the quadrotor UAV is verified by simulations in V-REP software - the Virtual Robot Experimentation Platform made by Coppelia Robotics [27]. The model of the quadrocopter used for simulation is courtesy of Eric Rohmer, the propellers are courtesy of Lyall Randell. The proposed geometric tracking controller for quadrocopter velocity control is implemented in Lua script for the quadrocopter model in V-REP software. The V-REP is using the bullet physics engine to simulate dynamics and object collisions.

The parameters of the quadrotor UAV are chosen the same as in [26].

$$J = \begin{bmatrix} 0.0820 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{bmatrix} kg \cdot m^2$$

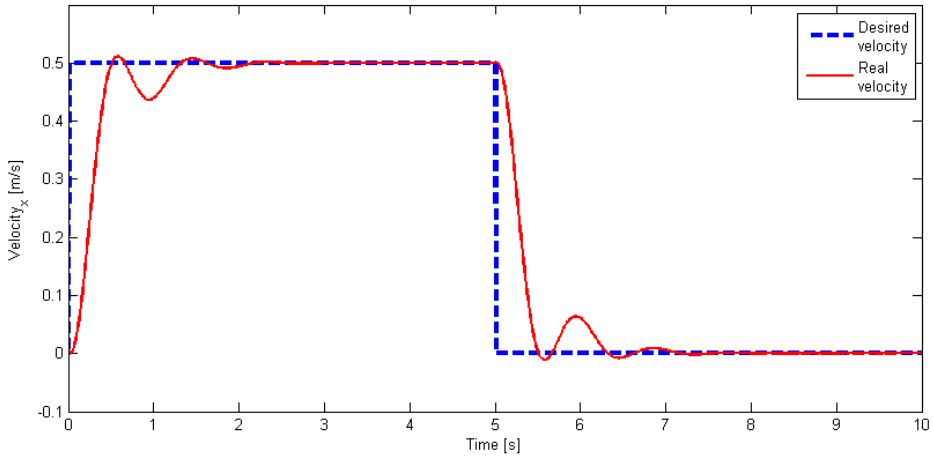
$$m = 4.34\ kg, \quad d = 0.315\ m$$

The controller parameters used in work [26] provide quality control of the quadrocopter, therefore the same values are used:

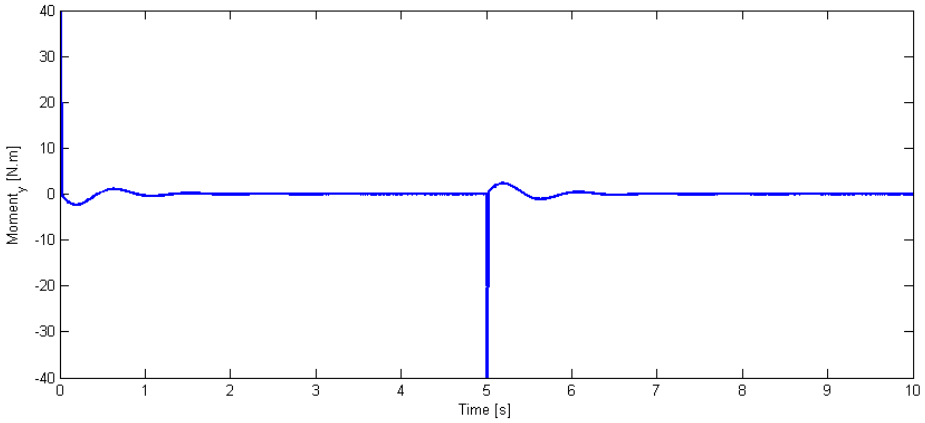
$$k_v = 5.6 \cdot m, \quad k_R = 8.81, \quad k_\Omega = 2.54.$$

Because the velocity controller is used in conjunction with MPC trajectory planning, where the trajectory consists of constant velocity elements, the proposed controller is verified on step functions of the desired velocity. The response to the step in desired velocity in direction of x-axis, $v_{x,des}$, is shown on Fig. 16 with the angular momentum of the quadrocopter measured around the local y-axis and the total thrust f of all four propellers.

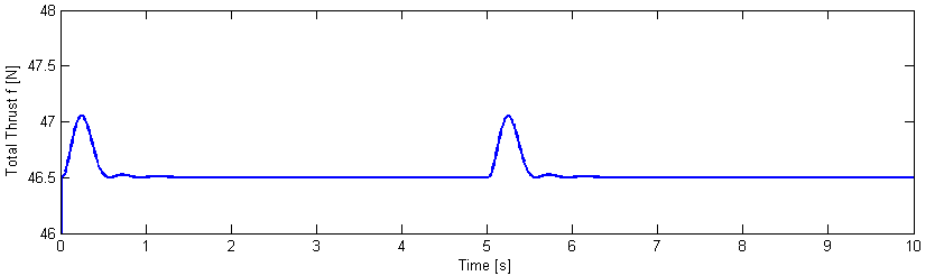
Note the oscillations that are caused by nonlinearities, specifically the limited thrust of the quadrocopter rotors and therefore limited moment produced by the quadrocopter.



a) The velocity v_x response to step.



b) The angular momentum M_y response to step.

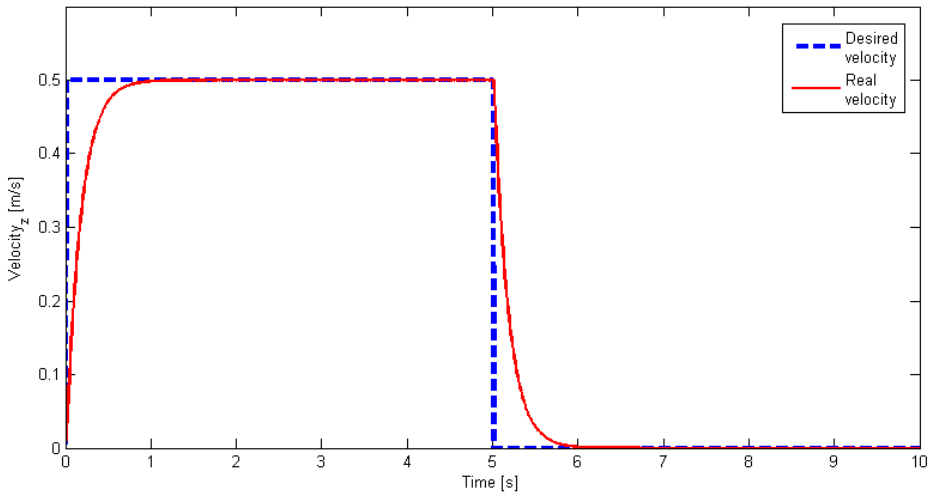


c) The total thrust f response to step.

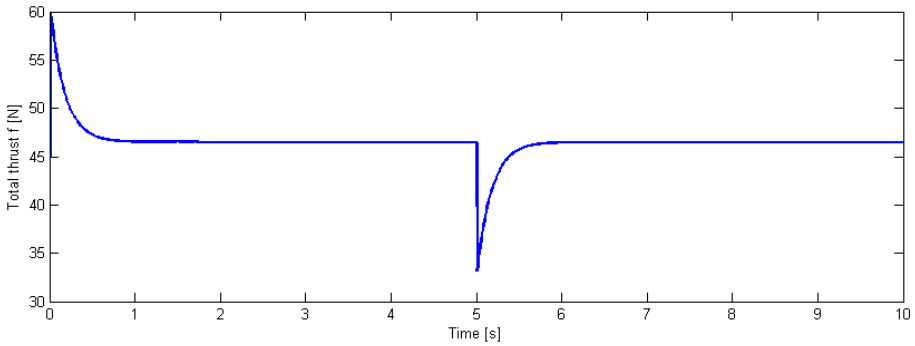
Figure 16 The response to the step in desired velocity of $v_{x,des}$. $v_{x,des}(t) = 0.5\ m{\cdot}s^{-1}, t \in < 0, 5)$ and $v_{x,des}(t) = 0\ m \cdot s^{-1}, t \in < 5, 10)$.

The step response in y-axis is the same as for x-axis, because the controller is controlling the velocity in whole $x - y$ plane the same way, by controlling the attitude of the quadrotor robot. A bit different situation is in controlling the z-axis velocity, because this component of the velocity vector is controlled by thrust of all quadrocopter propellers. The graph on Fig. 17 shows the response of the quadrocopter to the step of the desired velocity $v_{z,des}$.

Next graph on Fig. 18 shows the velocity tracking of the quadrotor UAV controller



a) The velocity v_z response to step.



b) The total thrust f generated as response to step.

Figure 17 The response to the step in desired velocity of $v_{z,des}$. $v_{z,des}(t) = 0.5\ m{\cdot}s^{-1}, t \in < 0, 5)$ and $v_{z,des}(t) = 0\ m \cdot s^{-1}, t \in < 5, 10)$.

for velocity input similar to the output of the MPC algorithm after optimization, that means the desired velocity is changing every time step, which is set to $\Delta t = 0.2\ s$. The changes in desired velocity are assumed to be small with possible exceptions, where the immediate change of the movement is necessary, e.g. collision avoidance, therefore the testing trajectory vector is chosen accordingly with small changes in velocity with the large change in velocity in time $t = 1.8\ s$, which corresponds with the control input at $v[10]$. The testing trajectory is defined by the following vectors:

$$v_x = 0.1 \cdot [4,\ 5,\ 5,\ 4,\ 4,\ 3,\ 2,\ 2,\ 1,\ -5,\ -5,\ -4,\ -4,\ -4,\ -2,\ -2,\ -1,\ -1,\ -1,\ 0]$$
$$v_y = 0.1 \cdot [0,\ 2,\ 2,\ 2,\ 2,\ 3,\ 4,\ 5,\ 5,\ -1,\ 0,\ 0,\ 0,\ -1,\ -2,\ -2,\ -2,\ -2,\ -1,\ 0]$$
$$v_z = 0.1 \cdot [0,\ 1,\ 1,\ 1,\ 1,\ 2,\ 1,\ 0,\ 0,\ -5,\ -5,\ -4,\ -3,\ 5,\ 5,\ 4,\ 4,\ 4,\ 0,\ 0]$$

The resulting trajectory is illustrated on Fig. 20, where the trajectory is shown as green bold line with starting point depicted as blue circle. The trajectory projections are shown as black lines.

6.3.1 The effects of airflow from the propellers on other quadrocopters

One of the objectives of this thesis is to study the influence of the airflow from the quadrocopter's propellers to other quadrocopters. This is allowed by V-REP simulation software [27] and the model of the quadrocopter, made by Eric Rohmer, with propellers, that simulates the airflow, that have been made by Lyall Randell.

During the experiments in simulation, it is assumed, that the formation of the autonomous quadrotor UAVs is always shaped in a way, that no quadrocopter will influence the other one during the whole flight. Therefore, the effects of airflow from the propellers are studied only as transient processes.

The testing environment for the airflow influence study consists of two same quadrocopters Q_1, Q_2 with straight trajectories in x-axis direction and one hovering quadrocopter Q_H. The quadrocopter Q_1 is used as reference, the trajectory of the second quadrocopter Q_2 leads under the hovering quadrocopter Q_H and the effects of the airflow from the hovering quadrocopter to the quadrocopter Q_2 is studied. The velocities of the quadrocopters Q_1, Q_2 are shown on Fig. 21 and the trajectories of both quadrocopters are shown on Fig. 22. The testing environment in V-REP with the airflow illustrated by light blue dots is shown on Fig. 23.

It can be seen on the figures 21, 22 and 23, that the effect of the airflow from the propellers to the trajectory of the influenced UAV is not very significant, as the controller tries to compensate the changing vertical speed, but on the other hand, it cannot be neglected. As it is shown in section 7.3, the small changes in position, caused by the airflow, are compensated by the MPC method, where the optimization of the trajectory leads to recovery of the desired position in formation.

6.3.2 Comparison with the simplified model used for MPC

As it is stated in section 3.3, the simplified model for use during the optimization of the formation members' trajectory is used to speed up the calculations. In this section, the experiments in simulation are performed to evaluate the accuracy of the simplified model and to determine the best proportional constant k_v for the simplified model.

The comparison is firstly done on simple step function for desired velocity v_x and the same step function for desired vertical velocity v_z, because the dynamics of the quadrocopter in XY-plane and in vertical direction is different. This comparison is shown on figures 24 and 25.

From the comparison of the velocities in direction of the global x-axis shown on Fig. 24, it can be seen, that good approximation of the real velocity is provided by the simplified model with proportional constant $k_v \simeq 5$. Important note is that the simple model used for quadrocopter's velocity prediction doesn't model the nonlinearities, propellers' limitations and complex dynamics of the real quadrocopter to keep the calculations as fast as possible, therefore it cannot follow the real velocity response much more accurately.

The effects of the missing nonlinearities, propellers' limitations and complex dynamics of the real quadrocopter, in the simplified model are much less evident in comparison of the velocities in vertical direction. As it was said in the beginning of the section 6.3, the vertical velocity is controlled by total thrust only, where the acceleration is directly proportional to the thrust, if using the thrust values in operational range. Therefore the simplified model with proportional controller approximates the vertical velocity very accurately with proportional constant tuned to $k_v = 5.5$, as it can be seen on Fig. 25. For all the following experiments, the proportional constant used in the simplified

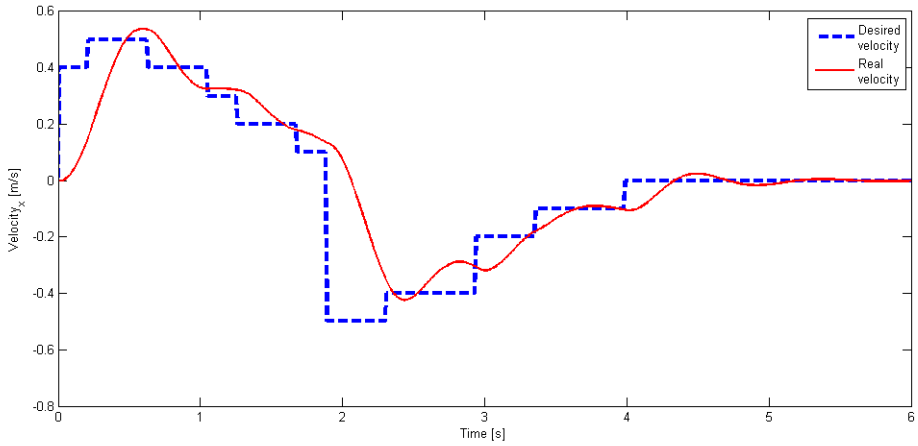
model is set to $k_v = 5.5$, because the approximation of the velocity in XY-plane is still good enough for the use in MPC method and the vertical velocity is approximated very accurately.

Next, the comparison is performed for the testing trajectory vector used in the beginning of the section 6.3. The comparison between the real/simulated velocity and the velocity predicted by simplified model from section 3.3 in all directions is shown on Fig. 26. Finally, the comparison of the resulting trajectories is presented on Fig. 27.

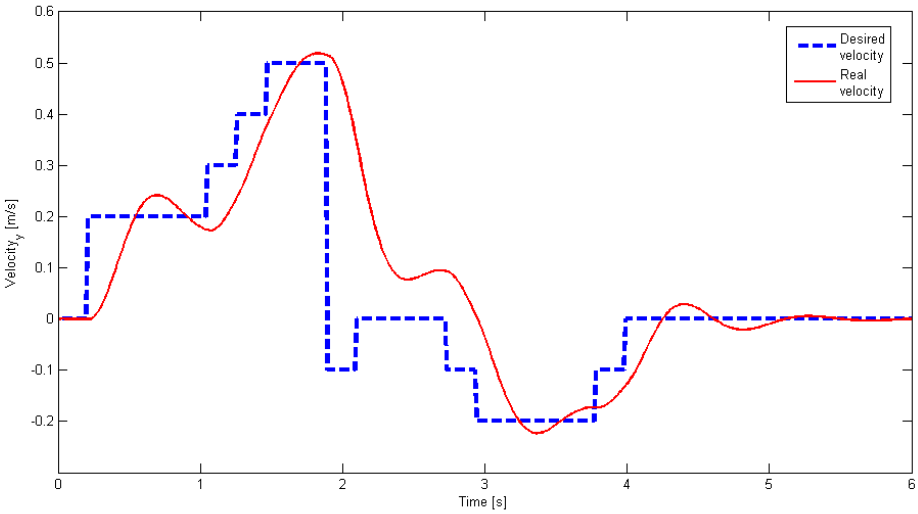
Again, it can be seen that the approximation, or better the prediction, of the velocity in XY-plane shown on figure 26 (a) and b)) is close to the real velocity with small deviations for most of the time. The largest deviations can be seen when large changes of the velocity is desired. It is assumed that such a situations occur only once in a while, because the small changes in velocities are preferred in optimization process.

The prediction of the vertical velocity is very accurate, shown on 26, and only expected deviations can occur due to the airflow influence from the other quadrocopters or due to unmodeled aerodynamics effects, such as ground effect when flying close to the ground or due to any other effects of the environment.

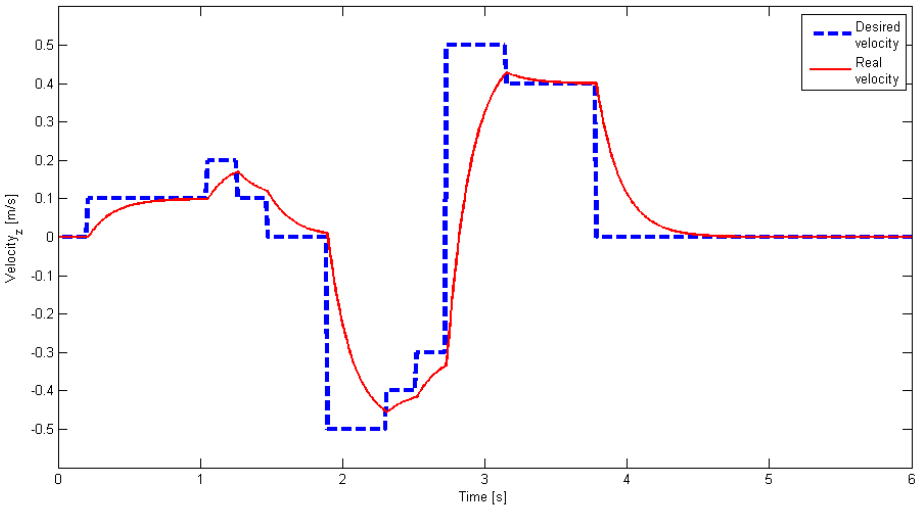
From the comparison of the resulting trajectories of the real/simulated quadrotor UAV and the simplified model on Fig. 27, it is evident, that the deviations between the trajectories increase with the time of the prediction, as the inaccuracies of the model cumulate. This is expected behavior and it is compensated by the nature of the model predictive control used for the trajectory planning, where the optimization of the trajectory is repeated every time step after applying the first control input from the trajectory (more detailed description of the MPC method is in chapter 2.2). The whole trajectory shown on Fig. 27 takes about 6 seconds and the time step of the MPC is chosen $t_{step} = 0.2\ s$. This ensures that the calculated trajectory will not lead the quadrocopter to collision because of the model inaccuracies.



a) The current velocity v_x for testing trajectory vector.

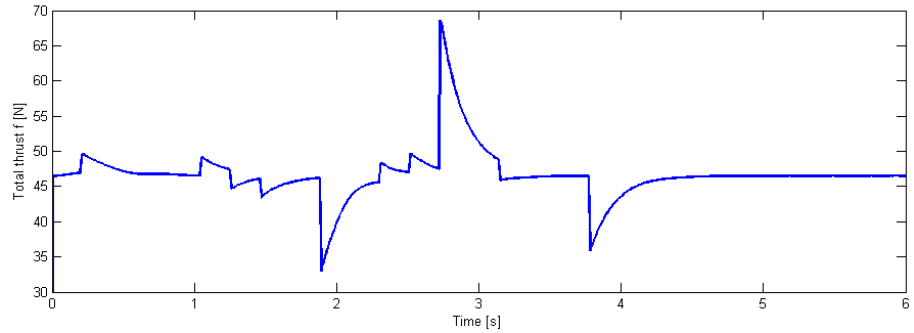


b) The current velocity v_y for testing trajectory vector.

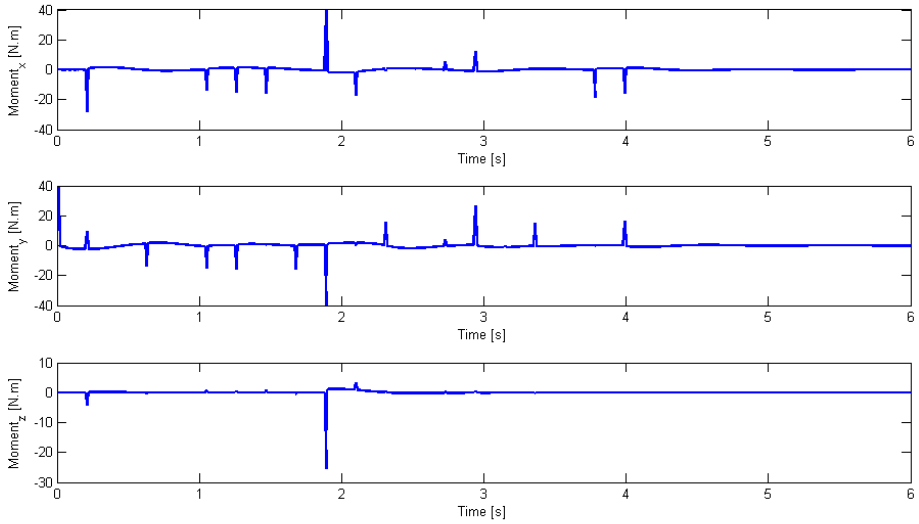


c) The current velocity v_z for testing trajectory vector.

Figure 18 The velocity tracking of the test trajectory vector.



a) The total thrust f for testing trajectory vector.



b) The angular momentums M_x, M_y, M_z for testing trajectory vector.

Figure 19 The total thrust and the angular moments for the test trajectory vector.

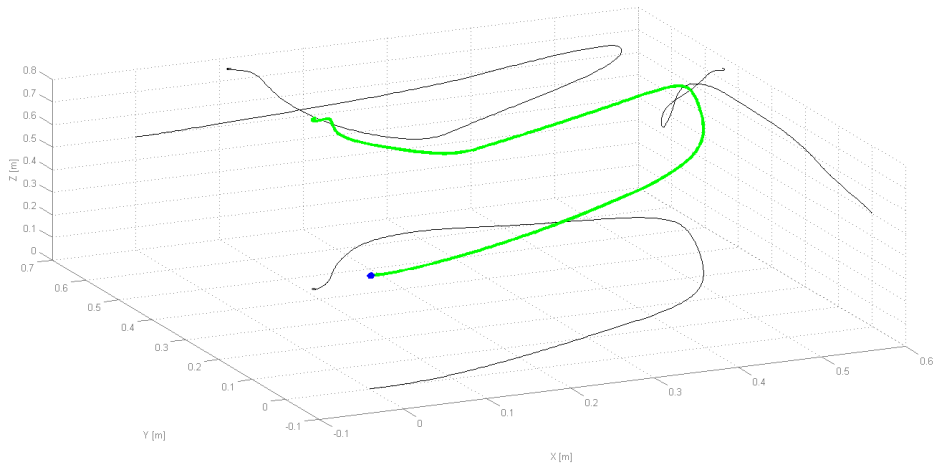


Figure 20 The resulting trajectory for test trajectory vector with shown projections.

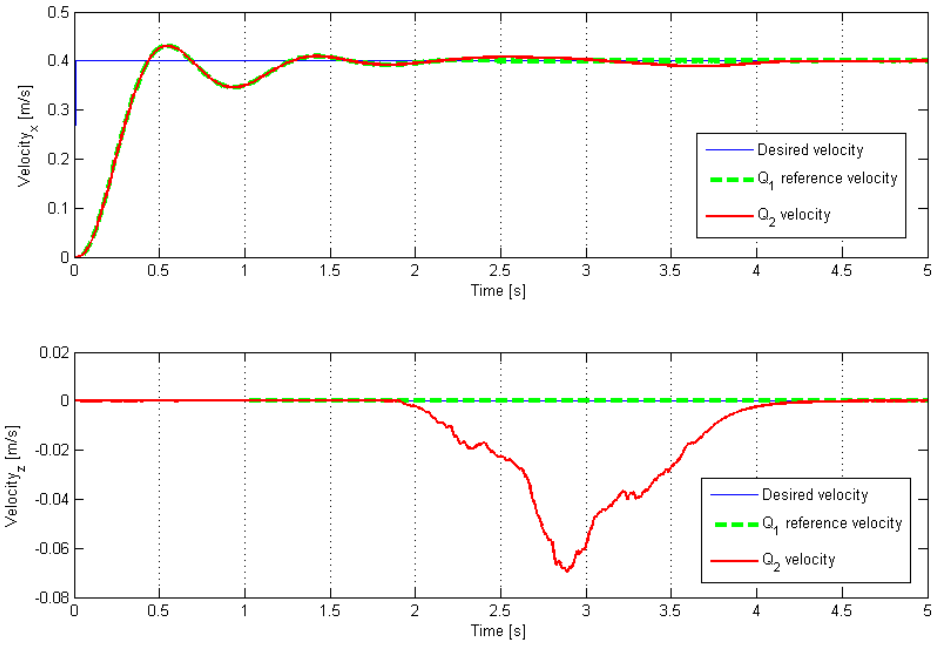


Figure 21 The resulting trajectory for test trajectory vector with shown projections.

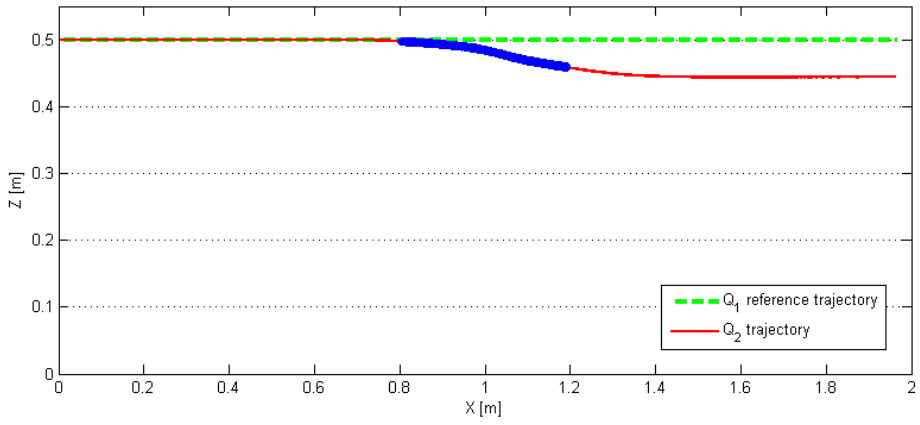


Figure 22 The trajectory of quadrocopters Q_1, Q_2, with the effect of the airflow shown on the quadrocopter Q_2. The area with the airflow applied is highlighted in blue.

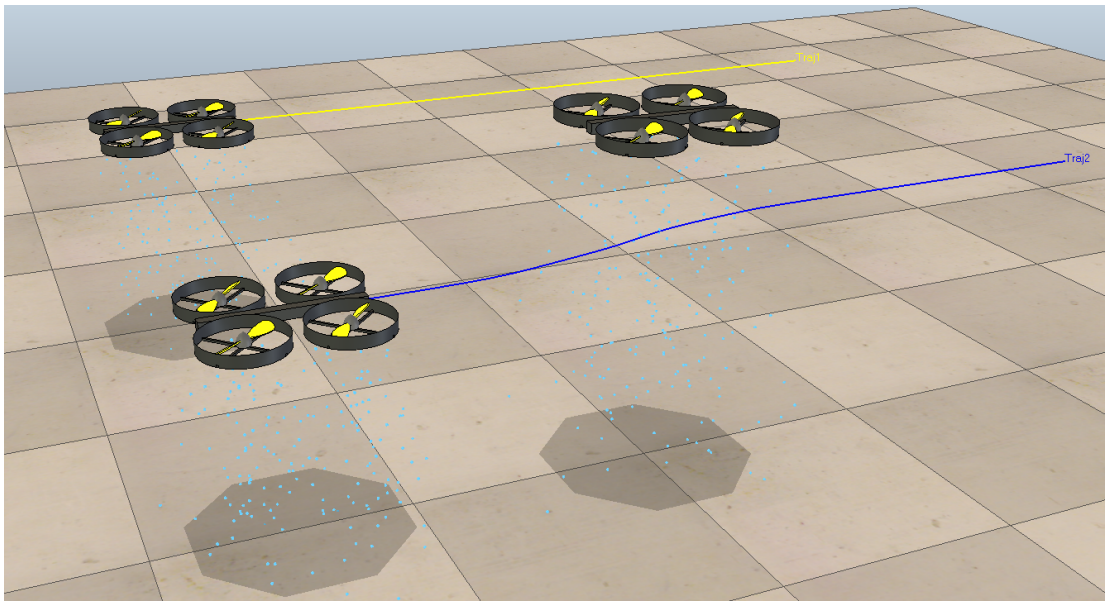


Figure 23 The testing environment setup in V-REP with the airflow illustrated by light blue dots and with trajectories of both quadrocopters, Q_1 in yellow, Q_2 in blue.

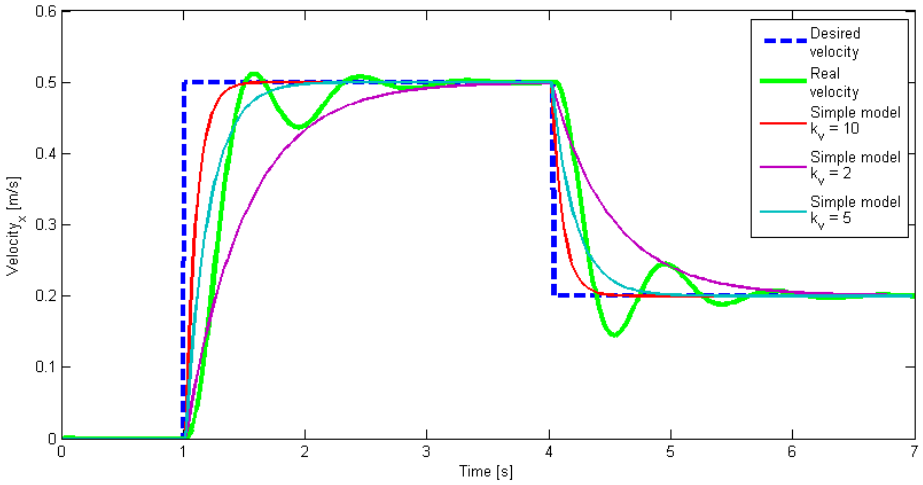


Figure 24 Comparison between the real/simulated velocity v_x and the velocity predicted by the simplified model used in MPC method calculations.

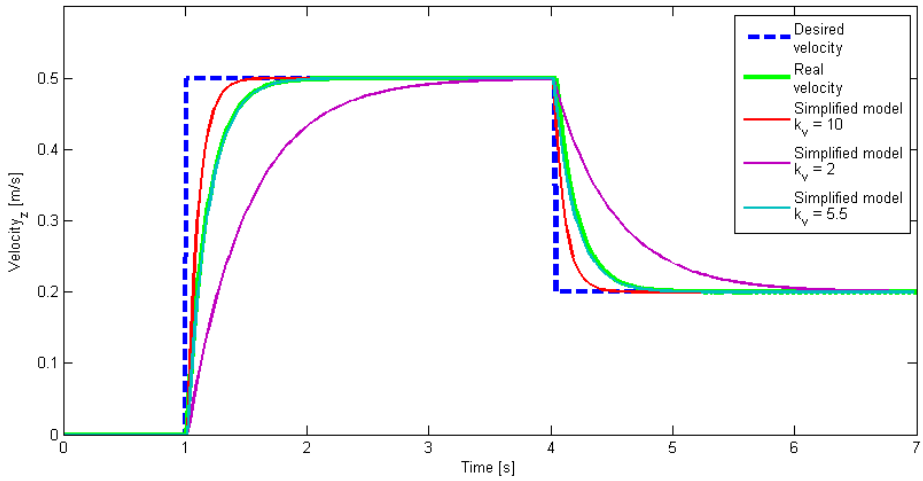
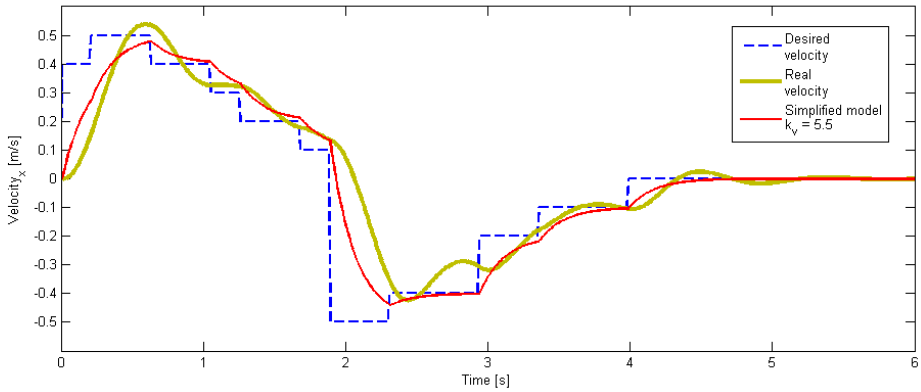
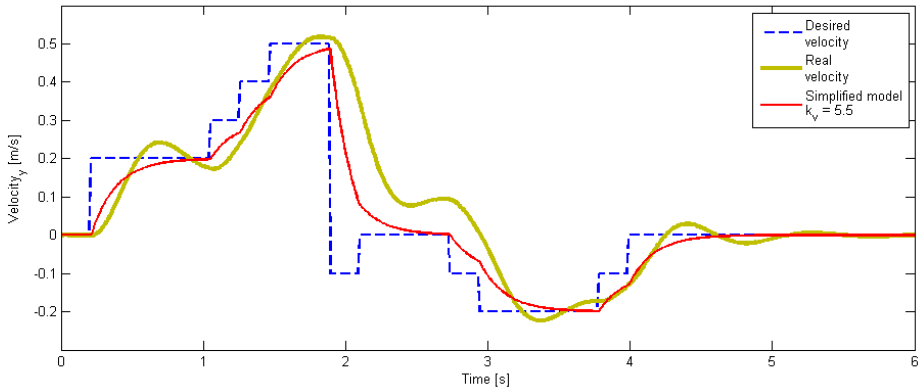


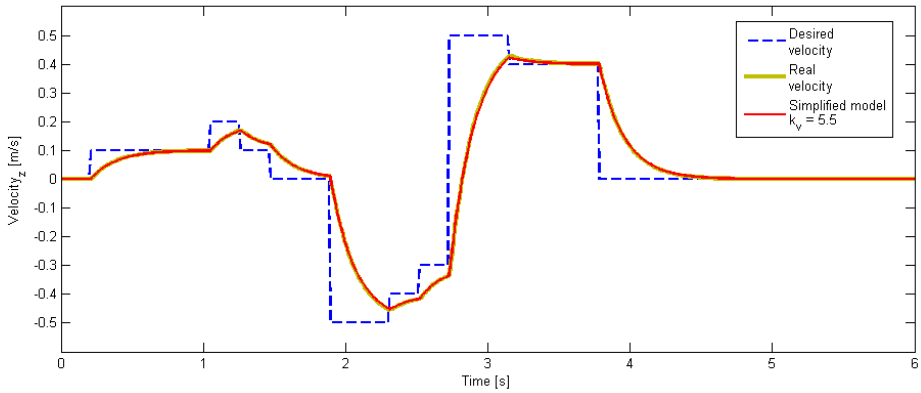
Figure 25 Comparison between the real/simulated vertical velocity v_z and the velocity predicted by the simplified model used in MPC method calculations.



a) Comparison between the real/simulated velocity and the predicted velocity in global x-axis direction.



b) Comparison between the real/simulated velocity and the predicted velocity in global y-axis direction.



c) Comparison between the real/simulated velocity and the predicted velocity in global z-axis direction.

Figure 26 Comparison between the real/simulated velocities and the velocities predicted by the simplified model used in MPC method calculations.

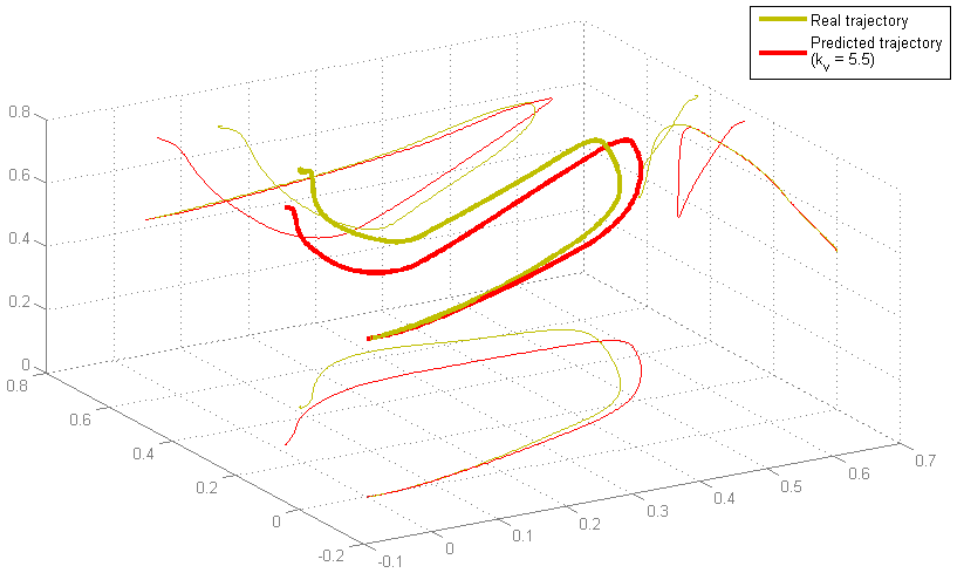


Figure 27 Comparison of resulting trajectories of the real/simulated quadrotor UAV and of the simplified model used in MPC method calculations.

7 Verification of the predictive control and planning

The verification of the complete system of predictive control and planning enabling complex maneuvers of formations of autonomous quadrocopters is provided in this chapter. The verification is done by simulations in the V-REP (virtual robot experimentation platform) simulation software. The overview of the simulation environment is illustrated on Fig. 28. The scene with the quadrocopters, obstacles and the target is created in the V-REP software. The software architecture was designed with focus on easy use of the trajectory planning for real robots. Therefore, the predictive control and trajectory planning described in this thesis was implemented as V-REP plug-in, using the V-REP API to get information about the scene. On the start of the simulation, the map is created based on the objects in the scene. Then, throughout the simulation, the "measurements" of the current objects' and obstacles' positions are done to simulate the real measurements by the robots sensors. All the physics with objects' dynamics, collisions and airflow simulation is done by V-REP bullet physics engine.

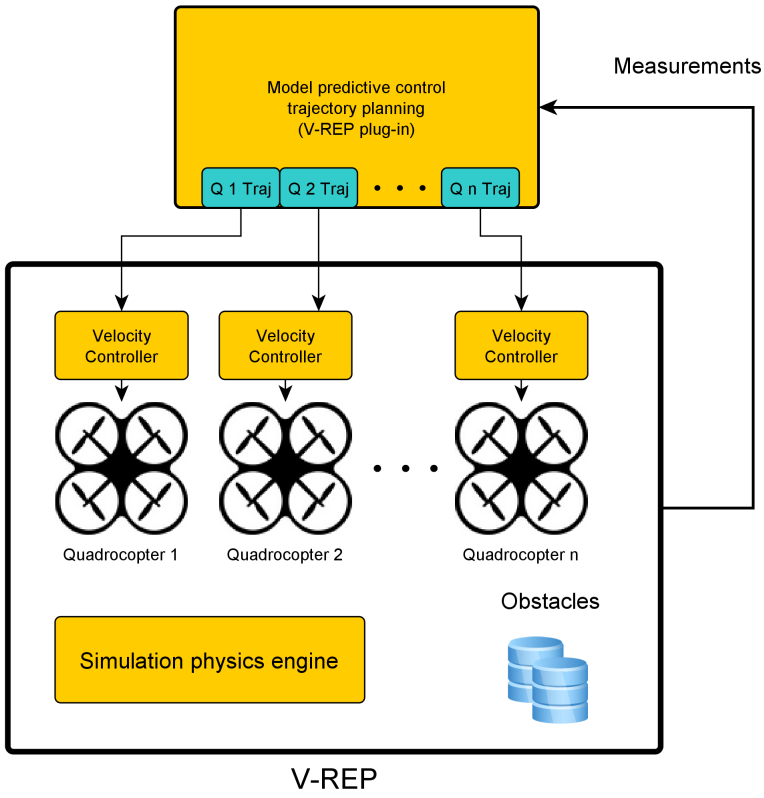
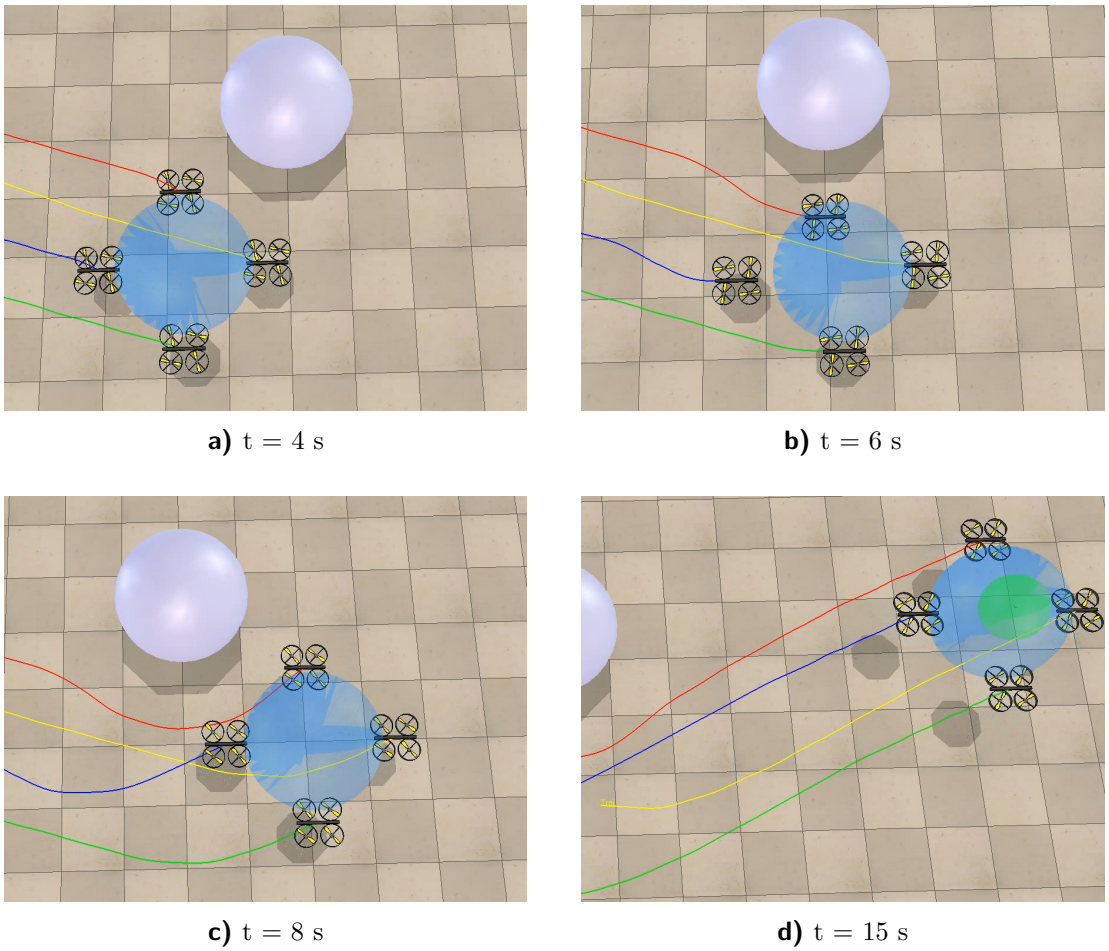


Figure 28 Overview of the simulation environment.

7.1 Collision avoidance

First experiments are done on a simple environment just with one obstacle represented by a sphere with radius $r = 0.5\ m$. The obstacle is located between the start position of the formation and the target region, therefore the formation has to fly around the obstacle to get to the target. The same testing scenario is used first with the safety radius from the obstacles $r_{s,VS} = 1\ m$, which means also the effective radius of the virtual structure is 1 m for the global trajectory planning. The trajectory leads in this case further from the obstacles keeping the deformations of the formation's shape on low level, as shown on Fig. 29. The video from this experiment is on the attached DVD in $Videos/1ObstLargeVs.avi$.

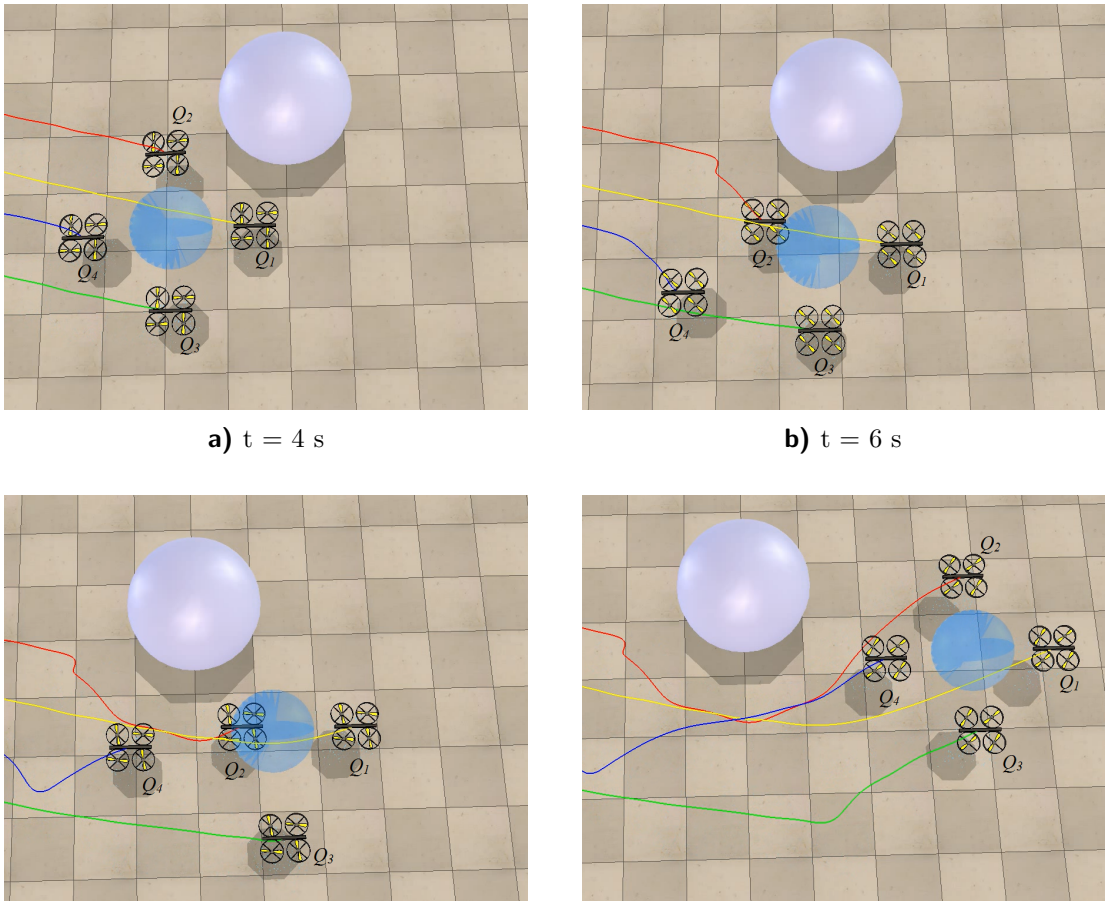


a) t = 4 s **b)** t = 6 s **c)** t = 8 s **d)** t = 15 s

Figure 29 Screenshots from the simulation of the trajectory planning for the formation of diamond shape with large safety radius around obstacles ($r_{s,VS} = 1\ m$). Situations: a) the formation is approaching to the obstacle - no deformation of the formation's shape; b) the formation flies around the obstacle - small deformation of the formation's shape; c) recovery of the formation's shape behind the obstacle; d) arrival of the formation to the target region (green sphere).

The second case with smaller safety radius around obstacles $r_{s,VS} = 0.6\ m$ is shown on Fig. 30. The video from this experiment is on the attached DVD in $Videos/1ObstSmallVs.avi$. It can be seen, that the flexibility of the virtual structure is increased as the trajectory of the virtual structure is passing by closer to the obstacles, causing the formation

members temporarily leave the exact position in the formation to avoid the collision with the obstacle. The desired shape of the formation is recovered in both cases, when the distance from the obstacles is safe.



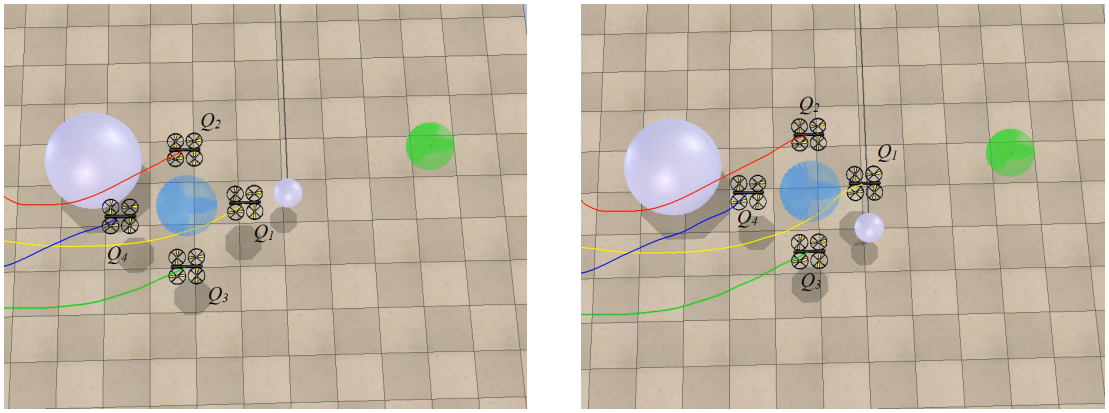
a) t = 4 s **b)** t = 6 s

c) t = 7 s **d)** t = 9 s

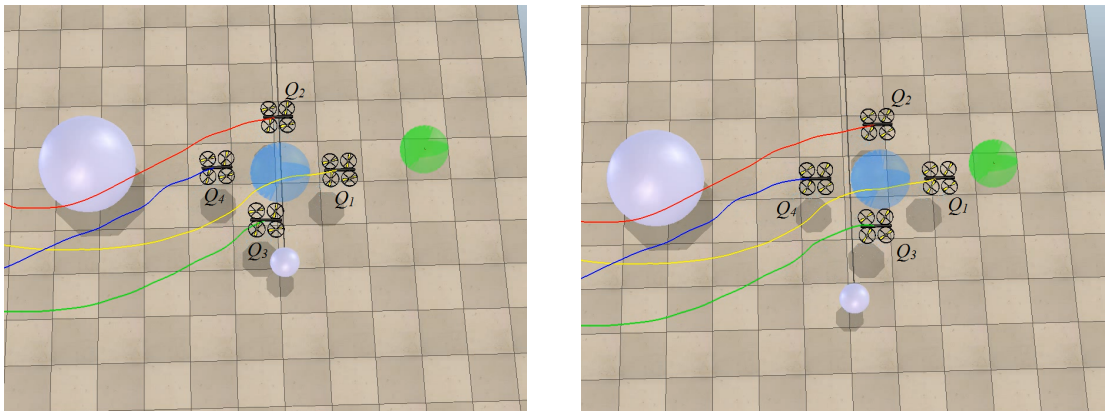
Figure 30 Screenshots from the simulation of the trajectory planning for the formation of diamond shape with small safety radius around obstacles ($r_{s,VS} = 0.6\ m$). Situations: a) the formation is approaching to the obstacle - no deformation of the formation's shape; b) the formation flies around the obstacle - more evident deformation of the formation's shape, the quadrocopter Q_4 is avoiding the collision with the quadrocopter Q_2, which is avoiding the obstacle; c) even more deformed shape of the formation; d) recovery of the desired shape of the formation behind the obstacle.

7.2 Dynamic obstacles

In this section, the experiment showing the capability of collision avoidance with the dynamic obstacles is provided. The collision avoidance for the dynamic obstacles is included in the trajectory planning as described in section 4.2.3. It can be seen from the screenshots shown on Fig. 31, that the formation members avoid the collision with the moving spherical obstacle, by predicting the movement for MPC method. From the side view to the trajectory on Fig. 32, it can be seen that the trajectory leads above the obstacles to avoid the collision. The video from this experiment is on the attached DVD in $Videos/1ObstDynSmallVs.avi$.



a) t = 8 s **b)** t = 9 s



c) t = 10 s **d)** t = 11 s

Figure 31 Screenshots from the simulation of the trajectory planning for the formation of diamond shape with small safety radius around obstacles ($r_{s,VS} = 0.6\ m$) and moving obstacle. Situations: a) the formation is approaching to the moving obstacle; b) the formation flies around the obstacle - the quadrocopter Q_1 is flying above the moving obstacle to avoid collision; c) the formation flies around the obstacle - the quadrocopter Q_3 is flying above the moving obstacle to avoid collision; d) recovery of the desired shape of the formation after passing by the moving obstacle.

7.3 Air flow effects

The effect of the airflow is shown on simple experiment, where two quadrocopters, Q_1 and Q_4, are flying under the quadrocopter, that is creating the airflow. The effect of the airflow can be seen on the Fig. 33 a), where the vertical position is slightly decreasing for about 2 *cm*, but then the desired position is recovered by the trajectory planning algorithm, as expected. It can be said, that the effect of the airflow is compensated by the repeatedly running trajectory optimization.

7.4 Complex maneuvers

Last experiments are done to prove the capability of the complex maneuvers requiring the sudden changes of the velocity of the formation and its members in the proposed and implemented system. Simple scenario with right angle turn is used in combination

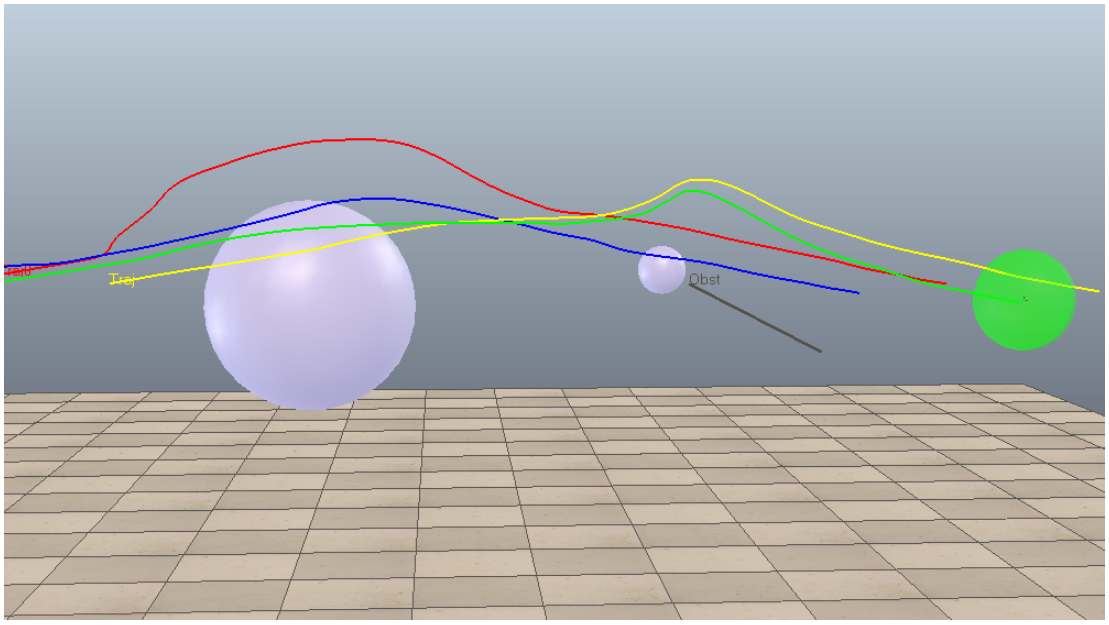
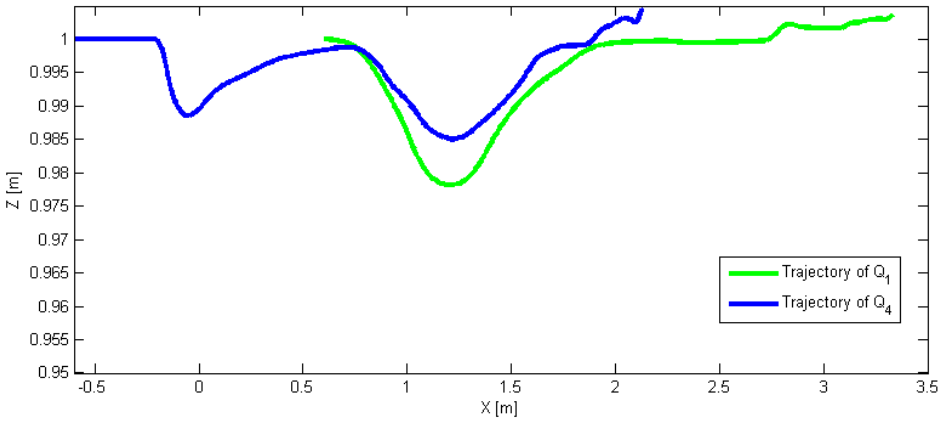


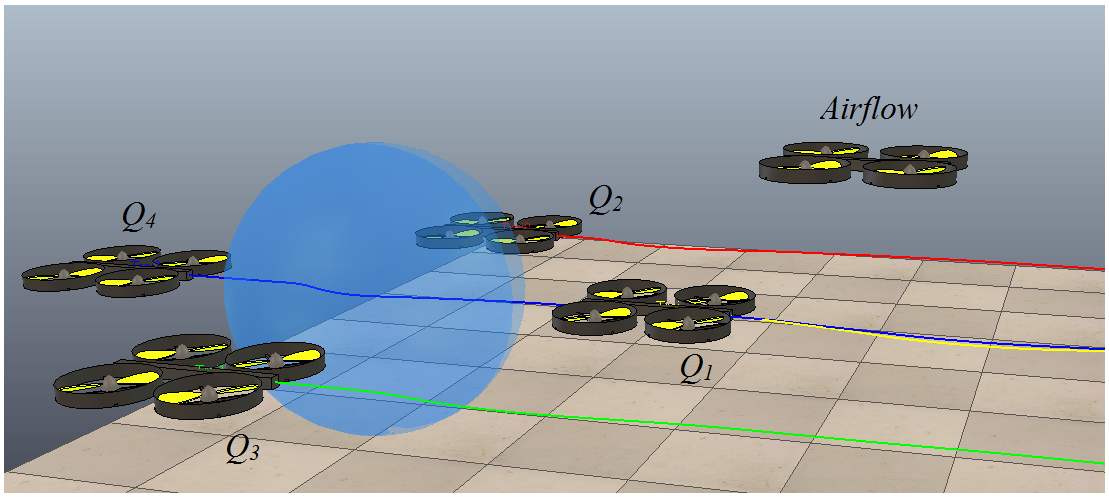
Figure 32 The view from the side to the trajectories of the formation members avoiding the collision with one static and one moving obstacle.

with the change of the formation shape from a diamond shape to a line formation. The result trajectories of the quadrocopters in formation are presented on Fig. 34 by screenshots from the simulation in different times. The video from the simulation is on the attached DVD in $Videos/maneuverTransform.avi$ and the turn without the formation change is in $Videos/maneuver.avi$.

It can be seen from the Fig. 34, that the proposed approach enables the complex maneuvers, that are needed for quick changes of the formation shape or to perform fast on-spot change of movement direction.

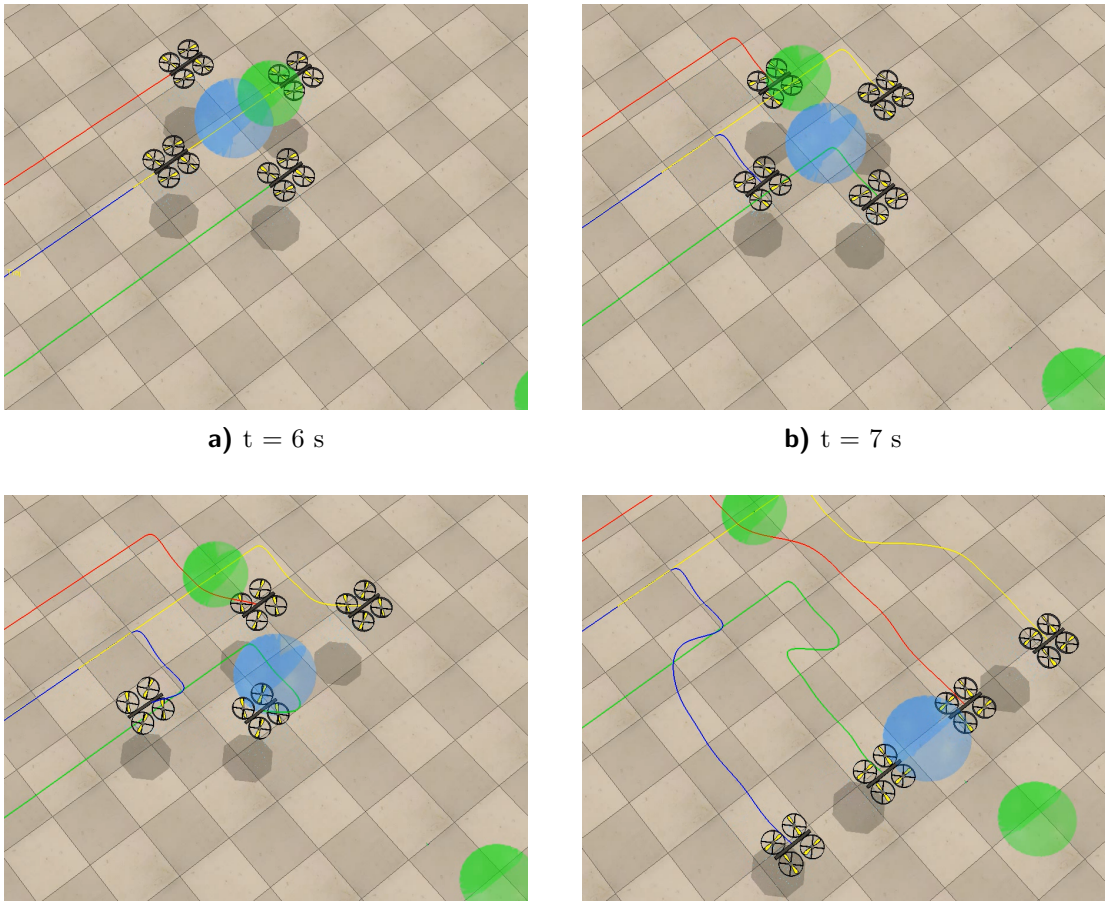


a) The trajectories in XZ-plane of the quadrocopters Q_1 and Q_4 that are flying under the quadrocopter creating the airflow.



b) The setup of the experiment for the airflow effects.

Figure 33 The trajectories from side view (XZ-plane) and the experiment setup in V-REP with illustrated trajectories.



a) t = 6 s

b) t = 7 s

c) t = 8 s

d) t = 10 s

Figure 34 Screenshots from the simulation of the trajectory planning for the formation of quadrocopters performing the right angle turn and changing the shape of the formation on the fly. Situations: a) the formation in diamond shape is arriving to the first waypoint; b) the formation in diamond shape has just performed the right angle turn; c) the formation is changing the shape to a line and still following the global trajectory; d) the change of the formation's shape is completed, continue to target area.

8 Conclusion

The main goal of this thesis was to extend the existing system designed for stabilization of formations of Micro Aerial Vehicles (MAV), presented in [1], specifically designed for the quadrotor Unmanned Aerial Vehicles (UAV), to enable complex maneuvers that require sudden changes of velocity of the group. This goal was accomplished by using the virtual structure approach for the formation representation and trajectory planning, described in chapter 3. Moreover, to enable passing through the narrow entrances or corridors, the flexible virtual structure is used. The majority of the existing trajectory planning system for the formation of UAVs using the model predictive control was utilized with necessary modifications to adapt the flexible virtual structure approach. Next step to allow the complex maneuvers in the formation was done in the trajectory representation by the elements with constant velocity vector, therefore enabling the quadrocopters to move instantly in any direction. In conjunction with the changed representation of the trajectory, the velocity controller for the quadrocopters was designed on the base of the position tracking controller presented in [26].

One of the particular objectives of this thesis was to implement a proper representation of the MAV formation by a convex hull and to design an algorithm that enables a movement of a virtual leader of the formation along the hull. The idea of using the existing leader-follower approach with the convex hull for the formation and design an algorithm that enables a movement of virtual leader along the hull was studied, but during the work on the thesis, the flexible virtual structure based approach was chosen to enable the complex maneuvers of the formation. This decision was made due to the advantages mentioned in section 2.1.3, shortly, it was due to the preserving the low count of the trajectory parameters to optimize and due to easier use of the relative visual localization used for autonomous robots [18].

The next particular objective was to adapt the existing system of motion planning for the new approach enabling the complex maneuvers with changes of formation velocity. As already mentioned, the flexible virtual structure is used instead of the leader-follower approach, therefore the system of motion planning for the formation of the quadrotor UAVs was adapted for use of the virtual structure for the formation control instead of the algorithm enabling the virtual leader migration along the convex hull of the formation.

The presented system using the flexible virtual structure for formation was then integrated into the V-REP (virtual robot experimentation platform) simulation software. The plug-in for V-REP, containing the trajectory planning system for the formation, represented by the flexible virtual structure, and using the MPC method for trajectory planning, was implemented in C++, based on the existing system for leader-follower approach and using the CFSQP library [17] for solving the optimization of the trajectory vector in MPC method. To integrate the system into the V-REP software, the velocity controller, described in chapter 6, was implemented in V-REP using the embedded script of the quadrocopter model. The existing quadrocopter model, made by Eric Rohmer, with the propellers enabling the air flow simulation, made by Lyall Randell, was used for the simulations in V-REP.

Last particular objective was to verify the implemented method in the V-REP simu-

lation engine and to study the influence of mutual interactions between the UAVs in the formation, caused by the air flow from their propellers. This objective was accomplished and the results of several experiments simulated in V-REP, including the experiments to study the effects of the propellers' air flow on the other quadrocopters are presented in chapter 7.

The drawback of the proposed approach is the time spent by the calculations for the trajectory planning, increasing with the complexity of the environment. In the experiments simulated in this thesis, the trajectory was optimized by the MPC method every 0.2 s, but the optimization itself took about 1 *minute*, depending on the testing scenario (on Intel Core-i7-3610QM 4x2.3 Ghz). To enable the on-board use of the trajectory planning, software optimization and parameters tuning, e.g. the number of elements in the trajectory, number of samples for the trajectory, has to be done to enable the real-time operation of the system.

To summarize the work done in this thesis, all of the given objectives and the main goal of this thesis were fulfilled. The contribution of this work is mainly in the novel approach combining the MPC-based trajectory planning with the flexible virtual structure defining the formation. As it can be seen from the experiments shown in the chapter 7, the proposed extension to the existing system for stabilization of formations of quadrotor UAVs, that enables complex maneuvers with sudden changes of velocity of the group and even the immediate changes of the velocity of the individual formation members in case of collision avoidance. Moreover, it is shown in the experiments in the chapter 7, that the proposed system is able to overcome disturbances caused by the air flow from the quadrocopters' propellers.

Appendix A

Content of enclosed DVD

The DVD attached to the printed version of this thesis contains the electronic version of this thesis in PDF format, the LaTeX source code of the thesis, the source code of the implemented plug-in to V-REP software with scenes and models used in V-REP for the thesis and video outputs from the experiments done in V-REP.

Directory/File	Description
MasterThesis	The LaTeX source code of this thesis.
Videos	Videos from the experiments exported from V-REP.
V-REP scenes	The scenes for V-REP used for the experiments.
V-REP models	The models used in simulations in V-REP.
jicha_MasterThesis.pdf	Electronic version of this thesis.

Table 1 Content on the attached DVD.

Bibliography

[1] M. Saska, Z. Kasl, and L. Preucil. "Motion Planning and Control of Formations of Micro Aerial Vehicles". English. In: *Proceedings of The 19th World Congress of the International Federation of Automatic Control*. Pretoria: IFAC, 2014, pp. 1228–1233. ISBN: 978-3-902823-62-5.

[2] M. Saska et al. "Control and Navigation in Manoeuvres of Formations of Unmanned Mobile Vehicles". English. In: *European Journal of Control* 19.2 (Mar. 2013), pp. 157–171. ISSN: 0947-3580. URL: http://www.sciencedirect.com/science/article/pii/S0947358013000204.

[3] M. Saska et al. "Ad-hoc Heterogeneous (MAV-UGV) Formations Stabilized Under a Top-View Relative Localization". English. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.

[4] M. Saska et al. "Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. Oct. 2012, pp. 2166–2171. DOI: 10.1109/IROS.2012.6385517.

[5] M. Saska, V. Vonasek, and L. Preucil. "Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs". In: *J. Intell. Robotics Syst.* 72.2 (Nov. 2013), pp. 239–261. ISSN: 0921-0296. DOI: 10.1007/s10846-013-9829-3. URL: http://dx.doi.org/10.1007/s10846-013-9829-3.

[6] D. M Stipanovic et al. "Cooperative avoidance control for multiagent systems". In: *Journal of Dynamic Systems, Measurement, and Control* 129.5 (2007), pp. 699–707.

[7] Y. Bouktir, M. Haddad, and T. Chettibi. "Trajectory planning for a quadrotor helicopter". In: *Control and Automation, 2008 16th Mediterranean Conference on*. June 2008, pp. 1258–1263. DOI: 10.1109/MED.2008.4602025.

[8] M.Anthony Lewis and Kar-Han Tan. "High Precision Formation Control of Mobile Robots Using Virtual Structures". English. In: *Autonomous Robots* 4.4 (1997), pp. 387–403. ISSN: 0929-5593. DOI: 10.1023/A:1008814708459. URL: http://dx.doi.org/10.1023/A:1008814708459.

[9] T. Balch and R.C. Arkin. "Behavior-based formation control for multirobot teams". In: *Robotics and Automation, IEEE Transactions on* 14.6 (Dec. 1998), pp. 926–939. ISSN: 1042-296X. DOI: 10.1109/70.736776.

[10] T. Paul, T.R. Krogstad, and J.T. Gravdahl. "UAV formation flight using 3D potential field". In: *Control and Automation, 2008 16th Mediterranean Conference on*. June 2008, pp. 1240–1245. DOI: 10.1109/MED.2008.4601984.

[11] T. Dierks and S. Jagannathan. "Neural network control of quadrotor UAV formations". In: *American Control Conference, 2009. ACC '09.* June 2009, pp. 2990–2996. DOI: 10.1109/ACC.2009.5160591.

[12] Manfred Morari and Jay H. Lee. "Model predictive control: past, present and future". In: *Computers & Chemical Engineering* 23.4–5 (1999), pp. 667–682. ISSN: 0098-1354. DOI: http://dx.doi.org/10.1016/S0098-1354(98)00301-9. URL: http://www.sciencedirect.com/science/article/pii/S0098135498003019.

[13] D.Q. Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814. ISSN: 0005-1098. DOI: http://dx.doi.org/10.1016/S0005-1098(99)00214-9. URL: http://www.sciencedirect.com/science/article/pii/S0005109899002149.

[14] Patrick Bouffard. "On-board Model Predictive Control of a Quadrotor Helicopter: Design, Implementation, and Experiments". MA thesis. EECS Department, University of California, Berkeley, Dec. 2012. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-241.html.

[15] Martin Saska. "Trajectory planning and optimal control for formations of autonomous robots". PhD thesis. 2011.

[16] Paul T. Boggs and Jon W. Tolle. "Sequential quadratic programming for large-scale nonlinear optimization". In: *Journal of Computational and Applied Mathematics* 124.1–2 (2000). Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations, pp. 123–137. ISSN: 0377-0427. DOI: http://dx.doi.org/10.1016/S0377-0427(00)00429-5. URL: http://www.sciencedirect.com/science/article/pii/S0377042700004295.

[17] J. Zhou, C. T. Lawrence, and A.L. Tits. *User's Guide for CFSQP Version 2.0: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints*. Tech. rep. 1994. URL: http://hdl.handle.net/1903/5496.

[18] J. Faigl et al. "Low-cost embedded system for relative localization in robotic swarms". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. May 2013, pp. 993–998. DOI: 10.1109/ICRA.2013.6630694.

[19] Andreas Masselli et al. "A Cross-Platform Comparison of Visual Marker Based Approaches for Autonomous Flight of Quadrocopters". English. In: *Journal of Intelligent & Robotic Systems* 73.1-4 (2014), pp. 349–359. ISSN: 0921-0296. DOI: 10.1007/s10846-013-9925-4. URL: http://dx.doi.org/10.1007/s10846-013-9925-4.

[20] Z. Kasl, M. Saska, and L. Preucil. "Rapidly Exploring Random Trees-based initialization of MPC technique designed for formations of MAVs". In: *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*. Vol. 02. Sept. 2014, pp. 436–443.

[21] F. Rak. "Hierarchical Approach to Motion Planning of Formations of Unmanned Helicopters in Complex Environment". Czech Technical University in Prague, Faculty of Electrical Engineering, 2014.

[22] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. ISBN: 9781139455176. URL: http://books.google.cz/books?id=Clg8SWNMSRAC.

[23] Steven M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep. Iowa State University, 1998.

[24] R. Omar. "Path planning for unmanned aerial vehicles using visibility line-based methods". PhD thesis. University of Leicester, 2011.

[25] K. Jiang, L.D. Seneviratne, and S.W.E. Earles. "Finding the 3D shortest path with visibility graph and minimum potential energy". In: *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*. Vol. 1. July 1993, 679–684 vol.1. DOI: 10.1109/IROS.1993.583190.

[26] Taeyoung Lee, M. Leoky, and N.H. McClamroch. "Geometric tracking control of a quadrotor UAV on SE(3)". In: *Decision and Control (CDC), 2010 49th IEEE Conference on*. Dec. 2010, pp. 5420–5425. DOI: 10.1109/CDC.2010.5717652.

[27] E. Rohmer, S.P.N. Singh, and M. Freese. "V-REP: A versatile and scalable robot simulation framework". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. Nov. 2013, pp. 1321–1326. DOI: 10.1109/IROS.2013.6696520.