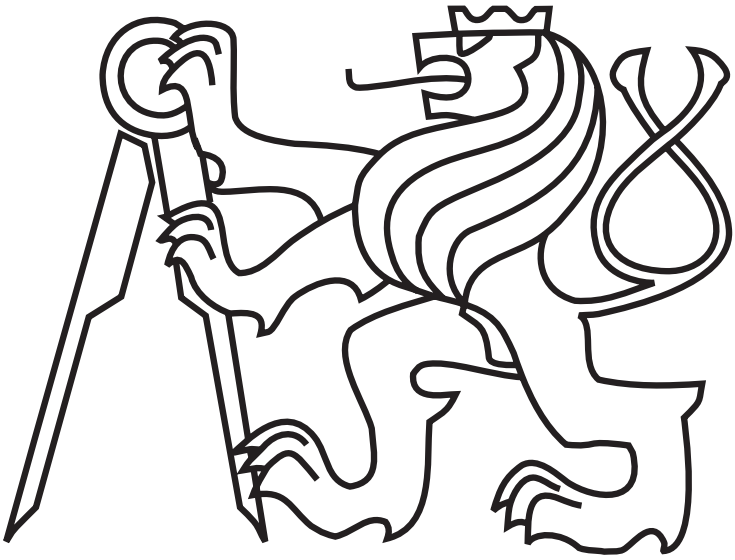
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

Bachelor's thesis



Daniel Heřt

Relatively Stabilized Unmanned Helicopters Led by Dynamic Virtual Leader

Department of Cybernetics

Thesis supervisor: **Dr. Martin Saska**

Prohlášení autora práce

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.

V Praze dne..... .....

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Daniel H e ř t

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: Relatively Stabilized Unmanned Helicopters Led by Dynamic Virtual Leader

Guidelines:

The aim of the thesis is to design and implement an algorithm for motion planning of formations of micro aerial vehicles (MAVs) based on a dynamic virtual leader with position which is optimized together with resulting trajectory. This mechanism enables to change motion direction of robots suddenly in 3D.
The designed method will be integrated into the system being developed by the Multi-robot Systems group for relative stabilization of MAVs.
The extended system will be verified in numerical simulations. The second half of the workload will be dedicated to HW experiment with quad-rotor helicopters or to statistical analysis of algorithm performance and reliability in simulator.
The decision will be taken by thesis advisor based on availability of robotic platform.

Bibliography/Sources:

[1] Saska, M. - Vonásek, V. - Přeučil, L.: Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs, Journal of Intelligent & Robotic Systems, 72(2): 239-261, November 2013.
[2] Saska, M. - Kasl, Z. - Přeučil, L.: Motion Planning and Control of Formations of Micro Aerial Vehicles. In Proceedings of The 19th World Congress of the International Federation of Automatic Control. IFAC, 2014.
[3] Barfoot, T.D. – Clark, C.M.:Motion planning for formations of mobile robots. Robotics and Autonomous Systems, 46(1):65-78, 2004.
[4] Das, A. – Fierro, R. – Kumar, V. – Ostrowski, J. – Spletzer, J. – Taylor, C. A vision-based formation control framework. IEEE Transactions on Robotics and Automation 18(5):813-825, 2003.

Bachelor Project Supervisor: Ing. Martin Saska, Dr. rer. nat.

Valid until: the end of the summer semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 23, 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Daniel H e ř t

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Vzájemně stabilizované autonomní helikoptéry vedené dynamickým virtuálním leaderem

Pokyny pro vypracování:

Cílem práce je navrhnout a implementovat algoritmus pro plánování pohybu formace bezpilotních helikoptér založený na dynamickém virtuálním leaderovi, jehož pozice je optimalizována spolu s výslednou trajektorií. Tento mechanismus umožní formaci náhle měnit směr pohybu ve 3D. Navržená metoda bude následně integrována do systému vyvíjeného pro vzájemnou stabilizaci helikoptér skupinou Multi-robotických systémů.
Student takto rozšířený systém verifikuje pomocí numerických simulací a v případě, že bude včas k dispozici robotická platforma i v reálném experimentu s kvadrokoptérami.
Na základě rozhodnutí vedoucího práce se student ve druhé polovině řešení zaměří na realizaci zmíněného reálného experimentu nebo na statistickou analýzu funkce algoritmu a jeho spolehlivosti v simulátoru.

Seznam odborné literatury:

[1] Saska, M. - Vonásek, V. - Přeučil, L.: Trajectory Planning and Control for Airport Snow Sweeping by Autonomous Formations of Ploughs, Journal of Intelligent & Robotic Systems, 72(2): 239-261, November 2013.
[2] Saska, M. - Kasl, Z. - Přeučil, L.: Motion Planning and Control of Formations of Micro Aerial Vehicles. In Proceedings of The 19th World Congress of the International Federation of Automatic Control. IFAC, 2014.
[3] Barfoot, T.D. – Clark, C.M.:Motion planning for formations of mobile robots. Robotics and Autonomous Systems, 46(1):65-78, 2004.
[4] Das, A. – Fierro, R. – Kumar, V. – Ostrowski, J. – Spletzer, J. – Taylor, C. A vision-based formation control framework. IEEE Transactions on Robotics and Automation 18(5):813-825, 2003.

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

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 23. 1. 2015

Acknowledgement

I would like to thank to the advisor of this thesis, Dr. Martin Saska, who dedicated a lot of time and patience to help me with this thesis. My thanks also belong to my family for their support.

Abstract

The aim of this thesis is to design and implement an algorithm for motion planning of formations of micro aerial vehicles based on a dynamic virtual leader, which migrates along a convex hull surrounding the formation. Position of the virtual leader driving the formation motion is optimized together with trajectory, which is feasible for the formation. This mechanism enables to change motion direction of robots suddenly in 3D. The designed system is verified in numerical simulation and by a statistical analysis.

Abstrakt

Cílem této práce je navrhnout a implementovat algoritmus pro plánování pohybu formace bezpilotních helikoptér založený na dynamickém virtuálním leaderovi, který migruje po konvexním tělese které obaluje formaci. Pozice virtuálního leadera, který vede formaci, je optimalizována spolu s výslednou trajektorií. Tento mechanismus umožní formaci náhle měnit směr pohybu ve 3D. Navržený systém je ověřen numerickými simulacemi a statistickou analýzou.

Table of contents

List of figures

List of tables

1 Introduction

Micro aerial vehicles (MAV) have been recently used in many applications. The favorite type of MAV is quad-rotor unmanned aerial vehicle (UAV), also called quadrotor. Quadrotor uses four counter-rotating rotors with fixed pitch. This configuration is much simpler than a classical helicopter, which uses variable pitch blades and a tail rotor. Quadrotors are controlled only by changing the speed of the rotation of individual rotors. They are also capable of vertical take-off and landing. Advances in technology allow producing various components like motors, batteries and control units that are relatively cheap and lightweight. This allows building small vehicles, which are man-portable, affordable, capable of operating in confined areas and have enough computing power and battery endurance.

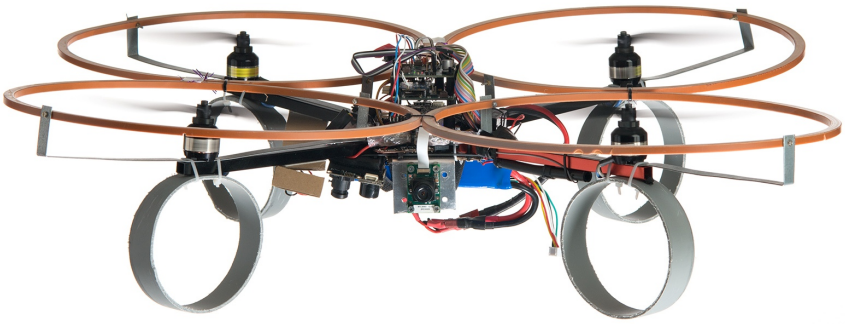


Figure 1: MikroKopter L4-ME quadrotor UAV

The goal of this thesis is to extend the 3D leader-follower formation control method presented in [8] to allow for flight in constrained areas with a lot of obstacles. This is done by introducing a migrating virtual leader, which migrates around the formation. This allows to execute rotation maneuvers and thus improves maneuverability of the formation.

This thesis is organized as follows. In chapter 3, specification of the task is presented. In chapter 4 preliminaries are presented, including Model Predictive Control, Sequential quadratic programming, trajectory representation and trajectory planning for the leader and followers. In chapter 5 the novel approach of the virtual migrating leader is presented. In chapter 6 experimental verification and statistical analysis of the proposed approach is presented.

2 State of the art

In the literature, there are many approaches which are dealing with coordination of robot formations. The most common one is the leader-follower approach, which designates one robot as a leader and positions of other robots are defined relatively to the leader. This approach is used for example in [11].

In [3], leader-follower approach is not used, instead distributed control scheme is proposed to solve the problem utilizing the notion of graph rigidity and persistence, as well as techniques of virtual target tracking and smooth switching.

Behavioral techniques based on observing flocks and herds of animals are described in [1]. Another approach is presented in [9], which is planning the motion of robots as a rigid body that represents the formation.

Coordinated formations of multiple MAVs have a lot of potential applications. They can be equipped with a range of various sensors including camera, smoke detectors or radiation detectors and therefore they are suited for search and patrolling operations, for example patrolling of a guarded area, patrolling of an area suspected of fire hazard or search missions during natural disasters. UAV patrolling is presented in [2].

Interesting use of quadrotor formations is described in [13]. Formation of quadrotors is used to lift a large dimensional payload, which is too heavy to be lifted by a single quadrotor. Master-slave approach is then used to control the formation. This approach is an important development in pushing quadrotor technology for industrial applications.

This thesis extends methods presented in thesis [8], where a formation driving method for UAVs is presented. In comparison with other approaches, the approach presented in this thesis is designed for formation flight in clustered environments, where sharp changes in motion of the robots is required.

3 Specification

Goal of this thesis is to extend 3D leader-follower formation driving method presented in [8]. In this method, the formation consists of constant number of quadrotors, one of which is designated as a leader, others as followers. Physical state of the quadrotor is known at all times. It is represented by position vector, rotation matrix, velocity vector and angular velocity vector. Quadrotors are aware of the positions of other quadrotors and thus are able to prevent collisions.

3D environment is represented by a map, which consists of obstacles and the goal of the flight. Positions of all obstacles and all quadrotors in the map are known.

Shape of the formation is defined by the position of followers in curvilinear coordinates relatively to the leader. The followers are trying to keep the shape of the formation constant during the flight, but it can be temporarily changed to allow obstacle avoidance.

The goal of this thesis is to find an optimal trajectory for the formation. The trajectory is optimized to achieve several objectives with weights defined by the user. The objectives are:

Distance to obstacles: Prevents collisions with obstacles.

Angular velocity: Penalty for curvature of the trajectory, straight trajectories are preferred.

Trajectory length: Shorter trajectories are preferred.

Time of flight: Trajectories that reach the goal faster are preferred.

4 Preliminaries

In this chapter, introduction of methods used in this thesis is given. In section 4.1, Model Predictive Control method is described. In section 4.2, sequential quadratic programming is described. In section 4.3, the representation of trajectory is described.

4.1 Model Predictive Control

The Model predictive control (MPC) method, also called the Receding Horizon Control (RHC), is a widely used control technique. It uses a dynamic model of the system, which is usually obtained by system identification. Current system state and the dynamic model are used to predict the state of the system and to optimize it in a short time horizon. The control problem is coded into an optimization vector with N control elements. N is the length of the horizon. Each element has constant duration. More details about MPC can be found in [6].

For the purposes of trajectory planning for robot formations, an extended optimization vector has been used in [8]. The MPC planning horizon is extended by an additional M vector elements with variable duration Δt_M. This extension allows for the whole trajectory to be encoded into the optimization vector, instead of just the short time horizon. This allows a global trajectory optimization.

The process of MPC can be described as follows: in each loop, an optimal control problem is solved from the known robot's state. The first part of optimized control inputs is then used to move the robot to a new state. The loop then continues until the robot reaches the goal. This process is illustrated in figure 2.

4.2 Sequential quadratic programming

Sequential quadratic programming (SQP) is a very powerful iterative method for nonlinear constrained optimization. The method generates steps by solving individual quadratic sub problems. The SQP method is suitable for solving the MPC problem of trajectory planning. In trajectory planning, fast recalculation of the optimal trajectory is needed to allow the robot to avoid newly discovered obstacles. Thus an efficient implementation of SQP should be used. In [8] and in this thesis, Feasible Sequential Quadratic Programming (CFSQP) library designed by Craig T. Lawrence, Jian L. Zhou, and André L. Tits is used. More information can be found in the CFSQP manual [4].

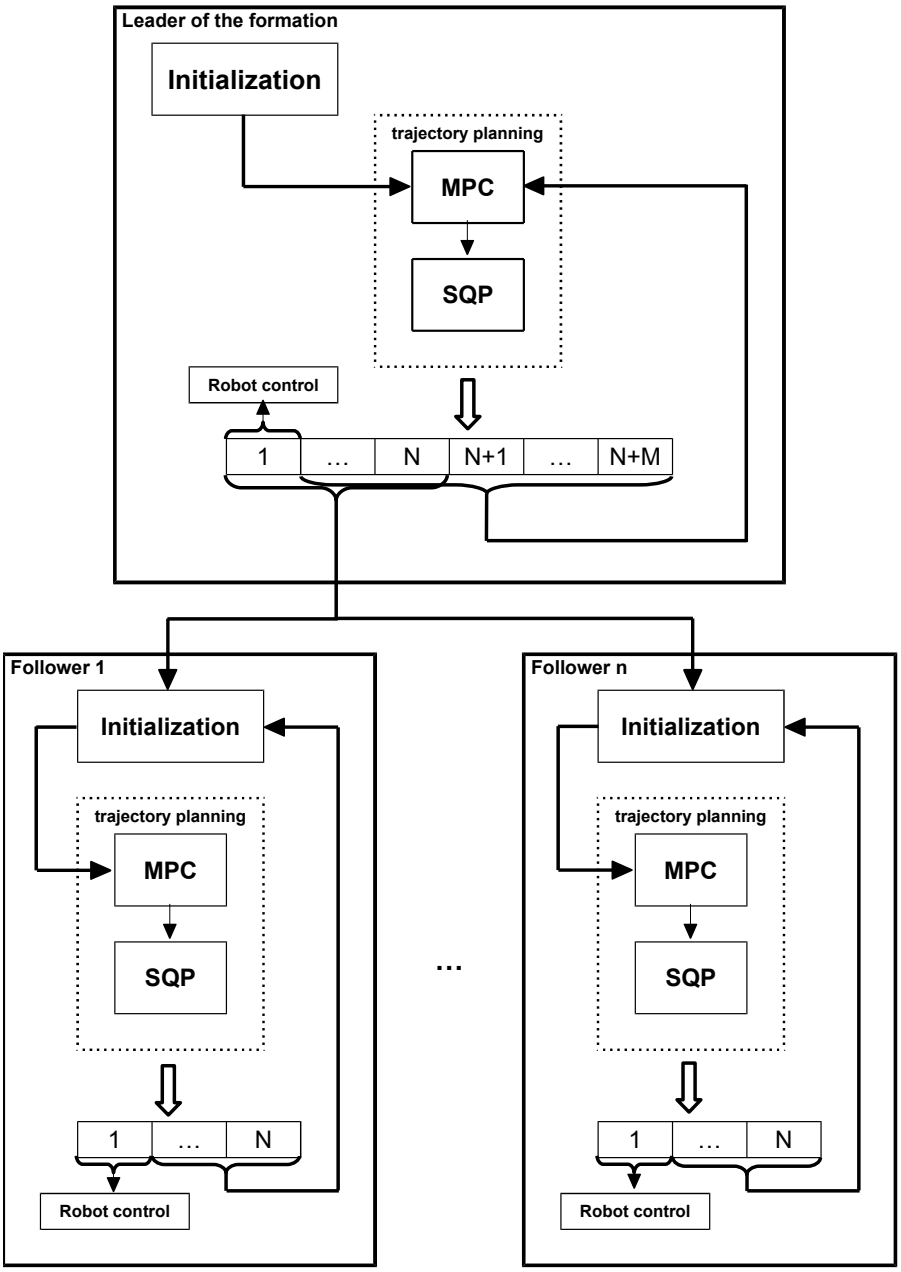


Figure 2: Extended Model Predictive Control method used in [8] and in this thesis.

4.3 Trajectory representation

The MPC approach described in 4.1 requires the trajectory to be coded into a vector of constant control inputs. In this section, the used vector representation is described. The vector used to represent the trajectory (described in figure 3) consists of four elements:

$v_t[m \cdot s^{-1}]$ is the tangential velocity,

$v_n[m \cdot s^{-1}]$ is the normal velocity,

$\omega[rad \cdot s^{-1}]$ is the angular velocity around an axis,

$t[s]$ is the time interval.

v_{t1}	v_{n1}	ω_1	t_1	v_{t2}	v_{n2}	ω_2	t_2	v_{t3}	v_{n3}	ω_3	t_3	...	v_{tn}	v_{nn}	ω_n	t_n

Figure 3: Representation of the quadrotor's trajectory as a vector of consecutive constant cotrol inputs. Each interval consists of three control inputs which are constant over a given time t_i.

4.4 Leader's trajectory planning

In this section, the approach designed for leader trajectory planning is described. Firstly, initial trajectory must be found. This trajectory is used as a basis of the optimization process. It needs to be feasible, but it does not have to be optimal. After this, the control loop is started. In the course of each iteration, vector representing the trajectory (shown in figure 3) is passed to the CFSQP solver. Optimization is done with objective and constraint functions. Objective function evaluates if the new iterate of trajectory is better than other. The evaluation considers multiple aspects of the trajectory:

Obstacle penalty: This component allows the quadrotor to avoid obstacles. It is equal to zero when the quadrotor is in a safe distance, and rises rapidly if it gets closer. To determine the obstacle penalty, the trajectory is sampled. The result of sampling a set of points in 3D space that covers the trajectory. Penalty is then calculated for each point. Maximal value is then taken as the resulting obstacle penalty.

Angular velocity: Penalty for curvature of the trajectory calculated as a sum of lengths of all angular velocities of all trajectory segments. Straighter trajectories are preferred, because they require less maneuvering by the robots.

Trajectory length: This component forces the solver to prefer shorter trajectories. It is equal to the sum of lengths of all segments of trajectory.

Time of flight: This component forces the solver to prefer trajectories with shorter time of flight. It is obtained as the sum of durations of all elements of the trajectory.

Distance to goal: The last component is not a classical objective, it is a doubled constraint that helps to stabilize and speed up the CFSQP optimization process.

Constraint function adds motion constraints into the process of optimization. In the CFSQP solver, if the value of the constraint is greater than zero, trajectory is considered as not feasible. Two constraints are used:

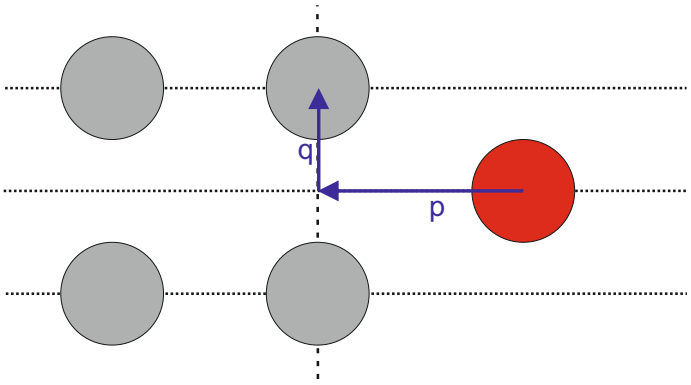
Obstacle distance: This constraint prevents the trajectory from intersecting with obstacles. It is calculated in a similar way as Obstacle penalty in the objective function. Obstacles have small defined safety radiuses around them. If the trajectory intersects with the safety radius or with the obstacle at any point, the trajectory is considered not feasible.

Distance to goal: This constraint ensures that leader reaches the specified goal. The trajectory is sampled, and the distance of the last point of the trajectory to the specified goal is calculated.

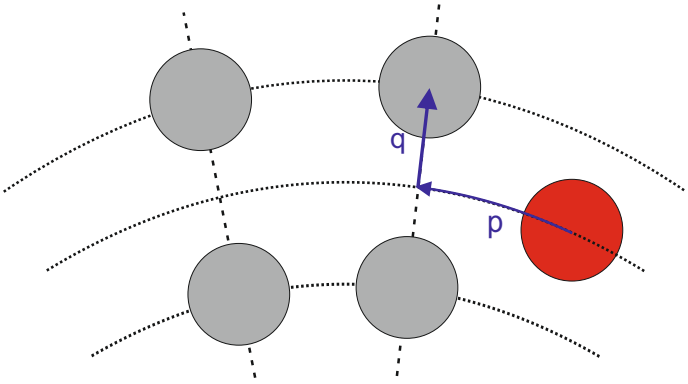
After the optimization process is finished, the first part of the resulting control inputs is used to move the leader to a new position. The loop then continues, until the formation reaches its goal.

4.5 Follower's trajectory planning

Follower's position is defined in curvilinear coordinates (p_i, q_i and r_i) relatively to the position of the leader. Coordinates p_i and q_i determine the position in a 2D plane, r_i then defines the z-coordinate. Meaning of coordinates is described in figures 4 and 5.



(a) Straight motion



(b) Curved motion

Figure 4: Positions of followers, keeping the same position in the formation defined relatively to the leader (red) by the p and q parameters in a straight and curved motion.

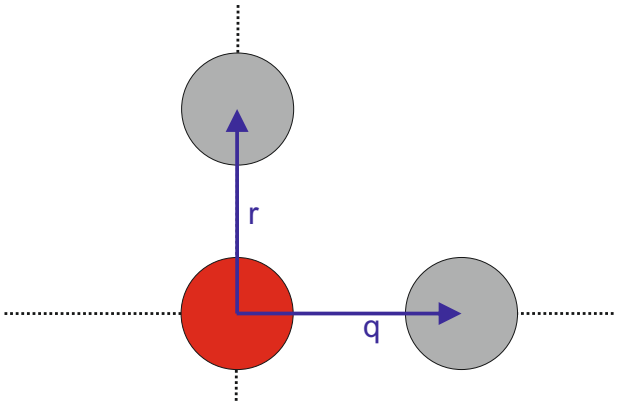


Figure 5: Positions of followers defined relatively to the leader (red) by r and q parameters.

In the presented approach, each follower tracks the leader and avoids obstacles and other followers. Only a small portion of the trajectory to the goal needs to be optimized, because followers do not need the global awareness about the environment for the purpose of trajectory planning. This is included in the leader's planning. As the initial trajectory for the followers, a small part of the leader's trajectory is used and transformed to correspond with the follower's position. Trajectory is then optimized similarly as the leader's trajectory. Objective function evaluates different aspects of the trajectory:

Obstacle penalty: This component works the same way as the leader's obstacle penalty, but takes into account not only obstacles, but also other followers.

Difference from the desired position in the formation: This component evaluates the distance of the follower from its desired position in the formation.

Angular velocity: This component is the same as the one in leader's objective function.

Control difference: This component helps to stabilize the leader's trajectory tracking problem. The value is calculated as a sum of differences between the follower's optimization vector elements and the leader's control. Parameters p_i and q_i have to be taken into account.

The constraint function for the followers works the same way as the leader's constraint function. Two constraints are set.

Obstacle distance: This constraint is the same as in the leader's constraint function.

Difference from the desired position in the formation: This component limits the motion of the follower to an area around its desired position.

For more details on the objective functions and constraints see [8], where their comprehensive description required for implementation is provided.

4.6 Obstacles

Obstacles are represented as spheres. This allows for an efficient calculation of the obstacle distance from the quadrotor. In any point p, distance r can be calculated as $r = |pc_0| - r_0$ where r_0 is the radius of the obstacle and $|pc_0|$ is the Euclidean distance between the point p and the center of the sphere c_0.

5 Trajectory planning with complex maneuvers

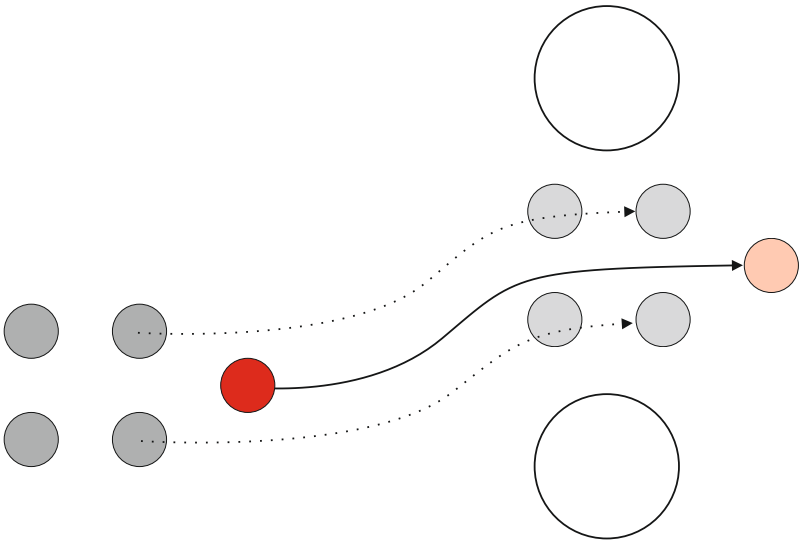
The implementation presented in [8] is limited by only two modes of motion of the formation. Quadrotors can fly either on straight or on curved trajectory, but they cannot perform sharp turns or complete turnabouts. This thesis purpose is to extend this implementation to allow above mentioned maneuvers. The work presented in this chapter is novel in sense of design, implementation and experimental verification.

5.1 Virtual migrating leader

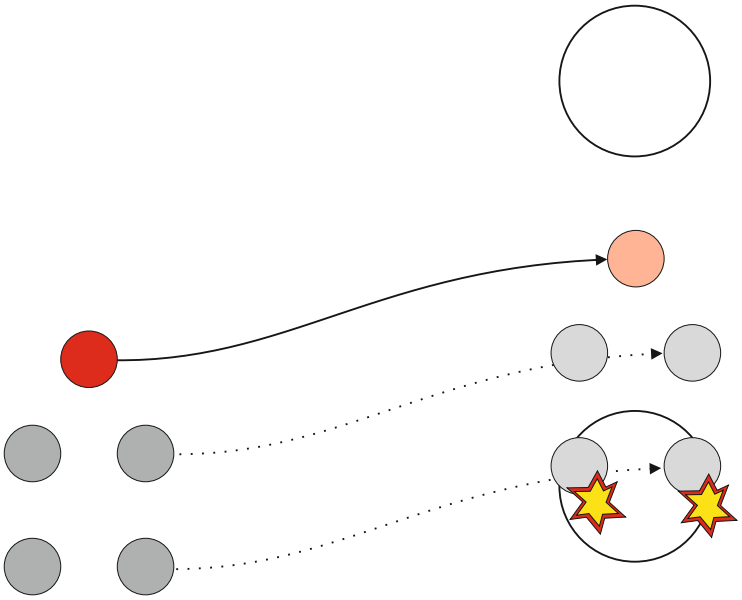
To extend the maneuverability of the formation, conventional leader used in [8] is replaced with a virtual migrating leader. Virtual leader has the ability to initialize or reinitialize itself anywhere on a circle of diameter d with its center in the center of the formation. This allows the leader to shift itself during the flight to a different position on the circle and to lead the formation to another direction.

It is important, that the leader is flying in front of the formation. If the leader would not be in front of the formation, followers would have to predict the leader's movement, which would bring impractical complexity and inaccuracy to the system. Leader should also fly on the axis of movement of the formation. This allows for effective non-collision trajectory planning of the formation, as illustrated in figure 6.

The migration of the virtual leader is illustrated in figure 7. This maneuver will be called the rotation maneuver from now on.

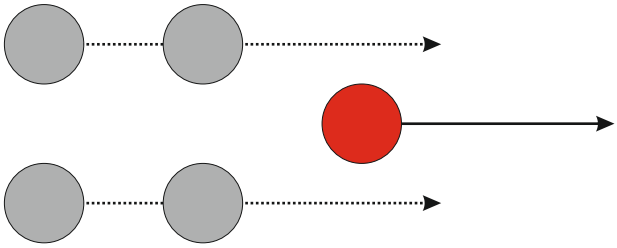


(a) Leader is on the axis of movement of the formation, which allows for an effective collision avoidance, using the presented approach.

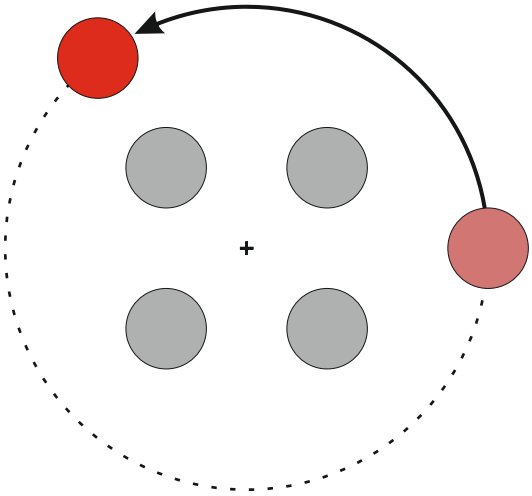


(b) Leader, which is offset of the axis of movement of the formation cannot effectively avoid obstacles, using the presented approach.

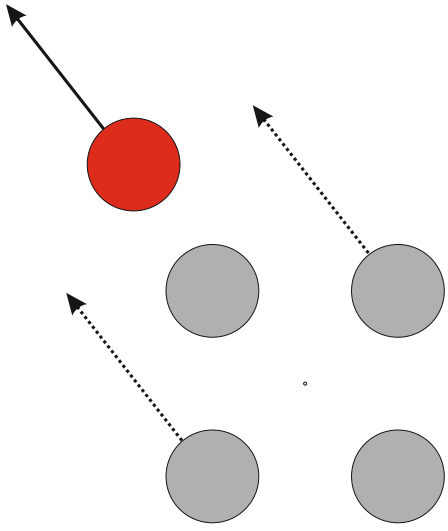
Figure 6: Obstacle avoidance scenario. Leader (red), followers (grey) and obstacles (transparent).



(a) Formation during straight flight.



(b) Migration of the virtual leader to enable sudden change of flight direction, if maneuvering in clustered environment is necessary.



(c) Flight continues in a different direction.

Figure 7: Virtual leader migration. Leader (red), followers (grey).

5.2 Rotation maneuver

To represent the rotation maneuver, vector of constant control inputs of the leader, presented in figure 3, is extended by optimization variable α. The vector is now composed of five control inputs:

$\alpha[rad]$ is the change of angle during the rotation maneuver,
$v_t[m \cdot s^{-1}]$ is the tangential velocity,
$v_n[m \cdot s^{-1}]$ is the normal velocity,
$\omega[rad \cdot s^{-1}]$ is the angular velocity around an axis,
$t[s]$ is the time interval.

α_1	v_{t1}	v_{n1}	ω_1	t_1	α_2	v_{t2}	v_{n2}	ω_2	t_2	...	α_n	v_{tn}	v_{nn}	ω_n	t_n

Figure 8: Extended vector of constant control inputs

The new control input α represents the angle of the rotation and allows the solver to calculate with it. The limits of α are $< -\pi, \pi >$. If α is equal to zero, no rotation is executed and the formation continues in normal flight. If α is not equal to zero, rotation maneuver is carried out. The maneuver itself consists of several steps:

1) Bringing the formation to a hover
2) Virtual leader migration
3) Recalculation of the p_i and q_i parameters
4) Continuation of the flight.

5.2.1 Bringing the formation to a hover

If the rotation maneuver was executed with the formation being in motion, unwanted residual lateral velocity could appear after the maneuver. To eliminate the possibility of the residual velocity, formation needs to be brought to a complete hovering mode. If a set of control inputs containing a non-zero α input needs to be used for moving the leader, formation needs to be brought to a hover, before the set of control inputs is used. Firstly, another set of control inputs is used, which forces all the followers and the leader to a hover as soon as possible. After that, the original set with a non-zero α input is used to move the leader and execute the migration.

5.2.2 Virtual leader migration

Once the formation is hovering, the virtual leader can migrate to a new position on the circle, defined by α. Firstly, center of the leader's migration circle has to be found. This is done by finding the two followers with the largest mutual distance. Center of the leader's migration circle is then set to be the midpoint between these two followers and the radius of the circle is calculated as follows:

$$r = \frac{d_{max}}{2} + r_s,$$

where d_{max} is the largest mutual distance between two followers and r_s is an additional safety radius which is applied to ensure that the leader is always in front of the formation. The leader's migration circle is shown in figure 9.

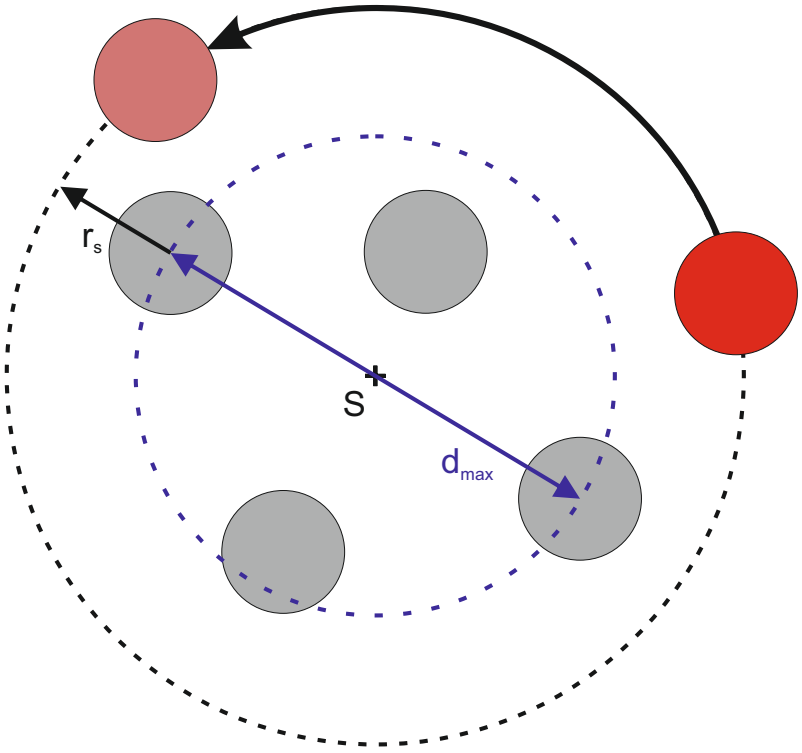


Figure 9: Leader's migration circle.

After the circle center is calculated, the leader migrates to its new position. The new leader's position is calculated as follows:

$$X = F_x + r \cdot cos(\alpha + \alpha_c),$$
$$Y = F_y + r \cdot sin(\alpha + \alpha_c),$$

where α is the angle of the rotation maneuver, α_c is be the current leader's yaw angle and F_x and F_y is the current calculated circle center.

Quadrotors are symmetrical machines, therefore they have the ability to fly in any direction. The system used in this thesis requires the quadrotors to have defined heading for the purposes of visualization. Thus the heading itself must be changed during the rotation maneuver, for the purposes of visualization. Let $\mathbf{R}$ be the current quadrotors rotation matrix. During the rotation maneuver, it is recalculated for all the quadrotors:

$$\mathbf{R} = \mathbf{R} \cdot \mathbf{R}_\alpha,$$

where $\mathbf{R}_\alpha$ is a matrix that represents the rotation:

$$\mathbf{R}_\alpha = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

5.2.3 Recalculation of the p_i and q_i parameters

After the leader migrates to its new position, the formation should keep its original shape and continue to fly in a new direction. But the original p_i and q_i parameters does not represent the formation after the leader's migration. Thus they must be recalculated.

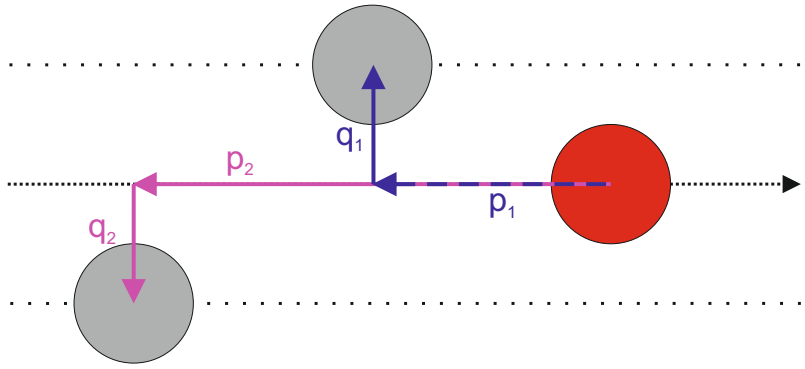
The actual position of the follower may be slightly different than the desired position. To keep the original shape of the formation, the desired position of the follower is used. It is calculated as follows:

$$F_{xdi} = F_{xl} - p_i \cdot cos(\alpha_c) - q_i \cdot sin(\alpha_c),$$
$$F_{ydi} = F_{yl} + q_i \cdot cos(\alpha_c) - p_i \cdot sin(\alpha_c),$$

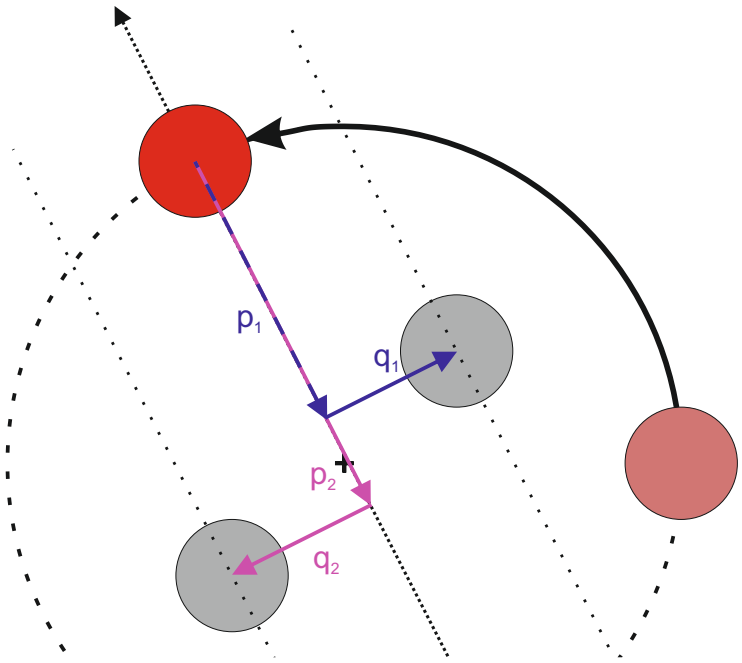
where F_{xl} and F_{yl} is the position of the leader before the rotation manouver is executed and α_c is the leader's current yaw angle. The new p_i and q_i parameters are then calculated as follows:

$$p_i = (F_x - F_{xdi}) \cdot cos(\alpha_c + \alpha) + (F_y - F_{ydi}\cdot) sin(\alpha_c + \alpha),$$
$$q_i = (F_{ydi} - F_y) \cdot cos(\alpha_c + \alpha) + (F_x - F_{xdi}\cdot) sin(\alpha_c + \alpha),$$

where F_x and F_y is the position of the leader after the rotation maneuver is executed and α is the angle of the rotation maneuver. Recalculation of the p_i and q_i parameters is shown in figure 10.



(a) Formation before the leader migration.



(b) Formation after the leader migration.

Figure 10: Recalculation of p_i and q_i parameters. Leader (red), follower's desired positions (grey).

5.2.4 Continuation of the flight

After the rotation maneuver is carried out, the formation can continue in flight. Followers aren't located usually exactly at the positions specified by the new p_i and q_i parameters, therefore they need to make corrections. Those corrections are realized with the MPC scheme in a feedback fashion.

5.3 Objective function

Objective function, as described in 4.4, evaluates quality of the trajectory regarding the specified objectives. The new rotation maneuver needs to be considered when calculating the objective function. If for example the rotation maneuver is not needed and the formation can easily reach the goal without it, solver should optimize the trajectory and remove the unnecessary rotation maneuver. The maneuver is difficult and time consuming, because the formation needs to completely stop. Therefore, it should only be used if a feasible trajectory without the maneuver does not exist, or if it is much longer and more complex.

If the rotation maneuver is used, penalty is added in form of time of flight penalty, which is described in section 4.4. This penalty represents the time needed to bring the formation to a hold. It depends on the speed of the robots and on their maximum acceleration. However this penalty does not depend on the value of α (which represents the angle of the rotation maneuver) changes. This means that it is difficult for the optimization process to get rid of unnecessary rotation maneuvers, due to missing gradient leading in elimination of the maneuver. To allow better and faster optimization, another objective is added into the leader's objective function:

Rotation maneuver penalty: The penalty is visualised in figure 11. It is calculated as follows[1]:

$$\text{if} \quad \alpha \in (-\alpha_{min}, \alpha_{min}) \qquad \text{then} \qquad pen = ln(|b \cdot \alpha| + 1)$$

$$\text{if} \quad \alpha \in < -\pi, -\alpha_{min} >, < \alpha_{min}, \pi > \qquad \text{then} \qquad pen = \frac{|\alpha|}{c} + ln(|b \cdot \alpha_{min}| + 1)$$

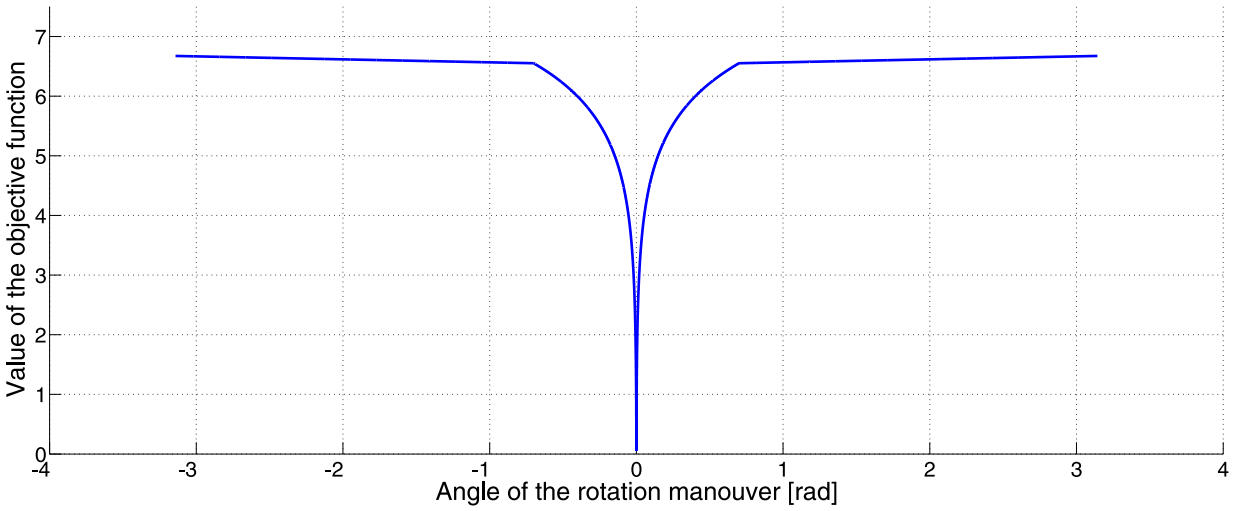


Figure 11: Leader's rotation maneuver penalty for $b = 1000$, $c = 20$ and $\alpha_{min} = 0.7$.

[1]Values $b = 1000$, $c = 20$ and $\alpha_{min} = 0.7$ are used in all of the experiments in this thesis.

For any value of α that is greater than α_{min}, the penalty is almost the same. However there is still a slight slope in the function, to help the solver to optimize towards removing the unnecessary maneuver. If $\alpha < \alpha_{min}$ the rotation maneuver can be replaced by trajectory curvature, most of the time, depending on the objective function. Therefore value of the penalty starts to drop significantly, to force the solver to use the value $\alpha = 0$. The weight of this objective can be modified by the user, similarly as the weight of the other objectives, to allow for different modes of optimization.

5.4 Obstacles

In [8], obstacles are implemented only as spheres of given center and diameter. This thesis extends the system with a maneuver, which is mainly used indoors or in confined areas. Spherical obstacles aren't suitable to represent obstacles like walls or corridors. Therefore the implementation of obstacles was changed.

5.4.1 Gilbert–Johnson–Keerthi distance algorithm

Gilbert–Johnson–Keerthi distance algorithm (GJK) is a method that determines the minimal distance between two convex sets. GJK is an iterative method that converges very fast, thus it is widely used in real time collision detection. For more information about GJK, see [7]. For the purposes of this thesis, implementation by Stephen Cameron is used [5]. Obstacles are defined as 8 points in 3D space, which forms a complex polyhedron.

6 Experimental verification

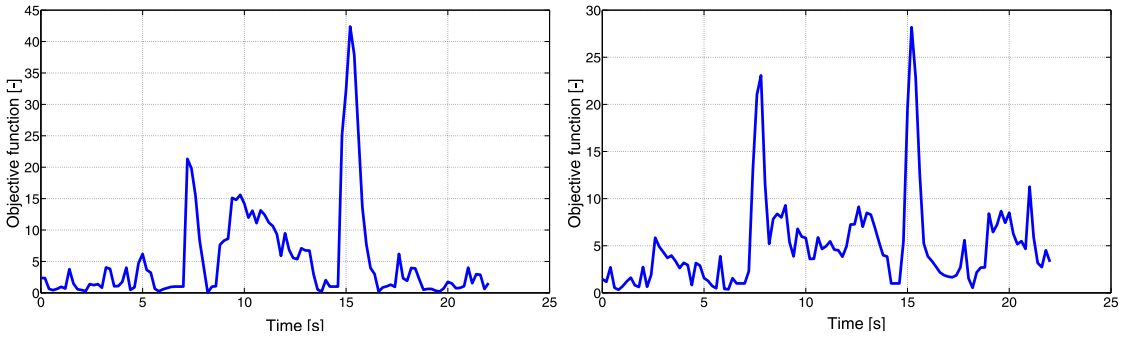
6.1 Rotation maneuver

In the first scenario, rotation maneuver is tested. The formation consists of a virtual leader (blue) and four followers (orange). Formation parameters are in table 1.

follower	1	2	3	4
p_i	0.3	0.3	0.8	0.8
q_i	0.3	-0.3	0.3	-0.3
r_i	0	0	0	0

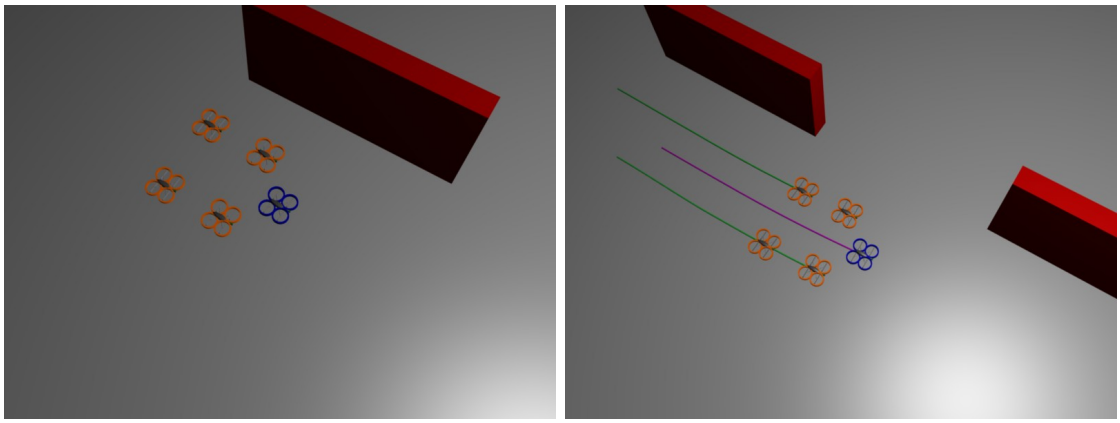
Table 1: Formation parameters

Obstacles are rendered red. The results of the simulation are visualized in figures 12, 13 and 14.

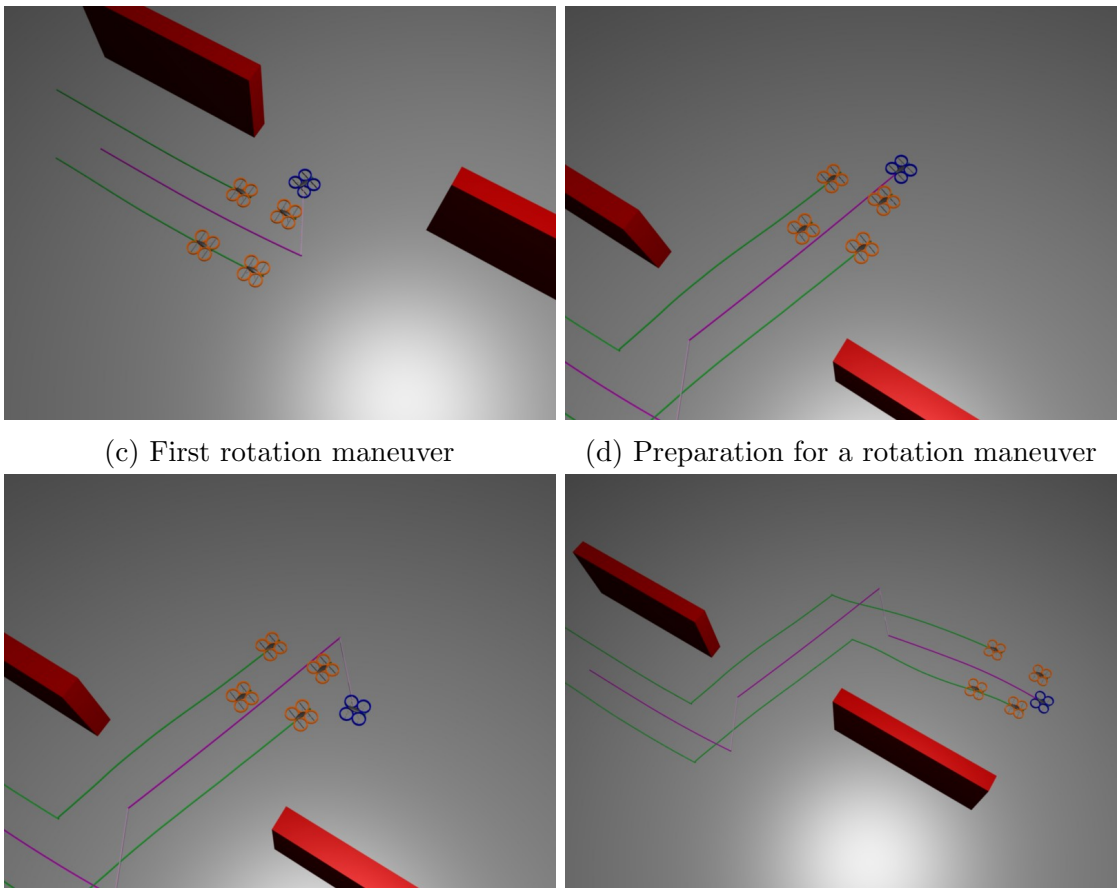


(a) Follower 2 (b) Follower 3

Figure 12: Objective function of follower 2 and 3 in time. Two peaks visible in both graphs are caused by the rotation maneuvers.



(a) Starting position (b) Preparation for a rotation maneuver



(c) First rotation maneuver

(d) Preparation for a rotation maneuver

(e) Second rotation maneuver

(f) Formation continues in flight

Figure 13: Testing of the rotation maneuver.

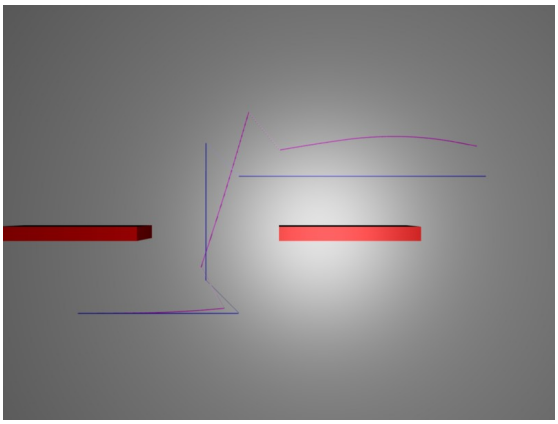


Figure 14: Comparison of the initial trajectory (blue) and actual leader's optimized trajectory (purple).

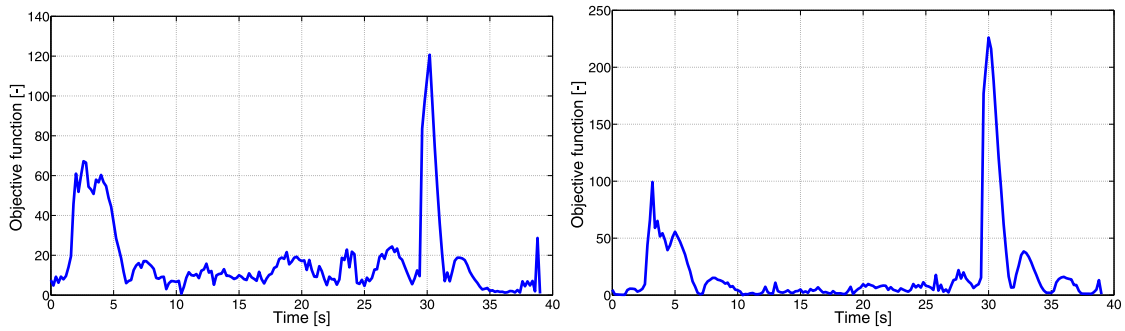
6.2 Obstacle avoidance

In this scenario, the ability of the system to avoid obstacles is tested. The space between the first set of obstacles is smaller than the formation itself, so the formation have to temporarily change its shape. Another obstacle is discovered during the flight, and formation must react to avoid it. The formation consists of a virtual leader (blue) and four followers (orange). Formation parameters are in table 2.

follower	1	2	3	4
p_i	0.6	0.6	1.0	1.0
q_i	0.4	-0.4	0.4	-0.4
r_i	0	0	0	0

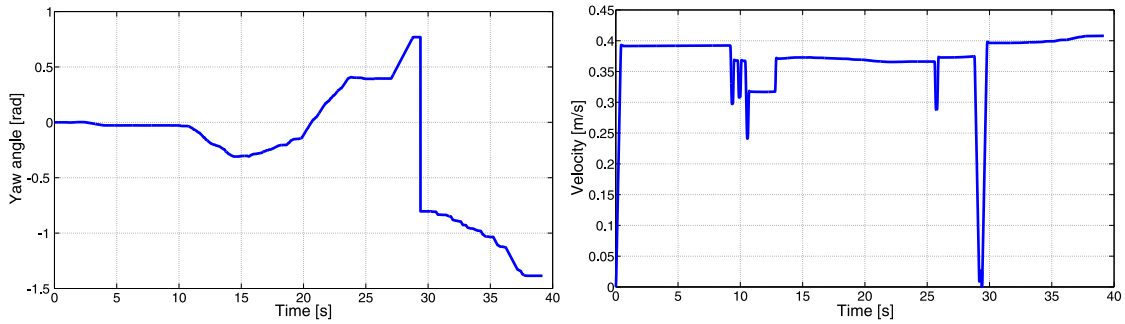
Table 2: Formation parameters

Obstacles are rendered red. The results of the simulation are visualized in figures 15, 16 and 17.



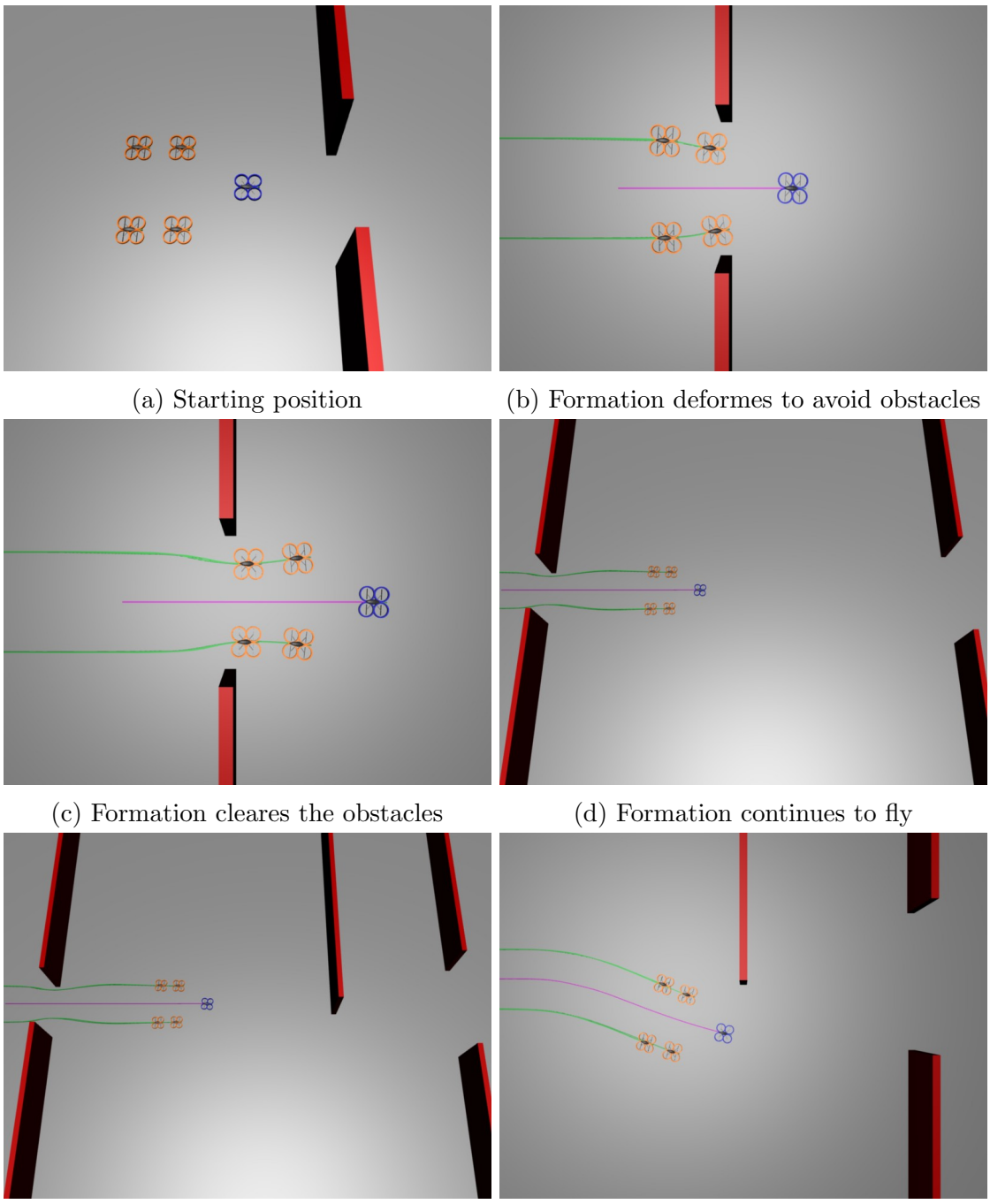
(a) Follower 1 (b) Follower 3

Figure 15: Objective function of followers in time. The first peak is due to the avoidance of the obstacles, while the second peak represents the rotation maneuver.



(a) Leader's yaw angle (b) Leader's velocity

Figure 16: Leader's yaw angle and velocity.



(a) Starting position

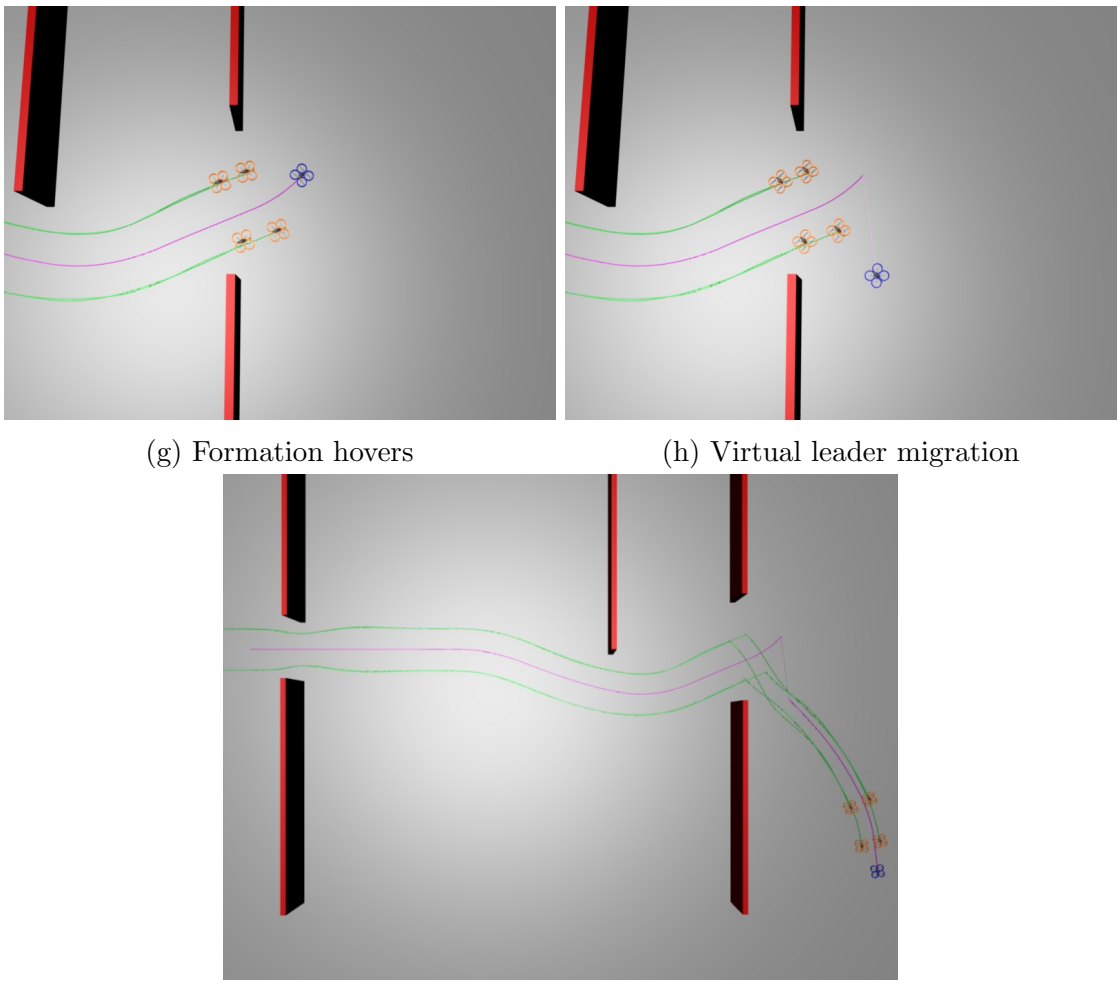
(b) Formation deformes to avoid obstacles

(c) Formation cleares the obstacles

(d) Formation continues to fly

(e) New obstacle is discovered

(f) Formation avoids the new obstacle



(g) Formation hovers

(h) Virtual leader migration

(i) Formation reaches the goal

Figure 17: Deformation of the formation and discovery of a new obstacle, combined with the rotation maneuver.

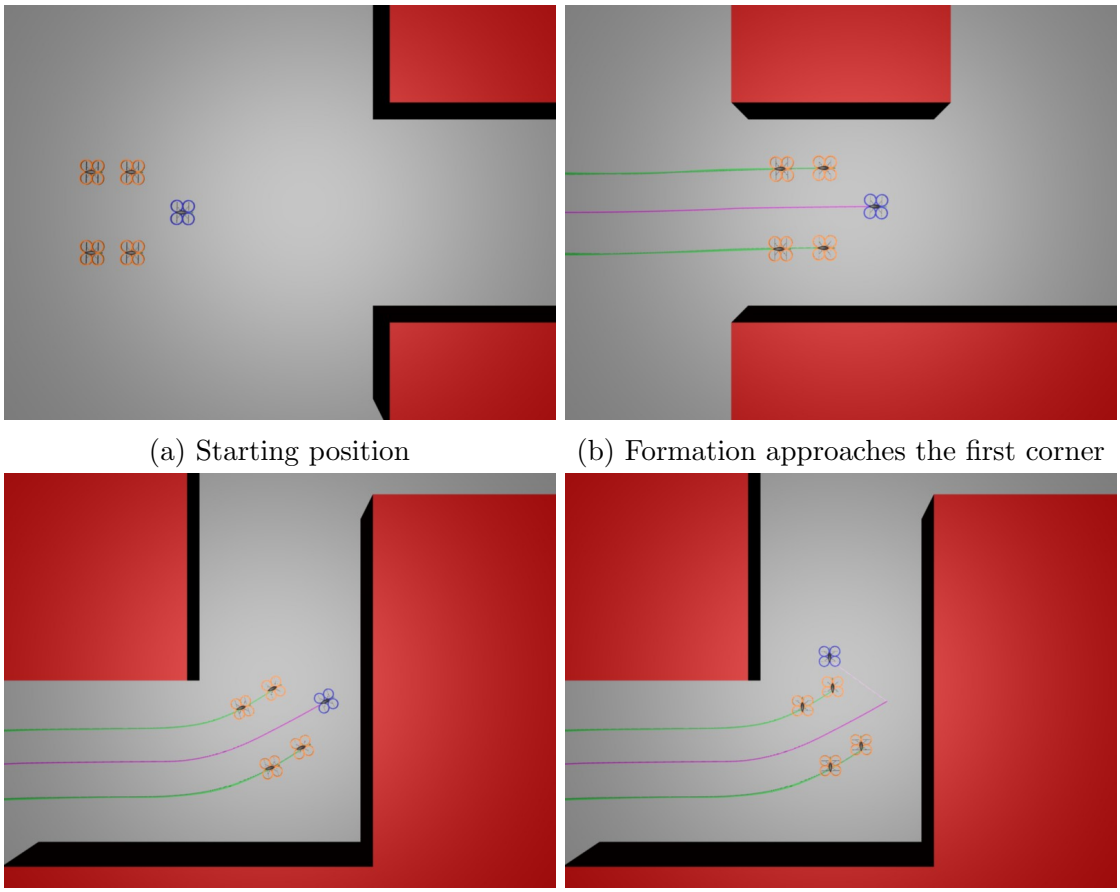
6.3 Corridor flight

Quadrotors are often used in buildings or corridors. Maneuvering in this environment often requires sharp turns to be made by the formation. This simulation represents an indoor scenario with three 90° turns. The formation consists of a virtual leader (blue) and four followers (orange). Formation parameters are in table 3.

follower	1	2	3	4
p_i	0.5	0.5	0.9	0.9
q_i	0.4	-0.4	0.4	-0.4
r_i	0	0	0	0

Table 3: Formation parameters

Obstacles are rendered red. The results of the simulation are visualized in figure 18.

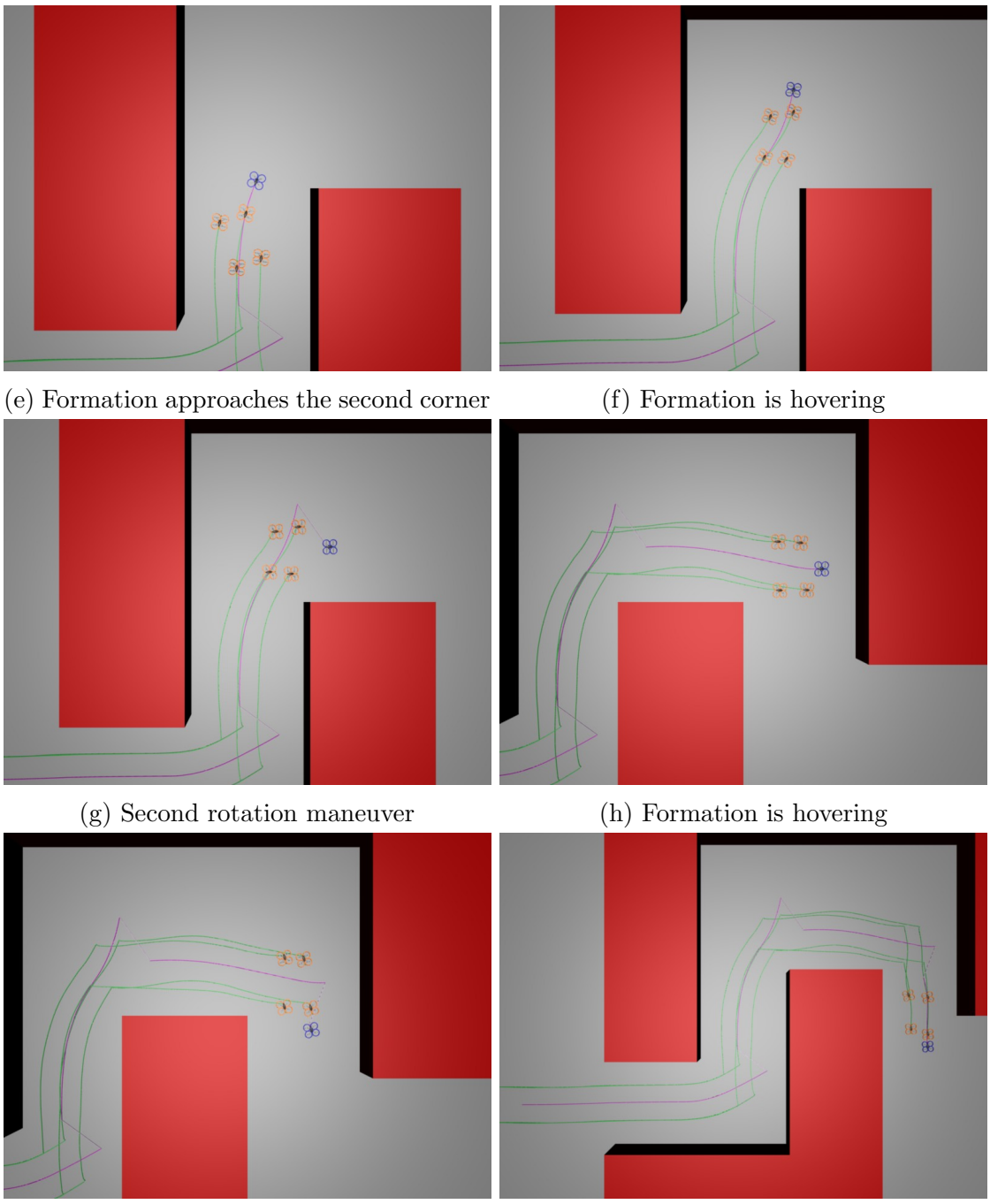


(a) Starting position

(b) Formation approaches the first corner

(c) Formation is hovering

(d) First rotation maneuver



(e) Formation approaches the second corner

(f) Formation is hovering

(g) Second rotation maneuver

(h) Formation is hovering

(i) Third rotation maneuver

(j) Formation reaches the goal

Figure 18: Formation flying in a corridor.

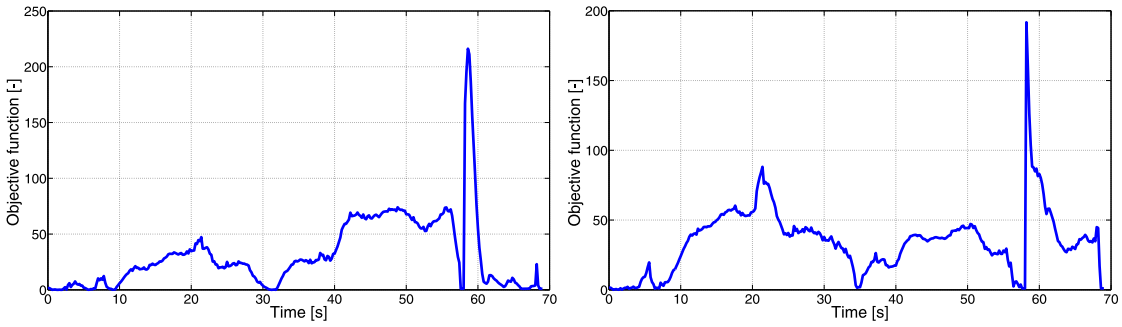
6.4 Slalom flight

This scenario demonstrates the ability of the system to fly on a highly curved trajectory combined with the usage of the rotation maneuver The formation consists of a virtual leader (blue) and four followers (orange). Formation parameters are in table 4.

follower	1	2	3	4
p_i	0.5	0.5	1.1	1.1
q_i	0.3	-0.3	0.3	-0.3
r_i	0.0	0.0	0.0	0.0

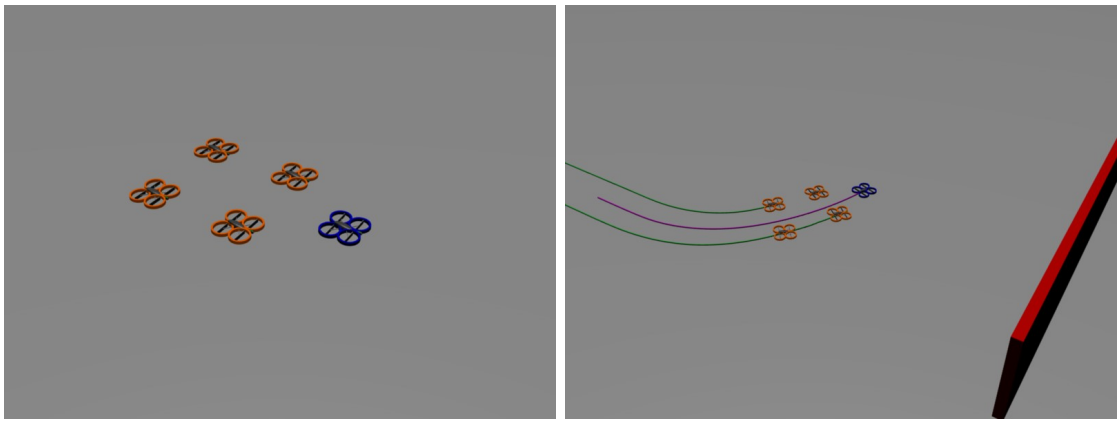
Table 4: Formation parameters

Obstacles are rendered red. The results of the simulation are visualized in figures 19 and 20.

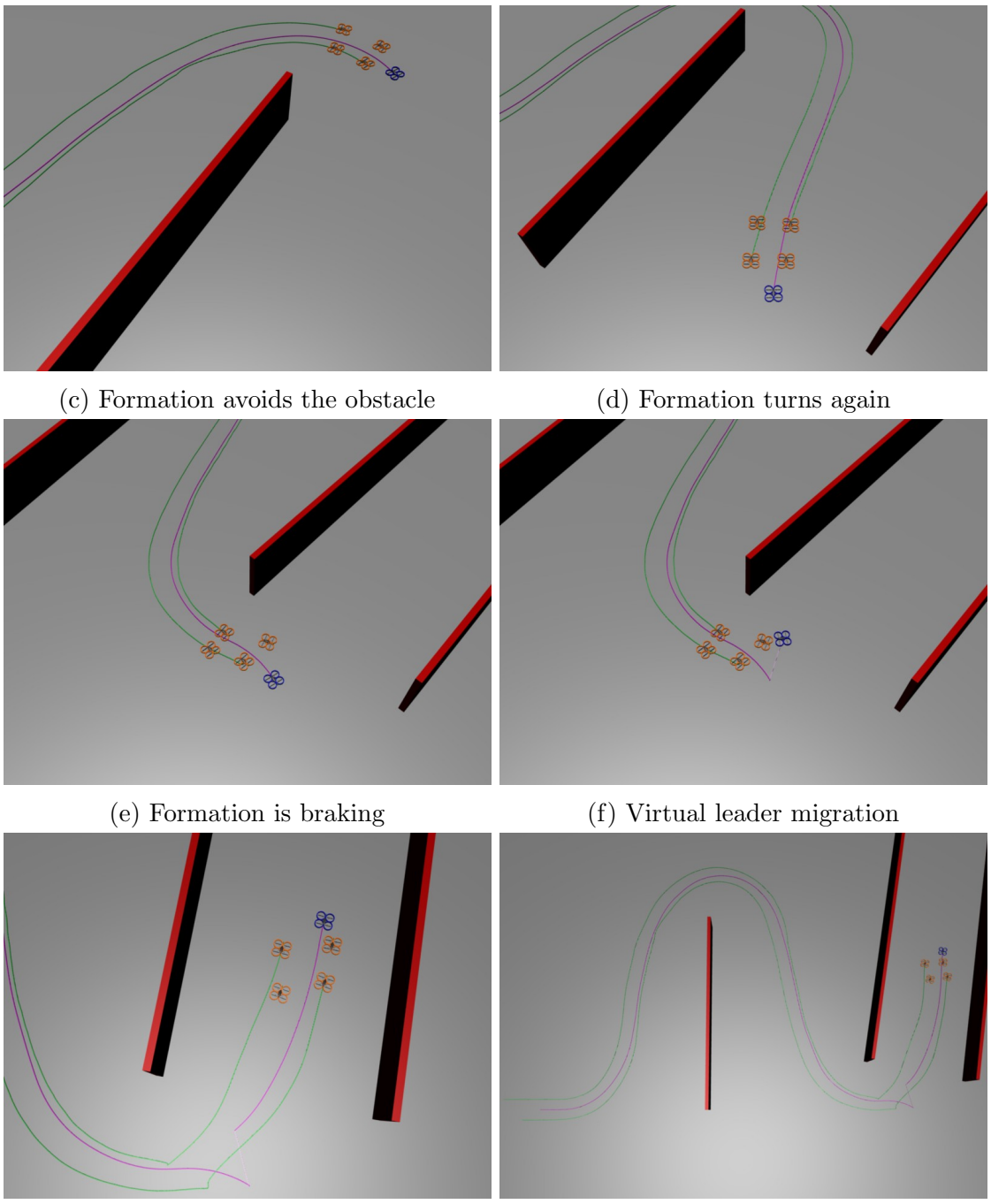


(a) Follower 2 (b) Follower 3

Figure 19: Objective functions of two followers in time.



(a) Starting position (b) Formation turns to avoid obstacles



(c) Formation avoids the obstacle

(d) Formation turns again

(e) Formation is braking

(f) Virtual leader migration

(g) Formation continues to fly

(h) Formation reaches the goal

Figure 20: Formation flying in a corridor.

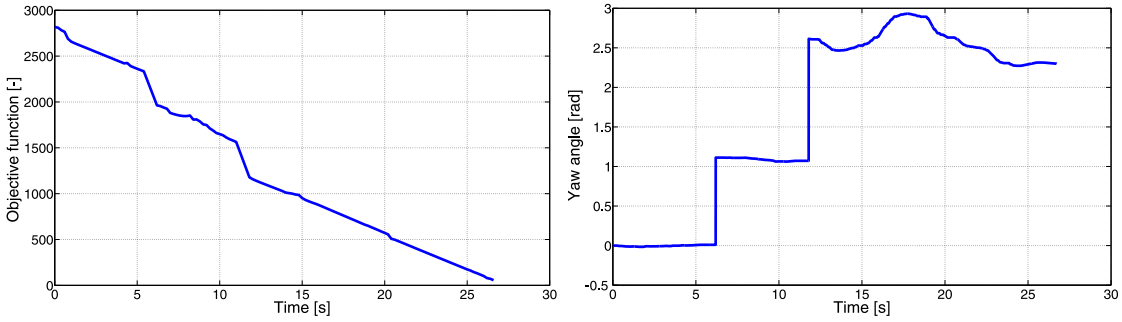
6.5 Flight with large number of MAVs

This scenario demonstrates the ability of the system to work with a large number of MAVs. The formation consists of a virtual leader (blue) and ten followers (orange). Formation parameters are in table 5. This experiment would be very difficult to perform with real quadrotors, due to the airflow from the top quadrotors affecting the bottom ones. For the purposes of testing, the airflow effect is not considered in this scenario.

follower	1	2	3	4	5	6	7	8	9	10
p_i	0.55	0.55	0.3	0.3	0.3	0.3	0.8	0.8	0.8	0.8
q_i	0	0.0	-0.4	-0.4	0.4	0.4	0.4	0.4	-0.4	-0.4
r_i	-0.25	0.25	0.25	-0.25	0.25	-0.25	0.25	-0.25	0.25	-0.25

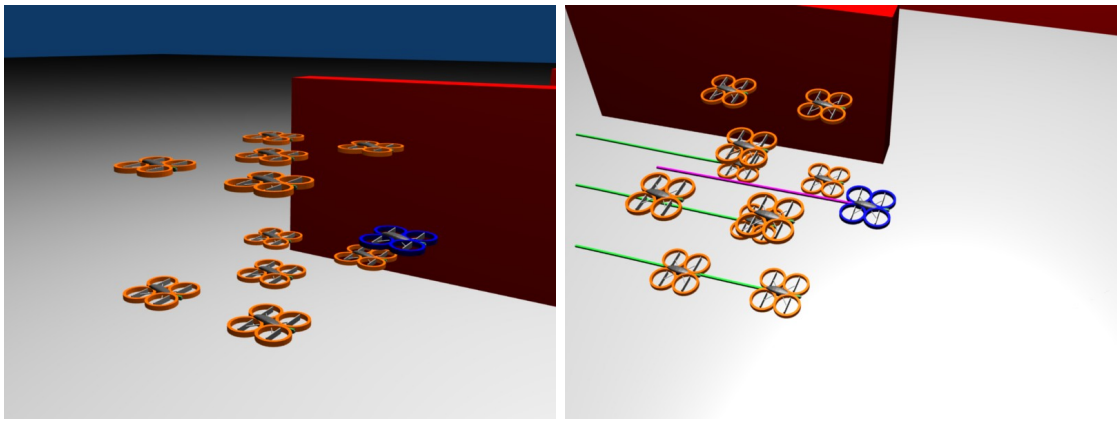
Table 5: Formation parameters

The results of the simulation are visualized in figures 21, 22 and 23.

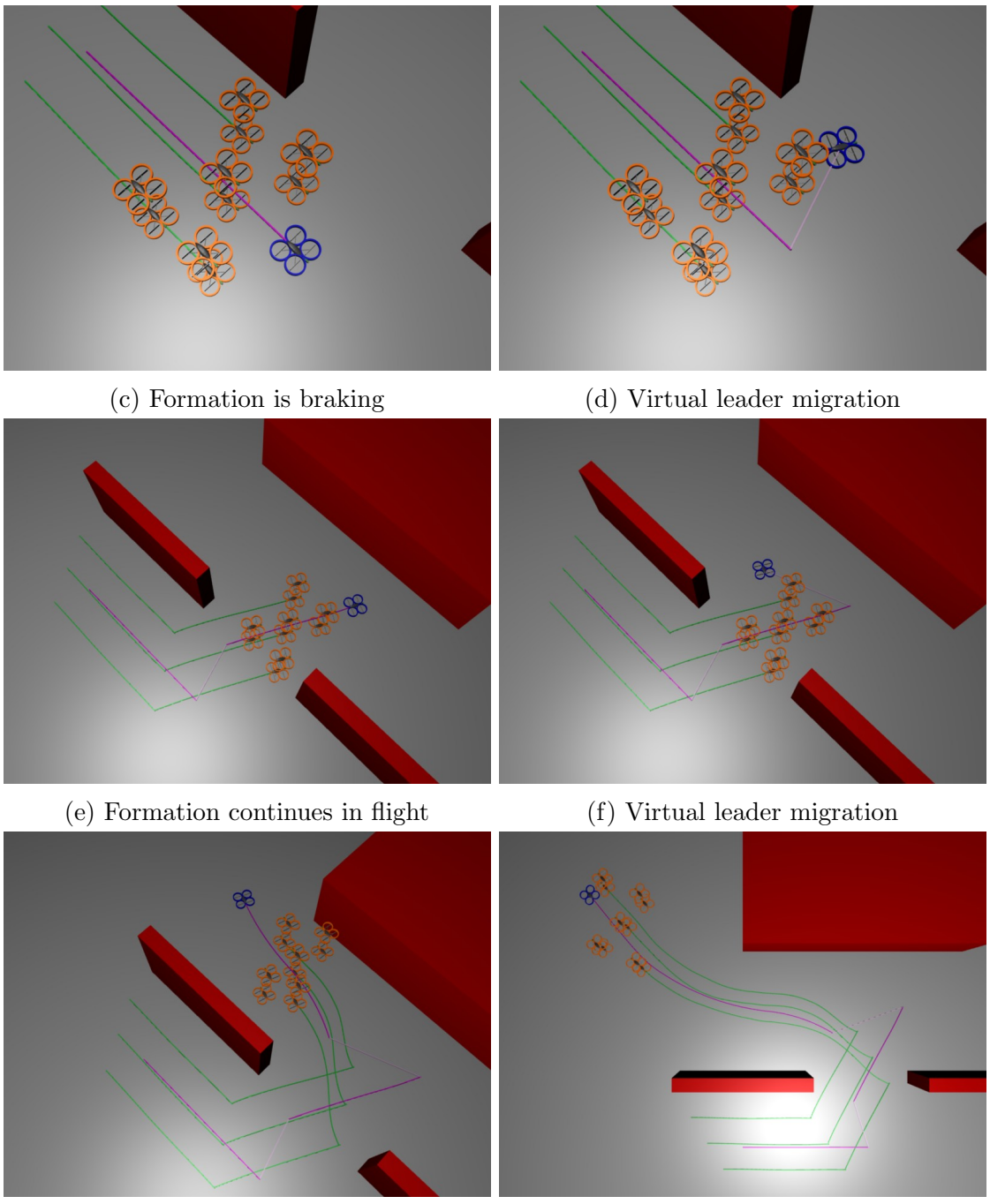


(a) Leader's objective function (b) Leader's yaw angle

Figure 21: Leader's objective function and yaw angle. The rotation maneuver is clearly visible in both graphs.



(a) Starting position (b) Formation approaches the obstacles



(c) Formation is braking

(d) Virtual leader migration

(e) Formation continues in flight

(f) Virtual leader migration

(g) Formation continues in flight

(h) Formation reaches the goal

Figure 22: Formation flying in a corridor. Airflow effect was not considered in this simulation

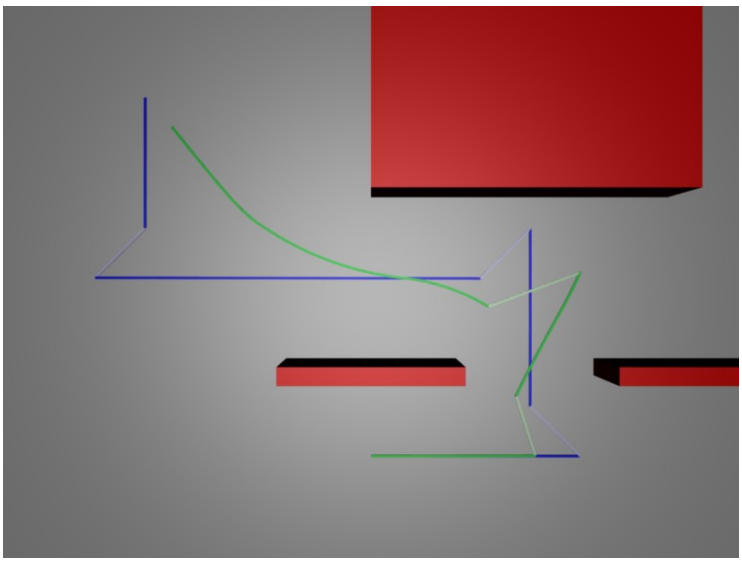
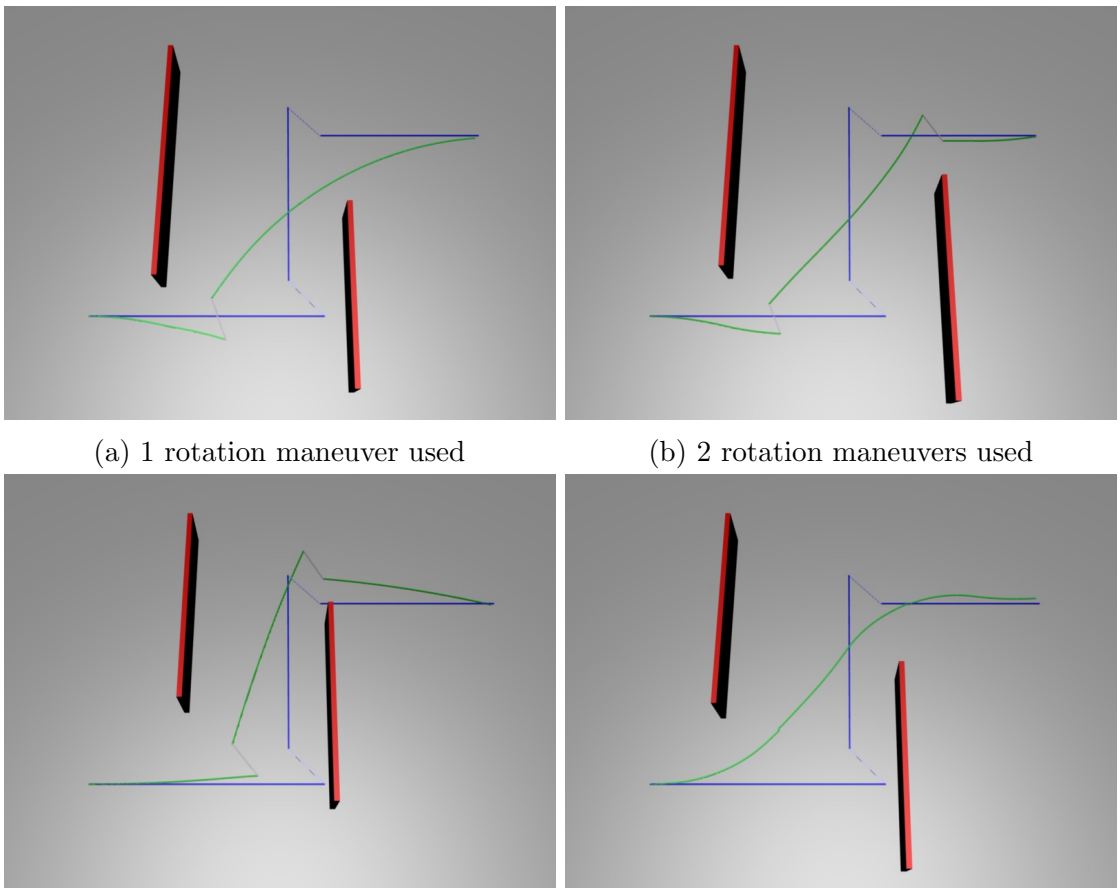


Figure 23: Comparison of the initial trajectory (blue) and actual leader's optimized trajectory (green). One rotation maneuver was autonomously replaced by a curve of the trajectory during the optimization process.

6.6 Statistical analysis

In this section, the rotation maneuver and its integration in the system is further demonstrated. 300 scenarios are generated for 7 different weights of the objective, to test the ability of the solver to plan trajectories using the rotation maneuver and to see the influence of different weights of the rotation maneuver penalty objective (presented in 5.3). The initial position, initial trajectory and goal are always the same. Two obstacles are randomly generated in such a way, that the initial trajectory remains feasible. Then the initial trajectory is optimized. 4 aspects of the trajectory are measured: time of flight, number of rotation maneuvers, minimal distance to obstacles and trajectory curvature, which is measured as the sum of absolute values of all the curvature control inputs multiplied by their duration. Four examples of generated scenarios are shown in figure 24. The solver was able to find a feasible solution in all of the tested scenarios.



(a) 1 rotation maneuver used

(b) 2 rotation maneuvers used

(c) 2 rotation maneuvers used

(d) No rotation maneuvers used

Figure 24: Examples of generated scenarios. Initial trajectory is blue, optimized trajectory is green and obstacles are red.

Results of the experiment are displayed as a box plot, where the red line represents the median, edges of the blue rectangle are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as red crosses. Points are drawn as outliers if they are larger than $q_3 + w(q_3 - q_1)$ or smaller than $q_1 - w(q_3 - q_1)$, where q_1 and q_3 are the 25th and 75th percentiles, respectively and w is the whisker length (value $w = 1.5$ was used in this case).

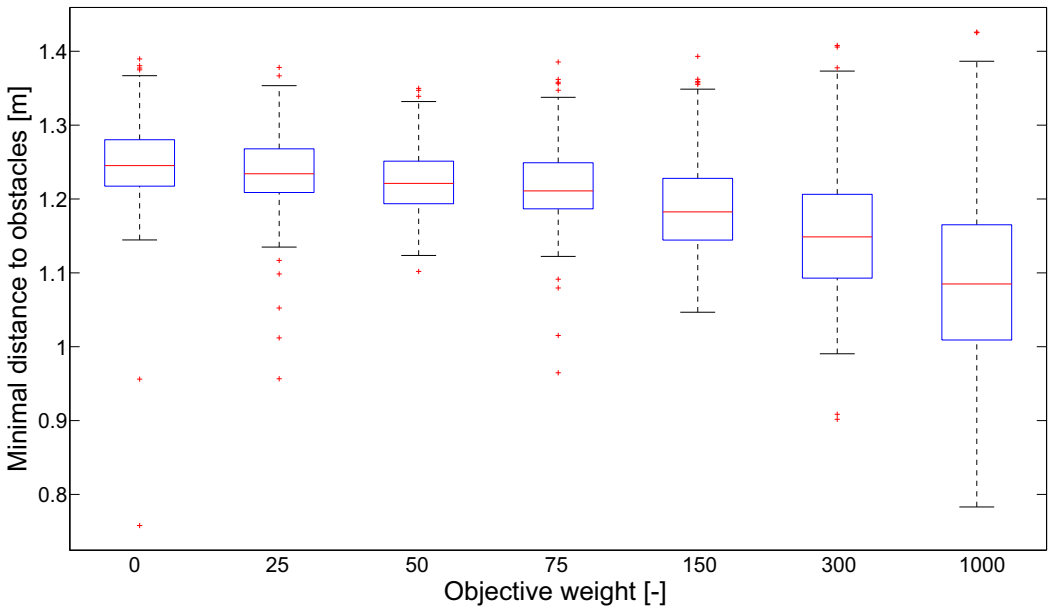


Figure 25: Minimal distance to obstacles

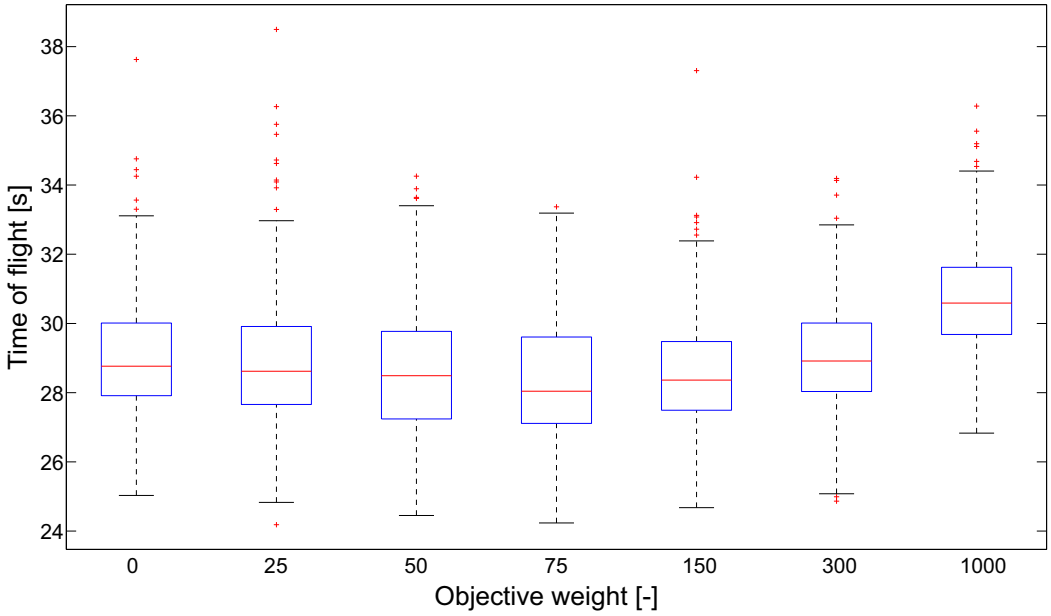


Figure 26: Time of flight

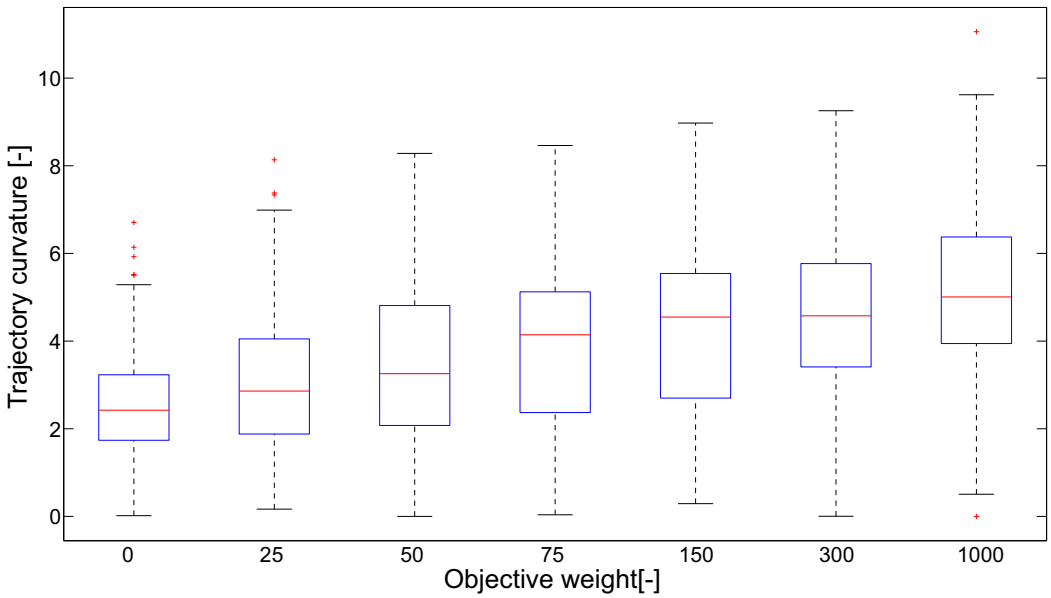


Figure 27: Trajectory curvature

objective weight	0	25	50	75	150	300	1000
2 rotation maneuvers	299	257	194	140	109	92	38
1 rotation maneuver	1	38	85	129	150	151	175
no rotation maneuvers	0	5	21	31	41	57	87
avg. time of flight	28.96	28.89	28.57	28.31	28.56	29.03	30.69
avg. min. obstacle distance	1.246	1.235	1.225	1.217	1.190	1.155	1.093
avg. trajectory curvature	2.564	3.082	3.470	3.891	4.171	4.433	5.035

Table 6: Influence of the rotation maneuver objective weight on the trajectory.

The influence of the rotation maneuver penalty objective weight is clearly visible in the results. With higher weight, lower number of rotation maneuvers is used. With the weight set to 0, two rotation maneuvers are used in almost all the cases. As the number of used rotation maneuvers gets lower, the curvature of the trajectory increases, as the rotation maneuvers are replaced with curves. Minimal distance to obstacles also decreases with higher weight. The shortest time of flight is achieved with the weight of 75. The trajectory curvature and minimal obstacle distance for the weight 75 is acceptable, and therefore weight of 75 is recommended for best results.

7 Conclusion

The goal of this thesis was to extend the 3D leader-follower formation driving method presented in [8], by a motion planning algorithm based on a dynamic virtual leader, which would allow the formation to perform sudden changes in the direction of flight. Used methods are described in chapter 4, including Model Predictive Control, Sequential quadratic programming, representation of the trajectory, trajectory planning for leader and for followers and obstacles.

The novel approach is described in the chapter 5. In section 5.1 the concept of a virtual migrating leader is introduced and its desired positioning is described. The rotation maneuver, which is described in section 5.2 is composed of four steps (formation hovering, virtual leader migration, recalculation of parameters, flight continuation). In section 5.3 the new objective is introduced to the objective function, to allow the solver to optimize the rotation maneuver. In section 5.4 Gilbert-Johnson-Keerhi distance algorithm is used to improve the representation of obstacles.

The functionality of the algorithm is then tested in chapter 6. Different scenarios are shown to demonstrate the capabilities of the system. In section 6.6, the system is subjected to a statistical analysis, which helps to determine the optimal rotation maneuver penalty objective weight and tests the system reliability.

All the points of the assignment were fulfilled. Dynamic virtual leader mechanism was designed, implemented and integrated into the system being developed by the Multi-robot Systems group for relative stabilization of MAVs. Work of the Multi-robot Systems group relevant to this thesis is presented in [10, 11, 12]. The system was verified in numerical simulations. The decision was taken by the thesis advisor to perform a statistical analysis, rather than HW experiment, due to unavailability of the robotic platform.

References

[1] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on*, 14(6):926 – 939, 1998.

[2] N. Basilico and S. Carpin. Online patrolling using hierarchical spatial representations. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2163 – 2169, 2012.

[3] I. Bayezit and B. Fidan. Distributed cohesive motion control of flight vehicle formations. *Industrial Electronics, IEEE Transactions on*, 60(12):5763 – 5772, 2012.

[4] J. Zhou C. Lawrence and A. Tits. User's guide for cfsqp version 2.5. University of Maryland, 1997.

[5] Stephen Cameron. Computing the distance between objects. Oxford University, 1996.

[6] C.V. Rao D.Q. Mayne, J.B. Rawlings and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[7] D.W Johnson E.G Gilbert and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *Robotics and Automation, IEEE Journal of*, 4(2):193 – 203, 1988.

[8] Zdeňek Kasl. 3d formations of unmanned aerial vehicles. Diploma thesis. Czech Technical University in Prague, Faculty of Electrical Engineering., 2013.

[9] W. Ren and R.W. Beard. Formation feedback control for multiple spacecraft via virtual structures. *Control Theory and Applications, IEE Proceedings*, 151(3):357 – 368, 2004.

[10] M. Saska, J. Chudoba, L. Preucil, J. Thomas, G. Loianno, A. Tresnak, V. Vonasek, and V. Kumar. Autonomous Deployment of Swarms of Micro-Aerial Vehicles in Cooperative Surveillance. In *Proceedings of 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, volume 1, pages 584–595, Danvers, 2014. IEEE Computer society.

[11] M. Saska, Z. Kasl, and L. Preucil. Motion Planning and Control of Formations of Micro Aerial Vehicles. In *Proceedings of The 19th World Congress of the International Federation of Automatic Control*, pages 1228–1233, Pretoria, 2014. IFAC.

[12] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil. Coordination and Navigation of Heterogeneous MAV UGV Formations Localized by a hawk-eye like Approach Under a Model Predictive Control Scheme. *International Journal of Robotics Research*, 33(10):1393–1412, September 2014.

[13] E. K. Wong T. Rakgowa, K.S. Sim and M.E. Nia. Master slave quadrotor formation for lifting force multiplication. *Applied Mechanics and Materials*, 705:133–136, 2014.

Appendix

CD Content

In table 7 are listed names of all root directories on CD

Directory name	**Description**
bp	bachelor thesis in pdf format.
sources	source codes
simulations	results of simulations

Table 7: CD Content