Martin Saska, Vojtěch Vonásek, Tomáš Krajník, Libor Přeučil

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, SAGE (2014).

DOI: 10.1177/0278364914530482

Copyright notice

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

Martin Saska[1,*], Vojtěch Vonásek[1], Tomáš Krajník[2], and Libor Přeučil[1]

[1] *Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague 6, Czech Republic*
[2] *Lincoln Centre for Autonomous Systems, School of Computer Science, University of Lincoln, Brayford Pool, Lincoln, UK*
[*] *Contact email: saskam1@fel.cvut.cz*

Abstract

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

Keywords:
formation control, model predictive control, unmanned aerial vehicles, unmanned ground vehicles, trajectory planning, obstacle avoidance, receding horizon control

1. Introduction

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

In this paper, we present a formation driving approach adapted for onboard visual relative localization of heterogeneous teams of unmanned helicopters (quadrotors) and ground robots. The localization is based on simple light-weight bottom cameras mounted on unmanned Micro Aerial Vehicles (MAVs). Identification patterns for the localization are placed on both the Unmanned Ground Vehicles (UGVs) and the MAVs. With this top-view approach, the problem of loss of direct visibility can be better tackled. Such problem occurs if systems of the visual relative localization are employed for ground robots operating in a workspace with scattered objects/obstacles, as is common in SAR scenarios. The possibility to localize robots from the top view increases robustness and precision in determining the relative position. Additionally, the top view brings another perspective for human operators supervising the mission. MAVs may also complement the team of UGVs with their ability to visit/search places inaccessible for ground vehicles, as we demonstrated in (Saska et al. 2012a). For more opportunities and advantages of MAVs, see the survey in (Kumar and Michael 2012).

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

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

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

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

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

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

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

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

In the literature, a direct inclusion of the trajectory planning into the formation driving is rarely found. To the best of our knowledge, we can mention only the leader-follower approach based on potential fields presented in (Garrido et al. 2011) as an appropriate example. This algorithm enables both formation stabilization and navigation of the formation into a desired goal. Although the method provides interesting results and seems to be computationally inexpensive, it has been developed for ground holonomic robots, and it suffers from the usual problems of algorithms inspired by potential fields. The authors state that their method behaved correctly in spite of the suddenly changed direction of movement around the sequence of points given by the planning method. However, it would be difficult to explicitly involve follower mobility constraints for a formation of nonholonomic robots. Further research would also be necessary to incorporate the requirements of 3D formations, and in particular the constraints given by the relative visual localization as proposed in the method presented here. Therefore, this method cannot be used for direct comparison with results achieved in this paper.

Finally, we should mention paper (Dorigo et al. 2012), which is similar to our work in terms of deployment of eye-bots on the ceiling with the aim to provide bird's eye view. In (Dorigo et al. 2012), the eye-bots are not moving together with ground robots if the top-view localization is in operation; they are fixed with the ceiling. Therefore, the motion coordination and formation driving of heterogeneous teams do not need to be solved there.

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

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

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

For descriptions and for a general survey of MPC methods, see (Barambones and Etxebarria 2000; Alamir 2006; Mayne et al. 2000) and the references reported therein. Early works applying MPC for formation control are presented in (Dunbar and Murray 2006; Franco et al. 2008). These papers utilized MPC for formation forming in a workspace without obstacles. Recently, researchers have taken advantage of MPC to respond to changes in a dynamic environment, again mainly in tasks including path tracking and formation stabilization (Chao et al. 2012; Defoort 2010; Zhang et al. 2010; Shin and Kim 2009; Chen et al. 2010; Saffarian and Fahimi 2009; Liu et al. 2011). In (Chao et al. 2012), the authors introduce a new cost penalty into MPC optimization to guarantee obstacle avoidance. A priority strategy is employed to ensure inter-vehicle collision avoidance. In (Defoort 2010), a decentralized receding horizon motion planner is developed to coordinate robots using neighbour-independent planning. This is followed by adjusting the plans with inter-team collisions using locally exchanged information. The trajectory tracking mechanism developed in (Zhang et al. 2010) is based on integrating a differential evolution algorithm into the MPC concept. In (Shin and Kim 2009), a heuristic approach is developed to reduce the required computational time of MPC iterations and to enable path tracking with an obstacle avoidance function. Formation stabilization on a pre-computed path based on the MPC leader-follower concept is presented in (Chen et al. 2010).

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

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

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

$n_r \in \mathbb{N}$	number of followers
$n_0 \in \mathbb{N}$	number of static and dynamic obstacles
L	variables related to the virtual leader
i	variables related to the i-th follower
j	variables related to the j-th entity, a follower or the virtual leader
C	configuration space of robots
C_{obs}	subspace of configurations of robots colliding with an obstacle
C_{free}	subspace of feasible configurations
o_l	the l-th obstacle
S_F	desired target region
$\varphi_j(t) \in \mathbb{R}$	heading of the j-th entity at time t
$\psi_j(t) \in \mathbb{R}^4$	configuration (position and heading) of the j-th entity at time t
$\bar{p}_j(t) \in \mathbb{R}^3$	position of the j-th entity at time t
$(x_j(t), y_j(t), z_j(t))$	position in Cartesian coordinates at time t
(p_i, q_i, h_i)	position of the i-th follower within the formation in curvilinear coordinates
$v_j(t) \in \mathbb{R}$	forward velocity of the j-th entity at time t
$K_j(t) \in \mathbb{R}$	curvature of the j-th entity at time t
$w_j(t) \in \mathbb{R}$	ascent velocity of the j-th entity at time t
$\bar{u}_j(t) \in \mathbb{R}^3$	control inputs (velocity and heading)
$\Delta t(k) \in \mathbb{R}$	time difference between the k-th and $(k+1)-$th transition points

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

2. Problem statement

In this paper, we consider the formation driving problem in scenarios motivated by search and rescue applications. We are interested in scenarios where a team of robots has to reach a desired target region or a sequence of target regions given by a supervising expert. During the movement between these given areas, the robots have to keep a fixed-shape formation satisfying the mission requirements. The robots can form a searching phalanx (a line formation) to be able to search for victims in large areas or they can form a compact fleet of vehicles (a formation of a general shape) for transportation purposes.

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

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

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

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

3. Preliminaries

Let $\psi_j(t) = \{x_j(t), y_j(t), z_j(t), \varphi_j(t)\} \in C$, with $j \in \{L, 1, \ldots, n_r\}$, denote configurations of a virtual leader L and n_r followers at time t. The virtual leader is positioned in front of the formation and on the axis of the formation, which is important for the symmetric obstacle avoidance function. C is the configuration space of the robots. The Cartesian coordinates $x_j(t)$, $y_j(t)$ and $z_j(t)$ define the positions $\bar{p}_j(t)$ of the robots, and $\varphi_j(t)$ denotes their heading. All MAVs and UGVs are denoted as followers in the approach presented here. For the MAVs, the heading $\varphi_j(t)$ becomes directly the yaw (see Fig. 1 for the coordinates system of the MAVs). The roll and the pitch do not need to be included directly in the kinematic model employed in the MPC. They depend on the velocity and the turning curvature, as shown for a quadrotor helicopter in Appendix A.

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

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

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

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

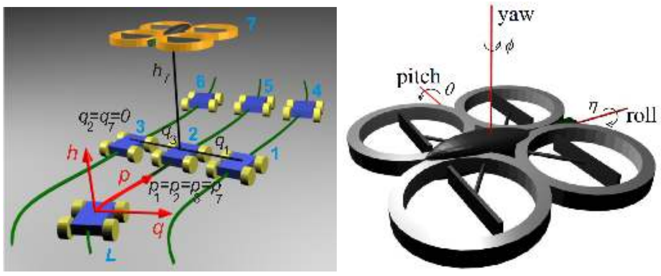
Forward velocity $v_j(t)$, curvature $K_j(t)$ and ascent velocity $w_j(t)$ represent control inputs denoted as $\bar{u}_j(t) = \{v_j(t), K_j(t), w_j(t)\}$. For UGVs (in the presented results for car-like robots), these control inputs can be directly employed for steering them[1]. In case of MAVs, $v_j(\cdot)$, $K_j(\cdot)$ and $w_j(\cdot)$ values are inputs for the controller shown in Appendix A.

Let us now define a time interval $[t_0, t_{end}]$ containing a finite sequence of elements of increasing time $\{t_0, t_1, \ldots, t_{end-1}, t_{end}\}$, such that $t_0 < t_1 < \ldots < t_{end-1} < t_{end}$. The control inputs are held constant in each time interval $[t_k, t_{k+1})$, where $k \in \{0, \ldots, end-1\}$. From this point we may refer to t_k by using its index k. By integrating the kinematic model over the interval $[t_0, t_{end}]$, we can derive the following model for *transition points* at which the control inputs change:

$$\begin{aligned} x_j(k+1) &= \begin{cases} x_j(k) + \frac{1}{K_j(k+1)} \Big[\sin\Big(\varphi_j(k) + \\ K_j(k+1) v_j(k+1) \Delta t(k+1)\Big) - \\ \sin\Big(\varphi_j(k)\Big)\Big], \text{if } \ K_j(k+1) \neq 0; \\ x_j(k) + v_j(k+1) \cos\Big(\varphi_j(k)\Big) \Delta t(k+1), \\ \text{if } \ K_j(k+1) = 0 \end{cases} \\ y_j(k+1) &= \begin{cases} y_j(k) - \frac{1}{K_j(k+1)} \Big[\cos\Big(\varphi_j(k) + \\ K_j(k+1) v_j(k+1) \Delta t(k+1)\Big) - \\ \cos\Big(\varphi_j(k)\Big)\Big], \text{if } \ K_j(k+1) \neq 0; \\ y_j(k) + v_j(k+1) \sin\Big(\varphi_j(k)\Big) \Delta t(k+1), \\ \text{if } \ K_j(k+1) = 0 \end{cases} \\ z_j(k+1) &= z_j(k) + w_j(k+1) \Delta t(k+1) \\ \varphi_j(k+1) &= \varphi_j(k) + K_j(k+1) v_j(k+1) \Delta t(k+1), \end{aligned} \tag{2}$$

where $x_j(k)$, $y_j(k)$ and $z_j(k)$ are Cartesian coordinates and $\varphi_j(k)$ is the heading angle at the transition point with index k for any robot $j \in \{L, 1, \ldots, n_r\}$. The sampling time $\Delta t(k+1)$ may not be uniform in the whole interval $[t_0, t_{end}]$, as shown below. The control inputs $v_j(k+1)$, $K_j(k+1)$ and $w_j(k+1)$ are constant between the transition points with indexes k and $k+1$. For each follower $i \in \{1, \ldots, n_r\}$, the control inputs are limited by vehicle kinematic constraints (i.e., implied by the steering limitations

[1] We assume that UGVs operate on a flat surface and that $z_j(\cdot) = 0$ and $w_j(\cdot) = 0$ for each of the UGVs.



(a) (b)

Figure 1: (a) The desired shape of the formation described in curvilinear coordinates. (b) Coordination system of the quadrocopter.

and the drive system) as $v_{min,i} \leq v_i(k) \leq v_{max,i}$, $|K_i(k)| \leq K_{max,i}$ and for the MAVs also $w_{min,i} \leq w_i(k) \leq w_{max,i}$. These values may differ for each of the followers.

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

3.1. Formation driving concept

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

In the proposed method, both types of followers, MAVs and UGVs, follow the same trajectory of the virtual leader in distances defined in the p, q, h curvilinear coordinate system, as visualized in Fig.1(a). The position of each follower i is uniquely determined: 1) by states $\psi_L(t_{p_i})$ in the *travelled distance* p_i from the actual position of the virtual leader along the leader's trajectory, 2) by the *offset distance* q_i from the trajectory in the perpendicular direction and, 3) by the elevation h_i above the trajectory. t_{p_i} denotes the time when the virtual leader was at the *travelled distance* p_i behind the actual position.

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

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

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

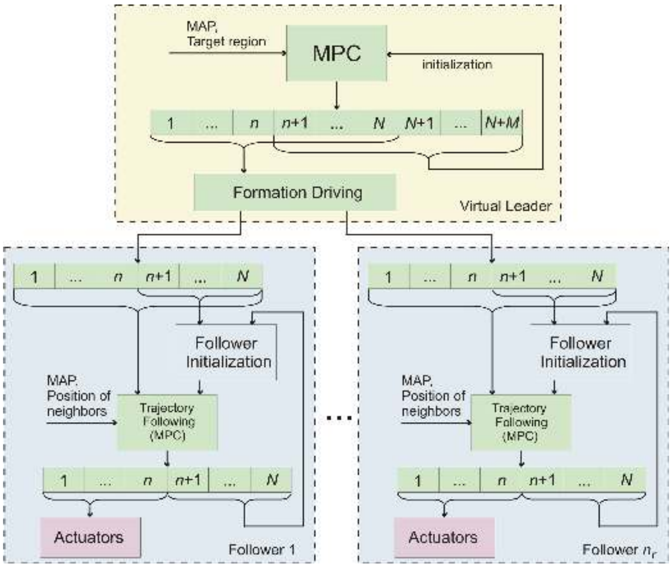


Figure 2: Scheme of the complete planning and control system.

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

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

These restrictions must be applied to satisfy different values for the curvature and the speed of the robots in different positions within the formation. Intuitively, the robot following the inner track during turning goes more slowly but with a bigger curvature than the robot further from the center of the turning. The equations arise from the fact that the followers turn around the same Instantaneous Center of Curvature (ICC) and at the same angular speed. These restrictions ensure that the formation remains compact while turning.

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

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

4.1. Method overview

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

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

The core of the second main block is the *Trajectory Following* module. This part enables the design of appropriate collision free control inputs for each of the MAV and UGV follow-

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

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

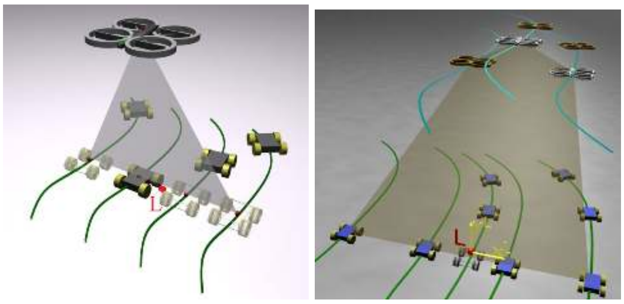


Figure 3: Two examples of convex hulls of asymmetric formations. The formation on the right side is utilized in the experiment in Fig. 6. The shaded contours represent projections of MAVs and UGVs into the plane of the virtual leader.

ers. It is responsible for avoiding impending collisions with obstacles or with other members of the team, and it corrects deviations from the desired trajectory provided by the virtual leader. Again this is ensured by the MPC concept using the actual data - states of neighbours and the map - which is shared within the team. Details on the control of followers can be found in Section 4.4.

4.2. Convex hull representing the formation in the trajectory planning

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

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

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

4.3. Trajectory planning and control for the virtual leader

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

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

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

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

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

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

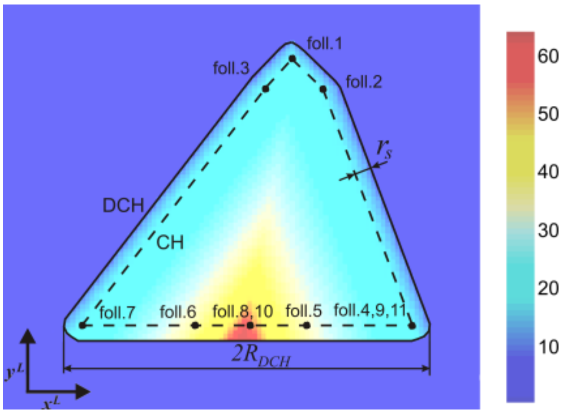


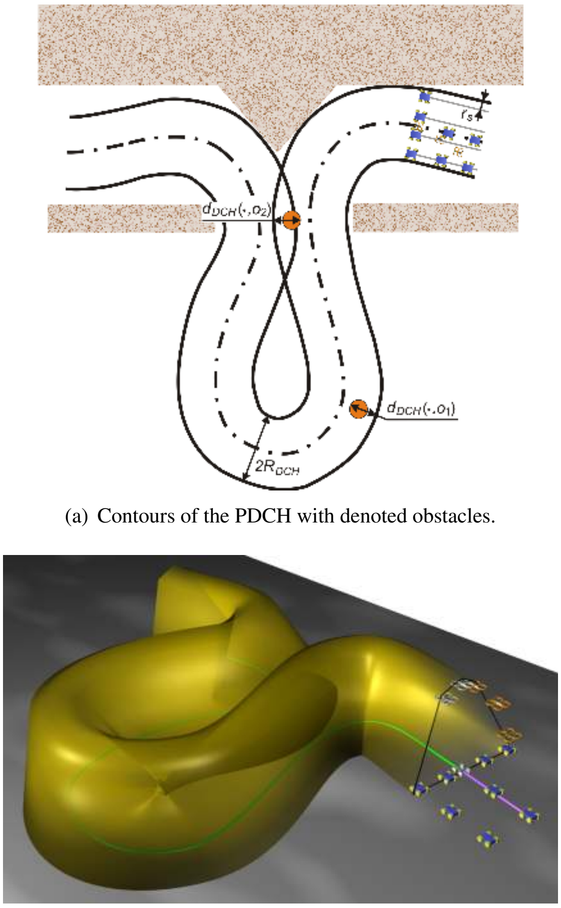
Figure 4: A color map of the function that ensures collision free trajectories for a formation operating under the *hawk-eye* concept. The color map was composed for the second formation introduced in Fig. 3.

zero value at the borders of PDCH are required properties necessary for optimization convergence into a feasible solution.

These properties are obtained as follows. The constant R_{DCH} is equal to half of the maximal width of DCH measured in the x^L coordinate. The meaning of R_{DCH} is also denoted in Fig. 4 and 5. The function $d_{DCH}(\Omega_L, o_l)$ provides the shortest distance in the direction of the x^L coordinate from the furthermost part of obstacle o_l to the borders of PDCH. See the obstacle o_1 and the denoted value $d_{DCH}(\cdot, o_1)$ for illustration in Fig. 5. The function value is positive if the obstacle is in PDCH and negative if the obstacle is completely outside the projected hull. If an obstacle occurs in several projections of DCH along Ω_L (e.g. in a sharp curve of the trajectory), it is counted only once in eq. (6). Always, only the largest value of the set of the shortest distances from the obstacle to the border of the relevant projection of DCH is used (see obstacle o_2 in Fig. 5). The direction of the gradient of such defined avoidance function (see the values of this function depending on the position of an obstacle in Fig. 4) is to the side of DCH in the x^L coordinate. This feature is important for the convergence of the optimization process into an obstacle free solution. If the resulting trajectory changes its shape during the optimization with the aim to have the obstacle outside DCH, the value of $J_{L,obstacles}(\Omega_L)$ decreases smoothly. This is important for the convergence of the optimization into a feasible solution.

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

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



(a) Contours of the PDCH with denoted obstacles.

(b) 3D visualization of the PDCH (the obstacles are hidden).

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

$\forall k \in \{N+1, \ldots, N+M\}$, for the planning horizon, characterize the limits on the control inputs (eq. (4)) of the virtual leader. Furthermore, the constraints $g_{T_M}(k)$ ensure that inequalities $\Delta t(k) \geq 0$ are satisfied for $\forall k \in \{N+1, \ldots, N+M\}$.

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

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

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

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

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

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

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

4.4. Trajectory Tracking for Followers

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

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

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

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

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

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

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

The shape of $J_i(\Omega_i)$ allows the repositioning of followers (UGVs and MAVs) with the aim of obstacle avoidance, compensation of actuators and sensors uncertainty or collision avoidance within neighbours. Each UGV follower i can change its position and heading by optimizing its curvature K_i and velocity v_i. The MAV followers may also change their altitude z_i by optimizing their ascent velocity w_i.

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

5. Analysis of formation convergence from a feasible initial solution

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

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

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

Assumption 1. *(Desired Reachability) At the initial time t_0, there exists a feasible solution of the optimization problem introduced in eq. (5). The solution represents the trajectory for the formation to reach the target region. It guarantees that the trajectory is situated at a sufficient distance from obstacles and that direct visibility between the robots is ensured, which is a crucial aspect of the relative localization under the hawk-eye concept. In addition, the utilized optimization method is capable of finding such a solution, not necessarily globally optimal, from the initial configuration $\psi_L(t_0) \in C_{free}$ to any configuration $\psi_L(t_f)$, with $t_f > t_0$, which is inside the target region.*

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

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

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

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

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

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

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

and

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

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

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

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

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

and

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

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

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

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

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

and similarly

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

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

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

$$J_L(\Omega_L(t_1;t_3)) \leq J_L(\Omega_L(t_1;t_2)) + J_L(\Omega_L(t_2;t_3)) \tag{19}$$

□

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

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

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

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

□

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

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

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

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

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

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

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

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

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

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

Collecting equations (21) and (22) we obtain

$$\begin{aligned} &- J_L(\Omega_L(\tau; n\Delta t + \tau)^\circ) - J_L(\Omega_L(n\Delta t + \tau; t_{T1})^\circ) + \\ &+ J_L(\Omega_L(n\Delta t + \tau; t_{T2})^\circ) < 0. \end{aligned} \tag{23}$$

Let us substitute

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

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

□

5.1. Analysis of results of the convergence proof

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

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

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

$$D(k) < n\Delta t. \tag{25}$$

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

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

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

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

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

Theorem 5.4 and the related proof can be utilized for the convergence analysis, similarly as was done for the static target region. Only the practical meaning of the perturbations $D(k)$ is changed. The uncertainty of the actuators and the imprecise kinematic model are still included in the perturbations according to eq. (24). Nevertheless, the prolonged total time to the goal given by the time difference $T_2 - T_1$ is caused mainly by the movement of the former formation, which is followed by the new formation. To better understand the problem, let us split the perturbations into two parts: $D(k)$, which represents the perturbations given by the imprecise actuators and model, and $D_{drift}(k)$, which includes the influence of the drift of the dynamic target region (the required place of the unstuck followers in the former formation). The equation $D_{total}(k) = D(k) + D_{drift}(k)$ holds for the situation with the dynamic target region, whereas, in the analysis in the previous subsection, the equations $D_{drift}(k) = 0$ and $D_{total}(k) = D(k)$ hold.

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

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

6. Experimental results

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

Table 3: Curvilinear coordinates of followers within the formation used in the experiment presented in Fig. 6.

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

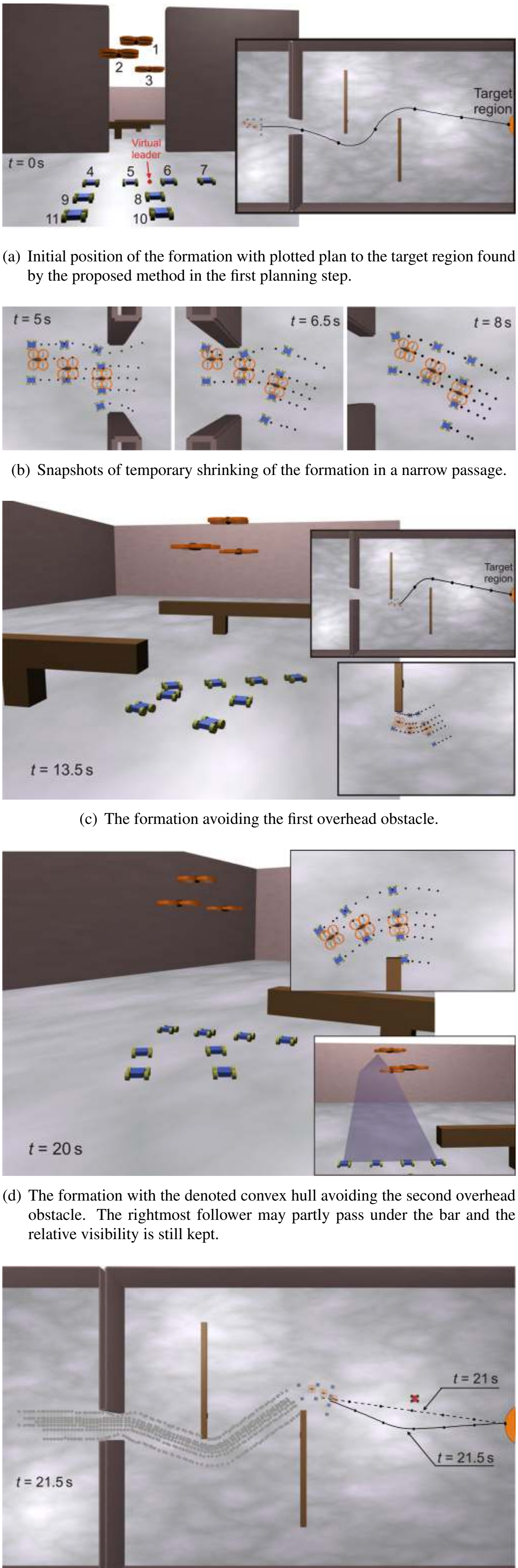
The presented results were obtained using the proposed algorithm with the following parameters: $n = 2$, $N = 4$, $M = 6$, $\alpha = 1$, $\alpha_i = 1$, $\beta_i = 1$ and $\Delta t = 0.25s$. We employed Sequential Quadratic Programming (SQP) (Nocedal and Wright 2006) for solving the optimization problems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

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

In the first experiment presented in this section, a formation of 11 followers (8-UGVs and 3-MAVs) has to move into a target region through an environment with two overhead obstacles and one dynamic obstacle. See Table 3 for the parameters of the formation and Fig. 6 for snapshots of the experiment. Fig. 7 illustrates a progress of the values of the cost function used for the virtual leader's trajectory planning (eq. (6)) and the values of the cost function employed for the stabilization of the followers (eq. (9)).

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

In Fig. 6(c) and 6(d), the formation passes by obstacles that verify the ability of the approach to keep direct visibility between team members. Although the UGVs could pass under the obstacles and the MAVs could fly over them (which would decrease the time to the goal), the planned trajectory leads around the obstacles to keep them outside the convex hull representing the formation. Due to the shape of the hull and the positions of the obstacles, the UGVs may go under the obstacles only partly to follow the trajectory that is as short as possible according to



(a) Initial position of the formation with plotted plan to the target region found by the proposed method in the first planning step.

(b) Snapshots of temporary shrinking of the formation in a narrow passage.

(c) The formation avoiding the first overhead obstacle.

(d) The formation with the denoted convex hull avoiding the second overhead obstacle. The rightmost follower may partly pass under the bar and the relative visibility is still kept.

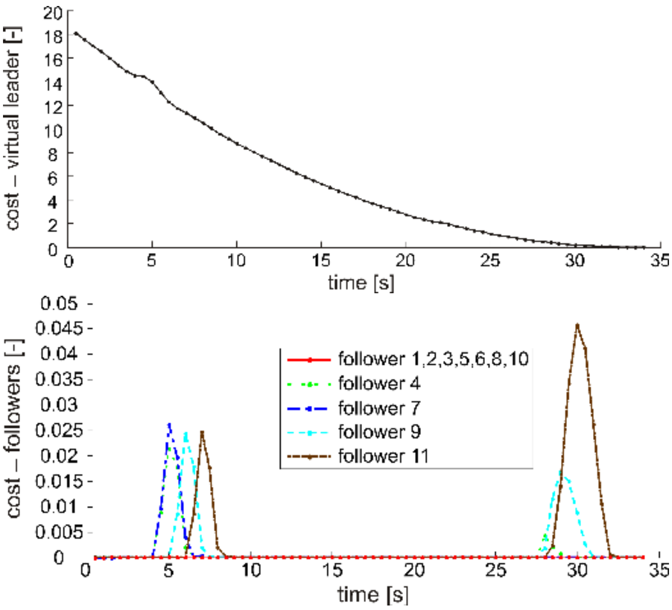


Figure 7: Progress of the values of the cost function used for the virtual leader's trajectory planning, eq. (6), and the cost function employed for stabilization of the followers in their desired positions behind the virtual leader, eq. (9). The decrease in the values of eq. (6) shows the convergence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

the optimization problem specified in eq. (5).

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

The results of a statistical test of the performance of the algorithm are presented in Table 4. The aim is to show the reliability and the practical utilization of the method with different values of parameter M, whose proper setting is crucial for the deployment of the system. For the test, a set of 1000 positions of the target region was randomly generated in the free space on the right side of the workspace introduced in Fig. 6(a). The algorithm was tested with different values of M for each of these configurations of the target, using Intel Core Duo CPU 3.2GHz, 4GB RAM. The simulations are counted as successful if the formation reached the desired target region without collisions and with relative visibility kept during the entire movement. The total time to reach the goal indicates the quality of the solutions.

The most time consuming part of the proposed MPC approach is evaluation of the objective function in eq. (6). This function is recalled in each iteration of the optimization process. The second term of the function, which implements the obstacle avoidance, represents major contribution to the computational demands. In particular, two variables influence computational complexity of the formation driving system that is proposed

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

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

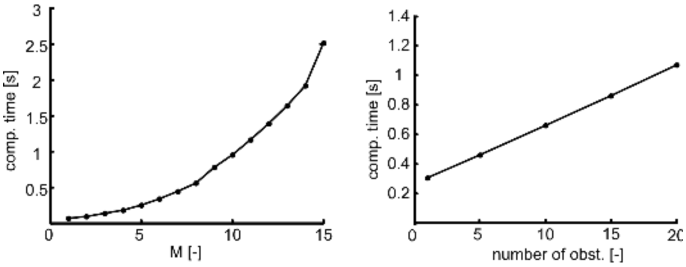


Figure 8: Computational demands of the method - influence of the number of obstacles and the number of transition points at the *planning interval* T_M. The mean computational time of one MPC planning step was obtained from 1000 runs of the algorithm.

here: the number of obstacles considered for the planning, and the dimension of the optimization vector. Fig. 8 shows that the mean computational time linearly depends on the number of obstacles. This confirms the expectations, since the distance from the convex hull is computed separately for each of the obstacles only in the second term of eq. (6). The distance-to-obstacle calculation is the most computationally intensive part of the algorithm.

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

The simulation presented in Fig. 9 shows the performance of the algorithm in the situation where an obstacle (the traverse beam under the ceiling) blocks the MAV follower from reaching the desired target region at the desired altitude (h_i coordinate). Similarly to the previous experiment, where the formation passed a narrow corridor, the formation is forced to change its desired shape temporarily.

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

In the real experiment presented in Fig. 11, a formation of three ground robots and one helicopter has to move from its initial locations into the desired target region. The trajectory planning method presented in this paper was employed to verify practically the usefulness of the visual relative localization and consequently the stabilization of followers from flying robots.

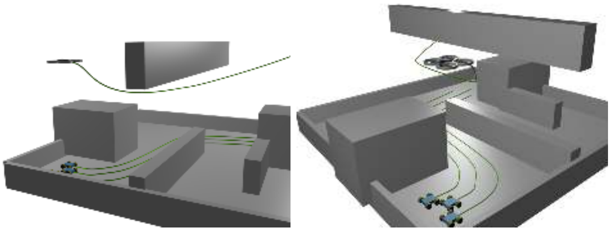


Figure 9: An example of the trajectories passed by a formation of four followers (3-UGVs and 1-MAV). The MAV follows the virtual leader, while it descends to avoid the top obstacle.

Two different UGVs, one G2Bot platform (Chudoba et al. 2006) (the bigger robot in Fig. 11) and two MMP5 platforms are used in this experiment. To verify the concept of the proposed 3D formations, Ar.Drone quadrocopter is used as a flying follower. The MAV is equipped with a vision system to be able to follow the proposed *hawk eye* approach. It carries a bottom monocular camera supplemented by a vision algorithm (Saska et al. 2012b) that is able to identify the location and the size of the color markers of the UGVs in the image. This information is used for the relative localization of all members of the formation. The estimated relative positions from such *hawk eye* are sent to the UGV followers over WiFi link as feedback to maintain the predefined formation shape.

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

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

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

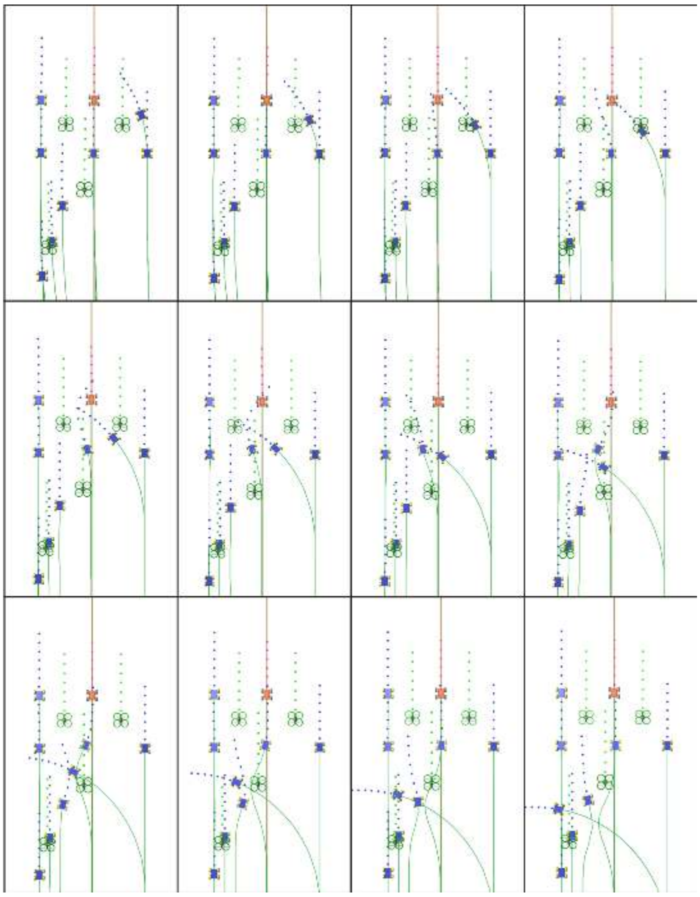
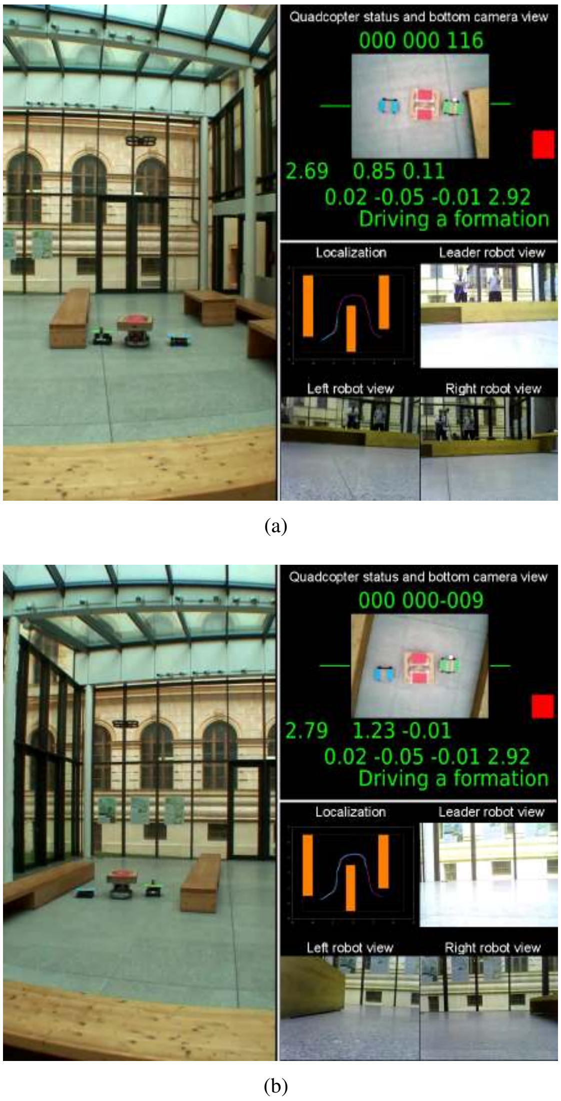


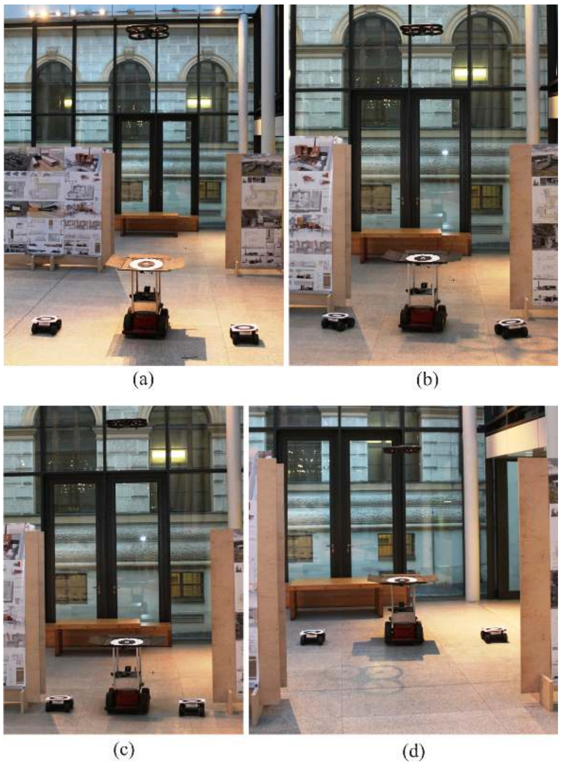
Figure 10: A sequence of snapshots presenting the failure tolerance of the system by simulating a follower failure.



(a)

(b)

Figure 11: Snapshots from the formation driving experiment with 3-UGVs and 1-MAV following a virtual leader.



(a) (b) (c) (d)

Figure 12: Experiment verifying formation shrinking while it is driven through a narrow corridor.

tails, software implementation and interfacing to AR-drone, see (Krajnik 2013). The system presented in this paper enables to stabilize the heterogeneous formation up to the speed of $0.7m/s$. For the experimental evaluation presented here, the speed of the leader was limited to $0.3m/s$ for safety reasons.

7. Conclusion

A control methodology developed for formation driving of 3D heterogeneous UGV-MAV formations stabilized via the *hawk-eye* like visual localization is presented in this paper. A novel MPC schema is introduced with an integrated obstacle avoidance function that ensures direct visibility between MAVs and UGVs. The visibility between MAVs and UGVs is crucial for the top-view relative localization of the team members. This may act as an enabling technique for real-world deployment of formations of micro-scale robots. Our experiments show the performance of the method and verify its robustness in an environment with dynamic obstacles. In addition, the requirements for practical utilization of the method are specified by sound theoretical analyses. As a result of these analyses, we propose a simple mechanism to detect and tackle eventual violations of the convergence of the 3D formations movement into the target region.

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

8. ACKNOWLEDGMENTS

The work was supported by the Grant Agency of the Czech Republic under postdoc grant no. P103-12/P756 and by MŠMT under project Kontakt II no. LH11053. The European Union supported this work within its Seventh Framework Programme project ICT-600623 “STRANDS”. Our special thanks go to Tom Duckett, Nicola Bellotto and Sam Malone from the University of Lincoln for their careful proofreading and valuable comments and to Dušan M. Stipanovic and Juan S. Mejía from the University of Illinois at Urbana-Champaign for very useful advises concerning theoretical issues of the paper. We would also like to thank the anonymous reviewers and the Editors for their valuable comments.

Appendix A. Low level controller designed for AR-Drone

In the kinematic model described in Section 3, it is assumed that the MAVs can follow a trajectory containing segments with a given curvature. Further, we assume that MAVs are controlled using forward speed v, curvature K and ascent w. We used the Ar.Drone quadrocopter, which allows changes in the speed of its rotors and consequently its pitch, yaw and roll angles (see Fig. 1(b) for the angles definition).

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

$$\tan\theta = -\frac{c_0 v}{F_g}. \tag{A.1}$$

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

$$\tan\eta = -\frac{v^2 K}{g}. \tag{A.2}$$

When the pitch and roll angles are small, the approximation $\tan x \sim x$ can be used in (A.1) and (A.2). To control the drone

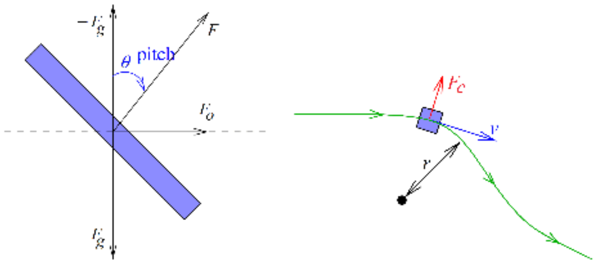


Figure A.13: Simplified model of the quadrocopter.

along a trajectory with the given curvature K and known forward speed v, pitch θ and roll η are controlled. Further details on Ar.Drone control can be found in (Krajník et al. 2011).

References

Abdessameud, A., Tayebi, A., 2011. Formation control of vtol unmanned aerial vehicles with communication delays. Automatica 47 (11), 2383 – 2394.

Alamir, M., 2006. Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes. Vol. 339 of Lecture Notes in Control and Information Sciences. Springer, Berlin / Heidelberg.

Barambones, O., Etxebarria, V., 2000. Robust adaptive control for robot manipulators with unmodelled dynamics. Cybernetics and Systems 31 (1), 67–86.

Barfoot, T. D., Clark, C. M., February 2004. Motion planning for formations of mobile robots. Robotics and Autonomous Systems 46, 65–78.

Beard, R., Lawton, J., Hadaegh, F., November 2001. A coordination architecture for spacecraft formation control. IEEE Transactions on Control Systems Technology 9 (6), 777 – 790.

Bullo, F., Cortés, J., Martínez, S., Jun. 2008. Distributed Control of Robotic Networks.

Burdakov, O., Doherty, P., Holmberg, K., Kvarnstrom, J., Olsson, P.-M., 2010. Relay positioning for unmanned aerial vehicle surveillance. International Journal of Robotic Research 29 (8), 1069–1087.

Chao, Z., Zhou, S.-L., Ming, L., Zhang, W.-G., 2012. Uav formation flight based on nonlinear model predictive control. Mathematical Problems in Engineering 2012 (1), 1–16.

Chen, J., Sun, D., Yang, J., Chen, H., May 2010. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. Int. Journal Robotic Research 29, 727–747.

Chudoba, J., Mazl, R., Preucil, L., 2006. A control system for multi-robotic communities. In: In ETFA 2006.

Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., Taylor, C., October 2003. A vision-based formation control framework. IEEE Transactions on Robotics and Automation 18 (5), 813–825.

Defoort, M., sept. 2010. Distributed receding horizon planning for multi-robot systems. In: IEEE International Conference on Control Applications (CCA). pp. 1263 –1268.

Desai, J., Ostrowski, J., Kumar, V., December 2001. Modeling and control of formations of nonholonomic mobile robots. IEEE Transactions on Robotics and Automation 17 (6), 905–908.

Do, K. D., Lau, M. W., 2011. Practical formation control of multiple unicycle-type mobile robots with limited sensing ranges. Journal of Intelligent and Robotic Systems 64 (2), 245–275.

Dong, W., 2011. Robust formation control of multiple wheeled mobile robots. Journal of Intelligent and Robotic Systems 62 (3-4), 547–565.

Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., Birattari, M., Bonani, M., 2012. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. IEEE Robotics & Automation Magazine, in press.

Dunbar, W., Murray, R., April 2006. Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42 (4), 549–558.

Faigl, J., Krajník, T., , Chudoba, J., Preucil, L., Saska, M., 2013. Low-cost embedded system for relative localization in robotic swarms. In: Proc. of IEEE International Conference on Robotics and Automation.

Franco, E., Magni, L., Parisini, T., Polycarpou, M., Raimondo, D., February 2008. Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. IEEE Transactions on Automatic Control 53 (1), 324–338.

Fredslund, J., Mataric, M., October 2002. A general algorithm for robot formations using local sensing and minimal communication. IEEE Transactions on Robotics and Automation 18 (5), 837–846.

Garrido, S., Moreno, L., Lima, P. U., 2011. Robot formation motion planning using fast marching. Robotics and Autonomous Systems 59 (9), 675 – 683.

Ghommam, J., Mehrjerdi, H., Saad, M., Mnif, F., 2010. Formation path following control of unicycle-type mobile robots. Robotics and Autonomous Systems 58 (5), 727 – 736.

Hengster-Movrić, K., Bogdan, S., Draganjac, I., 2010. Multi-agent formation control based on bell-shaped potential functions. Journal of Intelligent and Robotic Systems 58 (2), 165–189.

Hess, M., Saska, M., Schilling, K., October 2009. Application of coordinated multi vehicle formations for snow shoveling on airports. Intelligent Service Robotics 2 (4), 205 – 217.

Khalil, H., 2001. Nonlinear Systems, 3rd Edition. Prentice Hall, Michigan State University, MI.

Klančar, G., Matko, D., Blažič, S., 2011. A control strategy for platoons of differential drive wheeled mobile robot. Robotics and Autonomous Systems 59 (2), 57 – 64.

Krajnik, T., 2013. Ardrone quadcopter in robotics research [online]. http://labe.felk.cvut.cz/ tkrajnik/ardrone/ [cit. 2013-9-19].

Krajník, T., Faigl, J., Vonásek, V., Kosnar, K., Kulich, M., Preucil, L., 2010. Simple yet stable bearing-only navigation. Journal of Field Robotics 27 (5), 511–533.

Krajník, T., Vonásek, V., Fišer, D., Faigl, J., 2011. AR-Drone as a Platform for Robotic Research and Education. In: Research and Education in Robotics: EUROBOT 2011. Springer, Heidelberg.

Kumar, V., Michael, N., 2012. Opportunities and challenges with autonomous micro aerial vehicles. International Journal of Robotic Research 31 (11), 1279–1291.

Kushleyev, A., Mellinger, D., Kumar, V., 2012. Towards a swarm of agile micro quadrotors. In: Robotics: Science and Systems.

Langer, D., Rosenblatt, J., Hebert, M., December 1994. A behavior-based system for off-road navigation. IEEE Transactions on Robotics and Automation 10 (6), 776–783.

Lawton, J., Beard, R., Young, B., December 2003. A decentralized approach to formation maneuvers. IEEE Transactions on Robotics and Automation 19 (6), 933–941.

Liu, C., Chen, W.-H., Andrews, J., 2011. Piecewise constant model predictive control for autonomous helicopters. Robotics and Autonomous Systems 59 (78), 571 – 579.

Liu, Y., Jia, Y., 2012. An iterative learning approach to formation control of multi-agent systems. Systems & Control Letters 61 (1), 148 – 154.

Mastellone, S., Stipanovic, D. M., Graunke, C. R., Intlekofer, K. A., Spong, M. W., 2008. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. International Journal of Robotic Research 27 (1), 107–126.

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

Mellinger, D., Michael, N., Kumar, V., 2012. Trajectory generation and control for precise aggressive manuevers with quadrotors. International Journal of Robotics Research 31 (5), 664–674.

Michael, N., Kumar, V., 2009. Planning and control of ensembles of robots with non-holonomic constraints. International Journal of Robotics Research 28 (8), 962–975.

Min, H. J., Papanikolopoulos, N., 2012. Robot formations using a single camera and entropy-based segmentation. Journal of Intelligent and Robotic Systems (1), 1–21.

No, T. S., Kim, Y., Tahk, M.-J., Jeon, G.-E., 2011. Cascade-type guidance law design for multiple-uav formation keeping. Aerospace Science and Technology 15 (6), 431 – 439.

Nocedal, J., Wright, S. J., 2006. Numerical Optimization. Springer.

Olfati-saber, R., 2006. Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Transactions on Automatic Control 51, 401–420.

Ren, W., 2008. Decentralization of virtual structures in formation control of multiple vehicle systems via consensus strategies. European Journal of Control 14, 93–103.

Saffarian, M., Fahimi, F., 2009. Non-iterative nonlinear model predictive approach applied to the control of helicopters group formation. Robotics and Autonomous Systems 57 (67), 749 – 757.

Saska, M., 2013. Movies of simulations and experiments of the formation driving approach. http://imr.felk.cvut.cz/formationsijrr/ [cit. 2013-9-19].

Saska, M., Krajník, T., Faigl, J., Vonásek, V., Přeučil, L., 2012a. Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Saska, M., Krajník, T., Přeučil, L., 2012b. Cooperative micro uav-ugv autonomous indoor surveillance. In: IEEE Conference on Systems, Signals and Devices (SSD).

Saska, M., Vonásek, V., Krajník, T., Přeučil, L., 2012c. Coordination and navigation of heterogeneous uavs-ugvs teams localized by a hawk-eye approach. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Saska, M., Vonásek, V., Přeučil, L., May 2011. Roads sweeping by unmanned multi-vehicle formations. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA).

Shin, J., Kim, H., sept. 2009. Nonlinear model predictive formation flight. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39 (5), 1116 –1125.

Sira-Ramiandrez, H., Castro-Linares, R., 2010. Trajectory tracking for nonholonomic cars: A linear approach to controlled leader-follower formation. In: IEEE Conf. on Decision and Control (CDC).

Stipanović, D. M., Hokayem, P. F., Spong, M. W., Šiljak, D. D., 2007. Cooperative avoidance control for multi-agent systems. Journal of Dynamic Systems, Measurement, and Control 129, 699–707.

Tanner, H., Christodoulakis, D., 2007. Decentralized cooperative control of heterogeneous vehicle groups. Robotics and Autonomous Systems 55 (11), 811 – 823.

Turpin, M., Michael, N., Kumar, V., 2011. Trajectory design and control for aggressive formation flight with quadrotors. Autonomous Robots 33 (1-2), 143–156.

Xiao, F., Wang, L., Chen, J., Gao, Y., 2009. Finite-time formation control for multi-agent systems. Automatica 45 (11), 2605 – 2611.

Yang, H., ZHU, X., Zhang, S., 2010. Consensus of second-order delayed multi-agent systems with leader-following. European Journal of Control 16, 188–199.

Zhang, X., Duan, H., Yu, Y., 2010. Receding horizon control for multi-uavs close formation control based on differential evolution. Science China Information Sciences 53, 223–235.