Collision-Free Rendezvous Maneuvers for Formations of Unmanned Aerial Vehicles

A. Papen * **R. Vandenhoeck** * **J. Bolting** ** **F. Defay** **

* *Université de Toulouse, ISAE-SUPAERO, Toulouse, France (alexander.papen@student.kuleuven.be, ray.vandenhoeck@student.kuleuven.be)*
** *Université de Toulouse, ISAE-RESEARCH, Toulouse, France (jan.bolting@isae.fr, francois.defay@isae.fr)*

Abstract: This article discusses the rendezvous maneuver for a fleet of small fixed-wing Unmanned Aerial Vehicles (UAVs). Trajectories have to be generated on-line while avoiding collision with static and dynamic obstacles and minimizing rendezvous time. An approach based on Model Predictive Control (MPC) is investigated which assures that the dynamic constraints of the UAVs are satisfied at every time step. By introducing binary variables, a Mixed Integer Linear Programming (MILP) problem is formulated. Computation time is limited by incorporating the receding horizon technique. A shorter planning horizon strongly reduces computation time, but delays detection of obstacles which can lead to an infeasible path. The result is a robust path planning algorithm that satisfies the imposed constraints. However, further relaxation of the constraints and fine-tuning is necessary to limit complexity.

Keywords: Path planning, Optimal trajectory, Obstacle Avoidance, Mixed integer linear program, Unmanned aerial vehicle, Model predictive control, Collision avoidance, Receding horizon

1. INTRODUCTION

With the development of cost-competitive small fixed-wing unmanned aerial vehicles, interest in the control, trajectory generation, and formation flying of such vehicles has greatly increased. Many applications have been studied, such as surface-to-air missile jamming, monitoring of farms and irrigation control, and forest fire monitoring. (Chao et al., 2008) However, reliable real-time control of fleets of UAVs still remains difficult to achieve because of varying wind conditions (Wu et al., 2011) and limited communication range between UAVs (Beard and McLain, 2003). Furthermore, on-line calculation of the global optimal trajectory in terms of mission time or fuel consumption is often not feasible due to the required computation time. Therefore, a number of approximate methods have been developed such as a two-phase strategy and the receding horizon approach (Kamal et al., 2005; Richards et al., 2002; Bemporad and Rocchi, 2011; Hwangbo et al., 2007). These techniques do not calculate the globally optimal trajectory, but allow for a significant reduction in computation time.

One of the most crucial parts of formation flying is the rendezvous maneuver performed in order to achieve a specific formation. The principal objective of this study is to implement a path planning algorithm which minimizes the time required for this maneuver while satisfying an elaborate set of constraints. Collisions with other UAVs and static or dynamic obstacles have to be avoided at all times. The algorithm is based on Model Predictive Control (MPC) and reduces the optimization problem to a Mixed Integer Linear Program (MILP). A MILP approach can be applied for optimization problems of which part of the variables in the cost function and/or constraints are integers. Commercial packages, such as AMPL/CPLEX, are often used to solve this type of problem. In this paper, a receding horizon strategy (Kamal et al., 2005; Schouwenaars et al., 2001) is incorporated in the MILP framework and solved by means of the `intlinprog` command in Matlab.

Firstly, section 2 gives a precise problem statement. Secondly, an overview of the applied methodology is given in section 3. This section also elaborates on the cost function and the constraints of the MILP problem to be solved. Thirdly, the performance of the controlling algorithm is evaluated in section 4 for a benchmarking scenario. Special attention is devoted to the influence of different parameters on the computation time. Finally, possible improvements and extensions to the algorithm are discussed in section 5.

2. PROBLEM STATEMENT

A trajectory planning algorithm for a fleet of small fixed-wing UAVs was developed. The main goal of the controller is to minimize the time required for the rendezvous maneuver while avoiding collisions with static and dynamic obstacles. The considered maneuver consists of the rendezvous between a number of follower UAVs and a leader which follows a trajectory that is independent of the followers. A formation should be obtained in which all followers have a parent to which they maintain a fixed displacement vector expressed in the local reference frame of the leader, while avoiding the hazardous zones within

the wake vortices of other UAVs. However, maintaining the formation after a successful rendezvous does not lie within the scope of this project and can be achieved with a separate controller.

The trajectory generator is non-cooperative and information exchange between UAVs is limited to the positions and velocities of each UAV. The controller determines the position and velocity of the UAV at each time step. A separate controller, that is not considered in this paper, determines the commands that are applied to the UAV in order to follow this trajectory. Furthermore, computation time should be limited. It is assumed that wind conditions remain uniform. The algorithm was developed using Matlab/Simulink and was tested by means of an existing Multi-UAV simulation environment.

3. METHODOLOGY

This section discusses how the path planning problem and its constraints are formulated as a Mixed Integer Linear Program (MILP). Firstly, the main cost function is defined in order to minimize the rendezvous time. A zero-order hold discrete system is proposed for the dynamic model of the UAV. Furthermore, maximum velocity and acceleration, as well as minimum velocity, are imposed as constraints. Subsequently, avoidance of static and dynamic obstacles is implemented. Finally, the cost function is adapted to suit a receding horizon approach.

3.1 Cost function

The objective of the trajectory generator is to minimize the duration of the rendezvous maneuver while avoiding collisions. Consequently, the optimization problem can be written as follows:

$$\min_{\mathbf{s}_k} J = \min_{\mathbf{s}_k} N_G. \tag{1}$$

Where J is the cost function and $\mathbf{s}_k = (x\ y\ z\ v_x\ v_y\ v_z)_k^T$ the position- and velocity-vector of the UAV in the global reference frame at the k-th time step. The UAV reaches its goal position at the N_G-th time step, therefore N_G should be minimized. In order to determine N_G as a function of the optimization variables $\mathbf{s}_k$ while imposing that the UAV reaches its goal position with its goal velocity, a constraint is added to the optimization problem:

$$\min_{\mathbf{s}_k} N_G, \tag{2}$$

subject to

$$\exists! N_G \in \{1, ..., N_T\}$$
$$\mathbf{s}_{N_G} - \mathbf{s}_G = \mathbf{0}.$$

Where $\mathbf{s}_G = (x_G\ y_G\ z_G\ v_{x,G}\ v_{y,G}\ v_{z,G})^T$ is a vector with the goal position and goal velocity, and N_T the total number of considered time steps. The goal position is equal to the sum of the current position of the leader and a displacement vector. This vector is defined as a fixed vector in the local reference frame of the leader. In order to determine $\mathbf{s}_G$, this vector is expressed in the global reference frame using conventional transformation matrices based on the current attitude of the leader. This problem can be rewritten in a mixed integer linear form by adding a binary variable g_k for each time step. These variables denote whether the goal position is reached at the corresponding time step ($g_k = 1$). This method is similar to the one used by Richards et al. (2002) to detect when a UAV passes by a waypoint. This leads to the following optimization problem:

$$\min_{\mathbf{s}_k, g_k} N_G = \min_{\mathbf{s}_k, g_k} \sum_{k=1}^{N_T} k g_k, \tag{3}$$

subject to

$$\sum_{k=1}^{N_T} g_k = 1$$
$$\begin{aligned} \forall k \in \{1, ..., N_T\} \ : \ \mathbf{s}_k - \mathbf{s}_G &\leq M(1-g_k)\mathbf{1} \\ \mathbf{s}_G - \mathbf{s}_k &\leq M(1-g_k)\mathbf{1} \\ g_k &= 0 \text{ or } 1. \end{aligned}$$

Where $\mathbf{1}$ is a 6x1-vector of ones. M is an arbitrary number that is larger than the possible difference in position or velocity between the UAV and its goal. Therefore, if $g_k = 0$, the constraints of the k-th time step are satisfied even if the UAV has not yet reached its goal position or has passed it. If $g_k = 1$, the UAV reaches its goal at the k-th time step. By imposing that the sum of g_k equals one, the UAV reaches its goal at one time step only.

3.2 Model Predictive Control

The UAV is considered as a point mass with position and velocity $(x\ y\ z\ v_x\ v_y\ v_z)^T$. The control input is its acceleration: $(u_x\ u_y\ u_z)^T$. The equivalent zero-order hold discrete system is the following:

$$\begin{aligned} \begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix}_{k+1} &= \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix}_k + \\ &\quad \begin{pmatrix} \Delta t^2/2 & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & 0 & \Delta t^2/2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}_k \\ \Leftrightarrow \quad \mathbf{s}_{k+1} &= A\mathbf{s}_k + B\mathbf{u}_k. \end{aligned} \tag{4}$$

Where Δt denotes the length of each time step. This approach is also used by Kamal et al. (2005). Its limited complexity allows for a decrease in computation time while generating feasible trajectories. The exact model of the UAV is encapsulated by the assumed inner loop controllers that track the demanded accelerations $\mathbf{u}_k$, see e.g. (Bolting et al., 2016).

For each considered time step, the dynamic model of the UAV should be satisfied. Therefore, the above state space equations are added as constraints to the optimization problem.

3.3 Velocity and acceleration constraints

It is assumed that the maximum velocity and acceleration of each UAV are norm-constrained. Furthermore, a minimum velocity should be imposed for fixed-wing UAVs in

order to avoid stalling. These constraints are added to the optimization problem and can be written as follows:

$$\begin{aligned} \forall k \in \{1, ..., N_T\} \ : \ ||\mathbf{v}_k|| &\leq v_{max} \\ ||\mathbf{v}_k|| &\geq v_{min} \\ ||\mathbf{u}_k|| &\leq u_{max}. \end{aligned} \tag{5}$$

Where $\mathbf{v}_k$ and $\mathbf{u}_k$ are respectively the velocity and acceleration vector at the k-th time step. In order to preserve the linearity of the optimization problem, these constraints are linearized as explained in the following sections.

Maximum acceleration In order to linearize the constraint of maximum acceleration, u_{max} is imposed on the projection of $\mathbf{u}$ to different directions in 3D space, using a method similar to those used by Kamal et al. (2005); Culligan et al. (2007). The vector on which $\mathbf{u}$ is projected, is determined by two angles: θ in the xy-plane and φ in the perpendicular plane. These angles are discretized in order to determine all directions on which $\mathbf{u}$ is projected:
$\forall k \in \{1, ...N_T\} : \forall i \in \{1, ..., N_{d1}\} : \forall j \in \{1, ..., N_{d2}\}$:

$$\begin{aligned} \cos(\theta_i)\cos(\varphi_j)\, u_x + \sin(\theta_i)\cos(\varphi_j)\, u_y + \\ \sin(\varphi_j)\, u_z \leq u_{max}. \end{aligned} \tag{6}$$

Where θ is discretized in N_{d1} angles between 0 and 2π and φ in N_{d2} angles between $-\pi/2$ and $\pi/2$. The higher the discretization order, the more constraints are imposed, which in turn increases the computation time. However, if the discretization order is too small, some direction in 3D space will be favored by the path planning algorithm, as the maximum allowed acceleration in those directions is larger. Figure 1 shows the 3-dimensional boundary of the acceleration vector for $N_{d1} = 8$ and $N_{d2} = 5$. Ideally, the vector is bound by a sphere, but by discretizing θ and φ, the sphere is approximated. In the vertices of the approximated sphere, the acceleration constraint is maximum:

$$u_{vertex} = u_{max}\sqrt{\sec^2\left(\frac{\pi}{N_{d1}}\right) + \tan^2\left(\frac{\pi}{2(N_{d2}-1)}\right)}. \tag{7}$$

In order to ensure that the norm of $\mathbf{u}$ is always smaller than the maximum acceleration of the UAV, u_{vertex} should be equal to this value. The corresponding u_{max} can then be determined by evaluating the above equation. As illustrated in section 4.2, this method proves to be effective in imposing the maximum acceleration.

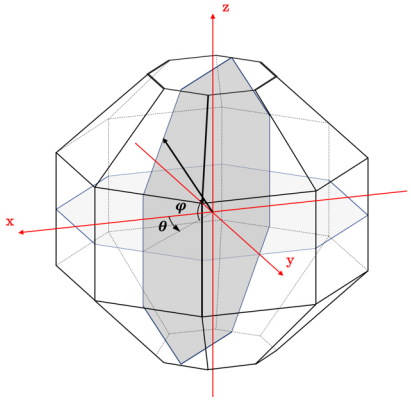


Fig. 1. Bounding polyhedron of the acceleration vector in 3D for $(N_{d1} = 8, N_{d2} = 5)$

Maximum velocity The maximum velocity constraint is integrated in a similar manner as for the maximum acceleration. However, slack variables are added, which relaxes the hard constraints to soft constraints. The slack variables represent by how much the maximum velocity is surpassed. They are added to the cost function with a large weight in order to minimize this violation. The trajectory generator has a tendency to keep the velocity close to v_{max} as this minimizes rendezvous time. Due to uncertainty and the use of an approximate dynamic model, it is possible that the real speed will be slightly larger than v_{max}. When the trajectory is recalculated at such a point, the hard maximum velocity constraint would be violated. Consequently, the algorithm would not be able to find a feasible path. Therefore, the introduction of slack variables increases its robustness.

The resulting constraints and auxiliary cost function J_{vmax} that is added to the main cost function of the optimization problem, are the following:

$$J_{vmax} = W_{vmax} \sum_{k=1}^{N_T} n_{vmax,k}, \tag{8}$$

subject to $\forall k \in \{1, ...N_T\} : \forall i \in \{1, ..., N_{d1}\} : \forall j \in \{1, ..., N_{d2}\}$:

$$\begin{aligned} \cos(\theta_i)\cos(\varphi_j)\, v_x + \sin(\theta_i)\cos(\varphi_j)\, v_y + \sin(\varphi_j)\, v_z \\ \leq v_{max}(1 + n_{vmax,k}) \\ 0 \leq n_{vmax,k} \leq 1. \end{aligned}$$

Where W_{vmax} is a large weight. The larger W_{vmax}, the more important the minimization of the maximum velocity slack variables is in comparison to the minimization of the rendezvous time.

Minimum velocity The same approach is applied in order to impose a minimum velocity. However, only one of the projections has to be larger than v_{min} as opposed to the maximum velocity, where every projection has to be smaller than v_{max}. Therefore, only one projection of $\mathbf{v}$ should be greater than v_{min} at a given time step. This is achieved by adding binary decision variables $b_{vmin,ijk}$:

$$J_{vmin} = W_{vmin} \sum_{k=1}^{N_T} n_{vmin,k}, \tag{9}$$

subject to $\forall k \in \{1, ...N_T\} : \forall i \in \{1, ..., N_{d1}\} : \forall j \in \{1, ..., N_{d2}\}$:

$$\begin{aligned} \cos(\theta_i)\cos(\varphi_j)\, v_x + \sin(\theta_i)\cos(\varphi_j)\, v_y + \sin(\varphi_j)\, v_z \\ \geq v_{min}(1 - n_{vmax,k}) - M(1 - b_{vmin,ijk}) \\ \sum_{i=1}^{N_{d1}} \sum_{j=1}^{N_{d2}} b_{vmin,ijk} \geq 1 \\ 0 \leq n_{vmin,k} \leq 1 \\ b_{vmin,ijk} = 0 \text{ or } 1. \end{aligned}$$

Where M is an arbitrary number greater than v_{min}. The binary variable $b_{vmin,ijk}$ is equal to one when the constraint is active. As this needs to be the case for at least one projection at every time step k, the sum of $b_{vmin,ijk}$ should be greater than or equal to one.

3.4 Static obstacles

Static obstacles are defined by the coordinates of their lower left and upper right corner points $(x_{min}\ y_{min}\ z_{min})^T$ and $(x_{max}\ y_{max}\ z_{max})^T$. At each time step, the UAV position has to be outside of the prism defined by these corner points in order to assure collision avoidance. As a consequence, at least one of the following constraints has to be satisfied at time step k and for the i-th static obstacle:

$$\begin{array}{lll} x_k \leq x_{min,i}\ , & y_k \leq y_{min,i}\ , & z_k \leq z_{min,i}\ , \\ x_k \geq x_{max,i}\ , & y_k \geq y_{max,i}\ , & z_k \geq z_{max,i}\ . \end{array}$$

The *or*-constraints are transformed into *and*-constraints by introducing binary variables $b_{obs,ikj}$. These variables denote whether the corresponding constraint is satisfied. Hence, at least one $b_{obs,ikj}$ must be equal to one. This is the same technique as the one applied in section 3.3.3. In addition, slack variables are added in order to avoid infeasibility of the trajectory. Since the trajectory will have a tendency to come as close as possible to obstacles in order to achieve an optimal path, it is possible that no feasible trajectory is found if the UAV is too close to an obstacle when the trajectory is recalculated. Relaxation of the collision constraints thus increases the robustness of the algorithm. Furthermore, a margin is added to the static obstacles. This way, a small violation of the collision constraint does not lead to a collision. By adding an auxiliary cost function, the slack variables are minimized. The final auxiliary cost function and constraints are the following:

$$J_{obs} = W_{obs} \sum_{i=1}^{N_o} \sum_{k=1}^{N_T} n_{obs,ik}, \tag{10}$$

$$\begin{aligned} &\text{subject to } \forall k \in \{1,...,N_T\}\ :\ \forall i \in \{1,...,N_o\}\ : \\ &\qquad x_k - x_{min,i} \leq M b_{obs,ik1} + D n_{obs,ik} \\ &\qquad x_{max,i} - x_k \leq M b_{obs,ik2} + D n_{obs,ik} \\ &\qquad y_k - y_{min,i} \leq M b_{obs,ik3} + D n_{obs,ik} \\ &\qquad y_{max,i} - y_k \leq M b_{obs,ik4} + D n_{obs,ik} \\ &\qquad z_k - z_{min,i} \leq M b_{obs,ik5} + D n_{obs,ik} \\ &\qquad z_{max,i} - z_k \leq M b_{obs,ik6} + D n_{obs,ik} \\ &\sum_{j=1}^{6} b_{obs,ikj} \leq 5 \\ &b_{obs,ikj} = 0 \text{ or } 1\ \forall j \in \{1,...,6\} \\ &0 \leq n_{obs,ik} \leq 1 \end{aligned}$$

Where N_o is the number of static obstacles and D the maximum amount by which the constraint may be violated. D is chosen equal to the margin added to the static obstacle, hence, collision can never occur. The binary variable $b_{obs,ikj}$ is equal to zero when the constraint is active which needs to be the case for at least one value of j.

3.5 Dynamic obstacles

During the rendezvous maneuver, collision between UAVs needs to be avoided at all times. The main difference with the static type is that dynamic obstacles move through space. Since the path planner is non-cooperative, a given UAV does not know the trajectories of the other UAVs. Therefore, it must determine its own path independently of the controllers of other UAVs.

When the trajectory is calculated, only the current position and velocity of the dynamic obstacles are known. The assumption is made that the velocities remain constant until recalculation. This is a good approximation for a short period of time. Therefore, the trajectory should be recalculated frequently in order to take into account the changing velocity vectors of the dynamic obstacles.

Dynamic obstacle constraints are implemented in the same way as for static obstacles in section 3.4 with the exception that the position of the obstacle is adjusted at each time step k. In this manner, the velocity of the obstacle is taken into account. The auxiliary cost function and constraints added to the mixed integer linear program are the following:

$$J_{do} = W_{do} \sum_{i=1}^{N_{do}} \sum_{k=1}^{N_T} n_{do,ik}, \tag{11}$$

$$\begin{aligned} &\text{subject to} \\ &\forall k \in [1,...,N_T]\ :\ \forall i \in [1,...,N_{do}]\ : \\ &\qquad \begin{array}{rll} x_k - x_{min,i} & -k v_{x,i} \Delta t \leq & M b_{do,ik1} + D n_{do,ik} \\ x_{max,i} - x_k & +k v_{x,i} \Delta t \leq & M b_{do,ik2} + D n_{do,ik} \\ y_k - y_{min,i} & -k v_{y,i} \Delta t \leq & M b_{do,ik3} + D n_{do,ik} \\ y_{max,i} - y_k & +k v_{y,i} \Delta t \leq & M b_{do,ik4} + D n_{do,ik} \\ z_k - z_{min,i} & -k v_{z,i} \Delta t \leq & M b_{do,ik5} + D n_{do,ik} \\ z_{max,i} - z_k & +k v_{z,i} \Delta t \leq & M b_{do,ik6} + D n_{do,ik} \end{array} \\ &\sum_{j=1}^{6} b_{do,ikj} \leq 5 \\ &b_{do,ikj} = 0 \text{ or } 1\ \forall j \in \{1,...,6\} \\ &0 \leq n_{do,ik} \leq 1. \end{aligned}$$

Where N_{do} is the number of dynamic obstacles, Δt the time between each time step, and $(v_{x,i}\ v_{y,i}\ v_{z,i})^T$ the velocity of the i-th dynamic obstacle.

3.6 Wake vortex avoidance

Behind every UAV, a wake vortex of turbulent air is formed. In these zones, it is difficult to maintain control over the UAVs which increases the risk of crashing. Therefore, wake vortices are treated as dynamic obstacles. The position of a UAV's wake vortex is estimated based on the orientation of its current velocity vector. The obstacle that represents the UAV is thus extended in the opposite direction of its velocity vector in order to represent the wake vortex. Since the orientation of the wake vortex is important, it cannot be modelled by a box whose faces are parallel to the xy-, xz-, and yz-plane. Therefore, the constraints presented in section 3.5 are modified in order to take into account the orientation of the dynamic obstacle.

The orientation of the obstacle is defined by the spherical coordinates θ and φ where $0 \leq \theta \leq 2\pi$ and $-\pi/2 \leq \varphi \leq \pi/2$. The current position of its center is given by $(x_c\ y_c\ z_c)^T$. The equations of the six faces of the obstacle can be written as follows:

$$\begin{aligned} c_\theta c_\varphi (x_k - x_c) + s_\theta c_\varphi (y_k - y_c) + s_\varphi (z_k - z_c) &= d_x \\ c_\theta c_\varphi (x_k - x_c) + s_\theta c_\varphi (y_k - y_c) + s_\varphi (z_k - z_c) &= -d_x - d_v \\ -s_\theta (x_k - x_c) + c_\theta (y_k - y_c) &= d_y \\ -s_\theta (x_k - x_c) + c_\theta (y_k - y_c) &= -d_y \\ -c_\theta s_\varphi (x_k - x_c) - s_\theta s_\varphi (y_k - y_c) + c_\varphi (z_k - z_c) &= d_z \\ -c_\theta s_\varphi (x_k - x_c) - s_\theta s_\varphi (y_k - y_c) + c_\varphi (z_k - z_c) &= -d_z. \end{aligned} \tag{12}$$

Where c_α is shorthand notation for $\cos\alpha$ and s_α for $\sin\alpha$, d_v is the length of the wake vortex and d_x is the

distance between the center of the obstacle and the face perpendicular to the x-axis. The distances d_y and d_z are defined in the same way.

Using the equations for the faces of the obstacle, the collision avoidance constraints can be written in a mixed integer linear form in a similar manner as presented in section 3.5:

$$J_{do} = W_{do} \sum_{i=1}^{N_{do}} \sum_{k=1}^{N_T} n_{do,ik}, \tag{13}$$

subject to

$\forall k \in [1, ..., N_T] \ : \ \forall i \in [1, ..., N_{do}] \ :$

$$\begin{aligned}
c_\theta c_\varphi(x_k - x_{c,i}) &+ s_\theta c_\varphi(y_k - y_{c,i}) + s_\varphi(z_k - z_{c,i}) \\
&\geq d_{x,i} - Mb_{do,ik1} - Dn_{do,ik} \\
c_\theta c_\varphi(x_k - x_{c,i}) &+ s_\theta c_\varphi(y_k - y_{c,i}) + s_\varphi(z_k - z_{c,i}) \\
&\leq -d_{x,i} - d_{v,i} + Mb_{do,ik2} + Dn_{do,ik} \\
-s_\theta(x_k - x_{c,i}) &+ c_\theta(y_k - y_{c,i}) \\
&\geq d_{y,i} - Mb_{do,ik3} - Dn_{do,ik} \\
-s_\theta(x_k - x_{c,i}) &+ c_\theta(y_k - y_{c,i}) \\
&\leq -d_{y,i} + Mb_{do,ik4} + Dn_{do,ik} \\
-c_\theta s_\varphi(x_k - x_{c,i}) &- s_\theta s_\varphi(y_k - y_{c,i}) + c_\varphi(z_k - z_{c,i}) \\
&\geq d_{z,i} - Mb_{do,ik5} - Dn_{do,ik} \\
-c_\theta s_\varphi(x_k - x_{c,i}) &- s_\theta s_\varphi(y_k - y_{c,i}) + c_\varphi(z_k - z_{c,i}) \\
&\leq -d_{z,i} + Mb_{do,ik6} + Dn_{do,ik} \\
\sum_{j=1}^{6} b_{do,ikj} &\leq 5 \\
b_{do,ikj} &= 0 \text{ or } 1 \ \forall j \in \{1, ..., 6\} \\
0 \leq n_{do,ik} &\leq 1.
\end{aligned}$$

with $x_{c,i} = x_i + kv_{x,i}\Delta t$ with x_i the x-position of the obstacle when the trajectory is calculated. The positions $y_{c,i}$ and $z_{c,i}$ are obtained in the same way. Furthermore, the following equations hold: $x_i = \frac{x_{min,i}+x_{max,i}}{2}$ and $d_{x,i} = \frac{x_{max,i}-x_{min,i}}{2}$, and analogously for y and z.

3.7 Receding horizon

In order to limit computation time, the trajectory is calculated up to a planning horizon instead of all the way up to the goal position. Because of the changing velocity of dynamic obstacles, the trajectory has to be recalculated frequently. As a consequence, it is not necessary to calculate up to the goal position. Of each trajectory, only the first N_a commands are applied to the UAV before a new calculation is done, this is called the execution horizon. Let T be the time between each recalculation, then:

$$N_a = \left\lfloor \frac{T}{\Delta t} \right\rfloor. \tag{14}$$

In a receding horizon approach, the number of time steps considered N_T is smaller than the number needed to reach the goal position. The distance between the final position and the goal position is minimized until the goal position lies within the planning horizon. The concepts of execution and planning horizon are illustrated in figure 2. As long as the goal position does not lie within the planning horizon, the cost function is the following:

$$J = (x_{N_T} - x_G)^2 + (y_{N_T} - y_G)^2 + (z_{N_T} - z_G)^2. \tag{15}$$

with $(x_{N_T} \ y_{N_T} \ z_{N_T})^T$ the final position at the planning horizon. However, this cost function is quadratic and needs

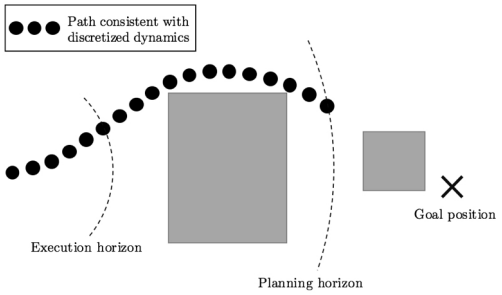


Fig. 2. A diagram of the execution and planning horizon

to be linearized in order to be applied to the MILP problem:

$$J = |x_{N_T} - x_G| + |y_{N_T} - y_G| + |z_{N_T} - z_G|. \tag{16}$$

The absolute value operators in the above cost function, can be replaced by adding following constraints, as is done by Kamal et al. (2005); Schouwenaars et al. (2001):

$$J = \sum_{i=1}^{3} w_i, \tag{17}$$

subject to

$$\begin{aligned}
x_{N_T} - x_G \leq w_1, \quad y_{N_T} - y_G \leq w_2, \quad z_{N_T} - z_G \leq w_3, \\
x_G - x_{N_T} \leq w_1, \quad y_G - y_{N_T} \leq w_2, \quad z_G - z_{N_T} \leq w_3.
\end{aligned}$$

Once the goal position lies within the planning horizon, the same cost function as for complete trajectory planning is used. There is no direct way to determine whether the goal position lies within the planning horizon without first calculating the trajectory. In order to reach the goal position when it lies within the planning horizon, an arbitrary large number W_{rh} is subtracted from the cost function if one of the binary variables g_k, that detect if the UAV reaches its goal, equals one. Furthermore, the minimization of the distance between the UAV and its goal at the end of the planning horizon is no longer important once the target lies within the planning horizon. Hence, a weight W_{N_G}, greater than any possible distance between the UAV and its goal, is given to $\sum_{k=1}^{N_T} kg_k$. This term is zero if the UAV can't reach its goal within the planning horizon and N_G otherwise:

$$J = \sum_{i=1}^{3} w_i + W_{N_G} \sum_{k=1}^{N_T} kg_k - W_{rh} \sum_{k=1}^{N_T} g_k, \tag{18}$$

subject to

$$\begin{aligned}
\sum_{k=1}^{N_T} g_k &\leq 1 \\
\forall k \in \{1, ..., N_T\} \ : \quad \mathbf{s}_k - \mathbf{s}_G &\leq M(1 - g_k)\mathbf{1} \\
\mathbf{s}_G - \mathbf{s}_k &\leq M(1 - g_k)\mathbf{1} \\
x_{N_T} - x_G &\leq w_1 \\
x_G - x_{N_T} &\leq w_1 \\
y_{N_T} - y_G &\leq w_2 \\
y_G - y_{N_T} &\leq w_2 \\
z_{N_T} - z_G &\leq w_3 \\
z_G - z_{N_T} &\leq w_3 \\
g_k &= 0 \text{ or } 1.
\end{aligned}$$

In this manner, if the goal position lies within the planning horizon, minimizing the time it takes to reach it is prioritized over minimizing the distance between the

final position of the UAV at the planning horizon and the goal position. If the goal position does not lie within the planning horizon, the term representing the time it takes to reach it reduces to zero and the distance between the position at the end of the planning horizon and the goal is minimized instead.

4. RESULTS

4.1 Benchmarking scenario

In order to test the performance of the developed algorithm and to investigate the influence of specific parameters, a benchmarking scenario was implemented. In this scenario, three UAVs are launched from the origin of the global reference frame $(0\ 0\ 0)^T$. The leader UAV has a fixed goal position at $(100\ 0\ -10)^T$. In its path lies a box-shaped static obstacle, for example a building, with corner points $(20\ -8\ 0)^T$ and $(40\ 8\ -30)^T$. After 2s, the second UAV is launched and will rendezvous with its leader, the first UAV. After another 2s, the third is launched which will also join the first UAV in formation. All UAVs have an initial velocity of $(10\ 0\ -1)^T$. A diagram of this scenario is presented in figure 3.

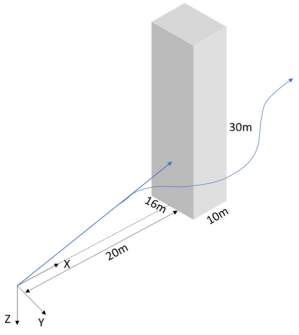


Fig. 3. A diagram of the benchmarking scenario with the UAV trajectory in blue

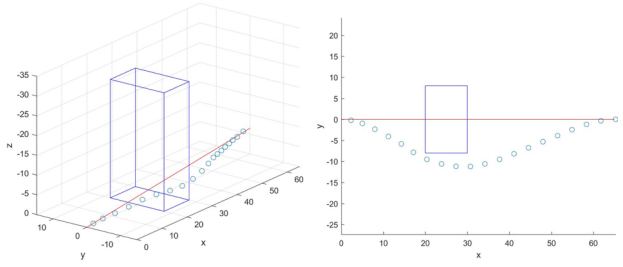
The UAVs have a maximum acceleration of $20\frac{m}{s^2}$, a minimum velocity of $10\frac{m}{s}$, and a maximum velocity of $20\frac{m}{s}$. For collision avoidance purposes, the UAVs are represented as cube-shaped obstacles with edges of 2m. The hazardous wake vortex has a length of three times the 2m wingspan.

4.2 Verification of trajectory constraints

The planning horizon is set to 4s in order to be able to plan a complete trajectory around the obstacle. The calculated trajectory is shown in figure 4. The norms of the velocity and acceleration vector during this trajectory are presented in figure 5. The velocity is always less than $20\frac{m}{s}$ and greater than $10\frac{m}{s}$. It is close to v_{max} during the majority of the flight path. When the UAV is launched, it accelerates up to its maximum velocity. After circling around the obstacle, its speed drops to that of its leader.

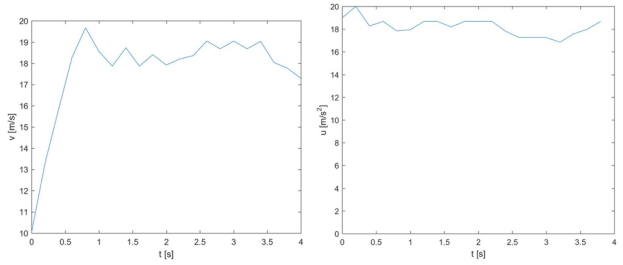
The acceleration is always smaller than the imposed $20\frac{m}{s^2}$. During the majority of the trajectory, the acceleration is close to its maximum, which indicates that the capacity of the UAV is used to its full potential. During the first time step, the speed of the UAV increases by $3.3\frac{m}{s}$ in 0.2s which corresponds to an acceleration of $16.5\frac{m}{s^2}$. Hence, in the beginning of the trajectory the orientation of $\mathbf{u}$ and $\mathbf{v}$ is nearly the same. Once $||\mathbf{v}||$ is close to v_{max}, the acceleration vector is nearly perpendicular to the velocity vector which allows the UAV to encircle the obstacle. This means that the acceleration capacity of the UAV is fully used to turn the velocity vector, hence $||\mathbf{v}||$ remains approximately constant while $||\mathbf{u}||$ is close to its imposed maximum.

Finally, the trajectory verifies the static obstacle avoidance constraints. A margin of 0.5m was added to the obstacle to allow for a small violation of the constraints caused by the slack variables. The smallest distance between the UAV and the obstacle during the trajectory is exactly equal to 0.5m. This means that all slack variables are zero at all time steps, which indicates that the UAV can still evade the obstacle while satisfying the collision constraints.



(a) 3D view (b) Top view

Fig. 4. Calculated trajectory, the red line represents the trajectory if the obstacle was not present



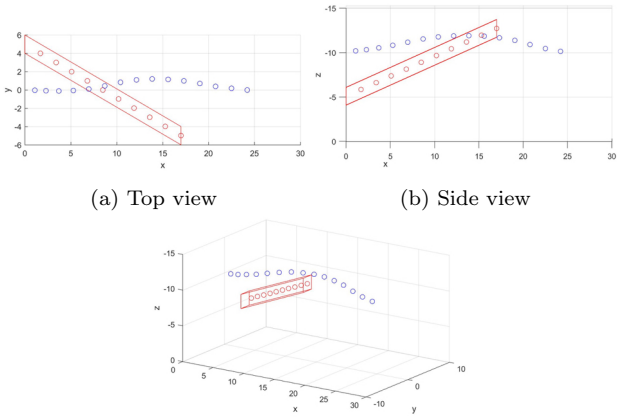
(a) Velocity $||\mathbf{v}||$ (m/s) (b) Acceleration $||\mathbf{u}||$ (m/s^2)

Fig. 5. Magnitude of velocity and acceleration

Dynamic obstacle In order to verify the dynamic obstacle constraints, the trajectory is determined for the following scenario: a UAV is launched from position $(0\ 0\ -10)^T$ with velocity $(10\ 0\ 0)^T$. Its goal trajectory is to fly at a constant velocity equal to its initial velocity. However, another UAV is launched at the same time from position $(0\ 5\ -5)^T$ and flies at a constant velocity of $(17\ -10\ -7.5)^T$. The first UAV will have to avoid collision with the second. The calculated trajectory is shown in blue in figure 6. The trajectory of the second UAV is shown in red. The first UAV is able to anticipate the movement of the second and avoid collision. When it crosses the second UAV, it flies over its wake vortex, represented by the red prism.

4.3 Minimum planning horizon for obstacle avoidance

The duration up to which the trajectory is calculated, is equal to $N_T \Delta t$. This time should be sufficiently large



(a) Top view (b) Side view

(c) 3D view

Fig. 6. Calculated trajectory of dynamic obstacle avoidance scenario

to detect obstacles fast enough. However, the planning horizon also increases the computation time, as illustrated in section 4.4.2, and should therefore be limited. Based on the dimensions of the obstacles, the maximum acceleration and velocity, an estimate can be made of the minimum planning horizon needed in order to ensure collision avoidance.

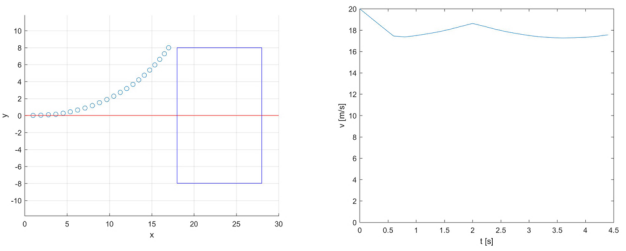
$N_T \Delta t$ should be large enough to allow a UAV, flying at its maximum velocity in the direction of an obstacle, to avoid that obstacle without having to decelerate. This would impact the optimality of the trajectory. When the UAV steers away from the obstacle, without decelerating, the minimum turning radius equals $r_{min} = \frac{v_{max}^2}{u_{max}}$. A diagram of this scenario is shown in figure 8. The time it takes for the UAV to fly the arc is equal to $t = \frac{r\theta}{v_m} = \frac{v_m \theta}{u_m}$. θ is the angle spanned by the arc and is equal to

$$\begin{aligned} \theta = \arccos\left(\frac{r-d}{r}\right) = \arccos\left(\frac{v_{max}^2/u_{max} - d}{v_{max}^2/u_{max}}\right) \\ \Rightarrow t = \frac{v_{max}}{u_{max}} \arccos\left(\frac{v_{max}^2/u_{max} - d}{v_{max}^2/u_{max}}\right). \end{aligned} \tag{19}$$

In this equation, d is half of the length of the obstacle in the direction in which it is easiest to evade. Using the above equations, the minimum planning horizon for the obstacle considered in the benchmarking scenario equals $t = 1.018s$. The trajectory for a scenario where the UAV is flying at maximum speed towards this obstacle is shown in figure 7a. The UAV has to decelerate slightly in order to avoid the obstacle as shown in figure 7b. It reaches the obstacle after 21 time steps of 0.05s, which is close to the calculated $t = 1.018s$. Furthermore, the execution horizon $N_a \Delta t$ must be smaller than the difference between the actual and minimum planning horizon. Otherwise, the UAV could move too close to an undetected obstacle before its trajectory is recalculated.

4.4 Sensitivity analysis

Influence of Δt and N_a on rendezvous time The rendezvous time for the benchmarking scenario is recorded for Δt ranging from 0.08s to 0.20s. At first, the slack variables from the static obstacle constraint are non-zero. The only possible way to reach the goal is by violating the static collision constraint. As Δt increases, the static obstacle will



(a) Trajectory (b) Velocity $||\mathbf{v}||$ (m/s)

Fig. 7. UAV flying at maximum speed towards an obstacle at the edge of the planning horizon

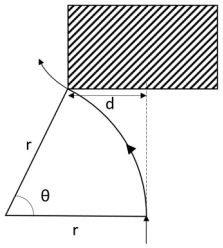
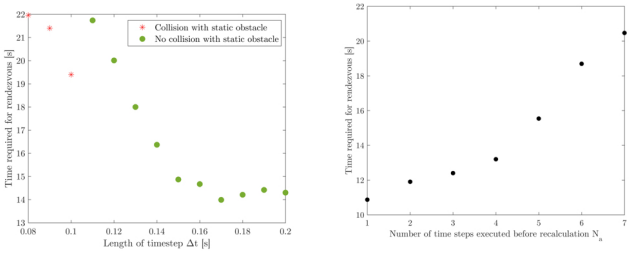


Fig. 8. A diagram of the UAV steering away from an obstacle at its minimum turning radius

lie within the planning horizon more quickly, allowing the controller to intervene faster. Therefore, the rendezvous time decreases as the UAV does not have to decelerate. The sudden jump between $\Delta t = 0.10s$ and $\Delta t = 0.11s$ indicates that the slack variables turn zero. The planning horizon is now sufficiently long to fully avoid the obstacle. Further increasing Δt decreases the rendezvous time as the obstacle is detected sooner. Therefore, a more optimal trajectory is calculated. At $\Delta t = 0.16s$, the rendezvous time begins to slightly increase due to a coarser discretization of the trajectory.

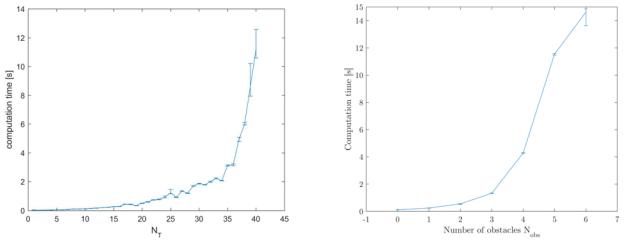


(a) Influence of Δt (b) Influence of N_a

Fig. 9. Influence of Δt and N_a on rendezvous time

Figure 9b shows that the rendezvous time increases linearly with the execution horizon N_a varying between 1 and 7. The execution horizon N_a corresponds to the amount of time steps executed before the trajectory is recalculated. The sooner recalculation occurs, the faster the optimal trajectory is updated which leads to a shorter rendezvous time. However, more computation time is necessary if the trajectory is recalculated more frequently. A trade-off needs to be made between path optimality and computation time depending on the resources available.

Influence of N_T and N_{obs} on the computation time This section discusses the influence of N_T and N_{obs} on the computation time of the `intlinprog` algorithm for one trajectory only. The algorithm is executed for the benchmarking scenario without the static obstacle. The computation times were measured with an Intel Core i7 processor of 2.6GHz and 16GB RAM. Figure 10a shows the computation time for an increasing amount of calculated time steps N_T. Each data point represents the average computation time of 10 simulations. The error bars extend from the minimum to the maximum measured computation time. Δt is held constant at 0.15s. As N_T increases, the computation time grows exponentially. This is due to the dependence of the amount of constraints on N_T: the total number of constraints in the MILP problem is equal to $75N_T + 7N_T N_{obs} + 12$ with N_{obs} the number of obstacles. However, the time complexity of the `intlinprog` command does not vary linearly with the amount of constraints. Figure 10b shows the influence of the number of static obstacles within the planning horizon on the computation time. For every obstacle added, the number of constraints of the MILP problem increases by $7N_T$. As a consequence, N_{obs} has an important influence on the computation time.



(a) Influence of N_T (b) Influence of N_{obs}

Fig. 10. Influence of N_T and N_{obs} on the computation time

5. POSSIBLE IMPROVEMENTS

Calculation of the optimal rendezvous trajectory in real-time has proven to be infeasible due to the elaborate set of constraints. In order to overcome this barrier, fine-tuning of the parameters and relaxation of constraints is necessary. For example, the planning and execution horizon, as well as Δt, can be varied on-line based on the current positions of obstacles and other UAVs. An optimal configuration of these parameters at each time step decreases computation time. Further improvements could be obtained by using specialized MILP software packages such as AMPL/CPLEX instead of the `intlinprog` command of Matlab. Furthermore, the GPS uncertainty on the position of the UAVs should be taken into account in order to make the algorithm more robust. Finally, coordination between the path planners could be considered if information exchange is not a limiting factor. This allows to compute globally optimal trajectories where other UAVs are not considered as obstacles with a constant velocity at each time step.

6. CONCLUSION

A path planning algorithm for the rendezvous maneuver of a fleet of UAVs has successfully been developed. The mathematical problem was formulated as a Mixed Integer Linear Program (MILP) by linearizing the constraints and transforming *or*- to *and*-constraints. The latter was achieved through the introduction of binary variables. The performance of the algorithm was investigated for a benchmarking scenario consisting of three UAVs and one static obstacle. A sensitivity analysis was conducted to investigate the influence of the algorithm's parameters on the rendezvous time and the computation time. The computation time strongly increases with the number of time steps N_T and the number of obstacles N_{obs}. Finally, further improvements are discussed to overcome the barrier of on-line computation.

REFERENCES

Beard, R.W. and McLain, T.W. (2003). Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, 25–30. IEEE.

Bemporad, A. and Rocchi, C. (2011). Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 7488–7493. IEEE.

Bolting, J., Fergani, S., Biannic, J.M., Defay, F., and Stolle, M. (2016). Discrete Sliding Mode control of small UAS in tight formation flight under information constraints. In *Proceedings of IFAC Symposium on Automatic Control in Aerospace (ACA) 2016*.

Chao, H., Baumann, M., Jensen, A., Chen, Y., Cao, Y., Ren, W., and McKee, M. (2008). Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control. *IFAC Proceedings Volumes*, 41(2), 11744–11749.

Culligan, K., Valenti, M., Kuwata, Y., and How, J.P. (2007). Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. In *American Control Conference, 2007. ACC'07*, 5322–5327. IEEE.

Hwangbo, M., Kuffner, J., and Kanade, T. (2007). Efficient two-phase 3d motion planning for small fixed-wing UAVs. In *Robotics and Automation, 2007 IEEE International Conference on*, 1035–1041. IEEE.

Kamal, W.A., Gu, D.W., and Postlethwaite, I. (2005). Real time trajectory planning for UAVs using MILP. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, 3381–3386. IEEE.

Richards, A., Bellingham, J., Tillerson, M., and How, J. (2002). Coordination and control of multiple UAVs. In *AIAA guidance, navigation, and control conference, Monterey, CA*.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. In *Control Conference (ECC), 2001 European*, 2603–2608. IEEE.

Wu, P.P., Campbell, D., and Merz, T. (2011). Multi-objective four-dimensional vehicle motion planning in large dynamic environments. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(3), 621–634.