

FUEL CONSUMPTION REDUCTION FOR UNMANNED AIR VEHICLES BY PREPLANNED FORMATION FLIGHTS USING MIXED INTEGER PROGRAMMING

T. Kopfstedt, J.W. Vervoort
Diehl BGT Defence GmbH & Co. KG
Alte Nussdorfer Strasse 13, 88662 Ueberlingen
Germany

ABSTRACT

This paper examines the optimization of the trajectories of formation flights for unmanned air vehicles (UAV). The optimization problem will be solved by a Mixed Integer Programming (MIP) – solver. The optimization problem itself is formulated as hybrid minimization problem with continuous and discrete optimization parameters. The dynamic UAV model, the description of the obstacle for different obstacles (forbidden flight areas, buildings) in 2D and 3D will be described as well as the description of the formation and the construction of the Mixed Integer Programming minimization problem. The effectiveness of this method will be shown for different scenarios in a simulation results chapter.

Index Terms: Mixed Integer Programming, cooperative control, optimal path generation

1. INTRODUCTION

In future the degree of autonomy of unmanned air vehicles (UAV) and their total number will increase significantly compared with today. Even if UAVs are used today mainly for military purposes they will be used more and more in civil markets as well like airfreight cargo flights and, depending on future laws, maybe also for passenger flights. The increasing number of UAVs will make it possible to create formations of UAVs over long distance flights. The creation and stability of formation flights itself has already been investigated for example in [4], [9], [16] and [17]. The advantage of formation flights is the significantly decreased consumption of fuel of the UAVs following the leading UAV, as it is investigated in [13] and [15]. Due to this possibility to save energy during each flight there will be a high interest in generating formations of UAVs, if there is an advantage for each UAV in the formation, so that all UAVs can reach a higher efficiency of their energy consumption during their flights and increase their payload capabilities because of a decrease in the fuel load.

In military missions UAVs are at the moment mainly used for intelligence missions. In such missions today only one UAV is fulfilling the mission on its own. This will change in future and the missions of UAVs will be more extensively integrated into the mission of normal planes as shown in [3]. Nowadays a fuel reduction due to formation flights is impossible during the intelligence mission as only a single UAV is patrolling in an operation sector, but such a mission contains more than only the intelligence part. At the beginning and at the end of the mission the UAVs for

different intelligence missions have to fly to their operation areas. If several operation fields are close together a number of UAVs can fly in formation for a certain part of the path from their base to their operation sectors in order to save fuel.

Another important point is the optimization of the trajectories between the different points of interest in a UAV intelligence mission. The algorithm that will be shown in this paper can also be used, as the calculated trajectories are the optimal paths between the start and goal points of a mission part, based on the parameters for the criteria of optimization.

Unlike [1] where the creation and changing of the formation is investigated using agents the solution in this paper optimizes the complete path of each vehicle and the creation of the formation and changes of formation become some subparts in the complete MIP minimization problem. The use of MIP for formation flights and flights of single vehicles is also investigated in [2], [7] and [14], but there the MIP is used for the receding horizon control in combination with a cost function to fulfil the missions, so that there the MIP solver must always solve only a part of the complete mission instead of the complete mission with all vehicles, all obstacles and for the complete path from start to goal positions in one optimization process.

At first we will describe the used dynamic UAV model for the Mixed Integer Programming optimization. Then the description for zones where a path through is prohibited is shown in chapter 3. In this chapter different versions of the method for the prohibited zones are shown. This is due to the fact that the prohibited zones are defined as obstacles and depending on the mission planning part the optimization can be done in 2D or 3D. With the information from chapter 2 and chapter 3 and a description of formation as such, that will be shown in chapter 4 the minimization problem itself can be constructed and solved by a MIP-solver. The results from solving the MIP minimization problem are shown in chapter 5, where the simulation results are presented.

2. DYNAMIC UAV MODEL

For the structure of the MIP minimization problem, which will be used for the generation of the optimal trajectories for the UAVs at first a description model of the behaviour of the UAVs is necessary.

For the planning of the trajectories of the UAVs it is possible to use a rather simple dynamic model to describe the capabilities of the UAVs. This is possible as the

control of each UAV is done in another subordinate control loop with a smaller loop time. According to that and in relation to other publications as [2], [8], [11] and [14] the following dynamic UAV model based on [11] is used for the description of the dynamics of the UAV

$$(1) \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_z} \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{k_x}{\tau_x} & 0 & 0 \\ 0 & \frac{k_y}{\tau_y} & 0 \\ 0 & 0 & \frac{k_z}{\tau_z} \end{bmatrix} \begin{bmatrix} f_x(t) \\ f_y(t) \\ f_z(t) \end{bmatrix}$$

This is a model that is representing a helicopter as well as a QuadRotor system, that can easily be adapted and modified to any other used UAV type, as only the decoupled and coupled axis, velocities and accelerations have to be modified depending on the used UAV type. The results are the velocities $\dot{x}(t)$, $\dot{y}(t)$, $\dot{z}(t)$ and the accelerations $\ddot{x}(t)$, $\ddot{y}(t)$, $\ddot{z}(t)$ of the UAV. The states are the actual position $x(t)$, $y(t)$, $z(t)$ of the UAV and as input the forces $f_x(t)$, $f_y(t)$, $f_z(t)$ from the engine of the UAV are used. In cases where the optimization is done only in 2D the equation (1) can be reduced to

$$(2) \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_x} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_y} \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{k_x}{\tau_x} & 0 \\ 0 & \frac{k_y}{\tau_y} \end{bmatrix} \begin{bmatrix} f_x(t) \\ f_y(t) \end{bmatrix}$$

which is the same as (1) except that the z - components are missing in (2).

3. DESCRIPTIONS OF OBSTACLES

Depending on the scenario and the kind of mission the obstacles can be described in different ways. In general the obstacles should be described as simple as possible and when 2D path planning is sufficient the path planning should not be done in 2D as the optimization of a 3D minimization problem lasts much longer than a 2D optimization problem. This is the similar problem with a high number of obstacles and/or a high level of detail of each obstacle. To allow a description of the obstacles as simple as possible in the following subchapters a set of possible descriptions of obstacles for different cases is shown. The use of the different descriptions is than shown in the chapter "Simulation Results".

3.1. Simple 2D Obstacles

The common way of describing obstacles for optimizations in publications using Mixed Integer Programming is the use of rectangles orientated along the coordinate system, as described for example in [2], [10] and [12] and for visualization shown in Fig. 1. These obstacles can be described by the system of equations

$$(3) \begin{aligned} x_{i,j} - \varepsilon_1 \cdot S &\leq x_{\min} - e \\ -x_{i,j} - \varepsilon_2 \cdot S &\leq -x_{\max} - e \\ y_{i,j} - \varepsilon_3 \cdot S &\leq y_{\min} - e \\ -y_{i,j} - \varepsilon_4 \cdot S &\leq -y_{\max} - e \end{aligned}$$

$$\sum_{i=1}^4 \varepsilon_i \leq 3$$

using $x_{i,j}$ and $y_{i,j}$ as the position of the vehicle j in the step t . With x_{\min} , y_{\min} the position of the lower left edge of the obstacle is described and with x_{\max} , y_{\max} the position of the upper right edge of the obstacle is described. The value e is the machine precision of the computer on which the optimization is running and ε_1 , ε_2 , ε_3 and ε_4 are Boolean variables in connection with the large positive number S .

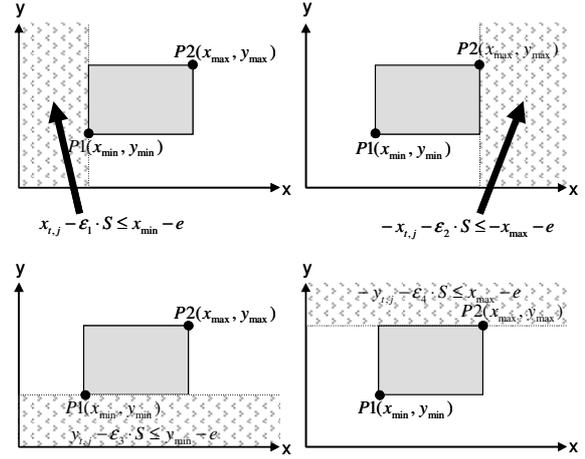


FIG 1. 2D Rectangle

The system of equations (3) works as shown in Fig. 1 and tests if the position $x_{i,j}$, $y_{i,j}$ of the vehicle is inside or outside of the obstacle. Therefore at least one of the four tests must succeed to fulfil the last equation. Based on this principle also more complex obstacles can be described as shown below.

3.2. Convex Obstacles

The method shown in chapter 3.1 is often not very useful for complex scenarios with narrow paths between several obstacles, as all obstacles can only be described by rectangles that are all orientated along the coordinate system.

Another method to describe obstacles in 2D is to describe them as polygons. The following algorithm is taken from [5] and allows to describe convex obstacles. An alternative method for the description of more complex obstacles is shown in [11].

For obstacles like the one shown in Fig. 2 the following description vector is necessary:

$$(4) \text{Obs}_i = [x_{start}, y_{start}, \alpha_1, \alpha_2, \dots, \alpha_{n-1}, l_1, l_2, \dots, l_{n-1}]$$

with x_{start} , y_{start} defining the point of suspension and α_1 , α_2, \dots for the angles between the following segments of the polygon and l_1, l_2, \dots for the length of the segments of the polygon.

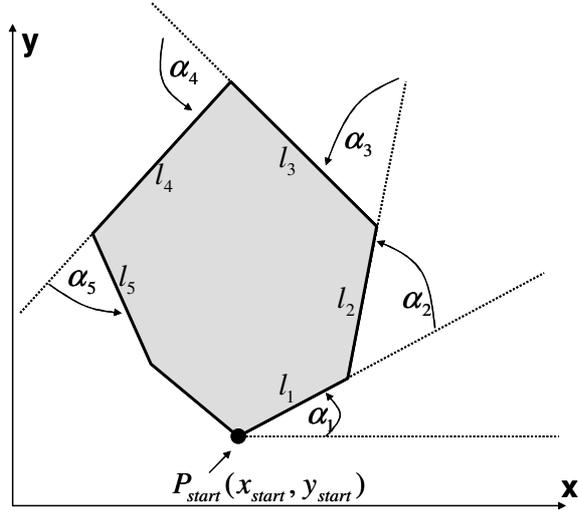


FIG. 2. Convex 2D Polygon

For the construction of the necessary equations to test if a position of the vehicle is inside or outside of the obstacle some calculations must be done a priori. At first the first edge of the polygon is defined as

$$(5) \quad P_1(x_1, z_1) = \begin{bmatrix} x_{start} \\ y_{start} \end{bmatrix}$$

$$\alpha_{obs_1} = 0$$

Then by using the information from (4) and from (5) the positions of all other edges are calculated in the global simulation coordinate system by

$$(6) \quad P_i(x_i, y_i) = \begin{bmatrix} x_{i-1} + l_{i-1} \cdot \cos(\alpha_{obs_i-1}) \\ y_{i-1} + l_{i-1} \cdot \sin(\alpha_{obs_i-1}) \end{bmatrix}$$

$$\alpha_{obs_i} = \alpha_{obs_i-1} + \alpha_{i-1}$$

When (6) is used to define all edges of the convex polygon the collision test can be described by

$$(7) \quad -x_{i,j} \cdot \text{sgn}(y_{i+1} - y_i) - \varepsilon_i \cdot S \leq -(x_i + e) \cdot \text{sgn}(y_{i+1} - z_i)$$

for all segments of the convex polygon that are orientated with $\frac{\pi}{2}$ or $\frac{3\pi}{2}$ towards the coordinate system. For all other cases the more complex version

$$(8) \quad y_{i,j} \cdot f_i - x_{i,j} \cdot f_i \cdot g_i - \varepsilon_i \cdot S \leq -f_i \cdot (e + h_i)$$

is necessary and needs (9), (10), and (11) to become solvable.

$$(9) \quad f_i = \text{sgn}(\cos(\alpha_{obs_i+1}))$$

$$(10) \quad g_i = \frac{z_{i+1} - z_i}{x_{i+1} - x_i}$$

$$(11) \quad h_i = g_i \cdot x_i - z_i$$

As the tests are working in similar to the demonstrated way in Fig. 1 the equation

$$(12) \quad \sum_{i=1}^n \varepsilon_i \leq n-1$$

is necessary to identify if at least one of the equations for the segments of the obstacles is fulfilled or not.

3.3. Extension of 2D Obstacle to 3D Obstacle

A path planning in 2D is not applicable for all missions. For the kind of missions where obstacles are not special zone in the air like flight permitted zones more complex descriptions are necessary. If obstacles in the optimizations represent real obstacles like mountains or high buildings not only flying past to the left or to the right sides is possible but also a flight over the top will be an alternative and depending on the obstacle maybe a shorter and more effective path. In this case the description of the 2D obstacle must be extended into 3D. We are therefore referring to [6] and are extending the description vector from (4) to the following form

$$(13) \quad Obs_i = [x_{start}, y_{start}, z_{start}, \alpha_1, \alpha_2, \dots, \alpha_{n-1}, l_1, l_2, \dots, l_{n-1}, z_{height}]$$

with z_{start} as the height of the base of the obstacle and z_{height} as the absolute height of the obstacle towards its base height. With this small adaptation the equations from chapter 3.2 can still be used and only two additional equations

$$(14) \quad z_{i,j} - \varepsilon_i \cdot S \leq z_{min} - e$$

$$(15) \quad -z_{i,j} - \varepsilon_i \cdot S \leq -z_{max} - e$$

must be added to the system of equations created by (7), (8) and (12).

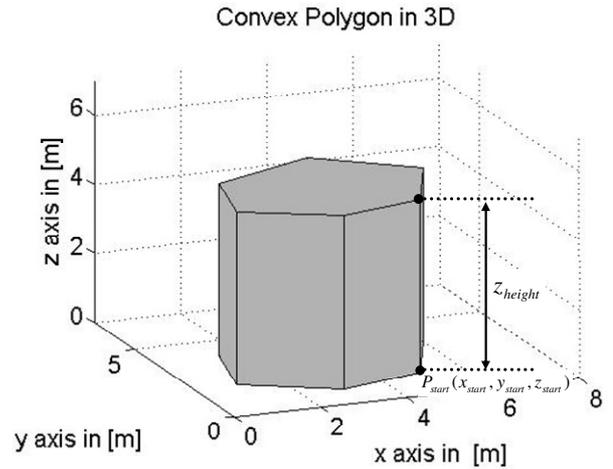


FIG. 3. Convex Polygon Extended to 3D

A possible obstacle that can be effectively described by this method is shown in Fig. 3. Another almost similar way for the description of such obstacles is shown in [11].

3.4. Virtual Obstacles

Depending on the possible step size for the trajectory points of the solution from the Mixed Integer Programming optimization the results may be as shown in Fig. 4. If illegal crossing of an obstacle or an edge of an obstacle happens it will be a big problem for the vehicle that is flying along this trajectory later, because a real vehicles has always a size different from an infinitesimal small point and the real vehicle is unable or not allowed to pass through an obstacle that was ignored by the optimization because of its small size.

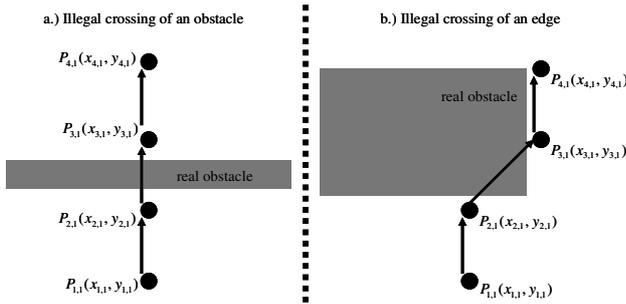


FIG 4. a) Illegal crossing of an obstacle
b) Illegal crossing of an edge

To avoid such situations the sizes of the real and defined obstacles must be increased. These increased obstacles that are larger than the real obstacles and where edge cutting would be no problem are described here as “virtual obstacles”. As shown in Fig. 5 the virtual edges can be constructed around the real obstacles.

The method is shown here for 2D cases but it can be used for 3D obstacles in the same way. For the calculation of the positions of the edges from the virtual obstacle some parameters from the vehicle and the optimization are necessary. The parameter with the largest influence normally is the possible step size between two points of the solution of the optimization. The maximum possible step size can be calculated by

$$(16) d_{step} = \sqrt{\Delta x_{max}^2 + \Delta y_{max}^2}$$

Also other parameters are important like the diameter $d_{vehicle}$ from the vehicle and the position accuracy d_{pos} to define the total distance necessary between the virtual and the real obstacles. The necessary distance can be calculated with

$$(17) d = \frac{d_{vehicle}}{2} + d_{pos} + \frac{d_{step}}{2}$$

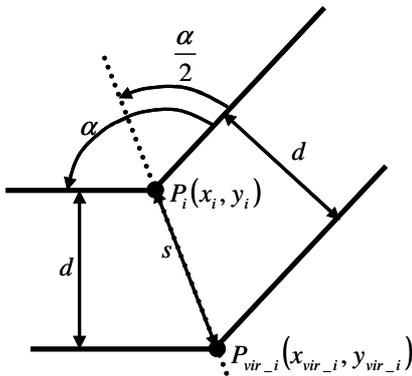


FIG 5. Construction of Virtual Edges

To apply the description of obstacles from chapter 3.2 to the virtual obstacles it is necessary to describe them in the same manner as the real obstacles. Therefore the information about the edges and angles of the real

obstacles can be used to define the positions of the edges of the virtual obstacles. These dependences are shown in Fig. 5 and the angle

$$(18) \alpha = \pi - (\alpha_{obs_i+1} - \alpha_{obs_i})$$

that uses the information from (6) is important to construct the direction for the new edge. The difference of position between the real edge and the virtual edge is defined as

$$(19) s = \frac{d}{\cos\left(\frac{\pi - \alpha}{2}\right)}$$

by using the equations (17) and (18). With this information it is possible to describe the virtual edges in the same manner as the real edges. Consequently for the general description of the obstacle (6) can be replaced by

$$(20) P_{vir_i} = \begin{bmatrix} x_{vir_i} \\ y_{vir_i} \end{bmatrix} = \begin{bmatrix} x_i + s \cdot \cos\left(\pi + \alpha_{obs_i+1} + \frac{\alpha}{2}\right) \\ y_i + s \cdot \sin\left(\pi + \alpha_{obs_i+1} + \frac{\alpha}{2}\right) \end{bmatrix}$$

so that all other equations can be used without changes.

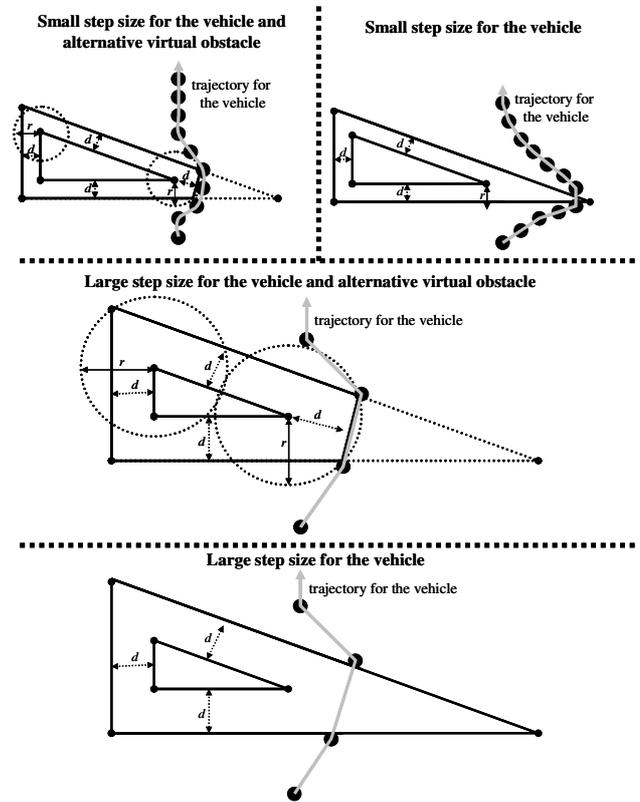


FIG 6. Alternative Description for Edges with Angles smaller than 90°

Depending on the form of the polygon and the influence of the step size between two trajectory points on (17) it can be helpful to increase the number of edges for the virtual obstacle in comparison to the number of edges of the real obstacle.

As shown in Fig. 6 in cases a small step size is used for a large vehicle the results from the optimization can be different depending on the design of the virtual obstacle. Sometimes additional edges should be added to avoid

that the optimization delivers trajectories that are not as good as they could be because of sub-optimal design of the virtual obstacles. Therefore a circle with the radius $r = d \cdot \sqrt{2}$ is taken around the edge of the real obstacle and than the intersections between this circle and the segments of the virtual obstacle are calculated. If two intersections are detected on a line between two edges of the virtual obstacles the intersection will be selected as new edge point of the virtual obstacle that is closer to the virtual edge tested,

By using this method for an edge the algorithm finds two new points for one native virtual edge in cases of sharp edges and by using them redesigns the edge as shown in Fig. 6.

4. DESCRIPTION OF THE MINIMIZATION PROBLEM

With the usage of the equations from chapter 2 and chapter 3 it is possible to construct a MIP minimization problem that contains the description of each UAV and the descriptions of all obstacles in the scenario. But this is not sufficient, in addition the description of the formation is necessary. Possible descriptions are shown for unmanned ground vehicles in [8] and [5]. The following descriptions of formation are an extension of the work from [5] into 3D and a more flexible formation, that allows the vehicles a movability inside of the formation in all three possible axis. Therefore the formation algorithm can be described as

$$(21) \quad \begin{aligned} x_{t,j+1} - x_{t,j} &\leq x_{dist_max,j} \\ -x_{t,j+1} - x_{t,j} &\leq -x_{dist_min,j} \\ x_{t,j+1} - x_{t,j} - k_{1,t,j} &= x_{dist} \\ y_{t,j+1} - y_{t,j} &\leq y_{dist_max,j} \\ -y_{t,j+1} - y_{t,j} &\leq -y_{dist_max,j} \\ y_{t,j+1} - y_{t,j} - k_{2,t,j} &= y_{dist} \\ z_{t,j+1} - z_{t,j} &\leq z_{dist_max,j} \\ -z_{t,j+1} - z_{t,j} &\leq -z_{dist_max,j} \\ z_{t,j+1} - z_{t,j} - k_{3,t,j} &= z_{dist} \end{aligned}$$

where $x_{dist_max,j}$, $y_{dist_max,j}$, $z_{dist_max,j}$ is the max. allowed distance between two connected UAVs described by (21) and $x_{dist_min,j}$, $y_{dist_min,j}$, $z_{dist_min,j}$ is the min. allowed distance between two connected UAVs described by (21). The optimal distance for the ideal flight formation is defined by x_{dist} , y_{dist} , z_{dist} as absolute distance between two by the equations for the description of formation connected UAVs.

In addition the optimization criteria for the formation stability is needed in the form

$$(22) \quad \min \sum_{l=1}^2 \sum_{t=1}^{T-1} \sum_{j=1}^M k_{l,t,j}^2$$

to ensure that the optimal relative positions between the vehicles are kept as often as possible as the ideal flight formation saves the most fuel/energy only in the ideal relative positions.

Using the description above for the description of the formation an additional minimization criterion for the path length must be added to the optimization criteria. This minimization criterion can be defined as

$$(23) \quad \min \sum_{t=1}^{T-1} \sum_{j=1}^M \Delta s_{t,j}^2$$

where $\Delta s_{t,j} = s_{t+1,j} - s_{t,j}$ is the difference of the position between two steps and $s_{t,j}$ is the representation of the vector $s_{t,j} = [x_{t,j}, y_{t,j}, z_{t,j}]^T$.

Now all necessary sub-problems for the MIP formulation are defined. Because of the design of the minimization criterion the problem has to be solved as a Mixed Integer Quadratic Programming (MIQP) minimization problem with a MIP solver like GLPK or CPLEX. The complete problem of optimal control can be formulated as

$$(24) \quad \min \left(\sum_{t=1}^{T-1} \sum_{j=1}^M \Delta s_{t,j}^2 + \sum_{l=1}^3 \sum_{t=1}^{T-1} \sum_{j=1}^M k_{l,t,j}^2 + \sum_{r=1}^T \sum_{r=1}^R a_r \epsilon_{t,r} \right)$$

subject to

$$\left[\begin{array}{c} \text{MLD system representation of (1) or (2)} \\ \text{description of obstacles} \\ (21) \\ s_{T,j} = s_{goal,j} \end{array} \right]$$

with T as the total number of steps, M as the total number of UAVs and $s_{T,j} = s_{goal,j}$ as the reaching goal criteria to ensure that after the maximum number of steps allowed all UAVs are at their goal positions for the optimized mission or mission part. The last double sum in the MIQP minimization criteria is representing the "in formation" and "out of formation" criteria. With the parameters a_r the importance of the flight in formation towards an out of formation flight can be defined.

5. RESULTS OF SIMULATION

For the verification of the MIQP minimization problem and the relevant equations, some test cases will be described in this chapter. The easiest way to test the functionality of the MIQP minimization problem and the result from solving it is a formation flight of UAVs in a 2D scenario. The 2D description of the obstacles, the dynamic model and the description of formation of the UAVs can be used if the flight level is already fixed by the mission plan and only the route of the vehicles can be optimized in the fixed height. In the case presented here the UAVs are already in formation and the formation can be used during the complete mission.

As shown in Fig. 7 the UAVs are able to avoid collisions with all possible obstacles that are used here to describe zones where a flight of the UAVs is prohibited. The virtual obstacles around the real obstacles are visualised in Fig. 7 in light grey while the real obstacles are shown in a darker grey inside of the virtual obstacles. Also it is obvious that the solution from the MIQP minimization problem is the optimal path through the scenario.

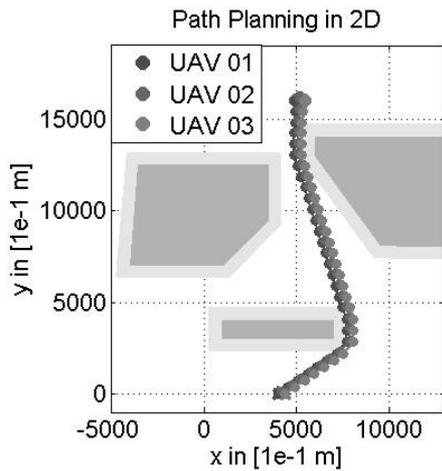


FIG 7. Path Planning in 2D

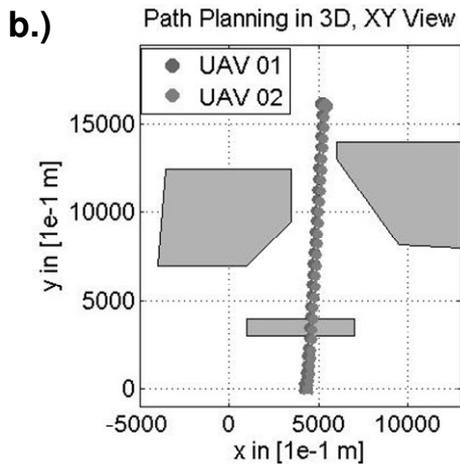
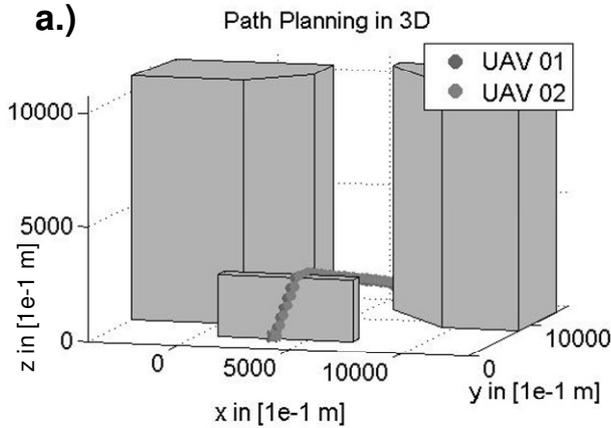


FIG 8. a.) Path Planning in 3D, as 3D view
b.) XY View on the Path Planning in 3D

If the height of the flight operation is not fixed by the mission plan the scenario from Fig. 7 can be described as a 3D MIQP minimization problem and than the solution differs from the one in Fig. 7. As shown in Fig. 8 the UAVs are flying above one of the obstacles and are than passing between the two other obstacles. In this scenario

the obstacle the UAVs are flying above is representing a large building complex while the other two obstacles are presenting areas where flights are not allowed so that in every possible height the UAVs are unable to pass over or below these obstacles. The effect of trajectory generation above obstacles in 3D environments instead of passing them at one side and the effects of possible factors of height limitation have already been investigated for single UAVs as in [7] and also in other publications for groups of UAVs using receding horizon control whereas it is here solved completely with the construction of a MIQP minimization problem and for a formation of UAVs and with a special optimization criteria to be able to switch into and to leave formations if this becomes necessary.

Temporary Formations in 2D, 1st Case

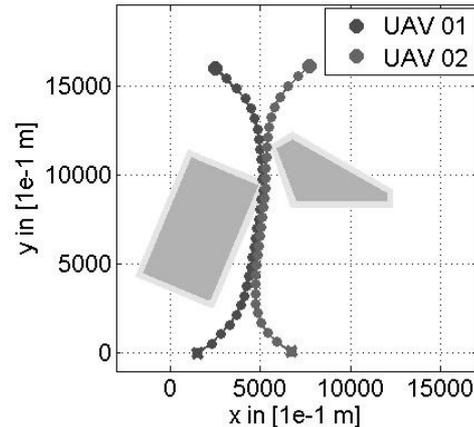


FIG 9. Temporary Formations in 2D, 1st Case

While in the two simulation runs before (Fig. 7 and Fig. 8) the UAVs are already in formation and they have only to keep this formation, the following scenario is different. There the UAVs are separated at the beginning and furthermore they have different target positions.

Temporary Formations in 2D, 2nd Case

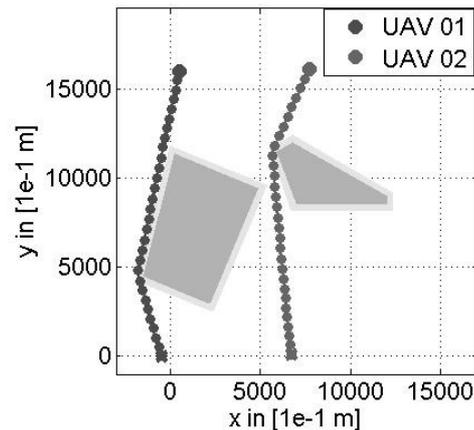


FIG 10. Temporary Formations in 2D, 2nd Case

Depending on the details of the scenario it will make sense for UAVs to create a temporary formation and to fly a certain part of their missions in formation, as shown in Fig 9. In the second case that is shown in Fig. 10 the UAVs do, however, not to go into formation flight because

at the beginning the distance of the UAVs is so far from each other that it is better for them to fulfil their missions independently without any temporary formation.

The example in Fig. 10 is only shown here to explain that the MIQP minimization problem is depending on its configuration parameters able to create solutions with temporary or permanent formation flights or to optimize the trajectories for every UAV without the creation of a formation. If the formation flight would result in a higher fuel/energy consumption due to a longer flight path than the formation flight effect reduces the fuel/energy consumption or if the formation flight would result in a mission that would last longer than the time limitation allows.

6. CONCLUSION

In this paper we have investigated a method for the description and optimization of UAV missions with a MIQP minimization problem using optimal trajectories generated from a MIP-solver. Therefore we have explained different possible methods for the description, of obstacles that represents in complex scenarios the mayor part of the equations from the MIQP minimization problem. Due to that effective methods for the description of obstacles are shown. In contrast to many other publications like [2] and [14] the MIQP minimization problem is formulated directly for the complete scenario and due to that the optimal trajectories according to the optimization problem parameters are directly delivered for the complete UAV mission. Furthermore it is shown that the minimization problem can be solved, so that the vehicles are able to create temporary formations and to leave them independently according to the tasks of the mission. Using such planning algorithms for real UAVs it will be possible to save fuel/energy as flying in formations saves significant amounts of energy according to [13] and [15].

REFERENCES

- [1] J. Beck, A. Knoll, "Autonome Organisation des Formationsflugs unbemannter Flugzeuge," *Deutscher Luft- und Raumfahrtkongress*, pp. 143-150, Nov. 6-9, 2006
- [2] J. Bellingham, A. Richards, J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," *Proceedings of the American Control Conference*, pp. 3741-3746, May 8-10, 2002
- [3] A. Brandstetter, F. Grässel, "Command and Control of Future Combat Air Systems in Network Enabled Operations," *Deutscher Luft- und Raumfahrtkongress*, pp. 135-142, Nov. 6-9, 2006
- [4] Y. Gu, B. Seanor, G. Campa, M. Napolitano, et. al., "Design and Flight Testing Evaluation of Formation Control Laws," *IEEE Transaction on Control Systems Technology*, Vol. 14, No. 6, pp. 1105-1112, Nov. 2006
- [5] T. Kopfstedt, M. Mukai, M. Fujita, O. Sawodny, "Formation Control for Mobile Robots in partially known Environments using Mixed Integer Programming and Fuzzy Systems," *SICE-ICASE International Joint Conference*, pp. 1832-1837, Oct. 18-21, 2006
- [6] T. Kopfstedt, "Möglichkeiten zur Leistungssteigerung durch den Einsatz von Telepräsenztechnologie bei der Robotersteuerung," *Deutscher Luft- und Raumfahrtkongress*, pp. 135-142, Nov. 6-9, 2006
- [7] Y. Kuwata, J. How, "Three Dimensional Receding Horizon Control for UAVs," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA2004-5144, Aug. 16-19, 2004
- [8] M. Mukai, T. Azuma, M. Fujita, "A Collision Avoidance Control for Multi-Vehicle Using PWA/MLD Hybrid System Representation," *IEEE Conference on Control Applications*, pp. 872-877, Sept. 2-4, 2004
- [9] W. Ren, W. Beard, E. Atkins, "Information Consensus in Multivehicle Cooperative Control," *IEEE Control Systems Magazine*, pp.71-82, April 2007
- [10] A. Richards, J. How, "Mixed-integer Programming for Control," *American Control Conference*, pp. 2676-2683, June 8-10, 2005
- [11] T. Schouwenaars, "Safe Trajectory Planning of Autonomous Vehicles," *PhD Thesis, Massachusetts Institute of Technology*, Feb. 2006, Chapters: 2.3 and 3.4
- [12] G. Tanaka, A. Kojima, "A modelling of crowd behaviour based on model predictive control," *SICE-ICASE International Joint Conference*, Oct. 18-21, 2006
- [13] M. Vachon, R. Ray, K. Walsh, K. Ennix, "F/A-18 Aircraft Performance Benefits Measured during the Autonomous Formation Flight Project," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Aug. 4-8, 2002
- [14] J. Vervoort, T. Kopfstedt, "Mission Planning for UAV Swarms," *52nd Internationales Wissenschaftliches Kolloquium*, Ilmenau, Germany, Sept. 10-13, 2007
- [15] M. Wagner, D. Jacques, W. Blake, M. Pachter, "Flight Test Results of Close Formation Flight for Fuel Savings," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Aug. 4-8, 2002
- [16] W. Williamson, et. al., "An Instrumentation System Applied to Formation Flight," *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 1, pp. 75-85, Jan. 2007
- [17] C. Xia, P. Wang, F. Hadaegh, "Optimal Formation Reconfiguration of Multiple Spacecraft with Docking and Undocking Capability," *Journal of Guidance, Control and Dynamics*, Vol. 30, No. 3, pp. 694-702, May-June 2007