

A Two Step Velocity Planning Method for Real-time Collision Avoidance of Multiple Aerial Robots in Dynamic Environments [★]

J.J. Rebollo I. Maza A. Ollero

*Robotics, Vision and Control Group, University of Seville, 41092
Seville, Spain, (email: juanjorebollo@hotmail.com,
imaza@cartuja.us.es, aollero@cartuja.us.es.)*

Abstract: This paper presents a collision avoidance method to improve the safety conditions in scenarios with multiple UAVs sharing the same aerial space with other aircrafts or mobile obstacles. The method modifies the velocity profile of the UAVs under control maintaining the paths initially planned. It is based on the combination of both a Search Tree algorithm which finds a solution if it exists, and the minimization of a cost function which tries to find the nearest solution to the initially planned trajectories for the UAVs. The Search Tree algorithm provides an initial valid *order of pass* for the vehicles involved in a given conflict and allows to formulate the minimization problem as a LP problem that can be efficiently solved. A model for the UAVs has been considered in the paper and several simulations are presented with promising results.

1. INTRODUCTION

The coordination of a team of autonomous aerial vehicles offers many application possibilities. Team members can exchange sensor information, collaborate to track and identify targets and perform detection and monitoring activities among other tasks. Furthermore, this approach leads to redundant solutions offering greater fault tolerance and flexibility than the use of a single Unmanned Aerial Vehicle (UAV).

In this paper, a disaster management scenario from the AWARE project (AWARE consortium [2006]) is considered. In this project several UAVs cooperate with a ground wireless sensor-actuator network in detection and tracking activities. Furthermore, these aerial robots are used to deploy the ground sensor nodes and other devices, and eventually to repair the connectivity of the network which could be damaged in disaster scenarios.

Different techniques for optimal coordination and cooperation of multiple robots have been proposed and some of them can be applied in multiple UAV systems and particularly in the AWARE scenario. However, the variability of the flying conditions, due for example to the wind, the faults that may affect to the UAVs, and the presence of other manned aircraft, including teleoperated aerial vehicles that cannot be controlled by the system, demand the implementation of real-time collision avoidance techniques.

One of the first approaches to the collision avoidance problem in robotics was proposed in Kant and Zucker [1986] where the problem is decomposed into the path planning problem (PPP) and the velocity planning problem (VPP). Then, once a path has been planned, a velocity profile that

avoids collisions in that path is found by means of the proposed VPP method.

The problem of collision avoidance of a single mobile robot with mobile obstacles is considered in Tsubouchi and Arimoto [1994]. The method in this paper computes a collision-free trajectory in the (x, y, t) space. First, the authors evaluate the position and speed of the mobile obstacles. Assuming that the obstacles speed remain constant, they compute a set of oblique cylinders in the (x, y, t) space to be avoided. The problem is then to find a trajectory connecting the initial position to a vertical line representing the goal. A speed planning method with mobile obstacle avoidance was presented in Cruz et al. [1998] where mobile obstacles are included as vehicle's motion constraints. In Fujimori and Teramoto [2000] the direction angle and the velocity of the mobile robots are used as control variables for navigation and collision avoidance, by assuring the avoidance for two vehicles. The method in Owen and Montano [2005] computes the trajectory of a vehicle in the velocity space to avoid mobile or static obstacles in its trajectory. The coordinated modification of the trajectories of several robots that could be involved in the collisions is not considered in all these methods.

In Ferrari et al. [1997] alternative collision avoidance solution paths are obtained by generating small variations of robot motions in space and time. The method assumes that the vehicles have rather simple dynamics, and does not consider mobile obstacles.

The aircraft trajectory planning with collision avoidance is studied in Richards and How [2002] where the problem is written as a linear program subject to mixed integer constraints, known as a mixed-integer linear program (MILP) which can be solved using commercial software. The problem has a significant complexity because of the high number of constraints. Furthermore, it does not consider mobile obstacles.

[★] This work has been partially funded by the European Union AWARE Project (IST-2006-33579) and the AEROSSENS Project of the Spanish Research and Development Program (DPI2005-02293).

In Pallottino et al. [2007] a plan is proposed for steering multiple vehicles between assigned independent start and goal configurations and ensuring collision avoidance. All the agents cooperate by following the same traffic rules. They move with a constant velocity, a safety area is defined and the velocity of movement of the safety area can be zero. However, this method usually leads to the modification of the paths which could be not needed if the collisions are avoided by simply modifying the velocity.

When considering real-time aircraft collision avoidance the easiest strategy is to modify the aircraft altitude (Bichi and Pallottino [2000]) but air space is commonly structured in layers and therefore not always altitude changes are possible. Furthermore, most vehicles have significant dynamics limitations that do not allow them to stop or modify their trajectories as fast as needed.

The collision avoidance problem can be approached in two different ways depending on the available computation time. Collision free trajectories can be initially calculated before the vehicles start moving. This approach has no significant computation time constraints and is directly related to the coordination of multiple robots sharing resources. In this paper, we consider a different approach for dynamic environments in which the collisions are solved in real-time once they are detected. In this case, computation time plays an important role.

In this paper, a 3-D motion planning problem for multiple UAVs sharing the space with other aircrafts or teleoperated vehicles that are non integrated in the system (considered as mobile obstacles) is solved. Taking into account that the initial trajectories of the UAVs are designed to execute cooperatively particular tasks in a given mission, the objective of the collision avoidance algorithm is to find a collision-free solution that changes the initial trajectory as little as possible, by changing the velocity profile of the vehicles. In the paper, the dynamic model and physical constraints of the aerial vehicles are considered.

The method is implemented in two steps. The first one is an efficient heuristic initialization algorithm that finds solutions much faster than optimal methods by exploring a discretized space to compute a valid order of pass for the vehicles in each conflict. In the second step, a linear programming (LP) problem is formulated to compute which of these solutions minimize the difference with the initial plans.

In the next section, the collision avoidance problem is formulated. The proposed method and the simulation experiments are presented in the next two sections. The conclusions, future work and references close the paper.

2. PROBLEM FORMULATION

The collision avoidance problem allows a wide range of possible solutions. A possible simplification consists in a discretization of the 3D space using cubic cells. It does not allow to find an optimal solution, but makes possible to use fast search algorithms to reach feasible solutions.

A trajectory can be described as a sequence of cubic cells, each of them with an associated entrance and departure time. Therefore, a conservative policy to ensure a collision-

free trajectory would be to allow only one vehicle in each cell along all the trajectories. It is assumed that each UAV knows the trajectories of other UAVs as a list of cells. This makes easier to check whether a collision will occur, because each UAV simply has to find temporal overlapping between a cell of its trajectory and a cell that belongs to another UAV trajectory.

The 3D grid proposed in this paper decreases the data transmission requirements among vehicles, because they do not have to transmit the full trajectory in the continuous (x, y, z, t) space. This strategy also decreases the time needed to detect potential collisions.

Let consider a scenario with N UAVs whose trajectories pass through M cubic cells (each cell with an unique identifier). If two or more UAVs pass through the same cell, it will be considered as a *conflict*. Let us consider a variable C_{ik} , whose value is p if the i^{th} UAV has a conflict in the k^{th} cubic cell of its trajectory, being p the unique identifier of this cell. On the other hand, C_{ik} will be 0 if there is no conflict.

Let t_{ik} be the amount of time that the i^{th} UAV spends in cell k and T_{ik} the interval of time $[\sum_{j=1}^{k-1} t_{ij}, \sum_{j=1}^k t_{ij}]$. If γ_{ik} is the identifier of the k^{th} cell in the trajectory of the i^{th} UAV, there is a *collision* in the conflict p , if:

$$\bigcap_{\gamma_{ik}: C_{ik}=p} T_{ik} \neq \emptyset, \forall i = 1 \dots N, k = 1 \dots M \quad (1)$$

On the other hand, the following model from McLain and Beard [2005] has been considered for the UAVs:

$$\begin{aligned} \dot{x}_i &= v_i \cos(\psi_i) \\ \dot{y}_i &= v_i \sin(\psi_i) \\ \dot{\psi}_i &= \alpha_\psi (\psi^c - \psi) \\ \dot{v}_i &= \alpha_v (v^c - v) \\ \ddot{h}_i &= -\alpha_h h_i + \alpha_h (h_i^c - h_i) \end{aligned} \quad (2)$$

where α_ψ , α_v , α_h and α_{h_i} are known parameters that depend on the particular characteristics of the UAV, (x_i, y_i, h_i) are the 3D coordinates and ψ_i is the heading of the UAV. Regarding the heading rate and velocity, we consider the following constraints:

$$\begin{aligned} -c &< \dot{\psi}_i < c \\ v_{\min} &< v_i < v_{\max} \end{aligned} \quad (3)$$

where c , v_{\min} and v_{\max} are positive constants that depend on the dynamics of the particular UAV.

The problem to be solved can be expressed as the computation of t_{ik} in order to minimize:

$$J = \sum_{i=1}^N \sum_{k=1}^M (t_{ik} - t_{ik_{\text{ref}}})^2 \quad (4)$$

subject to:

$$\begin{aligned} t_{ik} - T_{ik_{\max}} &\leq 0 \\ T_{ik_{\min}} - t_{ik} &\leq 0 \end{aligned} \quad (5)$$

$$\bigcap_{\gamma_{ik}: C_{ik}=p} T_{ik} = \emptyset, \forall p = 1 \dots L, i = 1 \dots N, k = 1 \dots M \quad (6)$$

where L is the number of conflicts and $t_{ik_{\text{ref}}}$ is the time that the initial reference trajectory takes to pass through

the k^{th} cell of the i^{th} UAV trajectory. The objective of the cost (4) is to find a solution close to the reference trajectory, which is the initial trajectory of the vehicles before detecting the potential collision. Minimizing this cost, the changes in the trajectories will be minimum. On the other hand, equation (5) comes from the consideration of the UAV's model. $T_{ik_{max}}$ and $T_{ik_{min}}$ are the maximum and minimum time that the i^{th} UAV can stay in the k^{th} cell, which depend on the distance traveled in the cell, the initial velocity and the model considered.

It should be noted that to solve the collisions, all the UAVs whose trajectories cut the trajectories of the vehicles initially involved in the collisions. In each collision, we have considered three types of vehicles:

- *Directly involved*: vehicles that are involved in a potential collision detected.
- *Indirectly involved*: vehicles whose trajectories are cut by the trajectories of the *directly involved* vehicles and can cooperate with other UAVs to avoid the collisions.
- *Non-cooperative aircrafts*: *indirectly involved* vehicles that can not cooperate in the collision avoidance process. This type of vehicle will be therefore considered as a *mobile obstacle*.

When solving a collision among directly involved vehicles, new conflicts could arise. In the collision avoidance method presented in this paper, only the trajectories of the directly and indirectly involved vehicles can be changed, whereas the non-cooperative aircrafts are treated as mobile obstacles. However, it should be noted that in some cases it could be convenient to consider some cooperative UAVs as mobile obstacles. It allows to reduce the information required to be exchanged among the vehicles, the computational complexity of the algorithms.

3. THE PROPOSED COLLISION AVOIDANCE METHOD

The objective of the algorithm is to find how long should stay each vehicle in each cell of its trajectory. In this paper, we have developed an heuristic method based on the combination of both a Search Tree algorithm which finds a solution if it exists, and the minimization of the cost function (4), that will use the information obtained by the Search Tree algorithm. The Search Tree will compute a valid *order of pass* for the vehicles in a given conflict, allowing to formulate the problem to be solved as a LP problem (instead of a full MILP problem). The algorithms consider the UAV model and the distances traveled by the UAVs in each cell. Finally, it will be assumed that the UAVs are initially moving at their maximum speed, as far as it allows to perform the missions in a minimum time.

3.1 Initialization Algorithm: Search Tree

This algorithm searches for a solution, by assuming that each vehicle involved in the collision goes as fast as possible. This algorithm is based on the idea that there is no collision-free solution if there is a vehicle traveling as fast as possible and another, as slow as possible, which has to pass the next one through a certain conflict collides with the first. The algorithm does not consider the cost function.

Let us define the *order of pass* as the order in which the UAVs pass through a given conflict. Therefore, a conflict with n vehicles involved, has $n!$ different orders of pass. The algorithm explores the different orders of pass in each conflict until a solution is found. First of all, the most logical orders of pass are tested – in which the vehicle that has to travel less distance to arrive to the conflict would pass first. For each order, it is possible to determine if a solution exists in short computational time. If there is no solution for a given order of pass, the algorithm changes first the order of the vehicles that have to travel more distance to arrive to the conflict.

If the problem has m conflicts with n_i vehicles involved in the i^{th} conflict, there are $n_0!n_1! \dots n_{m-1}!$ orders to check. When a given order is explored and there is no solution, the algorithm permutes the order of the conflict with a higher cost J_i defined as

$$J_i = \mu_i - \sigma_i \quad (7)$$

where μ_i and σ_i are the mean and the standard deviation of the distance that each vehicle involved has to travel to arrive to the conflict. Therefore, the algorithm permutes first the order of the vehicles involved in conflicts that are farther from the beginning of their trajectories. In those conflicts, another conflict closer to the beginning could affect the search criterion defined above (the vehicle which has to travel less distance to arrive to the conflict, passes first). Differences in the distance that the vehicles travel to arrive make the initial order of pass more suitable. The term σ_i in (7) copes with this concept.

Each UAV has an associated tree for a certain order of pass with each node of the tree associated with a cell. The depth of a node indicates the position of the cell in the trajectory. Therefore, nodes that have the same depth will be associated with the same cell. Let $\omega(i, d)$ be the weight associated with the edge $i - d$ of the tree. This parameter measures the time that the UAV will spend in the cell associated with the d node, according to the UAV model. The length of the edge $i - d$ is proportional to $\omega(i, d)$. Once the tree is completely built, if it exists a solution, the height of the tree will be the number of cells of the trajectory. Algorithm 1 shows how the trees are built.

Basically, the trees grow from the root node by calculating the minimum weights $\omega(i, d)$ according to the UAV model (maximum speed), until a conflict node is found. Once detected, the associated weight could be calculated or not depending on the tree's turn. That turn corresponds to the order of pass being checked by the tree in a given iteration. Therefore, it is a tree's turn if the weights of the edges of other trees associated with the same conflict cell and associated with UAVs that have to pass through the conflict before, were already generated. If it is the tree's turn, it evaluates if a collision occurs in the conflict by checking if there is temporal overlapping among the times (weights) associated with other trees' branches associated with the same conflict. If a collision appears, the algorithm has to increase the degree of the previous node by computing a new branch with a new weight associated (not calculated at the maximum speed) that should avoid the temporal overlapping found. If the temporal overlapping persists, the algorithm would repeat the previous process. If backtracking, the algorithm reach

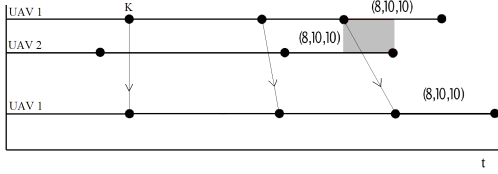


Fig. 1. Initial temporal overlapping between UAVs 1 and 2 in cell (8,10,10) corresponding to a potential collision (rectangular grey area between the timelines of the UAVs).

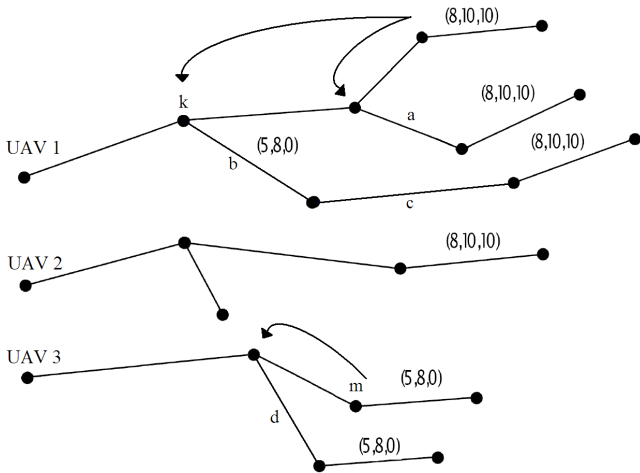


Fig. 2. The algorithm backtracks and creates new branches (*b* and *c*) which allow to avoid the collision. The three numbers between parenthesis are the identifiers of the cells involved.

the root node, there would not be any solution for the considered order of pass.

Figure 1 shows an example of an initial potential collision (temporal overlapping) between UAVs 1 and 2 in a cell with identifier (8,10,10).

Figure 2 shows the scheme used to solve that collision: the algorithm backtracks and creates new branches. Branch *a* is not valid because we would need to change the length of additional branches to avoid the collision and the UAV model does not allow to increase more the weight associated with branch *a*. Branches *b* and *c* are valid allowing to arrive later to the cell identified by (8,10,10) and solving the conflict (see the bottom timeline in Fig. 1 corresponding to the modified trajectory for the UAV with identifier 1).

The changes made by the Search Tree Algorithm to avoid the temporal overlapping could affect to other trees. Figure 2 shows that the tree associated with UAV 3 has to rebuild its branches before the node *m*, because UAV 1 passes first through the cell (5,8,0) and UAV 3 would collide with UAV 1. The time that UAV 3 stays in the cells previous to (5,8,0) has to be recomputed, because the condition that has to be fulfilled in the conflict has changed.

If the Search Tree algorithm does not find a solution for a certain order of pass, all the trees start from the

Algorithm 1 Search Tree Algorithm

```

while there is no solution and all the orders of pass have
not been explored do
    Start the tree associated with each UAV from the root
    node
    while the tree of each UAV is not complete do
        for each tree, if it is not complete do
            Calculate the weight of the next edge traveling at
            the maximum speed  $v_{\max}$  up to the next conflict
            node, creating the associated branches.
            if the end was not reached then
                if it is the tree's turn in the conflict then
                    Calculate the weight of the edge reaching the
                    conflict node
                    if there is a collision then
                        Go back and create new branches that
                        solve the collision, by associating higher
                        weights. Another tree may have to back-
                        track as well to ensure collision-free tra-
                       jectory
                        if the beginning of the tree is reached upon
                        backtracking, then there is no solution for
                        the order of pass being considered.
                    end if
                end if
            end if
        end for
    end while
    Get next order of pass
end while
    
```

beginning again using the next order of pass set by the search criterion defined in (7).

Directly and *indirectly* involved UAVs build the tree in the same way. However, the trees for the non-cooperative aircrafts are built from the beginning and are not changed (their trajectories can not be modified).

The Search Tree algorithm allows to find a solution in short time when comparing with methods that solve the problem of collision avoidance without considering a cell-divided space method.

3.2 LP Problem

If the Search Tree Algorithm finds a solution, we have a valid *order of pass* for the collision avoidance problem. It allows to transform a MILP problem (the full problem) into a LP problem. In fact, the binary variables of the MILP formulation allow to choose different *order of pass* in each conflict, and can be suppressed as far as we already have a valid order. Therefore, the problem to solve is to minimize:

$$J = \sum_{i=1}^N \sum_{k=1}^M (t_{ik} - t_{ik_{ref}})^2 \quad (8)$$

subject to:

$$\begin{aligned} t_{ik} - a_{ik}v_{ik} - b_{ik} &\leq 0 \\ b_{ik}v_{ik} + c_{ik} - t_{ik} &\leq 0 \end{aligned} \quad (9)$$

$$\begin{aligned} v_{ik} - v_{\max} &\leq 0 \\ v_{\min} - v_{ik} &\leq 0 \end{aligned} \quad (10)$$

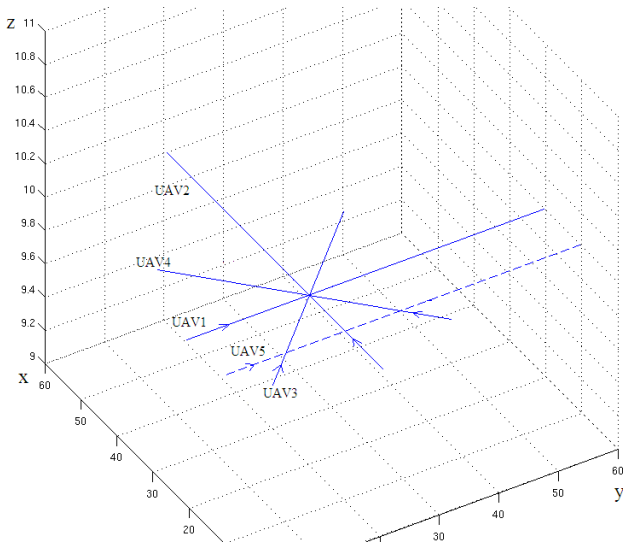


Fig. 3. Three dimensional paths of five UAVs used in a simulation to compare the performance of two different methods.

$$\forall i = 1 \dots N, k = 1 \dots M$$

and for each conflict and each UAV l that passes after another UAV m for the *order of pass* considered:

$$\sum_{k=1}^Q t_{mk} - \sum_{k=1}^P t_{lk} \leq 0 \quad (11)$$

where P indicates the cell previous to the conflict in the trajectory of the UAV l , and Q indicates the conflict cell in the trajectory of the UAV m . Equations (9),(10) take into consideration the UAV model. Equation (10), set the maximum and minimum time that an UAV can spend in each cell by linearizing $t(v_{ik})$ around the reference velocity in cell ik , allowing to have linear model constraints. In the next section, the effects of this approximation will be analyzed. v_{\max} and v_{\min} are the maximum and minimum velocities of the UAVs. Equation (11) ensures collision free trajectories, avoiding temporal overlapping.

The optimization problem shown here can be easily translated into the AMPL (Fourer et al. [1993]) modeling language. Finally, the CPLEX optimization software has been used to solve the problem.

4. SIMULATIONS

A scenario with five UAVs have been considered for the simulations (see Fig. 3). The method presented in this paper to minimize the cost function (4) has been compared in simulation with a Tabu Search (TS) method described in Rebollo et al. [2007] and the results are provided in the next section. The TS algorithm enhances the performance of a local search method by using memory structures to avoid local minima. On the other hand, TS does not consider a linearization of the UAV model, in contrast to the LP formulation of this paper.

4.1 Simulation Results

In this section, the results for the above mentioned methods with the paths shown in Fig. 3 are presented. UAVs

from 1 to 4 are considered *directly involved* vehicles and UAV 5 as a *mobile obstacle*. In Table 1, the different conflicts are summarized. There is a collision among four UAVs and four conflicts between two UAVs.

Table 1. Summary of the conflicts among UAVs for the scenario depicted in Fig. 3. There is a collision among four UAVs and four conflicts between two UAVs.

Cell	UAV
(20,20,10)	UAV 1, UAV 2, UAV 3, UAV 4
(20,10,10)	UAV 2, UAV 5
(10,10,10)	UAV 3, UAV 5
(30,10,10)	UAV 4, UAV 5
(29,10,10)	UAV 4, UAV 5

Table 2 shows how the four UAVs collision is solved, by comparing the entrance and leaving time in the conflict cell. The TABU, LP and initial reference times are provided.

Table 2. Solution computed to the four UAVs collision. The times are represented in seconds.

	$t_{in_{tabu}}$	$t_{out_{tabu}}$	$t_{in_{LP}}$	$t_{out_{LP}}$	$t_{in_{ref}}$	$t_{out_{ref}}$
UAV 3	13.8	15.3	18.3	19.4	19.2	20.4
UAV 2	15.3	15.9	19.4	20.3	20	21
UAV 1	16.1	18.1	20.3	21.3	20	21
UAV 4	19.2	20.5	21.3	22.7	19.4	20.8

It can be seen that the LP solution gives better results. But it should be noted that we are using a linearization of the UAV model to have the LP formulation. Therefore, it should be checked how far is the computed solution from the linearization point. The model was linearized around the reference velocity in each cell and by comparing the reference velocity and the solution velocity, we have the worst case with $v_{ref} = 1$ and $v = 1.11$. Then, it is near the linearization point and the solution is valid. In the LP solution, the value of the cost function is $J = 0.23$, which shows how near is the obtained solution from the initial reference trajectories.

In Table 3, the computational times with different number of involved UAVs are listed. The results clearly shows a reduction of the computational time thanks to the linearization of the UAV model and the formulation as a LP problem.

Table 3. Computational times of the two methods tested for different number of UAVs. The results are represented in seconds and have been obtained with a 1.7 GHz PC with 1GB RAM.

Number of UAVs	TABU	LP
2	0.74	0.025
3	1.21	0.028
4	1.72	0.032
5	2.67	0.04

The computational time does not depend on the shape of the path, because each path is a sequence of cells and the algorithm deals with them in the same way.

5. CONCLUSIONS AND FUTURE WORK

In this paper, a new method to solve the collision avoidance problem between multiple cooperative UAVs and

other aircrafts (mobile obstacles) sharing the same space has been presented. The proposed method is able to find a solution by changing in real time the trajectories of the vehicles as little as possible. This complex problem have been solved in two steps. In the first one a Search Tree algorithm is used to find an initial solution for the second step in which a Linear Programming method is used. The results obtained were satisfactory because the solution found allowed the initial trajectories to remain nearly unchanged and the execution time met hard time constraints.

In situations where it is not possible to find a solution without changing the path, some additional strategies has to be used. It would be possible to modify the altitude (Wollkind [2004]) or to create roundabouts (Massink and Francesco [2001]) to solve the collisions. But once the new path is obtained, the algorithm presented in this paper could still be used to find the velocity profile that provides a solution according to our requirements.

The proposed collision avoidance method can change or hold constant the trajectories of the different vehicles involved in a collision. Future work will include the design of a higher level entity that will decide how each vehicle should be treated to ensure a near optimal solution.

ACKNOWLEDGEMENTS

The authors would like to thank Boeing Research & Technology Europe for their approaches to the collision avoidance problem, which have been very useful to focus this research.

REFERENCES

- A. Bichi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–231, 2000.
- AWARE consortium. AWARE Project: Platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with aerial objects. <http://www.aware-project.net>, 2006.
- A. Cruz, A. Ollero, V. Munoz, and A. García-Cerezo. Speed planning method for mobile robots under motion constraints. In *Intelligent Autonomous Vehicles (IAV)*, pages 123–128, March 1998.
- C. Ferrari, E. Pagello, M. Voltolina, J. Ota, and T. Arai. Multirobot motion coordination using a deliberative approach. In *Second Euromicro Workshop on Advanced Mobile Robots (EUROBOT '97)*, page 96, 1997.
- R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Scientific Press, San Francisco, 1993. ISBN 0-89426-232-7.
- A. Fujimori and M. Teramoto. Cooperative collision avoidance between multiple mobile robots. *Journal of Robotic Systems*, 17(3):347–363, 2000.
- K. Kant and S. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3), 1986.
- M. Massink and N. De Francesco. Modelling free flight with collision avoidance. In *Proceedings of the Seventh International Conference on Engineering of Complex Computer Systems*, pages 270–279, 2001.
- T.W. McLain and R.W. Beard. Coordination variables, coordination functions, and cooperative timing missions. *Journal of Guidance, Control, and Dynamics*, 28:150–161, 2005.
- E. Owen and L. Montano. Motion planning in dynamic environments using the velocity space. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS 2005)*, pages 2833–2838, August 2005.
- L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi. Decentralized cooperative policy for conflict resolution in multi-vehicle systems. *IEEE Transactions on Robotics*, 2007. (accepted).
- J.J. Rebollo, I. Maza, and A. Ollero. Collision avoidance among multiple aerial robots and other non-cooperative aircraft based on velocity planning. In *In Proceedings of ROBOTICA 2007 Conference*, Paderne, Portugal, April 2007.
- A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of American Control Conference*, pages 1936–1941, 2002.
- T. Tsubouchi and S. Arimoto. Behavior of a mobile robot navigated by an iterated forecast and planning scheme in the presence of multiple moving obstacles. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2470–2475, 1994.
- S. Wollkind. *Using Multi-Agent Negotiation Techniques for the Autonomuos Resolution of Air Traffic Conflicts*. PhD thesis, University of Texas, 2004.