MILP AND ITS APPLICATION IN FLIGHT PATH PLANNING

W.A. Kamal, **D.-W. Gu**[1] and **I. Postlethwaite**

Control and Instrumentation Group, Department of Engineering, Univeristy of Leicester, Leicester, UK
Email: {wak5,dag,ixp} @le.ac.uk

Abstract: A finite receding horizon approach for path planning of uninhabited air vehicles (UAVs) is presented. The approach is based on mixed integer linear programming (MILP) techniques. Various constraints are formulated to avoid radar zones and collisions, etc. These constraints are extended to be both hard and soft so as to alleviate the infeasibility problem usually encountered. The finite receding horizon approach is numerically stable and can be applied to the path planning of a fleet of UAVs. Further improvements are possible for use in real time planning. The MILP is solved using commercially available software AMPL/CPLEX. *Copyright© 2005 IFAC*

Keywords: MILP, UAV, Autonomy, Flight Path Planning, Receding Horizon

1. INTRODUCTION

The mixed integer linear programming (MILP) approach is for optimisation problems which have integer variables in the cost function and/or the constraints. Such problems arise in engineering, economics, and many other disciplines. Commercially available software packages such as AMPL/CPLEX (Fourer *et al.*, 1993; Floudas, 1995) can be used to solve mixed integer optimisation problems. We shall show in this paper how the MILP can be used in the problem of flight path planning for Uninhabited air vehicles (UAVs).

UAVs are advantageous over manned counterparts in manoeuvrability, low human risk, low cost and light weight and is an important development area in the aerospace industry for the 21st century (Pachter and Chandler, 1998; Bortoff, 1999; McLain *et al.*, 2001; McLain, 1999; Chandler *et al.*, 2002). To be more powerful in applications, UAVs have to be more autonomous. Among many open issues in the development of autonomy, flight path planning, or trajectory selection, is crucial. Path planning algorithms must calculate a stealthy path which steers the vehicle away from potential dangers (threats, obstacles and collision with other vehicles if flying in a group). The path selected should be optimal in a certain sense and feasible for the vehicle to follow. The algorithm must be fast enough for real time use to deal with un-foreseen factors, and efficient in memory and computational demand to be run on airborne processors.

Path planners are generally divided into local and global ones and usually they do not take into account the dynamics of the vehicle. The former group work in on-line mode while the latter can be both on-line and off-line, though usually off-line. A global path planner requires all information beforehand. A clear disadvantage of a global planner is that a re-planning is necessary, which may take a long time, if the flight environment changes. Such changes happen frequently due to the uncertain environment, pop-up enemy threats, further information becoming available, etc. There is a tendency to design instead local path planners (Borenstein and Koren, 1991; Elnagar and Base, 1993). A local path planner does not suffer from the above disadvantage and can thus be

[1] Author for correspondence.

implemented in real time, though it may possibly lead to non-globally optimal flight paths. In this paper, a receding horizon strategy within the MILP framework is proposed which is based on the work of (Schouwenaars *et al.*, 2001; Richards *et al.*, 2001; Richards and How, 2002; Richards *et al.*, 2002). While it is very attractive in computation, the receding horizon strategy (Schouwenaars *et al.*, 2001) may fail to find a feasible solution in certain circumstance due to its local search characteristic. Hence, in this paper, it is proposed to include in the problem formulation "soft" constraints which accommodate infeasibility and find the least risk (most optimal) flight path in that situation.

It is shown that once the problem of path planning has been formulated as a mixed integer/linear constraints optimisation problem, it can be solved using commercially available software AMPL/CPLEX that uses branch and bound algorithms (Floudas, 1995). The optimisation problem is first translated into a program using the AMPL modelling language (Fourer *et al.*, 1993). The CPLEX optimiser is then applied to solve the problem (ILO, 1999).

The paper is organised as follows: Section 2 is concerned with problem formulation, including UAV dynamics, risk modelling and vehicle movement constraints. Section 3 introduces the receding horizon computation and the use of MILP. An example is used to demonstrate the algorithm in Section 4, with computational results and discussions. Conclusions are given in Section 5.

2. PROBLEM FORMULATION

2.1 Model of the Aircraft

The aircraft dynamics are expressed as a simple point mass in two diemansions with position and velocity $[x, y, v_x, v_y]^T$ as state variables and acceleration $[a_x, a_y]^T$ as control inputs. The measured outputs are the position variables.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0&0 \\ 0&0 \\ 1&0 \\ 0&1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

or

$$\dot{\mathbf{s}} = A_c \mathbf{s} + B_c \mathbf{u} \quad . \tag{1}$$

The zero-order hold equivalent discrete time system is

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1&0&\Delta t&0 \\ 0&1&0&\Delta t \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_k + \begin{bmatrix} (\Delta t)^2/2 & 0 \\ 0 & (\Delta t)^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}_k \tag{2}$$

or

$$\mathbf{s}_{k+1} = A\mathbf{s}_k + B\mathbf{u}_k \quad , \tag{3}$$

where k is a time step and Δt is the time interval between two steps. The control input $[a_x, a_y]_k^T$ remains constant over each time interval Δt under the zero-order hold assumption.

2.2 Constraints to Avoid Radar Zones

The radar areas are modelled as rectangles with $(x_{min}^{rad}, y_{min}^{rad})$ and $(x_{max}^{rad}, y_{max}^{rad})$ as the coordinates of the lower left and upper right corner points of the obstacle, respectively. At each time step i the position (x_i, y_i) of the vehicle must lie in the area outside of the risk area. In the case of N_V vehicles and N_R stationary obstacles, the general mixed integer linear constraints (Schouwenaars *et al.*, 2001)can be written as: $\forall p \in [1, \ldots, N_V], \forall c \in [1, \ldots, N_R], \forall i \in [1, \ldots, N]$:

$$\begin{aligned} x_{pi} &\le x_{c,min}^{rad} + M^{rad} b_{pci1}^{rad} \\ -x_{pi} &\le -x_{c,max}^{rad} + M^{rad} b_{pci2}^{rad} \\ y_{pi} &\le y_{c,min}^{rad} + M^{rad} b_{pci3}^{rad} \\ -y_{pi} &\le -y_{c,max}^{rad} + M^{rad} b_{pci4}^{rad} \\ \sum_{k=1}^{4} b_{pcik}^{rad} &\le 3 \\ b_{pcik}^{rad} &= 0 \texttt{ or } 1 \end{aligned} \tag{4}$$

In the situation of overlapped radar zones, the receding horizon algorithm may not work out feasible solutions due to such formulated hard constraints. For this consideration, the hard constraints can be transformed to soft ones by introduction of small variables as in the following $\forall p \in [1, \ldots, N_V], \forall c \in [1, \ldots, N_R], \forall i \in [1, \ldots, N]$:

$$\begin{aligned} x_{pi} &\le x_{c,min}^{rad}(1 - m_{pci1}^{rad}) + M^{rad}.b_{pci1}^{rad} \\ -x_{pi} &\le -x_{c,max}^{rad}(1 - m_{pci2}^{rad}) + M^{rad}.b_{pci2}^{rad} \\ y_{pi} &\le y_{c,min}^{rad}(1 - m_{pci3}^{rad}) + M^{rad}.b_{pci3}^{rad} \\ -y_{pi} &\le -y_{c,max}^{rad}(1 - m_{pci4}^{rad}) + M^{rad}.b_{pci4}^{rad} \\ \sum_{k=1}^{4} b_{pcik}^{rad} &\le 3 \\ 0 \le m_{pci1}^{rad},\ m_{pci2}^{rad},&\ m_{pci3}^{rad},\ m_{pci4}^{rad} \le 1 \end{aligned} \tag{5}$$

where m_{pci1}^{rad}, m_{pci2}^{rad}, m_{pci3}^{rad}, m_{pci4}^{rad} are auxiliary decision variables between 0 and 1. The idea is to reduce these variables to zero by incorporating them into the objective (cost) function. The problem formulation returns to the original setting when these $m's$ are zero. If it is not possible to reduce them to zero, i.e. the original hard constraints cannot be satisfied, the algorithm will have the flexibility to generate solutions which violate these constraints as little as possible.

2.3 Collision Avoidance Constraints

In the case of a fleet of UAVs, it is necessary to consider collision avoidance in path planning. This consideration can also be formulated using mixed integer linear constraints (Schouwenaars *et al.*, 2001). At each time step, every pair of vehicles p and q must be a minimum distance apart from each other in the x or/and y directions. At the i^{th} time step, let (x_{pi}, y_{pi}) and (x_{qi}, y_{qi}) be the positions of the vehicles p and q, respectively, and d_x^{col} and d_y^{col} the safety distances in the x and y directions, then the collision avoidance constraints can be set as (Schouwenaars *et al.*, 2001): $\forall i \in [1,\ldots,N], \forall p \in [1,\ldots,N_V], \forall q \in [p+1,\ldots,N_V]:$

$$\begin{aligned}
x_{pi} - x_{qi} &\geq d_x^{col} - M^{col} b_{pqi1}^{col} \\
x_{qi} - x_{pi} &\geq d_x^{col} - M^{col} b_{pqi42}^{col} \\
y_{pi} - y_{qi} &\geq d_y^{col} - M^{col} b_{pqi3}^{col} \\
y_{qi} - y_{pi} &\geq d_y^{col} - M^{col} b_{pqi4}^{col} \\
b_{pqik}^{col} &= 0 \text{ or } 1
\end{aligned} \quad (6)$$

Similarly these constraints can be converted to soft constraint to increase the solvability of the optimisation problem by introducing auxiliary variables as: $\forall i \in [1,\ldots,N], \forall p \in [1,\ldots,N_V], \forall q \in [p+1,\ldots,N_V]:$

$$\begin{aligned}
x_{pi} - x_{qi} &\geq d_x^{col}(1 - m_{pqi1}^{col}) - M^{col} b_{pqi1}^{col} \\
x_{qi} - x_{pi} &\geq d_x^{col}(1 - m_{pqi1}^{col}) - M^{col} b_{pqi2}^{col} \\
y_{pi} - y_{qi} &\geq d_y^{col}(1 - m_{pqi2}^{col}) - M^{col} b_{pqi3}^{col} \\
y_{qi} - y_{pi} &\geq d_y^{col}(1 - m_{pqi2}^{col}) - M^{col} b_{pqi4}^{col} \\
\sum_{k=1}^{4} b_{pqik}^{col} &\leq 3 \\
b_{pqik}^{col} &= 0 \text{ or } 1 \\
0 \leq m_{pqi1}^{col},\ &m_{pqi2}^{col} \leq 1
\end{aligned} \quad (7)$$

2.4 Speed and Acceleration Constraints

The maximum speed v_{max} is enforced by an approximation to a circular region in the velocity plane (Richards and How, 2002; Richards *et al.*, 2002). For each vehicle, the velocity vector is projected to different directions to obtain

$$\forall m \in [1,\ldots,N_C^{v_{max}}],\ \forall i \in [1,\ldots,N]:$$

$$\dot{x}_i \cos(\frac{2\pi m}{N_C^{v_{max}}}) + \dot{y}_i \sin(\frac{2\pi m}{N_C^{v_{max}}}) \leq v_{max} \quad (8)$$

The above constraints require that the velocity vector be inside a regular polygon with N_C sides circumscribed about a circle of radius v_{max}. A constraint on the minimum speed can be expressed in a similar way. However, it is different from the maximum speed constraint in that at least one of the constraints must be active instead of all of them,

$$\exists m \in [1,\ldots,N_C^v],\ \forall i \in [1,\ldots,N]:$$

$$\dot{x}_i \cos(\frac{2\pi m}{N_C^v}) + \dot{y}_i \sin(\frac{2\pi m}{N_C^v}) \geq v_{min} \quad (9)$$

where N_C is the order of the discretization of the circle. Equation 9 is a non-convex constraint and can be written as a mixed integer linear constraint

$$\forall m \in [1,\ldots,N_C^v],\ \forall i \in [1,\ldots,N]:$$

$$\dot{x}_i \cos(\frac{2\pi m}{N_C^v}) + \dot{y}_i \sin(\frac{2\pi m}{N_C^v}) \geq v_{min} - M^v(1 - b_{im}^v)$$

$$\sum_{m=1}^{N_C^v} b_{im}^v \geq 1 \quad (10)$$

Similarly, the constraint for the upper bound on acceleration can be written as:

$$\forall m \in [1,\ldots,N_C^a],\ \forall i \in [1,\ldots,N]:$$

$$\ddot{x}_i \cos(\frac{2\pi m}{N_C^a}) + \ddot{y}_i \sin(\frac{2\pi m}{N_C^a}) \leq u_{max} \quad (11)$$

2.5 Cost Function Selection

The objective function can be taken as the sum of two costs: a quadratic cost function and a cost to minimise the violation of constraints. We can minimise a quadratic function whose variables must satisfy the state space equation of the dynamic system (1) as follows

$$\min_{\mathbf{s},\mathbf{u}} J = \min_{\mathbf{s},\mathbf{u}} \int_0^\infty (\mathbf{s}^T Q \mathbf{s} + \mathbf{u}^T R \mathbf{u}) dt \quad (12)$$

subject to

$$\dot{\mathbf{s}} = A_c \mathbf{s} + B_c \mathbf{u} \quad (13)$$

where $\mathbf{s} \in \Re^n$ is the state vector and $\mathbf{u} \in \Re^{n_u}$ is the control. The system is assumed to start from some initial state $\mathbf{s}_0$ and final (target) state $\mathbf{s}_f$. A linear form of the quadratic cost is given by:

$$\min_{\mathbf{w}_{pi},\mathbf{v}_{pi}} \sum_{p=1}^{N_V} J_p = \min_{\mathbf{w}_{pi},\mathbf{v}_{pi}} \sum_{p=1}^{N_V} (\sum_{i=1}^{N-1} \mathbf{q}_p^T \mathbf{w}_{pi} + \sum_{i=0}^{N-1} \mathbf{r}_p^T \mathbf{v}_{pi} + \mathbf{p}_p^T \mathbf{w}_{pN}) \quad (14)$$

subject to

$$\begin{aligned}
s_{pij} - s_{pfj} &\leq w_{pij} \\
-s_{pij} + s_{pfj} &\leq w_{pij} \\
u_{pik} &\leq v_{pik} \\
-u_{pik} &\leq v_{pik} \\
\mathbf{s}_{p,i+1} &= A_p \mathbf{s}_{p,i} + B_p \mathbf{u}_{pi}
\end{aligned} \quad (15)$$

The weighting matrices Q and R of the quadratic formulation have been replaced by nonnegative weighting vectors $\mathbf{q}$ and $\mathbf{r}$, and s_{pfj} is the given final j^{th} state.
As discussed above, one way to increase solvability of the problem while keeping the violation of constraints to a minimum is to soften the constraints by including small variables $(m's)$ in the cost function (14) as follows:

$$\min_{m^{rad}_{pcif}, m^{col}_{pqig}} \sum_{p=1}^{N_V} \tilde{J}_p = \min_{m^{rad}_{pcif}, m^{col}_{pqig}} \sum_{p=1}^{N_V} (w^{rad} \sum_{c=1}^{N_R} \sum_{i=1}^{N} \sum_{f=1}^{4} m^{rad}_{pcif} + w^{col} \sum_{q=p+1}^{N_V} \sum_{i=1}^{N} \sum_{g=1}^{2} m^{col}_{pqig}) \quad (16)$$

So the overall objective function is $J_p + \tilde{J}_p$ for the p^{th} vehicle. The constraint list in (16) should be expanded to include other constraints considered earlier as per case to form the optimisation problem.

3. RECEDING HORIZON APPROACH AND MILP

3.1 Receding Horizon Control

A major difficulty in using MILP is the computational demand it requires. The computations increase dramatically with the number of time intervals (steps). On the other hand, big time steps could lead to inaccurate or non-implementable "solutions" . In order to make the algorithm workable in real time or near real time, a receding horizon approach has to be considered. In this approach, the path is computed online by solving an MILP over a limited horizon (in terms of time intervals) at each time step. The procedure is composed of a sequence of locally optimal segments. At each time step, the MILP is solved for T future time intervals, where the length T of the planning horizon is chosen as a function of the available computing resources as well as the individual problem. Solving this MILP provides the input commands for the T future time steps. The solution is of course only locally optimal. Only a subset of these T input commands is actually implemented. The process is then repeated and a new set of commands is generated for the next time window. Usually the applied subset is restricted to the first control input, such that a new set of input commands is calculated at each time step. Since the controller is designed at every sampling instant, disturbances can easily be dealt with. The concept is equally applicable to single-input, single-output (SISO) and multi-input, multi output (MIMO) systems, both linear and nonlinear.

3.2 Possible Infeasibility with Receding Horizon

When using receding horizon approaches, non-existence of feasible solutions may occur during the procedure of MILP, though in theory there are solutions to the whole problem. This is because the look ahead horizon is limited. The vehicle can be led to a critical state for which MILP has no solution in the next iteration. In other words, a feasible solution for T further time steps at current time step i does not guarantee a feasible MILP at the time step $i+1$. This can be further explained by the situation in which in the last time step of the planning horizon, the vehicle is moving at maximum speed, while its position is just outside an obstacle that has not yet been spotted. The position of the vehicle satisfies the anti-collision constraints and so corresponds to a feasible solution of the MILP. At the next time step, the obstacle is identified and the vehicle needs to brake or turn exceeding the constraints on acceleration or on the available manoeuvre space; a solution will not therefore be found. Increasing the time horizon will ease this kind of situation, but will also raise the computational demand.

3.3 Safe Feasible Mechanism

In the radar/SAM exposure minimisation problem, there are no physical obstacles. Rather we have radars of various detection ranges. We may have, say, three types of radars: long range SAM unit (65 km), medium range SAM unit (25 km), and short range SAM unit (7 km). We can approximately model these circular regions with squares of the same length as the radii of these circles. So in order to make the problem feasible some minimum violation of these constraints can be allowed. These radar ranges can overlap with one another. So if the UAV path is totally blocked by these overlapping radars or if the receding horizon approach with hard constraints (4), (6) and cost (16) is used, then MILP leads to an infeasible solution. But by using the soft constraints (5), (7) and including auxiliary variables in the cost (16), we can always obtain a feasible solution. Violations are kept to a minimum by use of the small variables $(m's)$ in the cost. If further reduction of violations is required, we may model the threats as squares of flexible size, slightly greater than the actual fitted square. The increment can be taken as 10% of the actual square. In this way, a vehicle can enter the radar zone but has to follow the safest possible path by optimising this flexibility.

4. EXAMPLE AND SIMULATION RESULTS

4.1 Scenario of the Example

An example is used to demonstrate the algorithm. This example considers path planning for

3 UAVs which start from different positions but fly to the same destination. The operation region is 180 km by 200 km as shown in Figure 1 and has 10 defence units (radar and SAM) shown as circles in the figure. Five units are of medium range (25 km) centered at coordinates $(100, 100)$, $(125, 65)$, $(125, 135)$, $(50, 155)$, $(50, 45)$, respectively, and the rest are of short range (7 km) centered at $(42, 102)$, $(167, 182)$, $(167, 127)$, $(167, 37)$, $(167, 77)$. The initial positions of UAV1, UAV2, UAV3 are $(10, 10)$, $(10, 120)$, $(10, 180)$, respectively. All three UAVs start at the same time and move towards a common goal at $(170, 100)$. The UAVs fly at an initial speed of 200 m/s and with an initial heading angle of $40°$ with regard to the horizontal. It is required that the final speed of each vehicle when arriving at the destination is 100 m/s and the heading is along the x-axis. The maximum flight speed is 300 m/s (see in Table 1). To use the formulation described earlier, the defence units are modelled as squares containing the circular threat regions. Adoption of "Soft"constraints in the finite horizon optimisation scheme alleviates the possibility of infeasibility outcomes. The vehicles may enter threat zones but will try to leave them as soon as possible to keep violation of constraints at a minimum. It has been observed that maximum violations will occur when the vehicle enters a threat zone at maximum speed and is perpendicular to the square boundary at a tangent point with the circle. That makes it possible that the vehicle will actually enter the dangerous zone before it can make a turn. Hence, in order to reduce this exposure, an additional safety measure can be taken by expanding the squares by 10% as is done in this exercise (the dotted lines in Figure 1).

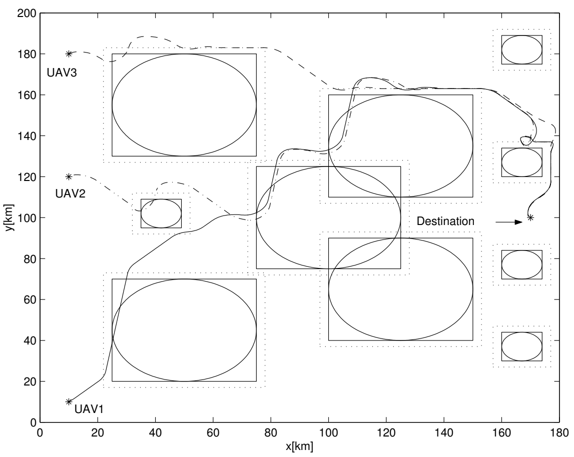

Fig. 1. Example Scenario and Obtained Flight Trajectories

4.2 Results and Discussions

In this exercise, the basic time interval is chosen as 1 second. The finite horizon T is chosen as 10

Table 1. Parameters used in the simulation

Parameter	Value	Parameter	Value
$\mathbf{q}_p^T$	$[1, 1, 1, 1]$	$\mathbf{r}_p^T$	$[1, 1]$
$\mathbf{p}_p^T$	$[1, 1, 1, 1]$	w^{obs}	10^{10}
w^{col}	10^{10}	d_x^{col}	1000
d_y^{col}	1000	M^{col}	$800,000$
M^{obs}	$800,000$	M^v	900
M^u	50	N_C^v	20
N_C^v	20	N_C^u	20
u_{max}	10	δt	1
v_{min}	100	v_{max}	300

seconds. That means at each iteration MILP finds an "optimal"solution for 10 seconds (a control sequence for the next 10 seconds). But of course only the first control input is to be implemented. All the parameters used in the computation are listed in Table 1. The computation is carried out using a PC machine with CPU of 2.66 GHz and RAM 1.048 Gb. The computation time for the full simulation is 27844 seconds that is the time when the last vehicle (UAV1) reaches the destination. The simulation times to destination for UAV2 and UAV3 are 27595 and 6386.3 seconds, respectively. The flight times taken by UAV1, UAV2, UAV3 are 1286, 1216, 856 seconds, respectively. Trajectories obtained for this example are shown in Figure 1. By comparing the computation and flight time, it is easy to see that a long computation time is required which makes it impracticable to be used in real time (on line) trajectory planning. Further improvements are definitely needed. A shorter horizon will reduce the computation time but will possibly make the solution less optimal. On the other hand, it is interesting to analyse in detail the time taken to solve this finite receding horizon MILP problem at each time step. This is shown in Figure 2. A high demand in computation occurs during the interval from 1030 to 1102 seconds with the peak demand (5473 seconds) at the 1036^{th} second. The trajectories for UAV1 and UAV2 around that particular time instant are shown in Figure 3. The UAVs are flying near the expanded boundary of the defence unit which is the last threat zone before they reach the destination. It can be seen in Figure 3 that at that time UAV1 takes a full turn to correct its direction immediately after exiting the (expanded) danger zone. In order to make a maximum turn, UAV1 has to reduce its speed to the minimum. On the other hand at that time instant, UAV2 is travelling exactly on the expanded boundary with minimum speed. Hence, the minimum speed constraint is active at the 1036 time step for both UAV1 and UAV2. As this constraint is non convex and involves integer variables, the computation of a solution is very hard and takes longer time. Similarly, for other iterations during the time interval $(1030 - 1102)$ seconds the minimum speed constraint remains active for either UAV1 or UAV2 (but not for both).

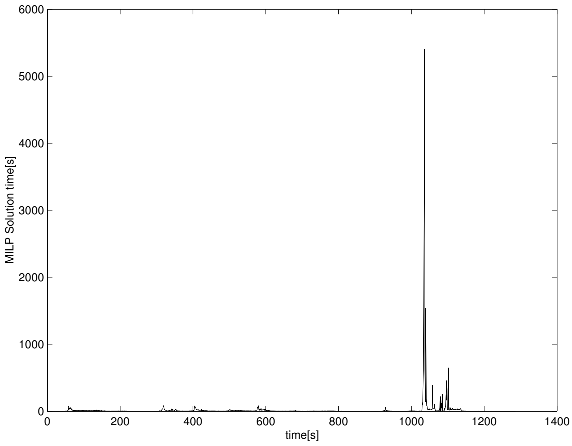

Fig. 2. Time taken by MILP to solve the problem at each time step for full simulation

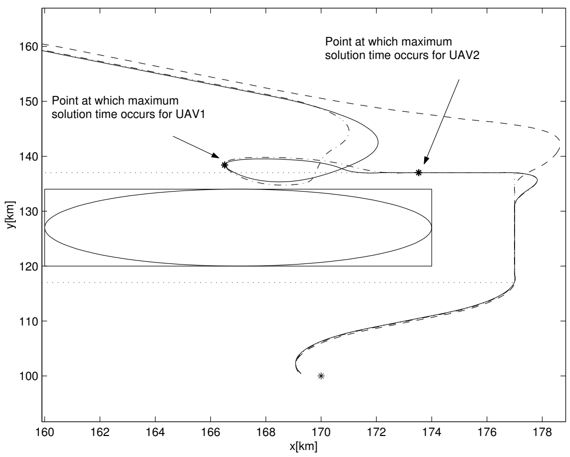

Fig. 3. Points on the trajectory where maximum solution time for both UAV1 and UAV2

5. CONCLUSIONS

In this work, we have shown that flight path planning for UAVs can be solved by a linear, constrained optimisation formulation with real and integer variables. A finite receding horizon method has been proposed which makes use of soft constraints. Available software packages such as AMPL/CPLEX can be applied in finding a solution. However, the MILP procedure requires a high computational demand. That makes it very difficult to perform in real time, though the introduction of finite receding horizon greatly helps the reduction of computation time. Further investigations are needed to formulate the constraints more effectively and to speed up the MILP computation. Future work will be focused to model radar zones with dynamic boundaries and comparison between the efficiency of MILP and of nonlinear optimisation will also be carried out.

6. ACKNOWLEDGEMENTS

The research work is supported by the UK Engineering and Physical Sciences Research Council and BAE Systems.

REFERENCES

Borenstein, J. and Y. Koren (1991). The vector field histogram, fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation* **7**(3), 278–288.

Bortoff, S. A. (1999). Path planning for unmanned air vehicles. Technical report. Air Force Research Laboratory. Air Vehicle Directorate.

Chandler, P., M. Pachter, D. Swaroop, J. Fowler, J. Howlett, S. Rasmussen, C. Schumacher and K. Nygard (2002). Complexity in uav cooperative control. In: *Proc. of the ACC*.

Elnagar, A. and A. Base (1993). Heuristics for local path planning. *IEEE Transactions on Systems Man and Cybernetics* **23**(2), 624–634.

Floudas, C. A. (1995). *Nonlinear and Mixed-Integer Programming-Fundamentals and Applications*. Oxford University Press.

Fourer, R., D. M. Gay and B. W. Kernighar (1993). *AMPL, A modeling language for mathematical programming*. The Scientific Press.

ILO (1999). *ILOG CPLEX User's guide*.

McLain, T., P. Chandler, S. Rasmussen and M. Pachter (2001). Cooperative control of uav rendezvous. In: *Proc. of the ACC*. pp. 2309–2314.

McLain, T. W. (1999). Coordinated control of unmanned air vehicles. Technical report. Air Vehicle Directorate, Wright-Patterson Air Force Base, Ohio.

Pachter, M. and P. Chandler (1998). Challenges of autonomous control. *IEEE Control Systems Magazine* pp. 92–97.

Richards, A. and J.P. How (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In: *Proceedings of the American Control Conference*.

Richards, A., J. Bellingham, M. Tillerson and J. P. How (2002). Coordination and control of multiple uavs. In: *Proc. of the AIAA GN&C Conf.*

Richards, A., J. P. How, T. Schouwenaars and E. Feron (2001). Plume avoidance maneuver planning using mixed integer linear programming. In: *Proc. of AIAA GN&C Conf.*

Schouwenaars, T., B. DeMoor, E. Feron and J. P. How (2001). Mixed integer programming for multi-vehicle path planning. In: *Proc. of the ECC Conf.*. Porto, Portugal. pp. 2603–2608.