40th Southeastern Symposium on System Theory
University of New Orleans
New Orleans, LA, USA, March 16-18, 2008

PATH PLANNING VIA CPLEX OPTIMIZATION

Taoridi A. Ademoye¶, Asad Davari¶, Charles C. Castello§, Sharon Fan§, Jeffrey Fan§
¶Department of ECE, West Virginia University Institute of Technology, Montgomery, West Virginia 25136
§Department of ECE, Florida International University, Miami, Florida 33174

Abstract — This paper presents an optimized solution of finding time-optimal trajectories for autonomous systems. These systems are subject to avoidance requirements, which include avoidance of collisions with other systems and obstacles, either static or dynamic. The necessary constraints for avoidance are added to a time-optimizing linear program by including a binary variable in the optimization. The resulting problem is a mixed-integer linear program (MILP). This will be solved using AMPL mathematical programming language in conjunction with CPLEX optimization software.

I. INTRODUCTION

The increasing power of computational resources makes possible the development of autonomous systems, such as Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs), which are capable of dealing with the complex task of path planning in dynamic environments. This paper presents the formulation of a mathematical program to optimize path planning techniques discussed in [3] and [4] which designs the optimal path between two states for a group of autonomous systems moving in a 2-Dimensional (2-D) environment and for a single system moving in a 3-Dimensional (3-D) environment.
The systems are constrained not to collide with each other or with any obstacle along the path of trajectory.

The use of autonomous systems in defense and security applications has increased significantly in recent years. Traditional applications for autonomous systems include Intelligence, Surveillance, and Reconnaissance (ISR), target acquisition, and communication relaying. Emerging uses include medical resupply, border patrol, maritime security, crowd control, search and rescue, and environmental monitoring. UAVs in this role provide reductions in manpower, as well as increased safety for personnel [1].

Previous work done on 3-D environments treats the problem by finding part of the trajectory using MILP and the other part using a cost map function. An algorithm is defined to connect the cost map and the detailed path in MILP [2]. This paper combines the work done in [3] and [4] by optimizing path planning techniques discussed in both research papers. This is done by developing a time optimal cost function that uses the necessary constraints of avoidance. The cost function is translated into an AMPL modelling language which is solved by CPLEX optimization software.

This paper begins with the kinematics model of both systems followed by the formulation of needed constraints. The trajectory optimization planning is later formulated separately for 2-D [3] and 3-D environments [4]. Examples are then presented to demonstrate the effectiveness of mathematical models introduced in this paper.

II. PROBLEM FORMULATION

A. Kinematics System Model

Two cases are considered for the kinematics model of the system depending on the dimension of the environment. In the first case, a situation where a number of vehicles or systems are moving in a 2-D environment is considered [5]. Systems are modeled as a point mass in 2-D or planar motion. In the second case, the system is modeled as a point mass in a 3-D environment. The discretized matrix form for the equation governing the motion of both systems is:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix}_{k+1} = A \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix}_{k} + B\mathbf{a}_k \tag{1}$$

where,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix},$$

$$A = \begin{bmatrix} I_3 & \Delta t.I_3 \\ O_3 & I_3 \end{bmatrix}, \quad B = \begin{bmatrix} O_3 \\ \Delta t.I_3 \end{bmatrix}$$

in a 3-D environment. The subscript k represents the discrete time-step, I_3 represents an identity matrix of size 3×3, and O_3 is a zero matrix of size 3×3. Vectors x, v, and a respectively represent the position, velocity, and acceleration input in the inertial frame [3].

Formulas used to govern motion for a 2-D environment are identical to formulas given above minus terms associated with the z-axis and where the identity and zero matrices are reduced to size 2×2, I_2 and O_2 respectively.

It is shown that by imposing a constraint on the maximum magnitude of the acceleration, the constraint on the maximum turning rate will also be satisfied [1]. In the remaining sections of this paper, the model of the vehicle presented in Eq. (1) will be used [4].

B. Two Dimensional

1) Dynamic Constraints

Assuming there are N vehicles, each approximated as a point mass moving in 2-D, the position of vehicle i at time step k is given by (x_{ik}, y_{ik}) and its velocity by (v_{xik}, v_{yik}), forming the elements of the state vector $\mathbf{X}_{ik}$. The control input is the acceleration of the vehicle (a_{xik}, a_{yik}) in the x- and y- directions respectively, forming the acceleration vector $\mathbf{a}_{ik}$. The discretized dynamics of the overall system, applied to all N vehicles up to T time steps, can be written as

$$\begin{aligned} &\forall i \in [1..N] \forall k \in [0..T] \\ &\mathbf{X}_{i(k+1)} = A\mathbf{X}_{ik} + B\mathbf{a}_{ik} \end{aligned} \tag{2}$$

If $\mathbf{X}_{iI}$ denotes the initial state of vehicle i, the initial conditions of the vehicle can be specified as

978-1-4244-1807-7/08/$25.00 ©2008 IEEE.

$$\mathbf{X}_{i0} = \mathbf{X}_{iI} \tag{3}$$

The following constraints limit the magnitude of the translational velocity and acceleration vectors, which in turn limits the magnitude of the rotational velocity of the vehicle [6]

$$|a_{xik}| \leq a_{\lim}, |a_{yik}| \leq a_{\lim} \tag{4}$$

$$|v_{xik}| \leq v_{\lim}, |v_{yik}| \leq v_{\lim} \tag{5}$$

Calculating the total velocity, v_k using v_{xk} and v_{yk} results in a nonlinear function, which is given as

$$v_{xik}^2 + v_{yik}^2 = v_{ik}^2 \leq v_{\max}^2 \tag{6}$$

It is shown from Eq. (6) that the summation of squared velocity components at any time must be within a circle whose radius is the maximum velocity squared in order to satisfy the magnitude constraint on the translational velocity. For an MILP representation, the linearized approach must be taken to constrain v_{xk} and v_{yk}. This is done by approximating the circle as a polygon with M sides where m represents a specific side. The corresponding constraint equations are thus given as

$$\begin{aligned} &\forall i \in [1..N] \forall k \in [1..T] \forall m \in [1..M] \\ &v_{xik}\cos(\frac{2\pi m}{M}) + v_{yik}\sin(\frac{2\pi m}{M}) \leq v_{\max} \end{aligned} \tag{7}$$

The constraints on the control input vector for each vehicle can also be written as

$$\begin{aligned} &\forall i \in [1..N] \forall k \in [1..T] \forall m \in [1..M] \\ &a_{xik}\cos(\frac{2\pi m}{M}) + a_{yik}\sin(\frac{2\pi m}{M}) \leq a_{\max} \end{aligned} \tag{8}$$

2) *Obstacle Avoidance*

The problem of obstacle avoidance is addressed in this subsection where each obstacle is approximated by a polygon. For simplicity, static obstacles are only considered in this paper, but dynamic obstacles are possible due to vehicle avoidance techniques, which are dynamic in nature. For any obstacle $c \in \{1,...,C\}$, let R_c denote the radius of the obstacle and (x_c, y_c) denote the coordinates of its center. Each obstacle is approximated by a polygon, denoted by $G(c)$, defined by a set of M_c inequalities. The polygon is given by

$$\begin{aligned} &\forall m \in \{1,...,M_c\} \\ &G(c) := \left\{ (x,y) : (x - x_c)\cos\frac{2\pi m}{M_c} \leq R_c \right\} \end{aligned} \tag{9}$$

In order for the vehicle to avoid the obstacles, its coordinates must be outside the region $G(c)$ at every time step. This condition can be written as $(x_k, y_k) \notin G(c)$. Because at least one constraint defining the region $G(c)$ must be violated in order to avoid the obstacle, the avoidance condition is equivalent to the following condition: there exists m such that

$$(x_{ik} - x_c)\cos\frac{2\pi m}{M_c} + (y_{ik} - y_c)\sin\frac{2\pi m}{M_c} > R_c \tag{10}$$

To express this avoidance constraint in an MILP problem formulation, it must be converted into an equivalent set of linear inequality constraints. This is achieved by introducing auxiliary binary variables $b_m \in \{0,1\}$ and the following M_c inequality constraints,

$$\begin{aligned} &\forall i \in [1..N] \forall k \in [1..T] \forall \forall m \in \{1,..,M_c\} \\ &(x_{ik} - x_c)\cos\frac{2\pi m}{M_c} + (y_{ik} - y_c)\sin\frac{2\pi m}{M_c} > R_c - Rb_m \end{aligned} \tag{11}$$

$$\sum_{m=1}^{M_c} b_m \leq M_c - 1, \quad b_m \in \{0,1\} \tag{12}$$

If b_m=1, the right side of Eq. (11) is significantly smaller than the other side due to R being a large arbitrary number. This makes the inequality trivially satisfied. If b_m=0, the equality is active and Eq. (12) ensures at least one constraint in Eq. (11) is active.

3) *Vehicle Avoidance*

The problem of vehicle avoidance will now be formulated in MILP form. In vehicle control, it is necessary to avoid other vehicles, stationary obstacles, and restricted regions. To safely avoid collision, there should be at least a safety distance apart between every pair of vehicles in each direction at each time step. Denoting the safety distances in the x- and y- directions by d_x and d_y, the constraints for collision avoidance between any two vehicles p and q can be written as

$$\forall k \in [1..T]$$

$$x_{pk} - x_{qk} \geq d_x - Rb_{pqk1} \tag{13}$$

$$x_{qk} - x_{pk} \geq d_x - Rb_{pqk2} \tag{14}$$

$$y_{pk} - y_{qk} \geq d_y - Rb_{pqk3} \tag{15}$$

$$y_{qk} - y_{pk} \geq d_y - Rb_{pqk4} \tag{16}$$

$$\sum_{t=1}^{4} b_{pqkt} \leq 3 \tag{17}$$

where b_{pqkt} is a binary variable (0 or 1). b_{pqkt}=0 indicates that there is at least a safety distance apart between vehicle p and q in one direction at time step k. Eq. (17) ensures that at least one b_{pqkt} is 0 at any time step, providing a safety distance in at least one direction.

4) *Time Optimal Cost Function*

This section addresses how MILP constraints can be used to include variable finishing times within a linear optimization. In this problem, vehicle i is required to reach its final destination

(x_{if}, y_{if}) at some time-step before the maximum time-step T. This problem can be solved by introducing the binary variable b_{ik}, which has a value of 1 if the i^{th} vehicle reaches its final destination at the k^{th} time step, and 0 otherwise. This is written in MILP constraints as

$$\forall i \in [1...N] \forall k \in [1...T]$$

$$x_{ik} - x_{if} \le R(1-b_{ik}) \tag{18}$$

$$x_{if} - x_{ik} \le R(1-b_{ik}) \tag{19}$$

$$y_{ik} - y_{if} \le R(1-b_{ik}) \tag{20}$$

$$y_{if} - y_{ik} \le R(1-b_{ik}) \tag{21}$$

$$\sum_{k=1}^{T} b_{ik} = 1, \quad b_{ik} \in \{0,1\} \tag{22}$$

where (x_{if}, y_{if}) denotes the destination coordinates for vehicle i. In the case of a single vehicle, it can also be seen that $b_{ik} = 1$ will force vehicle i to its destination. The minimum time solution for each vehicle is sought in this case by minimizing the sum of finishing times of each vehicle. The resulting cost function for multiple vehicles can thus be written as

$$\min_{\mathbf{X},\mathbf{a},b} J = \sum_{k=1}^{T}\sum_{i=1}^{N}\left\{ b_{ik}.\sum_{1}^{k}\Delta t + \tau(|a_{xik}| + |a_{yik}|)\right\} \tag{23}$$

The problem of finding the time optimal path for multiple vehicles is thus to minimize the cost function defined in Eq. (23) subject to dynamic, velocity, input, obstacle avoidance, and vehicle avoidance constraints in Eqs. (2), (3), (7), (8), and (11) – (22).

C. Three Dimensional

1) Dynamic Constraints

This subsection presents the dynamic constraints for a 3-D case which is similar to the 2-D case discussed earlier. The main difference is z-axis coordinates are needed for the position, velocity, and acceleration at any time step k. Hence, the discretized dynamics of the vehicle up to T time steps, can be written as

$$\forall k \in [0..T]$$

$$\mathbf{X}_{(k+1)} = A\mathbf{X}_k + B\mathbf{a}_k \tag{24}$$

The initial condition of the state vector is also specified as

$$\mathbf{X}_0 = \mathbf{X}_I \tag{25}$$

The magnitude limit constraints on the velocity and acceleration vectors, which in turn limits the maximum turning rate and the maximum pitch rate, provide that the optimization favours minimum time solutions [2] and [6], can be specified as

$$|a_{xk}| \le a_{\lim}, |a_{yk}| \le a_{\lim}, |a_{zk}| \le a_{\lim} \tag{26}$$

$$|v_{xk}| \le v_{\lim}, |v_{yk}| \le v_{\lim}, |v_{zk}| \le v_{\lim} \tag{27}$$

These constraints represented in Eqs (26) and (27) can be approximated by an arbitrary number (M) of constraints as

$$k \in [1..T] \forall m \in [1..M]$$

$$a_{xk}\cos(\frac{2\pi m}{M})\sin(\frac{\pi m}{M}) + a_{yk}\sin(\frac{2\pi m}{M})\sin(\frac{\pi m}{M})$$

$$+a_{zk}\cos(\frac{\pi m}{M}) \le a_{\max} \tag{28}$$

$$k \in [1..T] \forall m \in [1..M]$$

$$v_{xk}\cos(\frac{2\pi m}{M})\sin(\frac{\pi m}{M}) + v_{yk}\sin(\frac{2\pi m}{M})\sin(\frac{\pi m}{M})$$

$$+v_{zk}\cos(\frac{\pi m}{M}) \le v_{\max} \tag{29}$$

To prevent the vehicle from stonewalling, non-convex constraints are placed on the minimum speed along x- and y- directions. These constraints can be written as

$$k \in [1..T] \forall m \in [1..M]$$

$$v_{xik}\sin(\frac{2\pi m}{M}) + v_{yik}\cos(\frac{2\pi m}{M})$$

$$\le v_{\min} - 2v_{\max}b_{speed,m} \tag{30}$$

$$\sum_{m=1}^{M} b_{speed,m} \ge 1, b_{speed,m}\{0,1\} \tag{31}$$

Constraints on the maximum rate of climb and descent are also written as

$$v_{z,\min} \le v_z \le v_{z,\max} \tag{32}$$

2) Obstacle Avoidance

This subsection presents the obstacle avoidance constraints in MILP. For 3D environments, the most common obstacles are buildings which are defined by the coordinates of the lowest and highest corners or edges. Denoting the position of the vehicle at any time step k by (x_k, y_k, z_k) and the points representing the two corners of each obstacle by $(x_{low}, y_{low}, z_{low})$ and $(x_{high}, y_{high}, z_{high})$, the avoidance constraints can be expressed as

$$x_k \le x_{low} + Rb_{obst,1}$$

$$y_k \le y_{low} + Rb_{obst,2}$$

$$z_k \le z_{low} + Rb_{obst,3}$$

$$x_k \ge x_{high} - Rb_{obst,4}$$

$$y_k \ge y_{high} - Rb_{obst,5}$$

$$z_k \ge z_{high} - Rb_{obst,6} \tag{33}$$

$$z \ge 0 \tag{34}$$

$$\sum_{i=1}^{6} b_{obst,i} \le 5, \quad b_{obst,i} \in \{0,1\} \tag{35}$$

If $b_{obst,i}$ =1, the right side of the inequalities in Eq. (33) is a large, negative number that is always less than the left hand side. In this case, the inequality is inactive because it is trivially satisfied and if $b_{obst,i}$ =0, the inequality is active. Eq. (35) ensures that at least one of the constraints in Eq. (33) is active, and thus, obstacle avoidance is guaranteed.

3) Time Optimal Cost Function

This subsection addresses how MILP constraints are derived for a 3-D environment by modifying the results presented in the 2-D case. Here, the final destination of the vehicle is denoted by (x_f, y_f, z_f). The final value constraints are written as

$$\forall k \in [1 \ldots T]$$

$$x_k - x_f \le R(1-b_k) \tag{36}$$

$$x_f - x_k \le R(1-b_k) \tag{37}$$

$$y_k - y_f \le R(1-b_k) \tag{38}$$

$$y_f - y_k \le R(1-b_k) \tag{39}$$

$$z_k - z_f \le R(1-b_k) \tag{40}$$

$$z_f - z_k \le R(1-b_k) \tag{41}$$

$$\sum_{k=1}^{T} b_k = 1 \quad , \quad b_k \in \{0,1\} \tag{42}$$

If b_k =0, the inequalities in Eqs. (36) – (41) are relaxed. The vehicle will reach its destination only when b_k =1. Eq. (42) ensures that the vehicle reaches its destination at one time point. The minimum time solution is sought by minimizing the finishing time of the vehicle. The cost function is given as

$$\min_{\mathbf{X},\mathbf{a},b} J = \sum_{k=1}^{T} \{b_k . \sum_{1}^{k} \Delta t + \tau(|a_{xk}| + |a_{yk}| + |a_{zk}|)\} \tag{43}$$

The problem of finding the time-optimal path for a vehicle travelling in 3-D environments is solved by minimizing the cost function defined in Eq. (43) subject to dynamic, velocity, acceleration, and obstacle avoidance constraints in Eqs. (24) – (42).

III. EXAMPLES

The optimization problem described above can be easily translated to AMPL modelling language which is used with CPLEX optimization software to solve the problem [7].

A. *Two Dimensional Examples*

Consider a single vehicle moving through a field of fixed obstacles from a starting point to its final destination as shown in Fig.1. The vehicle is modeled as a simple 2-D discrete point mass with a discretization step of $\Delta t = 0.2s.$ The initial and final positions as well as the initial velocity of the vehicle are (3.2, 1.5), (13.7, 4.5) and (1, 0.5) units respectively. The other parameters used are:
M=8, $M_c = 8$, C=3, $v_{\max} = 5.0m/s$, $a_{\max} = 2.0m/s$, $\tau = 0.002$, R=800, T=50. The computational time is 5.0 seconds and the time optimal solution (i.e. the time it took the vehicle to cover this path) is 3.8 seconds. The linear program for this example has 1504 variables, 1153 integers, and 2146 constraints. In Fig.2 and Fig.3, the initial positions and destinations are points (0.5, 6.0) and (14.0, 5.2), respectively. Four obstacles are placed along the path of the vehicle in Fig.2. The linear program has 1866 variables, 1515 integers, and 2507 constraints. In Fig.3, another obstacle is added to the four in example two and the linear program for this example has 2238 variables, 1887 integers, and 2903 constraints. The optimization for both examples are solved in 9.0 seconds and 155 seconds respectively and the optimal solutions are 4.8 seconds for example two, and 5.0 seconds for example three.

In example four, a group of three vehicles is considered. The concern is to find optimal paths for three vehicles moving in a group from different initial positions, which are (3.2, 1.5), (3.2, 4.5), and (3.2, 7.8), and to different destinations, which are (13.7, 4.5), (13.7, 1.5), and (13.7, -1.5) respectively. The safety distance along each coordinate is assumed to be 0.8 units. Due to the increase in complexity of this problem, the computational time for this example is 8 minutes. The time optimal solution for each vehicle is 5.0 seconds. The linear program has 5243 variables, 3979 integers, and 8824 constraints.

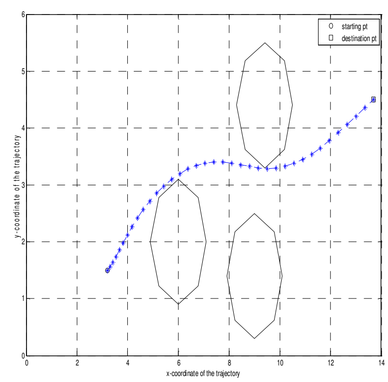

Fig.1 The designed path for a vehicle with three fixed obstacles

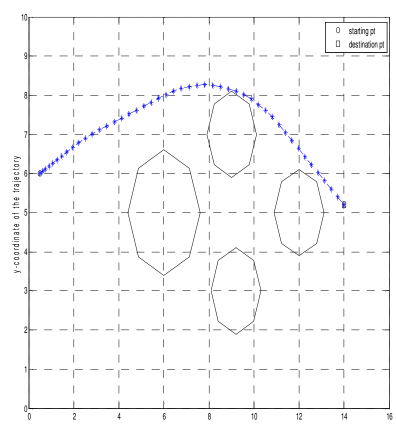

Fig.2 The designed path for a vehicle with four fixed obstacles

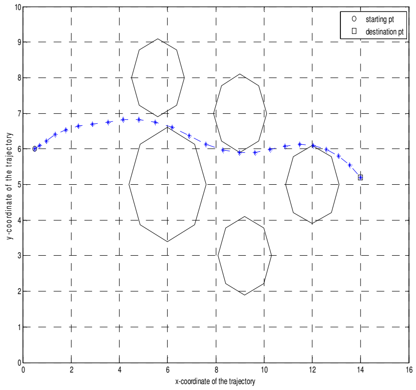

Fig.3 The designed path for a vehicle with five fixed obstacles

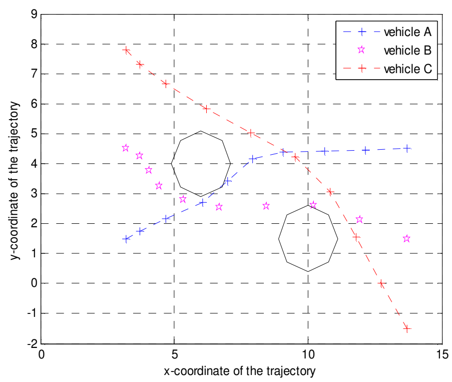

Fig.4 The designed time-optimal path for three vehicles with fixed obstacles

B. Three Dimensional Examples

The last two examples illustrate trajectory planning problems for 3-D cases. Consider an autonomous air vehicle that has to maneuver through a fixed rectangular shaped obstacle. The vehicle is modeled as a simple 3-D discrete point mass with discretization step $\Delta t = 0.2s$. Fig. 5 shows the trajectory when a single obstacle is placed between the initial position and the destination of the vehicle. The initial position is (1.0, 5.5, 2.0) and the destination is (12.0, 3.0, 1.0). This linear program has 979 variables, 378 binary and 601 linear, and 3612 constraints. The optimization problem was solved in 4 seconds and the time optimal path is 3.2 seconds or 16 time steps.

In Fig.6 the obstacle is increased to two. The initial position and the destination in this example are (1.0, 4.0, 3.0) and (10.0, 2.0, 0) respectively. The computational time is 7 seconds. The linear program has 1337 variables, 736 binary and 601 linear, and 4034 constraints. The time optimal path is 2.8 seconds or 14 time steps.

IV. Conclusions

This paper combines the working of [3] and [4] together where the first part deals with path planning for autonomous systems moving in 2-D environments and the second part discusses the same problem in 3 dimensions. This paper shows that obstacle avoidance, vehicle avoidance, and time optimization were incorporated and solved as one problem. Each problem was transformed into logical constraints that must be satisfied for optimization. It is also shown that with more constraints taken into consideration, computational time increases. For instance, in examples two and three, though the

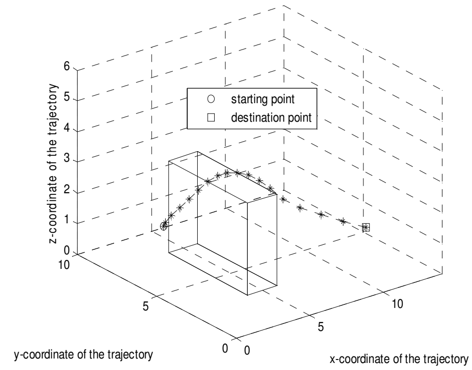

Fig.5 Time-optimal path for autonomous vehicle in 3-D environment

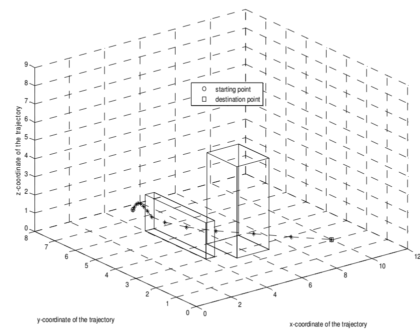

Fig.6 Time-optimal path for autonomous vehicle with double obstacles

vehicle was moving from the same initial positions to the same destination points, the computational times differed significantly because of the additional obstacle in example three. The numbers of constraints for both cases are 1515 and 2903 respectively. Therefore, the conclusion is the MILP method alone is not appropriate for finding the optimal path in a real time scenario although it is guaranteed to find the global-optimal solution.

References

[1] R. Fourer, D.M. Gay, & B.W. Kernighan, *AMPL – A Modeling Language for Mathematical Programming* (Thomson Brooks/Cole Academic Publisher, 2003).

[2] Y. Kuwata & J. How, *Three Dimensional receding Horizon Control for UAVs, in the proceeding of the AIAA Guidance*, Navigation and Control Conf., AIAA, Providence, Rhode Island, August 2004.

[3] T.A. Ademoye & A. Davari, *Trajectory Planning of Multiple Autonomous Systems Using Mixed Integer Linear Programming*, 38th Southern Symposium on System Theory, Cookeville TN, March 2006.

[4] T.A. Ademoye, A. Davari & Wei Cao, *Three Dimensional Obstacle Avoidance Maneuver Planning Using Mixed Integer Linear Programming*, 12th IASTED International Conference on Robotics and Applications, Honolulu, Hawaii, August 2006.

[5] T. Schouwenaars, B. De Moor, E. Feron, & J. Hao, *Mixed Integer Programming for Multiple – Vehicle Path Planning*, European Conference Control, 2001.

[6] Y. Hao, A. Davari, & A. Manesh, *Differential Flatness-Based Trajectory for Multiple Unmanned Aerial vehicles using Mixed-Integer Linear Programming*, in the proceeding of American Control Conference, Portland, 2005.

[7] *ILOG AMPL CPLEX System version 9.0 User's Guide*, September 2003