

MILP and NLP Techniques for Centralized Trajectory Planning of Multiple Unmanned Air Vehicles

Francesco Borrelli⁰, Dharmashankar Subramanian²,
Arvind U. Raghunathan¹ and Lorenz T. Biegler¹

Abstract—We consider the problem of optimal cooperative three-dimensional conflict resolution involving multiple Unmanned Air Vehicles (UAVs) using numerical trajectory optimization methods. The conflict problem is posed as an optimal control problem of finding trajectories that minimize a certain objective function while maintaining the safe separation between each UAV pair. We assume the origin and destination of the UAV are known and consider UAV models with simplified linear kinematics.

The main objective of this report is to present two different approaches to the solution of the problem. In the first approach, the optimal control is converted to a finite dimensional Nonlinear Program (NLP) by using collocation on finite elements and by reformulating the disjunctions involved in modeling the protected zones by using continuous variables. In the second approach the optimal control is converted to a finite dimensional Mixed Integer Linear Program (MILP) using Euler discretization and reformulating the disjunctions involved with the protected zones by using binary variables and Big-M techniques. Based on results of extensive random simulations, we compare time complexity and optimality of the solutions obtained with the MILP approach and the NLP approach.

NLPs are essential to enforce flyability constraints on more detailed UAV models. Moreover, any nonlinear extensions to the problem cannot be dealt with by MILP solvers.

The main objective of this paper is to open the route to the use of MILP solutions (based on simple linear UAV models) in order to initialize NLP solvers which allow the use of dynamic UAV models at any desired level of detail.

I. INTRODUCTION

The approaches to aircraft control for conflict resolution can differ on various criteria and objectives. A good review can be found in [20], [12], [13]. Our interest in resolution of conflicts between Unmanned Air Vehicles (UAVs) in free flight stems from the centralized computation of flyable trajectories for a flock of UAVs in a urban terrain.

In this paper we consider the problem of optimal cooperative three-dimensional conflict resolution involving multiple UAVs using mathematical programming techniques. We formulate a continuous-time optimal control problem where we minimize the inputs to a set of UAVs in order to let them move from a known initial configuration to a known final configuration without colliding with each other and without colliding with obstacles [14], [15], [16], [6]. UAVs are modeled as simple linear point-mass models with constraints on states and inputs. The protection zones associated with

each UAV pair and each UAV-obstacle pair are modeled using disjunctions. The main objective of this report is to compare two different approaches to the numerical solution of the optimal control problem: the Nonlinear programming approach presented in [20] and the Mixed Integer Linear programming approach presented in [18]. We do not make any attempt to improve the computational time of the respective mathematical programming solvers with formulation-specific heuristics. Such a study is beyond the scope of this paper.

There are three key steps required for the numerical solution of the optimal control problem: (i) the time-discretization of the continuous-time optimal control problem, (ii) the formulation of the disjunctions involved in modeling protection zones associated with each vehicle pair and vehicle-obstacle pair, and (iii) the choice of the mathematical programming solver used.

In the nonlinear programming approach [20] we employ collocation on finite elements to discretize the optimal control problem in time. In particular, the profiles of the optimization variables are approximated by a family of polynomials on finite elements. The disjunctions involved in modeling the protection zones are reformulated by using continuous variables and non-convex constraints. The resulting model is a nonconvex finite dimensional nonlinear program (NLP) which is solved by using an Interior Point algorithm that incorporates a novel line search method [20], [8]. The interior point algorithm is initialized as proposed in [7].

In the mixed integer programming approach [18] we employ a simple Euler discretization of the UAV dynamics. The disjunctions involved in modeling the protection zones are reformulated by using binary variables and big-M techniques. The resulting model is a finite dimensional Mixed-Integer Linear Program (MILP) which is solved by using a commercial branch and bound algorithm included in the CPLEX [17] package.

Based on results of extensive random simulations, we compare time complexity and optimality of the solution obtained with MILP approach and the NLP approach. The results of the comparison are summarized in Section VIII. The comparative study reported in this paper has several objectives.

NLPs are essential to enforce flyability constraints on more detailed UAV models. The first objective of this paper is to open the route to the use of MILP solutions (based on simple linear UAV models) in order to initialize NLP solvers which allow the use of dynamic UAV models at any desired level of detail.

The second objective of the study is to test the performance of the NLP solver presented in [20] in terms of

⁰ Dipartimento di Ingegneria, Università degli Studi del Sannio, 82100 Benevento, Italy, francesco.borrelli@unisannio.it.

¹ Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213.

² Honeywell Laboratories, Honeywell International, Minneapolis, MN 55418. Current affiliation: IBM Research, 1101 Kitchawan Road, Yorktown Heights, NY 10598.

optimality of the solution. As a last objective, we study the relation between time complexity of the optimal solution (both MILP and NLP) and some parameters of the optimization problem such as density of the obstacle field or the number of collisions when the vehicles fly along straight lines joining initial and final destination, at any speed.

The paper is organized as follows. In Section II we formulate the continuous time optimal control problem that applies to conflict resolution among multiple UAVs. In Sections III and IV we describe the NLP and the MILP approach, respectively, which have been used to numerically solve the optimal control problem. Section VI describes our method for generating random problem instances. Finally we conclude in Section VII with the results of the comparative study.

II. PROBLEM FORMULATION

In this section we describe the optimal control problem that applies to conflict resolution among multiple UAVs. We consider the conflict resolution of UAVs with dynamics described by a simple kinematic model

$$\begin{aligned}\frac{dx}{dt} &= v_x(t) \\ \frac{dy}{dt} &= v_y(t) \\ \frac{dz}{dt} &= v_z(t)\end{aligned}\quad (1)$$

where x, y and z represent the position of the UAV along the different axes. The velocities of the UAV in the x, y and z directions at time t are respectively $v_x(t), v_y(t)$ and $v_z(t)$. The maneuvers available to the UAV for conflict avoidance are changes in these velocities. The optimal conflict free trajectories are obtained as a result of minimizing the following objective,

$$\begin{aligned}\min \quad & \sum_{i=1}^n J_i(v_{x,i}^{(t_0,t_f)}, v_{y,i}^{(t_0,t_f)}, v_{z,i}^{(t_0,t_f)}) \\ \text{s.t.} \quad & \frac{dx_i}{dt} = v_{x,i}(t); \quad x_i(t_0) = x_{i,0}; \quad x_i(t_f) = x_{i,f} \\ & \frac{dy_i}{dt} = v_{y,i}(t); \quad y_i(t_0) = y_{i,0}; \quad y_i(t_f) = y_{i,f} \\ & \frac{dz_i}{dt} = v_{z,i}(t); \quad z_i(t_0) = z_{i,0}; \quad z_i(t_f) = z_{i,f} \\ & i = 1, \dots, n \\ & g_{ineq}(x_i(t), y_i(t), z_i(t), x_j(t), y_j(t), z_j(t)) \leq 0 \\ & \forall 1 \leq i < j \leq n \\ & g_{ineq}^{obs}(x_i(t), y_i(t), z_i(t)) \leq 0 \quad i = 1, \dots, n \\ & v_{x,i}(t)^L \leq v_{x,i}(t) \leq v_{x,i}(t)^U \quad i = 1, \dots, n \\ & v_{y,i}(t)^L \leq v_{y,i}(t) \leq v_{y,i}(t)^U \quad i = 1, \dots, n \\ & v_{z,i}(t)^L \leq v_{z,i}(t) \leq v_{z,i}(t)^U \quad i = 1, \dots, n\end{aligned}\quad (2)$$

where i indexes the UAV involved, t_0 is the initial time, t_f is the final time, $(x_{i,0}, y_{i,0}, z_{i,0})$ and $(x_{i,f}, y_{i,f}, z_{i,f})$ respectively denote the origin and destination of UAV i . The objective function in (2) weights the input profiles $v_{x,i}^{(t_0,t_f)}, v_{y,i}^{(t_0,t_f)}, v_{z,i}^{(t_0,t_f)}$ along the different axes from the initial time t_0 to the final time t_f of each vehicle and is related to the amount of fuel required for the maneuver. The set of constraints $g_{ineq} \leq 0$ are separation constraints between different pairs of UAVs. The set of constraints $g_{ineq}^{obs} \leq 0$ are separation constraints between UAV and static obstacles. $v_{x,i}(t)^L, v_{x,i}(t)^U, v_{y,i}(t)^L, v_{y,i}(t)^U, v_{z,i}(t)^L, v_{z,i}(t)^U$ denote point-wise upperbound and lower-bound on the velocity of the UAV i along the x, y and z axis, respectively.

In general, the optimal control problem (2) cannot be solved analytically. Numerical approximations are used to convert the infinite dimensional problem to a finite dimensional problem. In the next sections, we will briefly describe two approaches that we have used to numerically solve problem (2).

III. SOLUTION VIA NONLINEAR PROGRAMMING

In this section we briefly review the approach used in [20] to convert the infinite dimensional optimal control problem (2) into a finite dimensional nonlinear program. First, we detail the form of the cost function and of the constraints in (2) used in this approach. The cost function weights the one-norm of the UAV velocity vectors components

$$J_i(v_x, v_y, v_z) = \int_{t_0}^{t_f} [\|v_x(t)\|_1 + \|v_y(t)\|_1 + \|v_z(t)\|_1] dt \quad (3)$$

The inequality constraints g_{ineq} are chosen as

$$|x_i(t) - x_j(t)| \geq R \vee |y_i(t) - y_j(t)| \geq R \vee |z_i - z_j(t)| \geq H \\ i, j = 1, \dots, n, i \neq j \quad (4)$$

where $2R$ and $2H$ are base-length and height of square-base parallelepipedal protection zones around each UAV, respectively. The inequality constraints g_{ineq}^{obs} are chosen as

$$|x_i(t) - O_{xj}| \geq R_j \vee |y_i(t) - O_{yj}| \geq R_j \vee |z_i - O_{zj}| \geq H_j \\ i = 1, \dots, n, \quad j = 1, \dots, N_o \quad (5)$$

where (O_{xj}, O_{yj}, O_{zj}) are the centers of square-base parallelepipedal protection zones around the j th static obstacle. $2R_j$ and $2H_j$ are base-length and height of the parallelepipeds. N_o is the number of obstacles.

The key challenge in modeling the problem (2) is represented by the disjunctions in (4),(5) that model the protection zone around each UAV and each obstacle. The disjunctive equation can be formulated using continuous variables as described below. Suppose for simplicity that the disjunctive equation is given as,

$$g_1(t) \geq 0 \vee g_2(t) \vee g_3(t) \geq 0. \quad (6)$$

The above disjunction can be posed using continuous variables as follows

$$\begin{aligned}\lambda_1(t)g_1(t) + \lambda_2(t)g_2(t) + \lambda_3(t)g_3(t) &\geq 0 \\ \lambda_1(t) + \lambda_2(t) + \lambda_3(t) &= 1 \\ \lambda_1(t), \lambda_2(t), \lambda_3(t) &\geq 0.\end{aligned}\quad (7)$$

The second equation in the above formulation is critical to show equivalence to the disjunction (6). The equation denotes a normalization constraint for $\lambda_1(t), \lambda_2(t), \lambda_3(t)$. Of course, the right hand side could be any positive constant. In its absence, $g_1(\hat{t}) < 0, g_2(\hat{t}) < 0, g_3(\hat{t}) < 0, \lambda_1(\hat{t}) = \lambda_2(\hat{t}) = \lambda_3(\hat{t}) = 0$ for some \hat{t} , will satisfy the continuous formulation but violate the disjunction. Nevertheless, the first equation in (7) is nonconvex, and requires a good initialization to avoid local solutions or convergence to inconsistent points. Using the above formulation we can now discretize the optimal control problem to obtain a finite dimensional Nonlinear Program (NLP). In particular we use the monomial basis

representation [1] for the differential, algebraic and control variable profiles. The collocation points within each element are chosen to be at Radau points. The resulting NLP is solved using an interior point nonlinear programming solver, IPOPT [9], [10]. In Section III-A we give more details on the discretization procedure and on the NLP solver.

A. Collocation on Finite Elements and NLP Solver

We employ collocation on finite elements to discretize the optimal control problem (2). The profiles of the variables are approximated by a family of polynomials on finite elements. The time interval of interest $[t_0, t_f]$ is divided into ne finite elements of length h_i such that $\sum_{i=1}^{ne} h_i = t_f - t_0$.

Further we may define the time at the start of the intervals as, $t_i := t_0 + \sum_{l=1}^i h_l \forall i = 1, \dots, ne$. A monomial basis representation [1] is used for differential profiles $w_d(t)$, as follows:

$$w_d(t) = w_{d,n-1} + h_n \sum_{q=1}^{ncol} \Omega_q \left(\frac{t-t_{n-1}}{h_n} \right) \frac{dw_d}{dt}_{n,q} \quad (8)$$

$$t \in [t_{n-1}, t_n] \quad n = 1, \dots, N$$

where $w_{d,n-1}$ approximates the value of the differentiable profile $w_d(t)$ at the beginning of element n , $w_d(t_{n-1})$, $\frac{dw_d}{dt}_{n,q}$ approximates the first derivative in element n at the collocation point q and Ω_q is a polynomial of order $ncol$, satisfying,

$$\begin{aligned} \Omega_q(\rho_r) &= 0 \quad \forall q = 1, \dots, ncol \\ \dot{\Omega}_q(\rho_r) &= \delta_{q,r} \quad \forall q = 1, \dots, ncol \end{aligned}$$

where ρ_r is the r^{th} collocation point within each element, and $\delta_{q,r}$ is a Kronecker delta. We use Radau points as they allow to set constraints easily at the end of each element, and stabilize the system efficiently when high index constraints are present. In addition the monomial representation leads to smaller condition numbers and rounding errors [1]. Control and algebraic profiles ($w_a(t)$, $u(t)$, respectively) are approximated using the monomial basis representation as,

$$w_a(t) = \sum_{q=1}^{ncol} \psi \left(\frac{t-t_{n-1}}{h_n} \right) w_{a,n,q} \quad (9)$$

$$u(t) = \sum_{q=1}^{ncol} \psi \left(\frac{t-t_{n-1}}{h_n} \right) u_{i,q} \quad (10)$$

where $w_{a,n,q}$ and $u_{n,q}$ represent the values of the algebraic and control variables, respectively, in each element n at collocation point q , ψ_q is a Lagrange polynomial of order $ncol$ satisfying

$$\psi_q(\rho_r) = \delta_{q,r}.$$

The collocation formulation imposes continuity across the time elements for the differential variables (8), while the algebraic and control variables are allowed to have discontinuities at the boundaries of the elements.

We assume that the number of finite elements ne , and their lengths are pre-determined. With this assumption the substitution of the profile approximations and slacks for

the inequality constraints yields the following nonlinear programming problem,

$$\begin{aligned} J_{NLP}^* &= \min_{\xi \in \mathbb{R}^{n_\xi}} \phi(\xi) \\ \text{s.t.} \quad &c(\xi) = 0 \\ &\xi^L \leq \xi \leq \xi^U \end{aligned} \quad (11)$$

where $\xi = \left(\frac{dw_d}{dt}_{n,q}, w_{d,n}, w_{a,n,q}, u_{n,q} \right)$, $\phi : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}$ and $c : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}^{n_c}$.

The resulting NLP is solved using the interior point nonlinear programming solver IPOPT

The resulting NLP is solved using the interior point nonlinear programming solver IPOPT [10]. The reason is explained next. Optimal solutions for joint maneuvers typically bring UAVs to within the minimum separation. Different separation constraints are *active* depending on which pair of UAVs are at the minimum separation. Given the large number of separation constraints $N(N-1)/2(ne)(ncol)$ an *active set* algorithm can be expensive if a good guess for the optimal active set is not readily available. Barrier algorithms [4], [10] have been shown to overcome the combinatorial bottleneck of current Sequential Quadratic Programming (SQP) methods of identifying the active inequality constraints. Encouraging results have been obtained on a number of process engineering problems with a large number of inequality constraints [3]. The interested reader is referred to Wächter and Biegler [9], [10] and Biegler *et. al.* [4] for details on the convergence and implementation of IPOPT.

IV. SOLUTION VIA MIXED INTEGER LINEAR PROGRAMMING

In this section we describe the approach used to convert the infinite dimensional optimal control problem (2) into a finite dimensional Mixed Integer Linear Program (MILP) [18].

The objective function and constraints on the control inputs used in this approach are identical to the nonlinear programming approach. In the first step the UAV dynamics and cost function are discretized with a sampling time T_s by using a Euler discretization

$$\begin{aligned} x_i(k+1) &= x_i(k) + T_s v_{x,i} \\ y_i(k+1) &= y_i(k) + T_s v_{y,i} \\ z_i(k+1) &= z_i(k) + T_s v_{z,i} \end{aligned} \quad (12)$$

The disjunctions describing vehicles protection zones (4) are reformulated by using binary variables and “big-M” technique as described next. Consider two vehicles i and j , and six binary variables $b_{i,j}^1 \in \{0, 1\}, \dots, b_{i,j}^6 \in \{0, 1\}$. Constraint (4) is reformulated into the following set of mixed integer inequalities

$$\begin{aligned} x_i(k) &\geq x_j(k) + R - m_x b_{i,j}^1(k) \\ x_i(k) &\leq x_j(k) - R + M_x b_{i,j}^2(k) \\ y_i(k) &\geq y_j(k) + R - m_y b_{i,j}^3(k) \\ y_i(k) &\leq y_j(k) - R + M_y b_{i,j}^4(k) \\ z_i(k) &\geq z_j(k) + H - m_z b_{i,j}^5(k) \\ z_i(k) &\leq z_j(k) - H + M_z b_{i,j}^6(k) \\ \sum_{k=1}^6 b_{i,j}^k &\leq 5 \end{aligned} \quad (13)$$

where we assume that $x_i(k) \in \mathcal{X} \subset \mathbb{R}^n$, $y_i(k) \in \mathcal{Y} \subset \mathbb{R}^n$, $z_i(k) \in \mathcal{Z} \subset \mathbb{R}^n \forall i = 1, \dots, n$ and

\mathcal{X} , \mathcal{Y} , \mathcal{Z} are given bounded sets. The “bigM” constants m_x , m_y , m_z , M_x , M_y , M_z are lowerbounds and upperbounds on the difference between the position of two UAVs

$$\begin{aligned} M_x &\geq \sup_{x_1, x_2 \in \mathcal{X}} (x_1 - x_2 + R), \\ m_x &\leq \inf_{x_1, x_2 \in \mathcal{X}} (x_2 - x_1 + R), \\ M_y &\geq \sup_{y_1, y_2 \in \mathcal{Y}} (y_1 - y_2 + R), \\ m_y &\leq \inf_{y_1, y_2 \in \mathcal{Y}} (y_2 - y_1 + R), \\ M_z &\geq \sup_{z_1, z_2 \in \mathcal{Z}} (z_1 - z_2 + H), \\ m_z &\leq \inf_{z_1, z_2 \in \mathcal{Z}} (z_2 - z_1 + H), \end{aligned} \quad (14)$$

In the same way constraint (5) describing obstacles protection zones are reformulated into a set of mixed integer inequalities by using six binary variables $Ob_{i,j}^1 \in \{0,1\}, \dots, Ob_{i,j}^6 \in \{0,1\}$ for each obstacle-vehicle pair (i,j) and computing six constants $Om_x, Om_y, Om_z, OM_x, OM_y, OM_z$.

Given model (12) and constraints (13)-(14), problem (2) can be reformulated as a finite dimensional mathematical program which can be written as a mixed-integer linear program by using a standard approach which make uses of additional slack variables [21], [19]. We used the mathematical programming language AMPL [5] to transform problem (2) into an MILP of the form

$$\begin{aligned} \min_{\epsilon_c, \epsilon_b} \quad & f'_c \epsilon_c + f_b \epsilon_b \\ \text{subj. to} \quad & G_c \epsilon_c + G_b \epsilon_b \leq b \\ & \epsilon_b \in \{0,1\}^{n_b} \\ & \epsilon_c \in \mathbb{R}^{n_c} \end{aligned} \quad (15)$$

where $[\epsilon_b, \epsilon_c]$ is the optimization vector, which includes the inputs, the states and the binary variables associated with the vehicles and obstacles at all time steps, plus additional slack variables needed for the translation of the optimization problem in to an MILP. Then, the MILP (15) is solved by a *Branch and Bound* algorithm provided by the software CPLEX [17] which is interfaced with AMPL.

V. NLP/MILP FORMULATION DIFFERENCE

Before comparing the nonlinear and the mixed-integer approach presented in the previous sections, we highlight the only difference between the formulations used in the two approaches. The discretization in time of the original optimal control problem (2) is uniform over the horizon $t_f - t_o$ in the MILP approach, while it is a function of the position of the Radau points (see Section III-A) in the NLP approach. This implies that the NLP and the MILP optimal trajectories will fulfil the constraints at different points in time. For small sampling times (for the MILP) and big number of collocation points (for the NLP) this difference can be neglected. Also, since we are interested in the difference between the computational burden of the two techniques, the number of discretization points in time is more important than their collocation. We have used a sampling time T_s in (12) of 3s which results into 30 samples over an horizon of 90s. Accordingly in the NLP approach the number of Radau points over the same horizon have been chose equal to 30.

We want to mention here that we have also performed extensive simulations with NLP approach with cylindrical

protection zones around each UAV, and an objective function of minimizing the integral over time of a squared Euclidean norm of the UAV velocity vector (thus improving the smoothness of cost and constraint functions). For such formulation, the NLP solver performance was comparable to the formulation with a one norm in the objective function and parallelepipedal protection zones.

We remark that almost any changes in shape of models, flyability constraints, different flight models or objective function requires the solution of an NLP and cannot be recast as an MILP.

VI. RANDOM SCENARIOS GENERATOR

The comparison between the two solution methods presented in Section III and Section IV has been carried out on a set on problem instances that were randomly generated. In this section we give some details on the *instance-set generator*. A *instance-set generator* provides a set of random problem instances. For each problem instance belonging to the set, number of vehicles n and number of obstacles N_o are fixed. Each problem instance belonging to the set will be uniquely identified by the triplet (i, j, k) , $i \in \{n^{min}, \dots, n^{max}\}$, $j \in \{N_o^{min}, \dots, N_o^{max}\}$, $k \in \{1, \dots, N^{max}\}$. The problem (i, j, k) is the k -th random problem with $n = i$ vehicles and $N_o = j$ obstacles. Therefore, a set will be composed of $(n^{max} - n^{min} + 1)(N_o^{max} - N_o^{min} + 1)$ groups. Each group (i, j) is composed of N^{max} random problems with $n = i$ vehicles and $N_o = j$ obstacles. (i, j) will be the called the group number. We have considered random sets of problem instances with $N_o^{min} = 2, n^{min} = 2, N_o^{max} = 5, n^{max} = 10, N^{max} = 6$, for a total of 216 random problems for each *instance-set*.

For each problem instance belonging to the set, the algorithm randomly generates: (i) initial and terminal position of each vehicle: $(x_{i,0}, y_{i,0}, z_{i,0})$ and $(x_{i,f}, y_{i,f}, z_{i,f})$ for $i = 1, \dots, n$; (ii) N_o obstacles where the i -th obstacle O_i is described through the quintuple $(Ox_i, Oy_i, Oz_i, R_i, H_i)$. The algorithm makes sure that initial and terminal positions respect vehicles and obstacles protection zones.

In order to explore the possibility of defining a simple measure for time complexity of a problem instance, we associated a vector, $[a, b, c]$, to each problem instance. The component a denotes the density of the obstacle field measured as total volume of the obstacles field divided by the volume of field where UAVs are allowed to fly, $0 \leq a \leq 1$, $a \in \mathbb{R}$. The component b is the number of vehicle/vehicle collisions when vehicles fly straight paths from their initial position to their end position at any speed, $0 \leq b \leq n(n-1)$, $b \in \mathbb{N}$. The component c is the number of vehicle/obstacle collisions when vehicles fly straight paths from their initial position to their end position at any speed, $0 \leq c \leq nN_o$, $c \in \mathbb{N}$.

VII. RESULTS

Based on results of extensive random simulations we compared time complexity and optimality of the solution of the NLP approach presented in Section III and of the MILP approach presented in Section IV. A summary of the observation reported in this section is based on more than

10 random instance-sets (more than 2160 problem instances). We compared only problem instances that were feasible for both approaches. The number of infeasible problem instances was not significant compared to the total number of feasible instances. There are three relevant results we have obtained:

- 1) MILP was faster than the NLP for **all** problem instances we solved. As the problem size grows (in terms of number of vehicles and obstacles), the gap between the MILP solution time and the NLP solution time increases. Also the NLP reports much higher oscillations in solution times for larger problem instances, compared to the MILP solutions. NLP results are for the particular initialization in[7] and may be improved with other initializations.
- 2) MILP and NLP *always* provide optimal solutions with similar costs. We have computed the cost formulated for the NLP approach, for both MILP and NLP optimal trajectories. Then, we have defined a *relative cost error* as $\left| \frac{J(z_{MILP}^*) - J(z_{NLP}^*)}{J(z_{NLP}^*)} \right|$. where J is defined as

$$J(v_{x,1}, v_{y,1}, v_{z,1}, \dots, v_{x,n}, v_{y,n}, v_{z,n}) = \sum_{i=1}^n \int_{t_0}^{t_f} [\|v_x(t)\|_1 + \|v_y(t)\|_1 + \|v_z(t)\|_1] dt \quad (16)$$

and z_{MILP}^* and z_{NLP}^* are the MILP and NLP optimizers, respectively. The average relative cost error was always on the order of 10^{-1} , which implies a good performance by the NLP algorithm.

- 3) There is no correlation between the vector $[a, b, c]$ associated to a problem instance and the corresponding time complexity. We were expecting that for a fixed number of vehicles and obstacles the time complexity of a problem instance is related to parameters like density of the obstacles field or number of collision when vehicles fly along straight lines joining initial and final destination at any speed: this was not the case. We have found problem instances where the density of the field was small and where straight flight is a feasible solution which were more complex than problem instances with higher density of the obstacle field and where a straight flight would cause several collisions.

Next we report some data for one random instance-set with 216 problem instances, out of which 214 were feasible (for both the MILP and the NLP solvers). The MILP and NLP minimum solution times were 1.091 s and 1.422, respectively. The MILP and NLP maximum solution times were 18.20 s and 506.18 s, respectively. The MILP and NLP average solution times were 4.32 and 46.53, respectively. The relative maximum difference between the MILP and NLP optimal cost was 0.45%. The relative average difference between the MILP and NLP optimal cost was 0.22%. The data for all the groups of the instance-set are reported in Figure 1. Detailed data for some subgroups of the instance-set are reported in Tables I,II. Visualization of trajectories obtained for both approaches will be included in the final version of the paper.

Group Number	2-2	5-5	8-2	10-4	10-5
Feasible cases	6	6	6	6	6
Both infeasible cases	0	0	0	0	0
Min sol. time MILP	1.16	2.40	4.44	10.00	12.85
Min sol. time NLP	1.42	7.92	22.20	97.63	51.93
Max sol. time MILP	1.26	3.09	5.14	13.66	18.20
Max sol. time NLP	8.07	250.96	168.15	506.18	206.14
Ave. sol. time MILP	1.19	2.59	4.70	11.56	14.31
Ave. sol. time NLP	3.83	54.51	76.99	244.51	129.48
Max rel. cost error	0.45	0.30	0.22	0.23	0.21
Ave. rel. cost error	0.27	0.23	0.19	0.21	0.18

TABLE I
GROUPS COMPARISON

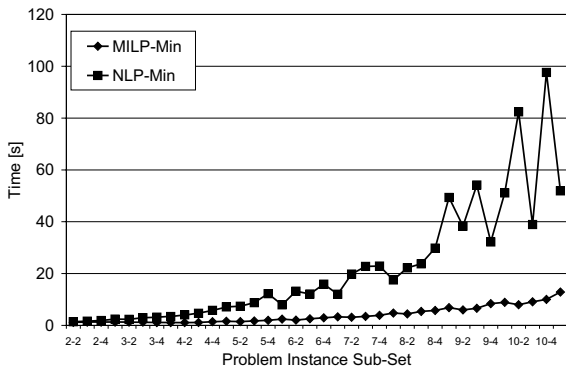
Group Number	2-2	5-5	8-2	10-4	10-5
NLP Instance					
# of variab.	1215	6450	9240	14250	15150
# of constr.	711	4515	6504	9930	10530
# of l. bounds	883	5150	7160	11650	12550
# of u. bounds	180	450	720	900	900
# of $\neq 0$ in Jacob.	2763	16755	25008	39360	41760
# of $\neq 0$ in Hess.	963	5850	10440	17550	18450
MILP Instance					
# of variab.	1626	8115	10824	18930	20730
of which binary	900	6300	7920	15300	17100
of which ≥ 0	180	450	720	900	900
of which ≤ 0	180	450	720	900	900
# of constr.	1422	8280	10728	19710	21810
of which equal.	372	930	1488	1860	1860
# of l. bounds	366	915	1464	1830	1830
# of u. bounds	366	915	1464	1830	1830
# of $\neq 0$ in constr.	3872	23430	33168	59460	64860
# of $\neq 0$ in object.	360	900	1440	1800	1800
# of $\neq 0$ in RHS	1062	7380	9288	17910	20010

TABLE II
SOLVER INSTANCES COMPARISON

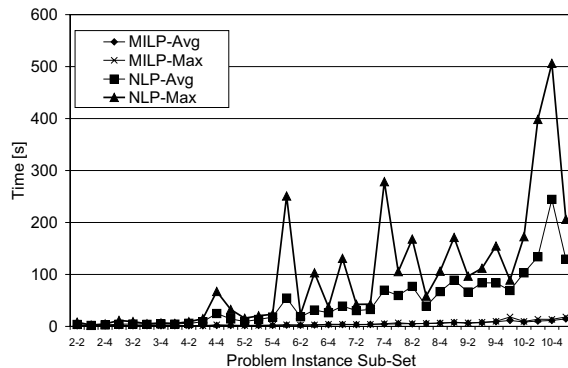
VIII. CONCLUSIONS

In this study, we have compared the time complexity and optimality of solutions obtained for multi-UAVs conflict resolution and obstacle avoidance with MILP and NLP approaches, based on the results of extensive random simulations. First, we have noticed that the MILP approach is always faster than the NLP approach, across the extensive set of random problem instances that we solved. In particular, when the optimal control problem becomes large, in terms of number of vehicles and obstacles (causing an explosion of combinatorics in the MILP case, and of non-convex constraints in the NLP case) the MILP approach outperforms the NLP approach. We used linear dynamical models for the UAVs. Conflict resolution algorithms that work with simple linear models may generate trajectories that are not necessarily flyable in practice. The issue of flyability of the optimal trajectories that are generated by any solution approach is critical to both safety and performance, and necessitates the use of more realistic and detailed dynamic models. Therefore, the first result of this paper opens the route to the use of MILP solutions (based on simple linear UAV models) in order to initialize NLP solvers which allow the use of more detailed dynamic UAV models. The second

objective of the study was to test the performance of the NLP solver presented in [20] in terms of optimality of the solution. In fact, while the NLP solver in [20] does not guarantee to converge to the global optimum, as opposed to the MILP approach, we notice that the costs of the solutions obtained with the MILP and the NLP algorithms are always comparable. Lastly, we have studied the relation between time complexity of the optimal solution (both MILP and NLP) and some physical parameters of the optimization problem such as density of the obstacle field or the number of collisions when the vehicles fly at any speed straight lines joining initial and final destination. Interestingly, we did not find any correlation between such physical parameters associated to a problem instance and its time complexity from our set of random problem instances. The results appear promising in the context of using linear MILP models, as well as for utilizing MILP-based solutions to provide a good initial guess to NLP formulations that can handle detailed dynamics.



(a) Minimum computational times as function of the group number



(b) Maximum and average computational times as function of the group number

Fig. 1. Comparison between the MILP and the NLP approach for a set of 216 random problem instances

REFERENCES

- [1] G. Bader and U. Ascher, A new basis implementation for mixed order boundary value ODE solver, *SIAM J. Sci. Comput.*, **8**(4): 483, 1987.
- [2] A.M. Cervantes and L.T. Biegler, A stable elemental decomposition for dynamic process optimization, *J. of Comp. and Appl. Math.*, 2001.
- [3] A.M. Cervantes, A.W. Wächter, R.H. Tütüncü and L.T. Biegler, A reduced space interior point strategy for optimization of differential algebraic systems, *Comp. and Chem. Eng.*, **24**(1): 39-51, 2000.
- [4] L.T. Biegler, A.M. Cervantes and A.W. Wächter, Advances in Simultaneous Strategies for Dynamic Process Optimization, *Chemical Engineering Science*, **57**(4):575-593, 2002
- [5] R. Fourer, D. Gay and B. Kernighan, *AMPL*, The Scientific Press, South San Francisco, California, 1993.
- [6] E. Frazzoli, Z.-H. Mao, J.-H. Oh and E. Feron, Resolution of conflicts involving many aircraft via semidefinite programming, submitted to *AIAA J. Guidance, Control and Dynamics*, 2001.
- [7] J. Hu, M. Pradini and S. Sastry, Three dimensional optimal coordinated maneuvers for aircraft conflict avoidance, *AIAA Journal of Guidance, Control and Dynamics*, vol. 25 (5), pp. 888-900, 2002.
- [8] M. Oishi, V. Gopal, C. Tomlin and D. Godbole, Addressing multi-objective control: safety and performance through constrained optimization, *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science 2034*, Springer, 459-472, 2001.
- [9] A.W. Wächter, *Interior Point Methods for Large-Scale Nonlinear Programming with Applications in Process Systems Engineering*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.
- [10] A.W. Wächter and L.T. Biegler, *On the Implementation of an Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*, *Mathematical Programming*, 106, 1, pp. 25-57, 2006.
- [11] Y. Zhao and R. Schultz, Deterministic resolution of two aircraft conflict in free flight, *AIAA Guidance Navigation and Control Conference*, New Orleans, LA, 1997.
- [12] J.K. Kuchar and L.C. Yang, A Review of Conflict Detection and Resolution Modeling Methods *IEEE Transactions On Intelligent Transportation Systems*, December, 2000.
- [13] T. S. Perry, In search of the future of air traffic control, *IEEE Spectrum*, vol. 34, pp. 18-35, August 1997.
- [14] P. K. Menon, G. D. Sweriduk, and B. Sridhar, Optimal strategies for free-flight air traffic conflict resolution, *Journ. of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 202-211, March-April 1999.
- [15] L. Pallottino, E. Feron and A. Bicchi, Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming, *IEEE Transactions On Intelligent Transportation Systems*, 3(1):3-11, March 2002.
- [16] C. Tomlin, G.J. Pappas, and S. Sastry, Conflict resolution for air traffic management: A case study in multi-agent hybrid systems, in *IEEE Trans. On Automatic Control*, vol. 43, no. 4, pp. 509-521, April 1998.
- [17] ILOG, Inc. *CPLEX 7.0 User Manual*. Gentilly Cedex, France, 2000.
- [18] A. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proc. American Contr. Conf.*, 2002.
- [19] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer, 2003.
- [20] A. Raghunathan, V. Gopal, D. Subramanian, L. Biegler, T. Samad, Dynamic Optimization Strategies for 3D Conflict Resolution of Multiple Aircraft, *AIAA J. of Guidance, Control and Dynamics*, 27 (4), pp. 586-594, 2004.
- [21] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407-427, March 1999.