2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Differential Flatness-Based Trajectory Planning for Multiple Unmanned Aerial Vehicles Using Mixed-Integer Linear Programming

Yongxing Hao*, Asad Davari†, Ali Manesh ‡
Department of Electrical and Computer Engineering
West Virginia University Institute of Technology
Montgomery, WV 25136, USA

March 1, 2005

Abstract

This paper provides a method for planning fuel-optimal trajectories for multiple unmanned aerial vehicles to reconfigure and traverse between goal points in a dynamic environment in real-time. Recent developments in robot motion planning have shown that trajectory optimization of linear vehicle systems including collision avoidance can be written as a linear program subject to mixed integer constraints, known as a mixed integer linear program (MILP). This paper extends the trajectory optimization to a class of nonlinear systems: differentially flat systems using MILP. A polynomial basis for a Ritz approximation of the optimal solution reduces the optimization variables and computation time without discretizing the systems. Based on the differential flatness property of unmanned vehicle systems, the trajectory planner satisfies the kinematic constraints of the individual vehicles while accounting for inter-vehicle collision and path constraints. The analytical fuel-optimal trajectories are smooth and continuous. Illustrative trajectory planning examples of multiple unmanned aerial vehicles are presented.

Keywords: Unmanned Vehicles, Differential Flatness, Trajectory Optimization, Linear Programming.

1 Introduction

Coordination and control of groups of autonomous unmanned aerial vehicles (UAVs) is emerging as a new and exciting field of research. Multi-UAV systems can provide redundancy and contribute cooperatively to solve the assigned task, or they may perform the assigned task in a more reliable, faster or cheaper way that is difficult or impossible to be accomplished by an individual vehicle.

*Research Assistant Professor, Ph.D., haoyx@ieee.org
†Professor, Ph.D., adavari@wvu.edu
‡American Science Technology

The goal in this paper is to develop a method for planning fuel-optimal trajectories for multiple unmanned aerial vehicles to reconfigure and traverse between goal points in a dynamic environment in real-time.

It is challenging for a trajectory generator to generate trajectories for a single vehicle in real-time satisfying system equations, operating and goal constraints, and minimizing cost criteria. The problem of coordination of systems in groups in a dynamic environment becomes even more challenging because of potential dynamic interactions and collisions between obstacles and members of the group.

The theory of dynamic optimization [1] provides a possible framework for the trajectory planning problem. However, this formulation often leads to problems that are very computationally intensive for on-line solutions. Another paradigm is provided by robotic path planning [2], where simplifying assumptions are made and some heuristic rules are used, so that a solution can be found in proper time. Despite its successful application to less complex systems, this latter approach may fail to comply with the system's dynamics, and perform poorly with respect to optimality.

A particular attempt to make dynamic optimization a viable tool for on-line optimal planning has been pursued in recent years. This effort takes place in the context of a recent advance in nonlinear systems theory: the study of differentially flat systems ([3], [4]). Dynamic optimization based on first-order and higher-order representations of a system have been compared showing substantial savings in computation [5]. Preliminary results of flatness-based planning of groups of autonomous vehicles were reported ([6], [7]).

By approximating the flat outputs using linear combinations of time-dependent functions [5], the dynamic optimization problem can be reduced into a common static optimization problem. Recent developments in robot motion planning have shown that trajectory optimization of linear vehicle systems including collision avoidance can be written as a linear program subject to mixed integer constraints, known as a mixed integer linear program (MILP) ([8], [9]). Some variables take only the values 0 or 1 which enable the inclusion of logical expressions in the optimization and global optimal solutions can be found. A key advantage of writing the trajectory optimization in the MILP form is the existence of highly-optimized, commercially available, efficient linear program softwares. So far, the MILP approach is based on discretized linear systems.

This paper extends the trajectory optimization to a class of nonlinear systems: differentially flat systems using MILP approach. Without discretizing the systems, the method described in this paper provides direct generation of fuel-optimal states and inputs trajectories which are smooth and continuous. A polynomial basis for a Ritz approximation of the optimal solution is formulated and collocation method is used to evaluate constraints at a finite number of points. Linear inequality constraints for a kinematic UAV model are derived. Based on the differential flatness property of unmanned vehicle systems, the trajectory planner satisfies the kinematic constraints of the individual vehicles while accounting for inter-vehicle collision and path constraints.

The outline of this paper is as follows: Section 2 presents the problem formulation, including flatness-based dynamic optimization, methods of solution, the UAV model and the fuel-optimal cost function. Linear inequality constraints are presented in Section 3. Illustrative examples of trajectory planning and optimization of multiple UAVs are shown in Section 4.

2 Problem Formulation

2.1 Flatness-based Dynamic Optimization

A differential system $\dot{x} = f(x,u)$, where x is a n-dimensional vector of states and u is a m-dimensional vector of control, is differentially flat if there exists variables y of the same dimension as controls, so that states and inputs can be algebraically expressed in terms of y and its higher-order derivatives: $[x,u] = F(y, y^{(1)}, ..., y^{(p)})$. The existence of a flat representation of the system implies that a dynamic optimization problem can be transformed to a problem in the calculus of variations with higher-order derivatives.

Thus, for a flat system, a fuel-optimal trajectory generation problem can be described by an unconstrained Lagrange problem with fixed final time and given terminal states:

$$\min J = \int_{t_0}^{t_f} L\left(y(t), y^{(1)}(t), ..., y^{(p)}(t), t\right) dt, \quad (1)$$

subject to

$$y(t_0) = y_0, y^{(1)}(t_0) = y_0^{(1)}, ..., y^{(p-1)}(t_0) = y_0^{(p-1)},$$
$$y(t_f) = y_f, y^{(1)}(t_f) = y_f^{(1)}, ..., y^{(p-1)}(t_f) = y_f^{(p-1)},$$
$$c(y(t), ...y^p(t), t) \leq 0.$$

2.2 Methods of Solution

A polynomial basis for a Ritz approximation of the optimal solution [5] can be chosen in the form

$$y = \varphi_0(t) + \sum_{i=1}^{n_\alpha} \alpha_i \varphi_i(t), \quad (2)$$
$$\varphi_{k-2p+l_k+1} = (t-t_0)^{p+l_k}(t-t_f)^{k-p-l_k},$$
$$d \geq k \geq 2p, k-2p \geq l_k \geq 0.$$

The α_i are unknown parameters. The function $\varphi_0(t)$ is a polynomial of degree $(2p-1)$ that satisfies the given terminal conditions.

In general, the inequality constraints in Eqn. (1) are nonlinear and the constraint region is nonconvex in multi-dimensional space. Determining the region requires solution of nonlinear equations, and in general such a solution is computationally expensive. To determine a real-time global optimal solution, an analytic linear solution of the feasible region is required. An example of such a region is the space within a polytope:

$$P = \{\bar{y} | a_{j0} + \sum_{k=1}^{n} a_{jk} y_k \leq 0, \quad j = 1...l.\} \quad (3)$$

Similarly, the cost function in Eqn. (1) also needs to be approximated by a linear one if it is nonlinear.

2.3 UAV Kinematic Model

Each UAV system is given a kinematic description using the well-known Frenet-Serret equations of motion. The fuel-optimal trajectories are derived by studying optimality criteria within the framework of our kinematic models. Assuming each UAV has some alternative method of determining the appropriate altitude, we reduce a three-dimensional model to a planar model. In this case, we allow for the possibility that the UAVs are flying at the same altitude, so that collision avoidance is a consideration.

The planar motion for UAV i is governed by the following equations

$$\dot{x}_i = v_i cos(\theta_i), \quad (4)$$
$$\dot{y}_i = v_i sin(\theta_i), \quad (5)$$
$$\dot{\theta}_i = \omega_i. \quad (6)$$

where (x_i, y_i) denotes the position of the vehicle in the inertial frame. θ_i denotes the orientation of the vehicle to the inertial frame. The inputs to the controller are v_i and ω_i which are, respectively, the translational speed and rotational speed of the vehicle. The control inputs are supplied to each UAV's on-board autopilot.

For a given $(x_i(t), y_i(t))$ trajectory of the vehicle, which are also the flat outputs of the system, the other states and inputs have the following expressions:

$$\theta_i = tan^{-1}\frac{\dot{y}_i}{\dot{x}_i}, \tag{7}$$

$$v_i = \dot{x}_i \cos\theta_i + \dot{y}_i \sin\theta_i, \tag{8}$$

$$\omega_i = \frac{\ddot{y}_i\dot{x}_i - \ddot{x}_i\dot{y}_i}{\dot{x}_i^2 + \dot{y}_i^2}. \tag{9}$$

2.4 Fuel-optimal Cost Function

The fuel-optimal cost function for multiple vehicles is

$$J = \sum_{i=1}^{K}\sum_{p=1}^{N}\sum_{j=1}^{n_u} r_{ij}|u_{ipj}| \tag{10}$$

where r_{ij} is the weight coefficient and u_{ipj} denotes the j^{th} component of the input vector of vehicle i at time step p. Thus the calculated trajectory is designed to minimize the fuel used for that specific time range.

This can be transformed into a linear form by introducing slack variables and additional constraints. By introducing the slack variables v_{ipj}, the optimization problem can be formulated as

$$\text{Min } J = \sum_{i=1}^{K}\sum_{p=1}^{N}\sum_{j=1}^{n_u} r_{ij} v_{ipj}, \tag{11}$$

$$\text{subject to} \quad u_{ipj} \leq v_{ipj}, \tag{12}$$

$$-u_{ipj} \leq v_{ipj}. \tag{13}$$

For the UAV systems (4)-(6), assuming no sharp turning, the rotational speed ω_i is negligible compared to the translational speed v_i. From Eqn. (4) and (5), the translational speed can be expressed as

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2. \tag{14}$$

Thus, minimizing the fuel consumption $|v_i|$ can be approximated by minimizing $|\dot{x}_i| + |\dot{y}_i|$. The cost function for the UAV trajectory optimization problem can be formulated as

$$\text{Min } J = \sum_{i=1}^{K}\sum_{p=1}^{N} v_{ip1} + v_{ip2}, \tag{15}$$

$$\text{subject to} \quad \dot{x}_{ip} \leq v_{ip1}, \tag{16}$$

$$-\dot{x}_{ip} \leq v_{ip1}, \tag{17}$$

$$\dot{y}_{ip} \leq v_{ip2}, \tag{18}$$

$$-\dot{y}_{ip} \leq v_{ip2}. \tag{19}$$

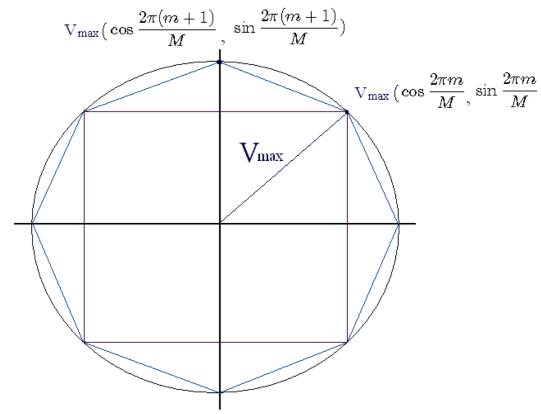

Figure 1: Approximations to translational magnitude limit. The circle is the feasible region for true magnitude limit. The square and polygon are two ways of approximating these regions with linear constraints.

If the constraints on states and inputs are linear, the resulting optimization problem is a linear program and can be solved using very efficient and highly optimized software packages.

3 Linear Inequality Constraints

3.1 Control Amplitude

The translational and rotational velocities acting upon the UAV are subject to magnitude constraints: $|v| \leq v_{max}$ and $|\omega| \leq \omega_{max}$. The exact representation of these constraints would be nonlinear, but they can be approximated by linear inequalities.

According to Eqn. (8), the true magnitude constraint of the translational velocity encloses a circle on the X-Y plane shown in Fig. 1. An arbitrary number (M) of constraints can be used to approximate the circle. These are given by

$$\begin{aligned} &\forall\ i \in [1...K]\ \ \forall\ p \in [1...N]\ \ \forall\ m \in [1...M] \\ &(\dot{y}_{ip} - v_{imax}\sin\frac{2\pi m}{M})(\cos\frac{2\pi(m+1)}{M} - \cos\frac{2\pi m}{M}) \\ &-(\dot{x}_{ip} - v_{imax}\cos\frac{2\pi m}{M})(\sin\frac{2\pi(m+1)}{M} - \sin\frac{2\pi m}{M}) \\ &\geq\ 0. \end{aligned} \tag{20}$$

The feasible region formed by four (M=4) or eight constraints (M=8) is shown in Fig. 1.

The linear inequality approximation of the rotational velocity of a single vehicle can be derived as follows

$$\omega_{max} \geq |\omega| = |\frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}| \ \geq\ |\frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{v_{max}^2}|,$$

and

$$\frac{|\ddot{y}\dot{x}-\ddot{x}\dot{y}|}{v_{max}^2} \leq \frac{|\ddot{y}\dot{x}|+|\ddot{x}\dot{y}|}{v_{max}^2} \leq \frac{|\ddot{y}v_{max}|+|\ddot{x}v_{max}|}{v_{max}^2} = \frac{|\ddot{y}|+|\ddot{x}|}{v_{max}},$$

We can choose

$$|\ddot{x}| \leq \frac{1}{2}w_{max}v_{max}, \quad |\ddot{y}| \leq \frac{1}{2}w_{max}v_{max}. \tag{21}$$

If Eqs. (21) are satisfied, the constraint on rotational velocity $|\omega| \leq \omega_{max}$ is satisfied.

The linear inequality approximation of the rotational velocity of multiple vehicles can be formulated as

$$\begin{aligned} &\forall\ i \in [1...K]\ \ \forall\ p \in [1...N] \\ &|\ddot{x}_{ip}| \leq \frac{1}{2}w_{imax}v_{imax}, \\ &|\ddot{y}_{ip}| \leq \frac{1}{2}w_{imax}v_{imax}. \end{aligned} \tag{22}$$

3.2 Obstacle Avoidance

In order to simplify the implementation, the obstacles are enlarged with the size of a single vehicle - allowing the vehicle to be treated as a point navigating through the planar space. Next, the obstacles are approximated by polygons. Let the position of one obstacle be denoted by the coordinates of its corner points (M) anticlockwise: (x_j, y_j). At each time step p, the position (x_p, y_p) of the vehicle must lie in the area outside of the obstacle. This can be formulated as

$$\begin{aligned} &\forall\ p \in [1...N] \\ &(y_p - y_1)(x_2 - x_1) - (x_p - x_1)(y_2 - y_1) \leq 0, \\ &... \\ \text{or}\ \ &(y_p - y_j)(x_{j+1} - x_j) - (x_p - x_j)(y_{j+1} - y_j) \leq 0, \\ &... \\ \text{or}\ \ &(y_p - y_M)(x_1 - x_M) - (x_p - x_M)(y_1 - y_M) \leq 0. \end{aligned}$$

By introducing binary slack variables and denoting $x_{M+1} = x_1$ and $y_{M+1} = y_1$, the above $or-$constraints can be replaced by the following mixed integer linear constraints

$$\forall\ p \in [1...N]\ \ \forall\ j \in [1...M]$$

$$(y_p - y_j)(x_{j+1} - x_j) - (x_p - x_j)(y_{j+1} - y_j) \leq Rb_{pj}, \tag{23}$$

$$\text{and} \quad \sum_{j=1}^{M} b_{pj} \leq M - 1. \tag{24}$$

where b_{pj} is a binary variable (0 or 1) and R is a positive arbitrarily large number. Eqn. (24) ensures that at least one of the original $or-$constraints is satisfied, which ensures that the vehicle avoids the polygon. The constraints (23-24) can be formulated for every obstacle situated in the maneuver space and for every vehicle in the multiple vehicle case. Eqs. (23)-(24) also work for moving obstacles with a predefined motion. In such situation, the obstacles change positions and new polygons can be defined at every time step, according to their predefined motion.

3.3 Vehicle Avoidance

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. Let the safety distances be denoted by d_x and d_y in the X and Y directions respectively. The constraints for collision avoidance between any two vehicles m and n can be written as

$$\forall\ p \in [1...N]$$

$$x_{mp} - x_{np} \geq d_x - Rc_{mnp1}, \tag{25}$$

$$x_{np} - x_{mp} \geq d_x - Rc_{mnp2}, \tag{26}$$

$$y_{mp} - y_{np} \geq d_y - Rc_{mnp3}, \tag{27}$$

$$y_{np} - y_{mp} \geq d_y - Rc_{mnp4}, \tag{28}$$

$$\sum_{k=1}^{4} c_{mnpk} \leq 3. \tag{29}$$

where c_{mnpk} is a binary variable (0 or 1) and R is a positive arbitrarily large number. If $c_{mnpk} = 0$, there is at least a safety distance apart between vehicles m and n in one direction at time step p. Eqn. (29) ensures that at least one c_{mnpk} is 0 at any time step, so there must always be safe separation in at least one direction.

4 Examples

4.1 Three UAVs in a Free Environment

The examples in this section demonstrate that the MILP approach can be applied to differentially flat UAV systems and generate fuel-optimal trajectories in real-time. LINGO optimization software [10] is used to solve the problem on a 3.2GHz Pentium IV PC with 1GB RAM.

Consider three vehicles moving through a free space and reconfiguring their formation. The initial and final states are shown in Table 1.

Table 1: Initial and final states for the three UAVs.

		Position (m)		Velocity (m/s)	
		X	Y	$\dot{X}$	$\dot{Y}$
UAV #1	*Initial*	3.2	1.5	1.0	0.5
	Final	13.7	4.5	2.0	0
UAV #2	*Initial*	3.2	4.5	1.0	– 0.5
	Final	13.7	1.5	2.0	0
UAV #3	*Initial*	3.2	7.8	1.0	– 1.0
	Final	13.7	– 1.5	2.0	0

The start and final times are chosen as $t_0 = 3s$ and $t_f = 12s$. The three UAVs avoid each other and reach their targets in 9 seconds. Each UAV's trajectory is divided into 30 points for collision and control amplitude check. The real vehicle has maximum translational speed $v_{max} = 5m/s$. Choosing the feasible region formed by four constraints in Fig. 1 at each point, the velocity limitations in x and y directions are $|\dot{x}| \leq 3.5m/s$ and $|\dot{y}| \leq 3.5m/s$. The acceleration limitations in x and y directions are chosen as $|\ddot{x}| \leq 2m/s^2$ and $|\ddot{y}| \leq 2m/s^2$. The safety distances between any two UAVs in x and y directions are $1.5m$.

Each vehicle's trajectory is described by two polynomials in x and y directions:

$$\begin{aligned} x_i &= a_{0i} + a_{1i}(t-t_0) + a_{2i}(t-t_0)(t_f-t) \\ &\quad + a_{3i}(t-t_0)^2(t_f-t) + a_{4i}(t-t_0)^2(t_f-t)^2, \\ y_i &= b_{0i} + b_{1i}(t-t_0) + b_{2i}(t-t_0)(t_f-t) \\ &\quad + b_{3i}(t-t_0)^2(t_f-t) + b_{4i}(t-t_0)^2(t_f-t)^2. \end{aligned}$$

This linear program has 2748 variables, 1080 integers and 1705 constraints. The optimization was solved in 5 seconds. The three UAVs' trajectories are plotted in Fig. 2. The circles mark the check points. The black dots mark the three UAVs' positions at time step #15. It is clear that the vehicles reach their targets and avoid each other during the maneuver. The velocities in x and y directions and the translational and rotational control inputs are plotted in Fig. 3. The absolute value of the rotational velocity ω is less than $60^o/s$. All states and control inputs trajectories are smooth and continuous.

The solved polynomial coefficients are listed in Table 2.

Table 2: Polynomial coefficients of the three UAVs.

UAV	a_0	a_1	a_2	a_3	a_4
#1	3.2	1.17	− 0.0185	− 0.0082	− 0.0018
#2	3.2	1.17	− 0.0185	− 0.0082	0.0050
#3	3.2	1.17	− 0.0185	− 0.0082	− 0.0063
	b_0	b_1	b_2	b_3	b_4
#1	1.5	0.333	0.0185	0.002	− 0.0027
#2	4.5	− 0.333	− 0.0185	− 0.002	− 0.0034
#3	7.8	− 1.033	0.0037	− 0.013	− 0.0023

4.2 Three UAVs in an Obstructed Environment

Consider three vehicles moving through a field of fixed obstacles and reconfiguring their formation as depicted in Fig. 4. The initial and final states are the same as before as shown in Table 1. Each UAV avoids all obstacles and other UAVs and reaches its target in 9 seconds. The solved polynomial coefficients are listed in Table 3.

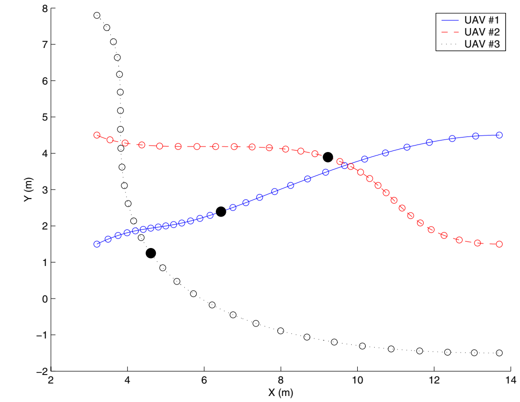

Figure 2: The planned trajectories for three UAVs in a free environment.

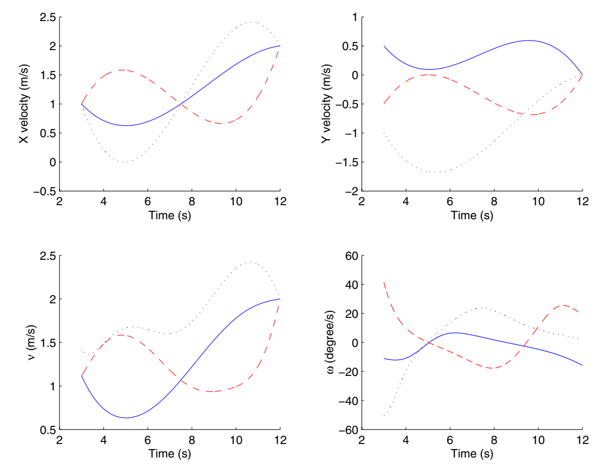

Figure 3: The velocities in x and y directions and the translational and rotational control inputs in a free environment.

The three UAVs' trajectories are plotted in Fig. 4. The velocities in x and y directions and the translational and rotational control inputs are plotted in Fig. 5.

Table 3: Polynomial coefficients of the three UAVs.

UAV	a_0	a_1	a_2	a_3	a_4
#1	3.2	1.17	− 0.0185	− 0.0082	0.0040
#2	3.2	1.17	− 0.0185	− 0.0082	− 0.0055
#3	3.2	1.17	− 0.0185	− 0.0082	− 0.0005
	b_0	b_1	b_2	b_3	b_4
#1	1.5	0.333	0.0185	0.002	− 0.0053
#2	4.5	− 0.333	− 0.0185	− 0.002	− 0.0047
#3	7.8	− 1.033	0.0037	− 0.013	− 0.0046

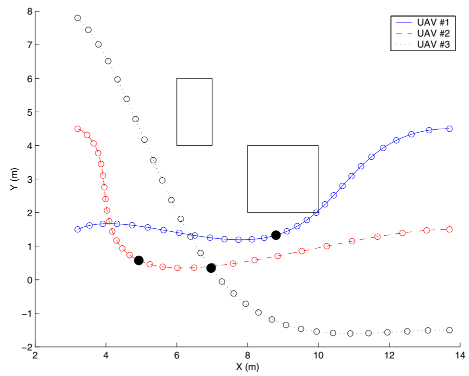

Figure 4: The planned trajectories for three UAVs in an obstructed environment.

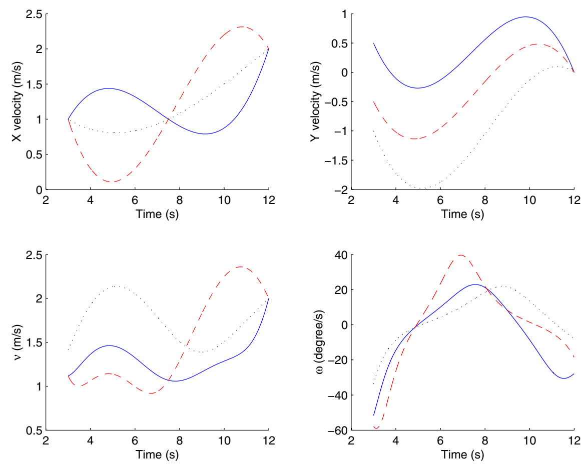

Figure 5: The velocities in x and y directions and the translational and rotational control inputs in an obstructed environment.

5 Conclusions

The requirements on the trajectory planner, namely the ability to generate trajectories in real-time satisfying system dynamic equations, operating and goal constraints, and minimizing cost criteria makes for a challenging computation problem. This paper extends the trajectory optimization to a class of nonlinear systems: differentially flat systems using MILP approach. Without discretizing the systems, the method described in this paper provides direct generation of fuel-optimal states and inputs trajectories which are smooth and continuous. A polynomial basis for a Ritz approximation of the optimal solution reduces the optimization variables and computation time. Collocation method is used to evaluate constraints at a finite number of points. Linear inequality constraints for a kinematic UAV model are derived. Based on the differential flatness property of unmanned vehicle systems, the trajectory planner satisfies the kinematic constraints of the individual vehicles while accounting for inter-vehicle collision and path constraints. Illustrative trajectory planning examples of multiple UAVs show promise of this approach.

Acknowledgments: We acknowledge the research support of DOD grant No. 8 SUB1166497RB.

References

[1] Bryson, A.E., Ho, Y.-C., *Applied Optimal Control*, John Wiley & Sons, 1975.

[2] Latombe, J.-C., *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[3] Fliess, M., Lévine, J., Martin, P., Rouchon, P., "Flatness and Defect of Non-linear Systems: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, N. 6, pp 1327-1361, 1995.

[4] Sira-Ramírez, H. and Agrawal, S.K., *Differentially Flat Systems*, Marcel Dekker Publishers, 2004.

[5] Veeraklaew, T. and Agrawal, S. K., "A New Computational Framework for Trajectory Optimization of Higher-order Dynamic Systems," *AAIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, pp 228-236, 2001.

[6] Hao, Y., Laxton, B., Agrawal, S.K., Lee, E. and Benson, E., "Planning and Control of UGV Formations in a Dynamic Environment: A Practical Framework with Experiments," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[7] Hao, Y., Laxton, B., Benson, E. R. and Agrawal, S. K., "Differential Flatness-Based Formation Following of a Simulated Autonomous Small Grain Harvesting System," *Transactions of the ASAE*, vol. 47(3), 933-941, 2004.

[8] Schouwenasrs, T., DeMoor, B., Feron, E. and How, J., "Mixed Integer Programming for Multi-Vehicle Path Planning," *Proceedings of the European Control Conference*, Portugal, 2001.

[9] Richards, A., How, J., Schouwenaars, T. and Feron, E., "Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, VA, 2001.

[10] *LINGO User's guide*, LINDO, 1999.