Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003

Hybrid System Design for Formations of Autonomous Vehicles

Shannon Zelinski
Aerospace Operations Modelling
NASA Ames Research Center
Moffett Field, CA 94035
szelinski@mail.arc.nasa.gov

T. John Koo
Department of EECS
Vanderbilt University
Nashville, TN 37212
john.koo@vanderbilt.edu

Shankar Sastry
Department of EECS
University of California
Berkeley, CA 94720
sastry@eecs.berkeley.edu

Abstract

Cooperative control of multiple unmanned aerial vehicles (UAVs) poses significant theoretical and technical challenges. Recent advances in sensing, communication and computation enable the conduct of cooperative multiple-UAV missions deemed impossible in the recent past. We are interested in solving the Formation Reconfiguration Planning (FRP) problem which is focused on determining a nominal state and input trajectory for each vehicle such that the group can start from the given initial configuration and reach its given final configuration at the specified time while satisfying a set of given inter- and intra- vehicle constraints. Each solution of a FRP problem represents a distinct reconfiguration mode. When coupled with formation keeping modes, they can form a hybrid automaton of formation maneuvers in which a transition from one formation maneuver to another formation maneuver is governed by a finite automaton. This paper focuses on the implementation of the optimized hybrid system approach to formation reconfiguration for a group of 1 real and 3 virtual UAVs. Experimental results performed in the Richmond Field Station by using a helicopter-based Berkeley Aerial Robot are presented.

1 Motivation

Cooperative control of multiple unmanned aerial vehicles (UAVs) poses significant theoretical and technical challenges. Recent years have seen the emergence of formation planning and control for UAVs as a topic of great interest. Much work has been done on the stabilization of groups of vehicles for keeping formation [1, 2, 3, 4]. While it is important that formations of UAVs be able to follow global trajectories while maintaining the group formation, it is also important for these groups of UAVs to be able to reconfigure when prompted without crashing into one another.

The most basic need of formation reconfiguration is obstacle avoidance. If the current formation of a group of vehicles is unable to avoid an upcoming obstacle without breaking formation, the group must reconfigure in order to avoid the obstacle. After the obstacle has been avoided, the formation may reconfigure again back to the original formation. Another application of formation reconfiguration is for a group of vehicles switching between two tasks that are suited to different formations. Formation reconfiguration could also be used in Air Traffic Management to reduce separation requirements and increase airspace capacity. Any group of vehicles in close proximity may be considered a formation.

In this paper, we propose a hybrid structure to enable effective switching between formation keeping modes and formation reconfiguration modes, while the trajectory generation for formation reconfiguration is formulated as an optimization problem with both inter- and intra-vehicle constraints. The main advantage of this approach is that on-line formation reconfiguration can now be performed more efficiently and safely since all possible next feasible formations from a current formation can be searched on a discrete graph and the execution is specified by a hybrid automaton in which each transition is guarded by both the discrete command and the continuous state. Furthermore, all the computationally intensive tasks which involve optimization are performed off-line.

In the next section, we will demonstrate how to generate a nominal state trajectory for a group of UAVs by minimizing a given cost function while satisfying a set of inter- and intra- vehicle constraints. Since there is a rich set of literature focusing on generating trajectories for formation keeping, we will omit the related discussion in this paper but assume all the necessary trajectories are designed properly. Then, we will propose a hybrid structure for formation keeping and reconfiguration so that the formation task can be performed efficiently and safely. Finally, we will show how these methods are implemented in a group of 1 real and 3 virtual UAVs. Experimental results performed in the Richmond Field Station by using a helicopter-based Berkeley Aerial Robot will be presented.

2 Trajectory Generation for Formation Reconfiguration

In studying the formation reconfiguration problem for a group of autonomous vehicles, we first consider if the problem is feasible assuming that all information is accessible. The *Formation Reconfiguration Planning* (FRP) feasibility problem addressed in this paper is:

Problem 2.1 *Given a group of autonomous vehicles, an initial configuration, a final configuration, a set of inter- and intra- vehicle constraints, and a time for reconfiguration determine a nominal state and input trajectory for each vehicle such that the group can start from the initial configuration and reach its final configuration at the specified time while satisfying the set of inter- and intra- vehicle constraints.*

For the feasibility of the FRP problems, please refer to [5] for details. In this paper, we focus on solving the FRP problem for a specific class of systems and a particular form of input signals so that we can represent the problem as an optimization problem that can be solved more efficiently especially for a large group of vehicles. For simplicity, a point mass model is used to model trajectory dynamics of each autonomous vehicle. Although this simple dynamical model is used, the results can be naturally extended to systems that can be feedback linearized [6] such as high-performance aircraft[7, 8] and helicopters[9]. Heterogeneities among vehicles can also be considered by incorporating additional constraints which can be used to specify maximal and minimal velocities or maximal turn rates of classes of vehicles.

Consider a group of autonomous vehicles each with the following dynamics:

$$\begin{bmatrix} \dot{p}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ a_i(t) \end{bmatrix} \tag{1}$$

where $p_i, v_i, a_i \in \mathbb{R}^3$ and $i = 1, \cdots, N$. Define $x_i(t) = [p_i^T(t)\ v_i^T(t)]^T$, $u_i(t) = a_i(t)$. Hence, (1) can be written as $\dot{x_i}(t) = A_i x_i(t) + B_i u_i(t)$ with

$$A_i = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \; B_i = \begin{bmatrix} 0_{3\times3} \\ I_{3\times3} \end{bmatrix}.$$

Denoting the group state as $x = [x_1^T, \cdots, x_N^T]^T$ and the group input as $u = [u_1^T, \cdots, u_N^T]^T$, the group dynamics can be written as $\dot{x}(t) = Ax(t) + Bu(t)$ where $A \in \mathbb{R}^{nN\times nN}$ and $B \in \mathbb{R}^{nN\times nM}$. Define the group configuration at time t as $g(t) = [x^T(t), u^T(t)]^T$ which specifies the state and input conditions for all the vehicles in the group at time t. Assume that all the inter- and intra-vehicle constraints are specified as a set of group state constraints $c_j(x(t)) \leq \beta_j$ for $j = 1, \cdots, M$. The admissible input for each vehicle is specified by an input constraint $b_i(u_i(t)) \leq \alpha_i$.

For example, the group state constraint for minimum separation can be specified as:

$$\|p_i(t) - p_j(t)\| \;\geq\; \epsilon \; \forall t \in [0, T], \; \forall_{i\neq j} i, j \tag{2}$$

where $i, j = 1, \cdots, N$, ϵ is the minimum allowable distance between vehicles. The input constraint of maximum acceleration can be specified as:

$$\|a_i(t)\| \;\leq\; \alpha, \; \forall t \in [0, T], \; \forall i \tag{3}$$

where α is the maximum allowable acceleration input at any time.

If the input trajectories and the initial conditions are provided, the state trajectories of each vehicle can be derived according to the vehicle dynamics. Point mass vehicle dynamics are written as follows.

$$v_i(t) = \int_0^t a_i(t)dt + v_i(0) \tag{4}$$

$$p_i(t) = \int_0^t v_i(t)dt + p_i(0) \tag{5}$$

The input signals are parameterized as polynomials of time. $a_i(t)$ is the acceleration vector of the i^{th} vehicle represented as:

$$a_i(t) = \sum_{k=0}^{K} a_{ki} t^k \tag{6}$$

where K is the order of the polynomial, and $a_{0i} \cdots a_{Ki}$ are the parameter vectors for the i^{th} vehicle. As with $a_i(t)$, all of the a_{ki} parameter vectors are in $\mathbb{R}^3$. In order to obtain feasible solutions for the FRP problem by considering only the dynamical and configuration constrains, $K \geq 4$. Again, see [5] for details.

The FRP problem can now be formulated as an optimization problem. The objective is to determine the parameters for the input trajectories, minimizing the cost function, subject to dynamical, configuration, inter- and intra- vehicle constraints.

The optimization problem used to derive the trajectory parameters that minimize the cost J defined by the total input energy consumed by a group of point masses subject to the minimum separation constraints is

$$\min_{a_{01}\cdots a_{KN}} \; J = \sum_{i=1}^{N} \frac{1}{T} \int_0^T a_i^T(t) W_i a_i(t) dt \tag{7}$$

subject to

$$\begin{bmatrix} \dot{p}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ a_i(t) \end{bmatrix} \tag{8}$$

$$g(0) = g_s \tag{9}$$

$$g(T) = g_f \tag{10}$$

$$\|p_i(t) - p_j(t)\| \geq \epsilon \; \forall t \in [0, T], \; \forall_{i\neq j} i, j \tag{11}$$

where W_i is a diagonal matrix of weighting constants for the i^{th} vehicle. Thus, distinct acceleration characteristics of vehicles can be properly specified by the weighting matrix W.

Figure 1 shows the trajectory solutions of 4 reconfigurations for 4 UAVs designed for experimental implementation. The polynomial order $K = 4$, the weighting matrix W is an identity matrix, the total time $T = 30$, and the minimum separation $\epsilon = 10$ is the only inter-vehicle constraint used. The large rectangle represents the space within which the real UAV (UAV1) must remain for safety with the group center specified at (65,-57.5). Virtual UAVs may go outside this boundary as seen with UAV4.

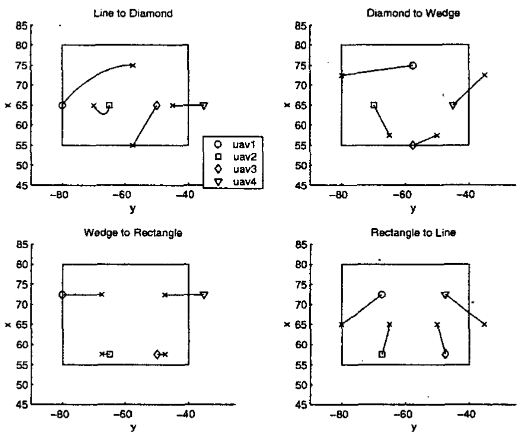

Figure 1: Four reconfiguration paths between four possible formations.

3 Hybrid Structure for Formation Keeping and Reconfiguration

Consider the four formation configurations shown in figure 1: line, diamond, wedge, and rectangle. In Section 2, we have shown how to generate reconfiguration trajectories from various pairs of initial and final formation configurations by solving the corresponding FRP problems. Since at any time, the operation mode of the group can be either formation keeping or formation reconfiguration, if we associate each mode of operation with the corresponding closed-loop dynamics, we can represent the group dynamics by the hybrid automaton[10] which is defined as a collection $H = (Q \times X, \Sigma, f, I, G)$ where $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$, $X \subseteq \mathbb{R}^{nN}$, $\Sigma = \{\sigma_1, \sigma_3, \sigma_5, \sigma_7\}$, f is the vector field, I assigns an invariant set to each q, and G assigns a guard to each edge. Modes q_1, q_3, q_5, q_7 represent the group dynamics in keeping formation in Line mode, Diamond mode, Wedge mode, and Rectangle mode, respectively. Each q_j for $j \in \{2,4,6,8\}$ represents the group dynamics in reconfiguration from mode q_{j-1} to mode q_{j+1} where $q_9 = q_1$. The hybrid state and input are denoted by $(q, x) \in Q \times X$ and $\sigma \in \Sigma$, respectively.

We assume that in each mode q_j, there is a desired group trajectory for the group to track and is denoted as $r_j(t) \in \mathbb{R}^{nN}$ for $j = 1, \ldots, 8$, where $r_j = [r_{j1}^T, \cdots, r_{jN}^T]^T$, and there is a trajectory tracking controller $u_i(t) = k_i(x(t), r_j(t))$ for each vehicle i for $i = 1, \ldots, N$. Depending on the sensor types, the communication protocols and the controller designs, the amount of information and the types of information flow needed for tracking the desired group trajectory may vary. Here, we just use a design of controller, which uses only its state trajectory and desired state trajectory, to illustrate the idea of the hybrid structure proposed for implementation. Given an admissible trajectory, we assume that each controller can keep the vehicle state trajectory staying with an open set provided that the vehicle state is started properly in another open set. Hence, the vector field is defined by $f(q, x, \sigma) = Ax + Bu$ with each $u_i(t) = k_i(x(t), r_j(t))$ for $i = 1, \ldots, N$. For $j \in \{2,4,6,8\}$, the invariants of q_j and q_{j+1} are defined by $B_{j1} \times \cdots \times B_{jN} \times \{\sigma_{j+1}\}$ and $B_{j+1\ 1} \times \cdots \times B_{j+1\ N} \times \{\sigma_{j+1}\}$, respectively, where B_{ji} is an open ball in $\mathbb{R}^n$ with center located at r_{ji} and radius ϵ_{ji}. For $j \in \{2,4,6,8\}$, the guards of (q_{j-1}, q_j) and (q_j, q_{j+1}) are defined by $B_{j1} \times \cdots \times B_{jN} \times \{\sigma_{j+1}\}$ and $B_{j+1\ 1} \times \cdots \times B_{j+1\ N} \times \{\sigma_{j+1}\}$. For example, when the group is in a Line keeping mode, q_1, the group state x and input σ must satisfy the invariant condition $I(q_1) = B_{11} \times \cdots \times B_{1N} \times \{\sigma_1\}$. If the input becomes σ_3 and $x \in B_{21} \times \cdots \times B_{2N}$, the group switches to the q_2 reconfiguration mode. While $\sigma = \sigma_3$, when the group state satisfies $B_{31} \times \cdots \times B_{3N}$, then the group transitions to Diamond keeping mode, q_3. Notice that each trajectory in a formation reconfiguration mode is by construction connected to an adjacent formation keeping mode and hence the transitions are possible with proper input signals. The resulting hybrid automaton is shown in Figure 2.

Although the hybrid automaton is useful for analysis, for implementing the design the hybrid automaton has to be refined by showing the interconnections between system components. In order to refine the hybrid automaton, we define the following signals $\delta_i \in \{\delta_{1i}, \ldots, \delta_{8i}\}$ for $i \in \{1, \ldots, N\}$ and $\delta_i = \delta_{ji}$ if $x_i \in B_{ji}$. The signals are used for indicating that $x_i \in B_{ji}$ and hence can be used for synchronizing among vehicles. For example, if the system is in mode q_1 and $\delta_1 = \delta_{21} \wedge \cdots \wedge \delta_N = \delta_{2N} \wedge \sigma = \sigma_2$, switching between mode q_1 and mode q_2 is feasible. The refined hybrid system contains N vehicles and each vehicle consists of a hybrid controller and a vehicle plant model. In each hybrid controller, there are N modes which each corresponds to a formation mode. In Figure 3, only vehicle 1 with three modes, namely q_1, q_2, q_3, is shown in detail.

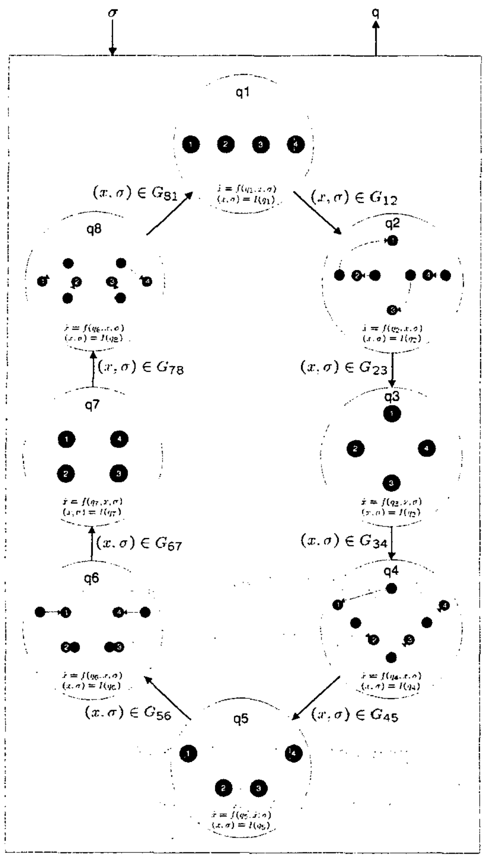

Figure 2: Hybrid automaton modelling of a group of vehicles in formation.

In order to make the decoupled hybrid automaton shown in Figure 3 resemble the hybrid automaton shown in Figure 2, it requires a strong synchronous assumption made on communication capabilities of the vehicles for exchanging events. On the other hand, the decoupled hybrid automaton allows further refinement by using asynchronous communication between components or by adding components for modelling sensors and actuators.

4 Experimental Results

The hybrid system method of formation reconfiguration was implemented on the Berkeley Aerial Robot (BEAR) platform. The BEAR helicopter is a highly versatile aerial platform which took years of work to develop. As shown in [9], a helicopter model is differentially flat, i.e. the state trajectory and nominal input trajectory can be reconstructed from the outputs and their derivatives. If the outputs are p_x, p_y, p_z, ψ, the Euler angles can be reconstructed by using $\ddot{p}_x, \ddot{p}_y, \ddot{p}_z$.

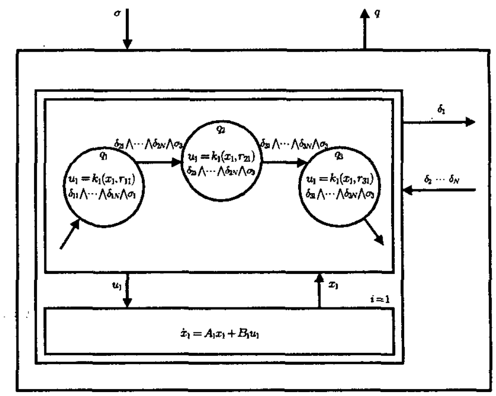

Figure 3: Component-based hybrid system design for multiple autonomous vehicles in formation.

When hovering, flight control takes an active role in stabilizing the helicopter about a given set point such that it emulates a point mass at low speeds which enables the use of equation 1 in the higher level path planning. However, when using the point mass model, we must still take enormous precautions at the path planning level, being careful to stay well within system limits. If one of the helicopters were to crash, it would be very dangerous and destroy years of hard work. Vertical movements in the z direction are extremely unstable, especially when ascending when there is danger of motor stalling. This is why all desired trajectories have been restricted to the x-y plane maintaining a constant elevation. Additionally, the desired trajectories are scaled with a timescale τ such that the fastest moving trajectory does not exceed $0.5\frac{m}{s^2}$ of acceleration and $1.0\frac{m}{s}$ of velocity.

Three systems must interact with one another in a hierarchy of control. The low level control is responsible for the stability of the helicopter about a desired set point. The next level, which we will call path planning, is responsible for providing the trajectory of an individual helicopter. At the highest level, a group controller decides how and when to reconfigure the group.

This experiment reconfigures one real BEAR helicopter in formation with three virtual helicopters. The real helicopter's navigation computer is responsible for low level control while the virtual navigation computers emulate both this low level control and the point mass dynamics. The control for the virtual navigation of the point mass dynamics is given as

$$a_i(t) = 2\eta\omega_n v_i(t) - \omega_n^2 \left(p_i(t) - p_{id}(t)\right) \qquad (12)$$

where $\eta = .7$ and $\omega_n = 10\pi$.

Both one real and three virtual path planners provide nominal trajectory set points calculated from on board position parameters to the low level controllers. These parameters are provided by the group controller which performs all centralized optimization and user interface. The group controller is also responsible for determining the trajectory of the group center so that the group as a whole may be moving during reconfiguration. However, due to space restrictions, the desired group center remains stationary throughout the sequence of reconfigurations in this experiment. The group controller also calculates the timescale τ which scales the trajectory solutions such that the fastest vehicle acceleration and velocity are less than or equal to the given maximum allowed acceleration a_{max} and velocity v_{max}.

First τ_a is calculated such that the maximum acceleration of all four trajectory solutions $\max a_{ki}$ is equal to the given maximum allowed acceleration a_{max}.

$$\tau_a = \frac{a_{max}}{\max a_{ki}} \tag{13}$$

Similarly, τ_v is calculated such that the maximum velocity of all four trajectories $\max v_{ki}$ is equal to the given maximum allowable velocity v_{max}.

$$\tau_v = \frac{v_{max}}{\max v_{ki}} \tag{14}$$

The used timescale τ is the lesser of τ_a and τ_v to ensure that $\max a_{ki} \leq a_{max}$ and $\max v_{ki} \leq v_{max}$.

The path planers use τ along with the stored parameters calculated using $T = 30$ to stretch or shrink the actual time required to complete the reconfiguration. The path planner evaluates each desired position set point p_{di} to be delivered to it's respective navigation computer at time t_e using position parameters $p_{0i} \cdots p_{6i}$ converted from it's stored parameters $a_{0i} \cdots a_{4i}$ as follows.

$$t_e = (t - t_0 + \Delta t)\tau \tag{15}$$

$$p_{di}(t_e) = \sum_{k=0}^{6} p_{ki} t_e^k \tag{16}$$

where τ is the timescale such that the reconfiguration time remains within $0 \leq t_e \leq T$, t_0 is the time at the start of the reconfiguration, and Δt is the set point delivery period. When $t_e \geq T$, the end of the trajectory has been reached and p_{di} will be evaluated at T.

Figure 4 shows the experimental setup. Vehicle one is a real BEAR helicopter which communicates serially to it's on board path planning computer via RS-232. All virtual path planning controllers are housed on a single lap top computer which communicate with their virtual helicopters and group controller via wireless TCP/IP connection. The group controller and virtual helicopters are housed on a single lap top computer used as the ground station. This does not in any way limit the degree of realism in system communication as the ground station and virtual helicopter do not communicate with one another directly.

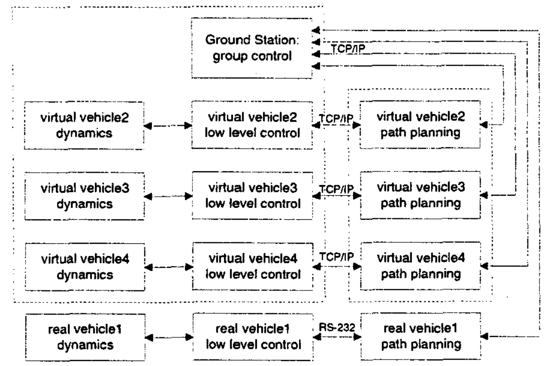

Figure 4: Controller housing and communication setup for a thee virtual and one real vehicle group.

Figures 5 and 6 show the results of a sequence of four reconfigurations between four formations from line to diamond to wedge to rectangle back to line.

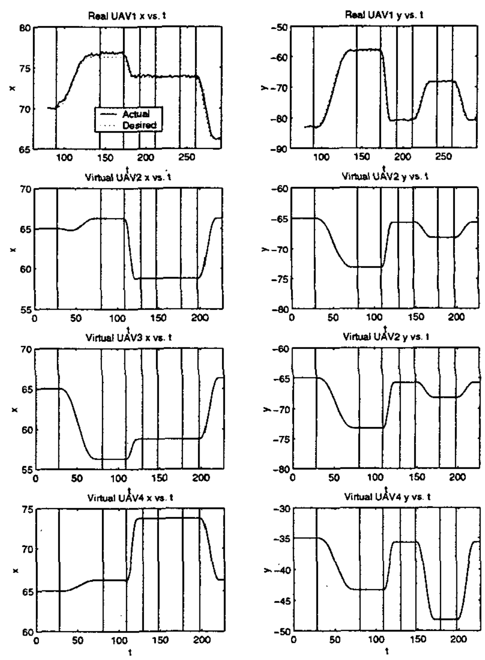

Figure 5: Experimental results reconfiguring 1 real and 3 virtual UAVs between 4 possible formations.

Figure 5 shows the desired and actual trajectories of the 1 real and 3 virtual helicopters. Notice how the virtual helicopter trajectory error is so small, the actual and desired trajectories shown are indistinguishable while the real helicopter had to deal with the strong winds of Richmond Field Station in order to follow the desired trajectory. The vertical lines show the moments of switching between formation keeping and formation reconfiguration modes.

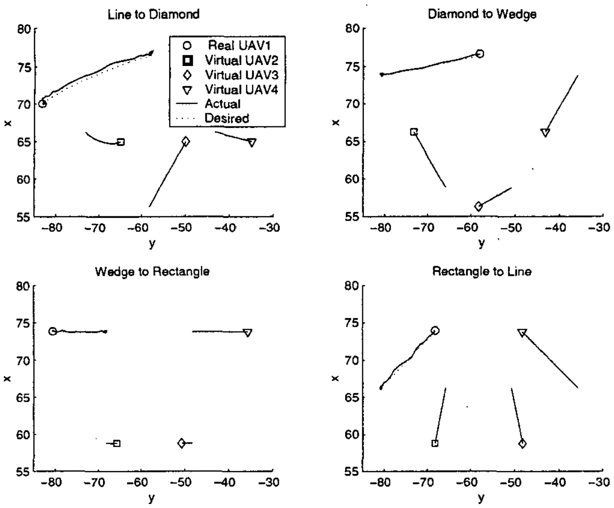

Figure 6: XY position of 1 real and 3 virtual UAVs during a reconfiguration experiment between 4 possible formations.

Figure 6 shows the desired and actual trajectories in xy space for each reconfiguration. Once again, desired and actual virtual helicopter trajectories are indistinguishable. By comparing the desired and actual trajectories of the real helicopter, one can tell that the wind was blowing strongly in the northwesterly direction with the position x axis oriented north.

5 Conclusion

In this paper, we proposed a solution to formation reconfiguration of multiple autonomous vehicles. Given a pair of initial and final formation configurations, the completion time, and a set of inter- and intra- vehicle constraints, we can find a nominal input trajectory for each vehicle such that the group can start from the given initial configuration and reach its given final configuration at the specified time while satisfying a set of given inter- and intra- vehicle constraints. By limiting the class of trajectories to polynomials, we can calculate the trajectories efficiently off line and store the parameters in reconfiguration libraries for each vehicle. Each set of reconfiguration parameters represents a hybrid reconfiguration mode. When coupled with formation keeping modes, they can form a hybrid system network of reconfiguration maneuvers. We implemented just such a hybrid system network with a series of 4 reconfigurations in a primitive information flow control scheme. We plan to further explore other information flows and assess their strengths and weaknesses when applied to the hybrid reconfiguration network. Special attention will be paid to the asynchronous characteristic of information structure.

Acknowledgement

The authors would like to thank H. Chung, D. Liccardo, O. Shakernia, C. Sharp, T. Pynn and P. Ray at Berkeley for technical assistance and to thank people at the Institute of Software Integrated Systems at Vanderbilt University and Aerospace Operation Modelling Division at NASA Ames Research Center for valuable comments. This work is supported by the DARPA SEC grant, F33615-98-C-3614.

References

[1] J. A. Fax, R. M. Murray. "Graph Laplacians and Stabilization of Vehicle Formations", CDS Technical Report 01-007. In *Proceedings of IFAC World Congress*, Barcelona, Spain, July 2002.

[2] A. Pant, P. Seiler, M. Broucke, T.J. Koo, J.K. Hedric. "Coordination and Control of Autonomous Vehicles". Submitted to *IEEE Transactions on Robotics and Automation*, March 2001.

[3] P. Tabuada, G. J. Pappas, P. Lima. "Feasible Formations of Multi-Agent Systems". In *Proceedings of the American Control Conference*, pp.56-61, Arlington, VA, June 2001.

[4] N. E. Leonard, E. Fiorelli, "Virtual Leaders, Artificial Potentials and Coordinated Control of Groups". In *Proceedings of the IEEE Conference on Decision and Control*, pp.2968-2973, Orlando, FL, December 2001.

[5] S. Zelinski, T. J. Koo, A. Sastry. Optimization-based Formation Reconfiguration Planning For Autonomous Vehicles. In *Proceedings of International Conference on Robotics and Automation*, Taipei, Taiwan, May 2003.

[6] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, 1995.

[7] R. Su, L.R. Hunt, G. Meyer. *Theory of design using nonlinear transformations.* In *Proceedings of the 1982 American Control Conference*, Vol. (1), pp. 247-251, 1982.

[8] C. Tomlin, J. Lygeros, L. Benvenuti, S. Sastry. *Output Tracking for a Non-Minimum Phase Dynamic CTOL Aircraft Model.* In *Proceedings of the 1995 IEEE Conference in Decision and Control*, pp. 1867-1872, Kobe, Japan, December 1996.

[9] T. J. Koo, S. Sastry. Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization. In *Proceedings of the 37th Conference on Decision and Control*, pp.3635-40, Tampa, Florida, December 1998.

[10] R. Alur and T.A. Henzinger. Modularity for timed and hybrid systems. In *Proceedings of the Eighth International Conference on Concurrency Theory (CONCUR)*, pages 74-88, 1997.