A Smooth Hybrid Symbolic Control for the Formation of UAVs over a Partitioned Space*

Ali Karimoddini[1], Hai Lin[2], Ben. M. Chen[3], and Tong Heng Lee[4]

***Abstract*— This paper presents and implements a smooth hybrid supervisory control mechanism for the formation of unmanned helicopters. A polar partitioning scheme is utilized to bisimilarly abstract the motion space to find a finite state model for the motion dynamics of UAVs. To implement this algorithm, a hierarchical control structure is introduced which uses a discrete supervisor on the top layer that is connected to the regulation layer via an interface layer. The implantation issues of the proposed algorithm are investigated and a control mechanism is introduced to smoothly transit over the partitioned space without any jump on the control signals while preserving the bisimulation relation between the abstract model and the original continuous system. Actual flight test results are presented to verify the algorithm and the control structure performance.**

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) can achieve a formation when they jointly move with a relatively fixed distance [1], [2]. This capability enhances the manoeuvrability of the team of UAVs to cooperatively accomplish different missions such as search and rescue in hazardous environments, aerial mapping and SLAM, area coverage and mutual defense. The formation problem usually consists of several subtasks. Starting from an initial state, the UAVs should achieve the desired formation within a finite time (*reaching the formation*). Then, they should be able to maintain the achieved formation, while the whole structure needs to track a certain trajectory (*keeping the formation*). Meanwhile, in all of the previous steps, the collision between the agents should be prevented (*inter-collision avoidance*).

In the literature there are some methods that can partly address the formation problem. For example, in [3], [4], [5], the problem of *reaching the formation* is investigated using optimal control techniques, navigation function, and potential field approaches. *Keeping the formation* can be seen as a standard control problem in which the system's actual position has slightly deviated from the desired position for which many control approaches have been developed such as feedback control, rigid graph, and virtual structure [6], [7]. Finally, in [8], [9], and [10], different mechanisms for *collision avoidance* have been introduced using probabilistic methods, MILP programming, and behavioral control. Nevertheless, there is still a lack of a unified solution to address the whole process starting from reaching formation, maintaining formation while avoiding collision. Furthermore, it is required to consider a decision making unit to apply discrete supervisory rules and switching logic to manage this multi-task structure. This suggests us to bring the formation problem to the context of hybrid modelling and control theory [11], [12] by which we can capture both the discrete and continuous dynamics of the system. In our recent study [13], a unified hybrid supervisory control framework was introduced to address all aspects of a formation control mission. The approach is based on the polar abstraction of the motion space and utilizing the properties of multi-affine functions over the partitioned space. This abstraction technique converts the original continuous system with infinite states into a finite state machine for which one can use the well developed theory of supervisory control of discrete event systems (DES) [14]. Due to the proven bisimulation relation, the abstracted system can behave as the same as the original system so that the discrete supervisor, designed for the discrete finite model, can be applied to the original system.

In this paper, we focus on the implementation issues of the proposed hybrid control algorithm. The main contribution of this paper lies in presenting a smooth hybrid supervisory control algorithm for the formation of unmanned helicopters. Using this control mechanism, the UAVs smoothly transit through the partitioning elements so that there is no jump in the generated control signal when the system transits from one region to its adjacent regions. The basic idea is to tune the value of the vector field at the vertices of the partitioning elements at the common edges to provide a smooth control signal. It is shown that this control mechanism preserves the bisimulation relation between the abstract model and the original continuous system so that there is no need to redesign the discrete supervisor. Secondly, an interface layer is introduced to connect the discrete supervisor layer to the continuous plant. This interface layer is responsible for converting the continuous signals of the plant into some symbols understandable by the discrete supervisor, and vice versa. Finally, a cooperative testbed is developed, and the proposed formation control algorithm has been verified through actual flight tests.

The rest of this paper is organized as follows. The pre-

[1]A. Karimoddini is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, USA. He is also with the Graduate School for Integrative Sciences and Engineering (NGS), National University of Singapore.

[2]H. Lin is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, USA.

[4]T. H. Lee is with Graduate School for Integrative Sciences and Engineering (NGS) and the Department of Electrical and Computer Engineering (ECE), National University of Singapore, Singapore.

[3]B. M. Chen is with Graduate School for Integrative Sciences and Engineering (NGS) and the Department of Electrical and Computer Engineering (ECE), National University of Singapore, Singapore.

*Corresponding author: H. Lin, hlin1@nd.edu, Tel. 574-6313177.

liminaries of the hybrid formation control algorithm are described in Section II. The control hierarchy including the discrete supervision layer, the interface layer, and the continuous low layer is discussed in Section III. Section IV describes implementation issues for the algorithm. Actual flight test results are presented in Section V, and the paper is concluded in Section VI.

II. Preliminaries on Hybrid Formation Control

For the implementation of the proposed hybrid formation algorithm we have used a set of two UAV helicopters, HeLion and SheLion which are developed by our research group at the National University of Singapore. The modelling and low level control structure of the NUS UAV helicopters are explained in [15], [16], [17]. For the regulation layer of these helicopters we have proposed a two-layer control structure in which the inner-loop controller stabilizes the system using H_∞ control design techniques, and their outer-loop is used to derive the system towards the desired location. As it has been discussed in [16], in this control structure, the inner-loop is fast enough to track the given references, so that the outer-loop dynamics can be approximately described as follows:

$$\dot{x} = u,\ x \in \mathbb{R}^2,\ \ u \in U \subseteq \mathbb{R}^2, \tag{1}$$

where x is the position of the UAV; u is the UAV velocity reference generated by the formation algorithm, and U is the velocity constraint set, which is a convex set.

Now, in a leader follower formation scenario, consider the follower velocity in the following form:

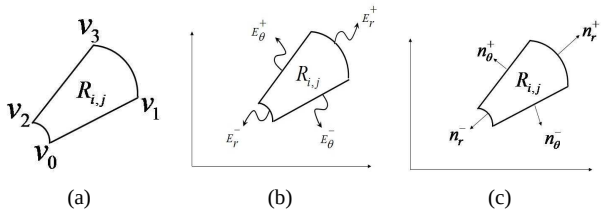
$$V_{follower} = V_{leader} + V_{rel}. \tag{2}$$

For these helicopters, our aim is to design the formation controller to generate the relative velocity of the follower, V_{rel}, such that starting from any initial point inside the control horizon, it eventually reaches the desired relative distance with respect to the leader, while avoiding the collision between the leader and the follower. Moreover, after reaching the formation, the follower UAV should remain at the desired position.

To solve this problem, in [13], a method is introduced for the polar abstraction of the state space which uses the properties of multi-affine vector fields over the polar partitioned space. Within this framework, a DES model can be achieved for which we can design a decentralized supervisor to achieve three major goals: *reaching the formation, keeping the formation,* and *collision avoidance*. This method is briefly explained in the following section.

A. Polar partitioning of the state space

Consider a relatively fixed frame, in which the follower moves with the velocity of V_{rel} and the leader has a relatively fixed position. In this framework, imagine a circle with the radius of R_m that is centered at the desired position of the follower. With the aid of the partitioning curves $\{r_i = \frac{R_m}{n_r-1}(i-1),\ i = 1,...,n_r\}$ and $\{\theta_j = \frac{2\pi}{n_\theta-1}(j-1),\ j = 1,...,n_\theta\}$, this circle can be partitioned into $(n_r-1)(n_\theta-1)$ partitioning elements. An element $R_{i,j} = \{p = (r,\theta) | \, r_i \le r \le r_{i+1},\, \theta_j \le \theta \le \theta_{j+1}\}$ has four vertices, v_0, v_1, v_2, v_3 (Fig. 1(a)), four edges, E_r^+, E_r^-, E_θ^+, E_θ^- (Fig. 1(b)), and correspondingly, four outer normal vectors n_{r}^+, n_{r}^-, n_θ^+, n_θ^- (Fig. 1(c)). In region $R_{i,j}$, the notation $E_{p,q}$ is used for the edge which is incident with the vertices v_p and v_q, and correspondingly, $n_{p,q}$ is used to denote its outer normal vector.



(a) (b) (c)

Fig. 1. (a) Vertices of the element $R_{i,j}$. (b) Edges of the element $R_{i,j}$. (c) Outer normals of the element $R_{i,j}$.

To implement the formation algorithm, we will deploy multi-affine functions over the partitioned space. A multi-affine function $f : \mathbb{R}^n \to \mathbb{R}^m$, has the property that for any $1 \le i \le n$ and any $a_1, a_2 \ge 0$ with $a_1 + a_2 = 1$, $f(x_1,...,(a_1x_{i_1} + a_2x_{i_2}),x_{i+1},...x_n) = a_1f(x_1,...,x_{i_1},x_{i+1},...x_n) + a_2f(x_1,...,x_{i_2},x_{i+1},...x_n)$. The following proposition shows that the value of a multi-affine function over the partitioning element $R_{i,j}$, can be uniquely expressed in terms of the values of the function at the vertices of $R_{i,j}$.

Proposition 1: [13] Consider a multi-affine function $g(x) : \mathbb{R}^2 \to \mathbb{R}^2$ over the region $R_{i,j}$. The following property always holds true:

$$\forall x = (r,\theta) \in R_{i,j} :\ g(x) = \sum_{m=0}^{3} \lambda_m g(v_m) \tag{3}$$

where λ_m, $m = 0,...,3$, are obtained as follows:

$$\lambda_m = \lambda_r^{\Psi_r(v_m)}(1-\lambda_r)^{1-\Psi_r(v_m)}\lambda_\theta^{\Psi_\theta(v_m)}(1-\lambda_\theta)^{1-\Psi_\theta(v_m)} \tag{4}$$

where $\lambda_{\mathrm{r}} = \frac{r-r_i}{r_{i+1}-r_i}$, $\lambda_\theta = \frac{\theta-\theta_j}{\theta_{j+1}-\theta_j}$, $\Psi_r(v_m) = \begin{cases} 0 & m = 0,2 \\ 1 & m = 1,3 \end{cases}$ and $\Psi_\theta(v_m) = \begin{cases} 0 & m = 0,1 \\ 1 & m = 2,3 \end{cases}$.

Remark 1: It can be verified that the resulting coefficients λ_m, $m = 0,1,2,3$, have the property that $\lambda_m \ge 0$ and $\sum_m \lambda_m = 1$.

The above proposition holds true for the edges as described in the following corollary.

Corollary 1: For a multi-affine function $g(x)$ defined over the element $R_{i,j}$ and for all edges E_q^s of $R_{i,j}$, $q \in \{r,\theta\}$ and $s \in \{+,-\}$, the following property holds true:

$$\forall x = (r,\theta) \in E_q^s :\ g(x) = \sum_{v_m \in V(\,E_q^s)} \lambda_m g(v_m), \tag{5}$$

where λ_m can be obtained as follows:

- For edges E_r^+ and E_r^- : $\lambda_m = \lambda_\theta^{\Psi_\theta(t)}(1-\lambda_\theta)^{1-\Psi_\theta(t)}$

- For edges E_θ^+ and E_θ^- : $\lambda_m = \lambda_r^{\Psi_r(u)}(1-\lambda_r)^{1-\Psi_r(u)}$

Next, the properties of multi-affine functions are utilized to form a hierarchical hybrid structure for the control of unmanned helicopters to achieve a desired formation.

III. Hierarchical Control Structure for the Formation of Unmanned Helicopters

For the above discussed model of the plant defined over the partitioned space, we will design a discrete supervisor which pushes the system trajectories to pass through the desired regions to achieve the desired behaviour. The designed discrete supervisor cannot be directly connected to the continuous plant. Hence, it is required to construct an interface layer which can translate continuous signals of the plant to a sequence of discrete symbols understandable for the supervisor. Also, the interface layer is responsible for converting discrete commands received from the supervisor, to continuous control inputs to be given to the plant. These two jobs are respectively realized by the blocks Detector and Actuator embedded in the interface layer as it is shown in Fig. 2. The elements of this control hierarchy are discussed in the following parts.

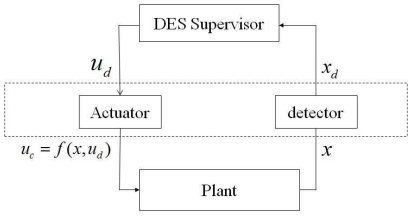


Fig. 2. Linking the discrete supervisor to the plant via an interface layer.

A. The interface layer

1) The detector block: When the system's trajectory crosses the boundaries of the region, a detection event will be generated which informs the supervisor that the system has entered a new region.

More specifically, a detection event $d_{i,j}$ will happen at $t(d_{i,j})$ when the system's trajectory $x(t)$ satisfies the following conditions:

- $\exists \tau > 0$ such that $x(t) \notin R_{i,j}$ for $t \in (t(d_{i,j})-\tau, t(d_{i,j}))$
- $\exists \tau_d > 0$ such that $x(t) \in R_{i,j}$ for $t \in [t(d_{i,j}), t(d_{i,j}) + \tau_d)$

Also, if the leader position is on the way of the follower towards the desired position, the event Ob will be generated to inform the supervisor about the risk of collision.

2) The actuator block: Having the information about the newly entered region, the supervisor can issue a discrete command to push the system trajectory to move towards the desired region. However, the discrete symbols generated by the supervisor need to be translated to a continuous form. For such a purpose, we utilize the properties of multi-affine functions by which we can design continuous controllers that drive the system's trajectory to either stay in the current region for ever or exit from one of its edges.

An invariant region can be defined as follows:

Definition 1: (**Invariant region**) In the circle C_{R_m} and the vector field $\dot{x} = g(x)\ ,\ g : \mathbb{R}^2 \to \mathbb{R}^2$, the region $R_{i,j}$ is said to be invariant region, if $\forall x(0) \in int(R_{i,j})$, and $x(t) \in R_{i,j}\ for\ t \geq 0$.

The following theorem and corollary show how we can construct an invariant region:

Theorem 1: Given a continuous multi-affine vector field $\dot{x} = g(x),\ g : \mathbb{R}^2 \to \mathbb{R}^2$, defined over the region $R_{i,j}$, the systems trajectory cannot leave the region through the edge $E_{p,q}$ with the outer normal $n_{p,q}$ if $n_{p,q}(y)^T.g(v_m) < 0$, for all $v_m \in \{v_p, v_q\}$ and all $y \in E_{p,q}$.

Proof: According to Corollary 1, $\forall x \in E_{p,q}$: $g(x) = \sum_{v_m} \lambda_m g(v_m)\ ,\ v_m \in \{v_p, v_q\}$. Substituting this value of $g(x)$ we will have $n_{p,q}(y)^T.g(x) = n_{p,q}(y)^T.\sum_{v_m} \lambda_m g(v_m) = \sum_{v_m} \lambda_m\ n_{p,q}(y)^T.g(v_m)$. Since, $n_{p,q}(y)^T.g(v_m) < 0$ for both $v_m = v_p$ and $v_m = v_q$ and all $y \in E_{p,q}$, and since $\lambda_m \geq 0$ and $\sum_m \lambda_{m\in\{p,q\}} = 1$, it can be concluded that $n_{p,q}(y)^T.g(x) < 0$ for all $x, y \in E_{p,q}$, which means that the trajectories of the system cannot leave $R_{i,j}$ through the edge $E_{p,q}$.■

Corollary 2: (**Sufficient condition for $R_{i,j}$ to be an invariant region**) For a continuous multi-affine vector field $\dot{x} = h(x, u(x)) = g(x),\ h : \mathbb{R}^2 \to \mathbb{R}^2$, $R_{i,j}$ is an invariant region if there exists a controller $u : \mathbb{R}^2 \to U \subseteq \mathbb{R}^2$, such that for each vertex v_m, $m = 0, 1, 2, 3$, with incident edges $E_q^s \in E(v_m)$, and corresponding outer normals n_q^s, $q \in \{r, \theta\}$ and $s \in \{+, -\}$:

$$U_m = U \cap \{u \in \mathbb{R}^2 |\, n_q^s(y)^T.g(v_m) < 0,\ for\, all\, E_q^s \in E(v_m),\ and\ \ for\, all\, y \in E_q^s\} \neq \emptyset, \tag{6}$$

where the convex set U represents the velocity bounds.

Proof: If (6) holds true, since $U_m \neq \emptyset$, there exists $u_m \in U_m$, $m = 0, 1, 2, 3$, such that based on Theorem 2, the value of the vector field at the vertices does not let the trajectory of the system leave the region from any of the edges. ■

Definition 2: (**Exit edge**)
In the circle C_{R_m} and the vector field $\dot{x} = g(x)\ ,\ g : \mathbb{R}^2 \to \mathbb{R}^2$, the edge $E_q^s,\ q \in \{r, \theta\}$ and $s \in \{+, -\}$, is said to be an exit edge, if $\forall x(0) \in int(R_{i,j})$, there exist $\tau\ (finite) > 0$ and $\tau_d > 0$ satisfying:

1) $x(t) \in int(R_{i,j}) \qquad for\ t \in [0, \tau)$
2) $x(t) \in E_q^s \qquad for\, t = \tau$
3) $x(t) \notin R_{i,j} \qquad for\ t \in (\tau, \tau + \tau_d)$

The following theorem shows the way that we can construct an exit edge:

Theorem 2: (**Sufficient condition for an exit edge**) For a continuous multi-affine vector field $\dot{x} = h(x, u(x)) = g(x),\ g : \mathbb{R}^2 \to \mathbb{R}^2$, E_q^s with the outer normal n_q^s, $q \in \{r, \theta\}$ and $s \in \{+, -\}$, is an exit edge if there exists a controller $u : \mathbb{R}^2 \to U \subseteq \mathbb{R}^2$, such that for each vertex v_m, $m =$

$0, 1, 2, 3$, the following property holds true:

$$U_m = U \bigcap \{u \in \mathbb{R}^2 |\ n_q^s(y)^T . g(v_m) > 0,\ for\, all\, v_m\ and\ for\, all\, y \in E_q^s\} \cap \{u \in \mathbb{R}^2 |\ n_{q'}^{s'}(y)^T . g(v_m) < 0,\ \ for\ all\ E_{q'}^{s'} \neq E_q^s\ and\ for\, all\, y \in E_{q'}^{s'},\ v_m \in V(E_{q'}^{s'})\} \neq \emptyset, \tag{7}$$

where the convex set U represents the velocity bounds.

Proof: Since $U_m \neq \emptyset$, there exists $u_m \in U_m$, such that we have $n_{q'}^{s'}(y)^T . g(v_m) < 0$, for all $E_{q'}^{s'} \neq E_q^s$ and all $y \in E_{q'}^{s'}$. Therefore, based on Theorem 1, the trajectories of the system do not leave $R_{i,j}$ through the non-exit edges. On the other hand, we have $n_q^s(y)^T . g(v_m) > 0$ for all v_m and all $y \in E_q^s$. According to Proposition 1, for the multi-affine function g, there exist λ_m such that $\forall x \in R_{i,j}$: $g(x) = \sum_m \lambda_m g(v_m),\ m = 0, 1, 2, 3$. Since $\lambda_m \geq 0$ and $\sum_m \lambda_m = 1$, then $n_q^s(y)^T . \lambda_m g(v_m) > 0$ for all v_m and all $y \in E_q^s$. This will lead to have $n_q^s(y)^T . g(x) > 0$ for all $x \in \bar{R}_{i,j}$, which means that the trajectories of the system have a strictly positive velocity in the direction of n_q^s steering them to exit from $R_{i,j}$ through the edge E_q^s. ■

Solving the inequalities given in Theorem 2 and Corollary 2, for the system dynamics given in (1), the following control values at the vertices of the region $R_{i,j}$ can make it an invariant region or can make one of its edges an exit edge. For the invariant controller, the control label is C_0 and the control values at the vertices are:

$$\begin{cases} u(v_0) = 1\angle(\theta_j + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_1) = 1\angle(\theta_j + \pi - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_2) = 1\angle(\theta_{j+1} - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_3) = 1\angle(\theta_{j+1} + \pi + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \end{cases}$$

To have the edge E_r^+ as the exit edge, the control label is C_r^+ and the control values at the vertices are:

$$\begin{cases} u(v_0) = u(v_1) = 1\angle(\theta_j + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_2) = u(v_3) = 1\angle(\theta_{j+1} - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \end{cases}$$

To have the edge E_r^- as the exit edge, the control label is C_r^- and the control values at the vertices are:

$$\begin{cases} u(v_0) = u(v_1) = 1\angle(\theta_j + \pi - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_2) = u(v_3) = 1\angle(\theta_{j+1} + \pi + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \end{cases}$$

To have the edge E_θ^+ as the exit edge, the control label is C_θ^+ and the control values at the vertices are:

$$\begin{cases} u(v_0) = 1\angle(\theta_j + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_1) = 1\angle(\theta_j + \pi - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_2) = 1\angle(\theta_{j+1} + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_3) = 1\angle(\theta_{j+1} + \pi - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \end{cases}$$

To have the edge E_θ^- as the exit edge, the control label is C_θ^- and the control values at the vertices are:

$$\begin{cases} u(v_0) = 1\angle(\theta_j - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_1) = 1\angle(\theta_j + \pi + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_2) = 1\angle(\theta_{j+1} - 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \\ u(v_3) = 1\angle(\theta_{j+1} + \pi + 0.5 \mid \theta_j - \theta_{j+1} + \frac{\pi}{2}\ |) \end{cases}$$

Now, the responsibility of the actuator is to relate the discrete symbol $u_d \in \{C_0, C_r^-, C_r^+, C_\theta^+, C_\theta^-\}$ to the continuous control signal $u_c(x)$. Using the properties of multi-affine functions as described in Proposition 1, the control signal can be constructed as $u_c(x) = f(x, u_d) = \sum_{m=0}^{3} \lambda_m(x) u(v_m)$, where $u(v_m), m = 0, ..., 3$, are the control values at the vertices corresponding to the control label u_d.

B. The supervisor layer

Using these control labels, a discrete supervisor is designed for a follower UAV involved in a formation mission. In this supervisor, shown in Fig. 3, when a detection event $d_{i,j}$ appears, the supervisor will be informed that the system has entered the new region $R_{i,j}$. If the detection event is $d_{1,j}$, it means that the system has entered the first circle of the partitioned space and the formation is achieved. Hence, to keep the formation, the system should remain in this region for the rest of the mission. In this case, keeping the formation can be done by activating the controller C_0. If the trajectory has not reached one of the partitions in the first circle ($i > 1$), then the event C_r^- should be activated to move towards the origin. Meanwhile if the leader is on the way of the follower towards the origin, the event Ob will be generated which alarms the supervisor about the collision. To avoid the collision, it is sufficient to drive the follower's path to turn anticlockwise, and then resume the mission. Hence, after observing the event Ob, the supervisor activates the event C_θ^+.

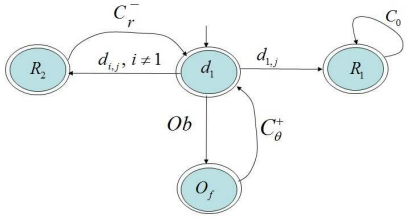


Fig. 3. The formation supervisor.

IV. Implementation Issues

A. Smooth Control

When the system trajectory enters a new region, a new discrete command will be generated. This may cause the discontinuity in the generated control signal to be applied to the lower levels of the control structure.

For example, Fig. 4 shows a case that the control command C_r^- has pushed the system's trajectory to transit from the region R_1 to the region R_2. After reaching the region R_2, the control command has changed from C_r^- to C_θ^+. Since the generated continuous control signal is a multi-affine function, based on Corollary 1, the control value at any point on the edges is determined by the control values at its vertices. In this example, $u(v_0(R_1)) = u(v_1(R_2))$ but $u(v_2(R_1)) \neq u(v_3(R_2))$. Since, the control values at the vertices of the common edge between R_1 and R_2 changes, there is a jump on the generated continuous control signal. Next theorem shows how we can resolve this problem.

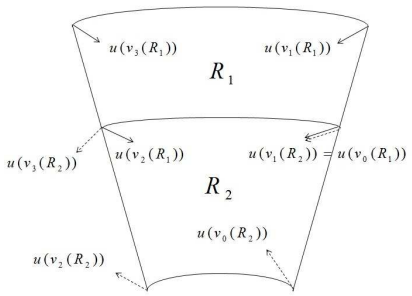


Fig. 4. The control values at the vertices when the system trajectory transits from region R_1 to region R_2 and the discrete command changes from C_r^- to C_θ^+.

Theorem 3: Let the command C_q^s steers the system's trajectory from the region $R_{i,j}$ to the region $R_{i',j'}$ and then, the supervisor issues the new command $C_{q'}^{s'}$. For this transition, the multi-affine controller $u(x) = \sum_{v_m \in V_c} \lambda_m \hbar(u(v_m)_{new}, u(v_m)_{old}) + \sum_{v_m \in V_n} \lambda_m (u(v_m))$ provides a smooth control signal, and drives all the system's trajectories to exit from the exit edge $E_{q'}^{s'}$.

Here λ_m, $m = 0, 1, 2, 3$, are given in Proposition 1, V_n is the set of vertices whose control values do not change due to the transition, and V_c is the set of vertices whose control values change after the system's trajectory enters the region $R_{i',j'}$. For these vertices, $u(v_m)_{old}$ and $u(v_m)_{new}$ are the control values at the vertex v_m before and after transiting to $R_{i',j'}$, respectively. The function $\hbar$ provides a smooth rotation from $u(v_m)_{old}$ to $u(v_m)_{new}$ and it can be presented as $\hbar(u(v_m)_{new}, u(v_m)_{old}) = \begin{cases} r_m \angle(\frac{t}{\Delta t}\theta_{m_{new}} + (1 - \frac{t}{\Delta t})\theta_{m_{old}}) & for\, t < \Delta t \\ r_m \angle \theta_{m_{new}} & for\, t \geq \Delta t \end{cases}$

where $u(v_m)_{new} = r_m \angle \theta_{m_{new}}$, $u(v_m)_{old} = r_m \angle \theta_{m_{old}}$. Also, Δt is the transition time.and then, resume

Proof: Let $C_q^s = C_r^-$ and $C_{q'}^{s'} = C_\theta^+$. As shown in Fig. 4, for this sequence of control commands, after transiting from $R_{i,j}$ to $R_{i',j'}$, the control value at the vertex v_3 changes from $u(v_3)_{old}$ to $u(v_3)_{new}$, and for the other vertices v_m, $m = 0, 1, 2$, there is no jump on the control value.

From the definition of the transition rule, $\hbar$, since for the whole transition time, the control values at the vertices satisfy the conditions of Corollary 2, the system's trajectory cannot leave the region through the non-exit edges $E_{0,2}$, $E_{0,1}$, $E_{1,3}$. Also, at the beginning of the transition mode, the control values at the vertex v_3 does not satisfy the conditions of Theorem 2, and hence, it cannot be concluded that the system's trajectory leaves the region through $E_{2,3}$. But, at some time, $u(v_3)$ will eventually reach $u(v_3)_{new}$, and the configuration of the vector field at the vertices will satisfy the conditions of Theorem 2 so that it can be guaranteed that the system's trajectory surely leaves the region $R_{i',j'}$ through the edge $E_{2,3}$, while there is no jump at the value of the control signal due to the smooth transition of the control values at the vertices. The same reasoning can be done for the other sequences of the control commands.■

Remark 2: In [13], it was shown that the polar abstracted model is bisimialr to the original system meaning that for any transition in the abstracted model, there is a transition in the original system and vice versa. From Theorem 3, it can be immediately concluded that the result is also valid for the case of smooth transition mechanism. This is due to the fact that based on Theorem 3, all of the trajectories finally will leave the region through the desired exit edge and the smooth transition mechanism does not let the system's trajectories exit from non-exit edges, leading to the following corollary:

Corollary 3: The smooth transition mechanism introduced in Theorem 3 preserves the bisimilarity relation between the abstracted model and the original hybrid system.

V. Implementation Results

To verify the algorithm, we have conducted a flight test in which the leader tracks a line path, and the follower should reach and keep the formation. In this test, the control horizon R_m is 50 meter, $n_r = 10$, and $n_\theta = 20$. The follower is initially located at a point which has a relative distance of $(dx, dy) = (-17.8, 11.4)$ with respect to the desired position and the distance between the desired position and the leader is $(dx, dy) = (-5, -15)$ as shown in Fig. 5.

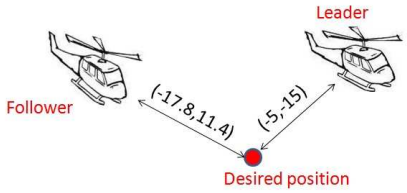


Fig. 5. The schematic of the scenario with for a leader-follower case tracking a line.

The position of the UAVs in x-y plane is shown in Fig. 6. The follower state variables and control signals are shown in Fig. 7 and Fig. 8, respectively. The relative distance of the follower UAV from the desired position is shown in Fig. 9. As it can be seen the follower UAV has finally reached the first circle after 17 sec and then, it has been able to maintain the formation. A video of this experiment is available at http://uav.ece.nus.edu.sg/video/hybridformation.mpg.

VI. Conclusion

In this paper a smooth hybrid supervisory control mechanism was proposed for the formation of unmanned helicopters. Using the polar partitioning of the motion space and utilizing the properties of multi-affine functions over the partitioned space, a finite state model was achieved which bisimulates the UAV motion dynamics and was used to design a discrete supervisor to satisfy the formation specification. To implement the algorithm, an interface layer was introduced which connects the discrete supervisor to the regulation layer of the UAV. This interface layer is composed of two main blocks: the detection block to generate the detection events based on the plant continuous signals; and the actuator block to convert discrete commands of the

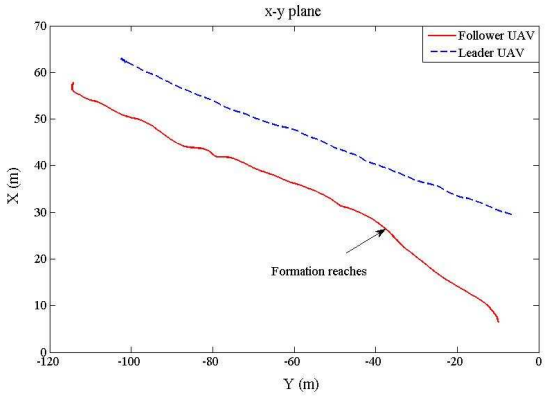


Fig. 6. The position of the UAVs in the x-y plane.

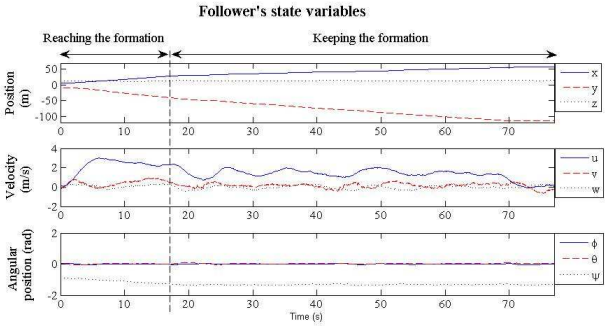


Fig. 7. The state variables of the follower.

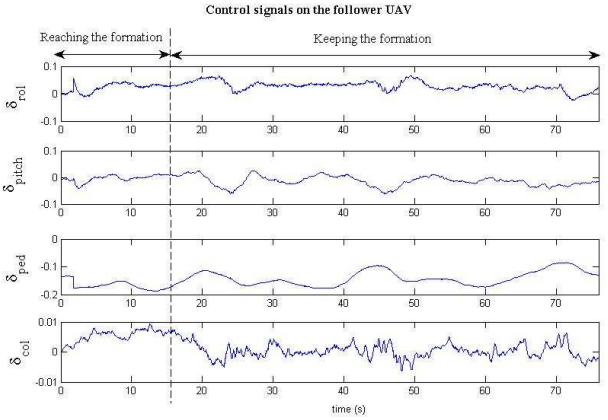


Fig. 8. Control signals of the follower UAV.

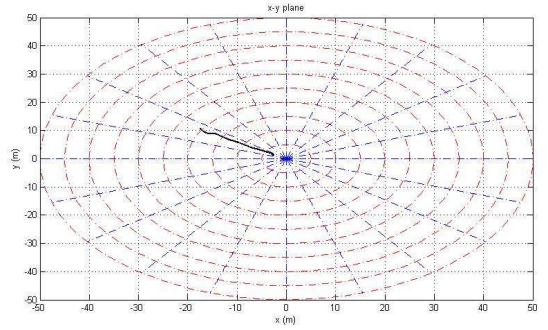


Fig. 9. The distance of the follower from the desired position.

supervisor to a continuous form, applicable to the plant. For the actuator block, a smooth control mechanism was introduced to avoid the jumps when the system transits over the partitioned space. The effectiveness of the algorithm was verified through the actual flight test results.

ACKNOWLEDGMENT

The financial supports from NSF-CNS-1239222 and NSF-EECS-1253488 for this work are greatly acknowledged. Also, The authors would like to thank Dr. Xiangxu Dong, Dr. Guowei Cai, and Dr. Feng Lin for their technical support during the implementations and flight tests.

REFERENCES

[1] R. Fierro, P. Song, A. Das, V. Kumar, Cooperative control of robot formations, in: R. Murphey, P. M. Pardalos, P. M. Pardalos, D. W. Hearn (Eds.), Cooperative Control and Optimization, Vol. 66, Springer, 2002, pp. 73–93.

[2] G. Lee, N. Y. Chong, Decentralized formation control for small-scale robot teams with anonymity, Mechatronics 19 (1) (2009) 85–105.

[3] J. How, E. King, Y. Kuwata, Flight demonstrations of cooperative control for uav teams, in: AIAA 3rd Unmanned Unlimited Technical Conference, 2004.

[4] M. De Gennaro, A. Jadbabaie, Formation control for a cooperative multi-agent system using decentralized navigation functions, in: American Control Conference, 2006.

[5] T. Paul, T. Krogstad, J. Gravdahl, Modelling of uav formation flight using 3d potential field, Simulation Modelling Practice and Theory 16 (9) (2008) 1453–1462.

[6] G. Hassan, K. Yahya, I. ul Haq, Leader-follower approach using full-state linearization via dynamic feedback, in: International Conference on Emerging Technologies, 2006, pp. 297–305.

[7] I. Shames, B. Fidan, B. D. Anderson, Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations, Automatica 45 (4) (2009) 1058–1065.

[8] J. Jansson, F. Gustafsson, A framework and automotive application of collision avoidance decision making, Automatica 44 (9) (2008) 2347–2351.

[9] R. Schlanbusch, R. Kristiansen, P. J. Nicklasson, Spacecraft formation reconfiguration with collision avoidance, Automatica 47 (7) (2011) 1443–1449.

[10] B. Cetin, M. Bikdash, F. Hadaegh, Hybrid mixed-logical linear programming algorithm for collision-free optimal path planning, Control Theory Applications, IET 1 (2) (2007) 522–531.

[11] X. Koutsoukos, P. Antsaklis, J. Stiver, M. Lemmon, Supervisory control of hybrid systems, Proceedings of the IEEE 88 (7) (2000) 1026–1049.

[12] P. Tabuada, Verification and control of hybrid systems: a symbolic approach, Springer-Verlag New York Inc, 2009.

[13] A. Karimoddini, H. Lin, B. M. Chen, T. H. Lee, Hybrid formation control of the unmanned aerial vehicles, Mechatronics 21 (5) (2011) 886–898.

[14] P. Ramadge, W. Wonham, The control of discrete event systems, Proceedings of the IEEE 77 (1) (1989) 8–98.

[15] G. Cai, B. M. Chen, K. Peng, M. Dong, T. H. Lee, Modeling and control system design for a uav helicopter, in: 14th IEEE Mediterranean Conference on Control and Automation, 2006, pp. 1–6.

[16] A. Karimoddini, G. Cai, B. M. Chen, H. Lin, T. H. Lee, Hierarchical Control Design of a UAV Helicopter," in Advances in Flight Control Systems, INTECH, Vienna, Austria, 2011.

[17] A. Karimoddini, G. Cai, B. M. Chen, H. Lin, T. H. Lee, Multi-layer flight control synthesis and analysis of a small-scale uav helicopter, in: IEEE Conference on Robotics Automation and Mechatronics, 2010, pp. 321–326.