Vol. 00, No. 00, Month 200x, 1–19

Hierarchical Hybrid Modeling and Control of an Unmanned Helicopter

Ali Karimoddini[a], Hai Lin[b*], Ben M. Chen,[c], and Tong H. Lee[c]

[a] *Department of Electrical and Control Engineering, North Carolina Agricultural and Technical State University Greensboro, NC 27411*

[b]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556

[c]Graduate School for Integrative Sciences and Engineering (NGS) and Department of Electrical and Control Engineering (ECE), National University of Singapore;

(*Received 00 Month 200x; final version received 00 Month 200x*)

In this paper, we propose a hybrid modeling and control design scheme for a unmanned helicopter. This control structure has a hierarchical form with three layers: the regulation layer, the motion planning layer, and the supervision layer. For each layer, a separate hybrid controller has been developed. Then, a composition operator is adopted to capture the interactions between these layers. The resulting closed-loop system can flexibly command the helicopter to perform different tasks, autonomously. The designed controller is embedded in the avionic system of the NUS UAV helicopter, and actual flight test results are presented to demonstrate the effectiveness of the proposed control structure.

Keywords: hybrid modelling and control theory, hierarchical control, unmanned aerial vehicles, flight control.

1 Introduction

Over recent years Unmanned Aerial Vehicles (UAVs) have attracted researcher from both military and academic sides due to the wide range of applications of UAVs. However, their complexity of modelling and control imposes many technical and theoretical challenges (Ollero and Merino 2004). A very important issue here is to develop a UAV which is able to perform different missions autonomously. A typical mission is composed of several tasks for which separate controllers are required to be designed. Then, a decision making unit needs to be embedded to coordinate the controllers based on assigned tasks. Hence, the control structure of a UAV has a hybrid nature, which includes both continuous and discrete dynamics that interactively coexist in the system (Sobh and Benhabib 1997). It is common to treat the discrete and continuous dynamics of the UAVs in a decoupled way, (Dong et al. 2007), (Fatemi et al. 2008), which simplifies the design procedure. However, the ignorance of the discrete dynamics and its coupling effect on the continuous evolution of the system is questionable and may degrade the reliability of the system (Karimoddini et al. 2009).

To address this problem, one solution is to use hybrid modelling and control theory to uniformly model and handle both discrete and continuous dynamics of the system (Antsaklis and Nerode 1998). To explore the applications of hybrid modelling and control theory in the sophisticated structures of UAVs, in (Bayraktar et al. 2004), a hybrid controller is developed for the control

Financial supports from NSF-CNS-1239222 and NSF-EECS-1253488 for this work are greatly acknowledged. A primary version of this paper was presented in the 18th World Congress of the International Federation of Automatic Control (IFAC 2012).

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

ISSN: 0020-7179 print/ISSN 1366-5820 online

DOI: 10.1080/0020717YYxxxxxxxx
http://www.informaworld.com

of the altitude and turning rate of a fixed wing UAV. For quadrotors, in (Gillula et al. 2010), a hybrid model for the backflit maneuvering is provided for which a forward reachability analysis guarantees the switching sequence for correct execution of the task. Similarly in (Naldi et al. 2009), a robust reachability analysis is given for taking off and landing of a ductedfan aerial vehicle. When the vehicle is landing, upon contacting with the ground, the control dynamics will be changed. So, the hybrid controller pushes the switching sequence to safely land on the ground. In (Frazzoli et al. 2000), the path planning of a UAV helicopter is translated to a robust hybrid analysis problem and the results are verified through simulation, and in (Schouwenaars et al. 2003), a hybrid controller for the velocity control of a helicopter is provided where Mixed Integer Linear Programming (MILP) is used for the optimal reference generation. In contrast, in this paper, instead of focusing on a specific task, our aim is to propose a framework for the hybrid control of a UAV helicopter so that it can autonomously accomplish the assigned mission. To reduce the complexity of the system and to facilitate the design procedure, we have developed a hierarchical control structure in a systematic way to distribute the control tasks among the layers. The use of hierarchical control and its application to coordination problems have been studied for a long time (Mesarović et al. 1970), (Findeisen et al. 1980); however, considering the concept of hierarchical control within hybrid framework still is a challenging problem.

Hence, the contribution of this paper is that firstly we have proposed a formal hierarchical hybrid modelling and control approach for UAV systems. The control system has three layers: the regulation layer, which is responsible for the low level control; the motion planning layer, which is responsible for path generation to be followed by the regulation layer, and the supervision layer, which is the decision making unit and is responsible for managing the switching scenario to perform a mission autonomously. Each layer has been modelled with an Input/Output hybrid automaton (Lynch et al. 2003). Then, we have introduced a composition operator to synchronise the layers and capture the interplay between them. The existing definitions of composition operator either are only useful for fully connected systems (Johansson 2005), or cannot refine the discrete transitions or states of the system (Lynch et al. 2001), (Rashid and Lygeros 1999). In contrast, in this paper, a new composition operator is proposed that is able to be used for partially connected systems and can refine the discrete transitions and states in an efficient way.

Finally, the designed controller is implemented on the NUS UAV helicopter (Peng et al. 2009), and real flight tests are conducted to evaluate the proposed hybrid control structure. The flight test results show that the designed control system can be effectively involved in a complex mission composed of several tasks.

The remaining parts of this paper are organized as follows. First, in Section 2, the model of the NUS UAV helicopter is described to be used in our further derivations. Then, in Section 3, a hierarchical hybrid framework has been developed for this UAV helicopter and the layers of this hierarchy are discussed in detail. The experimental results are presented in Section 4, and finally, the paper is concluded in Section 5.

2 The UAV Model and Structure

Before developing a hybrid controller for a UAV helicopter, its model and structure is briefly explained in this section. Here, the test-bed is the NUS UAV helicopter (Fig. 1), which is developed by our research group in the National University of Singapore. This helicopter is a Raptor-90 helicopter, which is equipped with an avionic system, including the onboard computer system, the sensors, and the actuators that together generate the control signals for an automatic flight. The construction procedure of such an autonomous UAV is described in (Cai et al. 2008), the hardware details are explained in (Cai et al. 2005), and its low level flight control performance is discussed in (Peng et al. 2009).

Based on the first-principles modeling approach detailed in (Cai et al. 2008), a nonlinear dynamic model for the NUS UAV helicopter has been obtained, which is highly accurate in a wide range of flight envelope. Using the trust-region dogleg method, the obtained model then

Figure 1. An Autonomous UAV Helicopter.

has been linearized at the hovering state in which the linear and angular velocities, the pitch angle, and the roll angle of the UAV are close to zero (Cai et al. 2006). To capture the UAV dynamics, it is required to consider two coordinate systems. The moment and force equations must be derived in a moving coordinate system whose origin is located at the center of gravity of the UAV, whereas to obtain the net displacement of the UAV, we need to consider a fixed coordinate system that is centered in the flight starting point. The moving and fixed coordinate systems are called the body frame and the ground frame, respectively.

Deriving the force and moment equations in the body frame of the UAV and linearizing the resulting nonlinear model at the hovering state, will result in the following model:

$$\dot{x}_{in} = Ax_{in} + Bu, \tag{1}$$

where $x_{in} = [\ V_x(m/s)\ V_y(m/s)\ \omega_x(rad/s)\ \omega_y(m/s)\ \phi(rad)\ \theta(rad)\ \tilde{a}_1(rad)\ \tilde{b}_1(rad)\ V_z(m/s)\ \omega_z(rad/s)\ w_{zf}(rad/s)\ ]'$ is the internal state of the system. Here, V_x, V_y, and V_z are the linear velocities; ω_x, ω_y, and ω_z are the angular velocities; ϕ is the roll angle; θ is the pitch angle; $\tilde{a}_1$ and $\tilde{b}_1$ are the flapping angles, and w_{zf} is the state variable of the gyro rate that introduces a first order differential equation to capture the effect of δ_{pedal} (Peng et al. 2006). Furthermore, $u = \left[\delta_{roll}(rad)\ \delta_{pitch}(rad)\ \delta_{col}(rad)\ \delta_{pedal}(rad)\right]'$ is the vector of the control input signals, to be given to the servos to control the angle of the blades and to drive the UAV in different directions. Finally, $w = (u_{wind}, v_{wind}, w_{wind})$ is the wind gust disturbance where u_{wind}, v_{wind}, w_{wind} affect the UAV velocities in the x, y and z directions, respectively. The state and input matrices A and B of the corresponding linearized model, and the disturbance matrix E are as follows:

$$A = \begin{bmatrix} A_2 & 0_{8\times 3} \\ 0_{3\times 8} & A_1 \end{bmatrix}, B = \begin{bmatrix} B_2 & 0_{8\times 2} \\ 0_{3\times 2} & B_1 \end{bmatrix}, E = \begin{bmatrix} E_2 & 0_{8\times 1} \\ 0_{3\times 2} & E_1 \end{bmatrix}$$

where

$$A_1 = \begin{bmatrix} -0.6821 & -0.1070 & 0 \\ -0.1446 & -5.5561 & -36.6740 \\ 0 & 2.7492 & -11.1120 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 15.6491 & 0 \\ 1.6349 & -58.4053 \\ 0 & 0 \end{bmatrix}, \quad E_1 =$$

$$\begin{bmatrix} -0.5995 \\ -1.3832 \\ 0 \end{bmatrix}, \quad B_2 \quad = \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.0496 & 2.6224 \\ 2.4928 & 0.1740 \end{bmatrix}, \quad E_2 \quad = \quad \begin{bmatrix} -0.1778 & 0 \\ 0 & -0.3104 \\ -0.3326 & -0.2051 \\ 0.0802 & -0.2940 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_2 \quad =$$

$$\begin{bmatrix} -0.1778 & 0 & 0 & 0 & 0 & -9.7807 & -9.7808 & 0 \\ 0 & -0.3104 & 0 & 0 & 9.7807 & 0 & 0 & 9.7807 \\ -0.3326 & -0.5353 & 0 & 0 & 0 & 0 & 75.7640 & 343.86 \\ -0.1903 & -0.2940 & 0 & 0 & 0 & 0 & 172.620 & -59.958 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -8.1222 & 4.6535 \\ 0 & 0 & -1 & 0 & 0 & 0 & -0.0921 & -8.1222 \end{bmatrix}.$$

To obtain the net displacement of the UAV, x_{out}, we should first obtain the velocity vector in the ground frame as a fixed coordinate system, and then, the integration of the velocity vector in the fixed frame will yield the net displacement:

$$\dot{x}_{out} = \Omega'(\Theta) C x_{in}, \tag{2}$$

where $x_{out} = [\ x(m)\ y(m)\ z(m)\ \psi(rad)]'$. Here, x, y, and z describe the position of the UAV in the ground frame, ψ is its heading angle, and $\Theta = [\phi,\ \theta,\ \psi]^T$ is the orientation vector. Matrix C and the block $\Omega(\Theta)$ are as follows:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \ \Omega(\Theta) = \begin{bmatrix} R(\Theta) & 0 \\ 0 & 1 \end{bmatrix},$$

where the block $R(\Theta)$ is a transformation matrix from the ground frame to the body frame and it has the following form:

$$R(\Theta) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{3}$$

.

The model diagram of the UAV helicopter is depicted in Fig. 2. In the next section, we will discuss about the control design for this semi-linearized model of the UAV within the hybrid modelling and control framework.

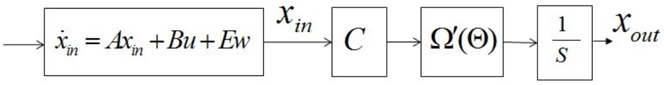


Figure 2. The diagram of the UAV model.

3 Hybrid Modelling and Control of an Unmanned Helicopter

3.1 *Hierarchical Hybrid Modelling and Control of an Unmanned Helicopter*

To design a fully autonomous controller for this helicopter, we propose a hierarchical hybrid control structure that consists of three layers: the *Regulation layer*, the *Motion planning layer*, and the *Supervision layer*. Each layer has a hybrid structure and is responsible to do a specific task. The relation between these layers can be described by hybrid composition operator. Fig .3 shows the overall picture of this system and describes the nature and objectives of each layer. The philosophy behind this hierarchy is that the lower levels are involved in more details such as reference tracking and stability analysis, while the higher levels mostly manage and coordinate the control scenarios to achieve the assigned task. The advantage of this structure is that it simplifies the design procedure so that each layer can be developed to accomplish a particular part of the control task. Next, we will describe the layers of this control hierarchy.

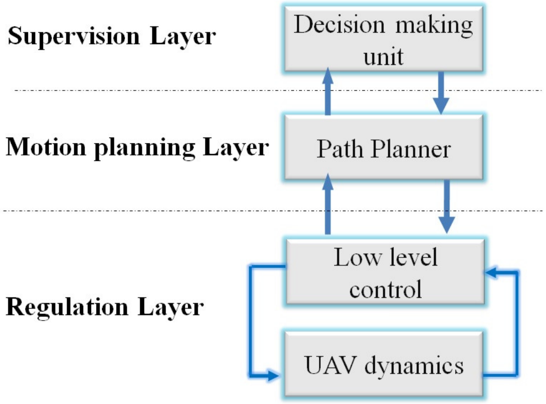


Figure 3. Hierarchical hybrid control structure of an autonomous UAV Helicopter.

3.2 *The Regulation Layer*

The regulation layer is directly connected to the UAV avionic system and can manipulate the actuators and gather the sensors reading for the control process. It also receives the task scheduling commands from the motion planning layer to activate proper control modes. For different velocities and situations, different controllers can be designed. For example, in (Cai et al. 2010), several controllers have been designed for different modes of operation of the NUS UAV helicopter. Then, the higher layers are responsible to activate the proper control modes. To elaborate the idea of hierarchical control, without loss of generality, here we consider two control modes for the regulation layer of this UAV as described in the following parts.

3.2.1 ***Velocity Control Mode***

In the velocity control mode (vc), one can stabilize the attitude of the helicopter and control the UAV to move with the desired velocity vector (v_x, v_y, v_z) and the desired yaw rate, w_z. For this purpose, we will use an H_∞ controller by which both the robust stability and a proper performance of the system can be achieved, simultaneously. To design a H_∞ controller, first, looking at matrices A, B, and E in (1), it can be seen that, the model is a decoupled system with two separate subsystems as follows:

$$\dot{x}_1 = A_1x_1 + B_1u_1 + E_1w_1 \tag{4}$$

$$\dot{x}_2 = A_2x_2 + B_2u_2 + E_2w_2 \tag{5}$$

where $x_1 = [\ V_{z_b}(m/s)\ \omega_{z_b}(rad/s)\ w_{zf}(rad/s)\ ]'$, $u_1 = [\ \delta_{col}\ \delta_{pedal}\ ]'$, $x_2 = [\ V_{x_b}(m/s)\ V_{y_b}(m/s)\ \omega_{x_b}(rad/s)\ \omega_{y_b}(rad/s)\ \phi(rad)\ \theta(rad)\ \tilde{a}_s(rad)\ \tilde{b}_s(rad)\ ]'$, and $u_2 = [\ \delta_{roll}(rad)\ \delta_{pitch}(rad)\ ]'$.

Now, starting with Subsystem 1, and using the notation analogous with (Chen 2000), we define the measurement output simply as the state feedback in the form of $y_1 = C_{11}x_1$ with $C_{11} = I$.

Also, we define the controlled output h_1 in the form of $h_1 = C_{12}x + D_{12}u$, where

$$C_{12} = \begin{bmatrix} & 0_{2\times 3} & \\ 3.1623 & 0 & 0 \\ 0 & 3.1623 & 0 \\ 0 & 0 & 1.7321 \end{bmatrix}, D_{12} = \begin{bmatrix} 44.7214 & 0 \\ 0 & 28.2843 \\ \multicolumn{2}{c}{0_{3\times 2}} \end{bmatrix} \tag{6}$$

The nonzero entries of C_{12} and D_{12} are used for tuning the controller. Here, they are determined experimentally to achieve the desired performance. Meanwhile, the H_∞ design guarantees internal stability and robustness of the system. Indeed, H_∞ control design reduces the effect of the wind gust disturbance on the control performance, by minimizing the H_∞ norm of the closed-loop transfer matrix from the disturbance w to the controlled output h_1, denoted by T_1. The H_∞ norm of the transfer function T_1 is defined as follows:

$$\|T_1\|_\infty = \underset{0\leq\omega<\infty}{sup}\ \sigma_{max}[T_1(j\omega)] \tag{7}$$

where $\sigma_{max}[*]$ denotes the maximum singular value of the matrix $*$.

Having the matrices C_{12} and D_{12}, one can find γ_∞^*, which is the optimal H_∞ performance for the closed-loop system from the disturbance input w to the controlled output h_1 over all the possible controllers that internally stabilize the system. As practically, γ_∞^* is not achievable, we will try to reach γ_∞, which is slightly larger than γ_∞^*.

With this choice of the control parameters, D_{11} and D_{12} are full rank and the quadruples $(A_1, B_1, C_{12}, D_{12})$ and $(A_1, E_1, C_{11}, D_{11})$ are left invertible and are free of invariant zeros. Therefore, we have a so-called regular problem, for which we can use the well-established H_∞ control theory (Chen 2000). As it was mentioned, the resulting closed loop system suboptimality minimizes the H_∞ norm of the transfer function from the disturbance w to the controlled output h_1. As a result, F_1 is the H_∞ control gain that can be achieved as follows:

$$F_1 = -(D_{12}'D_{12})^{-1}(D_{12}'C_{12} + B_1'P_1) \tag{8}$$

where matrix P_1 is the positive semi-definite solution of the following H_∞ algebraic Riccati equation:

$$\begin{gathered} A_1'P_1 + P_1A_1 + C_{12}'C_{12} + P_1E_1E_1'P_1/\gamma^2 - \\ (P_1B_1 + C_{12}'D_{12})(D_{12}'D_{12})^{-1}(D_{12}'C_{12} + B_1'P_1) = 0 \end{gathered} \tag{9}$$

For this system and these control parameters values, the value of γ_∞^* is 1.4516. Choosing $\gamma_\infty = 1.4616$, will lead to $F_1 = \begin{bmatrix} -0.0935 & -0.0005 & 0.0027 \\ 0.0008 & 0.0364 & -0.0481 \end{bmatrix}$. The same procedure can be followed for Subsystem 2, and the resulting feedback gain will be $F_2 = \begin{bmatrix} 0.0017 & -0.1683 & -0.0486 & 0.0081 & -1.9336 & -0.1974 & -0.3227 & -2.1444 \\ 0.0815 & -0.0461 & -0.0087 & -0.0535 & -0.3908 & -1.0690 & -1.1712 & -0.4659 \end{bmatrix}$. Then, considering these two subsystems together, the control law will be in the form of $u = Fx_{in} + Gr$ (Fig. 4), where matrix $F = \begin{bmatrix} F_2 & 0 \\ 0 & F_1 \end{bmatrix}$ was obtained through the robust H_∞ design technique, and $G = -(C(A+BF)^{-1}B)^{-1}$ is the feedforward gain, obtained from the inverse of the system steady state gain.

3.2.2 ***Position Control Mode***

The control objective in the position control mode, (pc), is to drive the UAV to follow the desired path. In other words, the state variable x_{out} should track the given reference r. The control law for this operation mode is $u = Fx_{in} + G\Omega K_p(r - x_{out})$. As it is shown in Fig. 5,

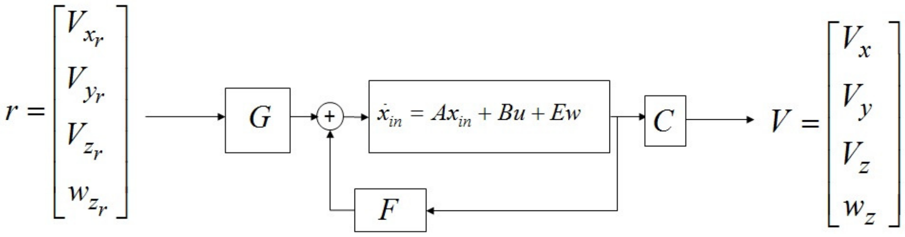

$r = \begin{bmatrix} V_{x_r} \\ V_{y_r} \\ V_{z_r} \\ w_{z_r} \end{bmatrix}$
G
+
$\dot{x}_{in} = Ax_{in} + Bu + Ew$
C
F
$V = \begin{bmatrix} V_x \\ V_y \\ V_z \\ w_z \end{bmatrix}$

Figure 4. The controller for the velocity-control of the UAV.

this controller consists of two layers: the inner-loop and the outer-loop. The inner-loop controller stabilizes the attitude of the UAV, and its parameters, F and G, are selected as the same as the velocity control mode. The outer-loop controller, however, smoothly drives the UAV to the desired position. In the outer-loop the block Ω is used to compensate for the transformation matrix Ω', as they have the property that $\Omega\Omega' = I$, and K_p is a P-controller. In (Karimoddini et al. 2010), a tractable procedure has been proposed for the design of a decentralized P-controller, K_p, for multi-variable systems, based on the generalized Nyquist theorem and disturbance analysis.

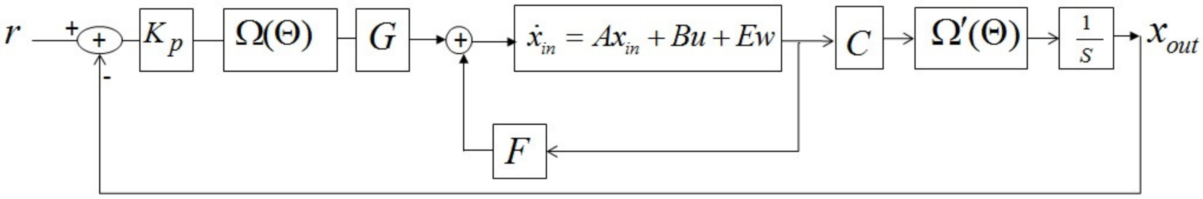


Figure 5. The controller for the position-control of the UAV.

3.2.3 *Hybrid Model of the Regulation Layer*

Now, we can present the hybrid model of the regulation layer based on what explained for each control mode. Both control modes have the same plant dynamics $\dot{x}_{in} = Ax_{in} + Bu$; however, the control law in the velocity control mode is $u = Fx_{in} + Gr$, and in the position control mode is $u = Fx_{in} + G\Omega K_p(r - x_{out})$.

The graph representation of the hybrid model of the regulation layer is shown in Fig. 6. Formally, this hybrid model of the regulation layer can be described by a hybrid automaton ((Lynch et al. 2003), (Liu et al. 1999)) $H_R = (V_R, X_R, U_R, Y_R, f_R, Init_R, Inv_R, E_R, Guard_R, Reset_R, h_R)$, where

- $V_R = \{start, vc, pc\}$ is the set of discrete states, where vc and pc stand for the velocity control mode and the position control mode, respectively. The $start$ mode is used for the initialization of the system to choose either of the modes.
- $X_R = [x_{in}, x_{out}]'$ is the continuous state of the system.
- $U_R = U_{D_R} \times U_{C_R}$ is the input space, where $U_{C_R} = r \subseteq \mathbb{R}^4$ is the continuous control input, and $U_{D_R} = \{cmd_V, cmd_P\}$ is the set of discrete inputs. The subscripts denote the corresponding ending discrete states in Fig. 6. For instance, cmd_P is the command that fires a transition to the position control mode.
- $Y_R = Y_{D_R} \times Y_{C_R}$ is the system output, where here, $Y_{C_R} = x_{out}$ and $Y_{D_R} = V_R$ feedback the current state of the system to the motion planning layer to be able to generate appropriate reference signals.
- $f_R : V_R \times X_R \times U_R \to X_R$ is the vector field description of the system that is defined as follows: $\dot{x} = f_R(v, x, u) = f_R(v, x, r) =$

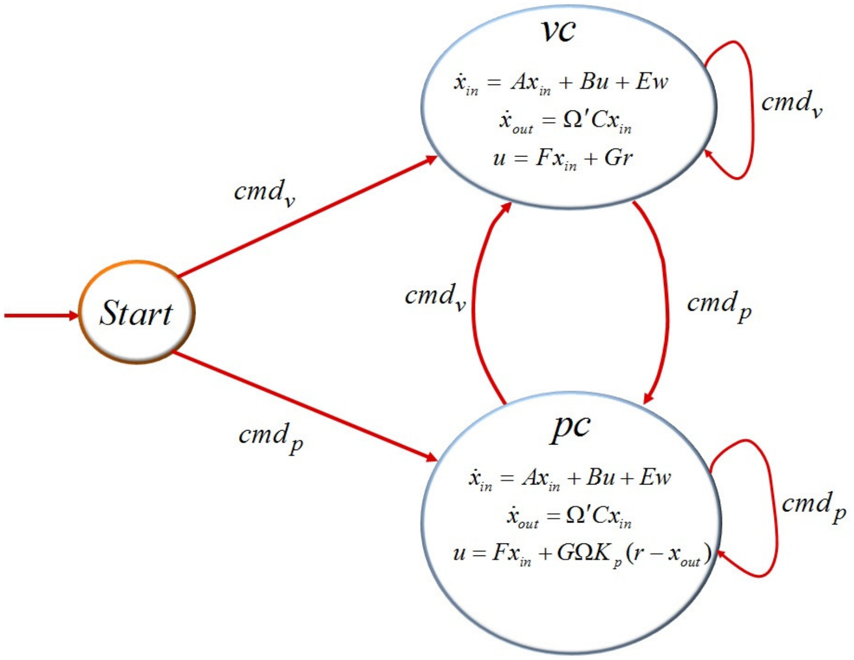

vc
$\dot{x}_{in} = Ax_{in} + Bu + Ew$
$\dot{x}_{out} = \Omega' Cx_{in}$
$u = Fx_{in} + Gr$
cmd_v
cmd_v
Start
cmd_v
cmd_p
cmd_p
pc
$\dot{x}_{in} = Ax_{in} + Bu + Ew$
$\dot{x}_{out} = \Omega' Cx_{in}$
$u = Fx_{in} + G\Omega K_p(r - x_{out})$
cmd_p

Figure 6. The hybrid model for the regulation layer.

$$\begin{cases} 0 & if\ v = start \\ \begin{bmatrix} (A+BF)x_{in} + BGr \\ \Omega' Cx_{in} \end{bmatrix} & if\ v = vc \\ \begin{bmatrix} (A+BF)x_{in} - BG\Omega K_p x_{out} + BG\Omega K_p r \\ \Omega' Cx_{in} \end{bmatrix} & if\ v = pc \end{cases}$$

- $Init_R = \{(start, 0)\} \subseteq V_R \times X_R$ is the set of initial states of the UAV.
- $Inv_R \subseteq V_R \times X_R \times U_R$ is the invariant condition. Here, it is required that for both discrete modes, $z > 0$, $v_x, v_y, v_x < 3.5m/s$, $\omega_z < 15deg/s$ and $a, b, \theta, \phi < \frac{\pi}{6}$.
- $E_R \subseteq V_R \times V_R$ is the set of discrete transitions. Here, $E = \{(Start, vc)$, $(start, pc)$, (pc, vc), (vc, pc), (pc, pc), $(vc, vc)\}$.
- $Guard_R : E_R \rightarrow 2^{X_R \times U_R}$ describes the guard conditions for the discrete transitions. For each discrete transition from the vertex v to v', the continuous state of the system and the control input should belong to $Guard(v, v')$. For instance, in Fig. 6, when the system is in mode vc, the control input cmd_P can cause a transition to the mode pc. In the guard map for this transition, no condition has been considered on the continuous state of the system, and only the discrete control input is used for the guard condition.
- $Reset_R : E_R \times X_R \times U_R \rightarrow 2^{X_R}$ describes the reset map. For instance, $z' \in Reset(v, v', z, w)$ shows that for $(v, v') \in E$, $z \in X$, and $w \in U$, there is a transition for which the continuous state of the system will be reset to z'. Here, the reset map is an identity map as there is no jump on the continuous state of the system. when the reset map is an identity map, it is not shown in the graph representation.
- $h_R : V_R \times X_R \rightarrow Y_R$ is the output map. Here we have $h(v, x) = x_{out}$.

3.3 *Motion Planning Layer*

Based on the feedbacked information received from the regulation layer, the motion planning layer can activate the corresponding control mode in the regulation layer and can generate proper control references in the form of a feasible path to be tracked by the regulation layer. The path generation mechanism could be done in an off-line manner or through a dynamic path planning mechanism:

3.3.1 ***Off-line Path Generation Mechanism***

In this method, based on the problem requirements, an optimal path can be generated and stored in the library of the system. As an example, we explain a motion planning layer that has been used in our flight tests using off-line path generation mechanism. The hybrid automaton for this model of the motion planning layer is $H_P = (V_P, X_P, U_P, Y_P, f_P, Init_P, Inv_P, E_P, Guard_P, Reset_P, h_p)$ where $X_p = (r_x, r_y, r_z, r_\psi)$ is the continuous state of the motion planning layer and indeed, it is the generated reference that is going to be given to the regulation layer. The discrete state is $V_p = \{Start_p$, $Path - Z_p$, $Path - C_p$, $Ascend_p$, $Hover_p$, Vel_p, $Descend_p$, $Emergency_p\}$ where $Start_p$, $PathZ_p$, $PathC_p$, $Ascend_p$, $Hover_p$, Vel_p, $Descend_p$, and $Emergency_p$ stand for starting the task, zigzag path tracking, circle path tracking, ascending, hovering, generating velocity references, descending, and emergency mode, respectively. Here, the control signal is $U_p = U_{C_p} \times U_{D_p}$ where $U_{C_p} = X_R$ is the current state of the system that is feedbacked from the regulation layer and $U_{D_p} = \{cmd_{PathZ}$, cmd_{PathC}, cmd_{Ascend}, cmd_{Hover}, cmd_{Vel}, $cmd_{Descend}$, $cmd_{Emergency}\}$ is the command received from the supervision layer. When the motion planning layer receives one of these commands, it switches to the corresponding discrete mode. $Y_p = Y_{D_p} \times Y_{C_p}$ is the layer output. Here, $Y_{C_p} = X_P$ is the continuous part, which informs the supervision layer about the current state of the motion planning layer and also, it will be given to the regulation layer as the generated reference to be tracked. $Y_{D_p} = Y_{D_{pr}} \times Y_{D_{ps}}$ is the discrete output signal where $Y_{D_{ps}} = V_p$ is given to the supervisor to inform about the current discrete mode of the motion planning layer and $Y_{D_{pr}} = \{cmd_p, cmd_v\}$ is the command that activates the proper control mode in the regulation layer:

$$Y_{D_{pr}} = \begin{cases} cmd_p & for\ V_p = PathC_p, PathZ_p, Ascend_p, Descend_p \\ cmd_v & for\ V_p = Vel_p, Emergency_p, , Hover_P \end{cases}$$

The dynamics of the motion planning layer is

$$\dot{X}_p(v) = [\dot{x}_r\ \dot{y}_r\ \dot{z}_r\ \dot{\psi}_r]^T = \begin{cases} (0, 0, f_{z_a}(t), 0) & v = Ascend_p\ \ f_z(t) > 0 \\ (0, 0, f_{z_d}(t), 0) & v = Descend_p\ \ f_z(t) < 0 \\ (f_{x_{pc}}(t), f_{y_{pc}}(t), f_{z_{pc}}(t), f_{\Psi_{pc}}(t)) & v = PathC_p \\ (f_{x_{pz}}(t), f_{y_{pz}}(t), f_{z_{pz}}(t), f_{\Psi_{pz}}(t)) & v = PathZ_p \\ (f_{x_v}(t), f_{y_v}(t), f_{z_v}(t), f_{\Psi_v}(t)) & v = Vel_p \\ (0, 0, 0, 0) & v = Emergency_p, Hover_p \end{cases}$$

In the graph representation for the hybrid model of the motion planning layer, all discrete states are connected, and the command cmd_* can fire a transition to the state $*$. There is no guard condition and jump for the discrete transitions. As this graph is tedious, we have not shown it here.

3.3.2 ***On-line Path Generation Mechanism***

Here, the objective is to generate the references in an on-line way to be tracked by the regulation layer. The basic path planning problem in which a robot have to be driven from the start point towards the destination point while respecting the constraints, is a standard optimal control problem and has been addressed with different methods such as potential function, mixed integer linear programming, cell decompositions and probabilistic roadmaps (Latombe 1990). But, these methods are not able to address more advance path planning problems when there are number of goals with a particular order of execution. The alternative solution is to utilize symbolic motion planning approaches (Belta et al. 2007) by which it is possible to generate a path associated with a sequence of symbols, which can follow logical supervisory rules. For this purpose, one can introduce an abstract system $\dot{x}_p(t) = f_p(x_p(t), u_p(t))$, which is simpler than the original model of the regulation layer as it ignores some unnecessary information. This abstract system should be approximately similar to the regulation layer dynamics so that the regulation layer can follow the generated reference. To elaborate the idea, let us work on the design of the motion planning layer for one of the NUS UAV helicopters that is involved in a leader follower formation mission as a follower. As we explained, for the regulation layer of this helicopter we have used a multi-layer control structure whose inner-loop controller stabilizes the system using H_∞ control design techniques and its outer-loop is used to drive the system towards the desired position (Fig. 7). Assuming that the inner-loop is fast enough to track the given references (Karimoddini et al. 2011), the inner-loop can be approximated by an identity matrix. Therefore, the regulation layer dynamics is approximately as $\dot{x}_p = u_p$, where x_p is the outer loop state variable, and u_p is a control parameter, which should be designed by the formation algorithm.

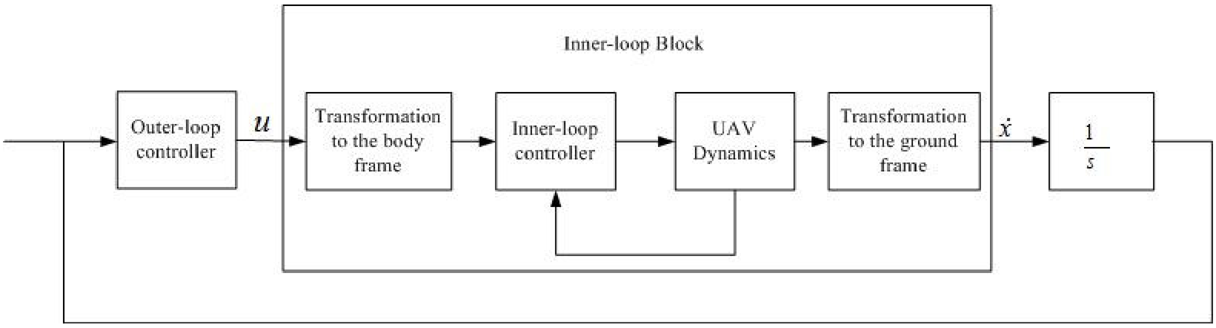


Figure 7. Control Structure of the UAV.

Considering the follower velocity in the form of $V_{follower} = V_{leader} + V_{rel}$, we can imagine a relative coordinate system in which the leader has a relatively fixed position and hence, the formation problem is reduced to drive the follower UAV towards the desired position. For this purpose, in (Karimoddini et al. 2012), we have introduced a hybrid symbolic approach based on spherically partitioning of the space. Consider an sphere S_{R_m}, with the radius of R_m that is centered at the desired position. The sphere is partitioned into several sectors as shown in Fig. 8. To reach the formation, the system's trajectory should reach one of the sectors adjacent to the sphere's origin, and to maintain the formation, the system trajectory should remain there for ever. Meanwhile, the follower UAV should avoid the collision with the leader UAV. These tasks can be achieved by properly driving the system trajectory through the partitioned space. Since the motion planning dynamics has a linear form, the control u_p can be constructed as the convex combinations of control signals on the vertices, so that, the system trajectory either remain inside one of the sectors or exit form a desired facet. The resulting control signal is in the form of $u_p(Cmd_*) = \sum_{v_m} \lambda_m u_{v_m}(Cmd_*)$, $m = 0, 1, ..., 7$, where $0 \leq \lambda_m \leq 1$ are coefficients, $u_{v_m}(Cmd_*)$ are the control values at the vertices, and Cmd_* is the discrete command, which could be Cmd_R, Cmd_K, or Cmd_C that stand for the commands for reaching the formation, keeping the formation, and collision avoidance, respectively. Further details about this online path generation mechanism are available in (Karimoddini et al. 2012). Using this method, the hybrid model for the motion planning layer of the follower unmanned helicopter is $H_P = (V_P, X_P, U_P, Y_P, f_P, Init_P, Inv_P, E_P, Guard_P, Reset_P, h_P)$ as the the hybrid model for this layer, where $X_p = (r_x, r_y, r_z, r_\psi)$ as the continuous state of the motion planning layer. The discrete state is $V_p = \{$ $Start_p$, $Hover_p$, $ReachFormation_p$, $KeepFormation_p$, $CollisionAvoidance_p\}$. Similar to the previous case, the control signal is $U_p = U_{C_p} \times U_{D_p}$ where $U_{C_p} = X_R$, and $U_{D_p} = U_{D_{pr}} \times U_{D_{ps}}$. The set $U_{D_{ps}} = \{cmd_H, cmd_R, cmd_K, cmd_C\}$ is the command received from the supervision layer, and $U_{D_{pr}}$ is the the information about the current discrete mode of the regulation layer. The subscripts R, K, C, and H stand for reaching the formation, keeping the formation, collision avoidance, and hovering, respectively. The output is $Y_p = Y_{D_p} \times Y_{C_p}$, where $Y_{C_p} = X_P$ is the continuous part and $Y_{D_p} = Y_{D_{pr}} \times Y_{D_{ps}}$ is the discrete output signal where $Y_{D_{ps}} = V_p$ is the discrete output to be given to the supervisor to inform about the current discrete mode of the motion planning layer and $Y_{D_{pr}} = \{cmd_p, cmd_v\}$ is the command that activates the proper control mode in the regulation layer:

$$Y_{D_{pr}} = \begin{cases} cmd_v & for\ V_p = Hover_P \\ cmd_p & for\ V_p = ReachFormation_p,\ KeepFormation_p,\ CollisionAvoidance_p \end{cases}$$

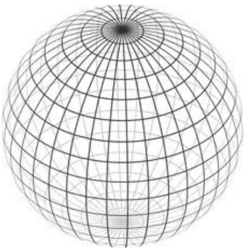


Figure 8. A spherically partitioned space.

The dynamics of the motion planning layer is as follows:

$$\dot{x}_p(v) = [\dot{x}_r\ \dot{y}_r\ \dot{z}_r\ \dot{\psi}_r]^T =$$

$$\begin{cases} \sum_{v_m} \lambda_m u_{v_m}(Cmd_R) & for\, m = 0, 1, ..., 7,\ v = ReachingFormation_p \\ \sum_{v_m} \lambda_m u_{v_m}(Cmd_K) & for\, m = 0, 1, ..., 7,\ v = KeepFormation_p \\ \sum_{v_m} \lambda_m u_{v_m}(Cmd_C) & for\, m = 0, 1, ..., 7,\ v = CollisionAvoidance_p \\ 0 & for\, v = Hover_p \end{cases}$$

The transitions for this hybrid model are shown in the graph representation of the system in Fig. 9.

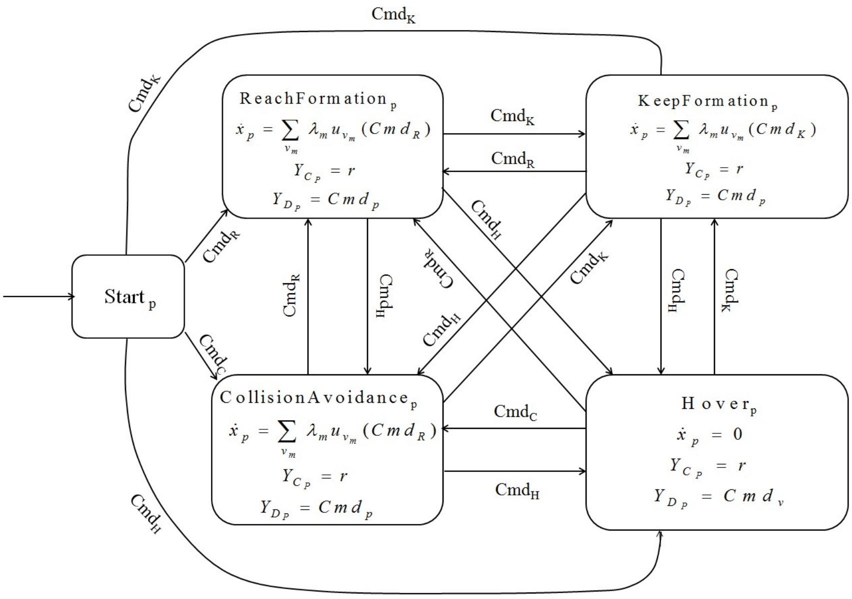


Figure 9. The hybrid model for the motion planning layer for a formation mission.

3.4 *Supervision Layer*

This layer is responsible for the decision making and task scheduling for the mission that should be performed by the UAV. The supervision layer can be presented by a purely discrete automaton (Ramadge and Wonham 1989) or a timed automaton (Alur and Dill 1994), which are subclasses of hybrid systems. Using the off-line path planning mechanism for the motion planning layer, described in the previous section, a supervision layer has been designed for a typical mission shown in Fig. 10. This mission starts with 8 meters ascending, followed by 15 Sec hovering, 60 Sec zigzag path tracking, 35 Sec velocity control, 42 Sec circle path tracking, 20 Sec hovering, and 8 meters descending. The mission ends with hovering. For safety issues, when the measured signals are out of range, the fuel level sensor alarms, or other possible problems occur, a fault signal is generated, which leads the system to the emergency mode. The discrete states and corresponding discrete outputs are shown in Fig. 10. These discrete outputs are commands that activate a control mode in the motion planning layer. The input space of this layer is in the form of $U_s = U_{C_s} \times U_{D_s}$ where $U_{C_s} = Y_{C_{ps}} = X_P$ is the current state of the path planner, and $U_{D_s} = U_{D_{se}} \times U_{D_{sp}}$ where $U_{D_{sp}} = Y_{D_{ps}} = V_p$ is the information about the current discrete mode of the motion planning layer, and $U_{D_{se}} = \{Cmd_{StartMission}, Fault\}$ are the external events generated by the other sources. Here, the command $Cmd_{StartMission}$ is generated by the ground station, and the command $Fault$ is generated by the UAV event generation mechanism for faulty cases (e.g., when the measurement values are out of range). The graph representation for this supervisor is shown in Fig. 10.

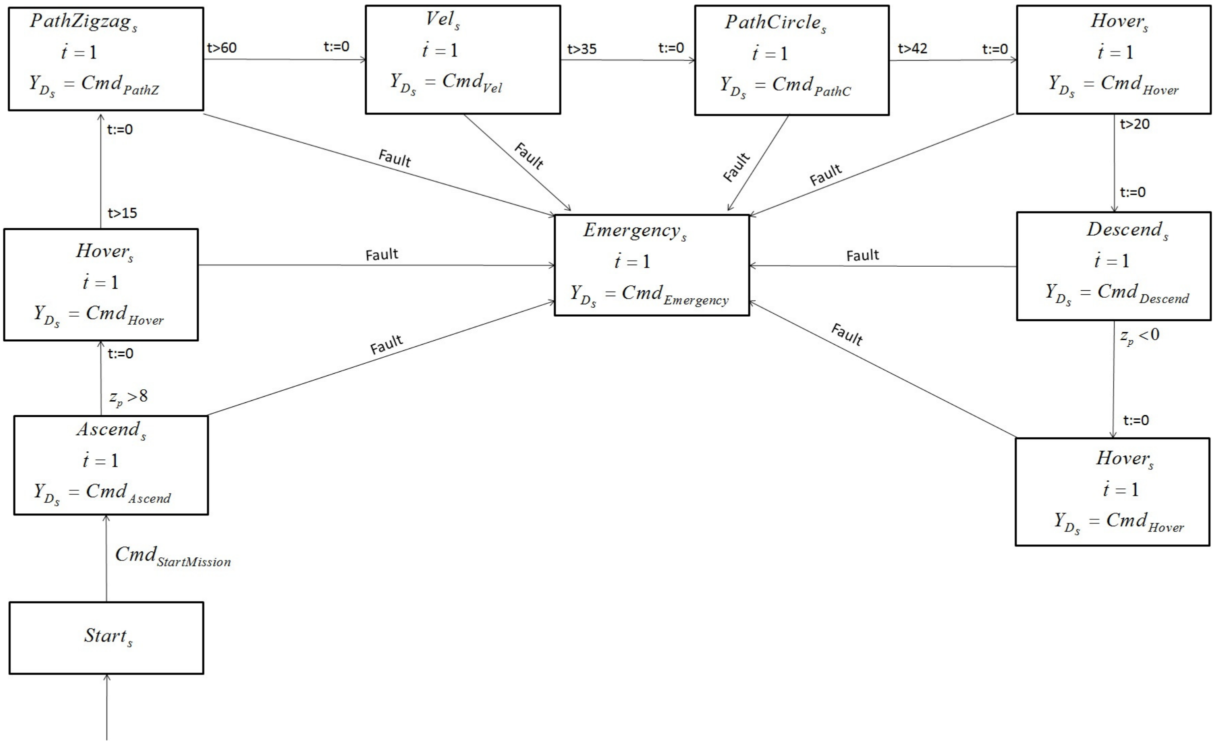


Figure 10. The supervision layer for a mission with successive tasks generated with off-line path generation mechanism.

As another example, using the motion planning layer for the on-line path planning, a supervisor has been designed for a follower UAV involved in a formation mission as shown in Fig. 11. It starts with the hovering. When the follower receives the event $Cmd_{StartFormation}$ from the leader, it switches to the $ReachFormation_s$ mode. If the supervisor detects a collision alarm, an event $Cmd_{CollisionAlarm}$ will be generated and the system switches to the $CollisionAvoidance_s$ mode. Disappearing the collision alarm, the command $Cmd_{AlarmRemoved}$ causes a transition to the the $ReachFormation_s$ mode to resume the formation. Finally, when the formation is achieved, the system switches to the $KeepFormation_s$ mode. The input space for this supervisor is in the form of $U_s = U_{C_s} \times U_{D_s}$ where $U_{C_s} = Y_{C_{ps}} = X_P$ is the current state of the path planner, and $U_{D_s} = U_{D_{se}} \times U_{D_{ss}} \times U_{D_{sp}}$ where $U_{D_{sp}} = Y_{D_{ps}} = V_p$ is the set of events received from the motion planning layer, $U_{D_{ss}} = \{Cmd_{CollisionAlarm}, Cmd_{AlarmRemoved}, Cmd_{KeepFormation}\}$ is the set of events observed by the supervisor, and $U_{D_{se}} = \{Cmd_{StartMission}, Cmd_{StartFormation}, Cmd_{EndFormation}\}$ is the set of external events received from other sources such as the ground station or the leader UAV. The output is in the form of $Y_S = Y_{D_s} = \{cmd_R, cmd_K, cmd_C, cmd_H, cmd_E\}$. These commands activate a proper control mode in the motion planning layer. The transitions and other details can be seen in Fig. 11.

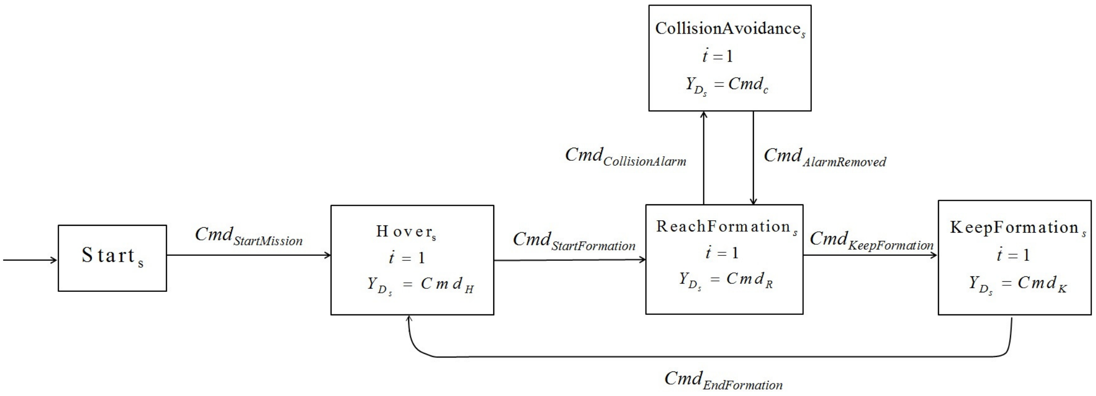


Figure 11. The supervision layer for a formation mission.

3.5 *Synchronizing the Layers of the Control Hierarchy*

To establish such a hierarchy, it is required to introduce a composition operator to synchronize the layers of the control hierarchy and to capture their relation (Karimoddini et al. 2011). In (Johansson 2005), a definition of parallel composition for fully connected hybrid systems is introduced. The resulting closed loop system for such a system is an autonomous unit which cannot be extended to a multi-agent scenario or a multi-layer structure. In (Lynch et al. 2001) and (Rashid and Lygeros 1999), a more general definition of composition of hybrid systems has been given in which the components need not to be fully connected. However in this method, the elements are only coexist in the combined system and there is no refinement on the transitions and states of the closed loop system. In contrast, here, a new definition of the composition operator is given for hybrid systems that can be used for hybrid-multi-agent systems or a multi-layer hybrid system. Furthermore, it considers a treatment on the discrete transitions and states of the composed system, which leads to a more simplified system. First, we need to define the composability condition:

Definition 3.1: Composability of hybrid automata
Hybrid automata H_1, H_2, ..., H_n are composable if:

(1) $Y_i \bigcap Y_j = \emptyset$, $V_i \bigcap V_j = \emptyset$, $X_i \bigcap X_j = \emptyset$ for all $i \neq j$ and $i, j = 1, ..., n$.
(2) $U_i \setminus Y_i = \emptyset$ for all $i = 1, ..., n$.

The first condition avoids the conflict between the components and the second condition guarantees the casuality condition.

Definition 3.2: Composition of hybrid automata
Consider two composable hybrid automata $H_1 = (V_1, X_1, U_1, Y_1, f_1, Init_1, Inv_1, E_1, Guard_1, Reset_1, h_1)$ and $H_2 = (V_2, X_2, U_2, Y_2, f_2, Init_2, Inv_2, E_2, Guard_2, Reset_2, h_2)$. The composition of H_1 and H_2, denoted by $H_1 \| H_2$, is the automaton $H = (V, X, U, Y, f, Init, Inv, E, Guard, Reset, h)$ where:

- $V = V_1 \times V_2$ and $X = X_1 \times X_2$.
- $U = (U_1 \setminus Y_2) \times (U_2 \setminus Y_1)$ and $Y = Y_1 \times Y_2$ (See Fig. 12).

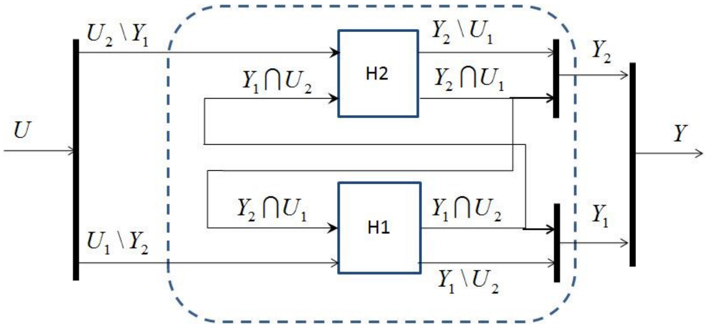


Figure 12. Input and output channels.

- $h : V \times X \to Y,\ where\ \ h = \begin{bmatrix} h_1 : V_1 \times X_1 \to Y_1 \\ h_2 : V_2 \times X_2 \to Y_2 \end{bmatrix}.$
- $f : V \times X \times U \to X$, and $f(v,x,u) = \begin{bmatrix} f_1(v_1,x_1,u_1) \\ f_2(v_2,x_2,u_2) \end{bmatrix} = \begin{bmatrix} f_1(v_1,x_1,(u_1\backslash y_2, u_1 \cap y_2)) \\ f_2(v_2,x_2,(u_2\backslash y_1, u_2 \cap y_1)) \end{bmatrix} = \begin{bmatrix} f_1(v_1,x_1,(u_{11},h_{21}(v_2,x_2))) \\ f_2(v_2,x_2,(u_{22},h_{12}(v_1,x_1))) \end{bmatrix}$, where $v = (v_1,v_2)$, $x = (x_1,x_2)$, $u = (u_1\backslash y_2,\ u_2\backslash y_1) = (u_{11},u_{22}),\ h_{12} : V_1 \times X_1 \to Y_1 \cap U_2,\ and\ h_{21} : V_2 \times X_2 \to Y_2 \cap U_1.$
- $Init = \{((v_1,v_2),(x_1,x_2)) | (v_1,x_1) \in Init_1\ \wedge\ (v_2,x_2) \in Init_2\}.$
- $Inv = \{((v_1,v_2),(x_1,x_2),(u_{11},u_{22})) |\ \exists\ u_1,\ u_2\ s.t.\,(v_1,x_1,u_1) \in Inv_1,\ (v_2,x_2,u_2) \in Inv_2,$ $u_1 = (u_{11},u_{12}),\ u_2 = (u_{22},u_{21}),\ \ u_{11} = u_1\backslash y_2,\ u_{22} = u_2\backslash y_1,\ u_{12} = u_1 \cap y_2 = h_{21}(v_2,x_2),$ $u_{21} = u_2 \cap y_1 = h_{12}(v_1,x_1)\}.$
- $E = \{e = ((v_1,v_2),(v_1',v_2')) \in V \times V | (v_1,v_1') \in E_1\ and\ (v_2,v_2') \in E_2\ and\ Guard(e) \neq \emptyset\}.$
- $Guard : E \to 2^{X \times U}$, which can be described as $Gaurd((v_1,v_1'),(v_2,v_2')) = \{((x1,x2),(u_{11},$ $u_{22})) \in X \times U | (v_1,v_2) \in E_1, (v_1',v_2') \in E_2, \exists u_1,u_2\,s.t.(x_1,u_1) \in G_1(v_1,v_1'),(x_2,u_2) \in G_2(v_2,v_2'), u_1 = (u_{11},u_{12}), u_2 = (u_{22},u_{21}), u_{11} = u_1 \setminus y_2, u_{22} = u_2 \setminus y_1, u_{12} = u_1 \cap y_2 = h_{21}(v_2,x_2), u_{21} = u_2 \cap y_1 = h_{12}(v_1,x_1)\}.$
- $Reset : E \times X \times U \to 2^X$ where for the composed system is defined as $Reset(((v_1,v_2),(v_1',v_2'$ $)),(x_1,x_2),(u_{11},u_{22})) = \{(x_1',x_2') \in X | \exists u_1 = (u_{11},u_{12}), u_2 = (u_{22},u_{21})\,s.t.\,((x_1,x_2),(u_{11},$ $u_{22})) \in G((v_1,v_2),(v_1',v_2')), x_1' \in Reset_1((v_1,v_1'),x_1,u_1), x_2' \in Reset_2((v_2,v_2'),x_2,u_2), u_{11}$ $= u_1 \setminus y_2, u_{22} = u_2 \setminus y_1, u_{12} = u_1 \cap y_2 = h_{21}(v_2,x_2), u_{21} = u_2 \cap y_1 = h_{12}(v_1,x_1))\}.$

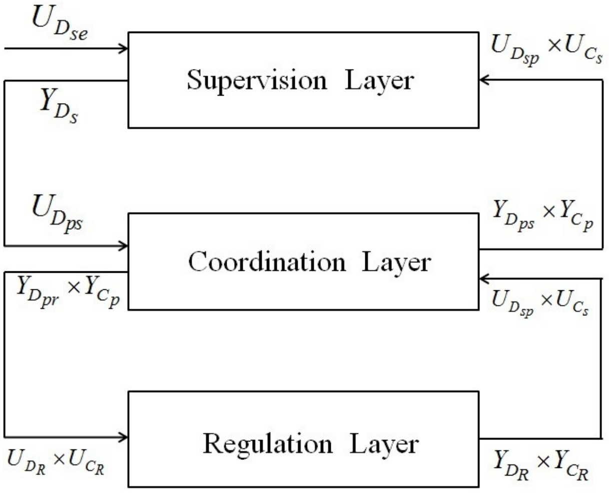


Figure 13. The layers of the control hierarchy.

The control hierarchy of the UAV and the data flow between the layers are shown in Fig.

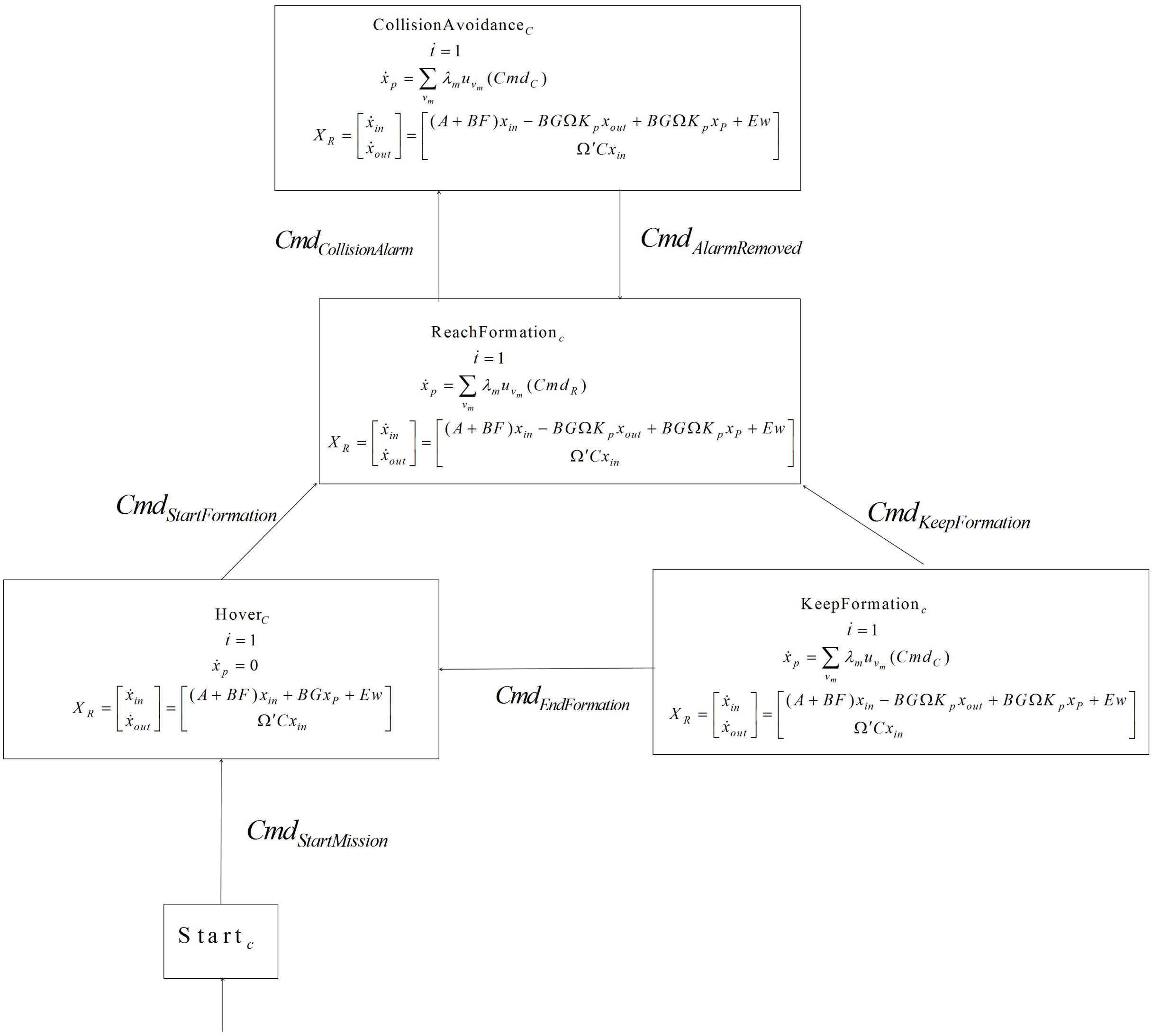

CollisionAvoidance$_C$
$i=1$
$\dot{x}_p = \sum_{v_m} \lambda_m u_{v_m}(Cmd_C)$
$X_R = \begin{bmatrix}\dot{x}_{in}\\ \dot{x}_{out}\end{bmatrix} = \begin{bmatrix}(A+BF)x_{in} - BG\Omega K_p x_{out} + BG\Omega K_p x_P + Ew\\ \Omega' C x_{in}\end{bmatrix}$
$Cmd_{CollisionAlarm}$
$Cmd_{AlarmRemoved}$
ReachFormation$_c$
$i=1$
$\dot{x}_p = \sum_{v_m} \lambda_m u_{v_m}(Cmd_R)$
$X_R = \begin{bmatrix}\dot{x}_{in}\\ \dot{x}_{out}\end{bmatrix} = \begin{bmatrix}(A+BF)x_{in} - BG\Omega K_p x_{out} + BG\Omega K_p x_P + Ew\\ \Omega' C x_{in}\end{bmatrix}$
$Cmd_{StartFormation}$
$Cmd_{KeepFormation}$
Hover$_C$
$i=1$
$\dot{x}_p = 0$
$X_R = \begin{bmatrix}\dot{x}_{in}\\ \dot{x}_{out}\end{bmatrix} = \begin{bmatrix}(A+BF)x_{in} + BGx_P + Ew\\ \Omega' C x_{in}\end{bmatrix}$
$Cmd_{EndFormation}$
KeepFormation$_c$
$i=1$
$\dot{x}_p = \sum_{v_m} \lambda_m u_{v_m}(Cmd_C)$
$X_R = \begin{bmatrix}\dot{x}_{in}\\ \dot{x}_{out}\end{bmatrix} = \begin{bmatrix}(A+BF)x_{in} - BG\Omega K_p x_{out} + BG\Omega K_p x_P + Ew\\ \Omega' C x_{in}\end{bmatrix}$
$Cmd_{StartMission}$
Start$_c$

Figure 14. The composed system for the formation mission.

13. Using the hybrid composition operator, the layers of this hierarchy can be synchronized as the discrete commands on the higher layers make a synchronize transition in the lower layers. Furthermore, using this composition operator, the closed loop system can be achieved. For instance, the regulation layer with the motion planning layer for the online path planning, and the supervision for the formation control have been composed and the result is shown in Fig. 14. This composed system gives an insight of the closed loop system for this controlled system. Also, since most of the hybrid tools are developed for a single layer hybrid system, for this composed hybrid model of the system we can apply hybrid analysis tools such as model checking (Henzinger et al. 1997) and verification (Alur et al. 1993).

4 Implementation and Experimental Results

The proposed control structure is implemented in the avionic system of thise NUS UAV and several flight tests have been conducted to evaluate this control hierarchy.

First, the supervision layer for the off-line path generation (Fig. 10) together with the motion planning layer discussed in Section 3.3.1 have been used to conduct a flight test. The assigned mission in this experiment is composed of several successive tasks. It starts with 8 meters ascending, followed by hovering, zigzag path tracking, velocity control, circle path tracking, hovering, and 8 meters descending. The mission ends with hovering. The state variables of the UAV are shown in Fig. 15. The control signals recorded in the flight test are shown in Fig. 16. To have a better sense of the system performance, the reference signals and actual flight test data in *Zigzag Path Tracking*, *Velocity Control*, and *Circle Path Tracking* modes are presented in Fig. 18. As it can be seen in this figure, the system is able to follow the given trajectory. Small deviations from the reference path could be due to the wind disturbances (around 2 to 3 m/s in the horizontal plane) and GPS signal errors as the position accuracy of GPS is 3m CEP. The video of this flight test is available at http://uav.ece.nus.edu.sg/video/hybridswitching2.avi.

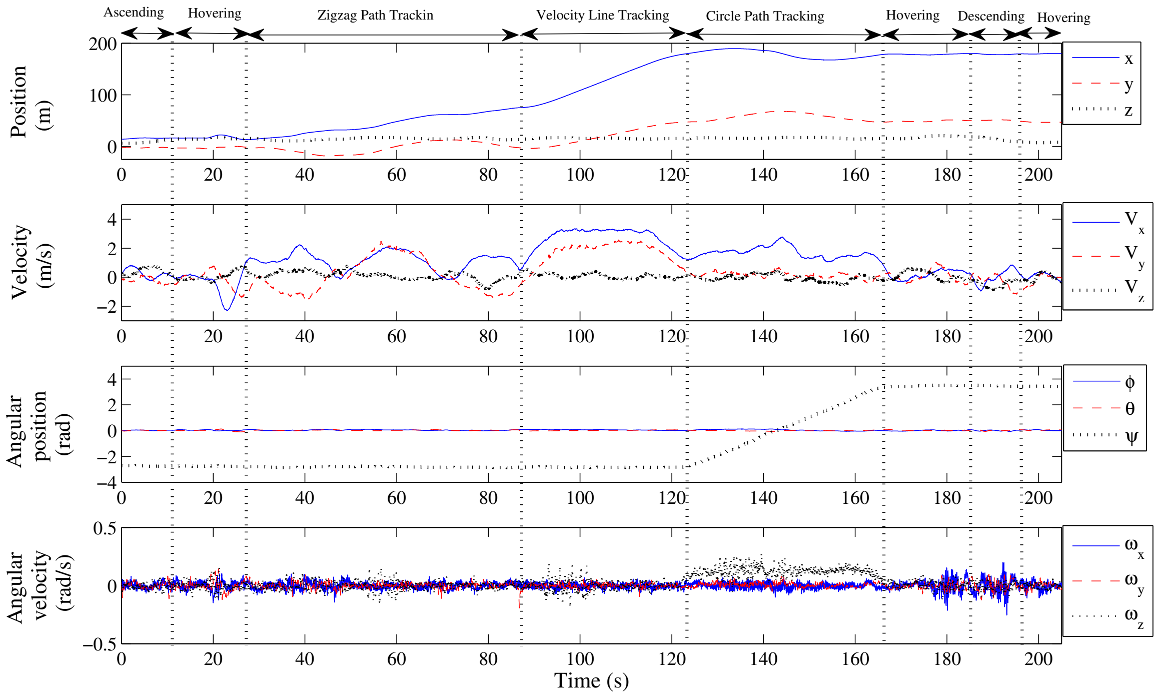


Figure 15. State variables of the UAV.

In the second experiment, we have implemented this control hierarchy in the avionic system of a follower UAV which is involved in a formation mission. For this experiment, we have used the supervision layer and the motion planning layer shown in Fig. 9 and 11, respectively. In this experiment, the leader follows a line path and the follower should reach and keep the formation. The follower is initially located at a point that has a relative distance of $(dx, dy) = (-17.8, 11.4)$ with respect to the desired position. Starting form a hovering mode, then the leader issues the start command, and after 17 Sec, the follower reaches the formation that has a relative distance of (dx, dy) = (-5, -15) with respect to the leader (Fig. 18(a)). The position of both follower UAV and the leader UAV are shown in Fig. 18(b). The video of this flight test is available at http://uav.ece.nus.edu.sg/video/hybridswitching2.avi.

5 Conclusion

In this paper we developed a hierarchical hybrid control structure for a UAV helicopter. This hierarchy consists of three layers: the regulation layer, which is responsible for reference tracking; the motion planning layer, which is responsible for the path planning, and the supervision layer, which is responsible for the task scheduling and decision making. Each layer was modelled by an

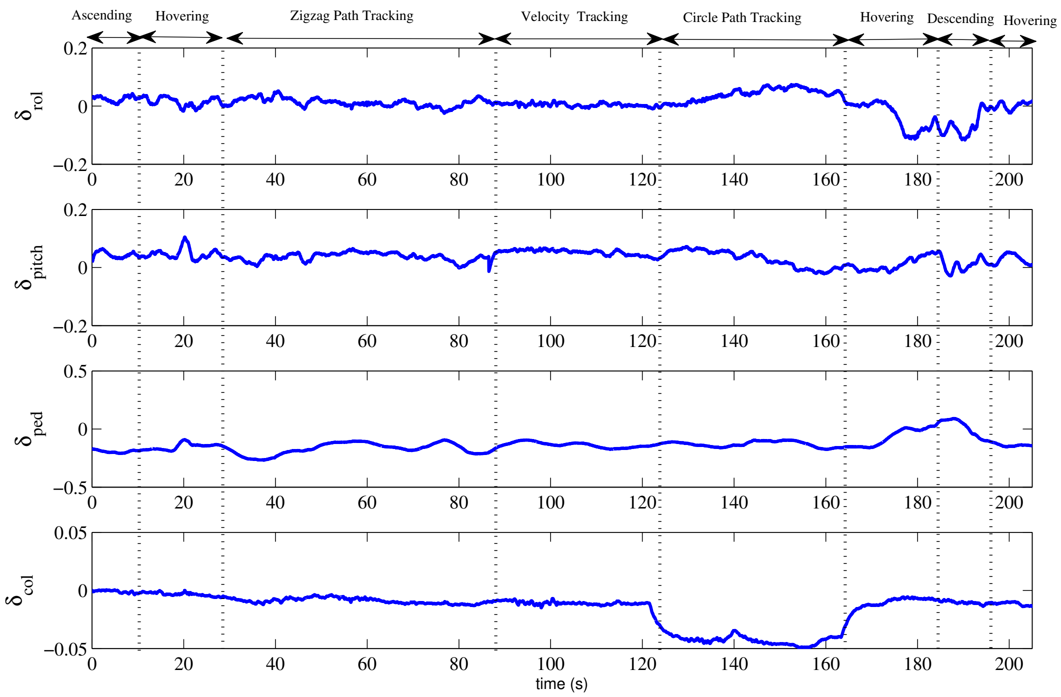
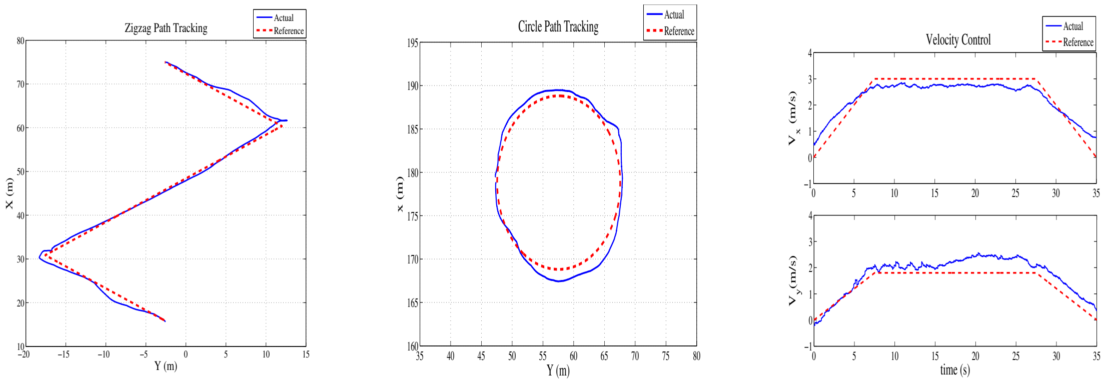


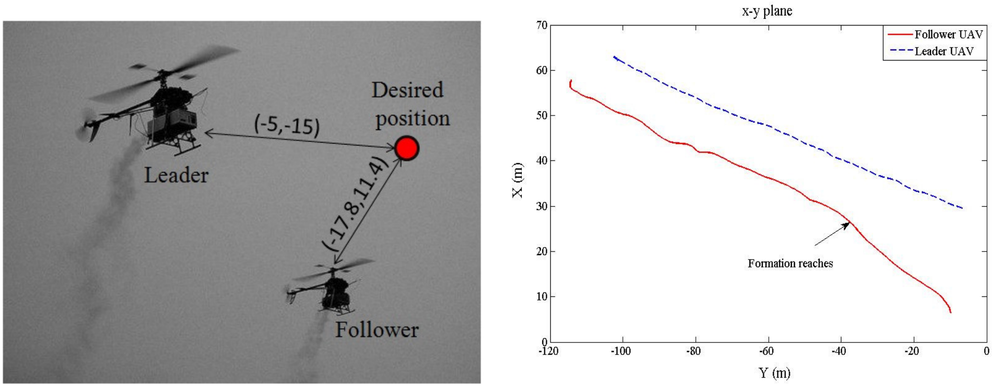
Figure 16. Control signals over the mission.



(a) (b) (c)

Figure 17. (a) Zigzag Path Tracking(b) Circle Path Tracking (c) Velocity Control.

Input/Output hybrid automaton and the discrete transitions and continuous dynamics of the system were simultaneously captured within the hybrid framework. Then, a composed hybrid operation was proposed to synchronise the layers of the control hierarchy and to obtain the whole closed-loop system. With this control scheme, two experiments were done to verify the proposed approach. In the first experiment the UAV was involved in a mission composed of several successive tasks, and in the second flight test, the UAV was involved in a formation mission as a follower UAV. Both scenarios were successfully implemented and the actual flight tests showed the effectiveness of the control structure.



(a) (b)

Figure 18. (a) The schematic of the scenario with for a leader-follower case tracking a line. (b) The position of the UAVs in the x-y plane.

References

Alur, R., Courcoubetis, C., Henzinger, T., and Ho, P. (1993), "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," (Vol. 736, eds. R. Grossman, A. Nerode, A. Ravn and H. Rischel, Springer Berlin, Heidelberg, pp. 209–229.

Alur, R., and Dill, D.L. (1994), "A theory of timed automata," *Theoretical Computer Science*, 126, 183 – 235.

Antsaklis, P., and Nerode, A. (1998), "Hybrid Control Systems: An Introductory Discussion to the Special Issue," *Automatic Control, IEEE Transactions on*, 43, 457–460.

Bayraktar, S., Fainekos, G., and Pappas, G. (2004), "Experimental cooperative control of fixed-wing unmanned aerial vehicles," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, dec., Vol. 4, pp. 4292–4298.

Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., and Pappas, G. (2007), "Symbolic planning and control of robot motion [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, 14, 61–70.

Cai, G., Chen, B.M., Dong, X., , and Lee, T.H. (2010), "Supplementary document: Linearized models of helion uav and the corresponding inner-loop controllers," Technical report, National University of Singapore, Tech. Rep.

Cai, G., Chen, B.M., Lee, T.H., and Lum, K.Y. (2008), "Comprehensive nonlinear modeling of an unmanned-aerial-vehicle helicopter," in *Proc. AIAA Conf. Guidance, Navigation and Control, Honolulu, Hawaii, USA.*

Cai, G., Chen, B.M., Peng, K., Dong, M., and Lee, T.H. (2006), "Modeling and Control System Design for a UAV Helicopter," in *Proc. 14th IEEE Mediterranean Conf. Control and Automation*, June, pp. 1–6.

Cai, G., Peng, K., Chen, B.M., and Lee, T.H. (2005), "Design and assembling of a UAV helicopter system," in *Proc. IEEE Conf. Control and Automation*, Vol. 2, pp. 697–702.

Cai, G., Feng, L., Chen, B.M., and Lee, T.H. (2008), "Systematic design methodology and construction of UAV helicopters," *Mechatronics*, 18, 545–558.

Chen, B.M., *ROBUST AND* H_∞ *CONTROL*, Springer, New York, London (2000).

Dong, M., Chen, B.M., Cai, G., and Peng, K. (2007), "Development of a real-time onboard and ground station software system for a UAV helicopter," *AIAA J. Aerosp. Comput., Inf., Commun*, 4, 933–955.

Fatemi, M., Millan, J., Stevenson, J., Yu, T., and O'Young, S. (2008), "Discrete event control of an unmanned aircraft," in *Pro- c. 9th IEEE Int. Workshop on Dis- crete Event Systems*, pp. 352–357.

Findeisen, W., Bailey, F.N., Brdeys, M., Malinowski, K., Tatjewski, P., and Wozniak, J., *Control and coordination in hierarchical systems*, Internat. Ser. Appl. Syst. Anal., Chichester: Wiley

(1980).

Frazzoli, E., Dahleh, M., and Feron, E. (2000), "Robust hybrid control for autonomous vehicle motion planning," in *Proc. 39th IEEE Conf. Decision and Control*, Vol. 1, pp. 821–826.

Gillula, J., Huang, H., Vitus, M., and Tomlin, C. (2010), "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Proc. IEEE Conf. Robotics and Automation*, May, pp. 1649–1654.

Henzinger, T.A., Ho, P.H., and Wong-Toi, H. (1997), "HYTECH: amodel checker for hybrid systems," *International Journal on Software Tools for Technology Transfer (STTT)*, 1, 110–122, 10.1007/s100090050008.

Johansson, K.H., "Introduction to hybrid systems," Lecture notes, Department of Signals, Sensors and Systems. Royal Institute of Technology (2005).

Karimoddini, A., Cai, G., Chen, B., Lin, H., and Lee, T. (2011), Vol. Advances in Flight Control Systems, "12," *Hierarchical control design of a UAV helicopter*, INTECH, pp. 239–260.

Karimoddini, A., Cai, G., Chen, B., Lin, H., and Lee, T. (2010), "Multi-layer flight control synthesis and analysis of a small-scale UAV helicopter," in *Proc. IEEE Conf. Robotics Automation and Mechatronics*, Jun., pp. 321–326.

Karimoddini, A., Dong, X., Cai, G., Feng, L., Lin, H., Chen, B.M., and Lee, T.H. (2011), "A composed hybrid structure for the autonomous flight control of unmanned helicopters," in *In Proceedings of the 18th IFAC World Congress*, Aug, pp. 2632–2637.

Karimoddini, A., Lin, H., Chen, B., and Lee, T.H. (2009), "Developments in hybrid modeling and control of Unmanned Aerial Vehicles," in *Proc. IEEE Conf. Control and Automation*, pp. 228–233.

Karimoddini, A., Lin, H., Chen, B.M., and Lee, T.H. (2012), "Hybrid three-dimensional formation control for unmanned helicopters," *Automatica*, 49, 424–433.

Latombe, J., *Robot motion planning*, Springer (1990).

Liu, J., Liu, X., Koo, T., Sinopoli, B., Sastry, S., and Lee, E. (1999), "A hierarchical hybrid system model and its simulation," in *Proc. 38th IEEE Conf. Decision and Control*, Vol. 4, pp. 3508–3513.

Lynch, N., Segala, R., and Vaandrager, F. (2001), "Hybrid I/O Automata Revisited," in *Proceedings Fourth International Workshop on Hybrid Systems: Computation and Control (HSCC'01*, Springer-Verlag, pp. 403–417.

Lynch, N., Segala, R., and Vaandrager, F. (2003), "Hybrid I/O automata," *Information and Computation*, 185, 105–157.

Mesarović, M.D., Macko, D., and Takahara, Y., *Theory of Hierarchical, Multilevel Systems*, Vol. 68 of *Mathematics in Science and Engineering*, Academic Press (1970).

Naldi, R., Marconi, L., and Gentili, L. (2009), "Robust takeoff and landing for a class of aerial robots," in *Proc. 48th IEEE conf. Decision and Control held jointly with 28th Chinese Control Conf. (CDC/CCC)*, pp. 3436–3441.

Ollero, A., and Merino, L. (2004), "Control and perception techniques for aerial robotics," *Annual Reviews in Control*, 28, 167–178.

Peng, K., Cai, G., Chen, B.M., Dong, M., and Lee, T.H. (2006), "Comprehensive Modeling and Control of the Yaw Dynamics of a UAV Helicopter," in *Control Conference, CCC 2006. Chinese*, aug., pp. 2087–2092.

Peng, K., Cai, G., Chen, B.M., Dong, M., Lum, K.Y., and Lee, T.H. (2009), "Design and implementation of an autonomous flight control law for a UAV helicopter," *Automatica*, 45, 2333–2338.

Ramadge, P., and Wonham, W. (1989), "The control of discrete event systems," *Proceedings of the IEEE*, 77, 81–98.

Rashid, S., and Lygeros, J., "Hybrid systems: modeling, analysis and control - open hybrid automata and composition," Lecture notes, University of California at Berkley (1999).

Schouwenaars, T., Mettler, B., Feron, E., and How, J. (2003), "Hybrid Architecture for Full-Envelope Autonomous Rotorcraft Guidance," in *American Helicopter Society 59th Annual*

Forum, Arizona, May.

Sobh, T., and Benhabib, B. (1997), "Discrete event and hybrid systems in robotics and automation: an overview," *Robotics Automation Magazine, IEEE*, 4, 16–19.