Design of Mode-to-Mode Fuzzy Controllers

Freeman Rufus and George Vachtsevanos*
Intelligent Controls Systems Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0250

The mode-to-mode transition problem involves taking initial states in the start mode to the equilibrium point of the goal mode, where each mode of operation corresponds to an operating regime about an equilibrium point. Like the problem of dynamic transitions between various equilibria, there is no consistent theory that deals with the mode-to-mode transition problem. This paper presents a method of designing mode-to-mode controllers by blending the start and goal mode controllers. The blending gains are determined by the phase portrait assignment algorithm. The phase portrait assignment algorithm is a systematic technique that uses dynamic programming and center-point cell mapping to design fuzzy logic controllers. A hover mode to forward flight mode controller for a small-scale helicopter is synthesized to illustrate the design methodology. Simulation results show that the controller is able to transition stably from hover to forward flight. Finally, sensitivity analysis of the hover to forward flight controller is performed for small parameter perturbations. © 2000 John Wiley & Sons, Inc.

1. INTRODUCTION

Large-scale dynamical systems such as airplanes, helicopters, automobiles, and industrial processes have several operating modes that require stable transitions between them. An operating mode is considered to be a region about the mode's equilibrium point in which linear approximations are valid. The mode-to-mode problem involves taking initial states in the start mode to the equilibrium point of the goal mode. The mode-to-mode problem can be considered as a transition problem between the two modes' equilibrium points with the assumption that the controller designed for the latter problem takes points in a small neighborhood of the start mode's equilibrium point to a neighborhood of the goal mode's equilibrium point. Three methods that can be applied to the design of mode-to-mode controllers are gain scheduling, phase-space control system design, and blending mode controllers 1–3.

Although, there is no consistent theory that deals with dynamic transitions between various equilibria, gain scheduling has been used to design equilibrium-to-equilibrium controllers. The technique of gain scheduling approximates a

*Author to whom correspondence should be addressed. e-mail: george.vachtsevanos@ee.gatech.edu.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 15, 657–685 (2000)

nonlinear control system with a piecewise-linear one and designs a linear controller for each linear piece. In gain scheduling, the transition between equilibria is governed normally by an auxiliary scheduling variable.[1,4] The rule of thumb to ensure stability is that the scheduling variable should vary slowly with respect to the states. The disadvantages associated with gain scheduling include a reliance on a long trial-and-error design process, a lack of adaptability to on-line variations, and poor robustness to uncertainties. Gain scheduling has been applied to process control[5] and the design of flight control systems for high performance aircraft.[6] The gain scheduling procedure is generally as follows: (1) for the start and goal equilibrium points, a local controller is designed using linear techniques; (2) then a viable path is chosen between the two equilibria depicting the desired transition between them; (3) afterward, N points are chosen along the path between equilibrium$_{\text{start}}$ to equilibrium$_{\text{goal}}$ and linear models of the system are created at these points; a linear controller is designed for each linear model having gain K_n, where $1 \leq n \leq N$; (4) finally, a multidimensional interpolation technique is used to interpolate between the gains K_n, for the equilibrium$_{\text{start}}$ to equilibrium$_{\text{goal}}$ transition.

The second approach of designing mode-to-mode controllers is based on phase-space design techniques. The Phase Space Navigator and Perfect Moment are two programs that use knowledge about the phase-space dynamics of a nonlinear system to synthesize nonlinear controllers between specified phase-space points. The Phase Space Navigator[2] finds optimal phase-space paths from an initial state to a goal state that consist of a sequence of path segments connected at intermediate points where the control parameter changes. It has two main modules, a planning module and a tracking module. The former synthesizes a reference trajectory based on the dynamics of the nominal model. The latter follows the reference trajectory and reactively corrects for local deviations. The Perfect Moment program uses the phase portraits constructed by varying the control parameter to synthesize a segmented path between two specified phase-space points. The search algorithm first finds a gross path between the regions surrounding the origin and destination, then iteratively reinvokes the mapping module on finer scales and uses those maps to find segments that connect the ends of the gross path to the origin and destination. Afterward, the system is routed along the segmented path by appropriate switches of the control parameter at the segment junctions. These two programs can be applied to the design of mode-to-mode controllers by specifying the equilibria of the two modes as phase-space points in which to form a segment.

In Ref. 3, a method of heuristically blending mode controllers by fuzzy logic was discussed in designing a forward flight mode to hover mode controller. Two methods were used in blending the mode controllers: (1) blending the output of the two mode controllers or (2) blending the reference setpoint of each mode controller. It was determined that (1) the outputs (actuator signals) of each mode controller should be blended if the coupling between the modes is small; and (2) the inputs (set points) of each mode controller should be blended if the coupling between the modes is high.

In this paper, a method is presented to design mode to mode fuzzy controllers. The method is based upon the blending of the output of the start and goal mode controllers using fuzzy logic. The blending weights for each output are determined by an algorithm called the Phase Portrait Assignment Algorithm (PPAA). Finally, this method is illustrated in the design of a hover mode to forward flight controller for a small-scale helicopter.

The Phase Portrait Assignment Algorithm (or PPAA) has been applied to the design of fuzzy logic controllers (FLCs) for an automotive engine,[7] the hover mode of a small-scale helicopter,[8,9] the single link robot arm,[10] and the nonlinear inverted pendulum problem.[11] The stability of the FLCs designed by the PPAA has been analyzed in Refs. 10, 12, and 13. The PPAA is similar to Hsu's approach[14] except that it produces an optimal control table containing fuzzy, rules.[11] The algorithm partitions both the state space and control variable space into cells approximated by their center points. The transition from one cell to another is accomplished via a center-point mapping of the cells under the applied control action. Simulation is performed until all the cells in the state space have been examined for all possible control action cells. Then, a modified A^* algorithm is applied to a database containing simulation data to create an optimal control table. Finally, fuzzy control rules are generated from the optimal table entries by fuzzifying the borders of cells in the state and control action spaces.[11]

The paper is organized as follows. In Section 2 the design of mode-to-mode controllers is described. This includes a brief discussion of cell-to-cell mapping, the phase portrait assignment algorithm, and sensitivity analysis. A heuristic model of a small scale helicopter is presented in Section 3, along with the design parameters for the hover to forward flight controller. In Section 4, a hover mode to forward flight mode controller is simulated with the small-scale helicopter. Afterward, sensitivity analysis is performed on the controller to determine parameters for optimum sensitivity. Finally, Section 5 contains conclusions.

2. DESIGN OF MODE-TO-MODE CONTROLLERS

A. Cell-to-cell Mapping

1. Introduction

Cell-to-cell mapping (or cell mapping)[15] is a powerful computational technique for analyzing the global behavior of nonlinear dynamical systems. It simplifies the task of analyzing a continuous phase space by partitioning it into a finite number of disjoint cells and approximating the system trajectories as cell transitions. The cell mapping is used mainly to determine stable periodic solutions and the corresponding domains of attraction of nonlinear systems. The cell mapping method is able to computationally perform global analysis of both low-order and high-order systems. However, the method has two main drawbacks: (1) the time required to obtain a complete characterization of the state

space is extremely high and (2) the memory requirements to manage all the data collected grows exponentially with the system dimension and with the desired accuracy.

The cell mapping method has been used to analyze the global properties of coupled van der Pol equations[15] in order to determine their limit cycles. The method has also been used to determine the global behavior of fuzzy dynamical systems.[16–18] Finally, cell mapping was extended by Hsu to optimal control problems.[14,19]

2. *Cell mapping concepts*

The cell mapping technique partitions a continuous state space of a system into a finite number of disjoint cells. Cells are created by dividing each axis x_i of an n-dimensional state space into intervals of size h_i; each interval is denoted by an integer z_i such that it contains all x_i satisfying $(z_i - 1/2)h_i \leq x_i \leq (z_i - 1/2)h_i$. A cell $\mathbf{z}$ is defined as an n-tuple of intervals $[z_1, z_2, \ldots, z_n]^T$. The union of all cells $\mathbf{z}$ yields an integer-valued n-dimensional cell space $\mathbf{Z}$. The portions of the state space lying outside the region of interest are lumped together in a single cell called the sink cell.

Cell mapping abstracts the system states in a cell $\mathbf{z_i}$ with the cell center-point $\mathbf{z_i^c}$. This allows real (or point-to-point) trajectories in the state space to be approximated by cell trajectories in the corresponding cell space. Figure 1 illustrates the approximation scheme for a real trajectory for the discrete time

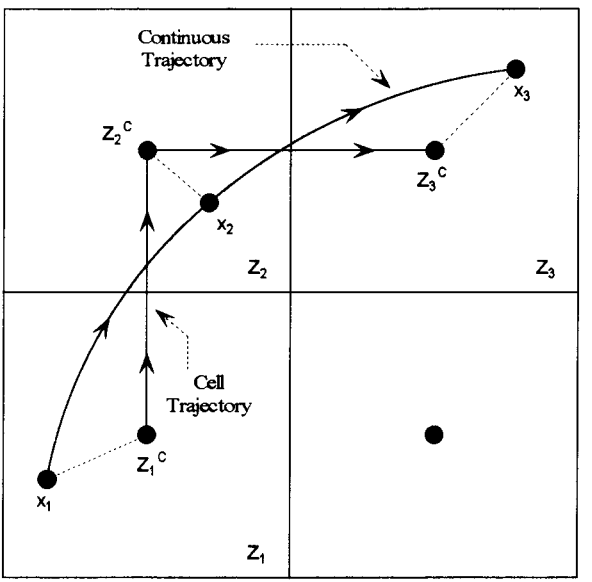


Figure 1. Cell space approximation technique.

system $\mathbf{x_{k+1}} = \mathbf{g}(\mathbf{x_k})$:

$$\mathbf{x_1} \rightarrow \mathbf{x_2} \rightarrow \mathbf{x_3}$$

The initial point $\mathbf{x_1}$ in the trajectory that lies in cell $\mathbf{z_1}$ is abstracted by the cell center-point $\mathbf{z_1^c}$. Then, $\mathbf{x_2'} = \mathbf{g}(\mathbf{x_1^c})$, which lies in cell $\mathbf{z_2}$, is abstracted by $\mathbf{z_2^c}$. Finally, $\mathbf{x_3'} = \mathbf{g}(\mathbf{x_2^c})$ in cell $\mathbf{z_3}$ is abstracted by $\mathbf{z_3^c}$. This procedure yields the cell trajectory:

$$\mathbf{z_1} \rightarrow \mathbf{z_2} \rightarrow \mathbf{z_3}.$$

Note that in order to minimize cell mapping errors, it is important that states $\mathbf{x_2'}$ and $\mathbf{x_3'}$ be located as close to each other as possible and lie in the same cells as the real trajectory states $\mathbf{x_2}$ and $\mathbf{x_3}$, respectively.

A cell mapping is formulated as a cell state space function:

$$\mathbf{C}: \mathbf{Z} \rightarrow \mathbf{Z}$$

Using this function, a k-step trajectory emanating from cell $\mathbf{z}$ is written as a cell sequence:

$$\mathbf{z} \rightarrow \mathbf{C}(\mathbf{z}) \rightarrow \mathbf{C}(\mathbf{C}(\mathbf{z})) \rightarrow \cdots \rightarrow \mathbf{C^k}(\mathbf{z})$$

A periodic motion with period K is a sequence of K distinct cells $\mathbf{z}_m$, $m = 0, \ldots, K-1$, satisfying the condition

$$\mathbf{z_m} = \mathbf{C^m}(\mathbf{z}) \qquad \text{and} \qquad \mathbf{z} = \mathbf{C^K}(\mathbf{z})$$

An *equilibrium cell* $\mathbf{z_e}$ is a cell that maps to itself, i.e.,

$$\mathbf{z_e} = \mathbf{C}(\mathbf{z_e})$$

Such a behavior results in a periodic motion with period 1. The r-step domain of attraction of a period motion is the set of all cells that are within r-steps of the periodic motion.

A cell map is constructed using an "unraveling algorithm"[15] to compute cell transitions and to identify all existing periodic motions and domains of attraction. The key to the unraveling algorithm is the computation of an image cell or one-step transition cell for each cell $\mathbf{z}$ within a specified "cell processing time period" t_m. The procedure assumes that all points in a cell $\mathbf{z}$ end up in the image of cell $\mathbf{z}$ within the cell processing time period t_m. Readers are referred to Ref. 15 for a detailed presentation of the unraveling algorithm.

Based on the computed image cells and cell trajectories, each cell $\mathbf{z}$ is assigned three numbers that characterize the dynamical behavior of the system:

(1) A group number $G(\mathbf{z})$.
(2) A step number $S(\mathbf{z})$.
(3) A periodicity number $P(\mathbf{z})$.

The same group number $G(\mathbf{z})$ is assigned to all cells in a periodic motion or domain of attraction of cell $\mathbf{z}$. The step number $S(\mathbf{z})$ of a cell $\mathbf{z}$ indicates the number of cell transitions needed to transit from cell $\mathbf{z}$ to a cell in a periodic motion. The period of a periodic motion is expressed by the periodicity number $P(\mathbf{z})$. The concepts of group number, step number, and periodicity number are illustrated in Figure 2. It is important to note that cells mapping to states lying outside the region of interest (sink cell) are identified with the sink cell, i.e., they belong to the same group as the sink cell.

B. Phase Portrait Assignment Algorithm

The phase portrait assignment algorithm is a systematic design procedure for fuzzy linguistic controllers that uses a hybrid methodology incorporating cell-to-cell mappings and AI algorithms. This method can be applied to a class of nonlinear systems which are subjected to dynamic or parametric disturbances. These issues are addressed by dividing the uncertain domain of interest into a finite number of manageable quantities and by assigning fuzzy sets to each linguistic representation; then, the relations that govern the control objectives may be easily derived since we are dealing with quantitative objectives (vector fields of invariant or switching manifolds) of an infinite point space in terms of qualitative reasoning that is expressed as a finite set of rules.

1. Method

A detailed description of the phase portrait assignment algorithm and its fuzzy hypercube implementation platform may be found in Refs. 7, 20, and 21. The following is an outline of the PPAA method:

- The state and control variable spaces are partitioned into cell-groups, where the size of each cell-group is determined according to accuracy and tolerance specifications, and the prevailing time and memory constraints.

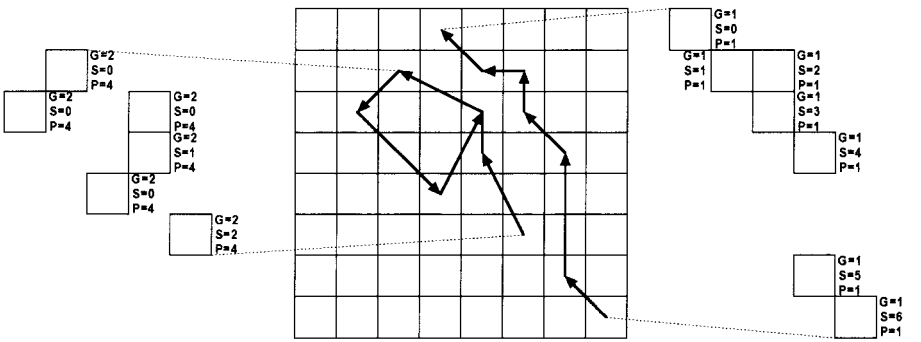


Figure 2. Cell characterization parameters.

- The center points of all cell-groups are chosen to anticipate the trajectories from one subspace to another. For each cell-group the center point is considered as the initial state for the simulation. Also, for each cell-group a set of admissible control inputs are applied, and for each control input which is held constant, a simulation run is performed until the system enters another cell-group or exceeds the maximum simulation time.
- For each complete simulation run of a constant input the time required for the transition is stored, as well as the control energy and the Euclidean distance of the state from the equilibrium point. This effort yields a large database with information about the global dynamical behavior of the system.
- The next step is to search the accumulated data file and come up with the best control in terms of minimum time, energy, or error that causes the system starting in any of the cell-groups to be driven to the target cell-group. The search procedure is based on a modified A^* algorithm. The algorithm is guaranteed to terminate in a finite number of steps and to find the best path from any node to the equilibrium node, provided such a path exists.
- Sink cells and periodic cells are thus eliminated before the controller rule based is derived. Ultimately, each cell transition in an optimal trajectory becomes a fuzzy control rule. Thus, the number of control rules is equal to the number of final cell-groups for which an optimal path exists to the target cell.
- The borders of cell-groups in the cell state space and control action space are fuzzified to generate smooth control actions. Linguistic values defined by functions with centers located at the center-points of cell-groups and ends located at the center-points of the adjacent cell-groups are employed. The antecedents and consequents of the final fuzzy control rules are defined using the linguistic values for the cell state space and control action cell space, respectively. The rules are fired using a standard fuzzification–inference–defuzzification algorithm.

2. *Stability*

Although fuzzy logic controllers have been applied to a variety of industrial applications, fuzzy logic has yet to be widely accepted in the control engineering discipline. This lack of acceptance is due primarily to the following reasons: (1) failure to address the problem of stability through analytical methods and (2) very few tools to evaluate the stability properties of fuzzy logic controllers. Since fuzzy controllers designed by the PPAA method incorporate information about the global dynamical behavior of the nonlinear system by cell mapping, the claim of asymptotic stability was put forth. In Ref. 10, the stability of FLCs designed by PPAA was analyzed using Lyapunov's direct method. The Lyapunov analysis was carried out on the basis of the fact that the PPAA forces the vector field of the system's differential equation to be directed toward a small neighborhood containing the equilibrium point. In Ref. 13, an exposition on the stability of the fuzzy logic controller based on the topology of the fuzzy center-point mapping was given. The two stability analysis approaches are implemented without using any model of the plant to be controlled. It should be noted that a control rulebase which is considered stable in the fuzzy domain does not have to automatically satisfy an analytical stability criterion. The control rules of FLCs designed by PPAA should be tested for stability where an approximate model is available. In Ref. 12, an input–output analysis is developed for a nearly linear plant that is being controlled by a fuzzy controller. In

this method of analysis, (1) the nearly linear assumption of the process setpoint control system is verified and (2) the linear part of the system is verified to be dissipative from an input–output point of view. Afterwards, the stability requirements are verified analytically, where possible, or via simulation. Finally, the stability of a closed-loop system (controller designed via PPAA) can be determined via a Lyapunov-like analysis:

(1) Determine a Lyapunov function $V(x, u)$ that is lower bounded.
(2) Show that $\dot{V}(x, u)$ is negative semi-definite.
(3) Show that $\ddot{V}(x, u)$ is bounded.
(4) If conditions (1)–(3) are met then $\dot{V}(x, u) \to 0$ as $t \to \infty$ implying that the closed-loop system is asymptotically stable.

C. Design: Mode-to-Mode Fuzzy Controller

1. *Statement of problem*

Given a large-scale dynamic interconnected system represented by the following state equation:

$$\dot{x} = F(x, u) \qquad x(t_0) = x_0 \qquad x \in R^n,\ u \in R^m \tag{1}$$

It is assumed that the system can be decomposed into N interconnected subsystems S_i, $i = 1, 2, \ldots, N$ where each subsystem represents the modes of operation and the ith subsystem's state equation is

$$\begin{gathered} \dot{x}_i = f_i(x_i, u_i) + \sum_{\substack{j=1 \\ i \neq j}}^{N} g_{ij}(x_j) \qquad x_i(t_0) = x_{i0} \\ x_i \in R^{n_i} \qquad u_i \in R^{m_i} \qquad g_{ij}(x_j) \in R^{n_i} \end{gathered} \tag{2}$$

where $g_{ij}(x_j)$ represents the coupling term due to the jth subsystem. How do we design a fuzzy controller that transitions from a starting mode to a desired mode of operation?

2. *Approach*

Let mode$_p$ and mode$_q$ denote the pth and the qth subsystem, respectively. Using the PPAA, a mode$_p$-to-mode$_q$ transitional controller is designs with knowledge about the states of mode$_p$ and mode$_q$, and the outputs of the mode$_p$-and-mode$_q$ controllers. The outputs of the mode$_p$-to-mode$_q$ controller are determined by blending the individual outputs of the mode$_p$ and mode$_q$ controllers. The blending weights for each mode controller are determined by the PPAA. The following is an outline for the design of the mode$_p$-to-mode$_q$ controller.

a. Design local controllers for mode$_p$ and mode$_q$

For the operating modes mode$_p$ and mode$_q$, models are constructed that capture their local dynamics. Afterwards, linear or nonlinear state feedback controllers are designed for the two operating modes. The controllers are designed to regulate initial states belonging to mode$_p$ and mode$_q$ to the equilibrium point of mode$_p$ and mode$_q$, respectively. Figure 3 shows the feedback structure of the mode controllers, where the equilibrium point of mode$_p$ and mode$_q$ is the desired command.

Equation 3 and 4 are the dynamical equations of mode$_p$ and mode$_q$, respectively:

$$\begin{aligned} &\dot{x}_p = f_p(x_p, u_p) \qquad x_p(t_0) = x_p^0 \qquad x_p \in R^{n_p},\ u_p \in R^{m_p} \\ &u_p = \varphi_p(x_p - x_p^*) \end{aligned} \tag{3}$$

$$\begin{aligned} &\dot{x}_q = f_q(x_q, u_q) \qquad x_q(t_0) = x_q^0 \qquad x_q \in R^{n_q},\ u_q \in R^{m_q} \\ &u_q = \varphi_q(x_q - x_q^*) \end{aligned} \tag{4}$$

where x_p^* and x_q^* denote the equilibrium points of mode$_p$ and mode$_q$, respectively; φ_p and φ_q are linear or nonlinear functions of x_p and x_q.

b. Model combined dynamics of mode$_p$ and mode$_q$

A model of the dynamical system is constructed that incorporates the dynamics of mode$_p$, mode$_q$ and their corresponding coupling dynamics:

$$\dot{x}_{pq} = f(x_{pq}, u_{pq}) \qquad x_{pq}(t_0) = x_{pq}^0 \qquad x_{pq} \in R^{n_{pq}},\ u_{pq} \in R^{m_{pq}} \tag{5}$$

where x_{pq} is the vector of distinct elements of $[x_p^T \quad x_q^T]^T$; the vector u_{pq} is the vector of distinct elements of $[u_p^T \quad u_q^T]^T$; x_p and x_q denote the states of mode$_p$ and mode$_q$, respectively; u_p and u_q denote the control inputs of mode$_p$ and

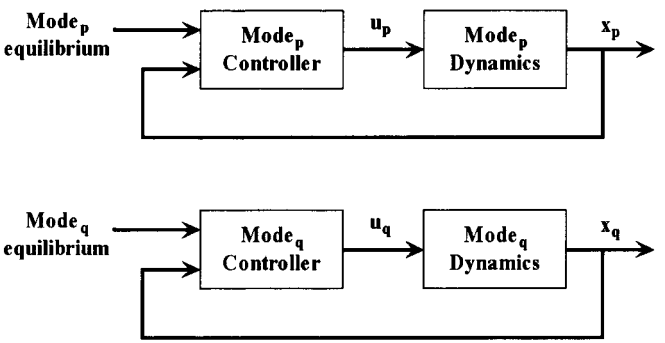


Figure 3. Block diagram of mode$_p$ and mode$_q$ feedback systems.

mode$_q$, respectively. Given the local models of mode$_p$ and mode$_q$, determining a model of the combined dynamics of mode$_p$ and mode$_q$ is really a problem of modeling the coupling terms of Eq. 2: $g_{pq}(x_q) \in R^{n_p}$ and $g_{qp}(x_p) \in R^{n_q}$. The coupling dynamics can be modeled by invoking first principles or produced through experimental data. In the latter case, neural network, fuzzy or neuro-fuzzy models can be constructed.

c. Determine the region of interest and the partition of the x_{pq} phase space

A region of interest of the phase space belonging to x_{pq} is chosen such that it contains the operating points of mode$_p$, mode$_q$ and any corresponding transitional paths between the two modes, if they exist. Constraints on x_{pq} can be used to determine the maximum region of interest. Choosing a large region of interest will generally result in more partitions for each state of x_{pq} to meet a desired cell resolution; and more partitions for each state of x_{pq} will lead to more cells in the phase space, which will lead to a longer time to perform the phase space simulation. However, choosing a small region of interest increases the chance of excluding possible paths between mode$_p$ and mode$_q$. Therefore, the region of interest should be determined on the basis of the system's dynamics. The region of interest of the phase space is described by the following inequality:

$$(x_{pq,\min})_i < (x_{pq})_i < (x_{pq,\max})_i \qquad \text{for } i = 1, \ldots, n_p \tag{6}$$

where $(x_{pq})_i$ is the ith state of x_{pq}; $(x_{pq,\min})_i$ and $(x_{pq,\max})_i$ denote the minimum and maximum values of $(x_{pq})_i$, respectively. The phase space is partitioned to have a desired cell resolution such that (i) a tolerance specification is met and (ii) the equilibrium points of mode$_p$ and mode$_q$ are in different cells near the center of their respective cells. The following describes the tolerance specification for the ith state of x_{pq}:

$$(\text{tolerance})_i \geq \frac{(x_{pq,\max})_i - (x_{pq,\min})_i}{N_i + 1} \tag{7}$$

where N_i is the number of interval divisions along the ith state of x_{pq}. The following expressions determine the relationship of N_i. $(x_{pq,\min})_i$ and $(x_{pq,\max})_i$ such that the equilibrium points of mode$_p$ and mode$_q$ can be placed in different cells near the center of their respective cells:

$$\begin{bmatrix} (x_{pq,\min})_i \\ (x_{pq,\max})_i \end{bmatrix} = (N_i + 1) \left\{ \begin{bmatrix} N_i + 1 + l_i^p & l_i^p \\ N_i + 1 + l_i^q & l_i^q \end{bmatrix} \right\}^{-1} \cdot \begin{bmatrix} \text{center}_i^p \\ \text{center}_i^q \end{bmatrix}$$

$$|\text{center}_i^p - \text{center}_i^q| \geq \frac{(x_{pq,\max})_i - (x_{pq,\min})_i}{N_i + 1} \tag{8}$$

where center_i^p is the ith component of mode_p's equilibrium point represented in x_{pq} space; center_i^q is the ith component of mode_q's equilibrium point represented in x_{pq} space; l_i^p denotes the interval of $(x_{pq})_i$ in which center_i^p lies; l_i^q denotes the interval of $(x_{pq})_i$ in which center_i^q lies; $1 \le l_i^p, l_i^q \le N_i$, $|l_i^p - l_i^q| \ge 1$, and $1 \le i \le n_{pq}$. The number of cells in the region of interest of x_{pq} is

$$\prod_{i=1}^{n_{pq}} N_i \tag{9}$$

d. Determine the region of interest and partition of the blending weights phase space

Assuming the outputs of the mode_p-and-mode_q controllers are linearly blended, the mode_p-to-mode_q controller C_{pq} will have the form

$$C_{pq}(\cdot) = K_p(x_{pq}) \cdot u_p + K_q(x_{pq}) \cdot u_q \tag{10}$$

where $K_p(x_{pq})$ and $K_q(x_{pq})$ are the blending matrices; $\dim(K_p(x_{pq})) = m_{pq} \times m_p$ and $\dim(K_q(x_{pq})) = m_{pq} \times m_q$; m_p elements of $K_p(x_{pq})$ are nonzero and m_q elements of $K_q(x_{pq})$ are nonzero. Let $k^{p1}, k^{p2}, \ldots, k^{pm_p}$ and $k^{q1}, k^{q2}, \ldots, k^{qm_q}$ denote the nonzero elements of $K_p(x_{pq})$ and $K_q(x_{pq})$, respectively. These nonzero elements of $K_p(x_{pq})$ and $K_q(x_{pq})$ are partitioned into admissible control inputs having the ranges

$$\begin{aligned} k_{\min}^{pr} \le k^{pr} \le k_{\max}^{pr} \qquad &\text{for} \quad r = 1, \ldots, m_p \\ k_{\min}^{qr} \le k^{qr} \le k_{\max}^{qr} \qquad &\text{for} \quad r = 1, \ldots, m_q \end{aligned} \tag{11}$$

satisfying the following conditions

(a) If $(u_{pq})_i$ is a control input belonging only to mode_p then

$$(u_{pq,\min})_i < k^{pr} \cdot (u_p)_r < (u_{pq,\max})_i \tag{12}$$

where $(u_p)_r$ corresponds to $(u_{pq})_i$ for some integer r.

(b) If $(u_{pq})_i$ is a control input belonging only to mode_q

$$(u_{pq,\min})_i < k^{qr} \cdot (u_q)_r < (u_{pq,\max})_i \tag{13}$$

where $(u_q)_r$ corresponds to $(u_{pq})_i$ for some integer r.

(c) If $(u_{pq})_i$ is a control input belonging to mode_p and mode_q

$$(u_{pq,\min})_i < k^{pr} \cdot (u_p)_r + k^{qs} \cdot (u_q)_s < (u_{pq,\max})_i \tag{14}$$

where $(u_p)_r$ and $(u_q)_s$ correspond to $(u_{pq})_i$, for some integers r and s, where $(u_{pq})_i$ is the ith element of u_{pq}; $(u_{pq,\min})_i$ and $(u_{pq,\max})_i$ denote the minimum and maximum values of $(u_{pq})_i$, respectively; $(u_p)_r$ and $(u_q)_s$ denote the rth and sth elements of u_p and u_q, respectively. Let $k_{pq} = (k^{p1}, k^{p2}, \ldots, k^{pm_p}, k^{q1}, k^{q2}, \ldots, k^{qm_q})$.

The elements of k_{pq} are partitioned heuristically. The number of cells in the region of interest of k_{pq} is

$$\prod_{i=1}^{m_p+m_q} M_i \tag{15}$$

where M_i is the number of interval divisions along the ith element of k_{pq}.

e. Determine the nonzero elements of K_p and K_q in order to transition from mode$_p$ to mode$_q$

The PPAA uses the region of interest and the partition information of x_{pq} and k_{pq} to produce a fuzzy controller that can be used to blend the outputs of the mode$_p$ and mode$_q$ controllers to obtain a mode$_p$-to-mode$_q$ transition. The inputs and outputs of the fuzzy controller are x_{pq} and k_{pq}, respectively. The elements of k_{pq} are determined by the compositional rule of inference and the modified mean-of-maxima defuzzifier. The fuzzy linguistic rules for the fuzzy controller have the form

$$\begin{aligned} &\text{If } (x_{pq})_1 \text{ is } L_{(x_{pq})_1} \text{ and } \cdots \text{ and } (x_{pq})_{n_{pq}} \text{ is } L_{(x_{pq})_{n_{pq}}} \\ &\text{Then } (k_{pq})_1 \text{ is } L_{(k_{pq})_1} \text{ and } \cdots \text{ and } (k_{pq})_{m_p+m_q} \text{ is } L_{(k_{pq})_{m_p+m_q}} \end{aligned} \tag{16}$$

where $(x_{pq})_i$ and $(k_{pq})_i$ are fuzzy representations for $(x_{pq}(t))_i$ and $(k_{pq}(t))_j$, the elements of $x_{pq}(t)$ and $k_{pq}(t)$, respectively; $L_{(x_{pq})_i}$ and $L_{(k_{pq})_i}$ are linguistic variables such as positive large, negative small, and so on.

Therefore, the mode$_p$-to-mode$_q$ controller will have the form

$$C_{pq}(x_{pq}, u_p, u_q) = K_p(x_{pq}) \cdot u_p + K_q(x_{pq}) \cdot u_q$$

where the nonzero elements of K_p and K_q are determined by a [n_{pq} input, ($m_p + m_q$) output] fuzzy controller.

D. Sensitivity Analysis of Mode-to-Mode Fuzzy Controller

Consider a nonlinear autonomous system that includes the dynamics of mode$_p$, mode$_q$, and their corresponding coupling dynamics,

$$\dot{x} = f(x, u, \alpha) \qquad x(t_0) = x_0 \tag{17}$$

where x denotes the n_{pq}-dimensional state vector, $\alpha = \alpha_0 + \Delta\alpha$ is the r-dimensional parameter vector, f is an n_{pq}-dimensional vector function, $u = C_{pq}(x, \alpha_0)$ is the m_{pq} dimensional input vector generated by the mode$_p$-to-mode$_q$ fuzzy controller, α_0 is the nominal value of α, $\Delta\alpha$ is the vector of small perturbations about the nominal value.

Let $x^*(t) = x(t, \alpha_0, x_0^*)$ denote the *nominal trajectory* that satisfies

$$\dot{x}^* = f(x^*, u, \alpha_0) \qquad x^*(t_0) = x_0^*.$$

Let $x(t) = x(t, \alpha, x_0)$ denote the *perturbed trajectory* that satisfies

$$\dot{x} = f(x, u, \alpha) \qquad x(t_0) = x_0$$

Define the error between the nominal trajectory and perturbed trajectory as

$$\begin{aligned} e(t) &= e(t, \alpha, x_0) \\ &= x(t\alpha, x_0) - x(t, \alpha_0, x_0^*) \\ &= x(t) - x^*(t) \end{aligned} \tag{16}$$

Since $x^*(t)$ converges asymptotically from the equilibrium of mode$_p$ to the equilibrium of mode$_q$, then the sensitivity analysis of the mode$_p$-to-mode$_q$ controller involves examining how close the perturbed trajectory remains to the nominal trajectory when the system is subjected to small perturbations of plant parameters. The stability envelope of $x^*(t)$, shown in Figure 4, denotes the region about $x^*(t)$ in which the perturbed trajectories must be constrained in

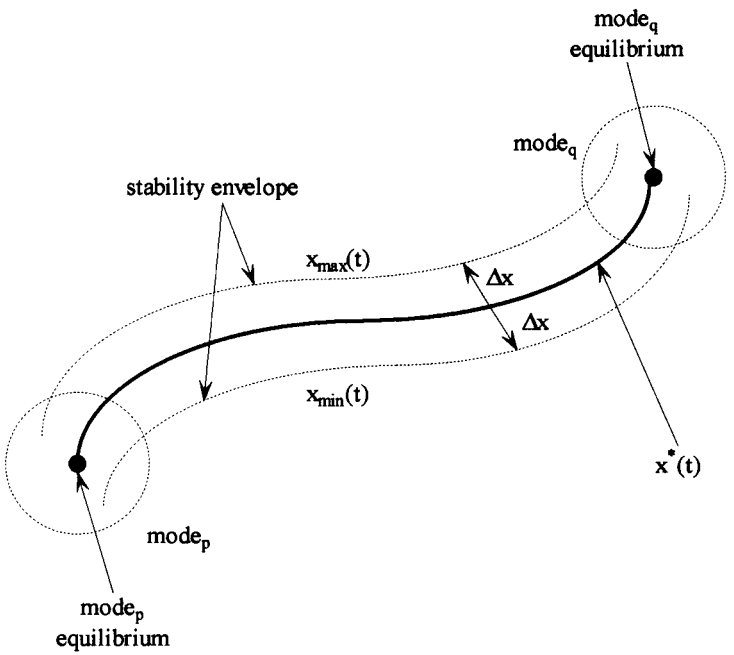


Figure 4. Stability envelope of mode$_p$-to-mode$_q$ trajectory.

order for the closed loop system to be considered stable and to have acceptable performance. The boundaries of the stability envelope satisfy the relationship

$$x_{\max}(t) = x^*(t) + \Delta x > x^*(t) > x^*(t) - \Delta x = x_{\min}(t) \tag{17}$$

where Δx is a positive vector. Therefore, the error between the nominal and perturbed trajectory satisfies the condition

$$|e_i(t)| \leq \Delta x_i \qquad \text{for} \quad i = 1, \ldots, n_{pq} \tag{18}$$

Let us define the performance measure as a lower-bounded function given by

$$V(e,t) = \int_{t_0}^{t} e^T(t) e(t)\, dt \tag{19}$$

where $e(t)$ is defined in Eq. 16 and $t \in [t_0, t_f]$. The fuzzy sensitivity of the real output function, $e(t, \alpha)$, with respect to the real parameters α_i, $i = 1, \ldots, r$, is expressed by Ref. 22:

$$S_\alpha^e(t) = \frac{1 - \mu_{\Delta e}}{1 - w_1 \mu_{\Delta \alpha_1} - w_2 \mu_{\Delta \alpha_2} - \cdots - w_r \mu_{\Delta \alpha_r}} \qquad \sum_{i=1}^{r} w_i = 1 \qquad w_i \in [0,1] \tag{20}$$

where $\mu_{\Delta \alpha_i}$ and $\mu_{\Delta e}$ are membership functions of the deviations and w_i are weights that are heuristically chosen to signify the importance of a parameter. If all parameters are equally important in a specific design, then $w_i = 1/r$ for all i.

The fuzzy sensitivity measure (FSM) is given by

$$\begin{gathered} \mathrm{FSM}(t) = \int_{t_0}^{t} \left(S_\alpha^e(t)\right)^T \left(S_\alpha^e(t)\right) dt \\ \text{where } S_\alpha^e(t) = \left[S_\alpha^{e_1}(t) \quad S_\alpha^{e_2}(t) \quad \cdots \quad S_\alpha^{e_{n_{pq}}}(t) \right]^T \text{ and } t \in \left[t_0, t_f\right]. \end{gathered} \tag{21}$$

The performance measure given in Eq. 19 gives a measure of how close the perturbed trajectory is to the nominal trajectory when the system is subjected to small parameter perturbations. The fuzzy sensitivity measure gives a measure of how large the sensitivities of the perturbed trajectories are for small parameter perturbations. Since we want $x(t)$ to remain close to $x^*(t)$ and $S_\alpha^{e_i}(t)$ to be small, then the sensitivity analysis will involve finding the set of parameter perturbations that minimize Eqs. 19 and 21. However, in general $V_{\min}$ does not correspond to $\mathrm{FSM}_{\min}$. Therefore, when attempting to select optimum parameters for robust performance, relaxation techniques will be used to compromise between minimizing the trajectory error $e(t)$ and $S_\alpha^{e_i}(t)$ over the time interval $[t_0, t_f]$.

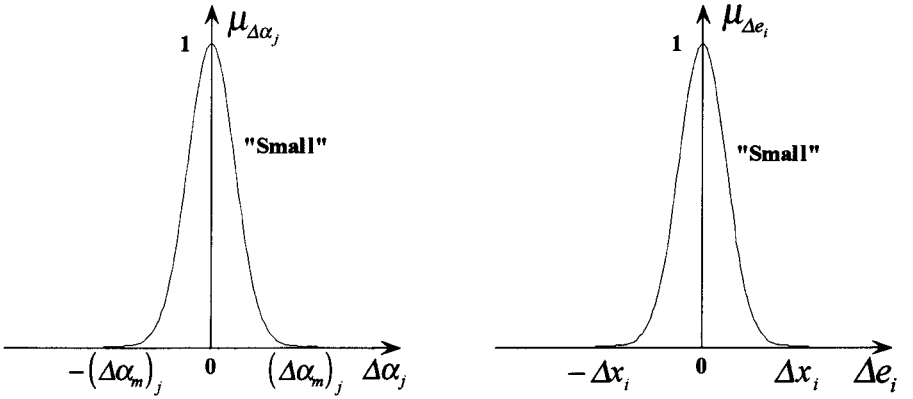


Figure 5. Fuzzy set for small perturbations and trajectory deviations.

The sensitivity analysis procedure is as follows:

- Simulate the system using the nominal parameter values.
- Determine the membership functions $\mu_{\Delta\alpha_i}$ and $\mu_{\Delta e_i}$ of the deviations for $\Delta\alpha_i$ and Δe_i.
  Figure 5 shows the fuzzy set for small parameter perturbations and trajectory deviations from the nominal trajectory.
- Determine $\text{FSM}_{\min}$ and $\text{FSM}_{\max}$.
  Figure 6 shows the fuzzy set for linguistic values of "not sensitive," not quite sensitive," "quite sensitive," "sensitive," and "very sensitive."
- Find $V_{\min}$ such that the FSM has linguistic value of "not sensitive."

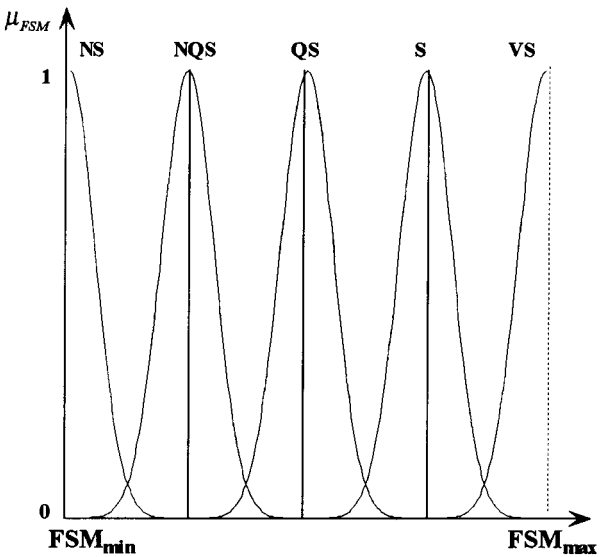


Figure 6. Fuzzy sensitivity measure.

3. EXAMPLE: DESIGN OF HOVER TO FORWARD FLIGHT FUZZY CONTROLLER

A. Parametric Model of Helicopter's Forward Dynamics

The approach discussed in Section 2c was used to design a hover to forward flight (FF) transition controller for the following model representing the longitudinal channel dynamics of a small scale helicopter constrained to have no vertical motion; only longitudinal and pitch rotation motions are allowed:

$$X = X_{\text{hov}} \cdot \mu_{\text{hov}} + X_{\text{FF}} \cdot \mu_{\text{FF}}$$

$$M = M_{\text{hov}} \cdot \mu_{\text{hov}} + M_{\text{FF}} \cdot \mu_{\text{FF}}$$

$$\ddot{x} = \frac{X}{m \cdot \cos(\theta)} - g \cdot \tan(\theta)$$

$$\ddot{\theta} = \frac{M}{I_Y}$$

$$X_{\text{hov}} = X_{\text{trim,hov}} + X_{\dot{x},\text{hov}}\left(\dot{x} - \dot{x}_{\text{trim,hov}}\right) + X_{\dot{\theta},\text{hov}}\left(\dot{\theta} - \dot{\theta}_{\text{trim,hov}}\right) + X_{\delta_e,\text{hov}}\left(\delta_e - \delta_{e,\text{trim,hov}}\right)$$

$$X_{\text{FF}} = X_{\text{trim,FF}} + X_{\dot{x},\text{FF}}\left(\dot{x} - \dot{x}_{\text{trim,FF}}\right) + X_{\dot{\theta},\text{FF}}\left(\dot{\theta} - \dot{\theta}_{\text{trim,FF}}\right) + X_{\delta_e,\text{FF}}\left(\delta_e - \delta_{e,\text{trim,FF}}\right)$$

$$M_{\text{hov}} = M_{\text{trim,hov}} + M_{\dot{x},\text{hov}}\left(\dot{x} - \dot{x}_{\text{trim,hov}}\right) + M_{\dot{\theta},\text{hov}}\left(\dot{\theta} - \dot{\theta}_{\text{trim,hov}}\right) + M_{\delta_e,\text{hov}}\left(\delta_e - \delta_{e,\text{trim,hov}}\right)$$

$$M_{\text{FF}} = M_{\text{trim,FF}} + M_{\dot{x},\text{FF}}\left(\dot{x} - \dot{x}_{\text{trim,FF}}\right) + M_{\dot{\theta},\text{FF}}\left(\dot{\theta} - \dot{\theta}_{\text{trim,FF}}\right) + M_{\delta_e,\text{FF}}\left(\delta_e - \delta_{e,\text{trim,FF}}\right)$$

$$\mu_{\text{hov}} = \begin{cases} 1 & \text{if } |\dot{x}| < 3 \\ 0 & \text{if } |\dot{x} - 17| < 3 \\ -\dfrac{\dot{x} - 14}{11} & \text{if } 3 \le \dot{x} \le 14 \end{cases}$$

$$\mu_{\text{FF}} = \begin{cases} 0 & \text{if } |\dot{x}| < 3 \\ 1 & \text{if } |\dot{x} - 17| < 3 \\ \dfrac{x - 3}{11} & \text{if } 3 \le \dot{x} \le 14 \end{cases}$$

where $\ddot{x}$, $\ddot{\theta}$, and δ_e represent the forward acceleration (ft/s^2), pitch angle acceleration (rad/s^2) and longitudinal cyclic input (rad), respectively. X represents the aerodynamic force along the "X axis" and M represents the pitching moment about the "Y axis." Figure 7 shows the axis system of the helicopter with respect to the sideview. The aerodynamic parameters and corresponding

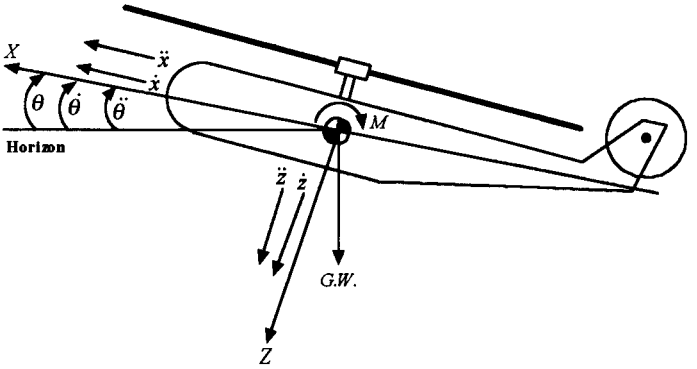


Figure 7. Side view of helicopter's axis system.

trim values for the hover and forward flight are given in Table I. These constant values were calculated by a trim analysis program using physical parameters from a Xcell 300 helicopter in hover and forward flight. The state vector of the helicopter model is $[x_1 \quad x_2 \quad x_3 \quad x_4]^T = [\dot{x} \quad \ddot{x} \quad \theta \quad \dot{\theta}]^T$. It is assumed that the output vector of the model is the same as the state vector.

In order to perform sensitivity analysis the model can be transformed into the form

$$\ddot{x} = \frac{X}{\tilde{\alpha}_3 c_{31} \cdot \cos(\theta)} - g \cdot \tan(\theta)$$

$$\ddot{\theta} = \frac{M}{\tilde{\alpha}_6 c_{61}}$$

$$X = X_{\text{hov}} \cdot \mu_{\text{hov}} + X_{\text{FF}} \cdot \mu_{\text{FF}}$$

$$M = M_{\text{hov}} \cdot \mu_{\text{hov}} + M_{\text{FF}} \cdot \mu_{\text{FF}}$$

$$X_{\text{hov}} = \tilde{\alpha}_1\left(c_{11} + c_{12}\Delta\dot{x}_{\text{hov}} + c_{13}\Delta\dot{\theta}_{\text{hov}} + c_{14}\Delta\delta_{e,\text{hov}}\right)$$

$$X_{\text{FF}} = \tilde{\alpha}_2\left(c_{21} + c_{22}\Delta\dot{x}_{\text{FF}} + c_{23}\Delta\dot{\theta}_{\text{FF}} + c_{24}\Delta\delta_{e,\text{FF}}\right)$$

$$M_{\text{hov}} = \tilde{\alpha}_4\left(c_{41} + c_{42}\Delta\dot{x}_{\text{hov}} + c_{43}\Delta\dot{\theta}_{\text{hov}} + c_{44}\Delta\delta_{e,\text{hov}}\right)$$

$$M_{\text{FF}} = \tilde{\alpha}_5\left(c_{51} + c_{52}\Delta\dot{x}_{\text{FF}} + c_{53}\Delta\dot{\theta}_{\text{FF}} + c_{54}\Delta\delta_{e,\text{FF}}\right)$$

where the parameters have the following values:

$\tilde{\alpha}_1 = 10.00$ is the nominal gain for the hover aerodynamic force

$\tilde{\alpha}_2 = 10.00$ is the nominal gain for the FF aerodynamic force

$\tilde{\alpha}_3 = 10.00$ is the nominal gain for the helicopter's mass

Table I. Aerodynamic parameters and the corresponding trim values.

	Value	Description
$X_{\dot{x}/\text{hov}}$	−0.0400	partial derivative of X w.r.t. $\dot{x}$ at hover
$X_{\dot{\theta},\text{hov}}$	1.1675	partial derivative of X w.r.t. $\dot{\theta}$ at hover
$X_{\delta_e,\text{hov}}$	21.2482	partial derivative of X w.r.t. δ_e at hover
$X_{\text{trim,hov}}$	−0.1011	trim value of aerodynamic force X at hover
$X_{\dot{x},\text{FF}}$	−0.0019	partial derivative of X w.r.t. $\dot{x}$ at forward flight
$X_{\dot{\theta},\text{FF}}$	1.2018	partial derivative of X w.r.t. $\dot{\theta}$ at forward flight
$X_{\delta_e,\text{FF}}$	26.9988	partial derivative of X w.r.t. δ_e at forward flight
$X_{\text{trim,FF}}$	−0.5411	trim value of aerodynamic force X at forward flight
$M_{\dot{x},\text{hov}}$	0.0000	partial derivative of M w.r.t. $\dot{x}$ at hover
$M_{\dot{\theta},\text{hov}}$	−1.8769	partial derivative of M w.r.t. $\dot{\theta}$ at hover
$M_{\delta_e,\text{hov}}$	−43.4060	partial derivative of M w.r.t. δ_e at hover
$M_{\text{trim,hov}}$	0.0000	trim value of aerodynamic moment M at hover
$M_{\dot{x},\text{FF}}$	0.0000	partial derivative of M w.r.t. $\dot{x}$ at forward flight
$M_{\dot{\theta},\text{FF}}$	−1.6336	partial derivative of M w.r.t. $\dot{\theta}$ at forward flight
$M_{\delta_e,\text{FF}}$	−37.4916	partial derivative of M w.r.t. δ_e at forward flight
$M_{\text{trim,FF}}$	0.0000	trim value of aerodynamic moment M at forward flight
$\dot{x}_{\text{trim,hov}}$	0.0000	trim value of forward velocity $\dot{x}$ at hover
$\dot{\theta}_{\text{trim,hov}}$	0.0000	trim value of pitch angle velocity $\dot{\theta}$ at hover
$\delta_{e,\text{trim,hov}}$	−0.0021	trim value of longitudinal input δ_e at hover
$\dot{x}_{\text{trim,FF}}$	17.0000	trim value of forward velocity $\dot{x}$ at forward flight
$\dot{\theta}_{\text{trim,FF}}$	0.0000	trim value of pitch angle velocity $\dot{\theta}$ at forward flight
$\delta_{e,\text{trim,FF}}$	−0.0421	trim value of longitudinal input δ_e at forward flight
$\theta_{\text{trim,hov}}$	−0.0037	trim value of pitch angle θ at hover
$\theta_{\text{trim,FF}}$	−0.0198	trim value of pitch angle θ at forward flight
m	0.8488	mass of the helicopter
I_Y	0.7656	moment of inertia along Y axis

$\tilde{\alpha}_4 = 10.00$ is the nominal gain for the hover aerodynamic moment

$\tilde{\alpha}_5 = 10.00$ is the nominal gain for the FF aerodynamic moment

$\tilde{\alpha}_6 = 10.00$ is the nominal gain for the moment of inertia along the Y axis

and the constant values are

$$c_{11} = -0.0101 \quad c_{12} = -0.0040 \quad c_{13} = 0.1168 \quad c_{14} = 2.1248$$

$$c_{21} = -0.0541 \quad c_{22} = -0.0002 \quad c_{23} = 0.1202 \quad c_{24} = 2.6999$$

$$c_{31} = 0.0849$$

$$c_{41} = 0.0000 \quad c_{42} = 0.0000 \quad c_{43} = -0.1877 \quad c_{44} = -4.3406$$

$$c_{51} = 0.0000 \quad c_{52} = 0.0000 \quad c_{53} = -0.1634 \quad c_{54} = -3.7492$$

$$c_{61} = 0.0766$$

B. Hover to Forward Flight Mode Controller

The following control law was used for the hover to FF controller

$$\delta_e = \delta_{e,\mathrm{hov}}(\dot{x}, \theta, \dot{\theta}) \cdot K_{\mathrm{hov}}(\dot{x}, \ddot{x}, \theta, \dot{\theta}) + \delta_{e,\mathrm{FF}}(\dot{x}, \ddot{x}, \theta, \dot{\theta}) \cdot K_{\mathrm{FF}}(\dot{x}, \ddot{x}, \theta, \dot{\theta})$$

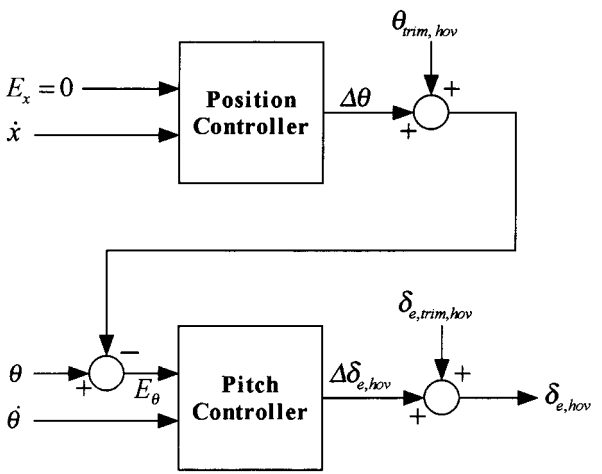
where $\delta_{e,\mathrm{hov}}(\cdot)$ and $\delta_{e,\mathrm{FF}}(\cdot)$ are fuzzy regulators for the hover and FF modes. Note that $\delta_{e,\mathrm{hov}}(\cdot)$, $\delta_{e,\mathrm{FF}}(\cdot)$, and δ_e are scalar. Figures 8 and 9 show the structure of the hover and forward flight controllers.

The hover and FF controllers regulate about the operating points

$$[\dot{x} \quad \ddot{x} \quad \theta \quad \dot{\theta}]^T = [0.0000 \quad 0.0000 \quad -0.0037 \quad 0.0000]^T$$

$$[\dot{x} \quad \ddot{x} \quad \theta \quad \dot{\theta}]^T = [17.0000 \quad 0.0000 \quad -0.0198 \quad 0.0000]^T$$

respectively. The scalar gains $K_{\mathrm{hov}}(\dot{x}, \ddot{x}, \theta, \dot{\theta})$ and $K_{\mathrm{FF}}(\dot{x}, \ddot{x}, \theta, \dot{\theta})$ are determined via the PPAA such that closed-loop system transitions from [0.0000 0.0000



E_x

$\dot{x}$ \ $\Delta\theta$	N	Z	P
N	NM	NM	PS
Z	NM	Z	PM
P	NS	PM	PM

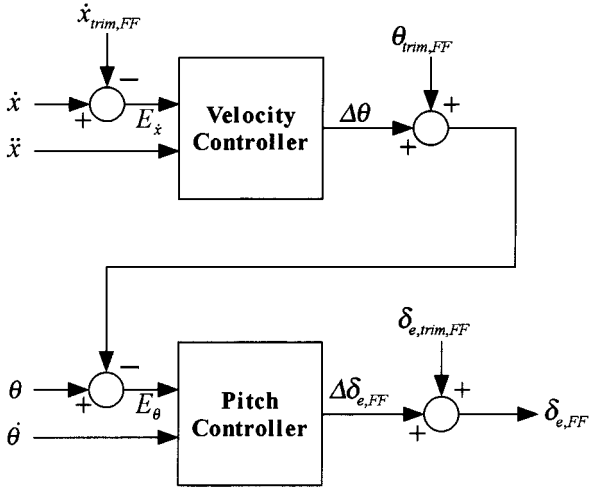
Position Controller Rulebase

E_θ

$\dot{\theta}$ \ $\Delta\delta_{e,hov}$	N	Z	P
N	NM	NM	PS
Z	NM	Z	PM
P	NS	PM	PM

Pitch Controller Rulebase

Figure 8. Structure of hover mode controller.



$E_{\dot{x}}$

$\ddot{x}$ \ $\Delta\theta$	N	Z	P
N	Z	Z	PM
Z	NS	Z	PS
P	NM	Z	Z

Velocity Controller Rulebase

E_{θ}

$\dot{\theta}$ \ $\Delta\delta_{e,FF}$	N	Z	P
N	NM	NM	PS
Z	NM	Z	PM
P	NS	PM	PM

Pitch Controller Rulebase

Figure 9. Structure of forward flight mode controller.

$-0.0037 \quad 0.0000]^T$ to $[17.0000 \quad 0.0000 \quad -0.0198 \quad 0.0000]^T$ in minimum time. Table II shows the parameters used by PPAA to design a minimum-time hover to FF transition controller, while Figure 10 shows the structure of this controller. A minimum-squared-error or minimum-gain-control transitional controller could also be designed, along similar lines.

Table II. PPAA parameters used to design hover to FF transition controller.

	Max Value	Min Value	Desired Value	Number of Divisions
$\dot{x}$	17.6296	-0.3148	17.0000	56
$\ddot{x}$	2.5000	-0.5000	0.0000	5
θ	0.0124	-0.1164	-0.0198	7
$\dot{\theta}$	0.2400	-0.2400	0.0000	5
k_{hov}	1.0625	-0.0625	0.0000	17
k_{FF}	1.0625	-0.0625	1.0000	17

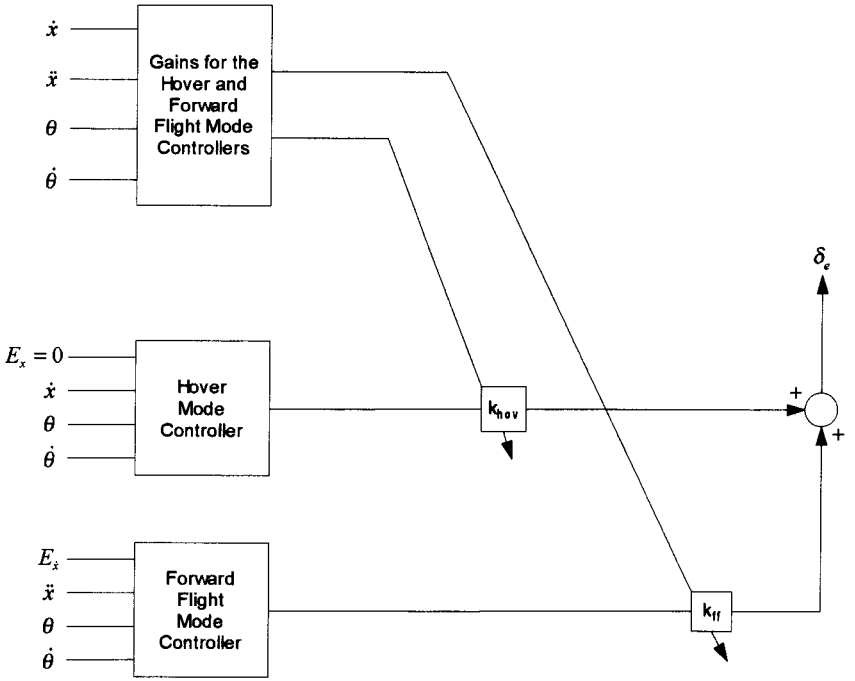


Figure 10. Structure of hover to FF transition controller.

4. RESULTS

A. Simulation with Nominal Parameters

Figures 11–17 show the simulation results of the hover to FF transition controller applied to the helicopter described above. Figures 11–14 show that the helicopter was able to stably transition from the hover mode to the FF mode, as desired. The helicopter went from $[0.0000 \quad 0.0000 \quad -0.0037 \quad 0.0000]^T$ to $[16.9993 \quad -0.0008 \quad -0.0198 \quad -0.0000]^T$ within 14 seconds and had a rise time of about 9.2 seconds. The velocity, acceleration and pitch angle profiles were fairly smooth. However, the pitch rate and control input profiles were not smooth. This is due to the fact that the controller was designed with one rule firing while the fuzzy implementation fires multiple rules.

B. Sensitivity Analysis

The sensitivity analysis of the $\text{mode}_{\text{hover}}$-to-$\text{mode}_{\text{FF}}$ fuzzy controller is studied when α_3 and α_6 are perturbed. This analysis is performed so that the perturbed trajectories are within the stability band represented by the following

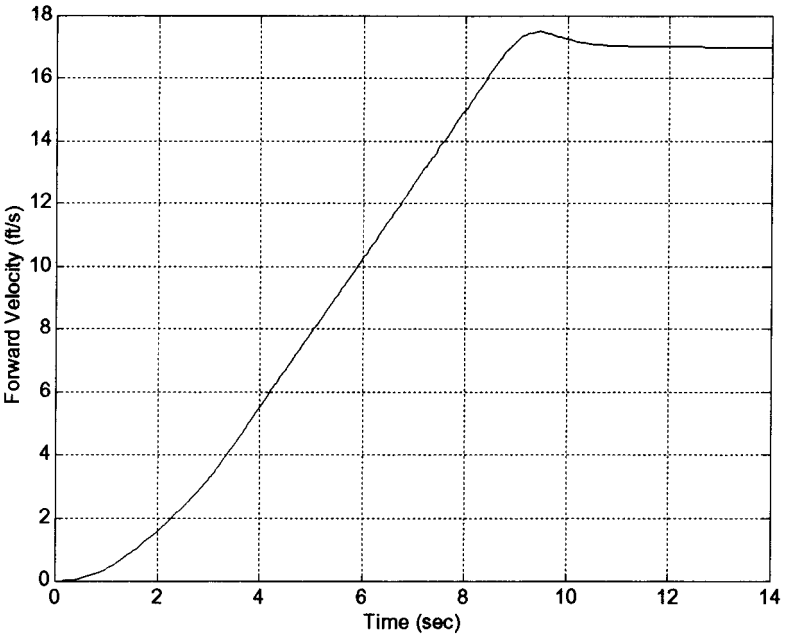


Figure 11. Plot of $\dot{x}$ versus time.

expression:

$$|e_i(t)| \leq \Delta x_i \qquad \text{for} \quad i = 1, \ldots, 4$$

where $\Delta x = [\Delta x_1 \quad \Delta x_2 \quad \Delta x_3 \quad \Delta x_4]^T = [0.50 \quad 0.80 \quad 0.01 \quad 0.10]^T$.

The parameters for $(\Delta \alpha)_{m1}$ and $(\Delta \alpha_m)_2$ are given below:

$$[(\Delta \alpha_m)_1 \quad (\Delta \alpha_m)_2]^T = [2 \quad 2]^T$$

The maximum and minimum fuzzy sensitivity measures were determined to be

$$\text{FSM}_{\max} = 1124.5000 \text{ at } \Delta \alpha_3 = -0.1000 \qquad \Delta \alpha_6 = -0.0010$$

$$\text{FSM}_{\min} = 0.1983 \text{ at } \Delta \alpha_3 = -0.1000 \qquad \Delta \alpha_6 = -1.0000$$

The linguistic values of "not sensitive," "not quite sensitive," "quite sensitive," "sensitive," and "very sensitive" are assigned to FSM in the following manner:

If $0.1983 \leq \text{FSM} \leq 140.5377$ then FSM is "not sensitive"

If $140.5377 < \text{FSM} \leq 421.6131$ then FSM is "not quite sensitive"

If $421.6131 < \text{FSM} \leq 702.6885$ then FSM is "quite sensitive"

If $702.6885 < \text{FSM} \leq 983.7639$ then FSM is "sensitive"

If $983.7639 < \text{FSM} \leq 1124.5000$ then FSM is "very sensitive"

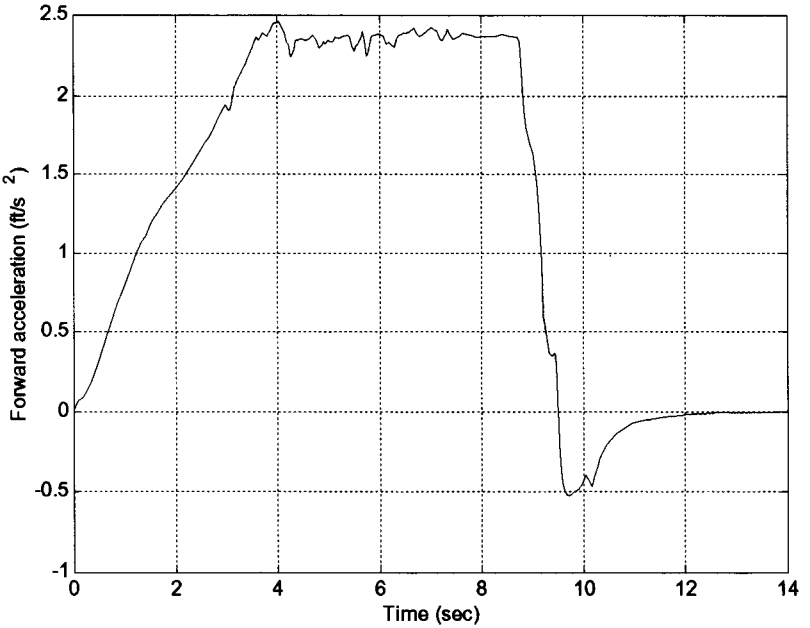


Figure 12. Plot of $\ddot{x}$ versus time.

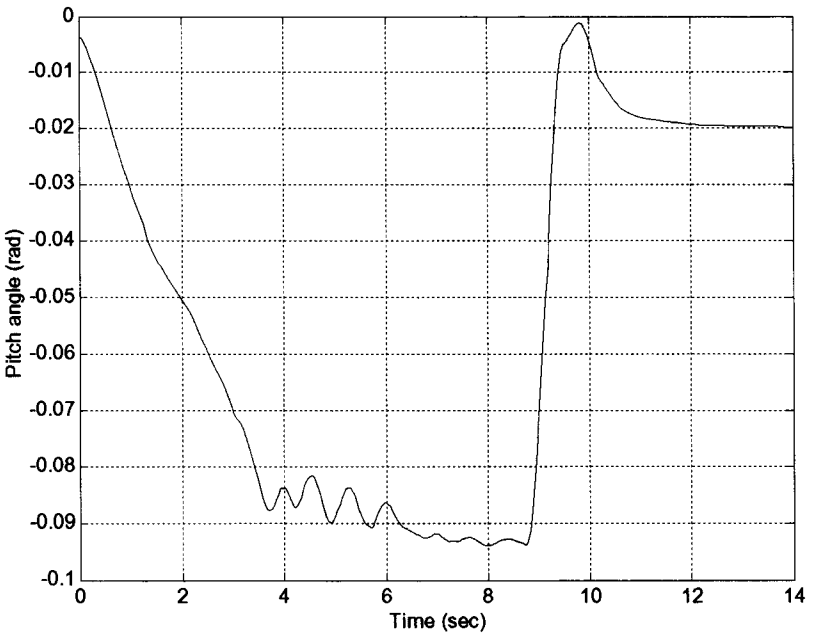


Figure 13. Plot of θ versus time.

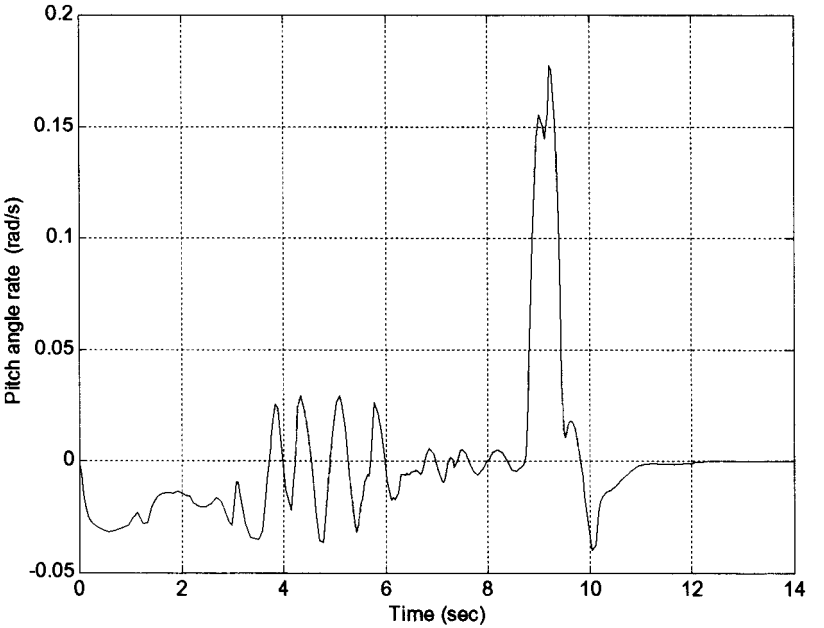


Figure 14. Plot of $\dot{\theta}$ versus time.

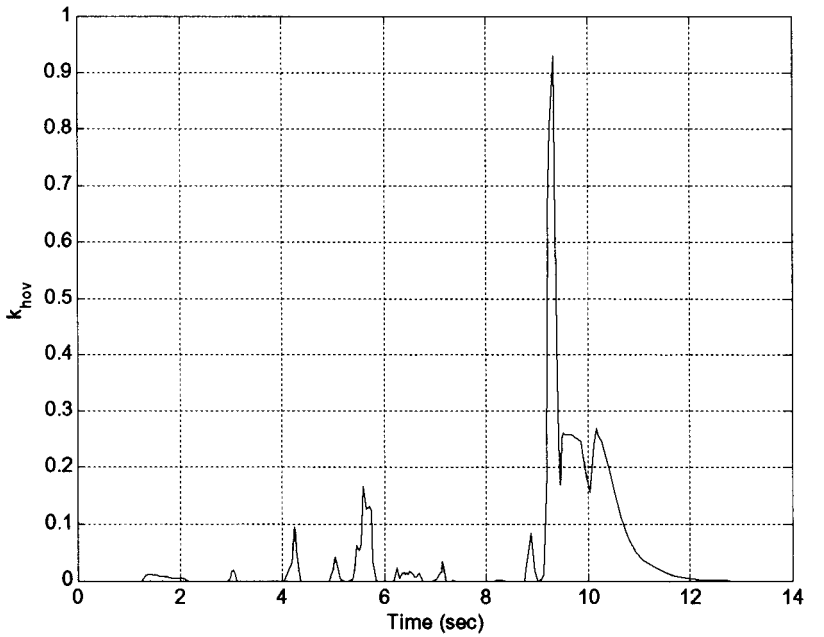


Figure 15. Plot of k_{hov} versus time.

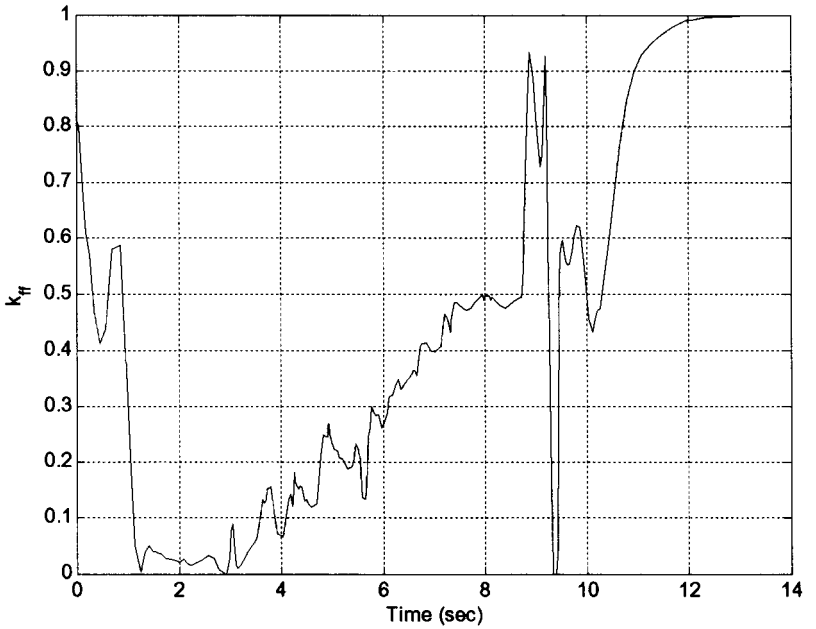


Figure 16. Plot of k_{FF} versus time.

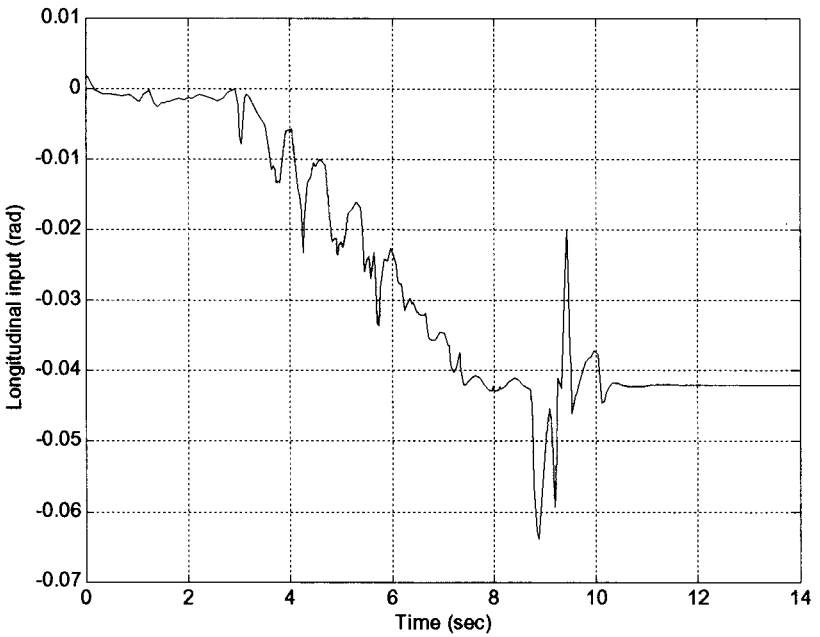


Figure 17. Plot of δ_e versus time.

The optimum sensitivity was found to occur at $\Delta\alpha_3 = -0.0800$, $\Delta\alpha_6 = -0.5000$ where $V_{\min} = 0.0181$ and FSM = 4.7031. The FSM value of 4.7031 corresponds to a linguistic value of "not sensitive." Figures 15–18 show the sensitivity of $\dot{x}$, $\ddot{x}$, θ, and $\dot{\theta}$ with respect to α, and the plots of the performance and fuzzy sensitivity measure for the optimum sensitivity.

5. CONCLUSIONS

Large-scale dynamical systems that have several operating modes require stable transitions between them. A design methodology for mode-to-mode controllers using the phase portrait assignment algorithm has been presented. The mode-to-mode controller design employs a hierarchical control scheme to blend the outputs of the controllers designed for the start and goal modes of operation. The phase portrait assignment algorithm determines the gains needed to blend the mode controllers such that the system transitions from the start mode to the goal mode of operation. This methodology was illustrated in the design of a hover mode to forward flight mode controller of a small helicopter. Simulation results show that the controller was able to transition the helicopter stably from hover flight to forward flight. Finally, sensitivity analysis of the hover mode to forward flight mode controller is performed for small parameter perturbations. The sensitivity analysis involved finding the optimum sensitivity such that the deviations and the sensitivity of the deviations from the nominal

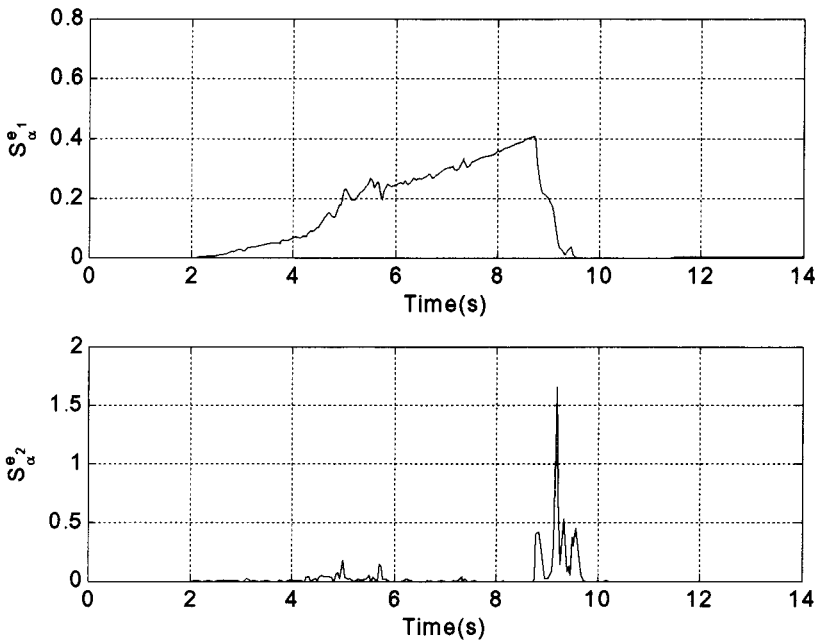


Figure 18. Sensitivity of $\dot{x}$ and $\ddot{x}$ for optimum sensitivity.

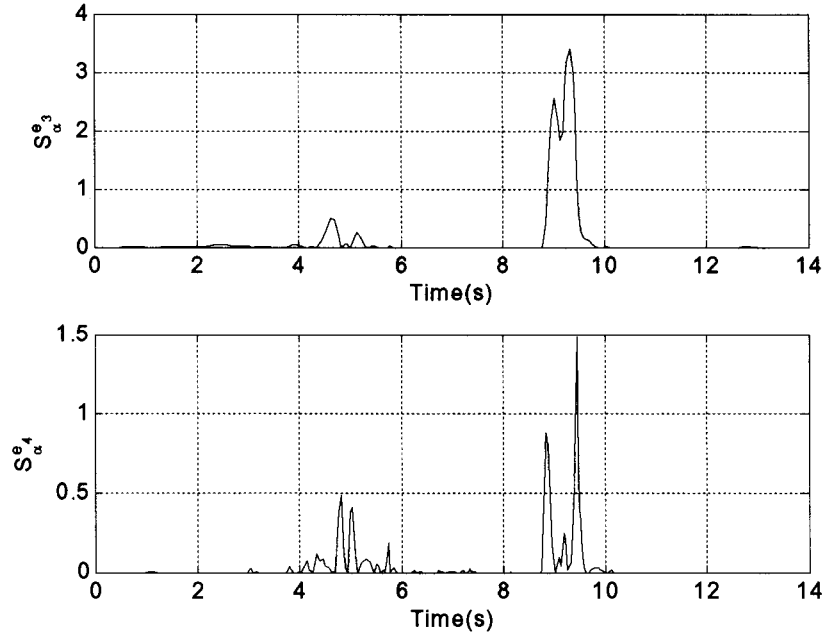


Figure 19. Sensitivity of θ and $\dot{\theta}$ for optimum sensitivity.

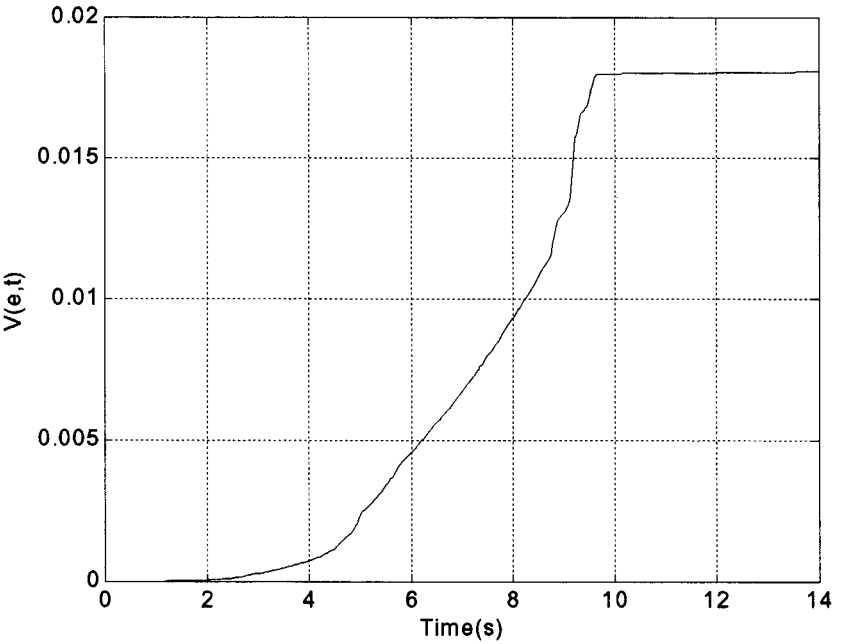


Figure 20. Performance measure for optimum sensitivity.

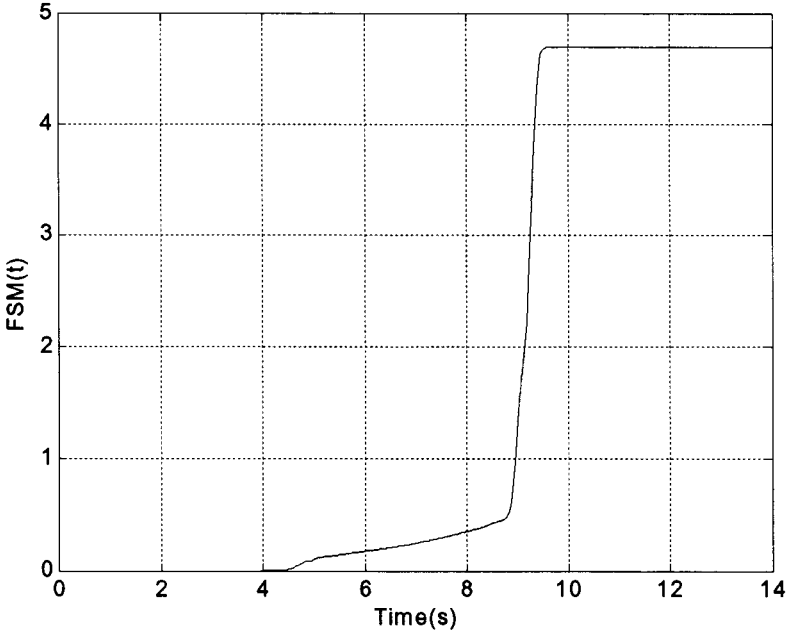


Figure 21. Fuzzy sensitivity measure for optimum sensitivity.

mode-to-mode trajectory is minimized. The methodology may be extended to include all critical mode transitions for a vehicle like a helicopter. Its application to small autonomous or unmanned vehicles will prevent unstable operating conditions when the latter undergo extreme or difficult maneuvering. A robust stability analysis of the dynamic system has been carried out and is presented in a companion paper.[23]

This work has been partially supported by DARPA under the Software Enabled Controls program, Contract No. E21-K59; their sponsorship and continued support is gratefully acknowledged.

References

1. Shamma JS, Athans M. Gain scheduling: Potential hazards and possible remedies. IEEE Control Systems Magazine 1992;101–107.
2. Bradley E, Zhao F. Phase-space control system design. IEEE Control Systems Magazine 1993;49:39–46.
3. Sugeno M, Griffin MF, Walker G, Bastian A. Issues for blending fuzzy controllers in hierarchical systems. Proceedings First Asian Fuzzy Systems Symposium, Singapore, Nov. 1993.
4. Shamma JS, Athans, M. Analysis of gain scheduled control for nonlinear plants. IEEE Trans Automatic Control 1990;35:898–907.
5. Ling C, Edgar TF. A new fuzzy gain scheduling algorithm for process control. Am Control Conf 1992;49:2284–2290.

6. Gonsalves PG, Zacharias GL. Fuzzy logic gain scheduling for flight control. Int J Control 1994;49:952–957.
7. Vachtsevanos G, Farinwata SS, Pirovolou DK. Fuzzy logic control of an automotive engine. IEEE Control Systems Magazine 1993;13:62–68.
8. Vachtsevanos G, Kim W, Al-Hasan S, Rufus F, Simon M, Schrage D, Prasad JVR. Autonomous vehicles: From flight control to mission planning using fuzzy logic techniques. Proc of the International Workshop on Breakthrough Opportunities for Fuzzy Logic, pp. 25–29, Yokohama, Japan, 1996.
9. Vachtsevanos G, Kim W, Al-Hasan S, Rufus F, Simon M, Schrage D, Prasad JVR. Mission planning and flight control: Meeting the challenge with intelligent techniques. J Adv Comput Intel 1997;1:62–70.
10. Kang H, Vachtsevanos G. Nonlinear fuzzy control based on the vector field of the phase portrait assignment algorithm. Proc Am Control Conf 1990;2:1479–1484.
11. Papa M, Tai HM, Shenoi S. Cell mapping for controller design and evaluation. IEEE Control Systems Magazine 1997;17:1331–1338.
12. Farinwata SS. On the stability of fuzzy control rulebase for a nonlinear process. IEEE Int Conf Fuzzy Systems 1994;2:924–929.
13. Farinwata SS, Vachtsevanos G. Stability analysis of the fuzzy logic controller designed by the phase portrait assignment algorithm. IEEE Int Conf Fuzzy Systems 1993;2:1377–1382.
14. Hsu CS. A discrete method of optimal control based upon cell state space concept. J Opt Theory Appl 1985;46:547–569.
15. Hsu CS. Cell-to-cell mapping. New York: Springer-Verlag, 1987.
16. Chen YY, Tsao TC. A description of the dynamical behavior of fuzzy systems. IEEE Trans Systems, Man and Cybernetics 1989;191:745–755.
17. Fei J, Isik C. The analysis of fuzzy knowledge-based systems using cell-to-cell mapping. IEEE Int Symp Intel Control 1990;1:633–637.
18. Fortuna L, Graziani S, Baglio S, Nunnari G. Improvements in fuzzy controller design. IEEE Int Conf Syst Eng 1991;1:237–240.
19. Bursal FH, Hsu CS. Application of a cell-mapping method to optimal control problems. Int J Control 1989;49:1505–1522.
20. Kang H. An automated rule design of fuzzy logic controllers for uncertain dynamic systems. IEEE Int Conf Fuzzy Systems 1993;1:261–266.
21. Kang H. Stability and control of fuzzy dynamic systems via cell-state transitions in fuzzy hypercubes. IEEE Trans Fuzzy Systems 1993;1:267–279.
22. Farinwata SS, Vachtsevanos G. Robust stability of fuzzy logic control systems. Proc American Control Conference 1995;5:2267–2271.
23. Rufus F, Vachtsevanos G. Robust stability of mode-to-mode fuzzy controllers, J Guidance, Control and Dynamics 1999;22:823–832.