HELICOPTER FLIGHT CONTROL WITH FUZZY LOGIC AND GENETIC ALGORITHMS

CHAD PHILLIPS
Logic Technologies, 421 E. Orange St., Tuscaloosa, AL 35401, USA
phillips@rotor.larc.nasa.gov

CHARLES L. KARR
Engineering Science and Mechanics Department, The University of Alabama, Tuscaloosa, AL 35487-0278, USA
ckarr@eng.ua.edu

GREGORY W. WALKER
Aerobotics Corp., 2105 Old York-Hampton Hwy., Yorktown, VA 23692, USA
gwalker@myhost.ccsinc.com

Researchers at the U.S. Bureau of Mines, in conjunction with researchers at the University of Alabama and the U.S. Army, have developed a fuzzy system for controlling the flight of UH-1 helicopters through various maneuvers. Since flying a helicopter is an extremely difficult task, the fuzzy logic controller was necessarily quite complex. In fact, the control tasks were distributed over four individual control units, each of which had its own rules and associated membership functions. Because the fuzzy logic controller was large and because the rules implemented in the individual control units were not necessarily those a human pilot would use, an efficient technique for writing the rules was required. A genetic algorithm was used to discover rules that provided for effective control of the helicopter. Genetic algorithms are search algorithms based on the mechanics of natural genetics and have demonstrated the ability to locate rules for fuzzy logic controllers. This paper describes the architecture of the helicopter fuzzy logic controller, provides the details of the genetic algorithm application, and presents the results of an actual flight test using the computer software.

Keywords: Fuzzy Logic, Genetic Algorithms, Helicopter, Flight Control.

1 Introduction

Developing an automatic pilot for a helicopter is a difficult problem because of the inherent instability, the high number of degrees of freedom, and the high degree of coupling between the state variables. [1] None the less, there are numerous reasons for undertaking this effort including rewards in terms of safety, commercialization, and flexibility. For instance, there are flight missions, both military and commercial, that place the pilot and the aircraft in danger. In these dangerous situations, it would certainly be desirable to replace the human pilot with an automatic control system. Additionally, there are some flight missions that simply preclude the use of human pilots such as the testing of weapons systems.

In fact, there are algorithms in use for attaining automatic helicopter flight control. However, these algorithms are highly aircraft specific; the algorithms have to be completely redesigned for each type of helicopter on which they are to be used. A flight control algorithm that could be easily adapted for use on a number of aircraft would save the helicopter industry, weapons manufacturers, and the military large sums of money, and would mark a major step forward in the area of helicopter flight control.

Fuzzy control systems, or fuzzy logic controllers, have become increasingly popular in the last decade. These systems have been used in a number of problem domains including chemical engineering, manufacturing, mineral engineering, and aerospace engineering. Recently, algorithms have been developed for automatically adapting fuzzy logic controllers to account for changes in the problem environment which they are manipulating. Procyck and Mamdani [2] were the first to develop an adaptive fuzzy logic controller, and their work involved using a derivative based search. Later, Karr and his coworkers [3,4,5,6] successfully used genetic algorithms to adapt fuzzy logic controllers in real-time. Thus, fuzzy logic controllers seem to be potential solutions to the problem of developing a helicopter control system that is easily adapted to various aircraft. And, since efficient helicopter flight control is obviously achieved by human experts, information needed to construct an effective fuzzy control system can be extracted via interviews with human pilots. Thus, the information and knowledge used by human pilots can be translated into a form such that it can be used in a fuzzy control system. The resulting fuzzy logic controller might eventually replace human pilots in dangerous missions, and at the same time be robust enough to be applicable to a variety of aircraft.

One of the difficulties associated with the development of a fuzzy system for helicopter flight control is tied to the complexity of the aircraft dynamics: the state variables are so highly coupled that it is exceedingly difficult to decouple the system adequately to implement conventional control strategies. Fortunately, the development of a fuzzy logic controller is not dependent on decoupling the system. A fuzzy logic controller can be developed that subdivides the tasks necessary to fly the aircraft. Individual fuzzy logic controllers can be written to achieve four goals associated with flying the aircraft: (1) the longitudinal velocity goal, (2) the vertical velocity goal, (3) the lateral velocity goal, and (4) the heading goal. This division of tasks produces a unique fuzzy logic controller architecture. Perhaps, more importantly, it reduces the size of the rule set needed in the control system to a point where it is manageable.

However, this subdivision of tasks does not reduce the size of the rule set to a point that the rules are immediately apparent. Thus, some mechanism for determining appropriate rules is necessary.

A genetic algorithm has been used successfully in the past to locate efficient fuzzy logic controller rules. [7] Genetic algorithms are search algorithms based on the mechanics of natural genetics. [8] They search large spaces efficiently without the need of derivative information. Their robust nature allows them to efficiently search for effective rules in the helicopter flight fuzzy logic controller. Additionally, the flexibility associated with determining the fitness function that drives a genetic algorithm makes it possible to design a controller that is capable of optimizing performance relative to time and energy simultaneously. This capability is not apparent in most optimal control algorithms. [9]

This paper describes the development of a fuzzy logic controller for a UH-1H helicopter. Although the control system has not yet been adapted for other aircraft, the architecture that has been developed is flexible enough so as to be applicable to a wide variety of helicopters. The sections that follow provide details of the fuzzy logic controller architecture and describe the approach used to determine an effective rule set using a genetic algorithm. Results are provided demonstrating the performance of the fuzzy logic controller on a computer simulation of a UH-1H helicopter, and on a test flight of an actual aircraft.

2 The Problem

An automated aircraft controller is useful in hazardous flight situations, both military and commercial, in which it is desirable to remove the pilot from the aircraft. For years, fixed-wing 'drones' have fulfilled this role, especially in military applications. There are, however, instances wherein an aircraft is required to maneuver in tight areas or maintain a specific position for an extended period of time. In these cases, an unmanned helicopter is better suited for the task. [10] However, control of a helicopter is much more difficult to achieve than control of a fixed-wing aircraft.

The difficulty in developing a controller for a helicopter stems from the inherent instability and high degree of coupling in the aircraft. If a pilot releases the controls of a typical fixed-wing aircraft, the aircraft will eventually reach some steady state flight condition. The same is not true for a helicopter. Without constant corrective control inputs, the aircraft will diverge from steady state. Also, the flight dynamics of a helicopter are highly coupled and vary from one aircraft to another as well as from one flight region to another. The coupling is due, in part, to the large gyroscopic moment created by the main rotor. Any forward pitch motion, i.e. "nose-down", in a helicopter results in a corresponding roll to the right. Control induced aerodynamic effects introduce additional coupling.

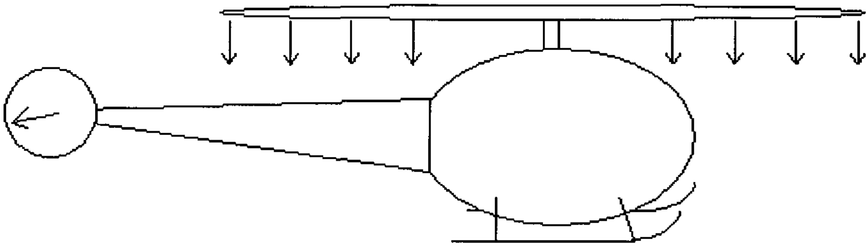

Figure 1: Force the helicopter produces on surrounding air.

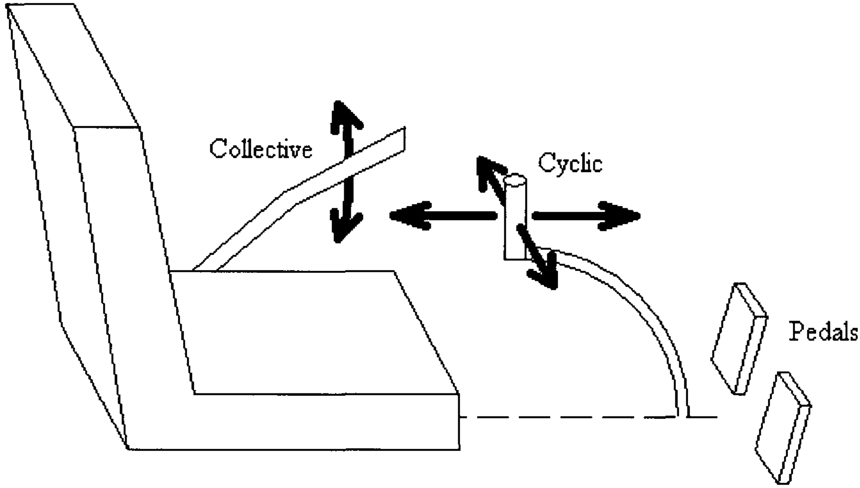

Figure 2: Three primary controls in the cockpit of a helicopter.

In a typical helicopter configuration as shown in the schematic of Figure 1, the body is suspended from a main rotor which provides lift. A tail rotor is employed to counteract the torque produced by the reaction of the air against the main rotor. Inside the cockpit, the pilot uses three primary controls to fly the helicopter (Figure 2): (1) the collective is used to adjust the pitch of the main rotor blades in order to increase or decrease lift, (2) the pedals are used to adjust the pitch of the tail rotor blades, thus producing more or less torque on the aircraft body as needed, and (3) the cyclic, for all practical purposes, determines the orientation of the main rotor with respect to the aircraft body. The main rotor can be thought of as an imaginary disk that produces lift. When the cyclic is pushed forward, the disk tilts forward with respect to the aircraft body causing a forward acceleration. When the cyclic is pushed to the right, the disk tilts to the right causing a rightward acceleration. Unfortunately, each control excites motion on multiple axes. In other words, the controls are highly coupled and, as a result, flying a helicopter is an extremely difficult task.

To demonstrate the coupling from the controls, consider a maneuver in which the pilot wants to ascend from a hover to some predetermined altitude. To increase the lift from the main rotor, and thus begin the ascent, the collective is pulled upward. This causes the pitch on the main rotor blades to increase, producing greater force on the air. As a result, the counter torque provided by the tail rotor is no longer sufficient to maintain a constant heading. Therefore, the pedals must be adjusted to change the pitch on the tail rotor blades, thereby providing the correct counter torque for the new collective position. Unfortunately, the torque provided by the tail rotor is not generally limited to the vertical axis of the helicopter. In most instances, the tail rotor affects torque about the longitudinal axis resulting in roll. The roll causes a pitch due to gyroscopic precession from the main rotor. Each control produces multiple effects in the motion of the aircraft. As a result, all of the controls must be meticulously coordinated in order to perform even the simplest maneuver.

3 The Helicopter

The helicopter used in this research is the UH-1 "Huey". While the actual flight tests were performed on a UH-1H, the controller was designed using a simulator developed at NASA Ames Research Center. The simulator is a nonlinear, total force and moment model of a single main rotor helicopter. The model considers ten degrees of freedom: six for rigid-body dynamics, three for rotor-flapping, and one for the rotation of the rotor. [11] The simulator was modified at the Georgia Institute of Technology to model the UH-1H specifically, and to allow simple interface to the search algorithm used in this research. [12,13]

In general, the state variables of an aircraft are defined about a right-handed aircraft body coordinate system with the x axis pointing out the nose, the y axis pointing out the right side, and the z axis pointing out the bottom. From this convention, the following state variables are defined:

u = $\frac{dx}{dt}$ (m/sec)	Forward velocity
v = $\frac{dy}{dt}$ (m/sec)	Rightward velocity
w = $\frac{dz}{dt}$ (m/sec)	Downward velocity
$\frac{du}{dt}$ (m/sec^2)	Forward acceleration
$\frac{dv}{dt}$(m/sec^2)	Rightward acceleration
$\frac{dw}{dt}$ (m/sec^2)	Downward acceleration
p (deg/sec)	Angular velocity around the x axis
q (deg/sec)	Angular velocity around the y axis
r (deg/sec)	Angular velocity around the z axis
R_u (m/sec)	Reference (or desired) forward velocity
R_v (m/sec)	Reference rightward velocity
R_w (m/sec)	Reference downward velocity
R_Ψ (deg)	Reference heading

Aside from the variables defined in body coordinates, the yaw, pitch and roll angles are defined as Euler angles in a ground reference frame. Obviously altitude is also defined with respect to the ground reference frame. Notice that all distances are measured in meters,0 all angles in degrees, and time in seconds.

Ψ (deg)	Yaw (heading)
Θ (deg)	Pitch (angle from horizon to the nose)
Φ (deg)	Roll (angle from horizon to the right wing)
h (m)	Altitude

4 Fuzzy Logic Controller Architecture

The goal of the current research is to develop a helicopter controller that is aggressively effective, computationally efficient, and easily adaptable to new rotorcraft. The controller should be capable of quickly and smoothly moving the helicopter from one steady state condition to another. It should be able to perform a simple set of maneuvers such as acceleration, deceleration, and steady turn. As with any computational system, the least rigorous algorithm that accomplishes a specific task is desirable. Finally, the controller should be as aircraft independent as possible.

While fuzzy control addresses many of these concerns directly, the particular architecture of the controller can mean the difference between success or failure in achieving the goals set forth. Architecture determines in what way state information flows through the controller. The flow of state information can be manipulated to take advantage of the knowledge of the system. Information that is unrelated, or not greatly coupled, can be sent to separate processing blocks. Using this technique, a controller is developed that has specialized sections through which information is distributed. Each section deals with a unique portion of the state space and addresses a specific aspect of control.

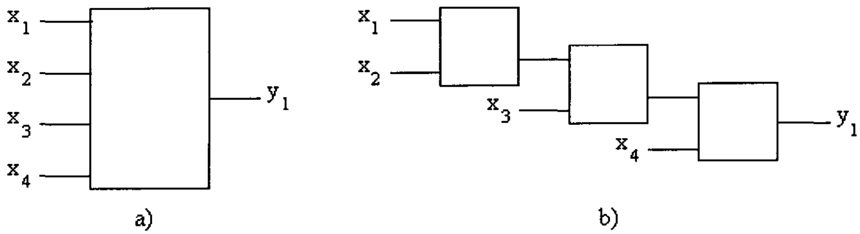

Figure 3: A classical fuzzy controller (a) can often be replaced by a distributed architecture (b) that performs adequately.

A classical fuzzy controller would typically consists of one fuzzy block with an input for each state variable (x_i) and an output for each control action (y_j) as shown in Figure 3a. Each input is explicitly related to the other inputs through rules in the rule base. For large problems, however, the classical approach to the development of fuzzy systems becomes impractical because the number of rules increases exponentially with the number of inputs. If each of the 17 variables used in the helicopter controller were described with 3 membership functions, the size of the rule base would be 3^{17}, or approximately 129 x 10^6 rules.

A distributed architecture, shown in Figure 3b, is an alternative to the classical, single block fuzzy controller. Using knowledge of the dynamics of the system to divide control tasks allows adequate control with much fewer rules than required by the classical fuzzy controller. If each input is described with 3 membership functions, the classical controller in Figure 3a requires $3^4 = 81$ rules. The distributed controller in Figure 3b, on the other hand, requires only $3^2 + 3^2 + 3^2 = 27$ rules. If the inputs are not tightly coupled, the distributed controller can achieve adequate performance with fewer rules.

Identification of a suitable architecture is often quite difficult. While it is desirable to generate a controller with a small number of rules, care must be taken to preserve important relationships between state variables. A careful balance between a small number of rules and accurate representation of the coupling in the system is needed to achieve an efficient and effective controller.

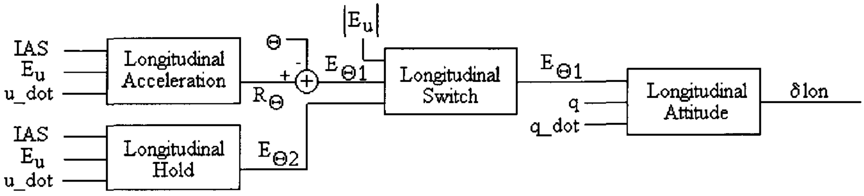

Figure 4: Section of the fuzzy controller used for longitudinal cyclic control.

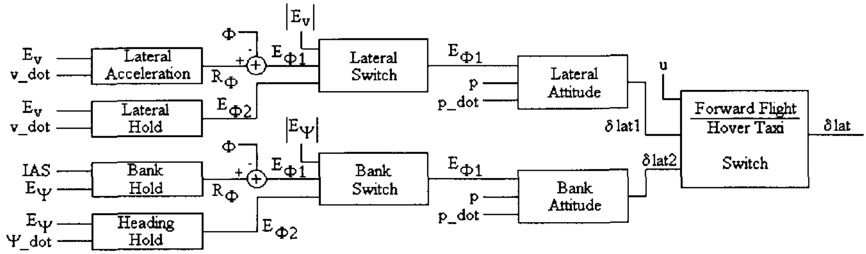

Figure 5: Section of the fuzzy controller used for lateral cyclic control.

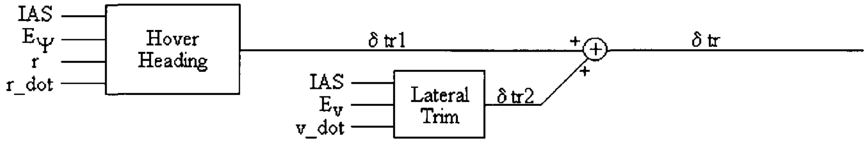

Figure 6: Section of the fuzzy controller used for pedal control.

Figure 7: Section of the fuzzy controller used for collective control.

Designing the helicopter controller architecture was largely a process of trial and error tempered with a minimal amount of human expertise. After several design iterations, the configuration shown in Figure 4 through Figure 7 was adopted. The controller is composed of four sections, one for each control freedom in the aircraft. The four sections represent control of the longitudinal cyclic (δ_{lon}) (Figure 4), lateral cyclic (δ_{lat}) (Figure 5), pedals (δ_{tr}) (Figure 6), and collective (δ_{col}) (Figure 7). Note that in these figures the extension dot represents a derivative with respect to time.

The longitudinal cyclic control motion is determined by the fuzzy block labeled Longitudinal Attitude. Given the error in pitch, q, and q_dot, the Longitudinal Attitude block infers a change in longitudinal cyclic. The Longitudinal Switch determines which of two control strategies is employed based on the error in forward velocity. If the forward velocity of the helicopter is far from the goal velocity, the error in pitch is determined by the Longitudinal Acceleration block. Otherwise, the error in pitch is determined by the Longitudinal Hold block. Notice that the Longitudinal Acceleration block infers a desired pitch angle while the Longitudinal Hold block infers an error in pitch angle. While at first this arrangement might seem overly complex it is designed to model the decision process of an actual pilot. When a pilot is near the desired forward velocity, he achieves the velocity goal by making small adjustments to the pitch of the aircraft. When a pilot is far from the desired forward velocity, he will typically maintain a desired pitch attitude to accelerate or decelerate the helicopter. In both cases, the result is a function of the current airspeed, the error in forward velocity and the forward acceleration.

Lateral cyclic control is split into two control branches (Figure 5) that each resemble the longitudinal section of the controller. The two branches represent the different functions of lateral cyclic at hover and at forward flight. At hover, the lateral cyclic is used to determine lateral velocity of the aircraft. In forward flight, however, the desired lateral velocity is almost always zero. In this case, the lateral cyclic is used to roll the aircraft into turns to achieve new headings. Notice in the figure that the top branch of the lateral controller uses lateral velocities to achieve control at hover. The bottom branch of the lateral controller uses heading information to achieve heading control in forward flight.

Control of the tail rotor (Figure 6) is also determined as a function of the current forward airspeed. While not obvious from the figure, the rules in the Hover Heading block become ineffective as airspeed increases. The opposite is true for the Lateral Trim block. At hover, the Hover Heading block controls the tail rotor. In forward flight, the Lateral Trim block controls the tail rotor. Between hover and forward flight, the actions of the two control blocks are linearly combined. The Purpose of the Lateral Trim block is to make very small adjustments to the tail rotor in forward flight to eliminate side-slip.

Finally, collective control (Figure 7) is determined as a function of the current airspeed, error in climb rate, and climb acceleration of the helicopter. The collective control section is the simplest and most effective section of the controller.

Fuzzy switches are used throughout the controller to provide specialized control for different flight regions. An example is the Forward Flight / Hover Taxi switch in the lateral cyclic controller (Figure 5). If the forward velocity of the helicopter is less than 6 m/s, then the result from the top section of the lateral controller is passed through the switch. If the forward velocity is greater than 12 m/s, the result of the bottom section of the lateral controller is passed through the switch. When forward velocity is between 6 m/s and 12 m/s, a linear combination of the results of the top and bottom sections of the lateral controller are provided as output from the switch. Thus, the switch is indeed fuzzy as opposed to discrete. The rules in the switches are predetermined, not manipulated by the search algorithm during rule discovery.

There are twelve fuzzy blocks in the helicopter controller, excluding the fuzzy switches. Each control block has several inputs which are described by a collection of triangular membership functions, or *members.* Also, the inputs are clipped to insure that they fall between a minimum and maximum value. These details are listed for each fuzzy control block in Table 1. The units for the values in the **Min** and **Max** columns of the table correspond to the units of the input variable they describe. The fuzzy switches are not listed; each has three inputs. The first input is the criteria for the switch, and is described by two membership functions. The other two inputs are the values to be switched between.

5 Search Implementation

The genetic algorithm used to contrive the rule base for the helicopter fuzzy logic controller is the simple genetic algorithm as described by Goldberg. [8] It employs the three classic genetic operators of *reproduction*, *mutation*, and *crossover.* To use the genetic algorithm as a search technique, two major issues must be addressed: (1) the coding of the parameters, and (2) the development of a fitness function.

The parameters for this problem are the fuzzy rules. These rules are encoded using a conventional multiparameter, mapped, Gray coding, [14] and appear in the bit string in the order they appear in the controller. The rules for each fuzzy control block are mapped using the range and number of bits shown in Table 2. The units for the values in the Min and Max columns of the

Table 1: Descriptions for the inputs of each fuzzy control block.

Block	Input	Number of Members	Min	Max
Longitudinal Acceleration	IAS	3	6	30
	Eu	3	−30	30
	u_dot	3	−1	1
Longitudinal Hold	IAS	3	6	30
	Eu	3	−6	6
	u_dot	3	−1	1
Longitudinal Attitude	E Theta	5	−6	6
	q	3	−10	10
	q_dot	3	−10	10
Lateral Acceleration	Ev	3	−6	6
	v_dot	3	−1	1
Lateral Hold	Ev	5	−6	6
	v dot	3	−1	1
Lateral Attitude	E Phi	5	−6	6
	p	3	−10	10
	p_dot	3	−10	10
Bank Hold	IAS	2	30	60
	E_Psi	3	−20	20
Heading Hold	E Psi	5	−10	10
	Psi_dot	3	−3	3
Bank Attitude	E Phi	7	-10	10
	p	3	−10	10
	p_dot	3	−10	10
Hover Heading	IAS	2	6	12
	E_Psi	7	−30	30
	r	3	−60	60
	r_dot	3	−30	30
Lateral Trim	IAS	2	6	12
	Ev	3	−6	6
	v_dot	3	−1	1
Climb Rate	IAS	2	6	12
	Ez_dot	3	−3	3
	z_dot dot	3	−3	3

Table 2: Parameters used to map the fuzzy rules into a bit string.

Block	# of Rules	Min	Max	Bits
Longitudinal Acceleration	27	−20	20	4
Longitudinal Hold	27	−6	6	4
Longitudinal Attitude	45	−2	2	4
Lateral Acceleration	9	−6	6	4
Lateral Hold	15	−6	6	4
Lateral Attitude	45	−2	2	4
Bank Hold	6	−20	20	4
Heading Hold	15	−10	10	4
Bank Attitude	63	−2	2	4
Hover Heading	126	−2	2	4
Lateral Trim	18	−0.2	0.2	4
Climb Rate	18	−2	2	4

table correspond to the units of the output variable associated with the rules they describe. Since there are 414 rules in the fuzzy controller, each mapped with 4 bits, the resulting encoded bit string is $4 \times 414 = 1656$ bits in length. This yields a search space with $2^{1656} = 3.2 \times 10^{498}$ possible solutions.

In most performance vs. time optimizations using a genetic algorithm, the fitness function is composed of a weighted sum of integrations. The difference between a set of states and their desired values is calculated as error at each time step. The absolute values of the errors are summed over the course of the simulation yielding a cumulative error for each of the states. These cumulative errors are combined in a weighted sum to produce the fitness. [3] In the helicopter problem, the fitness value typically consisted of a weighted sum of integrations of state error over several simulations. Several simulations with different initial conditions and goals were needed to adequately cover the parameter space being searched on. A typical fitness function for N separate simulations lasting T seconds is defined as:

$$fitness = \sum_{IC=1}^{N} \sum_{t=1}^{T} (E_u + E_v + E_w + E_\Psi). \quad (1)$$

The genetic algorithm was applied in two stages to search for the fuzzy rules. In the first stage, the genetic algorithm was employed to search the entire parameter space for a solution that was not immediately divergent. In this stage, the only region of flight considered was hover. In the second stage, each section of the controller was trained independently. First, all rules except those in the Longitudinal Attitude block were excluded from the search. The Longitudinal Attitude block was trained using a fitness function that evaluated short simulations from several sets of initial conditions to various attitude goals. In this way, the control block was trained to reach and maintain a desired pitch angle. Next, the rules for the Longitudinal Hold block were discovered. The fitness function consisted of several simulations at different airspeeds. The criteria for good performance was how well the controller could maintain airspeed despite perturbations.

At this point, the controller could maintain forward airspeed. The next step was to train the Longitudinal Acceleration block to allow airspeed changes. Once again, the fitness function entailed several simulations in which the controller was required to accelerate and decelerate the helicopter through a range of airspeeds. This procedure was repeated for the other three sections of the controller. A few iterations of this process were required to achieve good performance throughout the flight envelope.

6 Results

The helicopter fuzzy logic controller was developed to perform several maneuvers throughout the flight envelope of the UH-1H helicopter. Effective rules for the fuzzy controller were discovered using a genetic algorithm and a numerical model of a UH-1H. The performance of the controller was evaluated both in simulation and in actual flight tests.

Since the fuzzy controller was trained on the numerical model of the UH-1H, its performance on the model indicates the effectiveness of a genetic algorithm in generating fuzzy logic rules. Figure 8 shows the controller performing a simulated hover. In this figure, u, v, and w, are the forward, rightward, and downward velocities of the helicopter measured in meters per second, respectively. The heading (Ψ) is measured in degrees. After a brief period of transition, the controller brings all velocities and the heading to zero.

Figure 9 shows a transition from hover to 5 m/s forward taxi. The heading and lateral and vertical velocities are held near zero. In forward flight, a coordinated turn is performed by flying the helicopter to a new bearing without allowing lateral velocity. Figure 10 shows a coordinated turn at 34 m/s forward airspeed. The airspeed is maintained through a right turn of 60 degrees. Climb rate is held at zero for the duration of the turn. There is, however, oscillation in the lateral velocity. This is a result of a flaw in the controller design. The pedals are used to control heading at hover, but need not be used in forward flight except to trim the aircraft during transition to forward flight. The Lateral

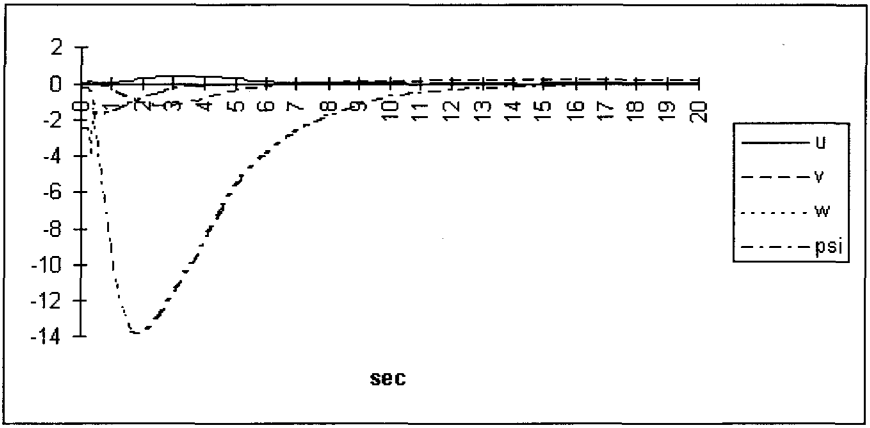

Figure 8: Hover in simulation.

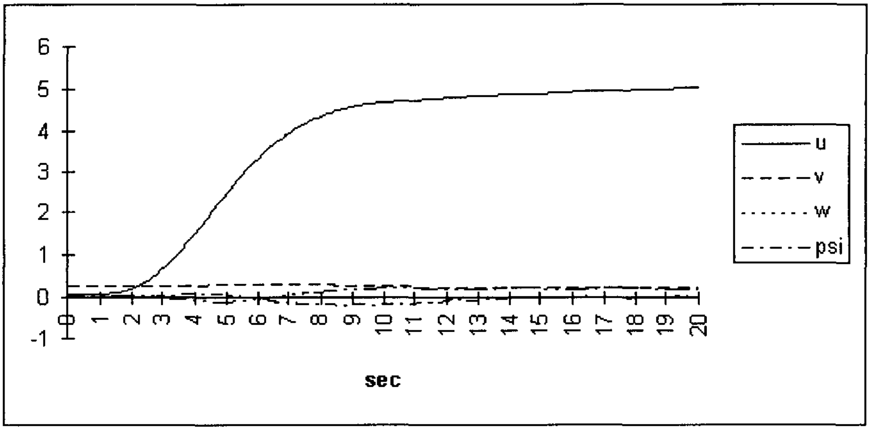

Figure 9: Forward taxi to 5 m/s in simulation.

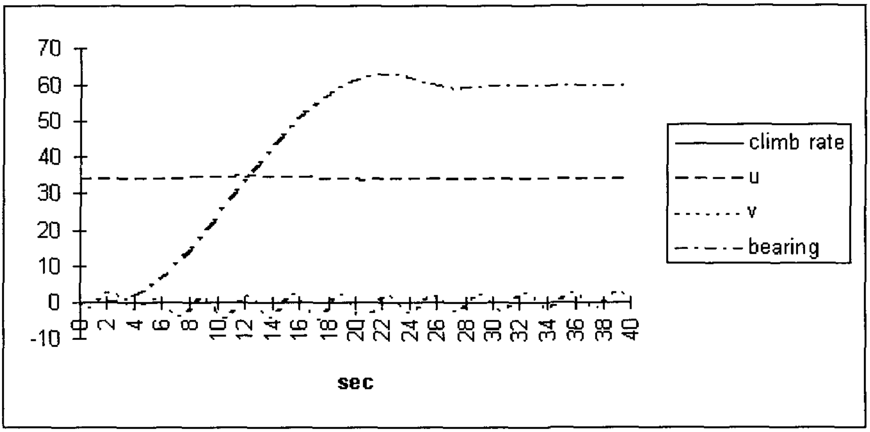

Figure 10: Coordinated turn in simulation.

Trim control block in the helicopter fuzzy controller constantly tries to correct lateral velocity errors in forward flight using the pedals. The fuzzy controller should be redesigned to provide trim during transition only.

Since the fuzzy logic controller performed well in simulation, it was used in an actual flight test. Due to differences between the dynamics represented by the numerical model and the dynamics of the actual aircraft, the performance of the fuzzy controller on the aircraft in actual flight tests indicates the robustness of fuzzy logic for flight control. Figure 11 shows the controller performing a hover in actual flight. It successfully holds all velocities and heading about zero for 90 seconds. In Figure 12, the controller switches the aircraft from hover to a forward taxi at about 5 m/s. Once again, the other velocities and heading remain about zero. Figure 13 shows a coordinated turn in forward flight. The bearing follows the bearing goal throughout the maneuver. While forward velocity is not maintained, the lateral velocity and climb rate are held near zero.

7 Conclusions and Future Work

A fuzzy controller capable of maneuvering a helicopter effectively and aggressively was successfully developed using a genetic algorithm to discover the fuzzy rules. Quality performance was demonstrated in both simulations and actual flight tests. The fuzzy controller architecture developed is general enough to be applicable a variety of rotorcraft. Moving the controller to a new helicopter

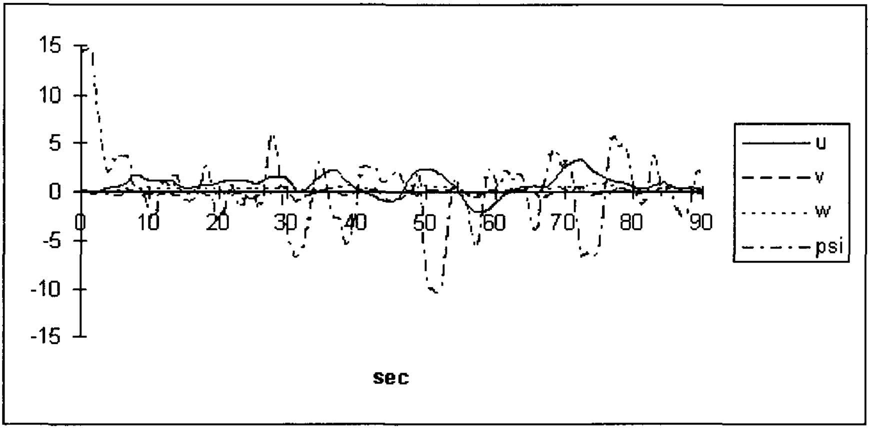

Figure 11: Hover in actual flight.

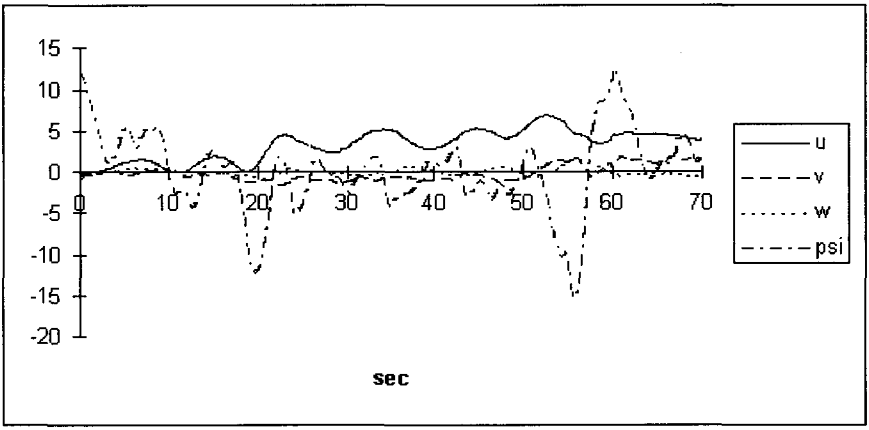

Figure 12: Forward taxi to 5m/s in actual flight.

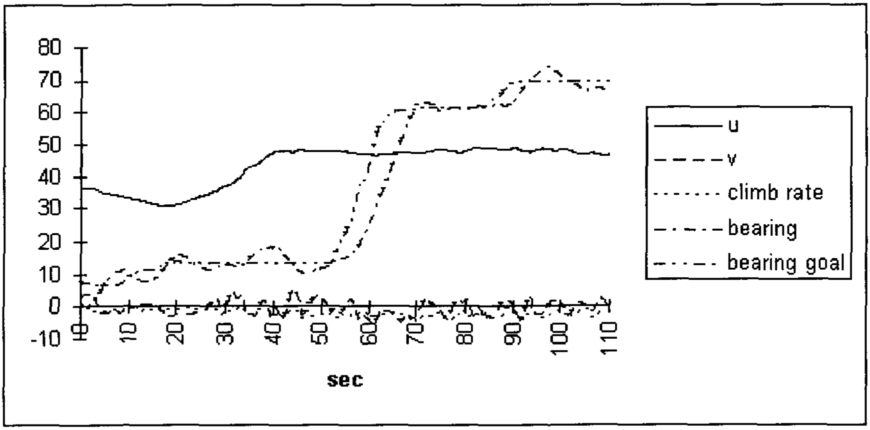

Figure 13: Coordinated turn in actual flight.

simply requires discovering appropriate fuzzy rules with the genetic algorithm.

Currently, the controller is being applied to model scale helicopters. To facilitate this effort, neural network simulators are being developed for the model helicopters. The simulators will be used by the genetic algorithm in discovering rules for the fuzzy controller. Eventually, the architecture of the fuzzy controller must be redesigned to correct problems with tail rotor control in forward flight.

8 References

1. G. H. Saunders, *Dynamics of Helicopter Flight* (Wiley, New York, 1975).
2. T. J. Procyk and E. H. Mamdani, "A linguistic self-organizing process controller," *Automatica* 15 (1978), pp. 15–30.
3. C. L. Karr, "Fine-tuning a cart-pole balancing fuzzy logic controller using a genetic algorithm," *Proceedings of The Applications of Artificial Intelligence VIII Conference* 1468 (1991), pp. 26–36.
4. C. L. Karr, L.M. Freeman and D.L. Meredith, "Improved fuzzy process control of spacecraft terminal rendezous using a genetic algorithm," *Proceedings of Intelligent Control and Adaptive Systems Conference* 1196 (1989), pp. 274–288.
5. C. L. Karr and T. R. Harper, "Genetic algorithms in adaptive fuzzy control," *Proceedings of the Eleventh Annual NAFIPS '92* (1992), pp. 506–514.

6. C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Transactions on Fuzzy Systems* 1(1) (1993), pp. 46–53.
7. C. L. Karr, "Genetic algorithms for fuzzy logic controllers," *AI Expert* 6(2) (1991), pp. 26–33.
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison Wesley, Reading, MA, 1989).
9. F. L. Lewis, *Optimal Control* (Wiley, New York, 1986).
10. M. Sugeno, M. F. Griffin and A. Bastian, *Proceedings of the IFSA Congress* 1(1) (1993).
11. P. Talbot, B. Tingling, W. Decker and R. Chen, "A mathematical model of a single main rotor helicopter for piloted simulation," NASA TM 84281, 1982.
12. M. Mittal and J. V. R. Prassad, "Modeling the UH-1H helicopter using the ARMCOP program," NASA Contract L27548DA, 1994.
13. W. Patterson, "Instruction manual for ARMCOP simulation interface program," NASA Contract L27548DA, 1994.
14. F. Gray *Pulse Code Communication* U. S. Patent 2 632 058, 1953, March 17.