JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 5, September–October 1995

Hybrid Fuzzy Logic Flight Controller Synthesis via Pilot Modeling

K. KrishnaKumar*
University of Alabama, Tuscaloosa, Alabama 35487
P. Gonsalves†
Charles River Analytics Inc., Cambridge, Massachusetts 02138
A. Satyadas‡
Flexible Intelligence Group, LLC, Tuscaloosa, Alabama 35486
and
G. Zacharias§
Charles River Analytics Inc., Cambridge, Massachusetts 02138

This paper presents an investigation of a hybrid technique developed for synthesizing fuzzy logic controllers as stability augmentation systems. The hybrid technique combines the control capabilities of fuzzy logic with the learning capabilities of genetic algorithms, to yield a fuzzy logic controller augmentation system optimized to satisfy desired handling qualities requirements. An optimal control model is used to provide the closed-loop handling qualities metrics. The optimal control model is an analytic model of the pilot which can support the computation of in-flight handling qualities metrics, such as the standard Cooper–Harper rating. A genetic algorithm is used to optimize the attributes of the fuzzy logic controller. These attributes include the parameters of the fuzzy logic controller membership functions and the rule structure. The hybrid technique was implemented and tested in an offline engineering simulation using a wide-envelope F/A-18 longitudinal model. The tests included examining the robustness of the fuzzy logic controllers, the robustness of the genetic algorithms optimization technique, and the effect of changing the number of rules used in the fuzzy logic controller. Results indicate that the approach provides a robust design technique for fuzzy logic controller stability augmentation system synthesis and also show that the synthesized fuzzy logic controllers possess good robustness qualities.

Nomenclature

G	= weighting matrix
HQR	= handling quality rating
K	= constant of proportionality
Q_x, Q_y, Q_u	= weighting matrices
q	= pitch rate, rad/s
V_t	= air speed, ft/s
α	= angle of attack, rad
δ_s	= stabilator angle, deg
θ	= pitch angle, rad
σ	= standard deviation

Introduction

MODERN high-performance aircraft rely heavily on the inner-loop control system augmentation to provide stability that has been traded for high performance and maneuverability. This poses a challenge to control system designers: to design control systems that can counteract static and dynamic instabilities, while providing satisfactory handling qualities. In addition to this, the ever expanding flight envelope and the requirements of large maneuvers introduce significant uncertainties in the dynamics under control. With this growing complexity, significant reduction in time and money involved in the design process could be achieved if an effective design technique brought to bear some of the emerging technologies now being applied to a number of nonflight control problems. These techniques include fuzzy logic control (FLC), genetic algorithms (GA), and artificial neural networks (ANN). In this paper, we examine the feasibility of using fuzzy logic controllers for providing acceptable inner-loop stability augmentation for a modern fighter aircraft. Fuzzy logic affords a mechanism for incorporating the uncertainty inherent in most control problems into conventional expert systems. FLC uses simple rules to compute control actions for the given state of the system. The robustness of FLC has been demonstrated for control problems ranging from acidity (pH) control[1] to helicopter control.[2]

Fuzzy control of complex, nonlinear systems require careful synthesis of membership functions and rules. General practice has been to either use experts' experience and knowledge or model an existing process. For example, in Ref. 2, the rules and the membership functions were iterated upon by researchers manually. In many instances this is just not possible or the time consumed might be prohibitively high. In such instances, a performance-driven synthesis can be achieved via exploratory search techniques. In the past, optimization techniques have been used to tune the parameters of predefined rules. Karr and Gentry,[1] Takagi and Sugeno,[3] Karr,[4] and Freeman et al.[5] have all used some form of a parameter optimization technique for fine tuning the membership functions. In all of these studies,[1–5] either all of the possible rules were used or the rules were predefined and only the parameters of the membership functions were altered. Depending on the number of membership functions and fuzzy variables, the number of possible rules can be very high. It is a known fact that finer fuzzy partitioning is needed for better control performance.[6,7] Given n variables and m fuzzy partitioning, the possible number of rules assuming one possible fuzzy relationship, is m^n. If $n = 4$ and $m = 4$, we have 256 possible rules. Using all possible rules is not recommended for two reasons: 1) computational efficiency associated with fuzzy logic is lost when using a high number of rules and 2) for control applications, robustness decreases with increasing number of rules. This poses a severe

Received July 5, 1994; revision received Oct. 26, 1994; accepted for publication Oct. 26, 1994. Copyright © 1995 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Associate Professor, Department of Aerospace Engineering. Senior Member AIAA.
†Scientist. Member AIAA.
‡Principal Scientist.
§Principal Scientist. Member AIAA.

limitation in the design of FLC for stability augmentation systems (FLC SAS). Another drawback of using FLC for inner-loop control augmentation emanates from the need for a synthesis procedure that provides a closed-loop system with acceptable handling qualities.

The major objective of this study is to outline a hybrid approach for synthesizing FLCs and to verify their capabilities by applying it to a longitudinal SAS design for a F/A-18 model. The uniqueness of this methodology lies in the combination of the control capabilities of fuzzy logic with the learning capabilities of genetic algorithms, to yield a FLC augmentation system optimized to satisfy desired handling qualities requirements. Also, the hybrid approach optimizes both the FLC rule structure and the membership function parameters. As stated earlier, this is different from the studies already referenced. A genetic algorithm optimization technique is successfully used to optimize the membership functional form and parameters, and the FLC rule base. Genetic algorithms are biologically inspired highly parallel mathematical search algorithms pioneered by Holland.[8,9] GAs combine a Darwinian principle of reproduction and survival of the fittest and natural genetic operations such as mutation and crossover. This technique has gained popularity in recent years as a robust optimization tool for a variety of problems in control.[10–12] Since GAs are robust over a wide spectrum of optimization criteria and problem domains, they are well suited for optimizing the FLC attributes based on an objective function computed via the use of the optimal control model (OCM). The OCM is an analytic model of the pilot that can support the computation of in-flight handling qualities metrics, such as the standard Cooper–Harper rating.[13] In essence, the OCM-based handling quality metric serves as the fitness function to be optimized by the GA, via appropriate tuning of the FLC. In this manner, we obtain the full design flexibility available with an FLC SAS design, while minimizing the heuristic (and often artistry) inherent in specifying many attributes of the FLC design components. It is noted here that this hybrid methodology could be used with any pilot modeling technique that gives an objective numerical measure. The choice of the OCM is based on its ability to handle multi-input/multi-output systems. In spite of many tunable parameters associated with the approach, researchers have found good success in applying OCM to many man-in-the-loop system designs.[14–21]

In what follows, we present brief overviews of FLC, GA, and OCM techniques and outline the proposed hybrid technique. Performance and robustness results obtained using F/A-18 longitudinal linear models are then presented.

Fuzzy Logic Control

In recent years, rule-based systems have become increasingly popular as practical applications of artificial intelligence. These expert systems have performed as well as humans in several problem domains. Their lack of flexibility in representing the subjective nature of human decision making, however, limits their performance in domains where it is difficult to formulate crisp rules and situations for acting on those rules. This uncertainty, inherent in human decision making, can be incorporated into expert systems using fuzzy set theory.[22] In fuzzy set theory, linguistic variables can represent abstract or subjective concepts. Linguistic variables and, thus, the uncertainty associated with human decision making, have been incorporated into expert systems in the form of fuzzy logic controllers.[23] FLCs are rule-based systems that use fuzzy linguistic variables to model a human's rule-of-thumb approach to problem solving. FLCs have performed successfully in a number of control problems.[23–25] These FLCs include rules to direct the decision process and membership functions to convert linguistic variables such as high and low into precise numeric values needed for the control execution. The rule set is gleaned from a human expert's knowledge, which is generally based on his experience. The FLC developer chooses the membership functions to represent the human expert's interpretation of the linguistic variables. The selection of high-performance membership functions is commonly the most time consuming phase of FLC development. A change in the membership functions alters the performance of the controller, because it is the membership functions that govern the effectiveness of a rule. Thus, for a given set of rules, the performance of the FLC can be significantly reduced by an inappropriate choice of membership functions. In the following paragraphs we further discuss the specifics of fuzzy logic control by using the FLC SAS developed in this study as an example.

Fuzzy Logic Control Stability Augmentation System Example

Fuzzy reasoning employs the techniques of fuzzy logic in an IF-THEN rule base to perform a given task. A functional block diagram of fuzzy logic for control is shown in Fig. 1. As shown, fuzzy reasoning consists of three main blocks: a fuzzifier, a decision logic or inference system, and a defuzzifier.

The first step in developing the FLC is to determine which variables will be important in choosing an effective control action. For the SAS problem studied in this paper, airspeed V_t, angle of attack α, pitch angle θ, and pitch rate q are the decision variables and stabilator δ_s is the control variable. Once the important decision and control variables have been identified, the linguistic variables that will be used to describe these variables must be defined (fuzzy sets). As an example, five fuzzy sets are used to characterize each of the four decision variables and the control (in the results presented later, we use eight fuzzy sets). Figure 2 illustrates an example fuzzy set for the decision and control variables using Gaussian membership functions (other functions such as trapezoidal or triangular can also be employed). The fuzzifier acts on the system measurements (V_t, α, θ, and q) and performs the mapping of deterministic numerical data (a crisp set) into fuzzy sets. Given a measurement value for V_t, α, θ, or q, the fuzzifier interprets it as a fuzzy set Fx with membership function $\mu(V_t, \alpha, \theta,$ and $q)$ with $\mu() \in [0, 1]$. Referring to Fig. 2, a θ of 0 deg might be viewed as $\mu(F\theta_1) = 0.0$, $\mu(F\theta_2) = 0.4$, $\mu(F\theta_3) = 1.0$, $\mu(F\theta_4) = 0.8$, and $\mu(F\theta_5) = 0.0$. When a fuzzy membership function has a value of $\mu = 1$, the confidence level of the precise numeric value being accurately described by the fuzzy set is maximum. On the other extreme, when $\mu = 0$, the confidence level of the precise numeric value being accurately described by the fuzzy set is minimum. It is important to realize that for each precise decision value each fuzzy set has a membership function value, i.e., $\theta = 0$ deg is $F\theta_3$ with a certainty of 1.0, $F\theta_4$ with a certainty of 0.8, $F\theta_2$ with a certainty of 0.4, and is described by each of the other two classes with a certainty of 0.

Now that the precise numeric conditions existing in the system at any given time can be categorized in a fuzzy set with some certainty, the fuzzy sets are processed via some decision logic consisting of

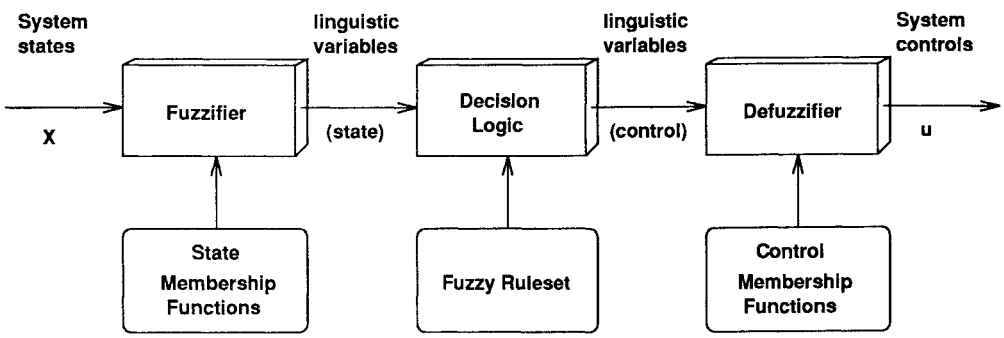

Fig. 1 FLC functional block diagram.

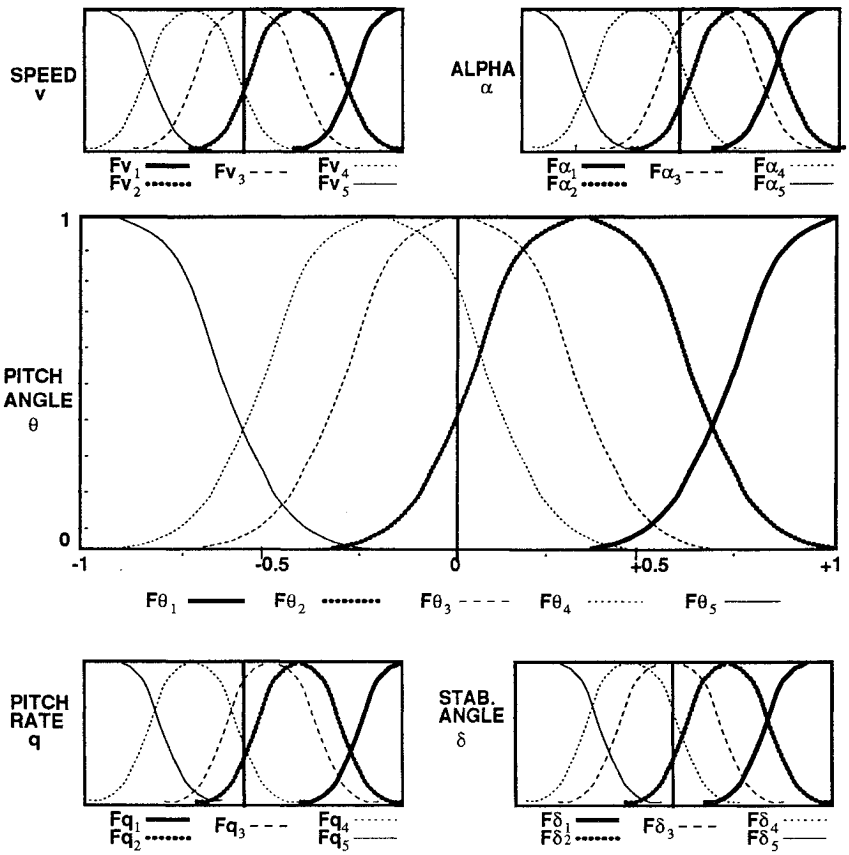

Fig. 2 Typical fuzzy sets.

linguistic rules. The formation of the rule set is comparable to the process that must exist in the development of any expert system, except that the rules incorporate linguistic variables (or fuzzy sets). As stated earlier, in situations in which simple rules cannot be formulated, determining effective rules become a difficult task. The choice of fuzzy sets just described allows for the possibility of $5^4 = 625$ different conditions that could exist when the rules are of the form

$$\text{IF } [\theta \text{ is } F\theta_1 \text{ AND } q \text{ is } Fq_2 \text{ AND } \alpha \text{ is } F\alpha_3 \text{ AND } V \text{ is } Fv_4]$$

$$\text{THEN } [\delta_s \text{ is } F\delta_1] \tag{1}$$

where $Fx = [F\theta_1, Fq_2, F\alpha_3, Fv_4]$ and $Fu = [F\delta_1]$ are fuzzy sets characterizing the respective variables.

The IF part of the rule is called the antecedent, and the THEN part is called the consequence. These rules thus relate the input measurements into the outputs. The mapping from the fuzzy set Fx into the fuzzy set Fu is called a fuzzy relation. Several methods are available for translating the fuzzy condition $Fx \rightarrow Fu$ into a fuzzy relation, including Zadeh's method[26] based on Lukaziewicz's implication. The procedure for computing the fuzzy relation curve for the action is outlined as follows.

For $i = 1$: Number of rules

$$Mx = \min[\mu(F\theta_j), \mu(Fq_j), \mu(F\alpha_j), \mu(Fv_j)]_i$$

$$j \in [1, 2, \ldots, \text{no. of fuzzy sets}]$$

$$h_i(\delta_s) = \min[1, (1 - Mx + F\delta_j(\delta s)]_i \quad \delta s_{\min} \leq \delta_s \leq \delta s_{\max}$$

end

For $\delta = \delta s_{\min} : \delta s_{\text{step}} : \delta s_{\max}$

$$H(\delta) = \min[h_1(\delta), h_2(\delta), h_3(\delta), \ldots, h_{\text{no. of rules}}(\delta)]$$

end

In this procedure, $h_i(\,)$ is the fuzzy action curve for the ith rule, $H(\delta)$ is the fuzzy relation curve for the control δ_s, and δs_{step} is the discretization step size. The final element of the functional block diagram for fuzzy reasoning is the defuzzification process. The defuzzifier acts on the decision logic output variables or fuzzy control actions and performs the mapping to the corresponding deterministic numerical data (a crisp set). That is, it converts the output from a fuzzy set to a numerical value used as a control input into a plant. In short, the defuzzification process involves determining the range of values of the output variables and performing the transformation that maps the fuzzified control action into a corresponding nonfuzzy action. Defuzzification is required because in most practical circumstances, a crisp output action is necessary. The main goal of defuzzification is to best represent the possibility distribution of an inferred fuzzy output action. Three popular techniques for defuzzification are[27] 1) max membership, which involves choosing the element that has the maximum membership in the output fuzzy set, 2) mean-of-maxima, which generates a value that represents the mean of control functions whose membership is maximum, and 3) centroidal or center of area, which uses the centroid of the possibility function of the control action. In this study, we use the centroidal method of defuzzification given as follows:

$$\delta s = \left[\int_{\delta s_{\min}}^{\delta s_{\max}} H(x)x\,dx\right] \Big/ \left[\int_{\delta s_{\min}}^{\delta s_{\max}} H(x)\,dx\right] \tag{2}$$

As we noted earlier, the key requirements for developing an effective FLC are 1) the appropriate specification of the membership functions, 2) the definition of the fuzzy production rules, and 3) the strategy chosen for defuzzification. Later we describe how these requirements can be met via the hybrid GA approach.

Genetic Algorithms

Genetic algorithms are different from normal search methods encountered in engineering optimization in the following ways.[9,10] 1) GAs work with a coding of the parameter set, not the parameters themselves. 2) GAs search from a population of points, not a single point. 3) GAs use probabilistic transition rules, not deterministic transition rules.

Genetic algorithms require the natural parameter set of the optimization problem to be coded as a finite length string. This string

could be either a string of binary numbers or real numbers. As an example, a parameter optimization with two parameters is considered. For this optimization problem, the two parameters are discretized by mapping from a smallest possible parametric set $K_{\min}$ to a largest possible parametric set $K_{\max}$. This mapping uses a 10-bit binary unsigned integer for both K_1 and K_2. In this coding a string code 0000000000 maps to $K_{\min}$ and a 1111111111 maps to $K_{\max}$ with a linear mapping in between. Next, the two 10-bit sets are chained together to form a 20-bit string representing a particular controller design. A single 20-bit string represents one of the $2^{20} = 1{,}048{,}576$ alternative solutions. Genetic algorithms work iteration by iteration, generating and testing a population of strings. This population-by-population approach is similar to a natural population of biological organisms where each generation successively evolves into the next generation by being born and raised until it is ready to reproduce.

Optimal strings are found through population reproduction via selection, crossover, and mutation. Selection is a process where an old string is carried through into a new population depending on its performance index (fitness function) value. Because of this move, strings with above average fitness values get larger numbers of copies in the next generation. This strategy, in which good strings get more copies in the next generation, emphasizes the survival-of-the-fittest concept of genetic algorithms.

A simple crossover follows selection in three steps. First, the newly selected strings are paired together at random. Second, an integer position n along every pair of strings is selected uniformly at random. Finally, based on a probability of crossover, the paired strings undergo crossover at the integer position n along the string. This results in new pairs of strings that are created by swapping all the characters between characters 1 and n inclusively. Mutation is simply an occasional random alteration of a string position (based on probability of mutation). In a binary code, this involves changing a 1 to a 0 and vice versa. The mutation operator helps in avoiding the possibility of mistaking a local minimum for a global minimum. When mutation is used sparingly (about one mutation per thousand bit transfers) with selection and crossover, it improves the global nature of the genetic algorithm search.

One of the benefits of genetic algorithm is the problem-independent characteristics of the search scheme. This enables a black-box treatment of the GA code. That is, the GA supplies the parameters to the optimization problem at hand and, in return, the specific problem-dependent software provides the fitness function. This fitness function is then utilized by the GA to evolve to the next generation.

Optimal Pilot Model

The optimal control model (OCM) of the human pilot is an information-processing model which has been developed within the systems framework of modern control and estimation theory.[14] The basic assumption underlying the model is that the well-trained well-motivated pilot behaves optimally in some sense, subject to inherent psychophysiological limitations that constrain the range of his behavior. The model is capable of predicting steady-state task performance (e.g., rms lock-on-error during vehicle pointing and weapons aiming), frequency-domain pilot transfer functions (e.g., stick response to a wind gust), frequency-domain pilot remnant (e.g., stick jitter), and time-domain dynamic histories (e.g., vehicle attitude during a hammerhead stall).

A general block diagram of the OCM is given in Fig. 3; a detailed description may be found in Ref. 14. As shown, the model comprises the following.

1) A set of linear system dynamics define the basic system states $\boldsymbol{x}$ to be controlled by the pilot.

2) A display interface converts system states $\boldsymbol{x}$ into displayed variables $\boldsymbol{y}$, seen by the pilot.

3) A perceptual model converts these display variables $\boldsymbol{y}$ into noisy and time-delayed perceived variables, denoted by $\boldsymbol{y}_p$.

4) An estimator/controller block converts these perceived variables $\boldsymbol{y}_p$ to commanded control actions $\boldsymbol{u}_c$. As shown, this consists of an optimum (Kalman) estimator which, in tandem with a predictor, generates a minimum variance estimate of the current system state $\boldsymbol{x}$ and a set of optimal gains that are chosen to ensure that the resulting command $\boldsymbol{u}_c$ will minimize a predefined cost function that expresses the task requirements.

5) A neuromotor model converts this command $\boldsymbol{u}_c$ into a piloted control action $\boldsymbol{u}$. As shown, this is accomplished by the formal addition of motor noise to the command $\boldsymbol{u}_c$, and limited-bandwidth filtering, which together account for the pilot's neuromotor bandwidth limitations, and his inability to generate perfectly precise control actions.

In this study, we used the OCM as a design evaluation tool for predicting expected pilot handling quality ratings (HQR), as a function of the FCL-based SAS design. The rationale for this approach comes from several previous studies which compared OCM-based predictions of HQR for a given configuration with actual requirements obtained during pilot-in-the-loop simulation testing of the same configuration. Hess[15,16] was the first to show that OCM-based HQR ratings could predict human pilot ratings to within ± 1 rating unit, if the HQR is computed according to

$$HQR = K \log(J) + (HQR)_0 \tag{3}$$

where

$$J = E\left(x^T Q_x x + y^T Q_y y + u^T Q_u u + \dot{u}^T G \dot{u}\right) \tag{4}$$

is the predefined quadratic cost function that defines the task requirements, and $(HQR)_0$ is a residual HQR parameter (a constant for a given problem[16]). In his study, Hess[15] showed this relationship to hold across 19 different control configurations evaluated in his study. Inclusion of a control rate term in the performance index accounts indirectly for the physiological limitations on the rate at which a human can effect control. The inclusion of this rate term

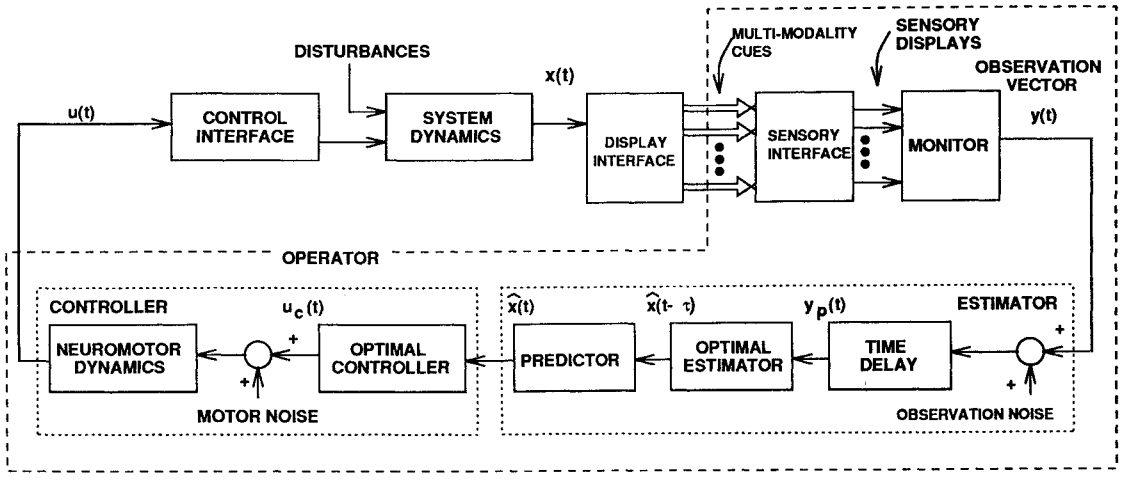

Fig. 3 Optimal control model of pilot/vehicle system.

results in a first-order lag being introduced in the optimal controller. This lag has often been attributed to the human neuromotor system. The selection of the weighting matrices, Q_x, Q_y, Q_u, and G is a nontrivial step in applying the OCM. The most commonly used method for selecting reasonable a priori estimates for the output weightings Q_y is to associate them with allowable deviation in the system variables.[14] The state and control related weightings (Q_x, Q_u, and G) may be chosen in a similar fashion or they may be picked to yield a desired response bandwidth. Several studies have made use of OCM-based HQR ratings with good success.[17–20]

Hybrid Fuzzy Logic Controller/Genetic Algorithm/Optimum Control Model

In these and other contemplated GA optimization efforts of the flight control system (FCS) design, we can use the OCM of the pilot to generate the GA fitness function [Eq. (4); since K and (HQR) are constants for a given problem, minimizing J is same as minimizing HQR]. As discussed earlier, this is a model-based prediction of the handling qualities rating for the given FCS configuration. What is involved here is a basic five-step process.

1) Define the piloting task and standard pilot model parameters specifying the pilot's inherent limitations and capabilities (Q_x, Q_y, Q_u, and G).

2) Define the non-FCS components of the overall loop, with particular attention paid to the unaugmented vehicle dynamics.

3) Define or modify the FCS configuration, structure, and controller parameters. For the FLC synthesis this involves a) identifying state and control criteria to be used in the FLC; b) defining the type of membership function to be used (Gaussian, triangular, etc.), number of linguistic variables, and number of rules; and c) identifying the appropriate fuzzy inference mechanism and defuzzification scheme.

4) Exercise the OCM to generate expected pilot-in-the-loop performance and handling qualities, and generate the HQR-based fitness value [Eq. (4)].

5) Search in the coded parameter space using the GA to minimize J as defined in Eq. 4. Iterate on the design by looping back to step 3 until an acceptable design is achieved.

We believe that this systematic approach, with its reliance on an objective model-based fitness function, can minimize the number of subjective design decisions facing the flight control system designer, while taking full advantage of the search strength of GAs in an FLC structure. The hybrid FLC SAS architecture used for this portion of the study is shown in Fig. 4. The software development was conducted using the MATLAB software package. Special routines for the GA and FLC modules were developed and integrated with an OCM module.

Objective Function Module

The objective function module synthesizes an optimal pilot model for a given fuzzy logic controller. Implicit in the optimal pilot model definition is a linear pilot model with state estimation and optimal feedback control. To compute this optimal model of the pilot, the system has to be linear. For nonlinear systems, this approach can be used as a quasilinear approach in which the system is linearized at several points along the trajectory and pilot models are derived for every linearized system. To have a valid performance measure, the system has to be simulated many times to sample the whole state space. Although this is feasible, when combined with the genetic algorithms search, the approach becomes computationally very expensive. In our study, we have used linear models that represent various operating points in the flight envelope and have used linearized versions of the FLCs. The nonlinear FLC is linearized about the operating point using three different perturbation scales (small, medium, and large), and the stability of the ensuing closed-loop system is verified. If the system is found to be unstable, the FLC is given a low-performance rating (a poor rating computed as 100 plus the maximum real part of the closed-loop eigenvalues). If the system is stable, then the closed-loop system corresponding to the small perturbation linear-FLC model is used to compute the optimal J [Eq. (4)]. This measure is then supplied to the GA module.

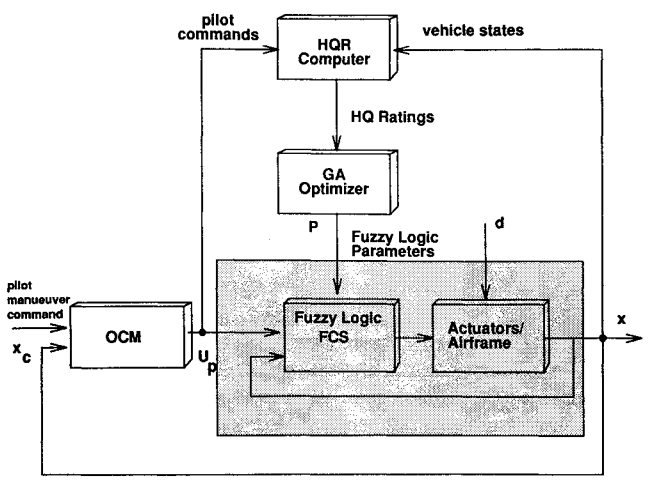

Fig. 4 Hybrid FCS architecture.

Table 1 Scaling values for fuzzy membership base set definition

Aircraft variables	Scaling values
Airspeed, ft/s	±50
Angle of attack, deg	±15
Pitch angle, deg	±15
Pitch rate, deg/s	±15
Stabilator, deg	±2.5

Fuzzy Logic Control Module

The FLC module consists of a rule base, an inference engine, and a defuzzifier.

The rule base subsystem consists of a set of states, control variables, and rules. In this study, the states are: airspeed V_t, angle of attack α, pitch angle θ, and pitch rate q. Stabilator angle δ_s is the action variable. Linguistic variables associated with the state and control variables are defined using symbols rather than words. Figure 2 presents a typical membership function set and a sample rule is presented in Eq. (1). For the membership functions, we have used the Gaussian function, whose scalar output y corresponding to a scalar input x is

$$y = \frac{1}{\sigma\sqrt{2\Pi}} e^{[-(x-\mu)^2/2\sigma^2]} \tag{5}$$

where the mean μ and standard deviation σ are the parameters. The parameters of the membership functions and the rules are supplied by the GA module. We use a base set of $[-1, +1]$ implying that all of the states and control are scaled to lie between -1 and $+1$. The scaling used in this study is given in Table 1.

The inference engine subsystem applies the rules and membership functions to determine the stabilator controls for given state values. In this study, symmetry in the rule structure is maintained to ensure zero control for zero state error.

The defuzzifier subsystem computes the centroid of the fuzzy conjunction curve [Eq. (2)]. This crisp value (stabilator controls) is the output of the FLC module. The GA, objective function, and FLC modules work together as a closed-loop control system during this adaptive evolutionary learning phase.

Genetic Algorithm Module

The GA module consists of encoding, evolution, and decoding subsystems. The population is represented in the GA domain as a character string of 0s and 1s. In Fig. 5, we show an example of the coding scheme used in this study. For the FLC implementation, we have four states (V_t, α, θ, q) and one control action (δ_s). Four linguistic variables (along with their mirror images) are defined for each of the states and control. For the membership function parameters, we choose four bits each for the mean and standard deviation of the Gaussian function. These four bits translate to a

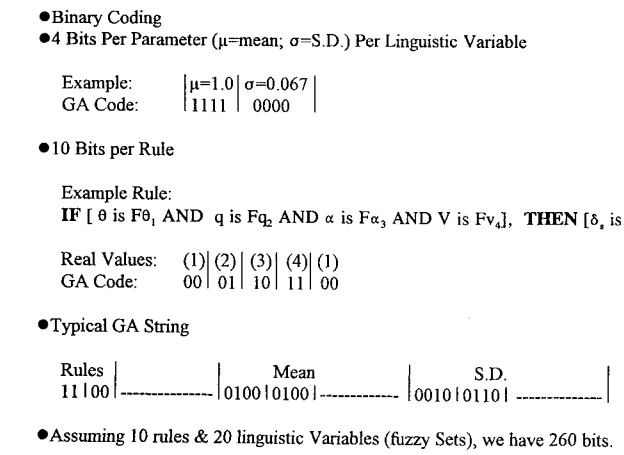

Fig. 5 GA coding.

resolution of 0.1333 and a range of -1 to $+1$ for the mean and a range of 0 to $+1$ with a resolution of 0.0667 for the standard deviation. Since each rule consists of four states and one control with four possible linguistic variables each, we have $4^5 = 2^{10}$ possible combinations. This translates to having a 10-bit string to represent each rule. The GA string is arranged with all the rule bits first, mean bits second, and the standard deviation bits last. For an example of 10 rules and 20 linguistic variables, we have a total string length of 260 bits (10 rules * 10 bits/rule +20 linguistic variables * 2 parameters/variable *4 bits/parameter = 260).

This N population m bit strings is first randomly generated by the GA. The decoding subsystem generates the actual values, which form the output of the GA module. The FLC module then uses the parameter values set by GA module. For each string in the population, the objective function module generates a fitness value that is based on the OCM HQR computation. This fitness sequence is repeated for all of the members of the population. The next step is to use this information to breed the next generation. The evolution subsystem breeds the next generation, using reproduction, crossover, and mutation. For crossover, two sites are selected: one for the rules and one for the parameters. This cycle is repeated for 30 generations. At the end of 30 generations, the best member of the population is decoded to define the parameters and rules for the FLC SAS.

Performance Results

Performance evaluation of the GA-optimized FLC SAS was carried out using linear longitudinal models derived from a F/A-18 nonlinear longitudinal simulation. These models are presented in the Appendix. The flight envelope ranged from 0 to 40,000 ft in altitude and 600 to 800 ft/s in speed. Performance evaluations were for a pitch initial condition disturbance (i.e., pitch angle initially set to 5 deg, all others states are zero). Initially, we specified the requisite parameters associated with the OCM. These are listed in Table 2 and were chosen based on acceptable handling qualities as specified by steady-state tracking performance and on previous experience in modeling the human pilot in flight control tasks.[21] We investigated performance of the GA optimized FLC SAS across the flight envelope for varying number of fuzzy rules, and lastly, we investigated robustness of the controller to system parameter variations. In synthesizing a FLC SAS using GA, the objective was not to find the optimal solution but to get an acceptable solution with a reasonable search effort. In all of the optimization runs, the GA used a population size of 40 and evolved for 30 generations resulting in a total of 1200 function evaluations.

Performance Across Flight Envelope

To investigate performance across the flight envelope, offline GA optimization runs were performed at many flight conditions within the flight envelope. These flight conditions and the appropriate system matrices are given in the Appendix. The GA optimizes the FLC membership function parameters (i.e., mean and standard deviations for Gaussian membership functions) and the FLC rule-base structure based on an OCM rating of HQ. A total of 20 rules were used for all of the runs. At each step in the optimization process, a linearization of the closed-loop FLC and vehicle dynamics is performed, a determination of the OCM-based HQR is made, and finally the FLC is altered by the GA. Figures 6 and 7 show the resulting system transient responses in nulling out an initial condition (IC) pitch offset of 5 deg for two example flight conditions. The upper portion of each figure shows the open-loop pitch response (dashed line) and the closed-loop GA-optimized FLC SAS pitch response (solid line); the lower portion shows the stabilator command. As seen, the FLC SAS provides good response to the initial disturbance with minimum controller requirements, as compared with the oscillatory and slow open-loop responses. It is noted here that the stabilator is highly effective at high-dynamic pressures (low altitudes and high airspeeds) and the effectiveness decreases with decrease in dynamic pressure. This is evident in all the stabilator plots shown.

Table 2 OCM parameters set for GA-optimized FLC SAS

Parameters	Values
Base	
Motor noise ratio, dB	−20
Perceptual noise ratio, dB	−40
Motor time constant, s	0.1
Perceptual time delay, s	0.2
Cost	
Velocity error, ft/s	10
Angle of attack error, deg	15
Pitch error, deg	10
Pitch rate error, deg/s	15
Stabilator, deg	30

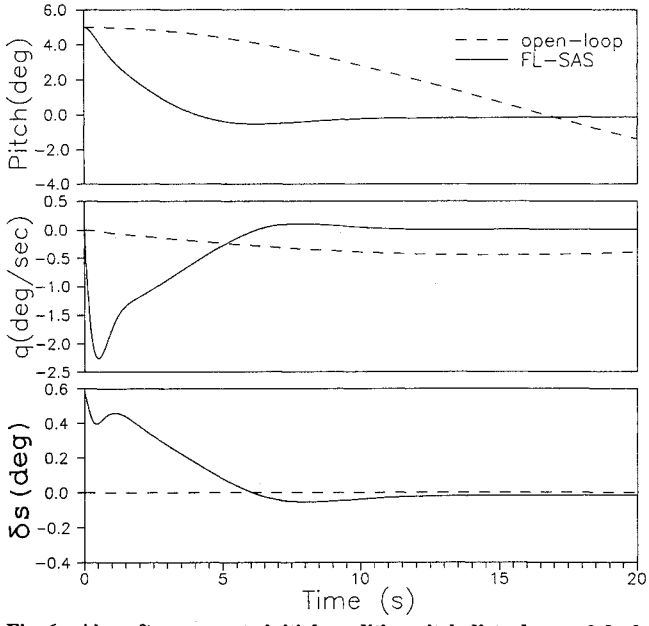

Fig. 6 Aircraft response to initial condition pitch disturbance, 0-ft altitude and 800-ft/s velocity flight condition.

Performance with Fuzzy Logic Controller Rule-Base Variation

To determine the effect of varying FL parameters, we investigated the performance of the FLC SAS with different numbers of rules in the decision-making rule base. The nominal number of rules used for runs across flight conditions was 20. Using the 0-ft flight condition, we compared pitch IC performance for increasing rule-base size to 30 rules, and decreasing the rule-base size to 10 rules. The results are shown in Fig. 8 for pitch, pitch rate, and stabilator responses. As shown, halving the rule-base size (i.e., to 10 rules) does not substantially change performance, nor does doubling the size (i.e., to 30 rules). As a matter of fact, the 10-rule case has lower overshoot and better performance. This is attributed to the fact that the problem

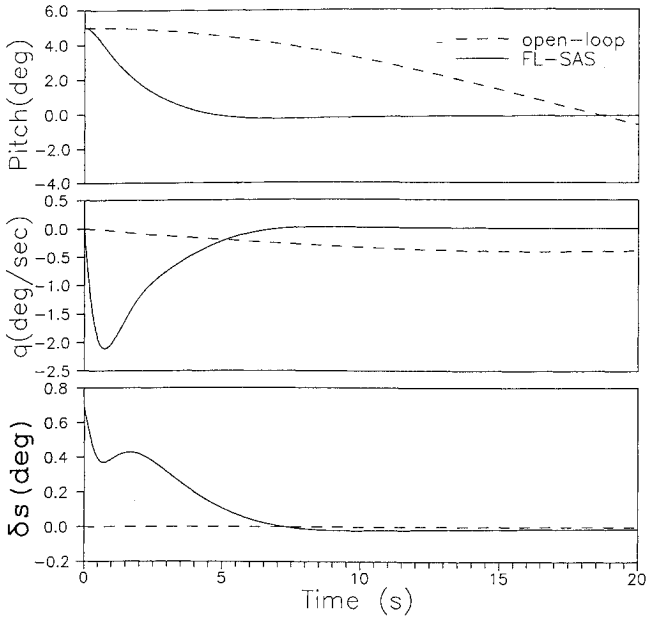

Fig. 7 Aircraft response to initial pitch disturbance, 20,000-ft altitude and 800-ft/s velocity flight condition.

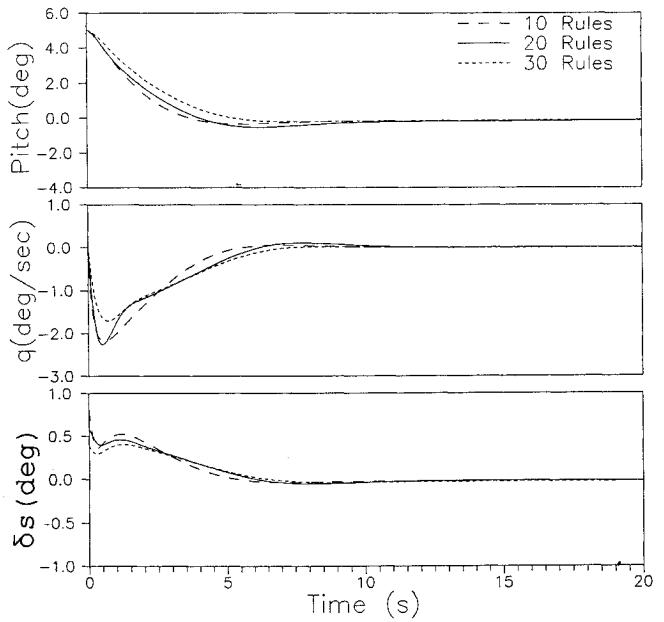

Fig. 8 Aircraft response to initial condition pitch disturbance, FLC number of rules variation.

with 10 rules had the lowest string length (260) resulting in a better solution obtained by GA in 1200 function evaluations. One can also conclude that acceptable performance can be obtained with as little as 10 rules. Efforts were not spent in trying to find the lowest possible set of rules that will provide an acceptable performance. This could be a topic of further research value in this field.

Fuzzy Logic Controller Robustness to System Variation

Finally, we investigate the robustness of the hybrid FLC SAS. Specifically, we are concerned with how well the GA-optimized FLC for a given flight condition performs across the flight envelope. For this demonstration, we chose the FLC SAS from the 20,000-ft flight condition and evaluated it at two other flight conditions: 1) 0-ft altitude and 600-ft/s velocity and 2) 40,000-ft altitude and 600-ft/s velocity. The resulting performance (as given by pitch, pitch rate, and stabilator responses) is shown in Fig. 9. For the 0-ft flight conditions, we see slightly slower, more highly damped performance using the 20,000-ft FLC SAS. The 20,000-ft FLC SAS

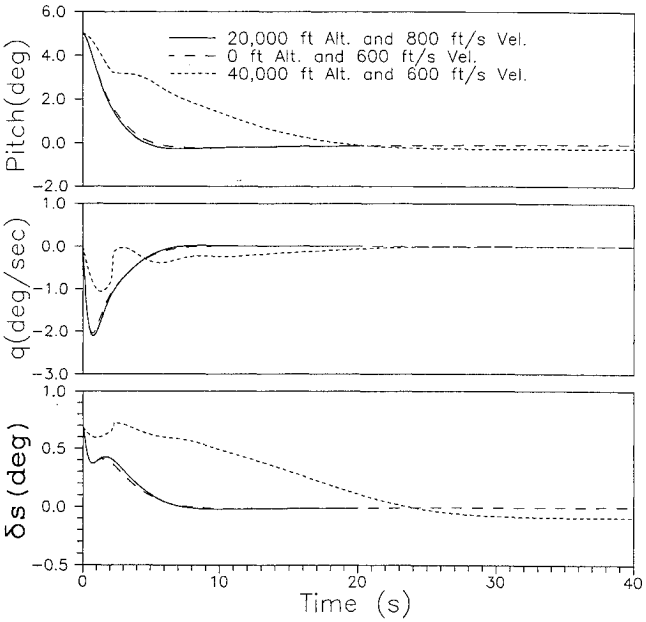

Fig. 9 Aircraft response to initial condition pitch disturbance, FLC SAS robustness across altitude and airspeed variations.

also maintains stability at the 40,000-ft altitude, which is open-loop unstable. These results point to the fact that the GA-optimized FLC SAS is fairly robust across the altitude and airspeed range tested, although additional testing is certainly necessary to evaluate full envelope performance.

Conclusions

This paper has demonstrated a hybrid technique for synthesizing fuzzy logic controllers (FLC) for inner-loop stability augmentation. The hybrid technique combines the learning capabilities of FLC with the optimization capabilities of a genetic algorithms (GAs) in finding fuzzy logic controllers that provide acceptable handling qualities. The handling qualities metrics are computed via an optimal control model (OCM) of the pilot. The GA was used to optimize the membership functions and the rule-base of the FLC. The results of applying this technique to a F/A-18 longitudinal model document the following: 1) robustness of the hybrid technique in finding suitable FLCs for different operating points with minimal user interaction, 2) robustness of the optimized FLC to operate at different operating conditions with no gain scheduling, and 3) the ability of the GA in finding a suitable FLC with as few as 10 rules in the rule base.

Several extensions to this study can be made. These include the following.

1) Examine the application to the nonlinear full envelope aircraft model.

2) Include a robustness measure in the GA optimization process. This could be achieved with a multiobjective optimization setup.

3) Include the number of rules as a parameter of the FLC system.

4) Examine other types of FLC membership function, such as trapezoidal, triangular, and sigmoidal.

All of the extensions proposed will not change the fundamental hybrid approach outlined in this paper. Also, the hybrid approach outlined in this paper could be easily modified and applied to other man–machine control problems.

Appendix: System Description

The nonlinear F/A-18 model is linearized for many different flight conditions. These equations are represented in the state space form as follows:

$$\dot{x} = Ax + Bu; \qquad x = [V_t\ \alpha\ \theta\ q]^T; \qquad u = \delta_s$$

The A and B matrices are as follows.

Operating point: 800 ft/s, 0-ft altitude:

$$A = \begin{bmatrix} -1.896e-2 & 2.620e+1 & -3.219e+1 & 0 \\ -1.027e-4 & -2.233 & 0 & 9.99e-1 \\ 0 & 0 & 0 & 1 \\ 1.1e-3 & -1.025e+1 & 0 & -5.193e-1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -2.953e-3 \\ 0 \\ -2.964e-1 \end{bmatrix}$$

Operating point: 800 ft/s, 20,000-ft altitude:

$$A = \begin{bmatrix} -1.1048e-2 & -1.2906e+1 & -3.2199e+1 & 0 \\ -1.1155e-4 & -1.198 & 0 & 9.995e-1 \\ 0 & 0 & 0 & 1 \\ 4.868e-4 & -3.63 & 0 & -3.0628e-1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -1.707e-3 \\ 0 \\ -1.69e-1 \end{bmatrix}$$

Operating point: 600 ft/s, 0-ft altitude:

$$A = \begin{bmatrix} -1.3566e-2 & -1.1206e+1 & -3.2199e+1 & 0 \\ -1.7885e-4 & -1.4815e-1 & 0 & 1.00099 \\ 0 & 0 & 0 & 1 \\ 0 & -4.1493 & 0 & -3.327e-1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -5.8723e-4 \\ 0 \\ -3.906e-2 \end{bmatrix}$$

Operating point: 600 ft/s, 40,000-ft altitude:

$$A = \begin{bmatrix} -1.675e-2 & -3.381e+1 & -3.2199e+1 & 0 \\ -1.92e-4 & -3.267e-1 & 0 & 9.99e-1 \\ 0 & 0 & 0 & 1 \\ -3.964e-4 & -1.439 & 0 & -1.1896e-1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -5.8723e-4 \\ 0 \\ -3.906e-2 \end{bmatrix}$$

Acknowledgments

This work was sponsored by the Naval Air Warfare Center, Warminster, PA, under Contract N62269-93-C-0434. The authors thank the Technical Monitor, Mark Steinberg, for his support and assistance.

References

[1]Karr, C. L., and Gentry, E. J., "Fuzzy Control of pH Using Genetic Algorithms," *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, 1993, pp. 46–53.

[2]Sugeno, M., Griffin, M. F., and Bastian, A., "Fuzzy Hierarchical Control of An Unmanned Helicopter," *Proceedings of the Fifth International Fuzzy Systems Association Congress*, Seoul, Republic of Korea, July 1993.

[3]Takagi, T., and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No.1, 1985, pp. 116–132.

[4]Karr, C., "Genetic Algorithms for Fuzzy Controllers," *AI Expert*, Feb. 1991, pp. 26–33.

[5]Freeman, L. M., KrishnaKumar, K., Karr, C. L., and Meredith, D., "Tuning Fuzzy Logic Controllers Using Genetic Algorithms—Aerospace Applications," *Proceedings of the AAAIC'90 Aerospace Applications of Artificial Intelligence Conference*, Netrologic, Inc., Dayton, OH, 1990, pp. 351–358.

[6]Lee, C., "Fuzzy Logic Controller—Part I," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, 1990, pp. 404–418.

[7]Lee, C., "Fuzzy Logic Controller—Part II," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, 1990, pp. 419–435.

[8]Holland, J., *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, MI, 1975.

[9]Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison–Wesley, Reading, MA, 1989.

[10]KrishnaKumar, K., and Goldberg, D. E., "Genetic Algorithms in Control System Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 735–740.

[11]Kristinsson, K., and Dumont, G. A., "System Identification and Control Using Genetic Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 5, 1992, pp. 1033–1046.

[12]Michalewicz, Z., and Krawczyk, J. B., "A Modified Genetic Algorithm for Optimal Control Problems," *Computers in Mathematical Applications*, Vol. 23, No. 12, 1992, pp. 83–94.

[13]Cooper, G. C., and Harper, R. P., Jr., "The Use of Pilot Ratings in the Evaluation of Aircraft Handling Qualities," NASA-TND-5153, April 1969.

[14]Kleinman, D. L., Baron, S., and Levison, W. H., "An Optimal Control Model of Human Response, Part 1: Theory and Validation," *Automatica*, Vol. 6, 1970, pp. 357–369.

[15]Hess, R. A., "Predictions of Pilot Opinion Ratings Using an Optimal Pilot Model," *Human Factors*, Vol. 19, No. 5, 1977, pp. 459–475.

[16]Hess, R. A., "A Method of Generating Numerical Pilot Opinion Ratings using the Optimal Pilot Model," NASA-TMX-73, 101, Feb. 1976.

[17]Levison, W. H., "A Model Based Technique for the Design of Flight Directors," *Proceedings of Ninth Annual Conference on Manual Control*, NASA-CR-142295, 1973.

[18]Hess, R. A., "Analytical Display Design for Flight Tasks Conducted under Instrument Meteorological Conditions," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-7, June 1977, pp. 453–462.

[19]Schmidt, D. K., "Optimal Flight Control Synthesis Via Pilot Modeling," *Journal of Guidance and Control*, Vol. 2, No. 4, 1979, pp. 308–312.

[20]Garg, S., and Schmidt, D. K., "Model-Based Analysis of Control/Display Interaction," *Journal of Guidance and Control*, Vol. 12, No. 3, 1987, pp. 342–350.

[21]Gonsalves, P. G., Kneller, E. W., and Zacharias, G. L., "Model-Based Method for Terrain-Following Display Design," Charles River Analytics, Inc., AAMRL-TR-89-039, Cambridge, MA, June 1989.

[22]Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, 1973, pp. 28–44.

[23]Sugeno, M. E., *Industrial Applications of Fuzzy Control*, North-Holland, Amsterdam, 1985.

[24]Tang, K. L., and Mulholland, R. J., "Comparing Fuzzy Logic with Classical Controller Designs," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 17, No. 6, 1987, pp. 1085–1087.

[25]Abdelnour, G. M., Chang, C. H., Huang, F. H., and Cheung, J. Y., "Design of a Fuzzy Controller Using Input and Output Mapping Factors," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 5, 1991, pp. 952–960.

[26]Zadeh, L. A., "A Theory of Approximate Reasoning, Univ. of California, Memo UCB/ERLM 77/58, Berkeley, CA, May 1977.

[27]Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1992, pp. 314–316.