WP1 - 5:15

1994 IEEE International Symposium on
Intelligent Control
16-18 August, 1994.
Columbus, Ohio, USA

Fuzzy Systems for Function Approximation with Applications to Failure Estimation[†]

Eric G. Laukonen and Kevin M. Passino [‡]
Department of Electrical Engineering, The Ohio State University
2015 Neil Avenue, Columbus, Ohio 43210

Abstract

A fuzzy system can be constructed to interpolate between input-output data to provide an approximation for the function that is implicitly defined by the input-output data pair associations. In this paper we begin by explaining how function approximation techniques can be used to solve nonlinear estimation and system identification problems. Next, we discuss several fundamental issues related to how to choose the input-output data pairs so that accurate function approximation can be achieved with fuzzy systems. Our main result is a new technique for function approximation via fuzzy systems where we specify membership functions and add rules to try to achieve a pre-specified function approximation accuracy. We illustrate the new technique on an actuator failure detection and identification problem for the F-16 aircraft.

I. Introduction

Fuzzy systems have been successfully applied in several areas within engineering including control, signal processing, and pattern recognition. Some recent work has focused on the idea of constructing fuzzy systems from a finite set of input-output training data in order to perform function approximation. This new focus is particularly important due to the fact that many problems in estimation and identification can be formulated as function approximation problems. For instance, in conventional system identification input-output data is gathered from a physical system and a least squares approach can be used to provide the best approximation for the linear function that maps the system inputs to its outputs. Similarly, in parameter estimation if one is given data that associates measurable system variables with an internal system parameter, a functional mapping may be constructed that approximates the process of estimation of the internal system parameter. In this paper we begin by precisely defining the function approximation problem and showing how nonlinear system identification and nonlinear estimation are special cases of this problem. Next we examine some theoretical issues associated with how to choose the input-output data so that good function approximation can be achieved. The main result of the paper provides a new technique for function approximation with fuzzy systems. In this technique we use ideas from the approaches in [1, 2, 3] and utilize a unique and novel way to position membership functions and add rules to a fuzzy system to try to achieve a pre-specified function approximation accuracy. While the approaches in [1, 2, 3] and related ones in [4, 5, 6, 7] have been used successfully for a variety of highly nonlinear identification and estimation problems, the problems studied to date have been of relatively low complexity. In this paper we evaluate our new function approximation approach by showing how it can be used to construct complex nonlinear estimators for aileron and differential elevator failures on an F-16 aircraft.

[†]This work was supported in part by an Ohio State University Interdisciplinary Seed Grant and by National Science Foundation Grant IRI 9210332.

[‡]Address all correspondence to Kevin Passino ((614)292-5716; email: passino@ee.eng.ohio-state.edu).

II. Background

Given some function $f : \mathcal{X} \subset I\!R^n \rightarrow \mathcal{Y} \subset I\!R$ where $\mathcal{Y}$ is a bounded set, we wish to construct a fuzzy system $g : X \subset \mathcal{X} \rightarrow Y \subset \mathcal{Y}$, where X and Y are some domain and range of interest, by choosing a parameter vector $\underline{\theta} \in \Theta$ so that

$$f(\underline{x}) = g(\underline{x}; \underline{\theta}) + e(\underline{x}) \tag{1}$$

for all $\underline{x} \in X$ where $e(\underline{x})$, the error in approximation, is as small as possible. We assume that all that is available to choose the parameters $\underline{\theta}$ of the fuzzy system $g(\underline{x}; \underline{\theta})$ is some part of the function f in the form of a finite set of input-output data pairs. The i^{th} input-output data pair for the system f is denoted by $(\underline{x}_i, y_i)$ where $\underline{x}_i \in X$, $y_i \in Y$ and $y_i = f(\underline{x}_i)$. We call the set of input-output data pairs the *training data set* and denote it by

$$F = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_{m_F}, y_{m_F})\} \subset X \times Y \tag{2}$$

where m_F denotes the number of I/O data pairs contained in F. Therefore, the problem we consider here is how to construct a fuzzy system $g(\underline{x}; \underline{\theta})$ so that $f(\underline{x}) \approx g(\underline{x}; \underline{\theta})$ for all $\underline{x} \in X$ when we have only limited information about f in the form of the training set F.

Many applications exist in the control and signal processing areas which may utilize nonlinear function approximation. One such application is system identification. System identification is the process of constructing a mathematical model of a dynamical system using experimental data from that system. Let f denote the physical system that we wish to identify. The training set F is defined by the experimental input-output data. In linear system identification an Autoregressive Moving Average (ARMA) model is often used where

$$y(k) = \sum_{i=1}^{q} \theta_{a_i} y(k-i) + \sum_{i=0}^{p} \theta_{b_i} u(k-i) \quad (3)$$

and $u(k)$ and $y(k)$ are the system input and output at time k. In this case $g(\underline{x};\underline{\theta})$, which is not a fuzzy system, is defined by (3) where

$$\begin{aligned} \underline{x} &= [y(k-1)\cdots y(k-q)u(k)\cdots u(k-p)]^T \quad (4) \\ \underline{\theta} &= [\theta_{a_1}\cdots\theta_{a_q}\,\theta_{b_0}\cdots\theta_{b_p}]^T. \quad (5) \end{aligned}$$

System identification amounts to adjusting $\underline{\theta}$ using information from F so that $g(\underline{x};\underline{\theta}) \approx f(\underline{x})$ for all $x \in X$. Clearly, restricting $g(\underline{x};\underline{\theta})$ to be linear may often make it difficult to achieve accurate identification (i.e. function approximation), especially if $f(\underline{x})$ is highly nonlinear.

Parameter estimation is the process of estimating the value of some system quantity based on measurable data from that system. In this case it is assumed that the parameter we wish to estimate is ξ where

$$\xi = f(\underline{x}) \quad (6)$$

Many techniques currently exist for parameter estimation which utilize model-based approaches. We utilize a parameterized fuzzy system $g(\underline{x};\underline{\theta})$ developed using a training set F to estimate the quantity of interest ξ. In this case $\underline{x}$ and F become,

$$\begin{aligned} \underline{x} &= [y(k)\cdots y(k-q)\; u(k)\cdots u(k-p)]^T \quad (7) \\ F &= \{(\underline{x}_1,\xi_1),\ldots,(\underline{x}_{m_F},\xi_{m_F})\}. \quad (8) \end{aligned}$$

Techniques which detect and isolate failures within complex systems in real time have been implemented in several applications [8, 9]. Although many of these model based techniques perform adequate fault detection in both failed and unfailed conditions if the real process acts as a known linear model and the necessary signal processing and decision logic is tractable, we consider the case where the model may be nonlinear and unknown. Therefore, we treat the fault estimation problem as a special case of parameter estimation where I/O data F (8) from the process is known and $f(\underline{x}_i)$ is the associated fault value. Therefore, the FDI problem we consider is equivalent to parameter estimation where a fuzzy system is constructed so that $\xi_i(k) = g(\underline{x},\underline{\theta})$ represents a fault value.

In this paper we will investigate the possibility of constructing a fuzzy system $g(\underline{x};\underline{\theta})$ by choosing $\underline{\theta}$ based on available training data F so that $e(k)$ is small for all k. Similar to conventional system identification we will utilize an appropriately defined "regression vector" $\underline{x}$ as specified in (4). Our hope is that since the fuzzy system $g(\underline{x};\underline{\theta})$ has more functional capabilities than the linear map defined in (3), we will be able to achieve more accurate identification for highly nonlinear systems by appropriate adjustment of its parameters $\underline{\theta}$.

A system which exhibits universal approximation is capable of approximating any real continuous function on a compact set to any arbitrary accuracy. Certain classes of fuzzy systems have the property of *universal approximation*. One common class of fuzzy systems we consider with singleton fuzzification, product inference, Gaussian membership functions, and centroid defuzzification is governed by the parameter set

$$\begin{aligned} \underline{\theta} = [&N \quad b_1 \;\cdots\; b_N \quad c_1^1 \;\cdots\; c_1^n \;\cdots\; c_N^1 \;\cdots \\ &c_N^n \quad \sigma_1^1 \;\cdots\; \sigma_1^n \;\cdots\; \sigma_N^1 \;\cdots\; \sigma_N^n]^T, \quad (9) \end{aligned}$$

and has,

$$g(\underline{x};\underline{\theta}) = \frac{\sum_{i=1}^{N} b_i \prod_{j=1}^{n} \exp(-\frac{1}{2}(\frac{x_i - c_i^j}{\sigma_i^j})^2)}{\sum_{i=1}^{N} \prod_{j=1}^{n} \exp(-\frac{1}{2}(\frac{x_i - c_i^j}{\sigma_i^j})^2)} \quad (10)$$

(see [10] for an interpretation of (10) and a fuller explanation of fuzzy systems). The size of $\underline{\theta}$ is governed by the number of fuzzy rules N and the number of inputs to the fuzzy system n. For the fuzzy system (10) the output membership function for the i^{th} rule is represented by the scalar point b_i (a "singleton"), the input membership function for the i^{th} rule and j^{th} input is a Gaussian type input membership function with a point of maximum c_i^j and a relative width term $\sigma_i^j > 0$. This form for a fuzzy system has the property of universal approximation [11], is continuously differentiable, and has nonzero input membership values over the domain of interest X.

III. Technique for Function Approximation with Fuzzy Systems

In this section we utilize some of the insight from the analysis performed in the previous section to propose a simple method for constructing a fuzzy system for nonlinear function approximation based on a training data set F defined by (2). The method we propose is an extension of other techniques proposed for nonlinear function approximation via fuzzy systems. Namely, we utilize a technique given in [3] entitled "learning by examples" and some ideas from [2] and [1] to develop a new method to construct a fuzzy system for nonlinear function approximation. We first develop the methodology and then present the function approximation algorithm.

A. *Function Approximation Algorithm*

Given the parameterized fuzzy system (10) we choose the parameters in $\underline{\theta}$ and the number of rules N so that we approximate a nonlinear function $f : X \subset \mathbb{R}^n \rightarrow Y \subset \mathbb{R}$ based on input-output data pairs generated via experiment or simulation.

The first question we address in the construction of a fuzzy system is the choice of the number of fuzzy rules N. More fuzzy rules means increased computational complexity in implementation. Therefore specific applications may limit the number of rules which may be utilized in implementation. On the other hand, the use of many fuzzy rules may produce a fuzzy system with more functional capability available for approximation. Therefore, a tradeoff exists between computational complexity and functional capability based on the number of fuzzy rules N.

We first define the quantity ϵ_f which characterizes the desired accuracy with which the fuzzy system performs function approximation. Specifically, the output of the current fuzzy system is compared to the output portion of the input-output training data point y_i. If this comparison is greater then ϵ_f then we choose to augment the current fuzzy system with an additional rule so that:

$$\begin{aligned} |g(\underline{x}_i;\underline{\theta}) - y_i| > \epsilon_f \quad &\Rightarrow \quad \text{modify} \\ |g(\underline{x}_i;\underline{\theta}) - y_i| \leq \epsilon_f \quad &\Rightarrow \quad \text{do not modify.} \end{aligned} \tag{11}$$

The choice for ϵ_f determines the number of fuzzy rules where a smaller value for ϵ_f generally means more fuzzy rules for a given training data set, and vice versa. Employment of ϵ_f in the function approximation algorithm is outlined below.

Algorithm To Construct Fuzzy System

1. Construct An Initial Fuzzy System
We assume that we have input-output training data pairs given by $(\underline{x}_i, y_i) \in F$ where $i = 1, \ldots, M$. We also choose an initial value for the input membership associated with the first fuzzy rule given by $\sigma_1^j = \sigma_0$ for all $j = 1, \ldots, n$. The initial choice σ_0 often has little effect on the resulting fuzzy system as training progresses, but it is necessary to initialize the fuzzy system. The first step in the algorithm is to form a fuzzy rule according to the first input-output training data pair $(\underline{x}_1, y_1)$ and the initial fuzzy system by
$$g(\underline{x};\underline{\theta}) = \frac{b_1 \prod_{j=1}^n \exp[-(\frac{x_j - c_1^j}{\sigma_1^j})^2]}{\prod_{j=1}^n \exp[-(\frac{x_j - c_1^j}{\sigma_1^j})^2]}. \tag{12}$$
where $c_1^j = x_1^j$, $b_1 = y_1$, and $\sigma_1^j = \sigma_0$ for $j = 1, \ldots, n$. Once the initial fuzzy system is constructed, the function approximation algorithm forms additional fuzzy rules and chooses parameters θ for the fuzzy system based on the remaining input-output training data pairs and the desired approximation accuracy ϵ_f in Steps 2 and 3.
2. Evaluate New Training Data Point
For each additional training data point $(\underline{x}_i, y_i) \in F$, for $i = 2, \ldots, M$ we evaluate (11) with the current fuzzy system. If an additional rule is necessary then we go to Step 3, otherwise we return to the start of to Step 2 to evaluate the next training data point according to (11).
3. Augment Current Fuzzy System
We modify some of the parameters of the original fuzzy system, namely σ_i^j, to account for the new information contained in $(\underline{x}_i, y_i)$ that passed the test in (11). A new rule is added to the fuzzy system so that the new fuzzy parameter set $\underline{\theta}$ is given by
$$\begin{aligned} N &:= N + 1 && (13) \\ b_N &:= y_i && (14) \\ c_N^j &:= x_i^j, \quad j = 1, \ldots, n && (15) \\ \sigma_N^j &:= \sigma_0, \quad j = 1, \ldots, n && (16) \end{aligned}$$
Moreover, we modify σ_i^j for $i = 1, \ldots, N$, $j = 1, \ldots, n$ to adjust spacing between the membership function for the rules in the augmented fuzzy system so that (i) the added rule does not distort what has already been learned, and (ii) we achieve a smooth interpolation between training points. Modification of σ_i^j is carried out by determining for each rule a nearest neighbor. We modify σ_i^j for the i^{th} rule according to the computed nearest neighbor which we denote by the index i^*
$$i^* = \arg\min\{||\underline{c}_{i'} - \underline{c}_i||_2 : i' = 1, \ldots, N, i' \neq i\} \tag{17}$$
where $\underline{c}_i = [c_i^1, \cdots, c_i^n]^T$. We update $\underline{\sigma}_i = [\sigma_i^1, \cdots, \sigma_i^n]^T$ according to the equation ($j = 1, \ldots, n$):
$$\underline{\sigma}_i^j = \frac{1}{W}|\underline{c}_i^j - \underline{c}_{i^*}^j| \tag{18}$$
where W is a weighting term which governs the input membership function overlap between nearest neighbor rules. A larger W means less overlap of membership functions and vice versa. This is done for all rules $i = 1, \ldots, N$. Once Step 3 is complete we go to Step 2 for the next training data point until the number of training data points are exhausted.

IV. Function Approximation Example: F-16 Aircraft

In this example, the function approximation algorithm is used to construct an estimator that can detect and identify actuator failures by using measurable F-16 aircraft data. The presence of an actuator failure is detected by comparing the actuator's commanded position and the estimated position. A failure is detected if the commanded position and the estimated position differ by more than some threshold for some window of time.

A. F-16 Aircraft Model and Failure Modes

The F-16 aircraft model used in this example is based on a set of five linear perturbation models (that are extracted from a non-linear model [1] at the five operating conditions) (A_i, B_i, C_i, D_i), $i \in \{1, 2, 3, 4, 5\}$:

$$\begin{aligned} \dot{x}_{f16} &= A_i x_{f16} + B_i u \\ y &= C_i x_{f16} + D_i u \end{aligned} \tag{19}$$

where the variables are defined as follows:

- Inputs $u = [\delta_e \quad \delta_{de} \quad \delta_a \quad \delta_r]^T$:
  - δ_e = elevator deflection (degrees)
  - δ_{de} = differential elevator deflection (degrees)
  - δ_a = aileron deflection (degrees)
  - δ_r = rudder deflection (degrees)
- System State $x_{f16} = [\alpha \quad q \quad \phi \quad \beta \quad p \quad r]^T$:
  - α = angle of attack (degrees)
  - q = body axis pitch rate (degrees/sec)
  - ϕ = Euler roll angle (degrees)
  - β = sideslip angle (degrees)
  - p = body axis roll rate (degrees/sec)
  - r = body axis yaw rate (degrees/sec)
- Outputs $y = [x_{f16}^T \quad A_z]^T$:
  - A_z = normal acceleration (g's)
- System Matrices (A_i, B_i, C_i, D_i): Provided by Wright Laboratories

The nominal control laws for the F-16 aircraft that were provided by the Wright Laboratories consist of two parts, one for the longitudinal channel and the other for the lateral channel. The inputs to the controller are the pilot commands and the F-16 system feedback signals. For the longitudinal channel, the pilot command is the desired pitch A_{zd}, and the system feedback signals are normal acceleration A_z, angle of attack α, and pitch rate q. Likewise, for the lateral channel, the pilot commands are the desired roll rate p_d as well as the desired yaw β_d, and the system feedback signals are the roll rate p, yaw angle r, and sideslip β. The controller gains for the longitudinal and for the lateral channels are scheduled as a function of different dynamic pressures. The dynamic pressure at all five perturbation models is fixed at 499.24 psf, which is based on an assumption that the F-16 aircraft will operate with constant speed and altitude. Hence, a gain schedule table, which is provided by Wright Laboratories, is used to determine the controller gains.

The transfer function $\frac{20}{s+20}$ is used to represent the actuator dynamics for each of the aircraft control surfaces, and the actuators have physical saturation limits so that: $-21° \le \delta_e \le 21°$, $-21° \le \delta_{de} \le 21°$, $-23° \le \delta_a \le 20°$, and $-30° \le \delta_r \le 30°$. The actuator rate saturation is $\pm 60°/s$ for all the actuators. To simulate the closed-loop system we interpolate between the five perturbation models based on the value of α. For all the simulations, a special "loaded roll command sequence", provided by Wright Laboratories, is used. For this command sequence: at time $t = 0.0$, a $60°/s$ roll rate command (p_d) is held for 1 second; at time $t = 1.0$, a 3g pitch command (A_{zd}) is held for 9 seconds; at time $t = 4.5$, a $-60°/s$ roll rate command (p_d) is held for 1.8 seconds; and at time $t = 11.5$, a $60°/s$ roll rate command (p_d) command is held for 1 second. The sideslip command β_d is held at zero throughout the sequence.

While many different failures can occur on a high performance aircraft such as the F-16 (e.g., performance degradation or structural damage), in this study we will focus on FDI for aileron and differential elevator stuck failures.

B. Failure Models and Training

In order to implement a fuzzy estimator for FDI on the F-16 aircraft it is necessary to determine those signals from the aircraft which enable us to deduce the position of the aileron and differential elevator. Specifically, we need to first determine the inputs $\underline{x}_i$ $(i = 1, \ldots, n)$ so that a fuzzy system (10) may produce accurate estimates ($\hat{\delta}_a$ and $\hat{\delta}_{de}$) for the aileron and differential elevator positions (δ_a and δ_{de}). We do this by examining the structure of the F-16 aircraft model and understanding the effect of the controller on the actuator positions. Specifically, we recognize that if a failure in the aileron actuator occurs, the controller compensates via the differential elevator actuator and vice versa. We also recognize that the roll command input p_d affects both the aileron and differential elevator positions. Therefore we utilize these signals as inputs to the fuzzy estimator to deduce the position of the aileron and differential elevator. In particular, we sample the aircraft responses and choose

$$\underline{x} = [\delta_a^c, \delta_{de}^c, p_d]^T, \tag{20}$$

where δ_a^c and δ_{de}^c are the signals commanded by the controller. The vector $\underline{x}$ $(n = 3)$ is used in the estimation of both the aileron position and the position of the differential elevator. Next, we let f_a and f_{de} denote the unknown functional mapping between the aircraft variables and the estimates of the actuator positions so that we let

$$\hat{\delta}_a(k) = f_a(\underline{x}), \tag{21}$$

[1] All information about the F-16 aircraft models was provided by Wright Laboratories.

and

$$\hat{\delta}_{de}(k) = f_{de}(\underline{x}). \tag{22}$$

Note that the choice of the inputs in (20) intuitively represents signals which may contribute to the functional mapping f_a or f_{de} in order to construct the fuzzy estimator. For instance, during no fault operation the aileron and differential elevator positions are related linearly to the roll command p_d. Basically, the estimator construction problem involves training the fuzzy system to approximate f_a and another to approximate f_{de}. With this we will have one fuzzy system to estimate the position of the aileron and one to estimate the position of the differential elevator.

We train the fuzzy systems for estimation by obtaining input-output training data developed through simulation with a sampling time of $T = 0.02$ sec. We alternately fail the aileron and differential elevator actuators at various positions (no failure, 1 sec., 3 sec., 5 sec., 7 sec.) and collect sampled data for $\underline{x}$. We then construct the fuzzy systems for estimation by means of the training algorithm which chooses values for N, b_i, c_i^j, and σ_i^j to achieve an accuracy of ± 1 degree on both the aileron and the differential elevator estimators during training (i.e. $\epsilon_f = 1$). The width scaling term W is 3. The resulting fuzzy system for estimation of aileron position contains 108 rules, and the resulting fuzzy system for estimation of differential elevator position contains 224 rules.

C. *Results*

The fuzzy estimator provides a simple FDI system to detect and isolate failures for the aileron and differential elevator actuators. We simply form the absolute value of the difference between the estimated value with the value commanded for the actuator. If one differs above some threshold for some period of time we may guess that a failure has occurred and in this situation one may want to reconfigure the control laws (see [12]). In our case we evaluate the median of the residual over 21 samples. If the resulting median exceeds a threshold of 3 degrees for the aileron and 1 degree for the differential elevator, then we flag the system that a specific actuator failure has occurred (we choose these thresholds via a simulation-based analysis of the no fault residuals). Once a residual exceeds a threshold the failure flag is set for the entire test. The results are given in Figures 1-3. Notice that in Figure 1 if there is no actuator failure then the failure estimators for both the aileron and differential elevator provide an estimate of the actuator positions that is reasonably close to the actual position. Clearly if we choose a detection threshold of 3 degrees for the aileron residual and a detection threshold of 1 degree for the differential elevator no failure will be indicated. Using the median filter described above, failures are correctly detected and determined for an aileron failure at 1 second (Figure 2) into the loaded roll command sequence with the filtered aileron residuals exceeding the threshold and the filtered differential elevator residual remaining below the threshold. Moreover, failures are correctly detected and determined for a differential elevator failure at 1 second into the loaded roll command sequence with the filtered differential elevator residuals exceeding the 1 degree threshold and the filtered aileron residuals remaining below the threshold (see Figure 3). We see that the estimators provide an adequate estimate of the position at which the aileron and differential elevator are stuck and also properly indicate which actuator has not failed. Our simulation results actually show that the fuzzy estimator can accurately discriminate between and aileron and differential elevator failures for the training data set we utilize. If failures occur other than the ones we have trained for, or if a different aircraft maneuver is initiated, accurate estimation may not be achieved.

V. Conclusions

In this paper we (*i*) concisely stated the function approximation problem in the context of fuzzy systems where input-output training data was available, (*ii*) illustrated possible applications for nonlinear function approximation in the system identification and signal processing fields, (*iii*) discussed the functional capability of fuzzy systems in terms of the universal approximation property and discussed contraints associated with the input-output training data set. In addition, we introduced a new method to perform function approximation that builds on the approaches in [1, 3] and uses some ideas from [2]. Finally, we used the function approximation algorithm to design failure estimators for one class of failures in the ailerons and differential elevator on an F-16 aircraft.

References

[1] A. T. Magan, "Fuzzy parameter estimation for failure identification," Master's thesis, Dept. of Electrical Engineering, Ohio State University, 1992.

[2] L.-X. Wang, "Training of fuzzy logic systems sing nearest neighborhood clustering," in *2nd IEEE Conference on Fuzzy Systems, San Fransico, California*, pp. 13-17, 1993.

[3] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from example," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, pp. 1414-1427, November 1992.

[4] H. Tanaka and H. Ishibuchi, "Identification of possibilistic linear systems by quadratic membership functions of fuzzy parameters," *Fuzzy Sets and Systems*, vol. 41, pp. 145-160, 1991.

[5] L.-X. Wang and J. M. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifiers," in *1st IEEE Conference on Fuzzy Fystems, San Diego, California*, pp. 1409-1418, 1992.

[6] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation and orthogonal least-squares learning," *IEEE Trans. on Neural Networks*, vol. 3, pp. 1–8, september 1992.

[7] S.-K. Sin and R. J. Defigueiredo, "Fuzzy system design through fuzzy clustering and optimal predefuzzification," in *2nd IEEE Conference on Fuzzy Systems, San Fransico, California*, pp. 190–195, 1993.

[8] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, vol. 3, no. 6, pp. 3–11, 1988.

[9] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy-a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[10] L. X. Wang, *Adaptive Fuzzy Systems and Control.* Prentice Hall, New Jersey, 1994.

[11] L.-X. Wang, "Fuzzy systems are universal approximators," in *1st IEEE conference on fuzzy systems*, pp. 1163–1170, March 1992.

[12] W. A. Kwong, K. M. Passino, and S. Yurkovich, "Fuzzy learning systems for aircraft control law reconfiguration." IEEE Int. Symposium on Intelligent Control, Columbus, Ohio, Aug., 1994.

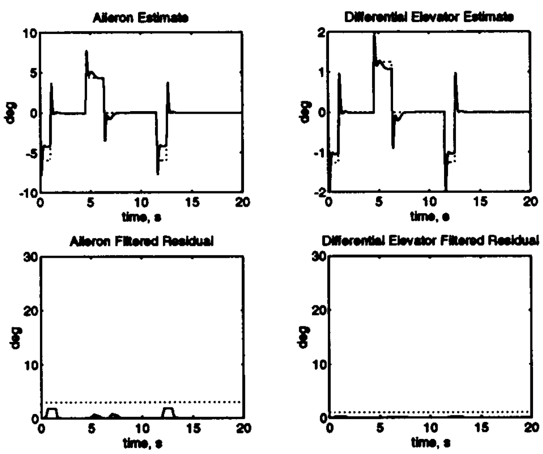

Figure 1: No Failure Case (solid line - estimate and residual, dotted line true value and threshold)

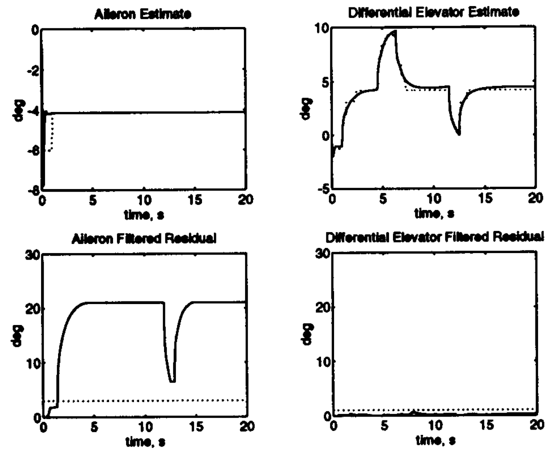

Figure 2: Aileron Failure at t=1 sec. (solid line - estimate and residual, dotted line true value and threshold)

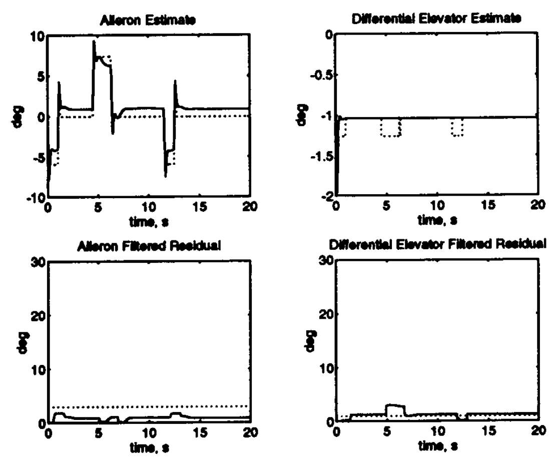

Figure 3: Differential Elevator Failure at t=1 sec. (solid line - estimate and residual, dotted line true value and threshold)