Brief Papers

Fault Detection and Isolation for an Experimental Internal Combustion Engine via Fuzzy Identification

E. G. Laukonen, K. M. Passino, V. Krishnaswami, G.-C. Luh, and G. Rizzoni

Abstract—**Certain engine faults can be detected and isolated by examining the pattern of deviations of engine signals from their nominal unfailed values. In this brief paper, we show how to construct a fuzzy identifier to estimate the engine signals necessary to calculate the deviation from nominal engine behavior, so that we may determine if the engine has certain actuator and sensor "calibration faults." We compare the fuzzy identifier to a nonlinear ARMAX technique and provide experimental results showing the effectiveness of our fuzzy identification based failure detection and identification strategy.**

I. Introduction

In recent years, more attention has been given to reducing exhaust gas emissions produced by internal combustion engines. In addition to overall engine and emission system design, correct or fault-free engine operation is a major factor determining the amount of exhaust gas emissions produced in internal combustion engines. Hence, there has been a recent focus on the development of on-board diagnostic systems which monitor relative engine health [1], [2]. Although on-board vehicle diagnostics can often provide detection and isolation of some major engine faults, due to widely varying driving environments they may be unable to detect minor faults which may affect engine performance. Minor engine faults warrant special attention because these faults do not noticeably hinder engine performance but may increase exhaust gas emissions for a long period of time without the problem being corrected. The minor faults we consider in this paper include "calibration faults" (for our study, the occurrence of a calibration fault means that a sensed or commanded signal is multiplied by a gain factor not equal to one, and in the no fault case the sensed or commanded signal is multiplied by one) in the throttle and mass fuel actuators, in the engine speed, and mass air sensors (we can also consider "bias" type faults even though we do not do so in this paper). The reliability of these actuators and sensors is particularly important to the engine controller, since their failure can affect the performance of the emissions control system. Next, we overview the experimental engine testbed and testing conditions that we use for developing failure detection and isolation (FDI) systems. Following this, we briefly overview the relevant literature and contents of this paper.

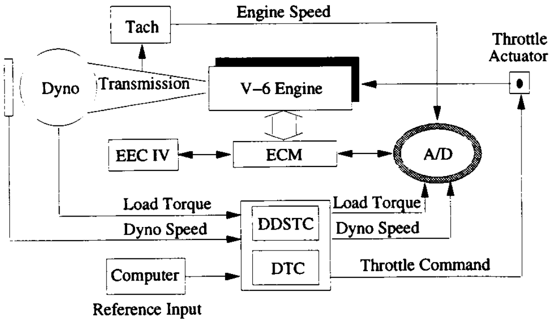


Fig. 1. The experimental engine test cell.

A. Experimental Engine and Testing Conditions

All investigations into the development of (FDI) strategies in this paper were performed utilizing the experimental engine test cell shown in Fig. 1 that is located within The Department of Mechanical Engineering at The Ohio State University [3], [4]. The experimental setup in the engine test cell consists of a Ford 3.0 L V-6 engine coupled to an electric dynamometer through an automatic transmission. An air charge temperature sensor (ACT), a throttle position sensor (TPS), and a mass air flow sensor (MAF) are installed in the engine to measure the air charge temperature, throttle position, and air mass flow rate. Two heated exhaust gas oxygen sensors (HEGO) are located in the exhaust pipes upstream of the catalytic converter. The resultant airflow information and input from the various engine sensors are used to compute the required fuel flow rate necessary to maintain a prescribed air/fuel ratio for the given engine operation. The CPU (EEC-IV) determines the needed injector pulse width and spark timing and outputs a command to the injector to meter the exact quantity of fuel. An electronic control module (ECM) break-out box is used to provide external connections to the EEC-IV controller and the

Manuscript received May 29, 1994; revised January 19, 1995. Recommended by Associate Editor H. P. Geering. This work was supported in part by an Ohio State University Interdisciplinary Seed Grant and by the National Science Foundation under Grants IRI 9210332 and DDM 9157211.

E. Laukonen and K. Passino are with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA.

V. Krishnaswami, G.-C. Luh, and G. Rizzoni are with the Department of Mechanical Engineering, The Ohio State University, Columbus, OH 43210 USA.

IEEE Log Number 9412863.

data acquisition system. The angular velocity sensor system consists of a digital magnetic zero-speed sensor and a specially designed frequency-to-voltage converter [3], [4] which converts frequency signals proportional to the rotational speed into an analog voltage. Data is sampled at every engine revolution. A variable load is produced through the dynamometer which is controlled by a DYN-LOC IV speed/torque controller in conjunction with a DTC-1 throttle controller installed by DyneSystems Company. The load torque and dynamometer speed are obtained through a load cell and a tachometer, respectively. The throttle and the dynamometer load reference inputs are generated through a computer program and sent through a RS-232 serial communication line to the controller. Physical quantities of interest are digitized and acquired utilizing a National Instruments AT-MIO-16F-5 A/D timing board for a personal computer.

Due to government mandates, periodic inspections and maintenance for engines are becoming more common. One such test developed by the Environmental Protection Agency (EPA) is the Inspection and Maintenance (IM) 240 cycle. The EPA/IM240 cycle (see Fig. 2) represents a driving scenario developed for the purpose of testing compliance of vehicle emissions systems for contents of carbon monoxide (CO), unburned hydrocarbons (HC), and nitrogen oxides (NO_x). The IM240 cycle is designed to be performed under laboratory conditions with a chassis dynamometer and is patterned after the Urban Dynamometer Driving Schedule (UDDS) which approximates a portion of a morning commute within an urban area. This test is designed to evaluate the emissions of a vehicle under real world conditions. In [5], Rizzoni and Krishnaswami have proposed an additional diagnostic test to be performed during the IM240 cycle to detect and isolate a class of minor engine faults which may hinder vehicle performance and increase the level of exhaust emissions. Since the EPA proposes to make the test mandatory for all vehicles, performing an additional diagnostic analysis in parallel would provide a controlled test which may allow for some minor faults to be detected and corrected thus reducing overall exhaust emissions in a large number of vehicles. We also use the IM240 cycle in this paper and take the same basic approach to FDI for minor engine faults as in [5], except that we utilize a fuzzy identification approach rather than the nonlinear ARMAX approach as in [5]. Related work on the use of nonlinear ARMAX is given in [4].

B. Relevant Literature and Paper Overview

In this paper, we investigate the use of fuzzy systems for use in FDI for the internal combustion engine described in the previous section. Specifically, we use a fuzzy clustering and optimal output predefuzzification technique outlined in [6] to model the dynamics of the experimental internal combustion engine and utilize these models together with a fault detection and isolation strategy given in [5] to identify a class of faults. Fuzzy systems have already 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 to perform system identification (e.g., the learning from examples, backpropagation, and clustering techniques in [7]). This is, however, what seems to be the first application of a fuzzy identification technique to FDI in an experimental setting. Moreover, since we provide a comparative analysis with a conventional nonlinear identification technique, we help to provide a more realistic assessment of the capabilities of the fuzzy methodology.

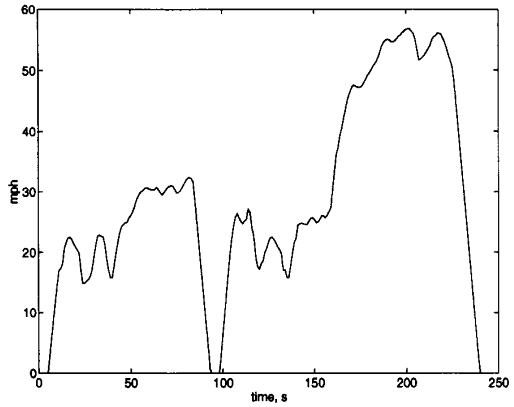


Fig. 2. The EPA IM240 engine cycle.

Following [6], in Section II we introduce the Takagi–Sugeno fuzzy system model [8], outline a fuzzy clustering technique called "Fuzzy c-Means" [9], and show how it can be applied to form the premise portion of the rules in a fuzzy system. We also explain a procedure for optimal output predefuzzification from [6] in which the output portion of the training data set, together with the cluster information, is used to perform a weighted least-squares approximation to construct appropriate output functions to represent the consequent portion of the rules in a fuzzy system. We apply the technique to perform modeling of various physical quantities associated with the emissions system for the experimental internal combustion engine described above. In Section IV we use the subsequent fuzzy system and a fault decoupling strategy given in [5] to implement a fault detection and isolation strategy for a class of calibration faults on the sensors and actuators ("input–output faults") which may occur in an internal combustion engine during the standard EPA/IM240 engine cycle. Analysis and discussion is included, and comparisons to a Nonlinear ARMAX technique from [10] are given.

II. Theoretical Background

In system identification, which forms the basis for our FDI technique, we wish to construct a model of a dynamical system using input–output data from the system [11]. Suppose that a system has input $\underline{x} = [x_1, \cdots, x_n]^T$ (T denotes transpose), output y, and that we can gather a set of m_F input–output training data pairs in the set

$$F = \{(\underline{x}_i, y_i) : i = 1, \cdots, m_F\} \tag{1}$$

where y_i is the output generated by the system at time k when the input is $\underline{x}_i$ at time k (hence x_1 is an input, and x_1^1 is a specified first component of the input portion of the first input–output training data pair). Conventional system identification techniques (such as least squares) are used to fit the best possible linear model to the data F so that i) it faithfully generates y_i when $\underline{x}_i$ is input to the system and $(\underline{x}_i, y_i) \in F$, and ii) properly interpolates when an $\underline{x}_i'$ is input to the system where $(\underline{x}_i', y_i) \notin F$. If input–output data from F truly represents a linear mapping, then conventional techniques often perform very well; however, if F represents a nonlinear mapping, the identification problem becomes more challenging.

The types of engine faults which the FDI strategy is designed to detect include calibration faults given in Table I. These faults directly affect the resulting fuel/air ratio and spark timing in combustion which subsequently effects the level of exhaust gas emissions. The fault detection and isolation strategy relies on estimates of ω (engine speed, rev/min), m_a (mass rate of air entering the intake manifold, lbm/sec), α (actuated throttle angle, expressed as a percentage of a full scale opening), m_f (mass of fuel entering the combustion chamber, lbm), and T_L (the load torque on the engine, lb-ft) (which we denote by $\hat{\omega}, \hat{m}_a$, $\hat{\alpha}$, $\hat{m}_f$, and $\hat{T}_L$, respectively) by identifying models f_ω, f_{m_a}, f_α, f_{m_f}, and f_{T_L} of how the engine operates. In particular, we have

$$\hat{\omega} = f_\omega(\underline{x}_\omega) \tag{2}$$

$$\hat{m}_a = f_{m_a}(\underline{x}_{m_a}) \tag{3}$$

$$\hat{\alpha} = f_\alpha(\underline{x}_\alpha) \tag{4}$$

$$\hat{m}_f = f_{m_f}(\underline{x}_{m_f}) \tag{5}$$

$$\hat{T}_L = f_{T_L}(\underline{x}_{T_L}) \tag{6}$$

where the inputs are given in (7)–(11) (k is a discrete-time in the crankshaft domain where physical quantities are sampled every turn of the engine crankshaft)

$$\begin{aligned}\underline{x}_\omega = [&\hat{\omega}(k-1), \hat{\omega}(k-2), \hat{\omega}(k-3), \alpha(k-1),\\ &\alpha(k-2), \alpha(k-3), m_f(k-1), m_f(k-2),\\ &m_f(k-3), \hat{T}_L(k-2)]^T\end{aligned} \tag{7}$$

$$\begin{aligned}\underline{x}_{m_a} = [&\hat{m}_a(k-1), \hat{m}_a(k-2), \hat{m}_a(k-3), \alpha(k-1),\\ &\alpha(k-2), m_f(k-1), m_f(k-2), m_f(k-3),\\ &\hat{T}_L(k-1), \hat{T}_L(k-3)]^T\end{aligned} \tag{8}$$

$$\begin{aligned}\underline{x}_\alpha = [&\hat{\alpha}(k-1), \hat{\alpha}(k-2), \hat{\alpha}(k-3), m_a(k-1),\\ &m_a(k-2), m_a(k-3), \omega_{\text{dy}}(k-1), \omega_{\text{dy}}(k-2)]^T\end{aligned} \tag{9}$$

$$\begin{aligned}\underline{x}_{m_f} = [&\hat{m}_f(k-1), \hat{m}_f(k-2), \hat{m}_f(k-3), m_a(k-1),\\ &m_a(k-2), m_a(k-3), \omega(k-1), \omega(k-2),\\ &\omega(k-3)]^T\end{aligned} \tag{10}$$

$$\begin{aligned}\underline{x}_{T_L} = [&\hat{T}_L(k-1), \hat{T}_L(k-2), \hat{T}_L(k-3), \alpha(k-1),\\ &\alpha(k-2), m_f(k-1), m_a(k-1), m_a(k-3),\\ &\omega_{\text{dy}}(k-1), \omega_{\text{dy}}(k-3)]^T\end{aligned} \tag{11}$$

where ω_{dy} is an output of the dynamometer. These regression vectors were chosen using simulation and experimental studies to determine which variables are useful in estimating others and how many delayed values must be used to get accurate estimation.

TABLE I
Types of Faults Detectable with FDI Strategy

Fault Location	Type of Fault	Description
m_a	sensor calibration	Measures amount of air intake for combustion
ω	sensor calibration	Measures engine speed
α	actuator calibration	Actuates the throttle angle
m_f	actuator calibration	Actuates the amount of fuel for combustion

One approach to nonlinear system identification that has been found to be particularly useful for our experimental testbed [10], [12] and which we will employ in the current study is the NARMAX (Nonlinear AutoRegressive Moving Average model with exogenous inputs) method which is an extension of the linear ARMAX system identification technique. The general model structure for NARMAX uses scaled polynomial combinations of the arguments contained in the regression vector; here we use the NARMAX model structure given by

$$\hat{y}(k) = \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij} x_i x_j \tag{12}$$

where $\beta_i, \beta_{ij} \in I\!R^n$ are parameters to be adjusted so that $\hat{y}(k)$ is as close as possible to $y(k)$ for all $\underline{x} \in \mathbb{R}^n$ (i.e., we only use one second-order polynomial term in our model structure). As is usual, in this paper we will use the standard batch least-squares approach to adjust the β_i, β_{ij} since they enter linearly (see [11] for details).

The primary focus of this paper, however, is to evaluate the use of a fuzzy identification technique as the basis for our FDI system. The particular fuzzy identification technique we study employs the Takagi–Sugeno model for a fuzzy system [8] as the model structure for identification and a clustering technique [9] combined with "optimal output predefuzzification" [6] to adjust the fuzzy system parameters, so that it generates a good estimate $\hat{y}$ for all $\underline{x} \in \mathbb{R}^n$.

Takagi and Sugeno's fuzzy system is composed of fuzzy IF-THEN rules of the form

$$\text{If } H^j \text{ Then } \hat{y}^j = c_0^j + c_1^j x_1 + \cdots + c_n^j x_n \tag{13}$$

$j = 1, \cdots, N$ (N is the number of rules) where input fuzzy set $H^j = \{(\underline{x}, \mu_{H^j}(\underline{x})) : \underline{x} \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_n\}$, $\mathcal{U}_i$ is the ith input universe of discourse, $\underline{x} = [x_1, \cdots, x_n]^T$ is the vector of fuzzy system inputs, $\mu_{H^j}(\underline{x})$ is the membership function for H^j, $\hat{y}^j = \underline{c}_j^T \underline{\tilde{x}}$ is a crisp output that is a linear function of the inputs given by the real-valued parameters $\underline{c}_j = [c_0^j, \cdots, c_n^j]^T$, and $\underline{\tilde{x}} = [1, \underline{x}^T]^T$. The output of the Takagi–Sugeno fuzzy system is a weighted average of $\hat{y}^j$ for $j = 1, \cdots, N$ given by

$$\hat{y} = g(\underline{x}; \underline{\theta}) = \frac{\sum_{j=1}^{N} \mu_{H^j}(\underline{x}) \hat{y}^j}{\sum_{j=1}^{N} \mu_{H^j}(\underline{x})} \tag{14}$$

where $\underline{\theta}$ is a vector of parameters which characterize the fuzzy system (given later).

In [6], Sin and Difigueiredo develop an algorithm to construct a Takagi–Sugeno fuzzy system that approximates the functional mapping represented in the input–output data F.

The algorithm contains two distinct steps for constructing a fuzzy system: first, fuzzy clustering is used to identify N cluster centers $\underline{v}_j \in \mathbb{R}^n, j = 1, \cdots, N$ where each cluster center represents the center of an input space fuzzy partition (i.e., the cluster centers are chosen as the center points for input membership functions), second, the consequent portion of each fuzzy rule is determined by calculating the coefficients $\underline{c}_j$ for N linear functions via a weighted least-squares computation that uses the output portion of the training data set.

A. Fuzzy Clustering

To specify the premise portion of the rules, we use the Fuzzy c-Means clustering algorithm [9], [13], [14] which employs a quantitative measure of similarity within data to determine the clustering partitions and degrees of membership associated with each partition. In our case, the similarity measure we use is a distance measure between points given by the input portion $\underline{x}$ of the training data in F. For Fuzzy c-Means we wish to minimize the objective function given by

$$J = \sum_{i=1}^{m_F} \sum_{j=1}^{N} (\mu_{ij})^m (\underline{x}_i - \underline{v}_j)^T (\underline{x}_i - \underline{v}_j) \tag{15}$$

where $m > 1$ is a design parameter, N is the number of clusters (and the number of rules), m_F is the number of training data points, $\underline{x}_i$ for $i = 1, \cdots, m_F$ is the input portion of the training data set $F, \underline{v}_j$ for $j = 1, \cdots, N$ are the "prototypes" (cluster centers), and μ_{ij} is the grade of membership for $\underline{x}_i$ in the jth cluster. In [9], Bezdek gives a set of necessary conditions for μ_{ij} and $\underline{v}_j$ to minimize (15) when $||\underline{x}_i - \underline{v}_j||_2^2 > 0$ for all j. This is given by

$$\underline{v}_j = \sum_{i=1}^{m_F} (\mu_{ij})^m \underline{x}_i \Big/ \sum_{i=1}^{m_F} (\mu_{ij})^m \tag{16}$$

for each $j = 1, \cdots, N$, and the scalars

$$\mu_{ij} = \left[\sum_{k=1}^{N} \left(\frac{||\underline{x}_i - \underline{v}_j||_2^2}{||\underline{x}_i - \underline{v}_k||_2^2} \right)^{m-1} \right]^{-1} \tag{17}$$

for each $i = 1, \cdots, m_F$ and $j = 1, \cdots, N$ ($||z||_2 = (z^T z)^{1/2}$). If, however, $||\underline{x}_i - \underline{v}_j||_2^2 = 0$ for some j, then μ_{ij} are any nonnegative numbers satisfying $\sum_{j=1}^{N} \mu_{ij} = 1$ and $\mu_{ij} = 0$ if $||\underline{x}_i - \underline{v}_j||_2^2 \neq 0$. The Fuzzy c-Means algorithm is an iterative algorithm based on (15), (16), and (17) which is given by the following steps:

1) Choose clustering parameters and initial cluster centers. We choose a "fuzziness factor" $m > 1$ which determines degree of fuzziness associated with the resulting membership functions in determining cluster centers. When m is close to 1, the clustering is determined with very "crisp" input membership functions, and with larger m, cluster centers are determined with more overlap between clusters. We also choose N or the number of clusters to calculate. The number of clusters, N, must be less than or equal to the number of input training data points ($N \leq m_F$). N will end up being the number of rules in our fuzzy system. We also choose $\epsilon_c > 0$ which is the error tolerance in calculating cluster centers as outlined in Step 2 below. To initialize the algorithm, we must choose initial cluster centers $\underline{v}_j^0, j = 1, \cdots, N$. This choice is somewhat arbitrary and may affect the final solution, but as is standard we choose initial cluster centers via a random number generator based on the interval present in the input data. That is we choose $\underline{v}_j^0$ as a random vector based on the range of the input portion of the training data so that

$$\begin{aligned} \underline{v}_j^0 \in & \left[\min_{1 \leq i \leq m_F} \{x_i^1\}, \max_{1 \leq i \leq m_F} \{x_i^1\} \right] \\ & \times \cdots \times \left[\min_{1 \leq i \leq m_F} \{x_i^n\}, \max_{1 \leq i \leq m_F} \{x_i^n\} \right] \end{aligned} \tag{18}$$

for all $j = 1, \cdots, N$.

2) Compute cluster centers.
In Step 2 we compute new cluster centers based on those computed in the previous iteration. In particular, we let

$$\mu_{ij}^{\text{new}} := \left[\sum_{k=1}^{N} \left(\frac{||\underline{x}_i - \underline{v}_j^{\text{old}}||_2^2}{||\underline{x}_i - \underline{v}_k^{\text{old}}||_2^2} \right)^{m-1} \right]^{-1} \tag{19}$$

for each $i = 1, \cdots, m_F$ and for each $j = 1, \cdots, N$ and also let

$$\underline{v}_j^{\text{new}} := \sum_{i=1}^{m_F} (\mu_{ij}^{\text{new}})^m \underline{x}_i \Big/ \sum_{i=1}^{m_F} (\mu_{ij}^{\text{new}})^m \tag{20}$$

for each $j = 1, \cdots, N$.

3) Compare to the tolerance.
In Step 3 we compute the Euclidean distance between the new cluster centers and those calculated in the previous iteration. The next step in the algorithm is determined by (for all $j = 1, \cdots, N$)

$$||\underline{v}_j^{\text{new}} - \underline{v}_j^{\text{old}}||_2 > \epsilon_c \Rightarrow \text{Go to Step 2} \tag{21}$$

$$||\underline{v}_j^{\text{new}} - \underline{v}_j^{\text{old}}||_2 \leq \epsilon_c \Rightarrow \text{Terminate Algorithm.} \tag{22}$$

The results obtained using the Fuzzy c-Means algorithm given above may calculate a local minimum or may not converge. Recent work has addressed some of these problems [13]–[15].

The Fuzzy c-Means algorithm requires three design parameters to be chosen to perform fuzzy clustering. These are the number of clusters (rules) N, the fuzziness factor m, and the tolerance ϵ_c. These are chosen to suit the particular application. For instance, the number of clusters, N, increases the functional capability of the resulting fuzzy system but also increases the computational complexity in the clustering algorithm and the overall fuzzy system. The fuzziness factor, m, reflects the amount of overlap between clusters. If the system to approximate behaves very differently for different clusters, then m should be chosen small. Otherwise, if the system to approximate is rather smooth, then m can be chosen larger. The tolerance ϵ_c provides a measure which is used to terminate the Fuzzy c-Means clustering algorithm. A smaller value of ϵ_c may result in more iterations of the clustering algorithm, but may result in obtaining more accurate cluster centers. In Section III, more details on the choice of $N, m,$ and ϵ_c are provided for our application.

B. Optimal Output Predefuzzification and the Resulting Fuzzy System

Optimal output predefuzzification is a technique to choose the parameters of the crisp output function for each fuzzy rule in the Takagi–Sugeno fuzzy model so that the relation fits the training data in a weighted least-squares sense. Specifically, each cluster calculated in the Fuzzy c-Means algorithm forms the premise portion of a single fuzzy rule. For each rule, we wish to minimize the squared error weighted by the value of the cluster membership between the output portion of the training data set and a parameterized function of the input portion of the training data set. We wish to minimize the cost

$$J_j = \sum_{i=1}^{m_F} \mu_{ij}(\underline{c}_j^T \tilde{\underline{x}}_i - y_i)^2 \tag{23}$$

for each $j = 1, \cdots, N$ where μ_{ij} is the membership value of the input portion of the ith training data point for the jth cluster, y_i is the output portion of the ith training data point, and the multiplication of $\underline{c}_j^T$ and $\tilde{\underline{x}}_i$ defines the output $\hat{y}^j$ associated with the jth rule for the ith training point. The solution to this minimization problem is a well known result from elementary linear algebra, can be found in [16], and is given by

$$\underline{c}_j = (\hat{X} D_j^2 \hat{X}^T)^{-1} \hat{X} D_j^2 Y \tag{24}$$

for $j = 1, \cdots, N$ where

$$\hat{X} = \begin{bmatrix} 1 & \cdots & 1 \\ \underline{x}_1 & \cdots & \underline{x}_{m_F} \end{bmatrix} Y = [y_1 \cdots y_{m_F}]^T,$$
$$\text{and} \quad D_j = \text{diag}([\mu_{1j} \quad \cdots \quad \mu_{m_F j}]). \tag{25}$$

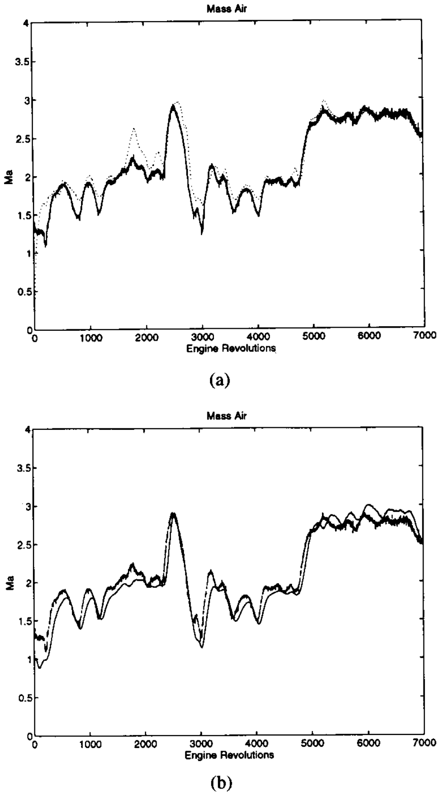
To avoid numerical problems associated with computing the inverse in (24), we actually used weighted recursive least squares [11], [17] to compute the values for $\underline{c}_j$ in (24).

The techniques of fuzzy clustering and optimal output predefuzzification are combined to construct a fuzzy system in the following manner. First, cluster centers $\underline{v}_j, j = 1, \cdots, N$ are calculated via Fuzzy c-Means clustering. These centers, together with the fuzziness factor m, determine the input membership functions given by

$$\mu_{H^j}(\underline{x}) = \left[\sum_{k=1}^{N} \left(\frac{||\underline{x} - \underline{v}_j||_2^2}{||\underline{x} - \underline{v}_k||_2^2} \right)^{m-1} \right]^{-1}. \tag{26}$$

Next, the technique of optimal output defuzzification is applied to the training data to calculate the linear function which defines the crisp output for each rule $\hat{y}^j = \underline{c}_j^T \tilde{\underline{x}}$. The resulting fuzzy system is given by (14) with

$$\underline{\theta} = [\underline{v}_1^T, \cdots, \underline{v}_N^T, \underline{c}_1^T, \cdots, \underline{c}_N^T]^T. \tag{27}$$



(a)

(b)

Fig. 3. Mass air—(solid) measured, (dotted) estimate. The data shown are voltage signals from the sensor so that the units for the vertical axes are in volts for both cases. Suppose that x denotes the voltage signal from the sensor. The actual value of $m_a = -4.2073 \times 10^{-4} + (5.2296 \times 10^{-3})x + (1.3396 \times 10^{-3})\ x^2 + (2.1860 \times 10^{-3})\ x^3$.

III. Identification and Fault Detection and Isolation Strategy

For training purposes, data were collected to calculate the necessary models f_ω, f_{m_a}, f_α, f_{m_f}, and f_{T_L}. Due to mechanical constraints on the electric dynamometer, we cut the IM240 cycle short to only 7000 engine revolutions for the tests that we ran. In addition, a uniformly distributed random signal was added to the throttle and torque inputs to excite the system. The data generated was utilized to construct five multi-input single-output fuzzy systems, one for each of (2)–(6). In fuzzy clustering, we choose 10 clusters ($N = 10$), a fuzziness factor $m = 2$, and tolerance $\epsilon_c = 0.01$ for each of the constructed fuzzy systems. These were derived via experimentation until desired accuracy was achieved (e.g., increasing N to more than 10 did not provide improved estimation accuracy). For comparison purposes, we also calculated models utilizing the Nonlinear ARMAX technique based on the same experimental data. Then the experimental test cell was run and the models derived through fuzzy clustering and the nonlinear ARMAX

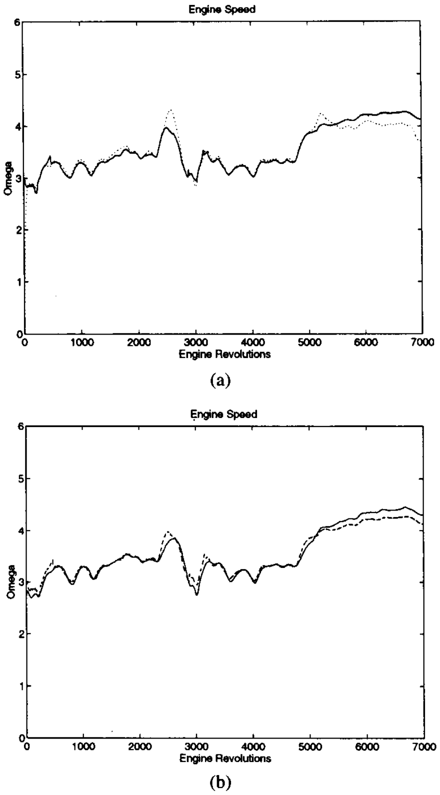


Fig. 4. Engine speed—(solid) measured, (dotted) estimate. The data shown are voltage signals from the sensor so that the units for the vertical axes are in volts for both cases. Suppose that x denotes the voltage signal from the sensor. The actual value of $\omega = -1990.5 + 1044.7x$.

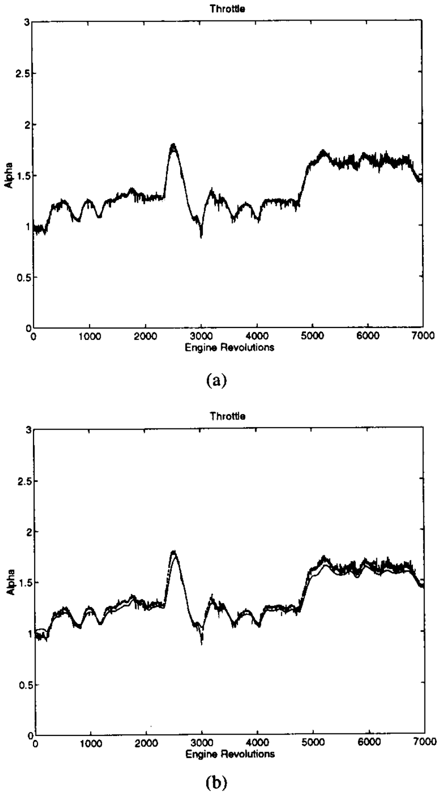


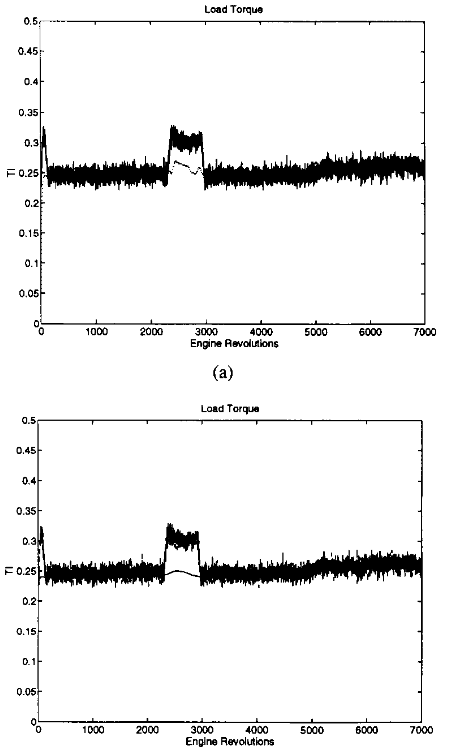
Fig. 5. Throttle—(solid) measured, (dotted) estimate. The data shown are voltage signals from the sensor so that the units for the vertical axes are in volts for both cases. Suppose that x denotes the voltage signal from the sensor. The actual value of $\alpha = -82.905 + 124.70x - 46.241x^2 + 9.1047x^3 - 0.67342x^4$.

technique were validated by collecting data for similar tests run on different days. The results in identification with the validation data (not the training data) for both techniques are given in Figs. 3–7. We measure approximation error by evaluating the squared error between the real and estimated value (which we denote by $\sum_k e^2$ where k ranges over the entire simulation time). As the results show, both techniques approximate the real system fairly well; however, for the mass air and engine speed estimates the ARMAX technique performed slightly better than the clustering technique. For the throttle, load torque, and the mass fuel estimates, the clustering technique estimated slightly better than the NARMAX technique.

We see that there is no clear overall advantage to using NARMAX or the fuzzy identifier, even though the fuzzy identifier performs better for estimating several variables. We would comment, however, that it took a significant amount of experimental work to determine where to truncate the polynomial expansion in (12) for the NARMAX model structure. The parameters N, m_F, and ϵ_c for the fuzzy identifier construction were, however, quite easy to select. Moreover, the fuzzy identifier approach provides the additional useful piece of information that the underlying system seems to be adequately represented by interpolating between 10 linear systems, each of which is represented by the output of the 10 rules ($N = 10$). Since the use of the NARMAX results for FDI system development were studied in [5], we will henceforth only use the fuzzy identifiers.

The models identified through fuzzy clustering allow us to utilize system residuals (e.g. $\hat{\omega} - \omega$, $\hat{m}_a - m_a$, $\hat{\alpha} - \alpha$, and $\hat{m}_f - m_f$) to detect and isolate failures. A specific fault may be isolated by referring to the fault isolation logic given in Table II that was developed by the "indirect decoupling method" outlined in [18]. In the body of Table II, we indicate a pattern of "zero," "nonzero," and "-" (don't care) residuals that will allow us to identify the appropriate failure. We use thresholds to define what we mean by "zero" and "nonzero" and explain how we choose these thresholds below. As an example, if the scaled values (we will explain how the residuals are scaled below) of $|\hat{\omega} - \omega|$, $|\hat{m}_a - m_a|$, and $|\hat{m}_f - m_f|$ are above some thresholds, and $|\hat{\alpha} - \alpha|$ is below some threshold, then we say that there is a m_f actuator calibration fault. For an m_a sensor calibration fault the (scaled) value of $|\hat{\omega} - \omega|$ should be nonzero since this residual is not completely decoupled, but it is very weakly coupled through the load torque model. Therefore we have the "-" (don't care) term for the $\hat{\alpha} - \alpha$ residual for an m_a sensor calibration fault.

The models developed via fuzzy clustering are only approximations of the real engine dynamics. Therefore, since the system residuals do not identically equal zero during

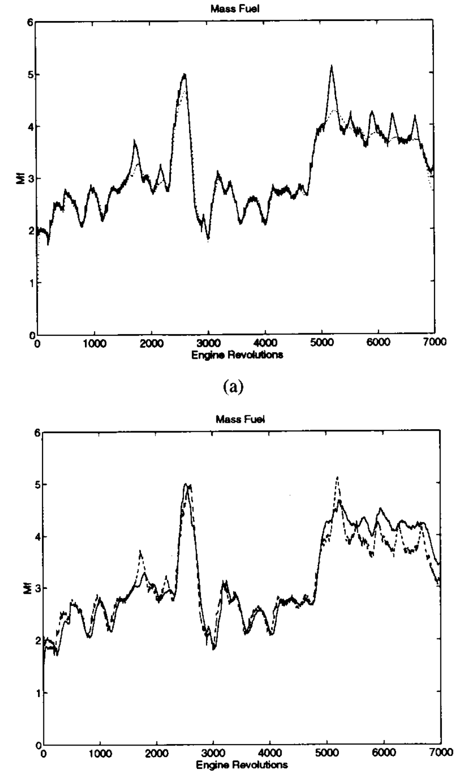


(a)

(b)

Fig. 6. Load torque—(solid) measured, (dotted) estimate. The data shown are voltage signals from the sensor so that the units for the vertical axes are in volts for both cases. Suppose that x denotes the voltage signal from the sensor. The actual value of $T_L = -2.5384 + 0.99857x$.

nominal no fault operation, it is necessary to perform some post processing of the residuals to detect and isolate the faults we consider. We perform a low-pass filtering of system residuals and a setting of thresholds to determine nonzero residuals. We implement a fourth-order Butterworth low-pass filter with a cutoff frequency of $\frac{\pi}{100}$ and pass the residuals through this filter. Next, we take the filtered residual and scale it by dividing by the maximum value of the signal over the entire IM240 cycle. The filtered and scaled residual is then compared against a threshold, and if the threshold is exceeded, then a binary signal of one is given for that particular residual for the remainder of the test. The threshold values for each residual used in the FDI strategy are computed empirically by analyzing the deviation of the residuals from zero during no fault operation. These thresholds are given in Table III (e.g., from Table III, if the filtered and scaled residual for m_a is greater than 0.3, then we say the $\hat{m}_a - m_a$ residual threshold has been exceeded, i.e., that it is "nonzero"). We perform tests utilizing the FDI strategy by simulating calibration faults and using the filtered residuals. Specifically, calibration faults are simulated by multiplying the experimental data for a specific fault by the desired calibration fault value. For instance, to obtain a 20% ω calibration fault, we multiply ω by 1.20. Through experimentation, we have found this to be an accurate representation of a true calibration fault.



(a)

(b)

Fig. 7. Mass fuel—(solid) measured, (dotted) estimate. The data shown are voltage signals from the sensor so that the units for the vertical axes are in volts for both cases. Suppose that x denotes the voltage signal from the sensor. The actual value of $m_f = 5.3 \times 10^{-6}(-1.2015 + 3.3304x)$.

TABLE II
CATALOG OF SYSTEM RESIDUALS AND CORRESPONDING FAULTS

Fault Location	$\hat{\omega} - \omega$	$\hat{m}_a - m_a$	$\hat{\alpha} - \alpha$	$\hat{m}_f - m_f$
m_a sensor	-	non-zero	non-zero	non-zero
ω sensor	non-zero	zero	non-zero	non-zero
α input	non-zero	non-zero	non-zero	zero
m_f input	non-zero	non-zero	zero	non-zero

TABLE III
THRESHOLDS FOR SYSTEM RESIDUALS

Residual	Threshold
$\hat{m}_a - m_a$ sensor	± 0.30
$\hat{\omega} - \omega$ sensor	± 0.10
$\hat{\alpha} - \alpha$ input	± 0.04
$\hat{m}_f - m_f$ input	± 0.15

We look at only a portion of the IM240 cycle when we test for faults. The portion we observe is between 3000 and 5000 revolutions of the engine (about 1 minute). During this portion, the best model matching occurred. Fig. 8 shows the residuals lying within the thresholds for the duration of the test signaling a no fault condition. In the second test, a 20% calibration fault exists in the throttle actuator meaning that the throttle angle is 1.20 times the commanded value. As Fig. 9 illustrates, all residuals exceed the threshold except the m_f

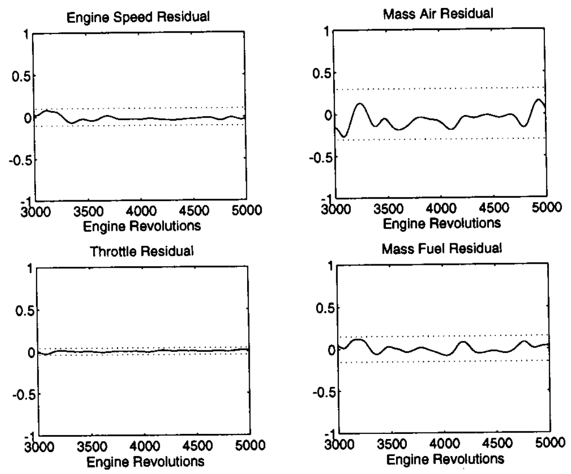


Fig. 8. System residuals with no fault. (The vertical axes are dimensionless; they represent the size of the filtered residual divided by the maximum value of the signal achieved over the IM240 cycle.)

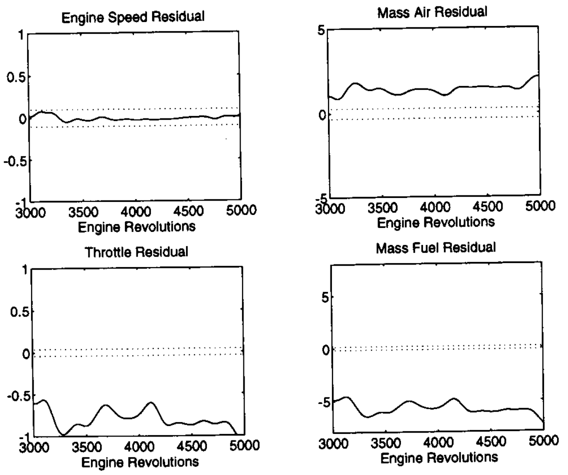


Fig. 10. System residuals with 20% mass air calibration fault. (The vertical axes are dimensionless; they represent the size of the filtered residual divided by the maximum value of the signal achieved over the IM240 cycle.)

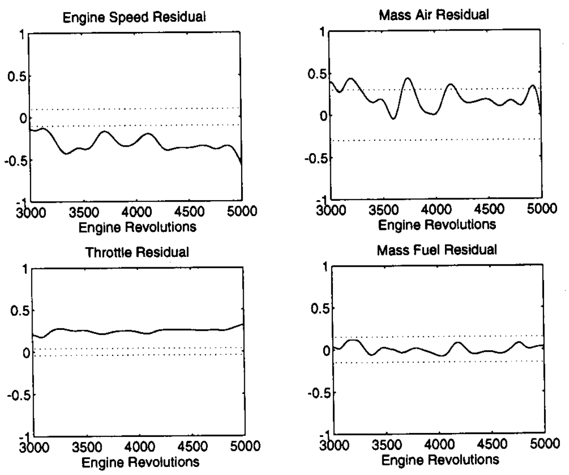


Fig. 9. System residuals with 20% throttle calibration fault. (The vertical axes are dimensionless; they represent the size of the filtered residual divided by the maximum value of the signal achieved over the IM240 cycle.)

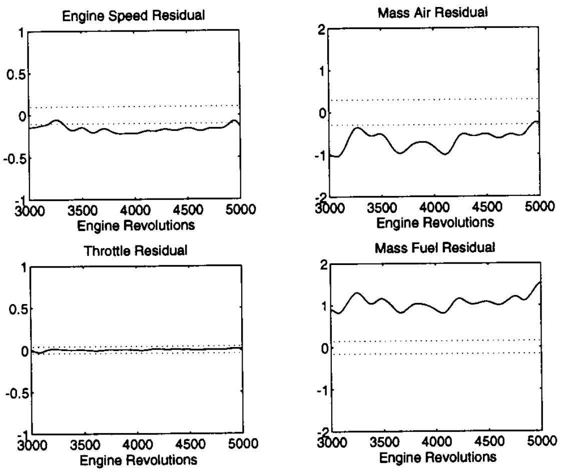


Fig. 11. System residuals with 40% mass fuel calibration fault. (The vertical axes are dimensionless; they represent the size of the filtered residual divided by the maximum value of the signal achieved over the IM240 cycle.)

residual which, by Table II, indicates that a throttle fault is present. In the third test, a 20% calibration fault exists in the mass air sensor meaning that the mass air sensor reads 1.20 times the real value. As Fig. 10 illustrates, all residuals exceed the thresholds except the engine speed residual which, by Table II, indicates a m_a sensor fault. In the fourth test, a 40% calibration fault exists in the mass fuel actuator meaning that the mass fuel actuator injects 1.40 times the commanded value. As Fig. 11 illustrates, all residuals exceed the thresholds except the throttle residual which, by Table II, indicates that a m_f fault occurred. In the final test, a 20% calibration fault exists in the engine speed sensor. As Fig. 12 illustrates, all residuals exceed the thresholds except the mass air residual signaling an engine speed sensor fault as shown in Table II. In a similar manner, engine failures can be detected utilizing the models calculated via the nonlinear ARMAX model; however, the resulting residuals are not shown here as they were very similar. Overall, we see that by combining the estimates from the fuzzy identifiers with the FDI logic in Table II, we were able to provide an effective FDI strategy for a class of minor engine failures.

IV. Conclusion

In this brief paper, we have utilized a method of fuzzy clustering and optimal output predefuzzification in [6] to perform identification of an experimental internal combustion engine to implement fault detection and isolation for a class of calibration faults during the IM240 testing cycle. We have demonstrated that fuzzy identification is one means to perform effective fault detection and isolation for the engine faults we considered, and we showed that the fuzzy

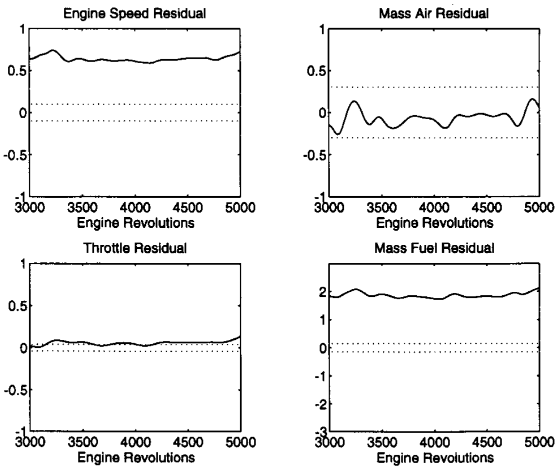


Fig. 12. System residuals with 20% engine speed calibration fault. (The vertical axes are dimensionless; they represent the size of the filtered residual divided by the maximum value of the signal achieved over the IM240 cycle.)

identification approach performed similarly to a nonlinear ARMAX technique utilized in [5].

References

[1] M. H. Costin, "On-board diagnostics of vehicle emission system components: Review of upcoming government regulation," in *IFAC/IMACS Symp Fault Detection Supervision Safety Tech. Processes*, 1991, pp. 235–239.

[2] J. J. Gertler, M. Costin, X. Fang, R. Hira, Z. Kowalczuk, and Q. Luo, "Model-based on-board fault detection and diagnosis for automotive engines," *Contr. Eng. Practice*, pp. 116–132, Feb. 1993.

[3] G.-C. Luh, "Multi-input multi-output modelling of nonlinear systems with application to internal combustion engine modelling," Ph.D. dissertation, Dept. of Mech. Engineering, Ohio State Univ., 1994.

[4] G.-C. Luh and G. Rizzoni, "Identification of a nonlinear mimo ic engine model during im/240 driving cycle for on-board diagnosis," in *Proc. Amer. Contr. Conf.*, Baltimore, MD, 1994, pp. 1581–1584.

[5] V. Krishnaswami, G.-C. Luh, and G. Rizzoni, "Fault detection in ic engines using nonlinear parity equations," in *Proc. Amer. Contr. Conf.*, Baltimore, MD, 1994, pp. 2001–2005.

[6] S.-K. Sin and R. J. Defigueiredo, "Fuzzy system design through fuzzy clustering and optimal predefuzzification," in Proc. *2nd IEEE Conf. Fuzzy Syst.*, San Francisco, CA, 1993, pp. 190–195.

[7] L. X. Wang, *Adaptive Fuzzy Systems and Control.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybernectics*, vol. SMC-15, no. 1, pp. 116–132, 1987.

[9] J. C. Bezdek, "A convergence theorem for the fuzzy isodata clustering algorithms," in *Fuzzy Models for Pattern Recognition*, J. C. Bezdek and S. K. Pal, Eds. Piscataway, NJ: IEEE Press, 1992, pp. 130–137.

[10] I. J. Leontaritis and S. A. Billings, "Input–output parametric models for nonlinear systems Part I: Deterministic nonlinear systems," *Int. J. Contr.*, vol. 41, no. 2, pp. 303–328, 1985.

[11] L. Ljung, *System Identification: Theory for the User.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[12] I. J. Leontaritis and S. A. Billings, "Input–output parametric models for nonlinear systems Part II: Stochastic nonlinear systems," *Int. J. Contr.*, vol. 41, no. 2, pp. 329–344, 1985.

[13] M. P. Windham, "Geometrical fuzzy clustering algorithms," in *Fuzzy Models for Pattern Recognition*, J. C. Bezdek and S. K. Pal, Eds. Piscataway, NJ: IEEE Press, 1992, pp. 123–129.

[14] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence theory for fuzzy c-means: Counter examples and repairs," in *Fuzzy Models for Pattern Recognition*, J. C. Bezdek and S. K. Pal, Eds. Piscataway, NJ: IEEE Press, 1992, pp. 138–142.

[15] M. J. Sabin, "Convergence and consistency of fuzzy c-means/isodata algorithms," in *Fuzzy Models for Pattern Recognition*, J. C. Bezdek and S. K. Pal, Eds. Piscataway, NJ: IEEE Press, 1992, pp. 143–150.

[16] D. Kincaid and W. Cheney, *Numerical Analysis.* California: Brooks/Cole, 1991.

[17] E. G. Laukonen, "Training fuzzy systems to perform estimation and identification," Master's Thesis, Dept. of Electrical Engineering, Ohio State Univ., 1994.

[18] V. Krishnaswami and G. Rizzoni, "Nonlinear parity equation residual generation for fault detection and isolation," in *Proc. IFAC/IMACS Symp. Fault Detection, Supervision Saftey Tech. Processes-SAFEPROCESS 1994*, Espoo, Finland, 1994, pp. 317–322.