International Journal of Industrial Electronics, Control and Optimization © 2019 IECO
Vol. 2, No. 4, pp. 331-342, Oct (2019)

Fault Detection and Identification of High Dimensional System by GLOLIMOT

S.M Emad Oliaee[1], Mohammad Teshnehlab[2†], and Mahdi Mahdi Aliyari Shoorehdeli [3]

[1,2]Control Engineering Dept., Electrical Faculty, K.N.Toosi University of Technology, Tehran, Iran
[3] Mechatronics Dept., Electrical Faculty, K.N.Toosi University of Technology, Tehran, Iran

ABSTRACT

The Local Model Network (LMN) is one of the common structures to model systems and fault detection and identification. This structure covers the disadvantages of training in fuzzy systems and interpretations in neural networks at the same time. But the algorithms that have been introduced to create LMN, such as LOcal LInear MOdel Tree (LOLIMOT), are very sensitive to the dimension of input space. In other words, the search space and the number of network parameters are increased exponentially by increasing the input dimension, which is called the curse of dimensionality. Therefore in this paper, the LMN structure has been developed, and a new incremental algorithm has been proposed which is based on Genetic algorithm and LOLIMOT algorithm that is called GLOLIMOT. The proposed idea reduces and optimizes the search space dimension. The proposed idea and the conventional structure are tested on single-shaft industrial gas turbine prototype model, which has high complexity and high dimension. The results indicate improvement in performance of the proposed structure and algorithm.

Article Info

Keywords:
FDI, Gas Turbine, Genetic Algorithm, GLOLIMOT, LMN, LOLIMOT

Article History:
Received 2018-10-25
Accepted 2019-01-22

I. Introduction

Local Linear Models (LMNs) are popular models for the systems identification and extracting the operating regions (partitions). The LMN describes training data by interpolating local models. The local models are polynomials that are interpolated by validity functions [1]. Gaussian and Sigmoid function are as the validity functions.

A. Survey over the Related Works

The design of the sub-models in the LMN is based on divide and conquer strategy. Hence, this strategy such as incremental partitioning has attracted more attention in recent years. Incremental partitioning is in contrast to experimental partitioning that requires prior knowledge. In other words, incremental partitioning requires very little or no prior knowledge [2]. These methods break down each part of the system into two smaller subsystems by partitioning and considering criteria in each step. Then the linear model is estimated for the two new local subsystems. In other words, a local model is added to the number of local models in each step. Popular partitioning strategies include axis-orthogonal, axis-oblique partitioning.

A.1. Orthogonal Partitioning

This strategy uses orthogonal axis in parallel to axis of input space for splitting. Fig. 1(left) represents the sketch of this splitting for 2-D input space. LOcally LInear MOdel Tree algorithm (LOLIMOT) [3, 4] and J&F algorithm [5] are the pioneers of this partitioning. Although these two algorithms are very similar, there are significant differences in the estimation of the linear model parameters and the location of the decomposition regimes. While LOLIMOT uses the heuristic method to find parameters and it performs much faster.

†Corresponding Author: teshnehlab@eetd.kntu.ac.ir
Tel: +98-21-840-62160,Fax: +98-21-840-62323, K.N.Toosi University

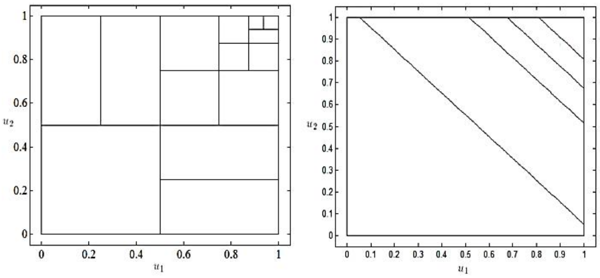

Fig. 1. (left) Axis-orthogonal partitioning and (right) Axis-oblique partitioning for 2-D input space

In the LOLIMOT algorithm, the decomposition of the input space is determined by the local error. Also, unlike the J&F algorithm, Local Models (LM) are easily estimated by the weighted least square method. That's why LOLIMOT's algorithm has become more popular.

Bänfer and Nelles [6] applied higher-degree polynomials in LMs and named this algorithm, the POLYnomial MOdel Tree (POLYMOT). In each iteration of this algorithm, there are two options. Choice of either increase of complexity of the worst LM or increase number of LMs with splitting of the worst LM. Also, Nekoui and Sajadifar [7] have optimized the nonlinear parameters of the validity functions by Particle Swarm Optimization (PSO) optimization. Moreover, Mehran [8] used the PSO optimization to find the best axis-orthogonal partition. The Expectation-Maximization (EM) algorithm is also implemented to identify local models by Rezaei [9]. Jakubek and Hametner [10] have optimized and determined the splitting location by the generalized total least squares and EM algorithms. Jakubek and Keuth [11] used clustering for partitioning and the local model statistic which approximates reliability of the obtained model to reduce the computational cost with the lower number of LMs. Sarabi-Jamab and N.Araabi have provided merge and split strategy to decrease the number of LMs by using Piecewise Linear Network (PLN) [12] and Simulated Annealing (SA) [13]. Also, a nonlinear fault detection and identification (FDI) of gas turbine basesd on deep neural networks and LMN is presented in [14].

Recently, the neuro-fuzzy structures with Genetic Algorithms (GA) have been utilized widely. Karaboga and Kaya have presented a review of Adaptive network fuzzy inference and its training approaches in [15]. The optimal division of the Piecewise Linearization for nonlinear function approximation has been carried out by GA based on clustering approach in [16]. Azimi et al.[17] have combined adaptive neuro-fuzzy inference systems and genetic algorithms to predict the discharge coefficient of rectangular side orifices. Furthermore, in [18], GA and a Fuzzy Logic are combined for network anomaly detection. The premise and the consequence parameters of the adaptive neuro-fuzzy inference system (ANFIS) are optimized with GA to control MIMO systems by Lutfy et al. [19].

A.2. Oblique partitioning

In this strategy, unlike axis-orthogonal partitioning, axis splitting of input space is performed with an angle. Fig. 1(right) shows the sketch of this partition. Breiman [20] was the first one who introduced this strategy with hinging hyperplanes.

Further, Ernst proposed an algorithm based on LOLIMOT in [21]. In every iteration of Ernst's algorithm, nonlinear optimization like gradient descent is performed to find unknown parameters of the hinge function of the two new LMs. Then Nelles added a series of steps to the LOLIMOT algorithm and the Ernst algorithm, and named it HILOMOT [22, 23]. The sigmoid functions are used Instead of hinge functions as validity functions in HILOMOT. Obviously, employing nonlinear optimization is necessary to reach the optimal splitting of the input space[1].

In continues, Fischer [1] eliminated the numerical gradient calculation and utilized quasi-Newton optimization to increase the speed of nonlinear optimization. Hartmann and Nelles [24] improved this algorithm by taking into account the smoothness in the determination of validity functions. They also combined the concepts of HILOMOT and POLYMOT and called it HILOMOT + [25]. Xu, Xuang, and Wange [26] used this strategy and the piecewise linear model's concept for control, prediction, and identification of the dynamic system.

B. Similar Works

A lot of works has been done to identify and detect the fault in the gas turbine up to now, that several effective cases will be reviewed in continue.

In all papers, the number of input feature does not exceed 15 features. Furthermore, in most of them, there has been no attempt to propose a nonlinear FDI method, while non-linear identification is more appropriate due to the presence of noise and uncertainty. For example [27] has used the classic observer to detect faults in a gas turbine, but this approach is applicable for linear systems not as the same as gas turbine. [28] has considered the dynamic observer and neural networks to detect sensor fault which faults were modeled as step functions. [29] has used a neuro-fuzzy- method to detect and isolate only two faults in gas turbine. In [30], four incipient faults are considered, and the linear dynamical identification has been used for FDI based on the observer. In [31], a combination of MLP and LMN structures are presented to robust FDI for the gas turbine in a steady state. Also, every four faults are considered as the ramp function.

C. Our Approach

All of these algorithms have poor performance when faced with high dimension in input space. Since the split and search space is enlarged exponentially and consequently, optimization becomes failed [1, 22-25]. Therefore, in this paper, the LMN structure has been developed. In this structure, only the size of split space is reduced by a matrix, while the input dimension is complete for the linear model in each LM. This matrix transfers the input data to new space with smaller dimension. The GA has been used to optimize the location of data in new space. In other words, this paper unlike other methods doesn't change the location of the splitting, but it attempts to project the input in such a way that it prepares optimal condition for the system's identification. Therefore a new incremental algorithm is proposed to create the developed LMNs. Since the proposed algorithm is based on GA and LOLIMOT; it is named GLOLIMOT.

In this paper, the proposed structure and algorithm are tested for FDI on the single-shaft industrial gas turbine prototype model. The simulated result illustrates the good performance of the proposed structure and algorithm.

D. Paper organization

The paper is organized as follows: LMN and LOLIMOT algorithm are reviewed in the next section. The developed LMN and its training algorithm (GLOLIMOT) are introduced in section III. Later, in section IV, the proposed structure is utilized for FDI on the model of gas turbine, and its result is compared with the conventional LMN. Finally, the conclusion is offered in section V.

II. LOCAL LINEAR MODEL AND LOLIMOT ALGORITHM

The neuro-fuzzy networks can be considered as a kind of fuzzy model that is not designed by the expert entirely, even at least, some parts are learned by data. In other words, the neuro fuzzy networks are the fuzzy model that is drawn in the structure of the neural network, and the concepts of neural network training methods have been implemented on it. Although many training methods were presented in continue which were not related to neural networks, the name of neuro-fuzzy networks has still retained for all fuzzy models that have data-learning education[32].

The Local Linear Neuro-Fuzzy models (LLNF) are one example of neuro-fuzzy networks. The local modeling method is based on divide and conquer strategy. It means that the complex problem is broken down into some simpler and smaller sub-models to be identified easily with simpler models such as linear models. The LLNF structure is called too, the LMN, the Takagi-Sugeno fuzzy model, the Piecewise Models, and Local Regression techniques [33]. The LMN is presented in Fig. 2.

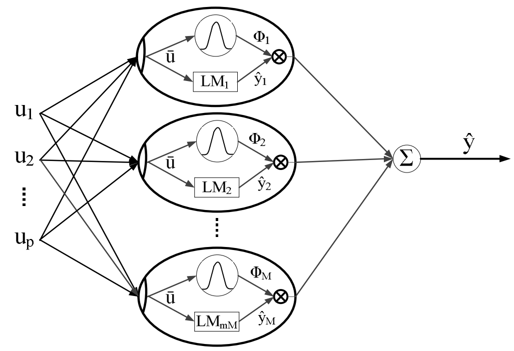

Fig. 2 LMN structure [4]

The output of the LMN structure ($\hat{y}$), is obtained from the weighted interpolation of M LMs (Eq. 1).

$$\hat{y}\left(u_i\right)=\sum_{j=1}^{M}\hat{y}_j\left(\underline{u_i}\right)\Phi_j\left(\underline{u_i}\right) \quad (1)$$

$$=\sum_{j=1}^{M}\left(w_{j,0}+w_{j,1}u_{i,1}+w_{j,2}u_{i,2}+\ldots+w_{j,p}u_{i,p}\right)\Phi_j\left(u_i\right)$$

Where $\underline{u_i}$ is the i-th sample of the input vector. Also, $\hat{y}_j(.)$ is the linear part of the j-th local model, like the consequent part of the j-th rule in the fuzzy structure. The polynomials are the most common functions for this part. This polynomial can be zero degree (constant), which is similar to the mamdani fuzzy system or Normalize Redial Base Function (NRBF) network. But the most popular option is first-order polynomial (linear) [35 , 34]. So $w_j=\left[w_{j,0}\ w_{j,1}\ \ldots\ w_{j,p}\right]^T$ are its linear coefficients. This coefficients vector are estimated by the weighted least squares (WLS) method. The WLS solution of the rule conclusion parameters is given by:

$$W_j=\left(X^TQ_jX\right)^{-1}X^TQ_jy \quad , \quad j=1,\ldots,M \quad (2)$$

Where X and Q are the regression matrix, and the weight matrix respectively:

$$X=\begin{bmatrix}1 & u_{1,1} & u_{1,2} & \cdots & u_{1,p}\\ 1 & u_{2,1} & u_{2,2} & \cdots & u_{2,p}\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & u_{N,1} & u_{N,2} & \cdots & u_{N,p}\end{bmatrix} \quad (3)$$

$$Q_j = \begin{bmatrix} \Phi_j(\underline{u}_1) & 0 & \cdots & 0 \\ 0 & \Phi_j(\underline{u}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Phi_j(\underline{u}_N) \end{bmatrix} \quad (4)$$

$\Phi_j(.)$ is the validity function of the j-th local model which plays the role of rules in the primary part of the fuzzy structure. It is created from the normalized membership function (Eq. 5).

$$\Phi_j(u_i) = \frac{\mu_j(u_i)}{\sum_{j=1}^{M}\mu_j(u_i)} \quad , \quad \sum_{j=1}^{M}\Phi_j(u_i) = 1 \quad (5)$$

Which $\mu_j(u_i)$ is the j-th membership function that is formed as Gaussian function (Eq. 6) by production in every dimension.

$$\mu_j(\underline{u}) = \exp\left(-\frac{1}{2}\left(\frac{(u_1 - c_{j1})^2}{\sigma_{j1}^2} + \ldots + \frac{(u_p - c_{jp})^2}{\sigma_{jp}^2}\right)\right)$$
$$= \exp\left(-\frac{1}{2} * \frac{(u_1 - c_{j1})^2}{\sigma_{j1}^2}\right) * \ldots * \exp\left(-\frac{1}{2} * \frac{(u_p - c_{jp})^2}{\sigma_{jp}^2}\right) \quad (6)$$

Where c_{ij} and σ_{ij} are the center coordinate and the individual standard deviation of Gaussian validity functions, respectively. Figure 3 shows a Gaussian membership function (a) and the normalized validity functions (b) for the two inputs.

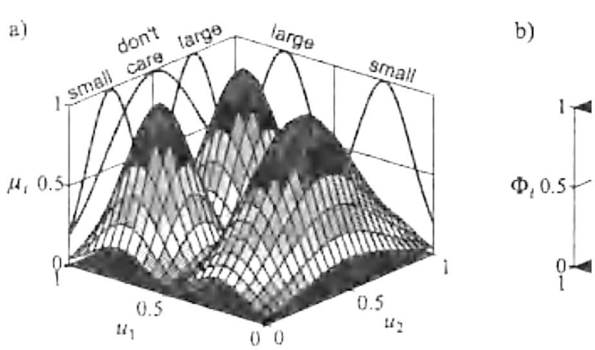

Fig. 3 (a) Gaussian membership function **(b)** the normalized validity functions for the two inputs[32].

A. LOLIMOT algorithm

The LOLIMOT algorithm is an incremental tree method. It means that the algorithm starts with a specific model and then the worst local model is broken down into two local models in each iteration. This action is done by orthogonal splitting in axis of input space with local cost function and general cost function criteria. After each splitting, the validity functions are calculated, and the local linear models have adapted with the Weighted Least Squares (WLS) method. Usually, Mean Square Error (MSE) is considered as the general cost function (general error) (Eq.7) for solving identification problems, while the local cost function (local error) (Eq.8) is used to decompose models [4].

$$J = \frac{1}{N}\sum_{i=1}^{N} e(u_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y(u_i) - \hat{y}(u_i)\right)^2 \quad (7)$$

$$J_j = \sum_{i=1}^{N} e(u_i)^2\,\Phi_j(u_i) \quad , \quad j = 1,\ldots,M \quad (8)$$

Where i is the number of input and J_j is the error of the j-th local model.

The steps in the LOLIMOT algorithm are as follows:

Step 1: Start with an initial model

Step 2: Find the worst LM based on the max local cost function (Eq.8)

Step 3: Break the worst LM

a. Splitting axis-orthogonal in each input dimension

b. Estimating the parameters of two new LLM by WLS (Eq. 2)

c. Calculating general cost function (Eq.7) for each splitting

Step 4: Select the best split based on the lowest general cost function

Step 5: Check the stop condition and go to step 2 if the condition is not met

B. The LMN and LOLIMOT Weakness

The learning ability of LMN is maintained properly with the appropriate set of data in size and complexity. The number of linear coefficients of the local model is related to the input size linearly, and the number of local models is determined by the complexity of the system. As the complexity of the problem increases, the more local models are required to cover the problem. Therefore the total number of parameters increases exponentially. This inefficiency results from both computational cost and the challenge of fuzzy rules generation. In other words, the performance of the algorithm is reduced greatly as the input dimension increases [36].

III. INTRODUCTION OF PROPOSED STRUCTURE AND ALGORITHM

One of the advantages of LMN is dividing the system into smaller ones. By doing so, the complex system is divided into several smaller sub-systems with less complexity. But as the input dimension increases, the search space grows up to generate the new rules. As a result, the performance of partitioning algorithms such as LOLIMOT goes inefficient as mentioned previously. To overcome this problem, the LMN structure has been developed as G-LMN in Fig.4.

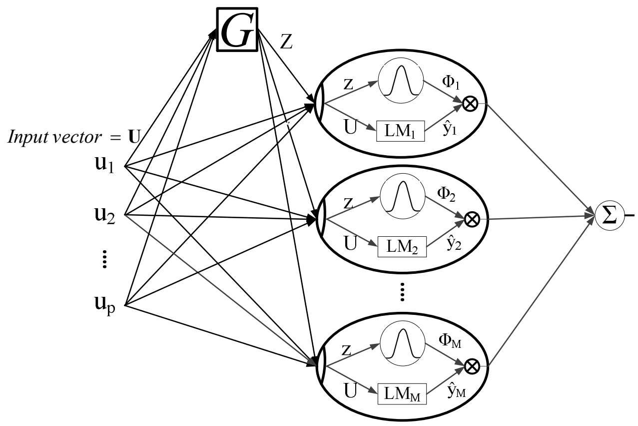

Fig. 4 Proposed structure (G-LMN)

In this figure, the G matrix reduces the size of the input vector (u) to the search space (Z) to increase the speed of the partitioning algorithm and also to rescues from the Curse of dimension. However, all input feature directly is used to construct the linear model in each local model, so that the identification accuracy is not reduced.

This linear model is trained by the WLS method, but the G matrix must be taken in such a way that: the new space is suitable, the vector Z is not outside the interval [-1,1], the identification error also decreases. Therefore, the genetic algorithm is used to optimize and to improve the new space. So, the fitness function of GA is defined as follows:

$$Fitnessfunction\,(u_i) = \frac{1}{2}\sum_{i=1}^{N} e^2(u_i) + \lambda\left[\max\left(1, \max_{i=1}^{N}(Z_i)\right) - 1\right] - \lambda\left[\min\left(-1, \min_{i=1}^{N}(Z_i)\right) + 1\right] \quad (9)$$

This fitness function selects the G matrix with this constraint: The first term reduces the identification error, 2nd and 3rd terms hold the vector Z in [-1,1]. It means that if there is an element in Z vector larger (smaller) than 1 (-1), its cost will be raised by 2nd term (3rd term), and if all elements in Z vector are smaller (larger) than 1 (-1), the 2nd term (3rd term) will be omitted. Also λ is the penalty parameter.

Furthermore, the initial population of the genetic algorithm is 500 chromosomes, and the number of generations is 100 generations. Since the best chromosomes are saved in each iteration, so each generation is in the line of the next generation. For example, to create ten rules (neurons), 10*100 generations are run. On the other hand, each chromosome has 144 genes. 144 is the number of elements in G matrix (48*3=144). Hence the crossover function is selected as Scattered. In this method, offspring are created by combining the genes of parents among random points. The mutation is also considered as Uniform with the rate of 0.01.

The proposed GLOLIMOT algorithm, which includes the genetic algorithm based on the LOLIMOT algorithm, is suggested as follows:

Step 1: Start with an initial model
Step 2: Find the worst LM based on the max local cost function (Eq.8)
Step 3: Break the worst LM
a. Split axis-orthogonal in each input dimension
b. Start GA for each split
b.1. Load the last population
b.2. Estimate the parameters of two new LLM by WLS for each population(Eq. 2)
b.3. Calculating general cost function (Eq.7) for each population
b.4. Find the best G-Matrix (Eq. 9)
Step 4: Select the best split based on the lowest general cost function and save the best population
Step 5: Check the stop condition and go to step 2 if the condition is not met

IV. Simulation and Result

The case study in this paper is a laboratory model of single-shaft industrial gas turbine Siemens V94.2, which is developed at the ALSTOM-ABB POWER center in the UK. This simulator generates the data set in four fault conditions which have been validated by real measurements in steady-state conditions [27]. The faults are as:

1. Compressor contamination fault
2. Thermocouple sensor fault
3. High-pressure turbine seal damage
4. Fuel actuator friction wear.

These faults incept at the 15-th second.

In this paper, the q_c, t_3, p_3, p_7 features are used as model outputs to FDI based on other articles in this field [27, 30, 37, 38]. Furthermore, 16 sensor measurements with their three dynamics (48 features totally), are considered as inputs. The nomenclature of these feature are described in table I. The number of the input dynamics is based on trial and error.

In this way, 4 LMNs are considered for modeling outputs as, p_3 in normal conditions, q_c in fault1, T_3 in fault2, and P_7 in fault3. For a better comparison, the data set are applied to both the proposed structure with GLOLIMOT and the conventional LMN with LOLIMOT and HILOMOT algorithm.

TABLE I
NOMENCLATURE OF INPUT FEATURE

a_v	*valve angle*
ff	*fuel flow*
q_c	*compressor torque*
t_i , $i=2,3,6,7$	*ith section (module) temperature*
p_i , $i=2,3,7$	*ith section (module) pressure*
m_i , $i=1,5$	*ith section (module) mass rate*
p_t	*turbine power*
p_a	*ambient pressure*
p_c	*compressor power*
w_t	*turbine angular rate*

The size of u vector and Z vector has been considered as 48 and 3 respectively in GLOLIMOT, whereas the size of u vector and the size of Z vector are same and equal to 48 in conventional LMN. Figure 5 shows Convergence behavior

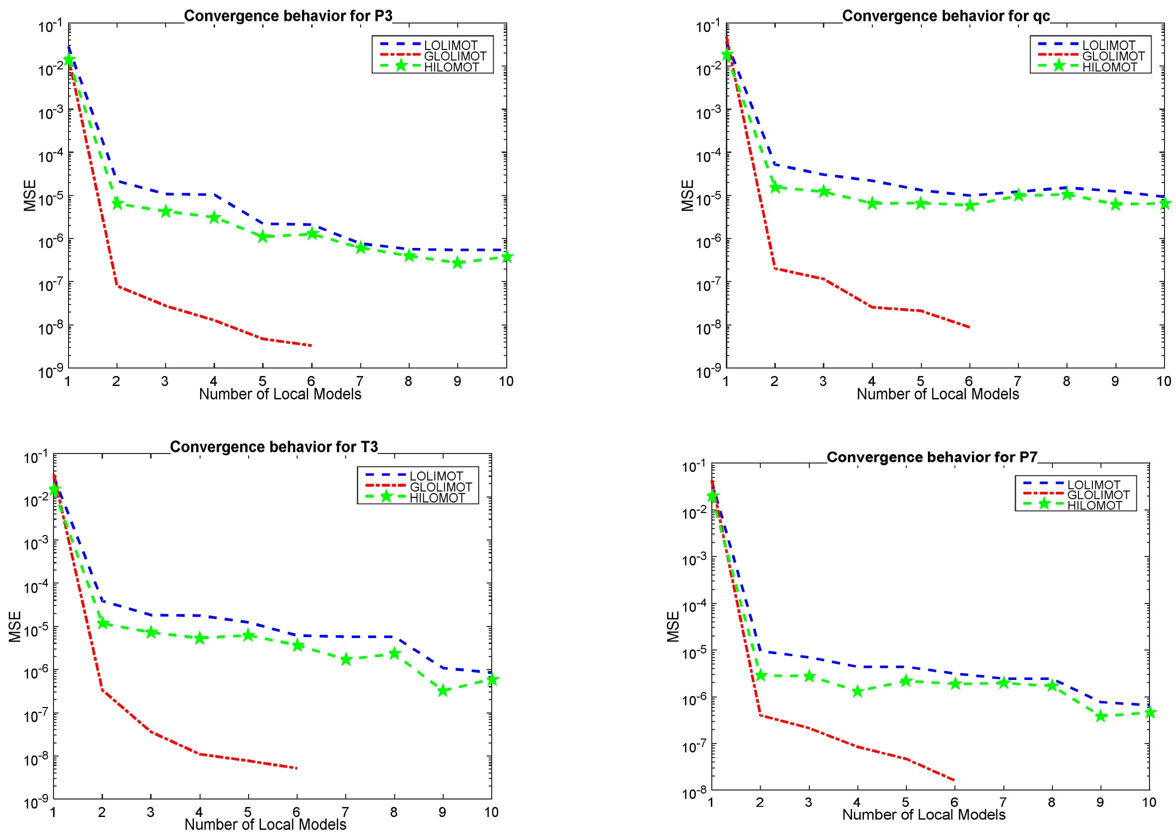

Fig. 5 Convergence behavior for GLOLIMOT and LOLIMOT

for GLOLIMOT and LOLIMOT and HILOMOT with respect to number of local models. The size of dataset from the transient state up to the steady state is 1287 samples in 16 sensors for each condition (normal and 4 faulty condition). Whereas, three dynamic is considered for each sensor, so the size dataset becomes 1287-by-48, which is selected as 70% and 30% for train and test data randomly. Furthermore, two other datasets from different operation points are generated to evaluate the generalization of methods in the identification part. The number of optimal local models and the MSE of each LMNs for both structure are given in table II.

TABLE II

MSE OF PROPOSED LMN AND CONVENTIONAL LMN

GLOLIMOT				
Models	Number LM	Number LM	Number LM	Number LM
$LMN(1) - p_3$	6	3.3e-9	3.7e-6	4.1e-6
$LMN(2) - q_C$	5	8.9e-9	5.9e-5	1.6e-4
$LMN(3) - t_3$	6	5.1e-9	1.6e-6	5.1e -6
$LMN(4) - p_7$	6	1.6e-8	2.7e-5	4.5e-5
LOLIMOT				
Models	Number LM	MSE train	MSE valid1	MSE valid2
$LMN(1) - p_3$	15	5.4e-7	1e-5	1.5e-5
$LMN(2) - q_C$	12	9.3e-6	2.2e-4	3.4e-4
$LMN(3) - t_3$	15	8.4e-7	2.3e-4	7.3e-4
$LMN(4) - p_7$	15	3.6e-7	3.2e-4	9.6e-5

Table II , continued

HILOMOT				
Models	Number LM	MSE train	MSE valid1	MSE valid2
$LMN(1) - p_3$	15	4.4e-7	7.4e-6	5e-6
$LMN(2) - q_C$	12	7.3e-6	1.2e-4	3.4e-4
$LMN(3) - t_3$	15	6.4e-7	1.3e-4	7.3e-4
$LMN(4) - p_7$	15	2.6e-7	1.2e-4	9.6e-5

As shown in Fig. 5 and table II, HILOMOT has a relatively better performance than LOLIMOT due to nonlinear optimization in the generation of rule, but totally both have failed due to the curse of dimension. On the other hands, decreasing and optimizing the search space, reduces the system identification error in the same number of local models. In addition, the number of non-linear parameters of each local model, such as the center and the variance of the membership functions, has fallen from 48*2 to 3*2 (it means about 90 parameters).

In other words, the required number of local models and its parameters are greatly reduced by decreasing the size of splitting space. Moreover, the MSE of the conventional LMN is much greater. This is the same curse of dimension that traps the fuzzy system.

The G matrix is obtained in table III. According to this matrix, each dimension of Z space is created by combination the input features with different weights. Some features play a bold role in some dimension as well as dim role in other dimensions. For example, the last feature is more effective in the first and second dimensions, but has less effect on the third one.

TABLE III
THE G MATRIX

0.0892	0.1964	0.1956
-0.1895	-0.158	0.0913
-0.1572	0.094	-0.2319
0.1791	0.2233	-0.103
0.0544	-0.1217	-0.0897
-0.0821	0.2357	-0.019
0.0363	-0.0359	-0.1173
-0.0179	0.2121	-0.2042
0.0116	0.0094	0.2374
-0.0569	-0.0767	0.0209
0.1848	0.1303	0.2216
-0.0221	-0.0155	0.2306
0.1048	0.1516	0.1074
-0.1192	-0.0819	-0.003
0.1889	-0.15	-0.1896
-0.0452	0.1121	-0.1001
0.0423	0.0479	-0.1492
0.0869	0.0001	-0.0871
0.1976	-0.0563	0.126
-0.2237	-0.1692	-0.2368
-0.2431	-0.2038	0.0595
-0.0539	0.0447	0.058
-0.0976	-0.0745	-0.0896
-0.0583	-0.2355	-0.2127
-0.2169	0.0875	-0.1528
0.0581	-0.0736	-0.0112
-0.0555	0.0143	0.0831
-0.1743	-0.24	-0.1596
-0.133	-0.1352	-0.0677
0.0778	-0.1467	0.2437
-0.1573	0.0328	-0.0086
-0.0535	0.1994	0.1321
-0.1946	0.1924	-0.101
-0.1811	0.1717	-0.1222
-0.2043	-0.1401	-0.1213
0.0175	-0.1277	-0.0178
-0.0861	-0.0519	0.1425
-0.0274	0.1788	-0.0444
-0.0087	0.2204	0.0188
0.0671	0.1683	0.0644
0.1836	-0.0055	0.157
0.0533	-0.1031	-0.0479
0.177	0.1974	0.1099
0.1521	-0.0384	-0.0314
-0.1991	-0.1625	-0.2054
-0.0075	0.145	0.1487
0.0329	0.2467	0.131
0.1823	0.2188	0.0343

A. Fault detection

In order to detect the faults, the calculation differences between G-LMN and the main system is necessary that is called residuals. On the other words, the faults can be detected by comparing the variation of the residuals. The residuals are shown by (R1, R2, R3, R4) in Fig. 6. Also, the residuals with applying different faults are presented in Figs. 7-10. According to these figures, each fault causes different variations and signature in the residuals (S_i). This characteristic is utilized to isolate faults in the next sub-section.

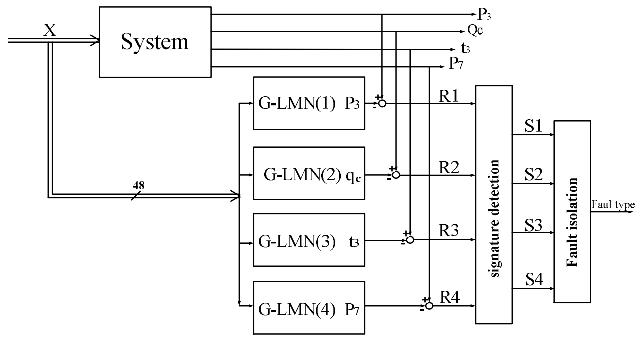

Fig. 6. The proposed combined structure of DNN and four proposed LMNs

Since the un-modeled disturbance and noise are the main enemy of FDI, Utilizing constant thresholding and adaptive thresholding are common ideas to reduce this malfunction. The thresholding method creates a band around the residuals to decide whether a fault occurred or not as follow:

$$\psi(t) = \begin{cases} 0 & \text{if} \quad Th_{Lower} \le R(t) \le Th_{Upper} \\ 1 & \text{if} \quad Th_{Lower} > R(t) \text{ or } R(t) > Th_{Upper} \end{cases} \tag{10}$$

Where ψ is the fault signature. Since simple thresholding (constant Th) increases the false alarm rate (Eq. 12), therefore the adaptive thresholding method is used to detect fault occurrences in this paper. Hence Th can be defined as follow:

$$Th_{Upper/Lower} = m_R \pm \eta S_R \quad , \quad \eta = 1 \text{ or } 2 \text{ or } \dots \tag{11}$$

Where m and S are mean and Standard deviation of residuals and η is a tuning parameter which can be chosen as 1, 2 or 3. Moreover, the False alarm rate is the one of the criteria which is used to evaluate the performance of fault detection. The lower this rate, the higher performance. The false alarm rate is given by:

$$\text{False Alarm rate} = \frac{NF}{N} \tag{12}$$

Where N is the total number of data in a class and NF is the number of data samples of the same class which detected as other class incorrectly. Thus, table IV shows the fault inception time and the fault detection time and the false alarm rate in GLOLIMOT and LOLIMOT.

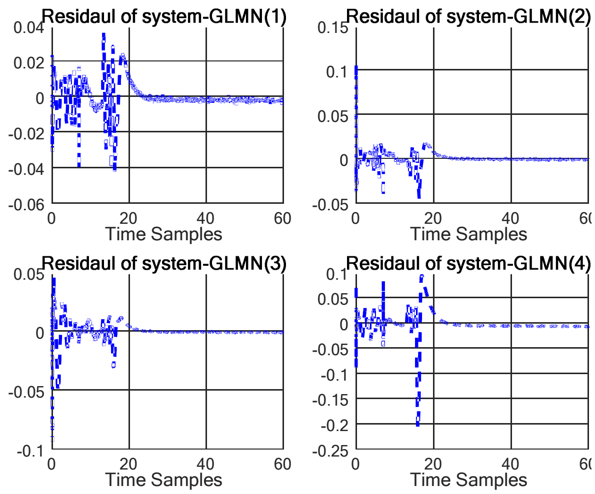

Fig. 7: Residuals of all four G-LMNs by applying the fault1

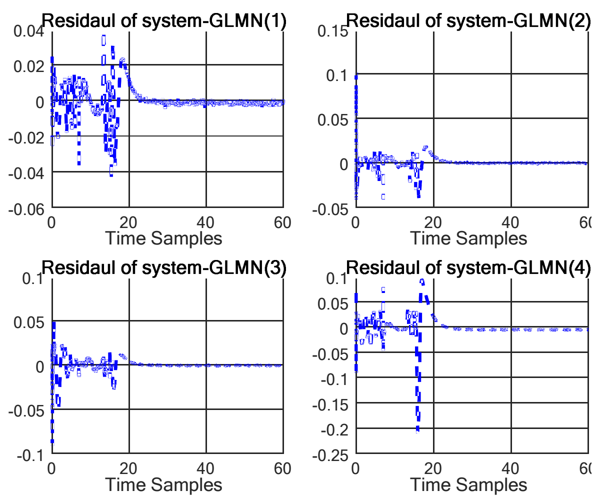

Fig. 8: Residuals of all four G-LMNs by applying the fault2

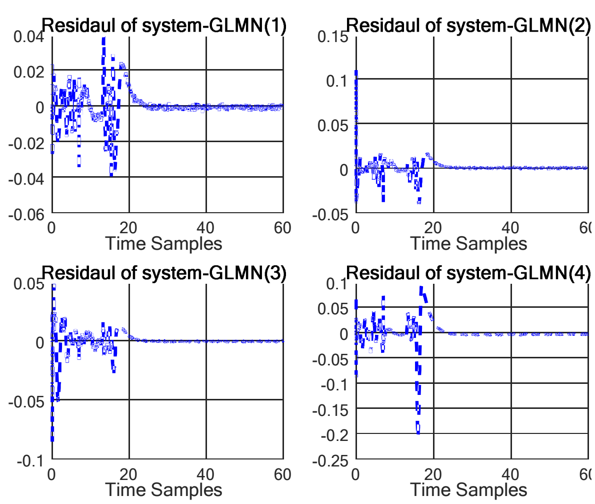

Fig. 9: Residuals of all four G-LMNs by applying the fault3

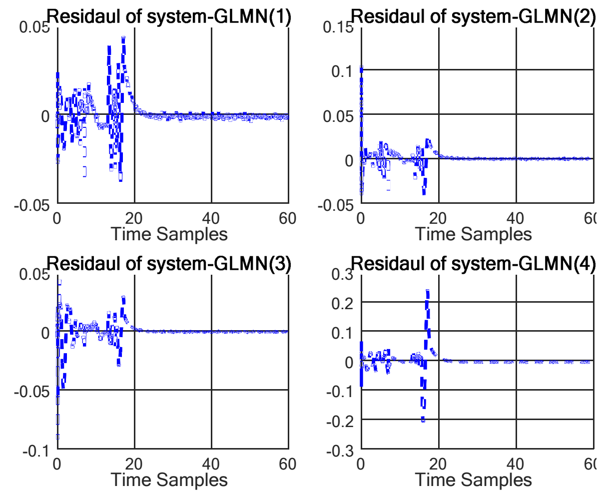

Fig. 10: Residuals of all four G-LMNs by applying the fault4

TABLE IV

FAULT DETECTION RESULTS

		GLOLIMOT		LOLIMOT	
	Fault inception time(s)	Detection time(s)	False alarm rate(%)	Detection time(s)	False alarm rate(%)
*Fault*1	15	15.60	10.42	16.12	22.73
*Fault*2	15	15.60	12.83	16.12	19.86
*Fault*3	15	15.60	8.1	16.12	17.2
*Fault*4	15	15.60	6.5	16.12	14.47

Table IV shows that the detection time and false alarm rate with GLOLIMOT are lower than LOLIMOT. The cause of improvement is more accurate identification with GLOLIMOT.

B. Fault isolation

Different variations and signature (S_i) in the residuals in Figs. 7-10 are useful to identify and isolate faults. Table V shows these signatures. The signs $\uparrow$ and $\downarrow$ mean that the residual is non-zero and oriented towards the upper or lower threshold. In other words, the sign $-$ indicates that the residual is close to zero. This table shows that any fault will make a unique signature in the residuals. So fault isolation was done with taking these sign into consideration. Hence, the fault isolation percentile is shown in a confusion matrix in table VI for GLOLIMOT and table VII for LOLIMOT for the test phase.

TABLE V

SI BASE UNITS

	*Fault*1	*Fault*2	*Fault*3	*Fault*4
$S1(signature\ of\ R1)$	↓	–	↑	↓
$S2(signature\ of\ R2)$	–	↑	↑	↑
$S3(signature\ of\ R3)$	↓	–	↑	–
$S4(signature\ of\ R4)$	↓	↓	–	↓

TABLE VI
CONFUSION MATRIX FOR FAULT ISOLATION (%) - GLOLIMOT

		Predicted				
		$Fault1$	$Fault2$	$Fault3$	$Fault4$	No class
Actual	$Fault1$	**80.21**	1.8	17.41	0	0.57
	$Fault2$	4.9	**81.11**	0	12.28	1.70
	$Fault3$	0.51	0	**97.3**	1.14	1.04
	$Fault4$	0	0.2	1.7	**97.49**	0.60

TABLE VII
CONFUSION MATRIX FOR FAULT ISOLATION (%) - LOLIMOT

		Predicted				
		$Fault1$	$Fault2$	$Fault3$	$Fault4$	No class
Actual	$Fault1$	60.24	18.35	14.47	5.31	1.62
	$Fault2$	12.41	61.72	8.45	15.26	2.15
	$Fault3$	7.18	5.38	75.62	10.14	1.67
	$Fault4$	3.36	2.59	5.91	86.84	3.29

By comparing tables VI and VII, the higher performance of GLOLIMOT than the LOLIMOT is clear. Table VI illustrates that almost all classes of conditions are separated with acceptable accuracy. Also, the percentage of non-class is located in the last column. Misdiagnosis or non-recognition of a class may be due to closeness or subscription of residual signs.

V. CONCLUSIONS

In this paper, an extended Local Linear Model (LMN) structure is presented that is useful to identify the high dimensional system. Hence, the space of the linear model differs from the splitting space (nonlinear part) in each local model. In other words, the size of splitting space is reduced by a matrix. Also, an incremental algorithm is proposed for creating the LMN and optimizing the matrix. This algorithm is based on two algorithms as Genetic algorithm and LOcal LInear MOdel Tree which has named GLOLIMOT.

We applied our algorithm to the gas turbine which is a system with high complexity and high dimensionality. This paper has attempted to Fault Detection and Identification (FDI) of this system based on the proposed structure and its algorithm. It has been shown that applying the proposed idea not only reduces both the number of local models and the number of parameter of each local model, also greatly improves the training accuracy.

The reasons for the improvement of our performance is the reduction of search space and project the input space to the new optimal space by GA. On the other hand, use of all information (all features) to model in local linear models, helps to more accurate identification. In continue, by comparing the simulation results of the proposed idea and conventional LMN, it can be said that FDI with the proposed structure is more sensitive to the faults. Since this method is offered in the presence of noise, it can be used in industrial gas turbine software applications, especially when measurements are unreliable due to noise and uncertainties.

REFERENCES

[1] T. Fischer, B. Hartmann, and O. Nelles, "Increasing the Performance of a Training Algorithm for Local Model Networks," in *World Congress of Engineering and Computer Science (WCECS). San Francisco, USA*, 2012.

[2] A. A. Adeniran and S. El Ferik, "Modeling and Identification of Nonlinear Systems: A Review of the Multimodel Approach--Part 1," 2016.

[3] O. Nelles, S. Sinsel, and R. Isermann, "Local basis function networks for identification of a turbocharger," in *Control'96, UKACC International Conference on (Conf. Publ. No. 427)*, 1996, pp. 7-12.

[4] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*: Springer Science & Business Media, 2013.

[5] T. A. Johansen and B. A. Foss, "Identification of non-linear system structure and parameters using regime decomposition," *Automatica,* vol. 31, pp. 321-326, 1995.

[6] O. Bänfer and O. Nelles, "Polynomial model tree (POLYMOT)—A new training algorithm for local model networks with higher degree polynomials," in *2009 IEEE International Conference on Control and Automation*, 2009, pp. 1571-1576.

[7] M. A. Nekoui and S. M. Sajadifar, "Nonlinear System Identification using Locally Linear Model Tree and Particle Swarm Optimization," in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, 2006, pp. 1563-1568.

[8] R. Mehran, A. Fatehi, C. Lucas, and B. N. Araabi, "Particle swarm extension to LOLIMOT," in *Sixth International Conference on Intelligent Systems Design and Applications*, 2006, pp. 969-974.

[9] J. Rezaie, B. Moshiri, A. Rafati, and B. N. Araabi, "Modified LOLIMOT algorithm for nonlinear centralized Kalman filtering fusion," in *Information Fusion, 2007 10th International Conference on*, 2007, pp. 1-8.

[10] S. Jakubek and C. Hametner, "Identification of neurofuzzy models using GTLS parameter estimation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 39, pp. 1121-1133, 2009.

[11] S. Jakubek and N. Keuth, "A local neuro-fuzzy network for high-dimensional models and optimization," *Engineering applications of artificial intelligence,* vol. 19, pp. 705-717, 2006.

[12] A. Sarabi-Jamab and B. N. Araabi, "PiLiMoT: A Modified Combination of LoLiMoT and PLN Learning Algorithms for Local Linear Neurofuzzy Modeling," *Journal of Control Science and Engineering,* vol. 2011, 2011.

[13] A. S. Jamab and B. N. Araabi, "A learning algorithm for local linear neuro-fuzzy models with self-construction through merge & split," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, 2006, pp. 1-6.

[14] S. M. E. Oliaee, M. A. Shoorehdeli, and M. Teshnehlab, "Faults detecting of high-dimension gas turbine by stacking DNN and LLM," in *Fuzzy and Intelligent Systems (CFIS), 2018 6th Iranian Joint Congress on*, 2018, pp. 142-145.

[15] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," *Artificial Intelligence Review,* pp. 1-31, 2018.
[16] A. Doroshenko, "Piecewise-Linear Approach to Classification Based on Geometrical Transformation Model for Imbalanced Dataset," in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, 2018, pp. 231-235.
[17] H. Azimi, S. Shabanlou, I. Ebtehaj, H. Bonakdari, and S. Kardar, "Combination of computational fluid dynamics, adaptive neuro-fuzzy inference system, and genetic algorithm for predicting discharge coefficient of rectangular side orifices," *Journal of Irrigation and Drainage Engineering,* vol. 143, p. 04017015, 2017.
[18] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença Jr, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications,* vol. 92, pp. 390-402, 2018.
[19] O. F. Lutfy, S. B. M. Noor, and M. H. Marhaban, "A simplified adaptive neuro-fuzzy inference system (ANFIS) controller trained by genetic algorithm to control nonlinear multi-input multi-output systems," *Scientific Research and Essays,* vol. 6, pp. 6475-6486, 2011.
[20] L. Breiman, "Hinging hyperplanes for regression, classification, and function approximation," *IEEE Transactions on Information Theory,* vol. 39, pp. 999-1013, 1993.
[21] S. Ernst, "Hinging hyperplane trees for approximation and identification," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, 1998, pp. 1266-1271.
[22] B. Hartmann, T. Ebert, T. Fischer, J. Belz, G. Kampmann, and O. Nelles, "LMNTOOL–Toolbox zum automatischen Trainieren lokaler Modellnetze," in *Proceedings of the 22. Workshop Computational Intelligence (Hoffmann, F.; Hüllermeier, E., Hg.), S*, 2014, pp. 341-355.
[23] O. Nelles, "Axes-oblique partitioning strategies for local model networks," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, 2006, pp. 2378-2383.
[24] B. Hartmann and O. Nelles, "On the smoothness in local model networks," in *American Control Conference (ACC), St. Louis, USA (June 2009)*, 2009.
[25] B. Hartmann and O. Nelles, "Structure trade-off strategy for local model networks," in *Control Applications (CCA), 2012 IEEE International Conference on*, 2012, pp. 451-456.
[26] J. Xu, X. Huang, and S. Wang, "Adaptive hinging hyperplanes and its applications in dynamic system identification," *Automatica,* vol. 45, pp. 2325-2332, 2009.
[27] S. Simani and C. Fantuzzi, "Dynamic system identification and model-based fault diagnosis of an industrial gas turbine prototype," *Mechatronics,* vol. 16, pp. 341-363, 2006.
[28] S. Simani, C. Fantuzzi, and R. Spina, "Application of a neural network in gas turbine control sensor fault detection," in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, 1998, pp. 182-186.
[29] V. Palade, R. J. Patton, F. J. Uppal, J. Quevedo, and S. Daley, "Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods," *IFAC Proceedings Volumes,* vol. 35, pp. 471-476, 2002.
[30] S. Simani, "Identification and fault diagnosis of a simulated model of an industrial gas turbine," *IEEE Transactions on Industrial Informatics,* vol. 1, pp. 202-216, 2005.
[31] H. A. Nozari, M. A. Shoorehdeli, S. Simani, and H. D. Banadaki, "Model-based robust fault detection and isolation of an industrial gas turbine prototype using soft computing techniques," *Neurocomputing,* vol. 91, pp. 29-47, 2012.
[32] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*: Springer, 2001.
[33] T. A. Johansen and R. Murray-Smith, "The operating regime approach to nonlinear modelling and control," *Multiple model approaches to modelling and control,* vol. 1, pp. 3-72, 1997.
[34] V. Kecman and B. Pfeiffer, "Exploiting the structural equivalence of learning fuzzy systems and radial basis function neural networks," in *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing EUFIT-94, Aachen, Gemany*, 1994, pp. 58-66.
[35] S. K. Halgamuge, *Advanced methods for fusion of fuzzy systems and neural networks in intelligent data processing*: VDI Verlag, 1996.
[36] O. Nelles and B. Hartmann, "Structure Trade-off Strategy for Local Model Networks," in *IEEE International Conference on Control Applications (CCA)*, Dubrovnik, Croatia, 2012, pp. 451-456.
[37] S. Simani and R. J. Patton, "Fault diagnosis of an industrial gas turbine prototype using a system identification approach," *Control Engineering Practice,* vol. 16, pp. 769-786, 2008.
[38] S. Simani, C. Fantuzzi, and R. J. Patton, "Model-Based Fault Diagnosis Techniques," in *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, ed: Springer, 2003, pp. 19-60.

Seyed-Mohamad-Emad Oliaee received his B.Sc. degree in Electrical Engineering from Hamedan University of Technology (HUT), Hamedan, Iran, in 2012, M.Sc. degree in Control System Design from K. N. Toosi University of Technology, Tehran, Iran in 2014. He is currently a PhD candidate in Control System Design at K. N. Toosi University of Technology, Tehran, Iran. His research interests include Systems Identification, Fault Detection and Identification, Intelligent Systems, Fuzzy Systems, Neural Networks, Neuro-Fuzzy Networks, Hybrid Control based on Classic and Intelligent systems, Deep Neural Networks, High-Dimensional Systems, Gas Turbine.

Mohammad Teshnehlab is currently a professor of control system in the Control Engineering Department, Faculty of Electrical and Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran. He received his B.Sc. degree in Electrical Engineering from Stony Brook university, New York, USA in 1980, master in Electrical Engineering from Oita university, Oita, Japan in 1991 and Ph.D in computational intelligence from Saga university, Saga, Japan in 1994. His main research interests are neural networks, fuzzy systems, evolutionary algorithms, swarm intelligent, fuzzyneural networks, pattern recognition, Meta heuristic algorithms, intelligent identification, deep neural network, prediction and control.

Mahdi Aliyari Shoorehdeli received his B.Sc. degree in Electrical Engineering in 2001, M.Sc. degree in Control System Design in 2003, Ph.D. degree in Control System Design in 2008 from K. N. Toosi University of Technology, Tehran, Iran. He is currently an Assistant Professor in the Department of Mechatronics Engineering of the K. N. Toosi University of Technology. His research interests include Fault Detection and Identification of Mechatronics Systems, Mechatronics Systems Identification, Intelligent Systems, Fuzzy, Neural Networks, Multi-Objective Optimisation, Hybrid Control based on Classic and Intelligent systems.

IECO

This page intentionally left blank.