Improved Flagging for Pattern Classifying Diagnostic Systems

Hsinyung Chin and Kourosh Danai

Abstract— **Fault detection and isolation (diagnosis) is based on residual generation and residual analysis. The model-based approach flags the residuals through thresholding, to isolate the effect of faults from noise, and performs diagnosis by mapping the residuals to a residual space with prespecified fault signatures. The main problem with this approach is that thresholds are not always able to differentiate between the effect of faults and noise, so this approach suffers from false alarms, undetected faults, and misdiagnosis. As an alternative to prespecified fault signatures and to cope with their variability, the use of pattern classification techniques has been proposed. However, since the fault signatures established by these classifiers are formed irrespective of diagnosability, this approach is also prone to misdiagnosis. In this paper we demonstrate the application of a flagging unit that enhances the quality of fault signatures. This Unit, which relies on a training set to tune its parameters, is shown to improve detection, reduce the number of false alarms and enhance diagnostics.**

I. Introduction

EFFICIENT detection and identification of faults is important for reliable performance of automated systems. Diagnostic systems can prevent damage to machinery by shutting down the process when necessary, or maintain productivity through corrective action when faults can be cured on-line. These systems can also reduce maintenance cost by eliminating routine disassembly, and can save inspection time by locating faulty components.

In general, fault diagnostic systems detect process abnormalities by monitoring the measurements and comparing them with their respective values during normal-mode operation [1]. Normal-mode values are determined analytically if the process can be represented by a mathematical model (model-based approach) [2,3], or are estimated empirically if such a model is not available [4]. In either case, the differences between the measurements and their normal-mode values, commonly referred to as residuals, are used as the basis for fault detection and isolation (see Fig. 1). In the model-based approach, the residuals are further mapped to a residual space with prespecified fault signatures so as to enhance diagnosability. That is, the residuals are transformed such that only distinct subsets of them are affected by individual faults ("structured residuals"), or that the residual vector is confined to specific directions for individual faults ("direction-fixed residuals") [5].

Residuals are usually contaminated with noise. As such, they are usually hard-limited to identify the effect of faults from noise. In this context, when a residual violates its threshold, a flag is posted to indicate abnormality, otherwise, the measurement is assumed to be normal. The problem with fault detection through hard-limiting is that thresholds cannot always distinguish the effect of faults from noise, so they result in false alarms, undetected faults (missed detection) and misdiagnosis [5]. False alarms occur when flags are unnecessarily posted, faults remain undetected when fault-affected measurements are not flagged, and misdiagnosis occurs when due to inefficient flagging the flagged residual vector does not match the correct fault signature in the diagnostic model.

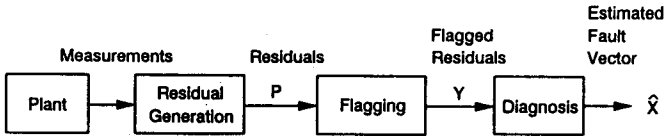


Fig. 1. Schematic of a fault diagnostic system.

In an attempt to cope with unreliable flagging, which precludes the use of prespecified fault signatures, pattern classification techniques have been employed to establish fault signatures through training. Pattern classifying diagnostic systems either define fault signatures in terms of the statistical properties of flagged residuals (e.g., Bayes method), or utilize nonparametric methods to estimate fault signatures independent of their statistical properties [6]. Although pattern classification improves diagnostics over the prespecified fault signature approach, it still cannot fully cope with the variability of fault signatures caused by inefficient flagging.

In this paper, we demonstrate the application of a flagging method [7] that improves the performance of pattern classifying diagnostic systems. A flagging unit is designed for this purpose so as to minimize false alarms, enhance detection, and stabilize fault signatures. This flagging unit consists of two hard-limiters and a quantization matrix. During a training session, the hard-limiters are adjusted to minimize the number of false alarms and undetected faults, and the components of the quantization matrix are tuned to enhance the uniqueness of fault signatures and reduce their variability. The fault signatures are estimated by the nonparametric pattern classification method MVIM [8], which has the capacity to quantify the uniqueness and variability of fault signatures after each training iteration (epoch).

The effectiveness of the flagging unit is demonstrated in simulation. For this purpose, two sets of data are generated to simulate the effect of faults on measurements as well as the presence of noise. One set is used for training the flagging unit,

Manuscript received January 9, 1992; revised August 5, 1992. This work is supported in part by the NSF Grants DDM-9015644 and MSS-9102149.

The authors are with the Department of Mechanical Engineering, University of Massachusetts, Amherst, MA 01003.

IEEE Log Number 9207524.

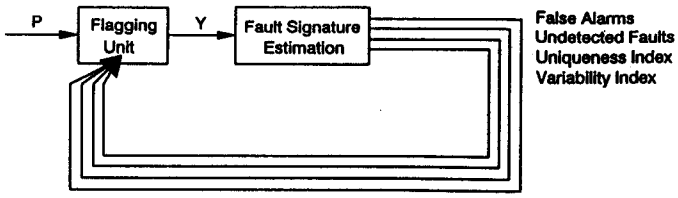


Fig. 2. Iterative tuning of the flagging unit based on feedback from its diagnostic model.

and to develop diagnostic models with different pattern classification techniques (MVIM, Bayes, and multilayer neural nets). The other set is used to test these models in diagnosis. The results indicate that the proposed flagging method improves diagnostics over the traditionally used method of flagging through simple hard-limiting, as well as over the inherent hard-limiting of nets performed through their activation function.

II. Flagging Method

The flagging unit [7] uses a sample set of measurement-fault vectors to tune its parameters iteratively. After each iteration through the training batch, the flagging unit counts the total number of false alarms and undetected faults, and estimates the signature of individual faults with a nonparametric pattern classification method. It then estimates the uniqueness and variability of the estimated fault signatures, so that it can use them as feedback in the next adaptation round (see Fig. 2). Adaptation stops when the total number of false alarms and undetected faults are minimized, and the uniqueness and consistency of fault signatures are enhanced.

A. Fault Signature Representation

Before the structure of the flagging unit is described, the basis upon which the uniqueness and variability of fault signatures are evaluated needs to be defined. Let us assume that fault signatures are represented by the n unit length columns $\bar{V}_j \in \mathcal{R}^m$ of a multi-valued influence matrix (MVIM) $\bar{A}$:

$$\bar{A} = [\bar{V}_1 \cdots \bar{V}_j \cdots \bar{V}_n] \tag{1}$$

such that m denotes the number of measurements, and n represents the number of faults. (We will explain later how the influence vectors $\bar{V}_j$ are estimated.) Based upon this influence matrix, the faults can be ranked according to their possibility of occurrence by the closeness of their influence vector to the vector of flagged measurements Y (see Fig. 3). Accordingly, the vector of *diagnostic certainty measures* which ranks the faults according to their possibility of occurrence is defined as

$$\begin{aligned}\hat{X} &= \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_j, \cdots, \hat{x}_n\}^T \\ &= \cos\{\alpha_1, \alpha_2, \cdots, \alpha_j, \cdots, \alpha_n\}^T\end{aligned} \tag{2}$$

where the $\hat{x}_j$ represent the individual *diagnostic certainty measures*, and α_j denote the individual angles between the influence vectors V_j and the flagged measurement vector Y (see Fig. 3). The angles α_j can be defined by their direction cosines as

$$\hat{x}_j = \cos\alpha_j = \bar{V}_j^T \bar{Y} \tag{3}$$

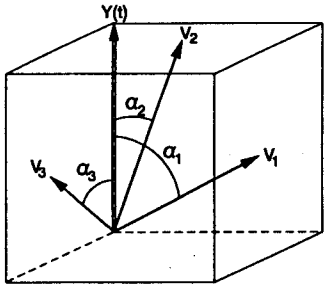


Fig. 3. Schematic of diagnostic reasoning in the MVIM method, illustrated in three-dimensional space.

where $\bar{Y}$ is the normalized form of vector Y, defined as

$$\bar{Y} = \frac{Y}{\|Y\|} = \left\{\frac{y_i}{\sqrt{\sum_{i=1}^m y_i^2}}\right\}. \tag{4}$$

Note that the diagnostic certainty measures of (2) are linear discriminant functions [6] of the form

$$\hat{x}_j \equiv g_j(Y) = \sum_{i=1}^m \bar{a}_{ij}\bar{y}_i \tag{5}$$

where the $\bar{y}_i$ denote the individual normalized flagged measurements (components of the normalized flagged measurement vector $\bar{Y}$), and $\bar{a}_{ij}$ represent the components of the influence vectors $\bar{V}_j$.

The influence vectors defined in (1) are not known *a priori* and need to be estimated. In the MVIM method, the error in diagnosis is used as the basis to estimate/update the influence vectors. For this purpose, the fault signatures are updated recursively after the occurrence of each fault to minimize the sum of the squared diagnostic error associated with individual faults:

$$\begin{aligned}E_j &= \sum_{k_j=1}^{N_j} e_j^2(k_j) \\ &= \sum_{k_j=1}^{N_j} [x_j(k_j) - \hat{x}_j(k_j - 1)]^2\end{aligned} \tag{6}$$

where $x_j(k_j)$ is always equal to 1 denoting the "ideal" value of the diagnostic certainty measure for the jth fault, and $\hat{x}_j(k_j - 1)$ represents the estimated value of the diagnostic certainty measure based on the current value of the influence vector $\hat{\bar{V}}_j(k_j - 1)$. Based on the above learning strategy, recursive least-squares (RLS) estimation [9] can be employed to adjust the orientation of individual influence vectors after the occurrence of each fault so as to improve diagnosis in the least-square sense. The proposed RLS algorithm as applied to the MVIM has the form

$$\begin{aligned}\hat{\bar{V}}_j(k_j) = \hat{\bar{V}}_j(k_j - 1) &+ \frac{G(k_j - 1)\bar{Y}(k_j)}{\lambda + \bar{Y}(k_j)^T G(k_j - 1)\bar{Y}(k_j)} \\ &\times [1 - \hat{x}_j(k_j - 1)]\end{aligned} \tag{7}$$

$$G(k_j) = \frac{1}{\lambda}\left[G(k_j - 1) - \frac{G(k_j - 1)\bar{Y}(k_j)\bar{Y}(k_j)^T G(k_j - 1)}{\lambda + \bar{Y}(k_j)^T G(k_j - 1)\bar{Y}(k_j)}\right] \tag{8}$$

where $k_j = 1, \cdots, N_j$ represents the number of occurrences

of the jth fault, $G(k_j)$ is the gain matrix, and λ is the "forgetting factor" [10] used to give more weight to the new data for time-varying systems. Note that the $\hat{\bar{V}}_j$ estimated by the above algorithm are not necessarily unit length. To enforce this condition, the $\hat{\bar{V}}_j$ are normalized after each iteration. The influence vectors estimated by the above learning algorithm comprise $\hat{\bar{A}}$.

B. Fault Signature Evaluation

One of the unique features of the MVIM method is its ability to evaluate quantitatively the uniqueness and variability of fault signatures, so that these quantitative measures can be used to tune the flagging unit (see Fig. 2). In the MVIM method, the uniqueness of fault signatures is characterized by the closeness of pairs of fault signatures. For this purpose, a diagnosability matrix $\mathcal{D}$ is defined to represent the closeness of the orientation of individual influence vectors as [8]

$$\mathcal{D} = \sin\left[\cos^{-1}\left(\hat{\bar{A}}^T\hat{\bar{A}}\right)\right]. \tag{9}$$

Matrix $\mathcal{D}$ in (9) is a 0 diagonal n-dimensional symmetric matrix representing the sine of the angles between pairs of influence vectors. The index of diagnosability d is defined as the smallest off-diagonal component of matrix $\mathcal{D}$ to denote the closest pair of fault signatures.

In the MVIM method, the variability of fault signatures is defined by their variance. For this purpose, the variance matrix associated with $\hat{\bar{A}}$ is estimated as [6]

$$\hat{\Sigma} = \left[1/N_j \sum_{k_j=1}^{N_j} (\bar{y}_i(k_j) - \hat{\bar{a}}_{ij}(k_j))^2\right] \tag{10}$$

to provide a measure of the variations of individual components of $\hat{\bar{A}}$. Since in the MVIM method the components of $\hat{\bar{A}}$ are adjusted recursively, matrix $\hat{\Sigma}$ can be readily estimated during training. In (10), the values of the normalized flagged measurements $\bar{y}_i$ at fault instances are perceived as the expected values of $\bar{a}_{ij}(k)$. The index of fault signature variability v in the MVIM method is defined as the largest component of matrix $\hat{\Sigma}$, to represent the largest variability in the components $\hat{\bar{a}}_{ij}$ of matrix $\hat{\bar{A}}$. Parameter N_j in (10) represents the total number of fault occurrences for the jth fault.

C. The flagging unit

The flagging unit processes the residuals as follows (see Fig. 4): The residual vector $\boldsymbol{P} \in \mathcal{R}^m$ (also shown in Fig. 2) is first passed through Hard-Limiter I (consisting of a vector of m thresholds) to produce a binary vector $\boldsymbol{Z} \in \mathcal{B}^m$. This vector is then multiplied sequentially by the normalized columns of the quantization matrix

$$\boldsymbol{Q} = [\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_i \cdots \boldsymbol{Q}_m] \tag{11}$$

and thresholded by Hard-Limiter II as

$$y_i = \begin{cases} 1 & \text{when } \boldsymbol{Z}^T\bar{\boldsymbol{Q}}_i \geq h_{2i} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

to produce the individual components of the flagged measurement vector $\boldsymbol{Y} \in \mathcal{B}^m$. The vectors $\bar{\boldsymbol{Q}}_i$ in (12) represent the normalized columns of the quantization matrix $\boldsymbol{Q}$ associated with individual measurements, and the h_{2i} represent the thresholds of Hard-Limiter II associated with the ith measurement. Note that the processing of residuals by the flagging unit is similar to residual processing in model-based methods – hard-limiting (by Hard-Limiters I and II) to produce "structured residuals," and residual transformation (by projecting $\boldsymbol{Z}$ on the individual columns of the quantization matrix) to produce "direction-fixed residuals" [5]—except for two major differences: 1) the flagging unit does not utilize prespecified thresholds or quantization vectors like model-based methods, and 2) it employs hard-limiting and residual transformation together, whereas in model-based methods these processes are commonly used independently. As will be explained later, the reason for combining these two types of residual processing is to accommodate the different types of learning necessary to reduce the number of false alarms and undetected faults, as well as to improve diagnosability and fault signature variability.

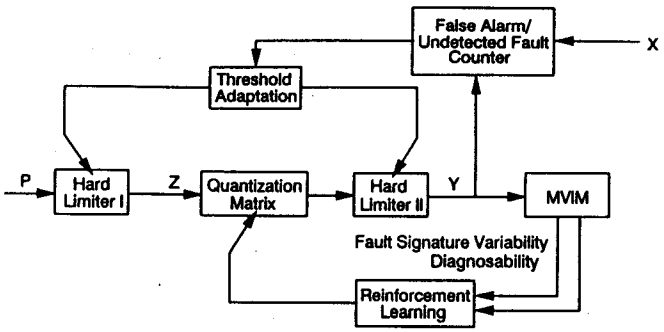


Fig 4. Schematic of the flagging unit.

The flagging unit is designed such that it can take full advantage of its feedback information during training (see Fig. 4). Ideally, the residuals should be flagged when a fault occurs and should be unflagged for false alarms. As such, the individual thresholds of Hard-Limiters I and II are adjusted to reduce the number of false alarms and undetected faults in the training batch. The adaptation algorithm used for these thresholds has the form

$$h_i(l+1) = h_i(l) + (\alpha f_i - \beta u)/N \tag{13}$$

where h_i denotes the threshold level (of Hard-Limiter I or II) corresponding to the ith measurement, f_i represents the number of false alarms in the batch corresponding to the ith measurement, u denotes the total number of undetected faults in the batch, l denotes the iteration round for the training batch, N represents the size of the training batch (number of measurement vectors used for adaptation), and α and β denote positive adaptation gains. The negative sign used in front of β indicates that fault undetectability is corrected by lowering all the threshold levels. Of course, it should be noted that the above algorithm alone does not guarantee the minimization of false alarms or undetected faults, as its convergence stops when an equilibrium point such as $\alpha f_i = \beta u$ for all measurements is reached. However, since the hard-limited residuals $\boldsymbol{Z}$ are subsequently passed through a quantization matrix whose columns are iteratively adapted to improve diagnosability and fault signature variability, there is always

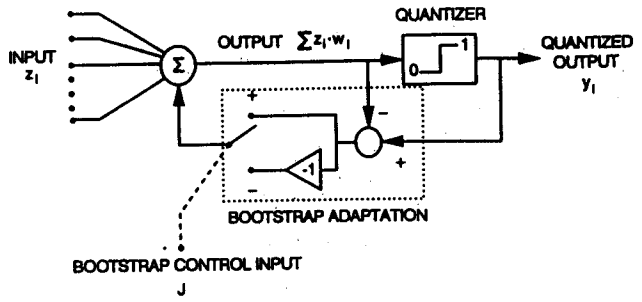


Fig. 5. Schematic of selective bootstrap adaptation.

a high probability that this equilibrium point will be disturbed and the thresholds will be adjusted further to reduce the number of false alarms and undetected faults.

In addition to reducing the number of false alarms and undetected faults, the flagging unit improves diagnosability and fault signature variability. For this purpose, it adapts iteratively the orientation of the quantization vectors $\boldsymbol{Q}_i$ through reinforcement learning, which uses a performance index J defined in terms of the process outputs to assess the relative success of the adaptation algorithm [11]. If J indicates success then the trend of adaptation is continued (reward), otherwise the trend is changed (punishment). The learning algorithm used for the quantization matrix is similar to Widrow *et al.*'s "selective bootstrap adaptation" [11] which uses a "critic" (bootstrap control input) to evaluate the relative success of the present decision with respect to the final results. In Widrow *et al.*'s algorithm (see Fig. 5), adaptation of weights w_i are adjusted according to the relationship,

$$\mathrm{w}_i(\mu) = \mathrm{w}_i(\mu-1) + \frac{\eta}{(n+1)}\Big[sgn(J)y_i(\mu-1) - \sum_{i=1}^{n+1} z_i(\mu-1)\mathrm{w}_i(\mu-1)\Big] \quad (14)$$

where $(n+1)$ denotes the total number of weights, η represents the learning rate, $sgn(J)$ is the signum of the boostrap control input J determining the sign of the quantized output y_i, and the error term inside the brackets represents the difference between the "bootstrapped" quantized output $y_i(\mu-1)$ and the output $\sum_{i=1}^{n+1} z_i(\mu-1)\mathrm{w}_i(\mu-1)$.

The learning algorithm used for the quantization matrix is similar to the algorithm in (14), except that 1) the term $sgn(J)$ is used outside the brackets to improve convergence [12], 2) the learning rate is reduced recursively to conform with adaptation gains in recursive least-squares estimation, and 3) the quantization vectors are nomalized so that $\boldsymbol{Z}^T\bar{\boldsymbol{Q}}_i$ represent the projection of vector $\boldsymbol{Z}$ onto the quantization vector $\boldsymbol{Q}_i$. This algorithm has the form

$$q_{ij}(\mu) = q_{ij}(\mu-1) + sgn(J)\, o_j(\mu-1) \times \left[y_i(\mu-1) - \boldsymbol{Z}^T(\mu)\bar{\boldsymbol{Q}}_i(\mu-1)\right]/N \quad (15)$$

where the q_{ij} denote the components of the quantization matrix, μ corresponds to the iteration step within the training batch, N represents the total number of measurement vectors in the batch (training batch size), and the o_j denote the components of the adaptation gain vector $\boldsymbol{O} \in \mathcal{R}^m$, updated

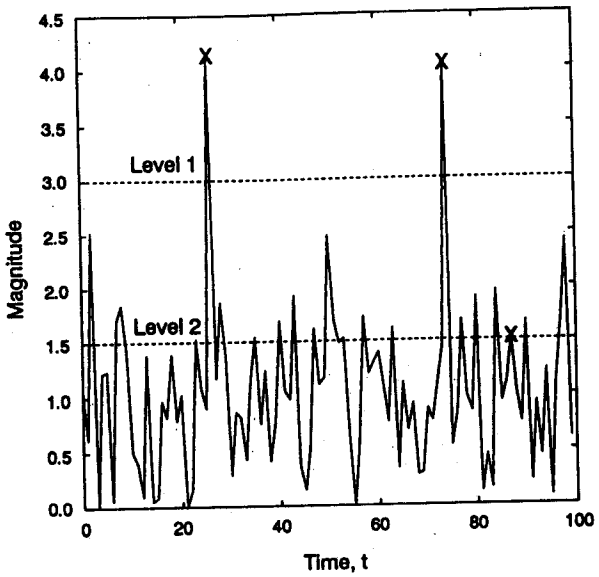


Fig. 6. Typical residuals produced through simulation (the × denote fault occurrences).

according to the relationship (Ljung, 1987)

$$\boldsymbol{O}(\mu) = \frac{\boldsymbol{R}(\mu-1)\boldsymbol{Z}^T(\mu)}{1+\boldsymbol{Z}(\mu)\boldsymbol{R}(\mu-1)\boldsymbol{Z}^T(\mu).} \quad (16)$$

The matrix $\boldsymbol{R}$ in (16) denotes the covariance matrix computed as

$$\boldsymbol{R}(\mu) = \boldsymbol{R}(\mu-1) - \boldsymbol{O}(\mu)\boldsymbol{Z}(\mu)\boldsymbol{R}(\mu-1). \quad (17)$$

Reward and punishment in this algorithm are defined as

$$sgn(J) = \begin{cases} + & d(l) > d(l-1), \text{ or } v(l) < v(l-1) \\ - & \text{otherwise} \end{cases} \quad (18)$$

where $d(l)$ and $v(l)$ are the diagnosability and fault signature variability indices provided by the MVIM after the lth iteration, respectively.

III. Simulation Results

The effectiveness of the proposed flagging unit in improving diagnostics was investigated in simulation. For this purpose, a system with four faults and six measurements was considered, and the residual vectors P (see Fig. 4) were simulated. In order to simulate the stochastic effect of faults on measurements, Bernoulli random numbers were generated according to class conditional probability densities included in matrix $\boldsymbol{B}$:

$$\boldsymbol{B} = \begin{bmatrix} 1.0 & 1.0 & 0.1 & 0.0 \\ 0.5 & 0.0 & 0.1 & 1.0 \\ 1.0 & 0.0 & 0.2 & 0.0 \\ 0.2 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.8 & 0.5 & 0.5 \\ 0.9 & 0.5 & 0.5 & 0.1 \end{bmatrix}. \quad (19)$$

Also, to simulate noise, uniformly distributed random numbers between 0 and 1 were added to these generated Bernoulli numbers. The obtained noise-contaminated residuals were thereafter scaled (multiplied by constants) to reflect reality.

TABLE I
DIAGNOSTIC RESULTS WHEN RESIDUALS WERE FLAGGED THROUGH SIMPLE HARD-LIMITING

Training Batch Size	Classifier	Correct Diagnosis	Misdiagnoses	False Alarms	Undetected Faults
10	MVIM	64	4	24	8
	Bayes	23	52	4	21
	Net (Flagged)	73	1	0	26
	Net (Unflagged)	76	9	7	8
20	MVIM	82	2	0	16
	Bayes	71	3	17	9
	Net (Flagged)	85	2	0	13
	Net (Unflagged)	79	5	1	15
30	MVIM	90	1	0	9
	Bayes	72	7	14	7
	Net (Flagged)	87	1	0	12
	Net (Unflagged)	87	5	2	6
40	MVIM	86	2	1	11
	Bayes	86	6	1	7
	Net (Flagged)	84	2	0	14
	Net (Unflagged)	88	6	4	2
50	MVIM	90	2	1	7
	Bayes	84	5	7	4
	Net (Flagged)	89	3	0	8
	Net (Unflagged)	92	5	0	3
100	MVIM	86	2	6	6
	Bayes	78	6	11	5
	Net (Flagged)	83	2	6	9
	Net (Unflagged)	93	5	1	1

A typical sample of residuals produced through simulation is presented in Fig. 6, which depicts a case where simple hard-limiting cannot reliably separate the effect of faults from noise. Note, for example, that a threshold set at Level 1 would not guarantee fault detection (i.e., no flag is posted when a fault occurs at t=87), whereas a threshold set at Level 2 would produce several false alarms.

A total of 100 sets of residuals were generated for training the flagging unit and estimating the fault signatures. Another 100 data sets were produced to test the fault signatures in diagnosis. As a first attempt, the residuals in the training batches were flagged by simple thresholds set at the mean plus one standard deviation of the residuals, and the fault signatures were estimated using three different classifiers: MVIM, Bayes, and a multilayer neural net using back-propagation learning. Note that each of the above classifiers in terms of the overall configuration of the diagnostic system in Fig. 2 represents the block "Diagnosis." For training the Bayes matrix, the individual class conditional probability densities b_{ij} in (19) were estimated using the maximum likelihood formula [6]

$$\hat{b}_{ij} = \frac{1}{N_j} \sum_{k_j=1}^{N_j} y_i(k_j). \tag{20}$$

where N_j denotes the total number of occurrences of the jth fault. For diagnostic purposes using Bayes classification, the set of discriminant functions $g_j(\boldsymbol{Y})$ associated with individual faults, which minimize the average probability of error for statistically independent y_i, were defined as

$$g_j(\boldsymbol{Y}) = P(x_j \mid \boldsymbol{Y}) = \prod_{i=1}^{m} \hat{b}_{ij}^{y_i}(1 - \hat{b}_{ij})^{1-y_i}. \tag{21}$$

The largest discriminant function in the above set was attributed to the occurred fault. During training, in order to establish the pattern of flagged measurements for no-fault cases, a fifth output was also considered to represent the no-fault case. Neural nets have an imbedded capability of flagging through their activation function (sigmoid function, in this case). In order to compare the quality of this imbedded flagging with hard-limiting and later with the results of the flagging unit, the net was also trained with residuals in their raw form (unflagged). The net used in all of the cases had 44 hidden units and was trained with the learning rate and momentum coefficient [13] set at 0.6 and 0.8, respectively. Note that a more suitable number of hidden units could perhaps provide better generalization and, consequently, better diagnostics. However, since the intention here was not to perform a comparative study between the above classifiers, but to investigate the effect of the flagging unit on their performance, we did not make any attempt to optimize the number of hidden units. For the neural nets, training was stopped when the mean square error associated with each output was less than 0.05.

The diagnostic results obtained from the test set reflecting the effect of training batch sizes on the performance of the

TABLE II
DIAGNOSTIC RESULTS WHEN RESIDUALS WERE FLAGGED WITH THE FLAGGING UNIT

Training Batch Size	Classifier	Correct Diagnosis	Misdiagnoses	False Alarms	Undetected Faults
10	MVIM	76	3	2	19
	Bayes	64	26	8	2
	Net (Flagged)	82	8	7	3
	Net (Unflagged)	76	9	7	8
20	MVIM	93	0	4	3
	Bayes	94	0	4	2
	Net (Flagged)	95	4	1	0
	Net (Unflagged)	79	5	1	15
30	MVIM	94	0	1	5
	Bayes	89	3	7	1
	Net (Flagged)	99	0	0	1
	Net (Unflagged)	87	5	2	6
40	MVIM	95	0	0	5
	Bayes	87	5	7	1
	Net (Flagged)	92	1	7	0
	Net (Unflagged)	88	6	4	2
50	MVIM	97	0	1	2
	Bayes	92	5	3	0
	Net (Flagged)	98	0	0	2
	Net (Unflagged)	92	5	0	3
100	MVIM	100	0	0	0
	Bayes	93	4	2	1
	Net (Flagged)	100	0	0	0
	Net (Unflagged)	93	5	1	1

above three classifiers are shown in Table I. In this table, *false alarms* denote cases where a fault was identified in a no-fault case, *undetected faults* represent fault incidents taken as a no-fault case, and *misdiagnoses* denote cases where faults were not correctly identified. The results in Table I indicate that the net trained with unflagged residuals produced better overall diagnosis than the net trained with hard-limited residuals. This suggests that the conversion of data into binary numbers limited the net's ability to classify the fault signatures, and that the net performed more effective "flagging" through its activation function. The results also indicate that the size of the training batch beyond 20 data sets did not have a pronounced effect on the performance of the classifiers. (This confirms the fact that large training sets do not necessarily improve generalization.)

The effectiveness of the flagging unit in improving the performance of the classifiers was investigated next. For this purpose, the flagging unit was tuned for each training set, with the initial thresholds of Hard-Limiter I set at the mean plus one standard deviation of the residuals, initial thresholds of Hard-Limiter II set at 0.5, and the initial quantization matrix set at identity. For all training cases, the values of α and β in (13) were both set at 1.0 for Hard-Limiter I and 0.5 for Hard-Limiter II. The convergence characteristic of the flagging unit during a tuning exercise with a training batch size of 20 is shown in Fig. 7. In all cases, the flagging unit was considered tuned when the number of false alarms and undetected faults converged to zero and the diagnosability and fault signature variability indices met the prespecified criteria of 0.7 and 0.1, respectively. For each case, the tuned flagging unit was then used to flag the residuals during both training and testing of any of the three classifiers. In these tests, the flagging unit performed the task of the block "Flagging" in terms of the diagnostic system of Fig. 1, and each of the classifiers MVIM, Bayes, and the neural net performed the task of the block "Diagnosis."

The diagnostic results produced by these newly trained classifiers are shown in Table II, along with the diagnostic results of the net trained with unflagged residuals. The results indicate that the flagging unit improved the performance of all of the classifiers considerably. For example, for a training batch size of 20 the average number of correct diagnosis improved over 30% for the Bayes classifier, that could not be achieved previously with any number of training data sets. The results also indicate that due to the flagging unit, the performance of the classifiers became much more competitive. Note that the number of *misdiagnoses*, *false alarms*, and *undetected faults* produced by the three classifiers has become quite comparable for all of the training batch sizes.

The diagnostic improvement provided by the flagging unit is a manifestation of the higher quality of fault signatures estimated by the classifiers. To demonstrate this point, let us consider the Bayes matrices estimated from a training batch size of 20 in Tables I and II. These estimated Bayes matrices are shown in Table III, along with the determinants of the

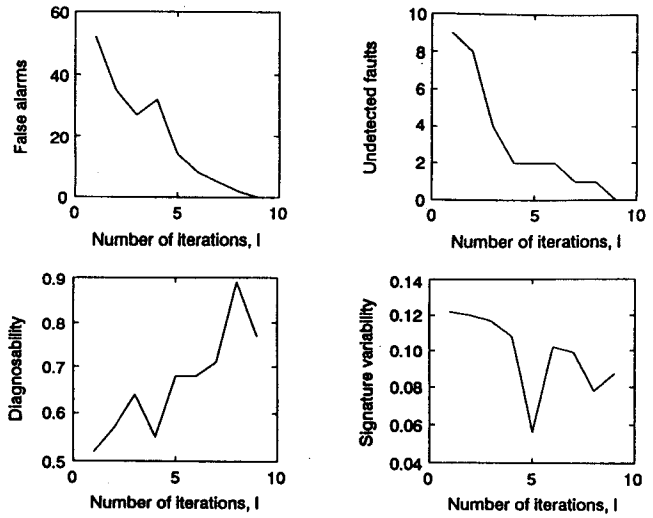


Fig. 7. Iterative elimination of false alarms and undetected faults, and improvement of diagnosability and fault signature variability by the flagging unit during adaptation.

TABLE III
ESTIMATED BAYES MATRICES AND THEIR COLINEARITY INDICES BEFORE AND AFTER THE APPLICATION OF THE FLAGGING UNIT, WITH A BATCH OF 20 TRAINING SETS

Estimated Bayes matrix before the application of the flagging unit	Estimated Bayes matrix after the application of the flagging unit
$\hat{\mathbf{B}}_b = \begin{bmatrix} 0.25 & 0.25 & 0.00 & 0.25 \\ 0.25 & 0.00 & 0.00 & 0.50 \\ 0.50 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.00 \\ 0.50 & 0.50 & 0.00 & 0.00 \\ 0.50 & 0.25 & 0.03 & 0.00 \end{bmatrix}$	$\hat{\mathbf{B}}_a = \begin{bmatrix} 0.50 & 0.50 & 0.00 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.50 \\ 0.50 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.00 \\ 0.50 & 0.50 & 0.00 & 0.75 \\ 0.50 & 0.25 & 0.00 & 0.00 \end{bmatrix}$
$\det(\hat{B}_b^T \hat{B}_b) = 0.0091$	$\det(\hat{B}_a^T \hat{B}_a) = 0.0281$

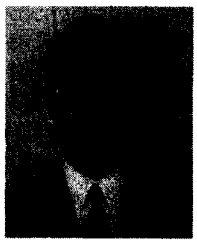
matrix $\hat{B}^T \hat{B}$ representing the degree of collinearity between the fault signatures [14]. The results indicate that the columns of the estimated Bayes matrix are less collinear when the residuals are flagged with the flagging unit. This implies that the improved diagnosis obtained with the flagging unit is a direct result of more distinct fault signatures estimated through the application of this unit.

IV. CONCLUSION

A flagging unit is introduced to enhance the performance of pattern classifying diagnostic systems. This unit operates in conjunction with the nonparametric pattern classifier MVIM which can evaluate the variability and uniqueness of fault signatures. Based on this evaluation of the fault signatures and the number of false alarms and undetected faults, which are obtained at the end of each training round, the flagging unit adjusts its parameters recursively to eliminate false alarms, ensure detection, and improve the uniqueness and variability of fault signatures. Simulation results indicate that with this flagging unit simpler classifiers such as the MVIM and Bayes can produce results comparable with a multilayer neural net. This suggests that improving flagging can be an effective method of enhancing diagnosis, perhaps as effective as adding to the complexity of decision regions.

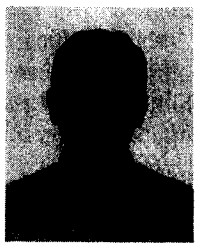
REFERENCES

[1] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, pp. 601–611, 1976.
[2] R. Isermann, "Process fault detection based on modeling and estimation methods—A survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.
[3] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Contr. Syst. Mag.*, pp. 3–11, Dec. 1988.
[4] R. A. Collacott, "Mechanical failure - diagnosis and monitoring," *CME*, pp. 63–69, July 1976.
[5] J. J. Gertler, "Analytical redundancy methods in fault detection and isolation," in *Proc. IFAC SAFEPROCESS Symp.*, (Baden-Baden, Germany), 1991.
[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[7] H. Chin and K. Danai, "A method of fault signature extraction for improved diagnosis," *ASME J. Dyn. Syst., Measurement, Contr.*, vol. 113, no. 4, pp. 634–638, 1991.
[8] K. Danai and H. Chin, "Fault diagnosis with process uncertainty," *ASME J. Dyn. Syst., Measurement Contr.*, vol. 113, no. 3, pp. 339–343, 1991.
[9] L. Ljung, *System Identification–Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[10] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[11] B. Widrow, N. K. Gupta, and S. Maitra, "Punish/reward: Learning with a critic in adaptive threshold systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 455–465, Sept. 1973.
[12] J. Hetrz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley, 1991.
[13] D. E. Hinton, G. E. Rumelhart and R. J. Williams, "Learning error representation by error propagation," in *Parallel Distributed Processing–Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: The MIT Press, 1988.
[14] J. O. Rawlings, *Applied Regression Analysis– A Research Tool*. Belmont, CA: Wadsworth, 1988.



Hsinyung Chin received the B.S. degree in mechanical engineering in 1984 from Tamkang University, Taiwan, and the M.S. degree in manufacturing engineering in 1990 and the Ph.D. degree in mechanical engineering from the University of Massachusetts, Amherst.

He is currently a post-doctoral research fellow in mechanical engineering at the University of Massachusetts, Amherst. His research interests involve: fault detection and diagnosis, parametric and nonparametric modeling, artificial neural networks, expert systems, instrumentation, and signal processing.



Kourosh Danai received the B.S., M.S., and Ph.D. degrees in mechanical engineering in 1980, 1982, and 1986, respectively, from the University of Michigan, Ann Arbor.

He joined the Department of Mechanical Engineering at the University of Massachusetts, Amherst in September 1987, where he is an Associate Professor. He has developed a method of fault diagnosis which has been successfully applied to tool breakage detection in machining, as well as to fault diagnosis of helicopter power transmissions. The latter was performed in collaboration with NASA and Sikorsky Aircraft Company.His research interests are automation with emphasis on modeling, system identification, and adaptive control.