15th Triennial World Congress, Barcelona, Spain

ASYMPTOTIC LOCAL APPROACH IN FAULT DETECTION WITH FAULTS MODELED BY NEUROFUZZY NETWORKS

Wang, Y. and Chan, C.W.

Department of Mechanical Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong, China
Email: mechan@hkucc.hku.hk   Fax: (852) 2859 7906

Abstract Since accurate models of nonlinear systems are difficult to obtain *a-priori* in practice, it is necessary to obtain these models from input-output data. As neurofuzzy networks can approximate nonlinear functions with arbitrary accuracy, and they can be trained from data, they are used here to model nonlinear systems. It is shown that the residuals generated from the model approximated by neurofuzzy networks is Guassian distributed, and that the asymptotic local approach can be applied to detect fault. Fault is detected when the residuals computed from the model exceeded a threshold determined by the χ^2-test for a given false alarm probability. The proposed fault detection procedure is demonstrated by an example.

Key words: fault diagnosis, neural network, residual, black-box model

1. INTRODUCTION

Fault diagnosis is crucial in monitoring industrial process, as demonstrated by the number of survey papers (Willsky, 1976, Gertler, 1988, Basseville, 1988, Frank, 1990, Isermann, 1993) and books (Patton *et al* 1989, Chen and Patton, 1999). A popular approach in fault detection is the model-based fault diagnosis methods derived based on the assumption that an accurate model of the system is available. By comparing the estimated output of the model and the actual output, faults are detected. This simple approach is adequate for large faults, but is difficult to detect smaller faults. To improve the reliability of fault detecting, asymptotic local approach is proposed such that fault detection for systems with known structure is reduced to statistical hypothesis test (Zhang, *et al*., 1998, Basseville, 1988).

However, model-based methods cannot be applied to systems that cannot be expressed explicitly by mathematical models. To overcome this difficulty, it is necessary to find a “universal” approximate model that can be used to represent most systems with arbitrary accuracy. Neural networks are such powerful tools. In input-output systems, neural networks are first trained to model faults in the system, and are then used to detect faults later (Patton, *et al*., 1994).

Wang *et al.* (2001) proposed to model the nonlinear system using a neurofuzzy network. The asymptotic local approach is then applied to detect fault from residuals generated from the neurofuzzy network. However, the threshold to detect fault has to be chosen empirically. In this paper, it is shown that the test is reduced to the χ^2-test, hence the threshold can be chosen for a given probability on the false alarm.

The paper is organized as follows. In Section 2, the model for different types of faults is presented, and the derivation of fault diagnosis of systems modeled by neurofuzzy networks using the asymptotic local approach is presented in Section 3. The proposed fault diagnosis procedure is demonstrated by a simulation example involving a nonlinear system.

2. NONLINEAR SYTEMS WITH FAULTS

Consider a general discrete single input single output (SISO) nonlinear system,

$y(t) = f(y(t\text{-}1), \ldots, y(t\text{-}n_y), u(t\text{-}d\text{-}1), \ldots, u(t\text{-}d\text{-}n_u))$ (1)

where u and y are the measured input and output, n_y and n_u are the maximum lags in the output and input, and d is the time delay, and $f(.)$ is an unknown smooth nonlinear function. Denote the input and the output by $x(t)$,

$$\begin{aligned} x(t) &= [x_1(t)\ \ x_2(t)\ \ \ldots\ \ x_n(t)]^T \\ &= [y(t\text{-}1)\ \ \ldots\ \ y(t\text{-}n_y)\ \ u(t\text{-}d\text{-}1)\ \ \ldots\ u(t\text{-}d\text{-}n_u)]^T \end{aligned}$$

where $n = n_y + n_y$ is the dimension of x. Neural networks with input $x(t)$ and output $y(t)$ are commonly used to approximate the nonlinear system (1). Neurofuzzy networks are used here to approximate $f(.)$, as they not only can approximate nonlinear functions with arbitrary accuracy, they can also be trained from data using linear parameter estimation techniques. Let there exist a neurofuzzy network that approximates closely $f(.)$, as given below (Brown and Harris, 1994),

$$y(t) = W_0^T \sigma_0(x) + e(t) \tag{2}$$

where W_0 is the weight vector, $\sigma_0(x)$ is the transformed input obtained by tensor product of the basis functions chosen as the membership functions of the fuzzy sets of x and $e(t)$ is a white noise with a variance of λ. As model (2) is not necessarily known *a-priori*, it is assumed that $y(t)$ is approximated by

$$y(t) = \overline{W}^T \sigma(x) + \xi(t) \tag{3}$$

where $\overline{W}$ is the weight vector, $\sigma(t)$ is the transformed input with a compatible dimension, and $\xi(t)$ is the sum of the modeling error and $e(t)$. If the dimensions of W is identical to the dimension of $\overline{W}$ then (3) is identical to (2). The output of the trained network (3) $\hat{y}(t)$ is,

$$\hat{y}(t) = \hat{W}^T \sigma(x) \tag{4}$$

where $\hat{W}$ is the estimated weight, and $\sigma(x)$ is the activation function vector. From (2) to (4), the modeling error $\varepsilon(t)$ is

$$\begin{aligned} \varepsilon(t) &= y(t) - \hat{y}(t) = W_0^T \sigma_0(x) - \hat{W}^T \sigma(x) + e(t) \\ &= W_0^T \sigma_0(x) - \overline{W}^T \sigma(x) + (\overline{W}^T - \hat{W}^T)\sigma(x) + e(t) \\ &= \Psi(x) + \tilde{W}\sigma(x) + e(t) \end{aligned} \tag{5}$$

where $\Psi(x) = W_0^T \sigma_0(x) - \overline{W}^T \sigma(x)$ and $\tilde{W} = \overline{W} - \hat{W}$. The modeling error given in (5) consists of three components: $e(t)$, the system noise, $\Psi(x)$, the modeling error for approximating the best neural network W_0, and $\tilde{W}\sigma(x)$, the estimation error of $\overline{W}$. If component fault or actuator fault occurs, the system becomes

$$y_f(t) = W_{0f}^T \sigma_0(x_f) + e(t) \tag{6}$$

where the subscript " f " denotes a system with faults. The output of the network when fault occurs is

$$\hat{y}_f(t) = \hat{W}^T \sigma(x_f) \tag{7}$$

The residual now becomes

$$\begin{aligned} \varepsilon(t) &= y_f(t) - \hat{y}_f(t) = W_{0f}^T \sigma_0(x_f) - \hat{W}^T \sigma(x_f) + e(t) \\ &= W_{0f}^T \sigma_0(x_f) - W_0^T \sigma_0(x_f) + W_0^T \sigma_0(x_f) - \overline{W}^T \sigma(x_f) \\ &\quad + (\overline{W}^T - \hat{W}^T)\sigma(x_f) + e(t) \\ &= \widehat{W}_0^T \sigma_0(x_f) + \Psi(x_f) + \tilde{W}^T \sigma(x_f) + e(t) \end{aligned} \tag{8}$$

where $\widehat{W}_0 = W_{0f} - W_0$. For sensor fault, the output of the system (2) is now

$$y_f(t) = y(t) + \Delta y(t) + e(t) = W_0^T \sigma_0(x) + \Delta y(t) + e(t) \tag{9}$$

where $\Delta y(k)$ arises from the sensor fault. As the neural network has not been updated, its output is still given by (7). From (5), (6) and (2), the residual is

$$\begin{aligned} \varepsilon(t) &= y_f(t) - \hat{y}_f(t) = W_0^T \sigma_0(x) - \hat{W}^T \sigma(x_f) + \Delta y(t) + e(t) \\ &= W_0^T \sigma_0(x) - \overline{W}^T \sigma(x) + (\overline{W}^T - \hat{W}^T)\sigma(x) + \hat{W}^T(\sigma(x) \\ &\quad - \sigma(x_f)) + \Delta y(t) + e(t) \\ &= \hat{W}^T(\sigma(x) - \sigma(x_f)) + \Delta y(t) + \Psi(x) + \tilde{W}\sigma(x) + e(t) \end{aligned} \tag{10}$$

It is shown in Wang *et al.* (2001) that it is difficult to detect small faults from the residual given by (5) or (10).

3. FAULT DIAGNOSIS BASED ON ASYMPTOTIC LOCAL APPROACH

The asymptotic local approach is a statistical technique that transforms a fault diagnosis problem into an asymptotically equivalent but simpler problem involving the detection of a change in the mean of a Gaussian process. It is shown by Zhang *et al.* (1998) that the approach can be applied to detect small and incipient faults in a class of nonlinear systems with known internal structure. In Wang *et al.* (2001), the method is extended to systems with unknown internal structure, modeled by neural networks. The asymptotic local approach is then used to develop the fault diagnosis technique, as presented below. A method to determine the threshold based on the χ^2-test is also presented.

3.1 Residual generation

Rewritten (2),

$$y(k) = W_0^T \sigma_0(x) + e(k) = \sum_{i=1}^{p} w_{0i} \sigma_{0i}(x) + e(k) \tag{11}$$

where $\sigma(x) = [\sigma_{01}(x) \quad \sigma_{02}(x) \quad \cdots \quad \sigma_{0p}(x)]^T$, $W_0^T = [w_{01} \quad w_{02} \quad \cdots \quad w_{0p}]$, and k is the sample number. Let $\sigma(x) = [\sigma_1(x) \quad \sigma_2(x) \quad \cdots \quad \sigma_m(x)]^T$ and $\overline{W} = [\overline{w}_1 \quad \overline{w}_2 \quad \cdots \quad \overline{w}_m]^T$, where m is the number of weights. For simplicity, let $m = p$. Then W_0 in (11) can be replaced by $\overline{W}$ and $\sigma_0(x)$ by $\sigma(x)$. Denote the estimate of $\overline{W}$ by $\hat{W}$. The output of the trained network is

$$\hat{y}(k) = \hat{W}^T \sigma(x) = \sum_{i=1}^{m} \hat{w}_i \sigma_i(x) \tag{12}$$

If m is chosen greater than p, then the estimated neural network is over-parameterized, but under-parameterized otherwise. Over-parameterization increases the computation burden, whilst under-parameterization increases the modeling errors. It is shown in the example presented in the next section that it is still possible to detect fault, though it may take longer.

Gradient descent algorithm is used to train the neural network (12), though other training algorithms can be used. At the kth training period, $\hat{w}_i^k$ is given by

$$\hat{w}_i^k = \hat{w}_i^{k-1} + \eta(y(k-1) - \hat{y}(k-1))\frac{\partial \hat{y}(k-1)}{\partial \hat{w}_i^{k-1}}\,;\ i = 1, \cdots, m \tag{13}$$

From equation (12), $\frac{\partial \hat{y}(k-1)}{\partial \hat{w}_i^{k-1}} = \sigma_i(x)$, hence equation (13) becomes

$$\hat{w}_i^k = \hat{w}_i^{k-1} + \eta h_i(k-1) \tag{14}$$

where $h_i(k) = \sigma_i(x)(y(k) - \hat{y}(k))$ is the residual and η is the learning rate. Define the residual vector, $H(k) = [h_1(k) \quad h_2(k) \quad \cdots \quad h_m(k)]^T$. After the network is trained using N input and output data, the residuals $H(k)$ are computed from the network. In the asymptotic local approach, the cumulative sum of the residual $D_M(\hat{W})$ computed for a window of M samples is normalized by $\sqrt{M}$, as given below (Benveniste *et al*, 1987).

$$\begin{aligned} D_M(\hat{W}) &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} H(k) \\ &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))(W_0^T \sigma_0(x(k)) - \hat{W}^T \sigma(x(k))) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))e(k) \\ &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))(\overline{W} - \hat{W}) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\Psi(x(k)) + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))e(k) \end{aligned} \tag{15}$$

Since the modeling error $\Psi(x)$ given by (5) is zero from the assumption that W_0 is replaced by $\overline{W}$ and $\sigma_0(x)$ by $\sigma(x)$, $\Psi(x)$ is ignored in the following analysis. The residual $D_M(\hat{W})$ can now be approximated by,

$$\begin{aligned} D_M(\hat{W}) &\approx \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))(\overline{W} - \hat{W}) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))e(k) \end{aligned} \tag{16}$$

It is well known that $\hat{W}$, the estimate of $\overline{W}$, is asymptotically normally distributed and approaches to $\overline{W}$ as M tends to infinity (Caines, 1988). Consequently, the first term on the right hand side of equation (16) is Gaussian distributed with zero mean. Since $e(k)$ is also Gaussian distributed, hence $D_M(\hat{W})$ is Gaussian distributed with zero mean.

When there is a component or an actuator fault, the system is now given by (6). Denote the optimal weight vector of the system with fault by $\overline{W}_f$, and ΔW_{ca} the change in the weight arising from the component or the actuator fault, then

$$\overline{W}_f - \overline{W} = \Delta W_{ca} \tag{17}$$

From (16) and (17), $D_M(\hat{W})$ becomes

$$\begin{aligned} D_M(\hat{W}) &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))(\overline{W}_f - \hat{W}) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))e(k) \\ &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))(\overline{W} - \hat{W}) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))e(k) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))\Delta W_{ca} \end{aligned} \tag{18}$$

Since ΔW_{ca} is non-zero, hence $D_M(\hat{W})$ is also non-zero, irrespective of a component or an actuator fault. For a sensor fault, the system is given by (9), and $D_M(\hat{W})$ is

$$\begin{aligned} D_M(\hat{W}) &= \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))\sigma^T(x(k))(\overline{W} - \hat{W}) \\ &\quad + \frac{1}{\sqrt{M}}\sum_{k=1}^{M} \sigma(x(k))(e(k) + \Delta y) \end{aligned} \tag{19}$$

Similarly, $D_M(\hat{W})$ is also non-zero if there is a sensor fault. An online fault detection scheme is derived in the following section using this result.

3.2 On-line fault detection procedure

After neurofuzzy network is trained off-line, the proposed fault detection procedure is given below.

Step 1 Select M, the window for computing the cumulative sum of residuals.

Step 2 Compute the mean of the residuals generated from the trained network as described below. This is a necessary step as $\Psi(x)$ is ignored, and also the mean of the residuals calculated from a finite number of samples may not be exactly zero.

$$b_0 = \frac{1}{N}\sum_{i=1}^{N} H(i)$$

Step 3 At the kth sampling period, compute the cumulative sum of residuals

$$D_M^k = \frac{1}{\sqrt{M}}\sum_{i=1}^{M}(H(i) - b_0) \qquad (20)$$

Because D_M^k is a vector, it is necessary to normalize the cumulative sum of residuals for detecting the fault,

$$S_M^k = (D_M^k)^T R(k)^{-1} D_M^k \qquad (21)$$

where $R(k)$ is the covariance matrix of $H(k)$.

Step 4 Fault is detected, if $S_M^k > \lambda$, where λ is the threshold. The selection of λ is discussed in the next subsection.

Step 5 Repeat Steps 3 and 4 for another sample if no fault has been detected.

3.3 Determination of threshold λ

The proposed procedure depends on the choice of λ. It is shown previously that for large *k*, fault detection is equivalent to the detection of a change in the mean of a Gaussian vector. It is well known that the detection of the change in the mean of a Gaussian vector can be formulated as a χ^2-test (Benveniste *et al*., 1987). Therefore, S_M^k given by (21) is χ^2 distributed as *M* tends to infinity (Caines, 1988). Therefore, the threshold λ can be chosen from the χ^2-table for a given false alarm probability (Zhang *et al*., 1998).

When computing the covariance matrix R_c given by (21), the following method is used to ensure that it is positive definite. Assume that $H(k)$, for k = 1, …, N are obtained when the system is under normal operating conditions. Divide the sequence $H(k)$ into L_1 segments of length L_2, then an estimate of R_c is

$$\hat{R}_c = \frac{1}{L_1}\sum_{i=1}^{L_1} D_{L_2,i} D_{L_2,i}^T \qquad (22)$$

where $D_{L_2,i} = \frac{1}{\sqrt{L_2}}\sum_{j=1}^{L_2} H(j + iL_2)$, and $L_1 L_2 = N$. For sufficiently large L_1, R_c is generally positive definite, and if L_2 is chosen sufficiently large, then $D_{L_2,i}$ is approximately Gaussian distributed. In general, there are a large number of choices of L_1 and L_2, such that R_c can be estimated accurately.

It is shown in the previous section that the dimension of $H(k)$ is the same as the number of weights in the neurofuzzy network. As the dimension of the neurofuzzy network, and hence the dimension of $H(k)$ can be very large, if too many B-spline functions are chosen for the network. In this case, R_c and hence its inverse are difficult to compute,. Therefore, the number of B-spline functions in the network should be kept as small as possible. Although the modeling error will be increased, fault can still be detected, as shown in the simulation example given later.

Remark 1: If the inverse of R_c does not exist or is difficult to compute, R_c is replaced by the norm $\sum_{i=1}^{M}\|H(i)\|^2$ in normalizing D_M^k (Wang *et al.*, 2001). However, the χ^2-test may not be appropriate in determining λ.

4. SIMULATION EXAMPLE

The following nonlinear system is simulated.

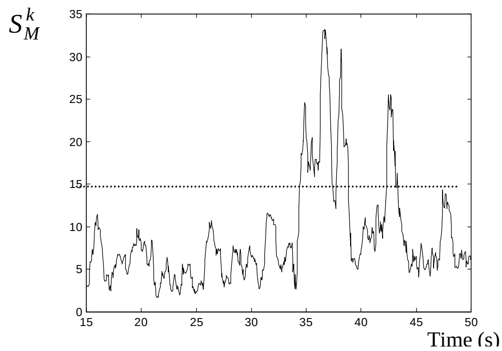
$$y(k) = \frac{ay(k-2)y(k-1)}{1 + y(k-1)^2 + y(k-2)^2} + b\cos\left(\frac{y(k-1) + y(k-2)}{2}\right) + cu(k-1) + e(k-1)$$

where $y(k)$, $u(k)$, and $e(k)$, are the output, the input and the white noise, and a = 2.5, b = 0.3 and c = 0.5. The variance of $e(k)$ is 0.2^2 and the sampling interval is 0.05*s*. The input of the neurofuzzy network is: ($u(k$-1), $y(k$-1), $y(k$-2)). Two triangular B-spline functions are chosen for each input, giving a total of eight weights. Although a small number of B-spline functions may lead to larger modeling errors, the covariance matrix R_c and its inversion can be computed more readily. From (23), 1000 input and output training data are generated with the input *u* randomly generated within the range of -2 and 2. After training, the proposed fault detection scheme described in Section 3.2 is applied to detect the following faults, which occurred separately one at time at 30*s*, (i) a_f = 2.2, (ii) b_f = 0.24, (iii) c_f = 0.45, and (iv) $y_f(k) = y_f(k) - 0.05$. Both (i) and (ii) represents component faults, (iii), an actuator fault, and (iv), a sensor fault. It is assumed the control *u* is given by, $u(t) = 1.2\sin(0.05\pi t)$.

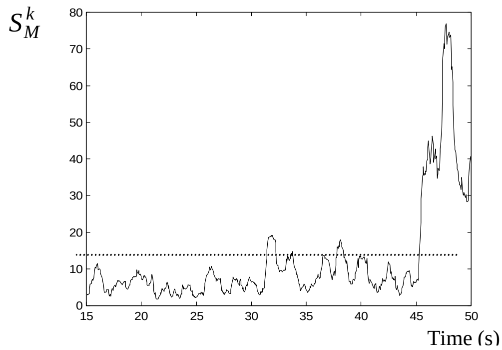
The residual $H(k)$ is obtained from the trained B-spline network after 15*s*. The proposed fault detection scheme is applied after the system has settled at t = 50s. For M = 100 (i.e., 5s), S_M^k for each fault are shown in Fig.1 (a) to (d), indicating it changed after the occurrence of each fault. The threshold λ determined from the χ^2-table for a 5% false alarm rate is 15.5073, as the dimension of $H(k)$ is eight. The time for the proposed fault detection scheme to detect each fault are 36.35s, 34.5s, 31.5s and 34.5s respectively, i.e., over 16 to 21s after each fault has occurred. To investigate the effect of M on the fault detection results, it is set respectively to 200 (i.e., 10s) and 300 (i.e., 15s). For lack of space, only S_M^k for M = 200 are plotted, as shown in Fig. 2 (a) to (d). Again, S_M^k changed after the fault has occurred. The time required to detect the faults for different M is shown in Table.1, illustrating that for large M longer time is required before faults are detected.

5. CONCLUSIONS

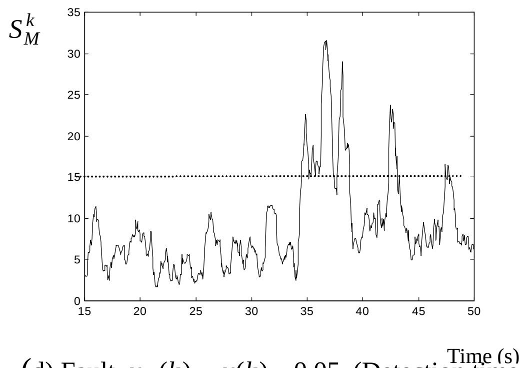
In this paper, a fault detection procedure for nonlinear systems based on the asymptotic local approach is derived. The system is modeled first by a neurofuzzy network, which is trained off-line. Residuals are then generated from the neurofuzzy network. Faults are detected if the normalized cumulative sum of the residuals exceeds a threshold determined by the χ^2-test for a given false alarm probability. The proposed procedure is demonstrated by a simulation example. It is also shown in the example that it requires longer time to detect the fault if M is large.



(b) Fault $b_f = 0.24$ (Detection time 34.5s)

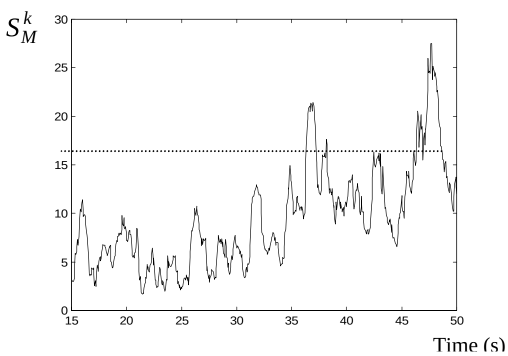


(c) Fault $c_f = 0.45$ (Detection time 31.5s)

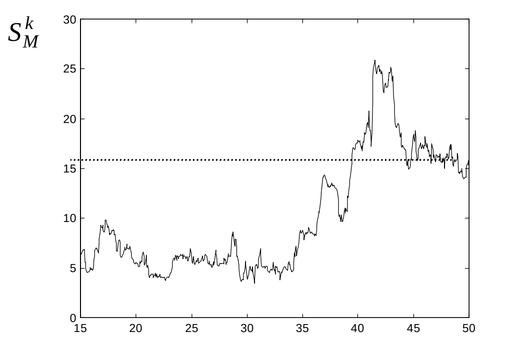


(d) Fault $y_f(k) = y(k) - 0.05$ (Detection time 34.5s)

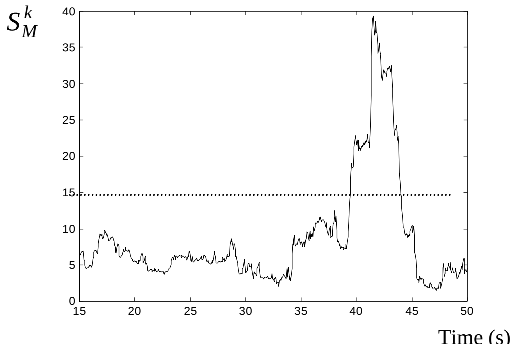
Fig. 1 Fault detection using the proposed scheme for $M = 100\,(5s)$



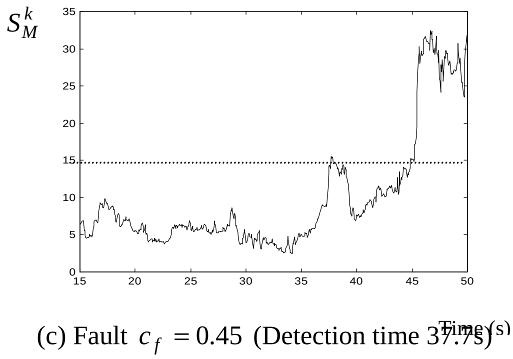
(a) Fault $a_f = 2.2$ (Detection time 36.35s)



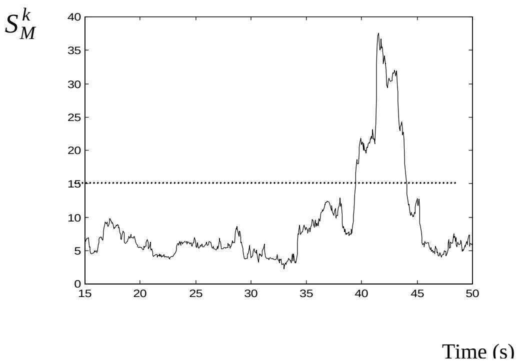
(a) Fault $a_f = 2.2$ (Detection time 39.4s)



(b) Fault $b_f = 0.24$ (Detection time 39.45s)



(c) Fault $c_f = 0.45$ (Detection time 37.7s)



(d) Fault $y_f(k) = y(k) - 0.05$ (Detection time 39.45s)

Fig. 2 Fault detection using the proposed scheme for $M = 200\,(10s)$

Table 1 Fault detection time with different M

	M =100	M = 200	M = 300
a_f = 2.2	36.35s	39.4s	39.85s
b_f = 0.24	34.5s	39.45s	41.25s
c_f = 0.45	31.5s	37.7s	43.65s
$y_f(k) = y(k)$ –0.05	34.5s	39.45s	40.95s

REFERENCES

Basseville, M., 1988, Detecting changes in signals and systems – A survey, *Automatica*, 24, 309-326.

Brown, M. and Harris, C. J.,1994, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall.

Caines, P. E., 1988, *Linear Stochastic Systems*, J. Wiley, New York.

Chen, J., Patton, R. J., 1999, *Robust Model-based Fault Diagnosis for Dynamical Systems*, Kluwer Academic Publishers, USA.

Frank, P. M., 1990, Fault diagnosis in dynamical systems using analytical and knowledge based redundancy – a survey and new results, *Automatica*, 26, 459-474.

Gertler, J. J., 1988, Survey of model based failure detection and isolation in complex plants, *IEEE Control Systems Mag.*, 8, 3-11.

Isermann, R., 1993, Fault diagnosis of machines via parameter estimation and knowledge processing – tutorial paper, *Automatica*, 29, 815-836.

Patton, R. J. Chen J. and Nielsen S. B., 1994, Fault diagnosis in nonlinear dynamical systems via neural networks, *Proc of IEE Int. Conf.: Control'94*, Warwick, UK, 1346-1351.

Patton, R. J., Frank, P. M. and R. Clark, 1989, *Fault Diagnosis in Dynamical Systems – Theory and Applicaition*, Prentice-Hall, London, UK.

Wang Y., Chan, C. W. and Cheung, K. C., 2001, On-line Fault Diagnosis Based On B-spline Neural Networks Using Asymptotic Local Approach, *Asian Journal of Control*, 3, 73-78.

Willsky, A. S., 1976, A survey of design methods for failure detection in dynamical systems, *Automatica*, 12, 601-611.

Zhang, Q, Basseville, M. and Benveniste, A., 1998, Fault detection and isolation in nonlinear dynamical systems: A combined input-output and local approach, *Automatica*, 34, 1359-1373