

Fault Detection of Redundant Systems Based on B-spline Neural Network

Hong Jin†, C. W. Chan*, H. Y. Zhang† and W. K. Yeung*

† Department of Automatic Control, Beijing University of Aeronautics and Astronautics
Beijing 100083, P. R. China, Email: hjin@ihw.com.cn; hoyzhang@public3.bta.net.cn

* Department of Mechanical Engineering, The University of Hong Kong
Hong Kong Pokfulam Road, Email: mechan@hkucc.hku.hk

Abstract: The fault detection and isolation of redundant sensor systems based on B-spline neural networks is presented in this paper. The network is trained using an algorithm with an adaptive learning rate. To further save computation time, the residual vector is transformed from a multivariate B-spline function to an univariate B-spline function. The detection of abrupt and drifting faults using the proposed method is discussed. The performance of the proposed method is illustrated by an example involving a redundant system consisting of six sensors.

Keywords: Fault detection, Redundant systems, Neural networks, B-spline functions.

1. Introduction

In many aerospace applications, reliability is a critical issue. To enhance reliability, redundant sensors are often used. For example, redundant gyrosensors arranged in a dodecahedron are common in navigation systems. In order that the reliability of inertial measurement systems with redundant sensors is significantly higher than that of systems with no redundant sensors, the redundant systems need to have a high degree of fault tolerance, which can be achieved passively using data selection techniques, or actively using fault detection and isolation (FDI) techniques [1]. Several FDI methods have been proposed, e.g., the well known Generalized Likelihood Test (GLT) [1], and the Optimal Parity Test (OPT). The OPT is more efficient in detecting and isolating faults as it is designed to be more sensitive to designated sensor faults, and less sensitive to other sensors faults in a redundant system [2][3]. However, a common assumption in both techniques is that the sensor mounting configuration is known [2]. Further, the statistical distribution of noise in the model is assumed known in order that the threshold to detect faults can be determined. In practice, mounting errors always exist, and these errors can reduce the reliability of these methods. To improve the reliability of fault detection in a redundant system, it is proposed in [4], that the estimate of the state should be compensated first before applying FDI techniques.

In recent studies of FDI, a popular approach is to use neural networks in fault detection, e.g., see [5,6]. In [5], multilayer feedforward neural networks are used, whilst in [6], associative B-spline networks are used to integrate quantitative and qualitative information. In this paper, an FDI technique based on the B-Spline neurofuzzy network (BSNN) is proposed. The main advantage of the BSNN is that computation time is much less than the multilayer feedforward neural networks, since only local adjustment are required [7].

2. Fault detection of systems with redundant sensors

Let the measurement of a system with redundant sensors be given by

$$M = Hx + \varepsilon \quad (1)$$

where $x \in R^n$ is the state vector, H , the measurement matrix for given a sensor configuration, $M \in R^m$, the measurement vector for m sensors, and $m > n$, ε , the zero mean noise with a covariance matrix $\sigma^2 I_m$, and σ , a positive constant, I_m , the $m \times m$ identity matrix. To determine the threshold in fault detection, ε is assumed to be a Gaussian noise,

$$\varepsilon \sim N(0, \sigma^2 I_m) \quad (2)$$

When there are sensor faults, the measurement equation (1) becomes

$$M = Hx + b + \varepsilon \quad (3)$$

where $b = (b_1, \dots, b_m)$ is the fault vector, b_i , the magnitude of the fault of the i th sensor. Let H be full column rank, i.e., $\text{rank}(H) = n$, then there exists an $m \times (m-n)$ matrix V of full column rank satisfying the orthogonal condition:

$$V'H = 0 \quad (4)$$

Let each column vector of V be defined as a parity vector, and the space spanned by the columns of V be the parity space [8]. The orthogonal projection matrix, P_V , on the parity space is given by,

$$P_V = V(V'V)^{-1}V' \quad (5)$$

A popular method to compute V is that proposed by Potter, and that V satisfies [1],

$$V'V = I \quad (6)$$

A brief discussion of the generalized likelihood test (GLT), and the optimal parity test in fault detection is presented here.

2.1 The generalized likelihood test [1,8]

For a given V , the measurement M is transformed yielding the residual vector

$$\rho = V'M \quad (7)$$

The fault detection decision function is given by

$$DF_D = \rho' Cov^{-1}(\rho)\rho = \sigma^{-2}\rho'(V'V)^{-1}\rho \quad (8)$$

The detection threshold T_D is chosen for a given probability of false alarm. Then

- (i) if $DF_D > T_D$, fault has occurred, or
- (ii) if $DF_D \leq T_D$, there are no faults.

2.2 The optimal parity test [3]

Let v_i be chosen such that it is more sensitive to faults in the i th sensor than that in the other sensors [3]. The optimal parity vectors for the other sensors are similarly chosen. The scaled residuals $\rho_i = v_i'M$, $i=1, \dots, m$ are computed, and normalized

$$\rho_i^* = \rho_i / (\sigma \|v_i\|) \quad (9)$$

such that the variance of ρ_i^* is 1. Select a detection threshold T for a given probability of false alarm. Determine the largest absolute value of the scale residual, ρ_k^* , $|\rho_k^*| = \max_i |\rho_i^*|$. Then

- (i) if $|\rho_k^*| > T$, a fault occurred in the k th sensor, or
- (ii) if $|\rho_k^*| \leq T$, there is no fault.

2.3 Remarks

The GLT and OPT are derived under two conditions:

- (i) σ is assumed known, as otherwise, DF_D in the GLT, and ρ_i^* in the OPT cannot be computed, and
- (ii) ε is assumed to be Gaussian in determining the thresholds T_D in the GLT, and T in the OPT.

Since these two conditions are restrictive, hence the application of these methods in practice can be limited. In this paper, BSNNs are proposed to improve the performance of these techniques. In the following discussion, it is assumed that ε is a zero mean noise with a covariance matrix $\sigma^2 I$, where σ is unknown, but satisfies $\sigma \in [a, b]$; where a and b are respectively the lower and upper bounds.

3. B-Spline neurofuzzy networks

For completeness, a brief discussion of the BSNN is given here, and the reader is referred to [7] for further details. Under certain conditions, the output, $\hat{y}(t)$, of a BSNN shown in Fig. 1 is given by,

$$\hat{y} = \sum_{j=1}^q w_j s_j(X) \quad (10)$$

where $\{w_j, j=1, q\}$, are the weights, $X \in R^n$, the input vector, and $\{s_j(x), j=1, q\}$, the transformed of X using B-spline function.

3.1 Training of the BSNN

The weights, $\{w_j, j=1, q\}$, are obtained by minimizing the performance index,

$$E = \sum_{i=1}^L \|y(i) - \hat{y}(i)\|^2 \quad (11)$$

where $y(i)$ is the reference value, and L is the number of training samples. Since the weights are independent of the transformed input $\{s_j(x), j=1, q\}$, they can be obtained by the least square method. Let

$$S = \begin{pmatrix} s'(X(1)) \\ \vdots \\ s'(X(L)) \end{pmatrix} = \begin{pmatrix} s_1(X(1)) & \dots & s_q(X(1)) \\ \dots & \dots & \dots \\ s_1(X(L)) & \dots & s_q(X(L)) \end{pmatrix}$$

The estimate of the weights \hat{w} is given by

$$\hat{w} = (S'S)^{-1} S'Y \quad (12)$$

where $Y = (y(1), \dots, y(L))'$. If S is full column rank, then \hat{w} can be determined uniquely. However, if $S'S$ is singular, e.g., $q > L$, then some components of w cannot be determined uniquely, then \hat{w} can be computed from the pseudo-inverse of $S'S$. When q is large, the computation of the inverse or the pseudo-inverse of $S'S$ can be complex. In this case, the following numerical methods can be used.

3.1.1 Steepest decent gradient method

The weights are computed by

$$w(k) = w(k-1) + \eta S' \Delta \hat{Y}^{(k-1)} \quad (13)$$

where η is the learning rate, k , the training cycle, $\Delta \hat{Y}^{(k)} = Y - \hat{Y}^{(k)}$, and $\hat{Y}^{(k)} = (\hat{y}^{(k)}(1) \dots \hat{y}^{(k)}(L))'$. A main problem here is the choice of η . For a small η , the convergence of the weights is slow, though the learning algorithm is more stable. However, increasing the learning rate by increasing η can lead to an unstable learning algorithm.

3.1.2 Adaptive learning rate

An adaptive learning rate (ALR) is proposed in [9], which is obtained by maximizing the reduction in the cost function E at each iteration, and given below.

$$\hat{\eta}_j = \|S' \Delta \hat{Y}_j^{(k-1)}\|^2 / \|S' \Delta \hat{Y}_j^{(k-1)}\|_G^2 \quad (14)$$

where

$$\Delta \hat{Y}_j^{(k)} = Y_j - \hat{Y}_j^{(k)},$$

$$\hat{Y}_j^{(k)} = (\hat{y}_j^{(k)}(1) \dots \hat{y}_j^{(k)}(L))', \text{ and } G = S'S.$$

$$\hat{\eta}_j = \|S' \Delta \hat{Y}_j^{(k-1)}\|^2 / \|S' \Delta \hat{Y}_j^{(k-1)}\|_G^2$$

3.1.3 Normalized least mean square method

In the normalized least mean square method, the weights are updated as follows,

$$w(t+1) = w(t) + \mu [y(t) - s'(t)w(t)]s(t) \quad (15)$$

where $w(t)$, $y(t)$, and $s(t)$ are defined previously, and μ , the feedback coefficient. It is proposed in [10] to update μ recursively as follows.

$$\mu(t) = \frac{q\mu_0}{\|s(t)\|^2} \quad (16)$$

where q is the dimension of $s(t)$, $\mu_0 > 0$ is a constant. The maximum value of μ_0 is given by

$$\mu_0 < 2\|s(t)\|^2/q$$

4. Fault detection procedure

4.1 Training of the BSNN

For a given V with full column rank, there exists a nonsingular matrix U such that

$$V'V = UU' \quad (17)$$

Let

$$V^* = VU^{-T} \quad (18)$$

where $V = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}$, $V^* = \begin{pmatrix} v_1^* \\ \vdots \\ v_m^* \end{pmatrix}$. From (18),

$$v_i^* = v_i U^{-T} \quad (19)$$

For a given constant c , $\{cv_i^*\}$ is taken as input data for all i .

The constant c is chosen depending on the range of σ . Since the dimension of V is $m \times (m-n)$, the number of the input of BSNN is $m-n$. Note that the last training input is a zero vector, representing no sensor fault in the system.

Since the number fault models is identical to the number of sensors, m , hence the number of output neurons of the BSNN is $m+1$ with the last one associated with the fault-free system. When the system is operating normally, all the output of the network are zero, except the last element. However, when a fault occurs, say in the i th sensor, the i th element of the output becomes 1, whilst the last element becomes zero, indicating that a fault has occurred. Let e_i be the i th column of the $(m+1) \times (m+1)$ identity matrix. Note that for an input training vector cv_i^* , the output vector is e_i , i.e., a vector with zeros everywhere, but unity at the i th element. If the input vector is zero, the output vector is e_{m+1} , i.e., unity at the $(m+1)$ th element.

A problem in this approach is the choice of V , since there exists an arbitrary large number of choices. Indeed, V can be constructed by any linear combination of $(m-n)$ columns of P_V . Consequently, there are infinite number of choice of the orthogonalized matrix V^* , implying the the training may be a function of the choice of V . However, it is shown below that the angle $\langle v_i^*, v_j^* \rangle$ is independent on the choice of V .

Let $\|\bar{v}_i^*\| = \sqrt{P_V(i,i)}$, where $P_V(i,j)$ is the i th row and j th column element of P_V . If the diagonal elements of P_V are

identical and equal to r^2 , i.e., $P_V(i,i) = r^2$, then the end point of \bar{v}_i^* is on a hyper-sphere with the center at the original and a radius of r .

From Eqs. (5), (17) and (19),

$$P_V(i,j) = v_i(V'V)^{-1}v_j'$$

$$\begin{aligned} \bar{v}_i^* \cdot \bar{v}_j^* &= v_i^*(v_j^*)' = v_i U^{-T}(v_j U^{-T})' = v_i U^{-T} U^{-1} v_j' \\ &= v_i(UU')^{-1}v_j' = v_i(V'V)^{-1}v_j' = P_V(i,j) \end{aligned}$$

then $\|\bar{v}_i^*\| = \sqrt{\bar{v}_i^* \cdot \bar{v}_i^*} = \sqrt{P_V(i,i)}$. Since

$$\cos \langle v_i^*, v_j^* \rangle = \frac{\bar{v}_i^* \cdot \bar{v}_j^*}{\|\bar{v}_i^*\| \cdot \|\bar{v}_j^*\|} = \frac{P_V(i,j)}{\sqrt{P_V(i,i)P_V(j,j)}}$$

and P_V is independent on the choice of V [8], hence the angle $\langle v_i^*, v_j^* \rangle$ is independent of V .

4.2 Fault detection

From (7) and (4), the residual vector $((m-n) \times 1)$, ρ , is

$$\rho = V'b + V'\epsilon \quad (20)$$

which is not a direct function of the system state. Note that the variance of $V'\epsilon$ is $\sigma^2 V'V$, hence ρ is a correlated vector. The computation of the multivariate B-spline function of ρ can be time-consuming. To save computing time, ρ is transformed as follows. From eqns. (17) and (18), ρ is transformed by

$$\rho^* = U^{-1}\rho \quad (21)$$

From (7),

$$\rho^* = V^*M \quad (22)$$

ρ^* is the input of the BSNN in fault detection. It can be readily verified that V^* satisfies Potter's condition given by (6) from (17). Since V^* is orthogonal to H , such that $Cov(\rho^*) = \sigma^2 I$, i.e., each element of ρ^* is uncorrelated to each other, and has the same variance σ^2 , hence only a univariate B-spline function needs to be computed, giving a significant saving in computing time. Further, it can be readily shown that V^* satisfies the orthogonal condition given by (4). From (4) and (22),

$$\rho^* = (V^*)'b + \epsilon^* = \sum_{i=1}^m (v_i^*)'b_i + \epsilon^* \quad (23)$$

where ϵ^* is mean-zero with a covariance matrix of $\sigma^2 I$. When the i th sensor fails, then $b_i \neq 0$, and $b_j = 0$, for $j \neq i$. Eq.(23) can be rewritten as

$$\rho^* = b_i(v_i^*)' + \epsilon^* \quad (24)$$

Taking expectation of Eq. (24), ρ^* has a bias of $b_i(v_i^*)'$. If b_i is large compared with σ , sensor fault can be detected and properly isolated. Hence, v_i^* can be considered as the fault input vector of the i th sensor in the space of the residual.

4.3 Computation of the transformed input vector

Without loss of generality, let $X = (x_1, \dots, x_n)'$ be the input of the BSNN (e.g. $X = \rho^*$ in section 4.2), and assume that each element of X is uncorrelated with others. The transformed input vector of X can be expressed directly as a vector consists of the transformed input vectors with n elements, i.e.,

$$s(X) = (s'(x_1), \dots, s'(x_n))' \quad (25)$$

where, $s(x_i)$ is defined previously. Let the dimension of $s(x_i)$ be m_i , then the dimension of $s(X)$ is: $\sum_{i=1}^n m_i$, instead of $\prod_{i=1}^n m_i$. The later is the number of multivariate spline functions obtained from the tensor operation. It indicates that the computing time decreases drastically using the proposed transformation.

5. Application Example

Consider a navigation system with six gyros mounted on the surface of a dodecahedron, and the configuration matrix for the sensors is

$$H = \begin{bmatrix} 0.52573 & 0 & 0.85065 \\ -0.52573 & 0 & 0.85065 \\ 0.85065 & 0.52573 & 0 \\ 0.85065 & -0.52573 & 0 \\ 0 & 0.85065 & 0.52573 \\ 0 & 0.85065 & -0.52573 \end{bmatrix}$$

In this example, the BSNN consists of 3 inputs and 7 output, and is trained by the ALR algorithm discussed in Section 3.1.2. Other details of the training are: the number of inner knots is 50, the order of spline function is 1, the number of training samples is 61, $c = 2, 1.8, 1.6, 1.4, 1.2, 1., 0.8, 0.6, 0.4, 0.2$, and V is constructed from the first three columns of P_V .

5.1 Using the proposed method

(i) Constant fault

In the simulation, the standard deviation of σ is set to 0.01. For the trainig period from 40 to 80, a fault of magnitude 2 occurred in the first sensor. For the testing period from 120 to 160, the magnitude of first sensor constant fault reduces to 1, whilst other sensors are working normally. The output of the first, the second and the seventh output neurons of the BSNN using the proposed method are shown in Fig. 2. The output of the other output neurons are not shown as they are similar to that of the second neuron. Clearly, for the periods from 40 to 80, and from 120 to 160, the output of the first output neuron is significantly larger than the output of the other output neurons, which are close to zero. Therefore, sensor faults can be detected from the seventh output neuron, whilst the ouput from the other output neurons to isolate sensor faults.

(ii) Drifting fault

Assume that the drifting fault b_1 of sensor 1 is given by,

$$b_1 = \begin{cases} 0.01(k-50) & k > 50 \\ 0 & \text{others} \end{cases} \quad (26)$$

In Fig. 3, the output of the first output-neuron using the proposed method is shown, illustrating that drifting fault can be detected shortly after it occurred in sensor 1.

5.2 Using the modified Krogmann's method

In Krogmann's method, only six vectors are considered, obtained from $cP_V(\cdot, i)$, for $i = 1, \dots, 6$, where c is constant. In this example, the Krogmann's method is modified as follows.

- (1) The constant c is chosen similar to the proposed method.
- (2) A zero vector is added to the input of the network, such that there are 7 instead of 6 output neurons. Now, the output neurons are the same in both the proposed and the modified Krogmann's method.
- (3) The BSNN, instead of BP neural network is used.
- (4) The testing input is $P_V M$.

For the drifting fault of the sensor 1 described by (26), the output of the first output-neuron using the modified Krogmann's method is shown in Fig. 4. The reason for adding a zero vector is clear from Fig. 5, as the output without the zero vector is biased.

From the discussion in Section 4.1, the choice of V has no effect in the fault detection in the proposed method. To illustrate this point, another V is chosen from the 3-5th columns of P_V . For the drifting fault of the sensor 1, the fault is similarly detected as shown in Fig. 6.

6. Conclusion

A technique for the detecting and isolating faults in redundant systems based on B-spline neurofuzzy networks is presented. It is shown that abrupt and drifting faults can be detected, and hence isolated for systems with redundant sensor based on a uncertainty measurement model. Another advantage of the proposed method is that its reliability can be improved using several models computed by several different choice of V . The performance of the proposed technique is illustrated by an simulation example, and compared with the Krogmann's method.

References

- [1] Daly, K. C., Gai, E., and Harrison, J. V., *Generalized likelihood test for FDI in redundant sensor configurations*, J. Guidance and Control, 2, 1979, 9-17.
- [2] Jin, H. and Zhang, H. Y., *Failure detection using optimal parity vector sensitive to special sensor failure*, Proc. IEEE/SICE/RSJ Int. Conf. on Multi-sensor, Fusion and Integration for Intelligent Systems, Dec. 8-11, 1996, 55-61.
- [3] Jin, H., and Zhang, H. Y., *Optimal parity vector sensitive to designated sensor fault*, IEEE Trans. on Aero. and Electr. Sys., Vol. 35, No. 4, 1999, 1122-1128.

- [4] Chan C. W., Hong Jin, K. C. Cheung, H. Y. Zhang, *State estimation with measurement error compensation using neural network*, Proc. IEEE Conf. on Contr. Appl., 1-4 Sept. 1998, Triests, Italy, 153-157.
- [5] Krogmann, U., *Artificial neural networks for inertial sensor fault diagnosis*, Symposium Gyro Technology, 1995, 5.0-5.9.
- [6] H. Benkhedda and R. J. Patton, *B-spline network integrated qualitative and quantitative fault detection*, IFAC 13th Triennial World Congress, San Francisco, USA, 1996.
- [7] Brown M. and C. Harris, *Neurofuzzy adaptive modelling and control*, Prentice Hall, 1994.
- [8] Jin, H. and Zhang, H. Y., *Configuration of redundant sensor system and its fault detection using parity vector method*, IFAC Symp. on Fault Detection, Supervision And Safety for Technical Processes, Aug. 26-28, 1997, U. K., 2, 843-848.
- [9] C. W. Chan, H. Jin, K. C. Cheung, and H. Y. Zhang, *An adaptive learning rate for the training of B-spline networks*, Proc. UKACC Int. Conf. on Control' 98, 1-4 Sept., 1998, Swansea, UK, 342-347.
- [10] N. J. Bershad and P. L. Feintuch, *A normalized frequency domain LMS adaptive algorithm*, IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-34, 1986, 452-461.

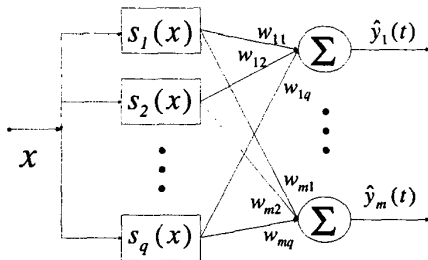


Fig.1 BSNN with multi-dimensional output

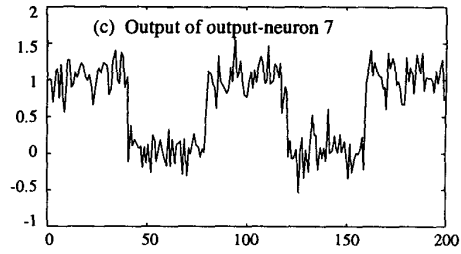
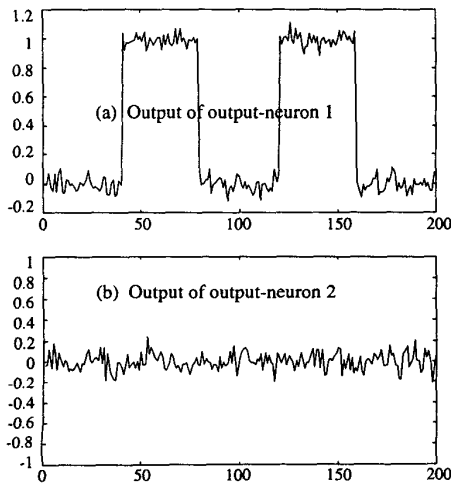


Fig.2 FDI of a constant fault using the proposed method

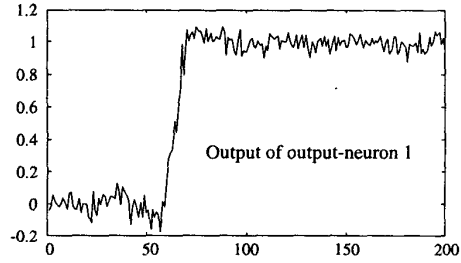


Fig.3 FDI of drifting fault using proposed method

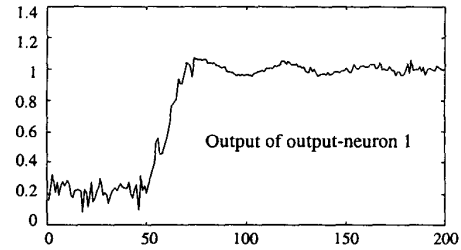


Fig.4 FDI of drifting fault using modified Krogmann's method with 6I6O BSNN

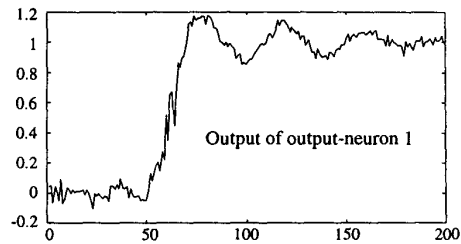


Fig. 5 FDI of a drift fault using modified Krogmann's method with 6I7O BSNN

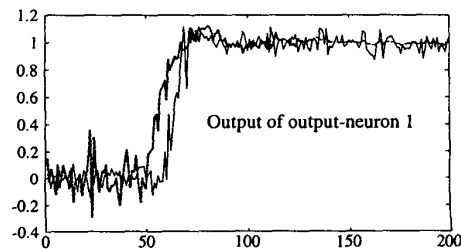


Fig. 6 FDI of drifting fault using another V