한국 퍼지 및 지능시스템학회 논문지 2000, Vol. 10, No. 3 (2000)211~217

Actuator Fault Diagnostic Algorithm based on Hopfield Network

Tae-Geon Park, Ji-Su Ryu*, Hak-Bom Hur, In-Mo Ahn*** and Kee-Sang Lee***

Dept. of Electronics, Okchon Provincial College
**Dept. of Electrical Eng., Dankook University*
***Dept. of Control and Instrumentation, Chonan Technical College*
****Dept. of Electrical Eng., Masan College*

ABSTRACT

A main contribution of this paper is the development of a Hopfield network-based algorithm for the fault diagnosis of the actuators in linear systems with uncertainties. An unknown input decoupling approach is introduced to the design of an adaptive observer so that the observer is insensitive to uncertainties. As a result, the output observation error equation does not depend on the effect of uncertainties. Simultaneous energy minimization by the Hopfield network is used to minimize the least mean square of errors of estimates of output variables. The Hopfield network provides an estimate of the gains of the actuators. When the system dynamics changes, identified gains go through a transient period and this period is used to detect faults. The proposed scheme is demonstrated through its application to a simulated second-order system.

1. Introduction

Artificial neural networks offer the advantage of performance improvement through learning using parallel and distributed processing. These networks are implemented using massive connections among processing units with variable strengths, and they are attractive for applications in system identification and control [1].

Hopfield and Tank [2] demonstrated that some classes of optimization problems can be programmed and solved on neural networks. They have been able to show the power of neural networks in solving difficult optimization problems [1]. The main advantage of the Hopfield network is that it can perform the least squares based minimization in parallel fashion. Furthermore, the Hopfield based neural network identification can be applied to nonlinear systems provided that the process model is linear in terms of the parameters to be estimated but nonlinear in terms of the process input and output [3]. In the recent years, there has been some success in using the Hopfield network for estimation purposes of system parameters [1,3]. Chu *et al.* [1] identified the system in the state-space form using Hopfield network. Srinivasan and Batur [3] implemented least squares algorithm using the Hopfield network to estimate the parameters of a discrete time system.

The object of this paper is to indicate how to apply the Hopfield network to the problem of identification of the gains of the actuators in linear systems, where the change of the gains of the actuators means the actuator fault. The mean-square error typically is used as a performance criterion in identification of the gains of the actuators. Our motivation is to study whether it is possible to express idenification problems of the gains of the actuators in the form of programming the Hopfield optimization network. By measuring inputs and outputs, a procedure is presented for programming the Hopfield network. The application of the proposed algorithm to actuator fault diagnosis is illustrated by a second-order system.

2. Hopfield network model

In the continuous Hopfield model, the behavior of a neuron is governed by the following differential equation [2]:

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} + \sum_{j=1}^{N} T_{ij} V_j + I_i,\ 1 \le i \le N, \tag{1}$$

where N is the total number of neurons, U_i is the neuron's internal state for neuron i, V_j is the output of neuron j, T_{ij} is the connection weight from neuron j to neuron i, and I_i is a bias input to neuron i. The neuron's output

$$V_j = g(\lambda_j U_j) \tag{2}$$

is a nondecreasing function of the activation level, where $g(\cdot)$ represents the input-output characteristic of a nonlinear amplifier and the scaling parameter λ_j effectively defines the steepness of the nonlinearity. Most implementation of Hopfield networks employ a

본 연구는 교육부 지정 마산대학 "메카트로닉스 자동화 분야 전문 인력양성" 특성화 사업의 지원에 의한 것입니다.

sigmoid activation function such as the tanh function:

$$V_j = g(\lambda_j U_j) = G_j \tanh(\lambda_j U_j), \quad (3)$$

where $\pm G_j$ indicates the asymptotic limits for $U_j = \pm\infty$. It has been shown by Hopfield [2] that if the following conditions are imposed then the network converges to a set of stable states:

① the synaptic weights are symmetric, $T_{ij} = T_{ji}$

② $g(\cdot)$ and its inverse are monotone increasing functions of their arguments as in the case of sigmoid function,

and furthermore, these states correspond to a local minima of the following energy function:

$$E = -[(1/2)V^T TV + IV], \quad (4)$$

when the gains of the activation function are sufficiently high, where V is an $N\times 1$ column vector containing the neuron's output, T is an $N\times N$ matrix, I is an $1\times N$ row vector.

3. Model of the faulty process

Consider a system described by the following equations:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ev(t)$$
$$y(t) = Cx(t), \quad (5)$$

where $x(t) \in \mathbf{R}^n$ is the unmeasurable state vector, $u(t) \in \mathbf{R}^m$ is the measurable input vector, $y(t) \in \mathbf{R}^p$ is the measurable output vector, $v(t) \in \mathbf{R}^q$ is an unmeasurable term representing model uncertainties or input noises of the system. A, B, E and C are known constant matrices with appropriate dimensions.

The actuator faults are modelled as

$$u_a(t) = f(t)u(t), \quad (6)$$

where $u_a(t)$ and $u(t)$ are the actual actuator output and the requested actuator input, respectively. $f(t) \in \mathbf{R}^{m\times m}$ is a diagonal matrix representing the gains of the actuators. Under normal operating conditions, the gains are equal to identity matrices. Under faulty operating conditions, the gains change, reflecting the effect of fault. For example, a stuck actuator valve will cause the corresponding element of $f(t)$ to change from a valve of unity to a valve of zero. Using (6) in (5), the dynamics of the faulty process can be modelled as

$$\dot{x}(t) = Ax(t) + Bf(t)u(t) + Ev(t)$$
$$y(t) = Cx(t), \quad (7)$$

It is assumed that the pair (A, C) is observable and the sensors are perfectly reliable, and only concern ourselves with identifying actuator faults. However, ignoring the case of sensor faults is only for the sake of brevity and simplifying the exposition. A detailed treatment of sensor failures can be found in [4].

4. Fault diagnosis: Identification of the gains of the actuators

Wang *et al.* [4] proposed the fault detection and diagnosis algorithm for linear actuators and sensors, where an adaptive observer was constructed to diagnose the faults. Using the augmented error technique from model reference adaptive control, an observation error model was formulated and used to establish an adaptive diagnostic algorithm that produces the estimates of the gains of the actuators and the sensors. In the algorithm, the adjustment of the gains is performed only when the magnitude of the augmented error is larger than a predetermined value. The method is relatively simple in the conditions for convergence and stability of an adaptive system. Although this method often yields acceptable designs, it has several drawbacks. For example, it requires a priori knowledge of the norms of an unmeasurable term representing model uncertainties or input noises of the system, which is used to achieve stable adaptive behaviour. Furthermore, uncertainties directly affect the adaptive observer and the diagnostic algorithm. So adaptability depends to a large extent on the degree and the frequency component of an unmeasurable term.

The mean square-based fault diagnosis is performed here by the Hopfield network. The main reason behind this choice is that one can formulate the fault diagnosis problem as a minimization of a performance index and then construct a well-defined neural network to determine a minima that corresponds to the estimated gains of the actuators. Furthermore, the computation associated with the proposed fault diagnosis algorithm can be performed in parallel fashion.

4.1 The monitoring system description free of unknown inputs

A necessary condition for decoupling an unmeasurable term $v(t)$ is that $rank(CE) = q$, $(q \leq p)$. This means that to implement an observer that is insensitive to an unmeasurable term, specific state variables must be measured or at least appear as a part of the output [5]. Therefore we assume $rank(CE) = rank(E)$. From the state-space description of (7), $\dot{y}(t)$ is obtained as

$$\dot{y}(t) = CAx(t) + CBf(t)u(t) + CEv(t). \quad (8)$$

Since CE has full column rank, the Moore-Penrose generalized inverse matrix $(CE)^+$ can be obtained by

$$(CE)^+ = [(CE)^T(CE)]^{-1}(CE)^T \in \mathbf{R}^{q\times p}. \quad (9)$$

Thus, from (8), we can estimate the effect of an unmeasurable term $v(t)$ by the following equation:

$$v(t) = (CE)^{+}\{\dot{y}(t) - CAx(t) - CBf(t)u(t)\}. \quad (10)$$

Substituting (10) into (7), we arrive at the following state equation independent of the unmeasurable term:

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}f(t)u(t) + E(CE)^{+}\dot{y}(t)$$
$$y(t) = Cx(t), \quad (11)$$

where

$$\tilde{A} = (I_n - E(CE)^{+}C)A, \ \tilde{B} = (I_n - E(CE)^{+}C)B. \quad (12)$$

4.2 Adaptive observer and its output estimation error equation

Assuming that the pair $(\tilde{A}, C)$ is observable, the adaptive observer is constructed in which unknown inputs are decoupled:

$$\dot{x}_m(t) = \tilde{A}x_m(t) + \tilde{B}\hat{f}(t)u(t) + E(CE)^{+}\dot{y}(t) + L(y(t) - y_m(t))$$
$$y_m(t) = Cx_m(t), \quad (13)$$

where $x_m(t) \in \mathbf{R}^n$ is the state vector of the observer, $y_m(t) \in \mathbf{R}^p$ is the observer output vector, $\hat{f}(t)$ is the estimate of $f(t)$, L is a pre-specified observer gain matrix that makes the matrix $\tilde{A} - LC$ stable. Defining a new state vector as

$$z(t) = x_m(t) - E(CE)^{+}y(t), \quad (14)$$

leads to the following state-space equations:

$$\dot{z}(t) = \tilde{A}_o z(t) + Gy(t) + \tilde{B}\hat{f}(t)u(t) \quad (15)$$

$$x_m(t) = z(t) + E(CE)^{+}y(t) \quad (16)$$

$$y_m(t) = Cx_m(t), \quad (17)$$

where

$$\tilde{A}_o = \tilde{A} - LC, \ G = \tilde{A}_o E(CE)^{+} + L. \quad (18)$$

(15) can be rewritten as

$$\dot{z}(t) = \tilde{A}_o z(t) + Gy(t) + \tilde{B}\sum_{i=1}^{m}\hat{f}_i(t)u_i(t)$$
$$= \tilde{A}_o z(t) + Gy(t) + \tilde{B}\sum_{i=1}^{m}u_i(t)I_m\hat{f}_i(t), \quad (19)$$

where $\hat{f}_i(t)$ is the ith column vector of matrix $\hat{f}(t)$ and $u_i(t)$ is the ith element of vector $u(t)$. Define the following state variable filters:

$$\dot{W}(t) = \tilde{A}_o W(t) + Gy(t), \ W(0) = 0 \quad (20)$$

$$\dot{P}_i(t) = \tilde{A}_o P_i(t) + \tilde{B}u_i(t)I_m, \ P_i(0) = 0, \ 1 \le i \le m, \quad (21)$$

where $W(t) \in \mathbf{R}^{n\times 1}$, $P_i(t) \in \mathbf{R}^{n\times m}$. Then state vector $z(t)$ can be written as

$$z(t) = [P_1(t) \ P_2(t) \cdots P_m(t)]\hat{\theta} + W(t) + \exp(\tilde{A}_o t)z_o \quad (22)$$

where $\hat{\theta} = [\hat{f}_1^T \ \hat{f}_2^T \cdots \hat{f}_m^T]^T \in \mathbf{R}^{mm\times 1}$, $z_o = z(0)$. Therefore the estimated state vector $x_m(t)$ can be written as

$$x_m(t) = [P_1(t) \ P_2(t) \cdots P_m(t)]\hat{\theta} + W(t) + \exp(\tilde{A}_o t)(x_{mo} - E(CE)^{+}y_o) + E(CE)^{+}y(t) \quad (23)$$

where $x_{mo} = x_m(0)$, $y_o = y(0)$. The output estimation error becomes

$$\tilde{y}(t) = y(t) - y_m(t)$$
$$= y(t) - [\phi^T(t)\hat{\theta} + CW(t) + y_a(t) + \tilde{y}_o(t)], \quad (24)$$

where

$$\phi^T(t) = C[P_1(t) \ P_2(t) \cdots P_m(t)] \quad (25a)$$

$$y_a(t) = CE(CE)^{+}y(t) \quad (25b)$$

$$\tilde{y}_o(t) = C\exp(\tilde{A}_o t)(x_{mo} - E(CE)^{+}y_o). \quad (25c)$$

4.3 The implementation of a mean square algorithm based on Hopfield network

The estimates of the gains of the actuators are obtained using a mean square algorithm that is implemented by the Hopfield network. Using a mean square algorithm, the estimates are obtained by minimizing the following performance index:

$$J(t) = \frac{1}{2(\Delta T)}\int_{t-\Delta T}^{t}\tilde{y}^T(\lambda)\tilde{y}(\lambda)[\exp(-(t-\lambda)/\mu)]d\lambda, \quad (26)$$

where ΔT is the time interval of interest. Note that an exponentially decaying window with time constant μ is used in the above energy function. This window has the effect of emphasing the most current estimation error and has a gradually fading memory of earlier errors. Substituting (24) into (26) and simplifying the resulting expression, the performance index $J(t)$ is given as

$$J(t) = J_1(t) + J_2(t) + J_3(t) \quad (27)$$

where

$$J_1(t) = \frac{1}{2(\Delta T)}\int_{t-\Delta T}^{t}\hat{\theta}^T\phi(\lambda)\phi^T(\lambda)[\exp(-(t-\lambda)/\mu)]\hat{\theta}d\lambda$$

$$J_2(t) = \frac{1}{(\Delta T)}\int_{t-\Delta T}^{t}[\phi(\lambda)(CW(\lambda) + y_a(\lambda) + \tilde{y}_o(\lambda) - y(\lambda))]^T \cdot [\exp(-(t-\lambda)/\mu)]\hat{\theta}d\lambda$$

$$J_3(t) = \frac{1}{(\Delta T)}\int_{t-\Delta T}^{t}[y(\lambda) - (CW(\lambda) + y_a(\lambda) + \tilde{y}_o(\lambda))]^T \cdot [y(\lambda) - (CW(\lambda) + y_a(\lambda) + \tilde{y}_o(\lambda))][\exp(-(t-\lambda)/\mu)]d\lambda,$$

where $J_3(t)$ is a non-negative value within each sampling time and has no effect on $dJ(t)/dt$ [2].

Consequently, the truncated version of (27) becomes

$$J(t) = J_1(t) + J_2(t) \tag{28}$$

Assume that $\hat{\theta}$ is constant. Letting the output of each neuron represent one of the unknown parameters, that is $\hat{\theta} = V$, and by comparing (4) with (28) the connection weights and biases of the Hopfield network can be represented as follows:

$$T(t)=\frac{\eta}{(\Delta T)}\int_{t-\Delta T}^{t}\phi(\lambda)\phi^T(\lambda)[\exp(-(t-\lambda)/\mu)]d\lambda \tag{29}$$

$$I(t)=\frac{\eta}{(\Delta T)}\int_{t-\Delta T}^{t}[\phi(\lambda)(y(\lambda)-CW(\lambda)-y_a(\lambda)-\tilde{y}_o(\lambda))]^T \cdot [\exp(-(t-\lambda)/\mu)]d\lambda, \tag{30}$$

where η is the learning gain. $T(t)$ is $mm\times mm$ time-varying matrix and $I(t)$ is $1\times mm$ row vector. Therefore the total number of neurons, N in Section 2 becomes that of the gains of the actuator to be estimated, mm. Note that the connection weights, $T(t)$ are symmetric in (29). The outputs of the Hopfield network with the derived connection weights (29) and biases (30) converge to the values of the gains of the actuators.

4.4 Design and implementation procedures of the proposed algorithm

The design and implementation procedures of Hopfield network based identification algorithm for actuator fault diagnosis are summarized as in the following:

Design procedures of the proposed algorithm

① Read system matrices A, B, E, C, the time interval of interest $\triangle T$, the learning gain η, and the time constant μ.

② Compute $E(CE)^+$, $\tilde{A}$ and $\tilde{B}$.

③ Choose L such that the matrix $\tilde{A}_o$ is stable, where $\tilde{A}_o = \tilde{A} - LC$, and find $\tilde{A}_o$ and G.

④ Obtain $T(t)$ and $I(t)$, where $T(t)$ is the function of $u(t)$ and $I(t)$ is the function of $u(t)$ and $y(t)$.

Implementation procedures of the proposed algorithm

① Read parameter τ in the Hopfield model (Eq. (1)), parameters λ_j, $(1\le j\le N, N = mm)$ (Eq. (2)), and parameters G_j, $(1\le j\le N)$ (Eq. (3)). Here the output of each neuron V_j is constrained within the subset of positive hypercubes G_j. Therefore, for any parameter f_{ki} in the gains of the actuators, G_j must be chosen such that $|f_{ki}| < G_j$, where $f_i(t) = [f_{1i}\ f_{2i}\ \cdots\ f_{mi}]^T$ is the ith column vector of matrix $f(t)$, $(1\le i\le m,\ 1\le k\le m)$. Also set initial values of the observer, x_{mo} and initial outputs, y_o.

② Obtain $u(t)$ and $y(t)$.

③ Compute $\phi^T(\lambda) = C[P_1(\lambda)\ P_2(\lambda)\ \cdots\ P_m(\lambda)]$, $W(\lambda)$, $y_a(\lambda)$ and $\tilde{y}_o(\lambda)$.

④ Find $T(t)$ and $I(t)$.

⑤ Compute the states and outputs of the Hopfield network, where $\hat{f} = [\hat{f}_1^T\ \hat{f}_2^T\ \cdots\ f_m^T]^T = V$.

⑥ If $t < t_{max}$, go to ②, where t_{max} is the implementation ending time. Otherwise terminate the implementation.

5. A numerical example

The above theoretical developments will be applied to the system [4]:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bf(t)u(t) + Ev(t)\\ y(t) &= Cx(t),\end{aligned} \tag{31}$$

where

$$x(t)=\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix},\ A=\begin{bmatrix}0.65 & -2.45\\0.3 & -0.9\end{bmatrix},\ B=\begin{bmatrix}0\\1\end{bmatrix},\ E=\begin{bmatrix}1\\1\end{bmatrix},$$

$$C=\begin{bmatrix}-3.5 & 0\\0 & 5.5\end{bmatrix} \tag{32}$$

5.1 Design of the proposed algorithm

step 1: $E(CE)^+ = \begin{bmatrix}-0.0824 & 0.1294\\-0.0824 & 0.1294\end{bmatrix}$, (33)

$$CE(CE)^+ = \begin{bmatrix}0.2882 & -0.4529\\-0.4529 & 0.7118\end{bmatrix}, \tag{34}$$

$$\tilde{A} = (I_n - E(CE)^+ C)A = \begin{bmatrix}0.2491 & -1.1032\\-0.1009 & 0.4468\end{bmatrix}, \tag{35}$$

$$\tilde{B} = (I_n - E(CE)^+ C)B = \begin{bmatrix}-0.7118\\0.2882\end{bmatrix}. \tag{36}$$

step 2: The pair $(\tilde{A}, C)$ is observable. For the system, the observer gains are selected as

$$L=\begin{bmatrix}-0.4997 & -0.2006\\0.0288 & 1.1721\end{bmatrix}, \tag{37}$$

which makes $(\tilde{A}_o = \tilde{A} - LC)$ stable, with two eigenvalues, -6 and -1.5. As a result, the following matrices can be defined:

$$\tilde{A}_o = \tilde{A} - LC = \begin{bmatrix}-1.5 & 0\\0 & -6\end{bmatrix}, \tag{38}$$

$$G = \tilde{A}_o E(CE)^+ + L = \begin{bmatrix}-0.3762 & -0.3947\\0.5229 & 0.3957\end{bmatrix} \tag{39}$$

step 3: Define $P_1(\lambda) = \begin{bmatrix} P_1^{11}(\lambda) \\ P_1^{21}(\lambda) \end{bmatrix}$ with $n = 2$ and $m = 1$. Then

$$\phi^T(\lambda) = \begin{bmatrix} -3.5P_1^{11}(\lambda) \\ 5.5P_1^{21}(\lambda) \end{bmatrix}, \quad t - \triangle T \leq \lambda \leq t, \tag{40}$$

where $P_1(\lambda)$ is determined from

$$\dot{P}_1(\lambda) = \tilde{A}_o P_1(\lambda) + \tilde{B}u(\lambda), \; P_1(0) = [0\ 0]^T. \tag{41}$$

step 4: Define $W(\lambda) = \begin{bmatrix} W^1(\lambda) \\ W^2(\lambda) \end{bmatrix}$ with $n = 2$.

$$\dot{W}(\lambda) = \tilde{A}_o W(\lambda) + Gy(\lambda), \; W(0) = [0\ 0]^T, \tag{42}$$

$$y_a(\lambda) = CE(CE)^+ y(\lambda), \tag{43}$$

$$\tilde{y}_o(\lambda) = C\exp(\tilde{A}_o\lambda)\,(x_{mo} - E(CE)^+ y_o). \tag{44}$$

step 5:

$$T(t) = -\frac{\eta}{(\Delta T)}\int_{t-\Delta T}^{t}[(-3.5P_1^{11}(\lambda))^2 + (5.5P_1^{21}(\lambda))^2]d\lambda. \tag{45}$$

$$I(t) = \frac{\eta}{(\Delta T)}\int_{t-\Delta T}^{t}[\phi(\lambda)(y(\lambda) - CW(\lambda) - y_a(\lambda) - \tilde{y}_o(\lambda))]^T d\lambda \tag{46}$$

where equations (45) and (46) stem from

$$J(t) = \frac{1}{2(\Delta T)}\int_{t-\Delta T}^{t}\tilde{y}^T(\lambda)\tilde{y}(\lambda)d\lambda \tag{47}$$

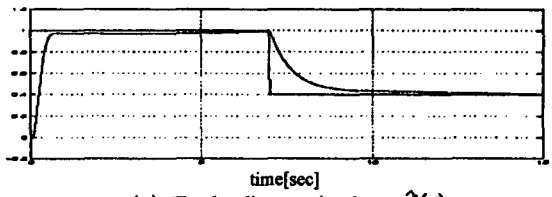
instead of $J(t)$ of (26).

step 6: From (1) and (3), $\hat{f}(t) = \hat{f}_1(t) = \hat{f}_{11}(t) = V = V_1$.

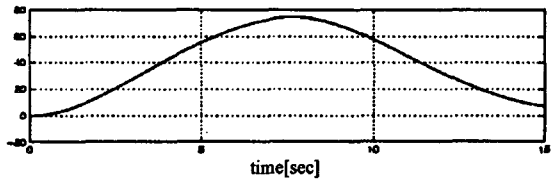
5.2 Simulation results

Computer simulations were carried out, where the system is subjected to a step input at $t = 0$. In simulating a high-gain limiting case with a sampling period of rm 10[msec], a large gain is used as $\lambda_1 = 100$. Some data are given as follows: $\eta = 1$, $\tau = 100$, and $x_{mo} = y_o = [0\ 0]^T$. The performance of the proposed fault diagnosis algorithm depends on the time interval of interest. For example, if the time interval of interest is big, the algorithm may not respond to faults fast enough due to the strong effect of the past data. On the other hand small time intervals will increase the variance of the identified gains of actuators. Here the time interval of interest, $\triangle T$ is chosen asrm 40[msec]. It is assumed that the healthy actuator has the gain $f_H = 1$. A fault is created as follows:

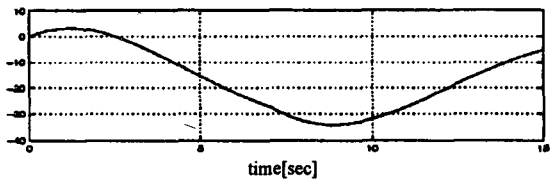
$$f(t) = \begin{cases} f_H = 1 & (t < 7[\text{sec}]) \\ 0.4 & (t \geq 7[\text{sec}]) \end{cases} \tag{48}$$



(a) Fault diagnosis for $\hat{f}(t)$.

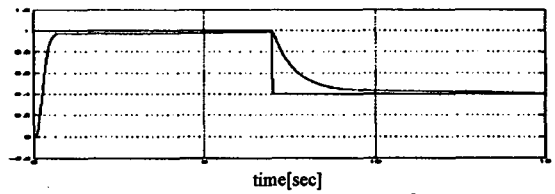


(b) The real state x_1 and its estimate.

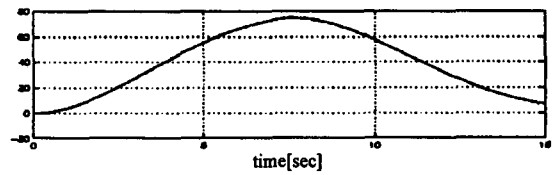


(c) The real state x_2 and its estimate.

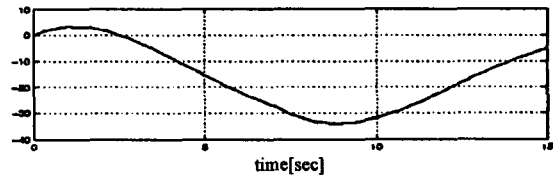
Fig. 1. Fault diagnosis and state estimation with $v(t) = 0.2\sin 30t$



(a) Fault diagnosis for $\hat{f}(t)$.

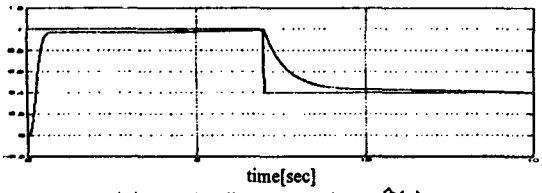


(b) The real state x_1 and its estimate.

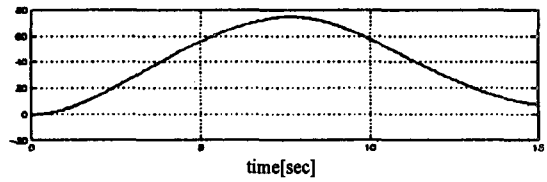


(c) The real state x_2 and its estimate.

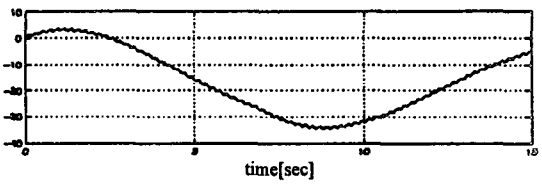
Fig. 2. Fault diagnosis and state estimation with $v(t) = 0.2\sin 10t$



(a) Fault diagnosis for $\hat{f}(t)$.

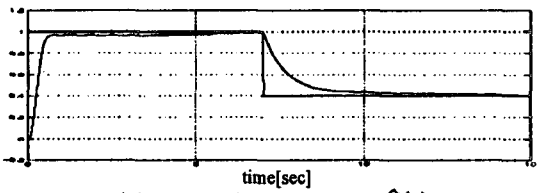


(b) The real state x_1 and its estimate.

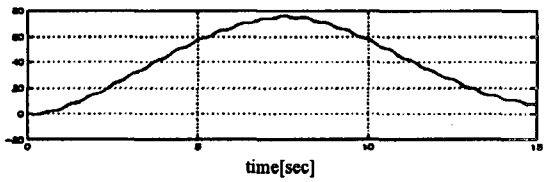


(c) The real state x_2 and its estimate.

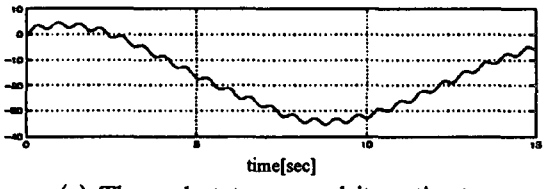
Fig. 3. Fault diagnosis and state estimation with $v(t) = 2\sin 30t$



(a) Fault diagnosis for $\hat{f}(t)$.



(b) The real state x_1 and its estimate.



(c) The real state x_2 and its estimate.

Fig. 4. Fault diagnosis and state estimation with $v(t) = 2\sin 10t$

where $f(t) = f_1(t) = f_{11}(t)$ with $m = 1$. The output of a neuron evolves within the prescribed hypercube where the size of the hypercube is chosen as $G_1 = 2$, that is twice of $|f_H|$. Moreover, a zero initial state of the neuron is assigned. Without a priori knowledge about uncertainties four different uncertainties are assumed in the simulation: $v(t) = 0.2\sin(30t)$, $v(t) = 0.2\sin(10t)$, $v(t) = 2\sin(30t)$ and $v(t) = 2\sin(10t)$. Figs. 1-4 represent the application results of the proposed algorithm. In each figure, (a) is the estimate of the gain of an actuator, $f(t)$, and (b) and (c) show the estimated states by the adaptive observer. It is seen from these figures that both the parameter $f(t)$ and two states are estimated quite well regardless of the magnitude and the frequency component of an unmeasurable term. When the system dynamics changes, identified gains go through a transient period and this period can be used to detect faults.

6. Conclusions

A technique for programming of the Hopfield network for identification of the gains of the actuators was developed. In this technique, the Hopfield network was used to implement a least-squares estimation for the gains of the actuators. When the system dynamics changes, identified gains go through a transient period and this period can be used to detect faults. It has been shown that the adaptive diagnostic algorithm can give the desired performance for fault diagnosis through its application to a second-order system.

References

[1] S. R. Chu, R. Shoureshi and M. Tenorio, "Neural networks for system identification," *IEEE Contr. Sys. Magazine*, Vol. 31, No. 4, pp. 31-35, 1990.

[2] J. M. Zurada, Introduction to artificial neural systems, West Publishing Company, 1992.

[3] A. Srinivasan and C. Batur, "Hopfield/ART-1 neural network-based fault detection and isolation," *IEEE Trans. Neural Networks*, Vol. 5, No. 6, pp. 890-899, 1994.

[4] H. Wang, Z. J. Huang and S. Daley, "On the use of adaptive updating rules for actuator and sensor fault diagnosis," Automatica, Vol. 33, No. 2, pp. 217-225, 1997.

[5] D. S. Hwang, S. K. Chang and P. L. Hsu, "A practical design for a robust fault detection and isolation system," *Int. J. of System Science*, Vol. 28, No. 3, pp. 265-275, 1997.

박 태 건 (Tae-Geon Park)

1992년 : 단국대학교 전기공학과(공학사)
1994년 : 단국대학교 전기공학과 (공학석사)
1999년 : 단국대학교 전기공학과 (공학박사)
2000년~현재 : 옥천전문대학 전자과 전임강사

허 학 범 (Hak-Bom Hur)

1971년 : 홍익대학교 전기공학과(공학사)
1974년 : 명지대학교 전자공학과 (공학석사)
1993년~현재 : 단국대학교 전기공학과 박사과정
1978년~현재 : 천안공업대학 제어계측과 교수
주관심분야 : 신호처리 응용

이 기 상 (Kee-Sang Lee)

1978년 : 고려대학교 전기공학과 졸업 (공학사)
1981년 : 고려대학교 대학원 전기공학과 졸업(공학석사)
1984년 : 고려대학교 대학원 전기공학과 졸업(공학박사)
현재 : 단국대학교 전기공학과 교수

류 지 수 (Ji-Su Ryu)

1990년 : 단국대학교 전기공학과 (공학사)
1996년 : 단국대학교 전기공학과 (공학석사)
1996년~현재 : 단국대학교 전기공학과 박사과정

안 인 모 (In-Mo Ahn)

1981년 : 동아대학교 전기공학과(공학사)
1983년 : 동아대학교 전기공학과 (공학석사)
현재 : 마산대학 전기과 교수