Proceedings of the
2005 IEEE Conference on Control Applications
Toronto, Canada, August 28-31, 2005

A Neural Network-Based Fault Detection Scheme for Satellite Attitude Control Systems

H.A. Talebi and R.V. Patel

Abstract— **This paper presents an actuator Fault Detection and Identification (FDI) scheme for satellite attitude control systems. A state-space approach is used and a nonlinear-in-parameters neural network (NLPNN) is employed to identify the general unknown fault. The recurrent network configuration is obtained by a combination of feedforward network architectures and dynamical elements in the form of stable filters. The neural network weights are updated based on a modified backpropagation scheme. The stability of the overall fault detection scheme is shown using Lyapunov's direct method. Simulation results are presented to show the performance of the proposed fault detection scheme.**

I. INTRODUCTION

The problem of fault detection and identification in satellite Attitude Control Systems (ACS) has received a great deal of attention during the past several years. The improvements in the accuracy and reliability of Attitude Control Systems (ACS) contributes directly to the success and reliability of satellites in space. Hence, an exceptional level of autonomy is required. Fault detection and identification is an essential component of an autonomous system. This has led to research in developing new methods for supervision, fault detection, fault isolation, and fault recovery. The inherent nonlinearity of their dynamics, however, makes the accurate and efficient fault detection of ACS a challenging problem.

There are a number of approaches for automated fault diagnosis for robotic systems. For instance, expert system methodologies [1] and time series analysis [2] among others are used for failure detection purposes. However, the two main approaches that use analytical redundancy are state and parameter estimation techniques. Parameter estimation methods require knowledge of the model structure of the investigated process and actual process measurements. The actual values of some critical model parameters (e.g. the mass matrix in a robotic system) are then determined from this information. Fault detection is accomplished by a comparison of identified and predetermined values by using some thresholds [3]. Observer-based methods require knowledge of the investigated process as well. The model operates parallel to the real process. Assuming an exact model of the plant, the difference between measured and calculated desired variables (e.g. the the joint velocities in robotic systems) will produce a non-zero value when a fault has occurred. This difference is called a residual. A fault is declared if the residual exceeds a certain threshold value. The main advantage of observer-based methods over parameter estimation methods results from the fact that no special persistent excitation is required. Observer-based methods also work in steady-state operating conditions. It is also well known that the fault capability of the nonlinear observer-based fault detection approach is significantly better than that of linear observer-based approach. Moreover, robotic system faults often cause unpredictable nonlinear changes in the dynamics of the system. Therefore, to capture a large class of practical failure situations, a nonlinear modeling framework is required.

Fault detection can be categorized as an estimation problem and fault isolation can be considered as a classification problem. The capability of neural networks for function approximation and classification and their ability to deal with uncertainty and/or parameter variations make them a good choice for use in the fault detection problem. Neural networks can be employed in the state/parameter estimation and/or fault isolation step. Using a neural network in the estimation step [4], [5] requires the network to identify a dynamical system. Using a neural network as a fault isolator [6], [7], [8] is more straightforward. Neural network learning and fault tolerant capabilities are used in [9], [10] for data fusion of multi-sensor measurements of remote sensing satellites. Sensor fault detection and identification for satellite attitude determination were considered in [11], [12], [13]. A supervisory switching logic algorithm is used in [14] to reconfigure the attitude control system in the case of actuator failures. Depending on the state of the actuator, the appropriate controller was selected from a bank of controllers.

In this paper, an actuator Fault Detection and Identification scheme for satellite attitude control systems is presented. A nonlinear-in-parameters neural network (NLPNN) is employed to identify a general unknown fault. The neural network weights are updated based on a modified backpropagation scheme. The stability of the overall fault detection scheme is shown using Lyapunov's direct method. The rest of the paper is organized as follows: In Section II, the neural network based fault detection scheme is introduced. Section II-A gives the stability analysis. Simulation results are shown in Section III and Section IV gives some conclusions.

This research was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada under grant RGPIN-1345 and grant 258007-02 .

H.A. Talebi is with Faculty of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran and with the Department of Electrical & Computer Engineering, University of Western Ontario, London, Canada htalebi@engga.uwo.ca

Rajni Patel is with the Department of Electrical & Computer Engineering, University of Western Ontario, London, Canada rajni@eng.uwo.ca

0-7803-9354-6/05/$20.00 ©2005 IEEE

II. Proposed Neural Network Structure

The dynamic equation of a satellite is given by

$$H\dot{\omega} = -S(\omega)H\omega + \tau \tag{1}$$

where $\tau \in R^3$ is the control input vector, $\omega = [\omega_1\ \omega_2\ \omega_3]^T$ is the angular velocity vector of the satellite, expressed in satellite body frame, H is the symmetric positive definite inertia matrix of the satellite and $S(\omega)$ is the cross product matrix given by

$$S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{2}$$

The attitude of the satellite can be represented in many ways using ,e.g., Euler angles or singularity-free Euler parameters leading [15] to

$$\Omega = \Pi(\Omega)\omega \tag{3}$$

where the definitions of Ω and Π depend on the choice of the kinematic representation. Although, the fault diagnosis algorithm proposed in this paper is independent of the choice of attitude representation, the Euler angle representation is considered for simulation purposes. Hence, the kinematic equations are described by

$$\begin{bmatrix} \dot{\Psi} \\ \dot{\theta} \\ \dot{\Phi} \end{bmatrix} = \Pi(\Psi,\theta,\Phi) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \tag{4}$$

$$\Pi(\Psi,\theta,\Phi) = \begin{bmatrix} 0 & S\Phi/C\theta & C\Phi/C\theta \\ 0 & C\Phi & -S\Phi \\ 1 & S\Phi tan\theta & C\Phi tan\theta \end{bmatrix},$$

where $\Omega = [\Psi\ \theta\ \Phi]^T$ is the Euler angle vector representing the orientation of the satellite body frame and C, S and tan denote the cos, sin and $tangent$ functions, respectively.

An actuator fault can be considered by adding an extra torque component τ_F to the dynamic equations:

$$H\dot{\omega} = -S(\omega)H\omega + \tau + \tau_F \tag{5}$$

where τ_F represents an unknown fault. Now, by defining $x = \begin{bmatrix} \Omega \\ \omega \end{bmatrix}$ and $u = \tau$, Equations (5) and (4) can be written as

$$\dot{x} = f(x,u) + T_F, \tag{6}$$

$$f(x,u) = \begin{bmatrix} \Pi(\Psi,\theta,\Phi)\omega \\ H^{-1}(\tau - S(\omega)H\omega) \end{bmatrix},$$

$$T_F = \begin{bmatrix} 0 \\ H^{-1}\tau_F \end{bmatrix},$$

where I_3 is the 3×3 identity matrix. By adding and subtracting Ax, we can write (6) as

$$\dot{x} = Ax + g(x,u) + T_F \tag{7}$$

where $g(x,u) = f(x,u) - Ax$ with A a Hurwitz matrix. Based on (7), a recurrent network model is constructed by parameterizing the mapping g by feedforward (static) neural network architectures, denoted by N. Therefore, the following model is considered for observer design.

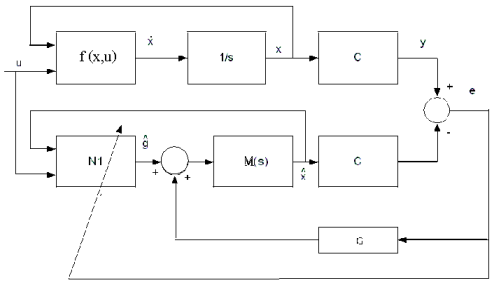

Fig. 1. Structure of the neural network estimator.

$$\dot{\hat{x}} = Ax + g(x,u) + \hat{T}_F(x,w), \tag{8}$$

where

$$\hat{T}_F(x,w) = \begin{bmatrix} 0 \\ H^{-1}\hat{\tau}_F(x,w) \end{bmatrix} \tag{9}$$

and $\hat{\tau}_F(x,w)$ is the mapping performed by the neural network (N). The structure of the estimator is shown in Figure 1. Corresponding to the Hurwitz matrix A, $M(s) := (sI - A)^{-1}$ is also shown which is an $n \times n$ matrix whose elements are stable transfer functions. Similar to [7], the neural network here is employed to identify the fault and produce zero output in non-faulty operation. When there is a fault, the output of the neural network $\hat{\tau}_F$ is nonzero and that is used as a residual to identify the type of fault. It has been shown by many researchers (e.g. see [16] and [17]) that for x restricted to a compact set S of $x \in R^n$ and for some sufficiently large number of hidden layer neurons, there exist weights and thresholds such that any continuous function on the compact set S can be represented as

$$\tau_F(x,w) = W\sigma(Vx) + \epsilon(x) \tag{10}$$

where W and V are the weight matrices, $\epsilon(x) \leq \epsilon_N$ is the neural network's bounded approximation error, and $\sigma(.)$ is the transfer function of the hidden neurons which is usually assumed to be a sigmoidal function:

$$\sigma_i(V_i x) = \frac{2}{1 + exp^{-2V_i x}} - 1$$

where V_i is the *ith* row of V, and $\sigma_i(V_i x)$ is the *ith* element of $\sigma(Vx)$. The function $\tau_F(x,w)$ can be approximated as

$$\hat{\tau}_F(x,w) = \hat{W}\sigma(\hat{V}x) \tag{11}$$

The estimator model can now be written as

$$\dot{\hat{x}}(t) = Ax + g(x,u) + \hat{W}\sigma(\hat{V}x) \tag{12}$$

Defining the estimation error as $\tilde{x} = x - \hat{x}$, and using (7), (11) and (12), the error dynamics can be stated as

$$\dot{\tilde{x}}(t) = A\tilde{x} + \tilde{W}\sigma(\hat{V}x) + w(t) \tag{13}$$

where $w(t) = W[\sigma(Vx) - \sigma(\hat{V}x)] + \epsilon(x)$ is a bounded disturbance term, i.e, $\|w(t)\| \leq \bar{w}$ for some positive constant $\bar{w}$, due to the sigmoidal function.

A. Stability Analysis

In this section, the learning rule for updating the weights of the neural network is given and the stability of the fault detection scheme is shown by Lyapunov's direct method.

Theorem 2.1: Consider the manipulator model (6) and the identifier model (7). If the weights of NLPNN are updated according to:

$$\dot{\hat{W}} = -\eta_1(\frac{\partial J}{\partial \hat{W}}) - \rho_1\|\tilde{x}\|\hat{W} \quad (14)$$

$$\dot{\hat{V}} = -\eta_2(\frac{\partial J}{\partial \hat{V}}) - \rho_2\|\tilde{x}\|\hat{V} \quad (15)$$

then $\tilde{x}, \tilde{W}$, and $\tilde{V} \in L_\infty$, i.e., the estimation error and the weights error are all ultimately bounded. In these equations, $\eta_1, \eta_2 > 0$ are the learning rates, ρ_1, ρ_2 are some small positive numbers, and $J = \frac{1}{2}(\tilde{x}^T\tilde{x})$ is the neural network objective function.

It should be noted that the first terms in (14) and (15) are the popular *backpropagation algorithm* and the last terms are the e-modification terms which add extra damping for robustness.

Proof: Let us define:

$$net_{\hat{v}} = \hat{V}x \quad (16)$$

$$net_{\hat{w}} = \hat{W}\sigma(\hat{V}x). \quad (17)$$

Therefore, $\frac{\partial J}{\partial \hat{W}}$ and $\frac{\partial J}{\partial \hat{V}}$ can be computed according to

$$\frac{\partial J}{\partial \hat{W}} = \frac{\partial J}{\partial net_{\hat{w}}}.\frac{\partial net_{\hat{w}}}{\partial \hat{W}}$$
$$\frac{\partial J}{\partial \hat{V}} = \frac{\partial J}{\partial net_{\hat{v}}}.\frac{\partial net_{\hat{v}}}{\partial \hat{V}} \quad (18)$$

where

$$\frac{\partial J}{\partial net_{\hat{w}}} = \frac{\partial J}{\partial \tilde{x}}.\frac{\partial \tilde{x}}{\partial \hat{x}}.\frac{\partial \hat{x}}{\partial net_{\hat{w}}} - \tilde{x}^T.\frac{\partial \hat{x}}{\partial net_{\hat{w}}}$$
$$\frac{\partial J}{\partial net_{\hat{v}}} = \frac{\partial J}{\partial \tilde{x}}.\frac{\partial \tilde{x}}{\partial \hat{x}}.\frac{\partial \hat{x}}{\partial net_{\hat{v}}} - \tilde{x}^T.\frac{\partial \hat{x}}{\partial net_{\hat{v}}} \quad (19)$$

and

$$\frac{\partial net_{\hat{w}}}{\partial \hat{W}} = \sigma(\hat{V}x)$$
$$\frac{\partial net_{\hat{v}}}{\partial \hat{V}} = x \quad (20)$$

We modify the original BP algorithm such that the static approximations of $\frac{\partial \hat{x}}{\partial net_{\hat{w}}}$ and $\frac{\partial \hat{x}}{\partial net_{\hat{v}}}$ ($\dot{\hat{x}} = 0$) can be used. Thus, using (12), (16), and (17), we can write

$$\frac{\partial \hat{x}}{\partial net_{\hat{w}}} = -A^{-1}$$
$$\frac{\partial \hat{x}}{\partial net_{\hat{v}}} = -A^{-1}\hat{W}(I - \Lambda(\hat{V}x)) \quad (21)$$

where $\Lambda(\hat{V}x) = diag\{\sigma_i^2(\hat{V}_i x)\}, i = 1, 2, ..., m$. Now, by substituting (19), (20), and (21) in (14) and (15), we have

$$\dot{\hat{W}} = -\eta_1(\tilde{x}^T A^{-1})^T(\sigma(\hat{V}x))^T - \rho_1\|\tilde{x}\|\hat{W} \quad (22)$$

$$\dot{\hat{V}} = -\eta_2(\tilde{x}^T A^{-1}\hat{W}(I - \Lambda(\hat{V}x)))^T x^T - \rho_2\|\tilde{x}\|\hat{V} \quad (23)$$

Defining $\tilde{W} = W - \hat{W}$ and $\tilde{V} = V - \hat{V}$, where W and V are the fixed ideal weights, the weight error dynamics can be described by:

$$\dot{\tilde{W}} = \eta_1(\tilde{x}^T A^{-1})^T(\sigma(\hat{V}x))^T + \rho_1\|\tilde{x}\|\hat{W} \quad (24)$$

$$\dot{\tilde{V}} = \eta_2(\tilde{x}^T A^{-1}\hat{W}(I - \Lambda(\hat{V}x))^T)x^T + \rho_2\|\tilde{x}\|\hat{V} \quad (25)$$

Now, consider the Lyapunov function candidate

$$L = \frac{1}{2}\tilde{x}^T P\tilde{x} + \frac{1}{2}tr(\tilde{W}^T\tilde{W}) + \frac{1}{2}tr(\tilde{V}^T\tilde{V}) \quad (26)$$

where $P = P^T$ is a positive-definite matrix satisfying

$$A^T P + PA = -Q \quad (27)$$

for the Hurwitz matrix A and some positive-definite matrix Q. The time derivative of (26) is given by

$$\dot{L} = \frac{1}{2}\dot{\tilde{x}}^T P\tilde{x} + \frac{1}{2}\tilde{x}^T P\dot{\tilde{x}} + tr(\tilde{W}^T\dot{\tilde{W}}) + tr(\tilde{V}^T\dot{\tilde{V}}) \quad (28)$$

Using (13), (24), (25), and (27), equation (28) can be written as

$$\begin{aligned}\dot{L} &= -\frac{1}{2}\dot{\tilde{x}}^T Q\tilde{x} + \tilde{x}^T P(\tilde{W}\sigma(\hat{V}x) + w) \\ &+ tr(\tilde{W}^T F_1\tilde{x}\sigma^T(\hat{V}x) + \tilde{W}^T\rho_1\|\tilde{x}\|(W - \tilde{W})) \\ &+ tr(\tilde{V}^T(I - \Lambda(\hat{V}x))^T\hat{W}^T F_2\tilde{x}x^T \\ &+ \tilde{V}^T\rho_2\|\tilde{x}\|(V - \tilde{V}))\end{aligned} \quad (29)$$

where $F_1 = \eta_1 A^{-T}, F_2 = \eta_2 A^{-T}$. In addition, we have

$$\begin{aligned}tr(\tilde{W}^T(W - \tilde{W})) &\leq W_M\|\tilde{W}\| - \|\tilde{W}\|^2 \\ tr(\tilde{V}^T(V - \tilde{V})) &\leq V_M\|\tilde{V}\| - \|\tilde{V}\|^2 \\ tr(\tilde{W}^T F_1\tilde{x}\sigma^T(\hat{V}x)) &\leq \sigma_m\|\tilde{W}^T\|\ \|F_1\|\ \|\tilde{x}\|\end{aligned} \quad (30)$$

where $\|W\| \leq W_M$, $\|V\| \leq V_M$ and $\|\sigma(\hat{V}\bar{x})\| \leq \sigma_m$. Note that the last inequality in (30) is obtained using the fact that, for two column vectors a and b, the following holds:

$$tr(ab^T) = b^T a. \quad (31)$$

By using $\|\hat{W}\| = \|W - \tilde{W}\| \leq W_M + \|\tilde{W}\|$, $1 - \sigma_m^2 \leq 1$, and (31), the following inequality can also be obtained:

$$tr(\tilde{V}^T(I - \Lambda(\hat{V}x))^T\hat{W}^T F_2\tilde{x}x^T) \leq \|\tilde{V}\|(W_M + \|\tilde{W}\|)\|F_2\|\|\tilde{x}\| \quad (32)$$

Using (30) and (32), we can write

$$\begin{aligned}\dot{L} &\leq -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P\|(\|\tilde{W}\|\sigma_m + \bar{w}) \\ &+ \sigma_m \|\tilde{W}\|\|F_1\|\|\tilde{x}\| + (W_M\|\tilde{W}\| - \|\tilde{W}\|^2)\rho_1\|\tilde{x}\| \\ &+ \|\tilde{V}\|\|F_2\|(W_M + \|\tilde{W}\|)\|\tilde{x}\| \\ &+ \rho_2\|\tilde{x}\|(V_M\|\tilde{V}\| - \|\tilde{V}\|^2) \triangleq G \end{aligned} \quad (33)$$

where $\lambda_{min}(Q)$ is the minimum eigenvalue of Q. Ultimate boundedness of the states requires conditions on $\|\tilde{x}\|$, independent of the neural networks weight error, that guarantees the negative definiteness of the time derivative of the Lyapunov function. Towards this end, first the squares on $\|\tilde{W}\|$ and $\|\tilde{V}\|$ should be completed. Thus, by adding $K_1^2\|\tilde{W}\|^2\|\tilde{x}\|$ and $\|\tilde{V}\|^2\|\tilde{x}\|$ to and subtracting from the right hand side of (33), we get

$$\begin{aligned} G &= -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 \\ &+ [(\|P\|\sigma_m + \sigma_m \|F_1\| + \rho_1 W_M)\|\tilde{W}\| \\ &- (\rho_1 K_1^2)\|\tilde{W}\|^2 + \|P\|\bar{w} \\ &+ (\rho_2 V_M + \|F_2\|W_M)\|\tilde{V}\| - (\rho_2 - 1)\|\tilde{V}\|^2 \\ &- (K_1\|\tilde{W}\| - \|\tilde{V}\|)^2]\|\tilde{x}\| \end{aligned} \quad (34)$$

where $K_1 = \frac{\|F_2\|}{2}$. In the next step, $K_2^2\|\tilde{x}\|$ and $K_3^2\|\tilde{x}\|$ are added to and subtracted from (34):

$$\begin{aligned} G &= -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + (\|P\|\bar{w} + (\rho_1 - K_1^2)K_2^2 \\ &+ (\rho_2 - 1)K_3^2 - (\rho_1 - K_1^2)(K_2 - \|\tilde{W}\|)^2 \\ &- (\rho_2 - 1)(K_3 - \|\tilde{V}\|)^2 - (K_1\|\tilde{W}\| - \|\tilde{V}\|)^2)\|\tilde{x}\| \end{aligned}$$

where K_2 and K_3 are defined as

$$\begin{aligned} K_2 &= \frac{\rho_1 W_M + \sigma_m\|F_1\| + \|P\|\sigma_m}{2(\rho_1 - K_1^2)} \\ K_3 &= \frac{\rho_2 V_M + (1 - \sigma_m^2)\|F_2\|W_M}{2(\rho_2 - 1)} \end{aligned}$$

Now assuming $\rho_1 \geq K_1^2$ and $\rho_2 \geq 1$, we get

$$\begin{aligned} \dot{L} &\leq -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|(\|P\|\bar{w} + (\rho_1 - K_1^2)K_2^2 \\ &+ (\rho_2 - 1)K_3^2) \end{aligned}$$

Then, the following condition on $\|\tilde{x}\|$ is obtained that ensures the negative semi-definiteness of $\dot{L}$

$$\|\tilde{x}\| > \frac{2(\|P\|\bar{w} + (\rho_1 - K_1^2)K_2^2 + (\rho_2 - 1)K_3^2)}{\lambda_{min}(Q)} = b \quad (35)$$

In fact, $\dot{L}$ is negative definite outside the ball with radius b described as $\chi = \{\tilde{x} \mid \|\tilde{x}\| > b\}$, and $\tilde{x}$ is uniformly ultimately bounded. The above analysis shows the ultimate boundedness of $\tilde{x}$. To show the boundedness of the weight errors $\tilde{W}$ and $\tilde{V}$, consider (24) and (25) which can be rewritten as

$$\begin{aligned} \dot{\tilde{W}} &= f_1(\tilde{x}, \hat{V}) + \rho_1\|\tilde{x}\|\hat{W} \\ \dot{\tilde{V}} &= f_2(\tilde{x}, \hat{W}, \hat{V}) + \rho_2\|\tilde{x}\|\hat{V} \end{aligned}$$

Now, by using the definition of $\tilde{W}$ and $\tilde{V}$, we get

$$\begin{aligned} \dot{\tilde{W}} &= f_1(\tilde{x}, \hat{V}) + \alpha_1 W - \alpha_1\tilde{W} \quad (36) \\ \dot{\tilde{V}} &= f_2(\tilde{x}, \hat{W}, \hat{V}) + \alpha_2 V - \alpha_2\tilde{V} \quad (37) \end{aligned}$$

where

$$\begin{aligned} f_1(\tilde{x}, \hat{V}) &= \eta_1(\tilde{x}^T A^{-1})^T(\sigma(\hat{V}\hat{\tilde{x}}))^T \\ f_2(\tilde{x}, \hat{W}, \hat{V}) &= \eta_2(\tilde{x}^T A^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\tilde{x}})))^T x^T \\ \alpha_1 &= \rho_1\|\tilde{x}\| \\ \alpha_2 &= \rho_2\|\tilde{x}\| \end{aligned}$$

It can be seen that $f_1(.)$ is bounded since $\tilde{x}$ and $\sigma(\hat{V}\hat{\tilde{x}}))$ are both bounded and A_c is a Hurwitz matrix . Given the fact that, the ideal weight W is also bounded, (36) can be regarded as a linear system with bounded input $(f_1(\tilde{x}, \hat{V}) + \alpha_1 W)$. It is clear that this system is stable since α_1 is positive and the system input remains bounded. Hence, the boundedness of $\tilde{W}$ is also ensured. Given that $\tilde{W} \in L_\infty$, it can be observed that $f_2(.)$ is also bounded since all its arguments are bounded including $\Lambda(.)$ as defined in (21). Consequently, similar analysis shows that (37) also represents a stable bounded input linear system and hence $\tilde{V} \in L_\infty$. The key to above analysis is that $\tilde{V}$ only appears in $f_1(.)$ and $f_2(.)$ as bounded functions ($\sigma(.)$ and $\Lambda(.)$).

III. Simulation Results

Simulation results performed to verify the performance of the proposed method are presented in this section. The following inertia matrix adopted from [18] was used in the simulations:

$$H = \begin{bmatrix} 5.5384 & -.0276 & -.0242 \\ -.0276 & 5.6001 & -.0244 \\ -.0242 & -.0244 & 4.2382 \end{bmatrix} kg.m^2 \quad (38)$$

The neural network has three layers, the input layer has 6 neurons, the hidden layer has 10 neurons with sigmoidal activation functions, and the output layer has 3 neurons with linear transfer functions. The input of the network is x. At first, a non-faulty situation was considered and a $.2sin\ 0.1t$ input trajectory was applied to the satellite. The output of the neural networks and the angular velocities of the body are shown in Figure 2. As can be seen, the output of the network goes to zero confirming that there is no fault in the system. Next, an actuator fault as a constant torque $(1N - m)$ in each channel is introduced in the system. The responses of the fault detection system are shown in Figure 3. It can be observed that the output of the neural network approaches to a nonzero torque showing that there is a fault in the system (in fact the magnitude of the torque is exactly the same as that introduced in the system). The orientation and angular velocities are also shown in Figure 3 which indicates that the states of the neural network follow the states of the system very accurately. In the last simulation, another type of fault, $cos\ t$, was considered. The output of the neural network, body orientation and angular velocities are shown in Figure 4. These figures confirm that the neural

network was able to recognize the existence as well as the type of the fault inserted in the system.

IV. Conclusions

A neural network based fault detection system was presented in this paper. The stable state estimation approach is *nonlinear-in-parameters* and can estimate actuator faults. The neural network weights are updated based on a combination of the backpropagation algorithm and e-modification that guarantees the boundedness of the estimation error. The stability of the overall system was shown by Lyapunov's direct method. The proposed approach can be applied both as an online and an off-line estimator. Simulation results for a satellite attitude control system confirm the reliable performance of the proposed fault detection approach.

References

[1] M.L. Visinsky, J.R. Cavallaro, and I.D. Walker, "Expert system framework for fault detection and fault tolerance in robotics," *Comput. Elect. Eng.*, pp. 65–75, 1994.

[2] W. Bucken and B. Freyermuth, "Fault detection in industrial processes by statistical methods in the example of an industrial robots," *ITG-Fachberichte*, pp. 123–136, 1991.

[3] B. Freyermuth, "An approach to model based fault diagnosis of industrial robots," in *IEEE International Conf. on Robotics and Automation*, pp. 1350–1356, 1991.

[4] R. Tinos, M.H. Terra, and M. Bergerman, "Fault detection and isolation in cooperative manipulators via artificial neural networks," in *IEEE International Conf. on Control Applications*, pp. 492–497, 2001.

[5] M.H. Terra and R. Tinos, "Fault detection and isolation in robotic systems via artificial neural networks," in *IEEE Conf. on Decision and Control*, pp. 1605–1610, 1998.

[6] J.M. Naughton, Y.C. Chen, and J. Jiang, "A neural network application to fault diagnosis for robotics manipulators," in *IEEE International Conf. on Control Applications*, pp. 988–993, 1996.

[7] A.T. Vemuri, M.M. Polycarpou, and S.A. Diakourtis, "Neural network based fault detection in robotic manipulators," *IEEE Trans. on Robotics and Automation*, pp. 342–348, April 1998.

[8] H. Schneider and P.M. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation," *IEEE Trans. on Control System Technology*, pp. 274–282, May 1996.

[9] W. Tian, Q. Li and Z.-H. Jin, "Application of neural networks for sensor fusion in a remote sensing satellite," in *IEEE International Symposium on Intelligent Control*, pp. 234 – 239, 2002.

[10] F.C. Fan, Z. Jin, J. Zhang and W. Tian, "Application of multisensor data fusion based on rbf neural networks for fault diagnosis of sams," in *7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, pp. 1557–1562, 2002.

[11] R. Mehra, S. Seereeram, D. Bayard and F. Hadaegh, "Adaptive kalman filtering, failure detection and identification for spacecraft attitude estimation," in *4th IEEE Conference on Control Applications*, pp. 176–181, 1995.

[12] T. Bak, R. Wisniewski and M. Blanke, "Autonomous attitude determination and control system for the rsted satellite," in *IEEE Aerospace Applications Conference*, pp. 173–186, 1996.

[13] S. Murugesan and P.S. Goel, "A scheme for fault tolerance in earth sensors," *IEEE Transactions on Aerospace and Electronic Systems, vol. 25, no. 1*, pp. 21–30, 1989.

[14] F. Bacconi, D. Angeli and E. Mosca, "Attitude control of asymmetric spacecrafts subject to actuator failures," in *IEEE Conference on Control Applications*, pp. 474–479, 2003.

[15] J.E. Slotine and M.D. Di Benedetto, "Hamiltonian adaptive control of spacecraft," *IEEE Transactions on Automatic Control, Vol. 35, NO. 7*, pp. 848–852, July 1990.

[16] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Int. Joint Conf. on Neural Networks*, pp. 593–605, 1989.

[17] G. Cybenko, "Approximation by superposition of sigmoidal functions," *Mathematics of Control, Signals and Systems,*, pp. 303–314, October 1989.

[18] L. Show, J.C. Juang and C.D. Yang, "Nonlinear h_∞ robust control for satellite large angle attitude maneuvers," in *American Control Conference*, pp. 1357–1362, 2001.

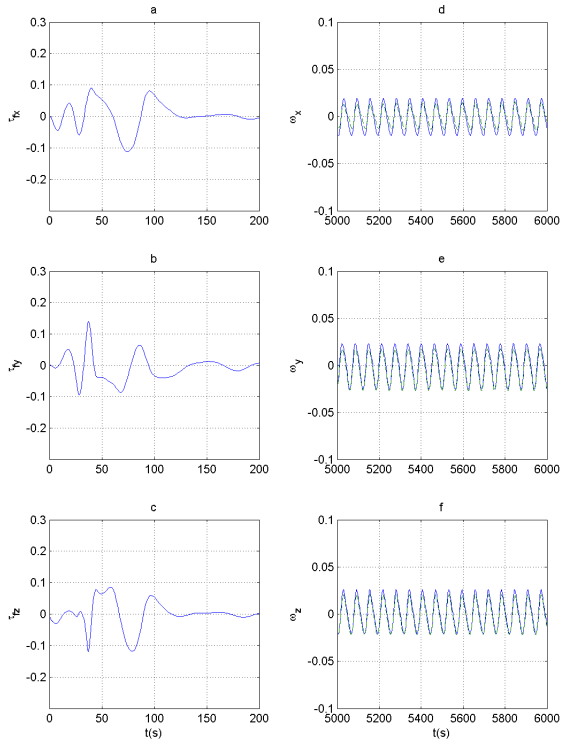

Fig. 2. Angular velocities of the satellite to $0.2sin\ 0.1t$ reference trajectory in fault free operation: (a)-(c) estimated faults; (d)-(f) satellite angular velocities, ω_x, ω_y, ω_z. The solid lines correspond to the actual angular velocities and the dashed lines correspond to their estimates.

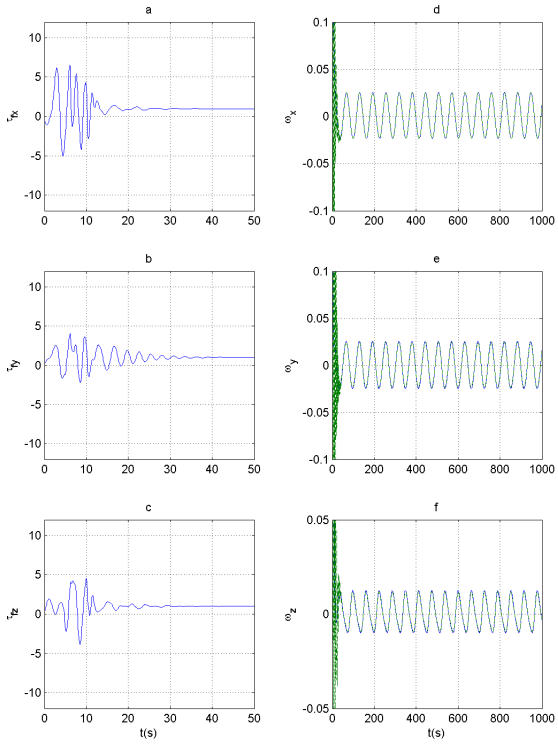

Fig. 3. Angular velocities of the satellite to $0.2sin\ 0.1t$ reference trajectory with 1.0 N-M constant fault in the actuators: (a)-(c) estimated faults; (d)-(f) satellite angular velocities, $\omega_x,\ \omega_y,\ \omega_z$. The solid lines correspond to the actual angular velocities and the dashed lines correspond to their estimates.

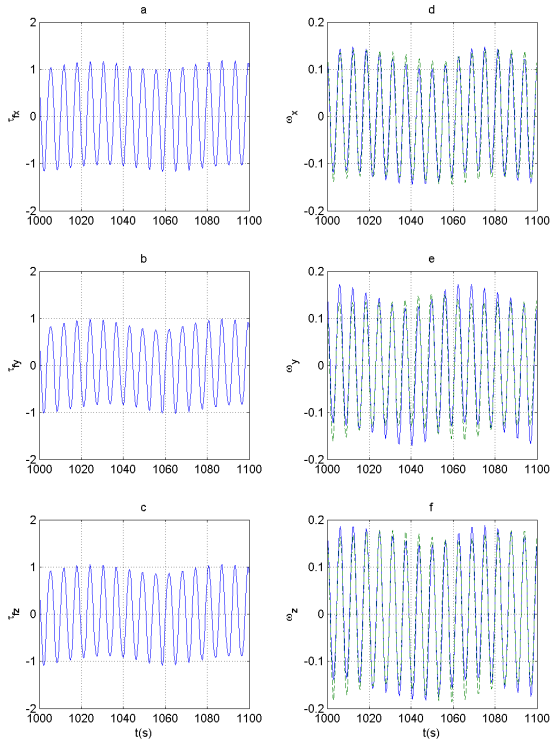

Fig. 4. Angular velocities of the satellite to $0.2sin\ 0.1t$ reference trajectory with 1.0 $cos\ t$ fault in the actuators: (a)-(c) estimated faults; (d)-(f) satellite angular velocities, $\omega_x,\ \omega_y,\ \omega_z$. The solid lines correspond to the actual angular velocities and the dashed lines correspond to their estimates.