# Actuator Fault Estimation Using Neuro-Sliding Mode Observers

Robert Fonod, Dušan Krokavec

# Actuator Fault Estimation Using Neuro-Sliding Mode Observers

Róbert Fónod and Dušan Krokavec
Technical University of Košice, Faculty of Electrical Engineering and Informatics
Department of Cybernetics and Artificial Intelligence
Košice, Slovak Republic
e-mail: robert.fonod@student.tuke.sk, dusan.krokavec@tuke.sk

***Abstract*—Reformulated principle for designing actuator fault estimation for continuous-time linear MIMO systems, based on neuro-sliding mode observer structure, is presented in this paper. Radial basis function neural network is used as a model-free fault approximator of the unknown additive fault. The method utilizes Lyapunov function and the steepest descent rule to guarantee the convergence of the estimation error asymptotically, where the design parameters can be obtained using LMI techniques. Finally, the proposed fault estimation scheme is applied to a nonlinear water tank system and simulation results illustrate its satisfactory performance.**

## I. Introduction

A particular class of observer-based methods uses sliding mode observers (SMO). In SMO a nonlinear discontinuous term is injected into the observer dynamics depending on the output estimation error. These observers are more robust than Luenberger observers, as the discontinuous term enables the observer to reject disturbances, and also a class of mismatch between the system and observer. Supported by existence of linear matrix inequality (LMI) solvers (see e.g. [1]), the intention is to reformulate given problem and optimize it over LMI constraints [2], [3].

Neural networks can be considered as an alternative model-free observer because it offers much potential benefit for nonlinear modeling. The dynamic neural networks have been also applied to design a Luenberger type of observer [4]. Combining sliding mode observer techniques with neural networks to design fault estimation structures offer new opportunity in design [5]. The main idea of this combination is to apply them at the same time, where sliding mode term is used to compensate the neural modeling error and the system uncertainties, while the neural network is used to approximate the unknown (in general nonlinear) fault function.

Radial basis function (RBF) neural networks seem to be one of the neural networks with high approximation and regularization capability [6], where the essential phenomenological rationale for the use of RBF rest in the realm of the purpose of feed-forward networks and feature extraction possibility. Gaussian RBF are employed the most frequently, since it is bounded, strictly positive and continuous on $I\!R$, and the optimization of RBF networks provides better approximation and interpolation capability as compared to the sigmoid functions. However, the performance of RBF neural networks (RBFNN) depends on the number of neurons in hidden layer, and on methods used for determining the network weights, where training of a RBFNN is, in general, a challenging nonlinear optimization problem.

Actuator faults detection and diagnosis (FDD) algorithm for linear systems is presented in this paper. The FDD scheme is based on neuro-sliding mode observer [7], which realizes a fault estimation using RBF neural network. When a fault occurs, the neural part in the observer is enabled. The switching function avoids the unnecessary chattering caused by sliding mode phenomena.

The main contribution of the paper is to adapt the work presented in [8], [9] to actuator faults, acting on the system in additive form. Moreover, the idea is extended to a class of continuous-time linear Multi-Input Multi-Output (MIMO) systems, using a bank of observers. Under defined conditions, the stability of the observer is assured, where the state observer error is bounded. In comparison with methods based on analytical models (see e.g. [2], [10], [11]) design conditions for specified system class are here derived using learning approach and sliding mode technique.

## II. Problem Description

Consider a state-space representation of the linear MIMO system in canonical form [3] with only actuator faults

$$\begin{aligned}\dot{\boldsymbol{q}}(t) &= \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\big(\boldsymbol{u}(t) + \boldsymbol{f}_a(t)\big) = \\ &= \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \sum_{i=1}^{r} \boldsymbol{b}_i f_a^i(\boldsymbol{q}(t))\end{aligned} \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) \tag{2}$$

where $\boldsymbol{q}(t) \in I\!R^n$, $\boldsymbol{u}(t) \in I\!R^r$, and $\boldsymbol{y}(t) \in I\!R^m$ are the state, input and output variables, respectively, and $\boldsymbol{A} \in I\!R^{n\times n}$, $\boldsymbol{B} \in I\!R^{n\times r}$, $\boldsymbol{C} \in I\!R^{m\times n}$ are real matrix parameters of the system. Actuator faults are modelled by an unknown additive vector function $\boldsymbol{f}_a(\boldsymbol{q}(t)) \in I\!R^r$, $\boldsymbol{b}_i \in I\!R^n$ is the $i^{th}$ column vector of the input matrix $\boldsymbol{B}$, and $f_a^i(\boldsymbol{q}(t)) \in I\!R$ represents the $i^{th}$ actuator fault.

The task is to design a neuro-sliding mode observer (NSMO) based estimation of the actuator faults using RBF neural networks.

## III. Neuro-Sliding Mode Observer Structure

The output variable $\boldsymbol{y}(t)$ and the output matrix $\boldsymbol{C}$ can be expressed as

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_1^T \\ \boldsymbol{c}_2^T \\ \vdots \\ \boldsymbol{c}_m^T \end{bmatrix} \boldsymbol{q}(t) \tag{3}$$

Thus, the system given by (1), (2) can be reduced, for the fault estimation purposes, in the following form

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \sum_{i=1}^{r} \boldsymbol{b}_i f_a^i(\boldsymbol{q}(t)) \tag{4}$$

$$y_j(t) = \boldsymbol{c}_j^T \boldsymbol{q}(t) \tag{5}$$

where the $y_j(t)$ is the $j^{th}$ output variable of the system, and $\boldsymbol{c}_j^T$ has to satisfy

$$\text{rank}\begin{bmatrix} \boldsymbol{c}_j & \boldsymbol{A}^T\boldsymbol{c}_j & \dots & (\boldsymbol{A}^T)^{n-1}\boldsymbol{c}_j \end{bmatrix}^T = n \tag{6}$$

i.e. the pair $(\boldsymbol{A}, \boldsymbol{c}_j^T)$ has to be an observable pair (see e.g. [10]).

In the next, we consider single actuator faults, i.e. only one actuator fault is occurring in the system. The particular NSMO of the simplified system (4), (5) for the $i^{th}$ actuator fault $f_a^i(\boldsymbol{q}(t))$ is expressed in the form

$$\begin{aligned} \dot{\boldsymbol{q}}_e(t) = \boldsymbol{A}\boldsymbol{q}_e(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{l}(y_j(t) - y_e(t)) + \\ + \boldsymbol{b}_i \hat{f}_a^i(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) + (1 - r(t))\boldsymbol{h}\nu(t) \end{aligned} \tag{7}$$

$$y_e(t) = \boldsymbol{c}_j^T \boldsymbol{q}_e(t) \tag{8}$$

where $\boldsymbol{q}_e(t) \in I\!R^n$ is the state observer variable, $y_e(t) \in I\!R$ is the estimated output, $\boldsymbol{l} \in I\!R^n$ is the traditional Luenberger observer gain vector to be designed, $\nu(t) \in I\!R$ is an external feed-forward compensation signal, $\hat{f}_a^i(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) \in I\!R$ is the unknown actuator fault approximation using RBFNN, and $r(t)$ is a switch function.

If $\text{rank}(\boldsymbol{B}) = r$, then for each actuator fault $f_a^i(\boldsymbol{q}(t))$, it is necessary to construct an own observer of the structure (7), (8), i.e. if all actuator faults are monitored, then a bank of $r$ observers (see Fig. 1) have to be designed.

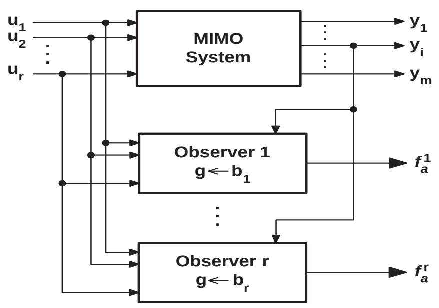


Fig. 1. Bank of observers for all actuator faults estimation

In the next, for the sake of simplicity, the index $i$ will be omitted, and the following equivalences are hold

$$f_a(\boldsymbol{q}(t)) \Leftrightarrow f_a^i(\boldsymbol{q}(t)) \tag{9}$$

$$\hat{f}_a(\boldsymbol{q}(t)|\boldsymbol{w}_a(t)) \Leftrightarrow \hat{f}_a^i(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) \tag{10}$$

$$\boldsymbol{g} \Leftrightarrow \boldsymbol{b}_i \tag{11}$$

The state $\boldsymbol{e}_q(t)$ and the output $e_y(t)$ estimation error are defined as follows

$$\boldsymbol{e}_q(t) = \boldsymbol{q}(t) - \boldsymbol{q}_e(t), \quad e_y(t) = y_j(t) - y_e(t) = \boldsymbol{c}_j^T \boldsymbol{e}_q(t) \tag{12}$$

The switch function, which switches between the pure neural and sliding mode observer in the dependence on the output error, i.e. with $a = \boldsymbol{m}^T\boldsymbol{m}$

$$r(t) = \begin{cases} 1, & \text{if} \|e_y(t)\|_a^2 \geq \tau; \quad \text{faulty} \\ 0, & \text{if} \|e_y(t)\|_a^2 < \tau; \quad \text{healthy} \end{cases} \tag{13}$$

where $\tau$ is a pre-specified threshold of the weighted modelling error, and $\boldsymbol{m}$ will be defined formally later.

## IV. Neural Observer Structure and Stability

If $r(t) = 1$, then (7), (8) becomes

$$\begin{aligned} \dot{\boldsymbol{q}}_e(t) = \boldsymbol{A}\boldsymbol{q}_e(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{l}(y_j(t) - y_e(t)) + \\ + \boldsymbol{g}\hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) \end{aligned} \tag{14}$$

$$y_e(t) = \boldsymbol{c}_j^T \boldsymbol{q}_e(t) \tag{15}$$

and the observer is pure neural observer.

### *A. RBF Neural Networks*

The model-free fault estimation employed in the next use a radial basis function neural network (RBFNN) to approximate the unknown actuator fault $f_a(\boldsymbol{q}(t))$. Supposing that there are $p$ receptive field units in the neural network hidden layer then the output of the RBFNN is

$$\hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) = \sum_{h=1}^{p} w_h(t)\varphi_h(\boldsymbol{q}_e(t)) = \boldsymbol{w}_a^T(t)\boldsymbol{\varphi}(\boldsymbol{q}_e(t)) \tag{16}$$

$$\varphi_h(\boldsymbol{q}_e(t)) = \exp\left(-\frac{\|\boldsymbol{q}_e(t) - \boldsymbol{c}_h^\diamond\|^2}{\sigma_h^2}\right) \tag{17}$$

$$\boldsymbol{w}_a^T(t) = \begin{bmatrix} w_1(t) & w_2(t) & \cdots & w_p(t) \end{bmatrix} \tag{18}$$

$$\boldsymbol{\varphi}^T(\boldsymbol{q}_e(t)) = \begin{bmatrix} \varphi_1(\boldsymbol{q}_e(t)) & \varphi_2(\boldsymbol{q}_e(t)) & \cdots & \varphi_p(\boldsymbol{q}_e(t)) \end{bmatrix} \tag{19}$$

where $\boldsymbol{c}_h^\diamond \in I\!R^n$, $0 < \sigma_h \in I\!R$ are the center and width (spread factor) of the neural cell of the $h^{th}$ neuron in the hidden layer, respectively, and $w_h(t) \in I\!R$ is the weight connecting the $h^{th}$ hidden layer neuron and the network output.

Thus, the optimal weight values of RBF neural network can be expressed as follows

$$\boldsymbol{w}^\star = \arg \min_{\boldsymbol{w}_a \in \Omega_f} \left( \sup_{\boldsymbol{q}_e \in \mathcal{S}_q} |f(\boldsymbol{q}_e(t)|\boldsymbol{w}_a) - f(\boldsymbol{q}_e(t))| \right) \tag{20}$$

where

$$\Omega_f = \{\boldsymbol{w}_a : \|\boldsymbol{w}_a\| \leq \delta_w\} \tag{21}$$

is a valid field of $\boldsymbol{w}_a(t)$, $\delta_w$ is a upper bound parameter, and $\mathcal{S}_q \subseteq I\!R^n$ is a variable space of $\boldsymbol{q}_e(t) \in \mathcal{S}_q$.

Radial basis function $\boldsymbol{\varphi}(\boldsymbol{q}_e(t))$ satisfies the so-called Lipschitz condition

$$\|\boldsymbol{\varphi}(\boldsymbol{q}(t))-\boldsymbol{\varphi}(\boldsymbol{q}_e(t))\|^2 \leq \kappa_\varphi \|\boldsymbol{q}(t)-\boldsymbol{q}_e(t)\|^2 \tag{22}$$

where $\|\cdot\|$ denotes Euclidian norm of a vector, and $\kappa_\varphi > 0$ is the Lipschitz constant of (17).

Defining the RBFNN approximation error as

$$\psi(t) = f_a(\boldsymbol{q}(t)) - \hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t)) \tag{23}$$

then, according to the Stone-Weierstrass theorem, the smooth function $f_a(\boldsymbol{q}(t))$ can be written as

$$f_a(\boldsymbol{q}(t)) = \boldsymbol{w}^{\star T}\boldsymbol{\varphi}(\boldsymbol{q}(t)) + \varepsilon(t) \tag{24}$$

where $\boldsymbol{w}^{\star T}$ is the fixed weight vector (20), and $\varepsilon(t)$ is is the smallest approximation (modelling) error satisfying condition

$$\varepsilon^2(t) \leq \delta \tag{25}$$

Then, according to (16), (23) is

$$\begin{aligned}\psi(t) &= \boldsymbol{w}^{\star T}\boldsymbol{\varphi}(\boldsymbol{q}(t)) + \varepsilon(t) - \boldsymbol{w}_a^T(t)\boldsymbol{\varphi}(\boldsymbol{q}_e(t)) = \\ &= \boldsymbol{w}^{\star T}(\boldsymbol{\varphi}(\boldsymbol{q}(t)) - \boldsymbol{\varphi}(\boldsymbol{q}_e(t))) + \varepsilon(t) + \\ &\quad + (\boldsymbol{w}^{\star T} - \boldsymbol{w}_a^T(t))\boldsymbol{\varphi}(\boldsymbol{q}_e(t))\end{aligned} \tag{26}$$

with notations

$$\Delta\boldsymbol{\varphi}(t) = \boldsymbol{\varphi}(\boldsymbol{q}(t)) - \boldsymbol{\varphi}(\boldsymbol{q}_e(t)) \tag{27}$$

$$\boldsymbol{w}_e^T(t) = \boldsymbol{w}^{\star T} - \boldsymbol{w}_a^T(t) \tag{28}$$

it yields

$$\psi(t) = \boldsymbol{w}^{\star T}\Delta\boldsymbol{\varphi}(t) + \boldsymbol{w}_e^T(t)\boldsymbol{\varphi}(\boldsymbol{q}_e(t)) + \varepsilon(t) \tag{29}$$

### *B. Neural Observer Error Dynamics*

Combining (4), (12), (14) and (29) implies the neural observer state error estimation dynamic

$$\begin{aligned}\dot{\boldsymbol{e}}_q(t) &= \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) - \boldsymbol{A}\boldsymbol{q}_e(t) - \boldsymbol{B}\boldsymbol{u}(t) - \\ &- \boldsymbol{l}(y_j(t) - y_e(t)) + \boldsymbol{g}\big(f_a(\boldsymbol{q}(t)) - \hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a(t))\big) = \\ &= (\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c}_j^T)\boldsymbol{e}_q(t) + \boldsymbol{g}\psi(t)\end{aligned} \tag{30}$$

$$\dot{\boldsymbol{e}}_q(t) = \boldsymbol{A}_e + \boldsymbol{g}\big(\boldsymbol{w}^{*T}\Delta\boldsymbol{\varphi}(t) + \boldsymbol{w}_e^T(t)\boldsymbol{\varphi}(\boldsymbol{q}_e(t)) + \varepsilon(t)\big) \tag{31}$$

respectively, where $\boldsymbol{A}_e = \boldsymbol{A} - \boldsymbol{l}\boldsymbol{c}_j^T$. Since only $e_y(t)$ is available, then (12) can be modified as

$$\boldsymbol{c}_j e_y(t) = (\boldsymbol{c}_j\boldsymbol{c}_j^T + \lambda\boldsymbol{I})\boldsymbol{e}_q(t) - \lambda\boldsymbol{I}\boldsymbol{e}_q(t) \tag{32}$$

where $0 < \lambda \in I\!R$, and

$$\boldsymbol{e}_q(t) = (\boldsymbol{c}_j\boldsymbol{c}_j^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{c}_j e_y(t) + \lambda(\boldsymbol{c}_j\boldsymbol{c}_j^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{e}_q(t) \tag{33}$$

$$\boldsymbol{e}_q(t) = \boldsymbol{m}e_y(t) + \lambda\boldsymbol{N}\boldsymbol{e}_q(t) \tag{34}$$

$$\boldsymbol{m}e_y(t) = (\boldsymbol{I} - \lambda\boldsymbol{N})\boldsymbol{e}_q(t) \tag{35}$$

respectively, where

$$\boldsymbol{N} = (\boldsymbol{c}_j\boldsymbol{c}_j^T + \lambda\boldsymbol{I})^{-1}, \quad \boldsymbol{m} = \boldsymbol{N}\boldsymbol{c}_j, \quad \boldsymbol{\Upsilon} = \boldsymbol{I} - \lambda\boldsymbol{N} \tag{36}$$

### *C. Neural Network Updating Law*

Since NN have to be discrete updated, and only $e_y(t)$ can be used in the updating law, the neural network weights can be determined by using the steep descent method

$$\boldsymbol{w}_e(i+1) = \boldsymbol{w}_e(i) - r(i)\mu(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(t))\boldsymbol{g}^T\boldsymbol{m}e_y(t) \tag{37}$$

where $\mu(i) > 0$ is the neural network learning rate.

### *D. Neural Observer Stability*

Defining Lyapunov function candidate as

$$v(\boldsymbol{w}_e(i)) = \boldsymbol{w}_e^T(i)\boldsymbol{w}_e(i) = \|\boldsymbol{w}_e(i)\|^2 \tag{38}$$

then with $r(i) = 1$ yields

$$\begin{aligned}\Delta v(\boldsymbol{w}_e(i)) &= \|\boldsymbol{w}_e(i+1)\|^2 - \|\boldsymbol{w}_e(i)\|^2 = \\ &= \left\|\boldsymbol{w}_e(i) - \mu(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\boldsymbol{g}^T\boldsymbol{m}e_y(i)\right\|^2 - \|\boldsymbol{w}_e(i)\|^2 < 0\end{aligned} \tag{39}$$

$$\begin{aligned}\Delta v(\boldsymbol{w}_e(i)) &= \mu^2(i)\left\|\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\boldsymbol{g}^T\right\|^2\|\boldsymbol{m}e_y(i)\|^2 - \\ &- 2\mu(i)\left\|\boldsymbol{w}_e^T(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\boldsymbol{g}^T\boldsymbol{m}e_y(i)\right\| \leq \\ &\leq \alpha(i) - \mu(i)\xi\beta(i) < 0\end{aligned} \tag{40}$$

respectively, where

$$\alpha(i) = \mu^2(i)\left\|\boldsymbol{g}\boldsymbol{\varphi}^T(\boldsymbol{q}_e(i))\right\|^2\|\boldsymbol{m}e_y(i)\|^2 \tag{41}$$

$$\beta(i) = 2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{g}\boldsymbol{w}_e^T(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\right\|, \quad \xi = \|\boldsymbol{\Upsilon}\| \tag{42}$$

$$\left\|\boldsymbol{w}_e^T(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\boldsymbol{g}^T\right\| = \left\|\boldsymbol{g}\boldsymbol{w}_e^T(i)\boldsymbol{\varphi}(\boldsymbol{q}_e(i))\right\| \tag{43}$$

Since from the Lyapunov theory of stability [12] yields $\boldsymbol{e}_q^T(i)\dot{\boldsymbol{e}}_q(i) < 0$, then (43) with (31) gives

$$\begin{aligned}&\beta(i) = \\ &= 2\left\|\boldsymbol{e}_q^T(i)\big(\dot{\boldsymbol{e}}_q(i) - \boldsymbol{A}_e\boldsymbol{e}_q(i) - \boldsymbol{g}(\varepsilon(i) + \boldsymbol{w}^{\star T}\Delta\boldsymbol{\varphi}(i))\big)\right\| \leq \\ &\leq \left\{\begin{array}{c} 2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{g}\boldsymbol{w}^{\star T}\Delta\boldsymbol{\varphi}(i)\right\| - 2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{A}_e\boldsymbol{e}_q(i)\right\| - \\ -2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{g}\varepsilon(i)\right\| - 2\left\|\boldsymbol{e}_q^T(i)\dot{\boldsymbol{e}}_q(i)\right\| \end{array}\right\}\end{aligned} \tag{44}$$

For a smooth stable system and any $i > 0$ yields

$$2\|\dot{\boldsymbol{e}}_q(i)\| \leq i\|\boldsymbol{e}_q(i)\| \tag{45}$$

then

$$-2\left\|\boldsymbol{e}_q^T(i)\dot{\boldsymbol{e}}_q(i)\right\| \geq -i\|\boldsymbol{e}_q(i)\|^2 \tag{46}$$

Since $\boldsymbol{A}_e < 0$ is designed to be stable, then second term in (44) can be rewritten

$$-2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{A}_e\boldsymbol{e}_q(i)\right\| \leq -\sigma_{max}(\boldsymbol{A}_e)\|\boldsymbol{e}_q(i)\|^2 \tag{47}$$

where $\sigma_{max}(\boldsymbol{A}_e)$ is the largest singular value of the matrix $\boldsymbol{A}_e$. Third term in (44) can be rewritten in the form

$$\begin{aligned}-2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{g}\varepsilon(i)\right\| &\geq -\|\boldsymbol{e}_q(i)\|^2 - \|\boldsymbol{g}\|\,\|\varepsilon(i)\|^2 = \\ &= -\|\boldsymbol{e}_q(i)\|^2 - \|\boldsymbol{g}\|\,\delta\end{aligned} \tag{48}$$

It is obvious that (22) implies

$$\begin{aligned}2\left\|\boldsymbol{e}_q^T(i)\boldsymbol{g}\boldsymbol{w}^{\star T}\Delta\boldsymbol{\varphi}(i)\right\| &\leq \|\boldsymbol{e}_q(i)\|^2 + \|\boldsymbol{g}\boldsymbol{w}^{\star T}\|\|\Delta\boldsymbol{\varphi}(i))\|^2 \leq \\ &\leq \big(1 + \kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\|\big)\|\boldsymbol{e}_q(i)\|^2\end{aligned} \tag{49}$$

then with notation $\sigma_{max}(\boldsymbol{A}_e) = \sigma_m$

$$\begin{aligned}\mu(i)\xi\beta(i) \le -\mu(i)\xi\|\boldsymbol{g}\|\delta+ \\ +\mu(i)\xi\left(\kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\| - \sigma_m - i\right)\|\boldsymbol{e}_q(i)\|^2\end{aligned} \tag{50}$$

Thus, combining (41) and (50) gives

$$\begin{aligned}\Delta v(\boldsymbol{w}_e(i)) = \alpha(i) - \mu(i)\xi\beta(i) = \mu(i)\xi\|\boldsymbol{g}\|\delta- \\ -\mu(i)\begin{pmatrix}-\mu(i)\left(\|\boldsymbol{g}\boldsymbol{\varphi}^T(\boldsymbol{q}_e(i))\|^2+1-1\right)\|\boldsymbol{m}e_y(i)\|^2+ \\ +\left(\kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\| - \sigma_m - i\right)\xi\|\boldsymbol{e}_q(i)\|^2\end{pmatrix}\end{aligned} \tag{51}$$

Setting

$$\mu(i) = \frac{\kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\| - \sigma_m - i}{1 + \|\boldsymbol{g}\boldsymbol{\varphi}^T(\boldsymbol{q}_e(i))\|^2} \tag{52}$$

then the Lyapunov function difference (51) takes the form

$$\Delta v(\boldsymbol{w}_e(i)) = \mu(i)\xi\|\boldsymbol{g}\|\delta - \mu^2(i)\|\boldsymbol{m}e_y(i)\|^2 < 0 \tag{53}$$

$$\Delta v(\boldsymbol{w}_e(i)) = -\mu(i)\left(\mu(i)\|e_y(i)\|_a^2 - \xi\|\boldsymbol{g}\|\delta\right) < 0 \tag{54}$$

Because $\|e_y(i)\|_a^2 \ge \tau$, it is enough to select, such a constant $\delta$, that the sign inside the brackets of (54) is positive, then the neural network is stable trained.

Here the initial condition $\boldsymbol{w}^{\star T}$ can be selected as

$$\kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\| < 1 + i + \sigma_m \tag{55}$$

which gives

$$0 < \kappa_\varphi\|\boldsymbol{g}\boldsymbol{w}^{\star T}\| - i - \sigma_m < 1 \tag{56}$$

and it is evident that (52) implies $0 < \mu(i) < 1$.

## V. Sliding Mode Observer Structure and Stability

If at time $t = t_0$ is $r(t) = 0$, then the weights become constant, i.e. $\boldsymbol{w}_a = \boldsymbol{w}_a(t_0)$, and the observer (7), (8) becomes pure sliding mode taking the form

$$\begin{aligned}\dot{\boldsymbol{q}}_e(t) = \boldsymbol{A}\boldsymbol{q}_e(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{l}(y_j(t) - y_e(t))+ \\ +\boldsymbol{g}\hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a) + \boldsymbol{h}\nu(t)\end{aligned} \tag{57}$$

$$y_e(t) = \boldsymbol{c}_j^T\boldsymbol{q}_e(t) \tag{58}$$

The discontinuous term $\nu(t)$ enters the observer dynamics through the design vector $\boldsymbol{h} \in I\!R^n$, and is defined by

$$\nu(t) = \begin{cases} -\rho(t)\operatorname{sign}(e_y(t)) = -\rho(t)\frac{e_y(t)}{\|e_y(t)\|} & \text{if } e_y \ne 0, \\ 0 & \text{otherwise} \end{cases} \tag{59}$$

where $\rho(t) > 0, \rho(t) \in I\!R$ is the scalar gain function, high enough to enforce the sliding motion.

### A. Sliding Mode Observer Stability Conditions

The error dynamics obtained from (4), (12) and (57) is

$$\dot{\boldsymbol{e}}_q(t) = \left(\boldsymbol{A} - \boldsymbol{l}\boldsymbol{c}_j^T\right)\boldsymbol{e}_q(t) + \boldsymbol{g}\Delta f_a(t) - \boldsymbol{h}\nu(t) \tag{60}$$

where

$$\Delta f_a(t) = f_a(\boldsymbol{q}(t)) - \hat{f}_a(\boldsymbol{q}_e(t)|\boldsymbol{w}_a) \tag{61}$$

is the neural modelling error when the weights of the neural network are fixed as $\boldsymbol{w}_a$.

Defining the Lyapunov function of the form

$$v\big(\boldsymbol{e}_q(t)\big) = \boldsymbol{e}_q^T(t)\boldsymbol{P}\boldsymbol{e}_q(t) > 0 \tag{62}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0, \boldsymbol{P} \in I\!R^{n\times}$, and taking the time derivative of $v\big(\boldsymbol{e}_q(t)\big)$ results in

$$\dot{v}\big(\boldsymbol{e}_q(t)\big) = \dot{\boldsymbol{e}}_q^T(t)\boldsymbol{P}\boldsymbol{e}_q(t) + \boldsymbol{e}_q^T(t)\boldsymbol{P}\dot{\boldsymbol{e}}_q(t) < 0 \tag{63}$$

inserting (60) into (63) gives

$$\dot{v}\big(\boldsymbol{e}_q(t)\big) = \boldsymbol{e}_q^{\circ T}(t)\boldsymbol{P}^\bullet\boldsymbol{e}_q^\circ(t) < 0 \tag{64}$$

where

$$\boldsymbol{e}_q^{\circ T}(t) = \begin{bmatrix} \boldsymbol{e}_q^T(t) & \Delta f_a(t) & \nu(t) \end{bmatrix} \tag{65}$$

$$\boldsymbol{P}^\bullet = \begin{bmatrix} \boldsymbol{P}(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c}_j^T)+(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c}_j^T)^T\boldsymbol{P} & \boldsymbol{P}\boldsymbol{g} & -\boldsymbol{P}\boldsymbol{h} \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} < 0 \tag{66}$$

(Hereafter, $*$ denotes the symmetric item in a symmetric matrix).

Defining the congruence transform matrix

$$\boldsymbol{T}_q = \begin{bmatrix} \boldsymbol{I} & & \\ & \boldsymbol{I} & \boldsymbol{I} \\ & & \boldsymbol{0} \end{bmatrix} \tag{67}$$

then pre-multiplying left-hand side of (66) by $\boldsymbol{T}_q$, and right-hand side of (66) by $\boldsymbol{T}_q^T$ gives

$$\begin{aligned}\boldsymbol{P}^\circ = \boldsymbol{T}_q\boldsymbol{P}^\bullet\boldsymbol{T}_q^T = \\ = \begin{bmatrix} \boldsymbol{P}(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c}_j^T)+(\boldsymbol{A}-\boldsymbol{l}\boldsymbol{c}_j^T)^T\boldsymbol{P} & \boldsymbol{P}(\boldsymbol{g}-\boldsymbol{h}) \\ * & 0 \end{bmatrix}\end{aligned} \tag{68}$$

Since (68) can be rewritten as

$$\begin{aligned}\begin{bmatrix} \boldsymbol{P}\boldsymbol{A}+\boldsymbol{A}^T\boldsymbol{P} & \boldsymbol{P}(\boldsymbol{g}-\boldsymbol{h}) \\ * & 0 \end{bmatrix} - \\ -\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{l}\begin{bmatrix} \boldsymbol{c}_j^T & \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c}_j \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{l}^T\begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \end{bmatrix} < 0\end{aligned} \tag{69}$$

then using the orthogonal complement matrix

$$\boldsymbol{c}_j^{\circ\perp} = \begin{bmatrix} \boldsymbol{c}_j \\ \boldsymbol{0} \end{bmatrix}^\perp = \begin{bmatrix} \boldsymbol{c}_j^\perp & \\ & \boldsymbol{I} \end{bmatrix} \tag{70}$$

and pre-multiplying left side of (69) by $\boldsymbol{c}_j^{\circ\perp}$ and right side of (69) by $\boldsymbol{c}_j^{\circ\perp T}$ gives

$$\begin{bmatrix} \boldsymbol{c}_j^\perp\left(\boldsymbol{P}\boldsymbol{A}+\boldsymbol{A}^T\boldsymbol{P}\right)\boldsymbol{c}_j^{\perp T} & \boldsymbol{c}_j^\perp\boldsymbol{P}(\boldsymbol{g}-\boldsymbol{h}) \\ * & 0 \end{bmatrix} < 0 \tag{71}$$

Setting

$$\boldsymbol{h} = \boldsymbol{g} \tag{72}$$

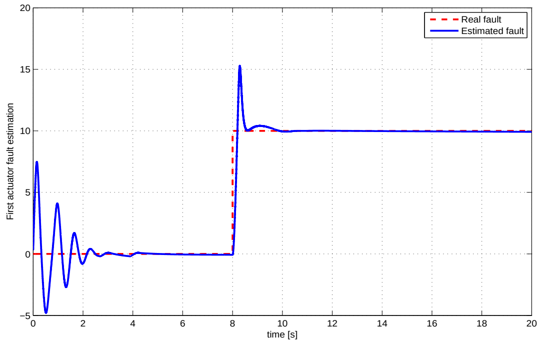


Fig. 2. First actuator fault and its estimation

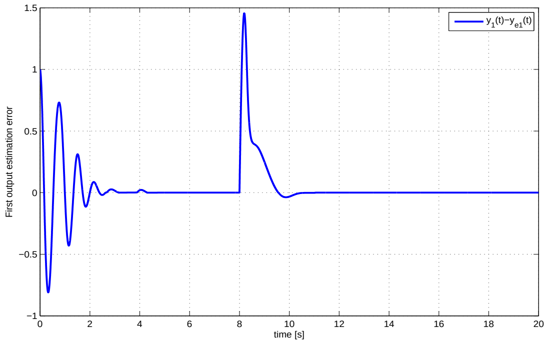


Fig. 3. First output estimation error (residual)

then (71) implies

$$c_j^{\perp}\left(\boldsymbol{P}\boldsymbol{A}+\boldsymbol{A}^T\boldsymbol{P}\right)c_j^{\perp T}<0, \quad \boldsymbol{P}=\boldsymbol{P}^T>0 \tag{73}$$

which gives sufficient condition for the stable sliding mode existence.

## VI. Design Conditions

The observer gain vector $\boldsymbol{l}$ can not be directly computed from (73), hence introducing a new LMI variable $\boldsymbol{z}\in I\!R^n$ and using assignment

$$\boldsymbol{z}=\boldsymbol{P}\boldsymbol{l} \tag{74}$$

then (69) implies

$$\boldsymbol{P}\boldsymbol{A}+\boldsymbol{A}^T\boldsymbol{P}-\boldsymbol{z}\boldsymbol{c}_j^T-\boldsymbol{c}_j\boldsymbol{z}^T<0 \tag{75}$$

The observer gain vector $\boldsymbol{l}$ can be set as

$$\boldsymbol{l}=\boldsymbol{P}^{-1}\boldsymbol{z} \tag{76}$$

Solving the above specified inequalities (73), (75) with respect to the LMI variables $\boldsymbol{P},\boldsymbol{z}$ the sliding mode observer (57), (58) is asymptotically stable. The observer parameters are then given by (72) and (76).

The sliding mode neural observer (7), (8) requires two additional design parameters: switch constant $\tau$ and the upper bound $\Delta$ of neural modelling error $\Delta f_a(t)$ when start the sliding mode compensation. Here, $\tau$ decide when the neural network learning is stopped and sliding mode observer is started. The bigger $\tau$, the shorter training time the neural observer has and since the neural modelling error is bigger, so $\Delta$ should be bigger. Usually $\Delta>\tau$, because $\tau$ corresponds to the modelling error with optimal synaptic weights, while $\Delta$ corresponds to the modelling error when $\|e_y(i)\|_a^2<\tau$.

## VII. Illustrative Example

Considering a nonlinear model of three coupled water tank system [13], linearized around the working points and transformed into the canonical form results in

$$\boldsymbol{A}=\begin{bmatrix}-0.0156 & 0.0078 & 0.0078\\ 0.0044 & -0.0044 & 0.0000\\ 0.0044 & 0.0000 & -0.0044\end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix}0.000 & 0.000\\ 1.234 & 0.000\\ 0.000 & 1.234\end{bmatrix}, \quad \boldsymbol{C}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

A bank of two observers have to be designed (see Fig.1 with $r=2$). Using $\boldsymbol{c}_1^T$ for the first, and $\boldsymbol{c}_2^T$ for the second observer, and solving the inequalities (73) with respect to the LMI variables $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, the problem was solved as feasible with Self-Dual-Minimization (SeDuMi) package for Matlab [1], and giving the solutions

$$\boldsymbol{P}_1=\begin{bmatrix}0.8143 & -0.3823 & 0\\ 0.1554 & 0.5816 & 0\\ 0 & 0 & 1\end{bmatrix} \quad \boldsymbol{P}_2=\begin{bmatrix}1.6477 & 0.2545 & 0\\ -0.2309 & 1.6634 & 0\\ 0 & 0 & 1\end{bmatrix}$$

consequently solving the second inequality (75) with respect to LMI vector variables $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$

$$\boldsymbol{z}_1=\begin{bmatrix}0.0077 & 0.1474 & 0.0014\end{bmatrix}^T$$
$$\boldsymbol{z}_2=\begin{bmatrix}0.0160 & -0.0011 & 0.1522\end{bmatrix}^T$$

Thus, $\boldsymbol{l}_1$, $\boldsymbol{l}_2$ can be set according to (76) as

$$\boldsymbol{l}_1=\boldsymbol{P}_1^{-1}\boldsymbol{z}_1=\begin{bmatrix}0.1141\\ 0.2229\\ 0.0014\end{bmatrix}, \quad \boldsymbol{l}_2=\boldsymbol{P}_2^{-1}\boldsymbol{z}_2=\begin{bmatrix}0.0096\\ 0.0007\\ 0.1522\end{bmatrix}$$

It is evident that (72) implies

$$\boldsymbol{h}_1=\boldsymbol{b}_1=\begin{bmatrix}0.000\\ 1.234\\ 0.000\end{bmatrix}, \quad \boldsymbol{h}_2=\boldsymbol{b}_2=\begin{bmatrix}0.000\\ 0.000\\ 1.234\end{bmatrix}$$

The result of several simulations was, that after the hidden neurons number is more than 7, the estimation accuracy is not improved a lot. The chosen parameters for RBF neural network are $\mu_1=\mu_2=0.2$, $\tau_1=10^{-3}$, $\tau_2=5.10^{-4}$. The centers $\boldsymbol{c}_h^{\diamond}$ and widths $\sigma_h$ were determined using the Matlab function newrb, and were left out due to the space limitations. The sliding mode gain was chosen as $\rho_1=\rho_2=1$.

The presented simulation results show, that the method based on neuro-sliding mode observer structure is very well adapted to obtain a faithful estimation of a actuator fault signal, see Fig.2. The residual signal, Fig.3, indicates only the fault existence, but says nothing about the shape and amplitude of

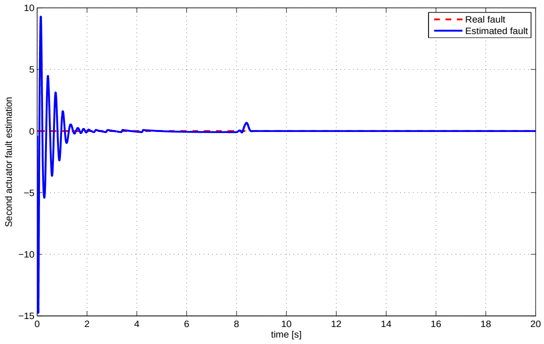


Fig. 4. Second actuator fault and its estimation

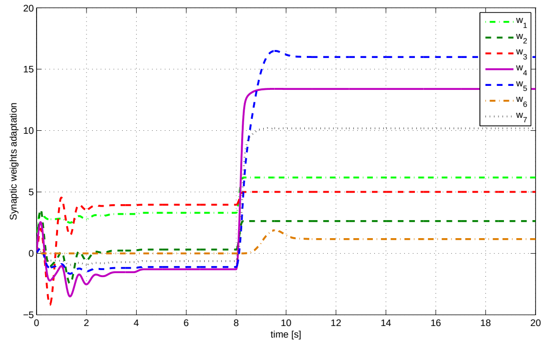


Fig. 6. Synaptic weights adaptation of the first observer

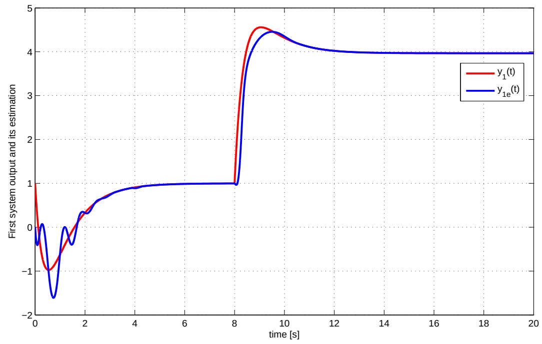


Fig. 5. Response of the first output and its estimation

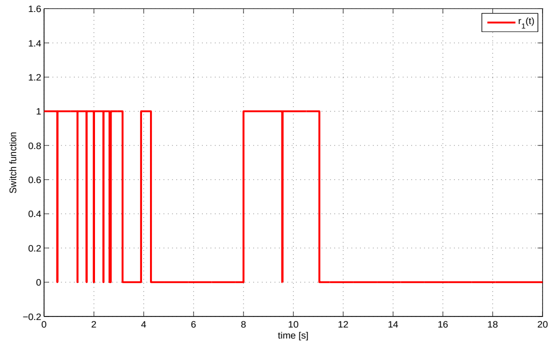


Fig. 7. Switch function behaviour of the first observer

the fault signal, i.e. the output of the RBFNN gives more information about the present fault. Fig.4 shows, that the the second RBFNN output is not coupled by the occurrence of the first actuator fault. The results depicted in Fig.5 and 6 represents the first system output and its estimation, and show the synaptic weights adaptation.

## VIII. Concluding Remarks

In this paper, an approach for actuator fault detection and estimation for a class of linear MIMO systems is investigated. Reformulated design method is presented where the fault estimation is achieved by inclusion of the RBF neural network into the sliding mode observer. The neural networks weights are updated based on a steepest descent rule. Unlike many previous methods in the literature, the proposed fault estimation scheme does not rely on the availability of full state measurements. The stability analysis is given using the Lyapunov method and LMI observer condition to guaranty observer error convergence. Under certain conditions, the estimated signal can approximate the fault signal to any accuracy even in the presence of unmodelled dynamics. Simulation results provided demonstrate and illustrate the effectiveness and capabilities of our proposed actuator fault estimation strategy.

## Acknowledgement

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11. This support is very gratefully acknowledged.

## References

[1] D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz, "User's Guide for SeDuMi Interface 1.04," *LAAS-CNRS, Toulouse*, 2002.
[2] H. Choi and K. Ro, "LMI-based sliding-mode observer design method," in *IEE Proceedings-Control Theory and Applications*, vol. 152, no. 1. IET, 2005, pp. 113–115.
[3] A. Filasová and and D. Krokavec, "On sensor faults estimation using sliding mode observers," in *Conference on Control and Fault-Tolerant Systems (SysTol)*, oct. 2010, pp. 44 –49.
[4] Y. Kim, F. Lewis, and C. Abdallah, "Nonlinear observer design using dynamic recurrent neural networks," in *Proceedings of the 35th IEEE Decision and Control*, vol. 1, dec 1996, pp. 949–954 vol.1.
[5] M. Chen, C. Jiang, and Q. Wu, "Sensor fault diagnosis for a class of time delay uncertain nonlinear systems using neural network," *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 401–405, 2008.
[6] M. Buhmann, *Radial basis functions: theory and implementations*. Cambridge University Press, 2003, vol. 12.
[7] J. Resendiz, W. Yu, and L. Fridman, "Discrete-time sliding mode neural observer for continuous time mechanical systems," in *47th IEEE Conference on Decision and Control*, dec. 2008, pp. 2838 –2843.
[8] R. Fónod and D. Gontkovič, "Fault estimation in a class of first order nonlinear systems," in *IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, jan. 2011, pp. 317 –321.
[9] D. Krokavec and A. Filasova, "Estimation of nonlinear system parameter faults via sliding mode observer," in *Third Pacific-Asia Conference on Circuits, Communications and System (PACCS)*, july 2011, pp. 1–4.
[10] D. Krokavec and A. Filasová, *Discrete–Time Systems (in Slovak)*. Elfa, 2008.
[11] S. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Verlag, 2008.
[12] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1910 –1914, sep 1994.
[13] R. Youssef and H. Peng, "Piecewise sliding mode decoupling fault tolerant control system," *American Journal of Applied Sciences*, vol. 7, no. 1, pp. 102–109, 2010.