Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008



# Practical stability analysis for DNN observation

I. Chairez, A. Poznyak and T. Poznyak

*Abstract*— **The most important fact for differential neural networks dynamics is related to its weights time evolution. This is a consequence for the higher nonlinear structure describing the matrix differential equations, which are associated with the adaptive capability for this kind of neural networks. However, as we know, there is no any analytical demonstration of the weights stability. In fact, this is the main inconvenient to design real applications of differential neural network observers, especially for control uncertain nonlinear systems. This paper deals with the stability proof for the weights dynamics using an adaptive procedure to adjust the weights ODE. A new dynamic neuro-observer, using the classical Luenberger structure based on practical stability theory, is suggested. This methodology aviods the averaged convergence for the state estimation and provides an upper bound for the weights trajectories. A numerical example dealing with the ozonization process state estimation is presented to illustrate the effectiveness of the suggested approach.**

## I. INTRODUCTION

The modelling theory (supported by different physical principles, chemical laws and so on) represents the most extended manner to formalize the knowledge on systems dynamics. However, in many real situations, the modelling rules not always may generate an acceptable reproduction of the reality. In those cases, the nonparametric identification (using adaptive methods) can be successfully applied to cover the deficiencies of classical modelling approaches. In this area, the function approximation technique plays an important role since this method avoids the needs of any mathematical description of the plant.

In this direction, neural networks (NN) have become an attractive tool for modelling complex non-linear systems due to its inherent ability to approximate arbitrary continuous functions. Neural networks provide especially powerful tools for handling large scale problems. It has been proven that artificial neural networks can approximate a wide range of nonlinear functions to any desired degree of accuracy under certain conditions [1]. However, the implementation of neural networks suffers from the lack of efficient constructive approaches, both for choosing network structures and for determining the neuron parameters. It is generally understood that selection of the neural network training algorithm covers an important role for most neural network applications. For example, in the conventional gradient-descent-type weight adaptation, the sensitivity of the unknown system is required to have an on-line training process [2]. Nevertheless, it is difficult to acquire sensitivity information for unknown or highly nonlinear dynamics. Besides, the local minimum of the performance index remains to be the main inconvenient in the implementation of NN [3].

Radial Basis Function (RBF) networks are often used to improve the ANN learning efficiency. T. Poggio and F. Girosi ([4] and [5]) analyze various networks architectures for their approximation abilities and point out that RBF networks possess the property of best approximation. These advantages are further strengthened with the introduction of different activation functions into neural network [6] or dynamic models for the ANN description. Currently, many researches have been done on applications of recurrent neural networks (RNN) [6]. At this point, all the RNN application have been developed for the, so-called, static schemes (like multilayer structure). However, exploiting the fact of being universal approximations, it makes possible straightforwardly substitute uncertain continuous system uncertainties by ANN containing large number of unknown parameters (*weights*) to be adjusted. In general, this class of ANN is known as differential neural networks or DNN for short. They have two important characteristics: their adjustable parameters may appear as linear elements in the ANN description and they may be modified using differential equations [3], [7]. This approach transforms the original problem into a nonlinear robust adaptive feedback one. The DNN approach permits to avoid many problems related to global extremum search converting the learning process to a particular feedback design [7]. If the mathematical model of a considered process is incomplete or partially known, the DNN theory provides an effective instrument to attack a wide spectrum of problems such as non-parametric trajectory identification, state estimation, trajectories tracking and etc. [8]. Mostly real systems are really difficult to be controlled because of the lack of information on its internal structure or-and their current states. Due to continuity of DNN models, more detailed techniques should be applied to resolve important questions on new ANN proposal (convergence for example). The Lyapunov's stability theory (specially the, so-called, controlled Lyapunov theory) has been used within the neural networks for control literature [9], [7]. This is the main tool to justify the DNN improvements on the estimation problems or in the control actions design. Even though there still exists a general trend to enlarge the nonlinear systems for which the aforementioned works can be applied, for instance: results on stability, convergence to arbitrarily small sets and robustness to modeling imperfections and external perturbations of the closed-loop system.

Nevertheless, the advantages, reported for DNN application, have not so nice properties related to the convergence

[1]UPIBI, IPN. M\'{e}xico D.F., E-mail:jchairez@ctrl.cinvestav.mx
[2]CINVESTAV, IPN. M\'{e}xico D.F.,E-mail:apoznyak@ctrl.cinvestav.mx
[3]ESIQIE, IPN. M\'{e}xico, D.F. UPALM Email:tpoznyak@hotmail.com

results. Many papers dealing with DNN application report the estimation error convergence in an average sense. This fact introduces a lot of inconveniences related with DNN-parameter dynamics. Moreover, under the previous techniques, it is not possible to demonstrate the stability of weights trajectories. This paper addresses the application of the practical stability theory over the DNN structures (state estimation algorithm for uncertain systems affected by external perturbations). The new approach is fully explained, including the observer structures, the corresponding convergence schemes and the continuous learning algorithms. The effectiveness of the novel DNN topology is illustrated by the numerical simulation of ozone reaction to eliminate contaminants from water that itself represents the important challenge for the environmental engineers.

## II. DNN OBSERVATION WITH STABLE LEARNING

### A. Class of nonlinear systems

The class of uncertain nonlinear SISO systems considered throughout this paper is governed by a set of $n$ nonlinear ordinary differential equations (ODE) and algebraic state-output mapping given by

$$\dot{x}_t = f(x_t, u_t) + \xi_{1,t}, \quad y_t = Cx_t + \xi_{2,t} \tag{1}$$

where $x_t \in \Re^n$ is the system state at time $t \geq 0$, $y_t \in \Re$ is the system output, $u_t \in \Re$ is the control action, $C \in \Re^{1\times n}$ is an a priory known output matrix. The vectors $\xi_{1,t} \in \Re^n$ and $\xi_{2,t} \in \Re$ represent the state and output deterministic bounded (unmeasurable) disturbances, i.e.,

$$\left\|\xi_{j,t}\right\|^2_{\Lambda_{\xi_j}} \leq \Upsilon_j, \ \Lambda_{\xi_1} \in \Re^{n\times n} \tag{2}$$

with $\Lambda_{\xi_1}^{\intercal} = \Lambda_{\xi_1} > 0, \Lambda_{\xi_2} > 0$. Suppose that

$$\begin{array}{c}\|f(x,u) \text{-} f(w,v)\| \leq L_1\|x\text{-}w\| + L_2|u\text{-}v| \\ w, x \in \Re^n \ u, v \in \Re, \ 0 \leq L_1, L_2 < \infty\end{array} \tag{3}$$

which automatically implies the following property

$$\|f(x,u)\|^2 \leq C_1 + C_2\|x\|^2 + C_3\|u\|^2 \tag{4}$$

$\left(C_k \in \Re^+, k = \overline{1,3}\right)$ valid for any $x$ and $u$. This class of nonlinear system is not very restrictive because an enormous class of non doscontinuous systems are included. Notice that (1) always could be represented as

$$\dot{x}_t = f_0(x_t, u_t \mid \Theta) + \tilde{f}_t + \xi_{1,t} \tag{5}$$

where $f_0(x_t, u_t \mid \Theta)$ is the *nominal dynamics* while $\tilde{f}_t := f(x_t, u_t) - f_0(x, u \mid \Theta)$ is a vector treated as the *modelling error*. Here the parameters $\Theta$ are suggested to be adjusted to minimize the approximation of the nominal part. In particular, according to the DNN approach [7], the nominal dynamics may be defined by the following nonlinear structure

$$\begin{array}{c}f_0(x, u \mid \Theta) = Ax + W_1^0\sigma(x_t) + W_2^0\varphi(x_t)u \\ A \in \Re^{n\times n}, \ W_1^0 \in \Re^{n\times l}, \ W_2^0 \in \Re^{n\times s}, \ \sigma \in \Re^{l\times 1} \\ \varphi \in \Re^{s\times 1} \ \Theta := \left[W_1^0, W_2^0\right] \in \Re^{n \times (n+l+s)}\end{array} \tag{6}$$

The validation of such approximation is based on the approximative Kolmogorov theorem [10], the Stone-Weierstrass theorem [11] on sigmoid approximation and the Lipschitz property (in fact, *quasi-linearity*) (3). Here the matrix $A$ is selected as a stable one and such that the pair $(A, C)$ is observable that supposed to be hold. The vector-functions $\sigma(\cdot) := [\sigma_1(\cdot), \ ..., \ \sigma_l(\cdot)]^{\intercal}$ and $\varphi(\cdot) := [\varphi_1(\cdot), \ ..., \ \varphi_s(\cdot)]^{\intercal}$ are usually constructed with sigmoid function components (following the standard neural networks design algorithms): $x = [x_1, \ ..., \ x_n]^{\intercal}$

$$a\left(1 + b\exp\left(-\sum_{j=1}^{n} c_j x_j\right)\right)^{-1} \tag{7}$$

with $a, b, c_j \in \Re^+$. Nonlinear functions $\sigma(x)$ and $\varphi(x)$ satisfy

$$\begin{array}{c}\|\sigma(x_1) - \sigma(x_2)\|^2 \leq l_\sigma\|x_1 - x_2\|^2 \\ \|\varphi(x_1) \text{-} \varphi(x_2)\|^2 \leq l_\varphi\|x_1 \text{-} x_2\|^2, \ \|\varphi\| \leq \varphi^+\end{array} \tag{8}$$

The admissible control set is supposed to be described by a state estimated feedback controllers defined (in general) by:

$$U^{adm} := \left\{u = u(\hat{x}) : u^2 \leq v_0 + v_1\|\hat{x}\|^2_{\Lambda_u^s}\right\} \tag{9}$$

$(0 < \Lambda_u^s = [\Lambda_u^s]^{\intercal}, \ \Lambda_u^s \in \Re^{n\times n})$ where $\hat{x} \in \Re^n$ is a state estimation defined by any suitable (adaptive and stable) nonlinear observer. By (9) and in view of (4), the following upper bound for the modelling error dynamics $\tilde{f}_t \in \Re^n$ usually takes place:

$$\left\|\tilde{f}\right\|^2_{\Lambda_f} \leq \tilde{f}_0 + \tilde{f}_1\|x\|^2_{\Lambda_{\bar{f}}} + \tilde{f}_2\|\hat{x}\|^2_{\Lambda_{\tilde{f}}} \tag{10}$$

Here $\tilde{f}_0, \tilde{f}_1, \tilde{f}_2 \in \Re^+$, $\Lambda_{\bar{f}}, \Lambda_{\tilde{f}} \in \Re^{n\times n}$, $0 < \Lambda_{\tilde{f}} = \Lambda_{\tilde{f}}^{\intercal}$ $0 < \Lambda_{\bar{f}} = \Lambda_{\bar{f}}^{\intercal}$. Assumed that:

A1. $A$ is a stable matrix.
A2. Any of unknown controlled SISO ODE has solution and it is unique.
A3. The unmeasured disturbances for the uncertain dynamics $\xi_{1,t}$ and the output signal $\xi_{2,t}$ satisfy (2) and they do not violate the existence of the ODE solution (1).
A4. Admissible controls satisfy the sector condition (9), and again, do not violate the existence of the solution to ODE (1).

### B. DNN observer

The DNN observer, which can be used to reproduce the unknown $x_t$ vector, usually is as follows:

$$\begin{array}{c}\frac{d}{dt}\hat{x}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\varphi(\hat{x}_t)u_t + K_1[y_t \text{-} C\hat{x}_t] \\ \forall t \geq 0, \ A \in \Re^{n\times n}, \ K_1 \in \Re^{n\times 1} \\ W_{1,t} \in \Re^{n\times l}, \ W_{2,t} \in \Re^{n\times s}, \ \hat{x}_0 \text{ is fixed}\end{array} \tag{11}$$

where the weight matrices $(W_{j,t}, \ j = 1, 2)$ are *updated* by a *nonlinear learning* law

$$\dot{W}_{j,t} = \Phi_j(W_{j,t}, \hat{x}_t, y_t, u_t, t \mid \Theta) \tag{12}$$

to be designed. Notice this nonlinear adaptive observer reproduces (as it usually called in the state estimation theory) the nominal plant structure (or its approximation) with additional output based correction term proportional to the output error. The correction matrix $K_1$ should be selected as it is described below.

### C. Problem Statement

The main idea is to estimate the states of the uncertain nonlinear system (1) on the presence of external perturbations. This problem can be formulated as follows:

*Under the assumptions A1-A4 for any admissible $u_t$ control strategy* (9) *select the adequate matrices $A$, $K_1$and the update law (12) (including the selection of $W_j^0$, $j=1,2$) in such a way that the upper bound for the estimation error $\beta$ defined as*

$$\beta := \overline{\lim_{t\to\infty}} \|\Delta_t\|^2_{Q_0} \tag{13}$$

where $Q_0^\intercal$=$Q_0>0$, $\Delta_t$:=$\hat{x}_t$-$x_t$ *would be as less as possible.*

### D. Adaptive weights learning law with bounded dynamics

To adjust the weights of the neural observer (11), let us apply the following learning law :

$$\begin{array}{c}\dot{W}_{i,t} = -k_i \left[e_t^\intercal C N_\delta P_1\right]^\intercal \Pi_i^\intercal(\hat{x}_t) - \\ k_i\left[\Pi_i^\intercal \tilde{W}_{i,t}^\intercal P_1 N_\delta \Xi_i N_\delta P_1\right]^\intercal \Pi_i^\intercal + \\ k_i\left[2\Pi_i \hat{x}_t^\intercal P_2\right]^\intercal \Pi_i^\intercal - \dfrac{\alpha \tilde{W}_{i,t}}{2}\end{array} \tag{14}$$

where $\Pi_1=\sigma(\hat{x}_t)$, $\Pi_2=\varphi(\hat{x}_t)u_t$ and

$$\Xi_i = \left(C^\intercal \Lambda_{\xi_2}^{-1} C + \Lambda_i\right),\ \alpha := \min\left(\alpha_{Q_1}, \alpha_{Q_2}\right)$$
$$\alpha_{Q_j} = \lambda_{\min}\left(P_1^{-1/2} Q_{0j} P_1^{-1/2}\right)$$

The matrix $N_\delta \in \Re^{n\times n}$ is defined by $N_\delta := \left(CC^\intercal + \delta I_{n\times n}\right)^{-1}$ with $\delta$ a small positive scalar value (typically 0.01) The matrices $\tilde{W}_{1,t}\in\Re^{n\times l}$ and $\tilde{W}_{2,t}\in\Re^{2n\times s}$ represent the distance between the current values of the adjustable parameters $W_{1,t}$, $W_{2,t}$ to some *fitted* values $W_1^0$ and $W_2^0$, that is $\tilde{W}_{i,t}=W_i^0-W_{i,t}$.

The matrices $\Lambda_i\in\Re^{n\times n}$, $i=1,2$ are symmetric positive definite matrices ($0<\Lambda_i=\Lambda_i^\intercal$). The time varying function $e_t\in\Re$ is the output error defined by $e_t := y_t - \hat{y}_t \in \Re$. The time varying parameters $k_j$, $j=1,2$ are such that $k_j$, $P_j$ ($j=1,2$) are the positive definite solutions for the following algebraic Riccati equations [12]:

$$\begin{array}{c} P_jA_j + A_j^\intercal P_j + P_jR_jP_j + Q_j = 0, \\ A_1 := (A-K_1C),\quad A_2 := A \\ R_1 := W_1^0\Lambda_\sigma^{-1}\left(W_1^0\right)^\intercal + W_2^0\Lambda_\varphi^{-1}\left(W_2^0\right)^\intercal \\ +\Lambda_f^{-1} + \Lambda_{\xi_1}^{-1} + K_1\Lambda_{\xi_2}^{-1}K_1^\intercal \\ Q_1 := \delta^2\Lambda_1^{-1} + \delta^2\Lambda_2^{-1} + 2\tilde{f}_1\Lambda_{\bar{f}} + \\ \lambda_{\max}(\Lambda_\sigma)\, l_\sigma + \Lambda_{K1}^{-1} + Q_{01} \\ R_2 := K_1\left(C\Lambda_{K1}C^\intercal + \Lambda_{\xi_2}^{-1}\right)K_1^\intercal \\ Q_2 := 2\tilde{f}_1\lambda_{\max}\left(\Lambda_{\bar{f}}\right) + \\ \tilde{f}_2\lambda_{\max}\left(\Lambda_{\tilde{f}}\right) + 2\varphi^+ v_1\Lambda_u^s + W_1^0\Lambda_\sigma^{-1}\left(W_1^0\right)^\intercal \\ +W_2^0\Lambda_\varphi^{-1}\left(W_2^0\right)^\intercal + \lambda_{\max}(\Lambda_\sigma)\, l_\sigma + Q_{02} \end{array} \tag{15}$$

### E. Main Result

*Theorem 1:* If there exist positive definite matrices $\Lambda_1$, $\Lambda_2$, $Q_{01,}$ $Q_{02}$, $\Lambda_{K1}$ and positive constants $\delta, v_1$ such that the matrix Riccati equations (15) have positive definite solutions, then the DNN observer (11), supplied by the learning law (14) with any matrix $K_1$ guarantying that the close-loop matrix $A_1 := (A+K_1C)$ is Hurwitz, provides the following upper bound for the state estimation process:

$$\begin{array}{c}\overline{\lim_{t\to\infty}} \|\Delta_t\|^2_{P_1} \le \alpha^{-1}\rho^+ \\ \rho^+ := 4\Upsilon_2 + \Upsilon_1 + \tilde{f}_0 + 2\varphi^+ v_0\end{array} \tag{16}$$

*Proof:* The proofs of this theorem is given in Appendix. ■

*Remark 1:* In the previous publications ([7], [12], [8] for example), there was guaranteed the upper bound ($\rho$) only for **averaged** estimation error, that is, $\overline{\lim_{t\to\infty}} \frac{1}{T}\int\limits_{t=0}^{T} \|\Delta_t\|^2_{P_1}\, dt \le \rho$.

*Remark 2:* If there are no noises ($\Upsilon_1=\Upsilon_2=0$) in the system dynamics and the output measurements and if the class of uncertain systems and the control functions are "*zero*- cone" type, i.e., ($\tilde{f}_0 = v_0 = 0$), then $\rho^+$=0 and the asymptotic error convergence $\Delta_t \to 0$ ($t\to\infty$) is guaranteed.

### F. Training Algorithm

To realize the learning algorithms (14) one needs the knowledge of the nominal matrices $W_s^0$, $s=1,2$ incorporated in $\tilde{W}_{s,t}$, $s=1,2$. The, so-called, *training process* consists in a suitable approximation (or estimation) of these values. It can be realized *off-line* (before the beginning of the state estimation) by the selection of the nominal parameters $\Theta := [A, W_1^0, W_2^0]$ using some *available* experimental data $(u_{t_k}, x_{t_k})\mid_{k=1,N}$ (or fictitious data) and a numerical interpolator algorithm allowing to manage these data as a semi-continuous signals. Obviously, the data must be sampled with a fixed supplied frequency to contain enough information to process a special kind of parametric identification [13] including the "*persistent excitation"* condition and so on. Here we suggests to apply the least-mean square algorithm to attain this aim.

**Mean Least Square (MLS) application.** Rearranging (5) in its integral form is $Y_t$:=$\Phi X_t$+$\zeta_t$ where

$$\zeta_t := \int\limits_{s=t-h}^{t} \left(\tilde{f}_s + \xi_{1,s}\right) ds$$
$$\Phi := \left[\begin{array}{cc} W_1^0 & W_2^0 \end{array}\right]$$
$$X_t := \int\limits_{s=t-h}^{t} \left[\begin{array}{c}\sigma(x_s) \\ \varphi(x_s)\, u_s\end{array}\right] ds$$

The *Matrix Least Square* estimate $\Phi_t^{iden}$ of $\Phi$ is given by $\Phi_t^{iden} := \Psi_t\Gamma_t$ where

$$\Gamma_t^{-1} := \int\limits_{\tau=0}^{t} X_t X_t^\intercal d\tau$$
$$\Psi_t := \int\limits_{\tau=0}^{t} Y_t X_t^\intercal d\tau$$

or in differential form

$$\frac{d}{dt}\Phi_t^{iden} = \left(Y_t - \Phi_t^{iden}X_t\right)X_t^{\intercal}\Gamma_t \\ \frac{d}{dt}\Gamma_t := -\Gamma_t X_t X_t^{\intercal}\Gamma_t, \tag{17}$$

where $t \geq t_0 =: \inf_t \left(\Gamma_t^{-1}>0\right)$ . With this result, it is possible to realize the numerical solution for the DNN observer.

## III. Numerical Results

Ozone has been often used for the treatment of domestic and industrial wastewaters [14]. However, those ozonation applications have been limited by the high cost of ozone production and the low ozone efficiency due to its poor mass transfer rate [15].

First and foremost important problem for effective continuous ozonation system is related to the on-line control methods application, since the classic controllers design methods (such as PID) require total knowledge on the variables involved in ozonation (the actual contaminants concentration value and the ozone mass transfer characteristics). The on-line concentration sensors for few contaminants can be applied in water and wastewater treatments. However, in the case of the complex mixtures in real wastewater treatment, the on-line measurement of all contaminants is unrealistic due two main facts: a high cost for that class of sensors and the impossibility to design a particular sensor for each organic contaminant. Here, the DNN observer seems to be an effective instrument to deal with this particular system (organics ozonation). Indeed, in the case of the residual water treatment by ozone, any mathematical model can not be directly applied because of the complex organics composition and the effect related to the combination of two different reaction mechanisms [12] (depending on the pH value while the reaction occurs): by molecular ozone and by indirect reactions using the free radicals. The ozonation process for two contaminants can be described by:

$$\begin{aligned} \dot{x}_{1,t} &= \frac{W_{gas}}{V_{gas}}\left(u_t\text{-}x_{1,t}\right) - \frac{K_{sat}}{V_{gas}}\left(Q_{max} - x_{2,t}\right) \\ &+ \frac{x_{2,t}}{V_{gas}}\left(k_{c1}x_{3,t} + k_{c2}x_{4,t}\right),\ y_{Oz} = x_{1,t} \\ \dot{x}_{2,t} &= K_{sat}\left(Q_{max}\text{-}x_{2,t}\right)\text{-}\left(k_{c1}x_{3,t}\text{+}k_{c2}x_{4,t}\right)x_{2,t} \\ \dot{x}_{3,t} &= -k_{c1}V_{liq}^{-1}x_{2,t}x_{3,t},\ \dot{x}_{4,t} = -k_{c2}V_{liq}^{-1}x_{2,t}x_{4,t} \end{aligned} \tag{18}$$

where $x_{1,t}$ $(mol/l)$ is the ozone concentration in the output of the reactor (this is ozone which doesn't react with organic compounds dissolved in water), $x_{2,t}$ $(mol)$ is ozone dissolved in a liquid phase, $x_{3,t}$ $(mol/l)$ is the first organic compound concentration at time $t$ while $x_{4,t}$ $(mol/l)$ is the concentration of the second contaminant. The parameters involved in the model description, have the following physical meaning: $V_{gas} = 0.4l$ is the volume of gas phase in the reactor, $W_{gas} = 0.02\ l/s$ is the oxygen gas flow in the inlet of the reactor, $K_{sat} = 0.02\ s^{-1}$ is the ozone saturation constant, $Q_{max} = 4 \cdot 10^{-6}\ mol$ is the maximum of ozone being in the saturated state liquid phase under the given conditions, $k_{c1} = 756\ l/(mol \cdot s)$ is the ozonation rate constant for the first contaminant and $k_{c2} = 1534\ l/(mol \cdot s)$ is the rate constant for the second compound, $V_{liq} = 0.6l$ is the liquid phase volume in the reactor. The state estimate for the first contaminant decomposition is depicted in Fig. (1).Second contaminant decomposition dynamics is depicted in the following figure:

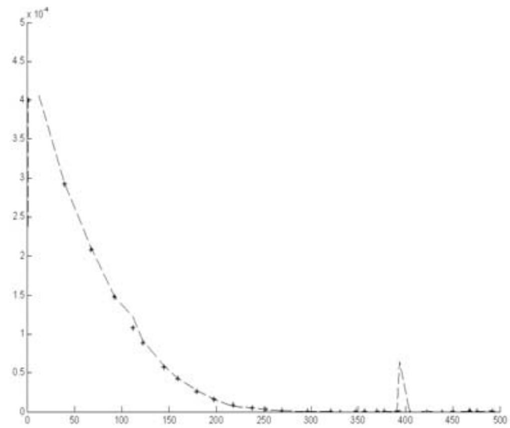

Fig. 1. State estimation for the first contaminant.

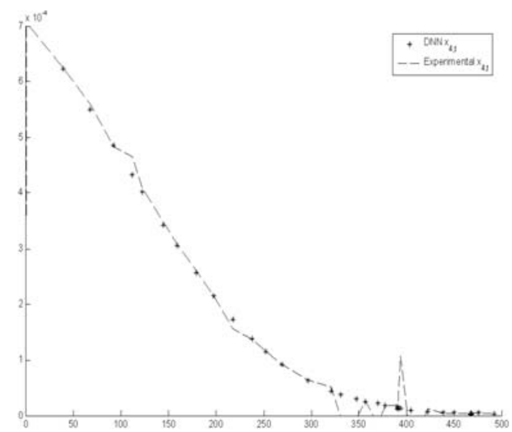

Fig. 2. State estimation for the second contaminant.

## IV. Conclusion

The suggested approach in this work solves one of the most important problems related with the, so-called, differential neural networks: the boundedness property of the dynamic evolution for the weights parameters. The asymptotic convergence for the observing error has been demonstrated applying a Lyapunov analysis. Besides the same analysis leads to the generation of the corresponding conditions for the possible learning rate function. The numerical example of the ozonation methodology to decompose contaminants shows the simulation efficiency for this new kind of the DNN learning procedure.

## REFERENCES


[1] O. Omidvar and D. L. Elliott, *Neural Systems for Control*. New York: Academic Publishers, 1997.

[2] F. J. Lin, W. J. Hwang, and R. J. Wai, "Ultrasonic motor servo drive with on-line trained neural network model-following controller," in *Proc. Inst. Elect. Eng. Electr. Power Applications*, vol. 145, pp. 105–110, 1998.

[3] S. Haykin, *Neural Networks. A comprehensive Foundation*. New York: IEEE Press, 1994.

[4] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proceedings of the IEEE CDC 2006*, vol. 78, pp. 1481–1497, 1990a.

[5] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978–982, 1990b.

[6] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 889–898, November 1992.

[7] A. Poznyak, E. Sanchez, and W. Yu, *Differential Neural Networks for Robust Nonlinear Control (Identification, state Estimation an trajectory Tracking)*. Worl Scientific, 2001.

[8] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 1–11, 1996.

[9] G. Rovithakis and M. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Systs., Man and Cyben.*, vol. 24, pp. 400–412, 1994.

[10] V. Kurkova, "Kolmogorv's theorem and multilayer neural networks," *Neural Networks*, vol. 5, no. 3, pp. 501–506, 1992.

[11] J. B. Prolla, "On the weierstrass-stone theorem," *Journal of approximation theory*, vol. 78, no. 3, pp. 299–313, 1994.

[12] T. Poznyak, I. Chairez, and A. Poznyak, "Application of a neural observer to phenols ozonation in water. simulation and kinetic parameters," *Water Research*, vol. 39, pp. 2611–2620, 2005.

[13] L. Lung, *System Identification, Theory for the user*. Springer-Verlag, 1979.

[14] R. Rice, "Applications of ozone for industrial wastewater treatment- a review," *Ozone Sci. Eng.*, vol. 18, pp. 477–515, 1997.

[15] Y. C. Hsu, Y. F. Chen, and J. H. Chen, "Decolorization of dye rb-19 solution in a continuous ozone process," *J. Environ. Sci. Heal.*, vol. A39, pp. 127–144, 2004.

[16] A. Poznyak, *Sliding Modes: From Principles to Implementation*, ch. Chapter 3: Deterministic Output Noise Effects in Sliding Mode Observation, pp. 123–146. IEE Press, 2004.


## V. APPENDIX

Define the state estimation error as $\Delta_t = x_t - \hat{x}_t$ and the output error as

$$e_t = y_t - \hat{y}_t = Cx_t + \xi_{2,t} - C\hat{x}_t = C\Delta_t + \xi_{2,t}$$

for which the following identities hold:

$$\begin{aligned} C^\intercal e_t &= N_\delta^{-1}\Delta_t + C^\intercal \xi_{2,t} - \delta\Delta_t, \Delta_t \\ &= N_\delta \left(C^\intercal e_t\text{-}C^\intercal \xi_{2,t}\text{+}\delta\Delta_t\right) \end{aligned}$$

The dynamics of $\Delta_t$ is governed by the following ODE:

$$\begin{gathered} \dot{\Delta}_t = \dot{x}_t - \frac{d\hat{x}_t}{dt} = A\Delta_t + \tilde{W}_{1,t}\sigma(\hat{x}_t) + \\ W_1^0\tilde{\sigma}(x,\hat{x}_t) + \tilde{W}_{2,t}\varphi(\hat{x})u_t - \\ W_2^0\tilde{\varphi}(x,\hat{x}_t)u + \tilde{f}_t + \xi_{1,t} - K_1(y_t - \hat{y}_t) \end{gathered} \tag{19}$$

where $\tilde{\sigma}_t := \sigma(x_t) - \sigma(\hat{x}_t)$ and $\tilde{\varphi}(x,\hat{x}_t) = \varphi(x) - \varphi(\hat{x}_t)$. Define the following Lyapunov-like (energetic) function as ($i = 1, 2,\ \kappa > 0$):

$$\begin{gathered} V := V\left(\Delta, \hat{x}, \tilde{W}, t\right) = \|\Delta_t\|_{P_1}^2 + \|\hat{x}_t\|_{P_2}^2 + \\ 2^{-1}\mathrm{tr}\left\{k_1^{-1}\tilde{W}_{1,t}^\intercal\tilde{W}_{1,t} + k_2^{-1}\tilde{W}_{2,t}^\intercal\tilde{W}_{2,t}\right\} \end{gathered} \tag{20}$$

which time derivative is

$$\begin{gathered} \dot{V} = \Delta_t^\intercal P_1\dot{\Delta}_t + 2\hat{x}_t^\intercal P_2\frac{d}{dt}\hat{x}_t + \\ \mathrm{tr}\left\{k_1^{-1}\tilde{W}_{1,t}^\intercal\dot{W}_{1,t} + k_2^{-1}\tilde{W}_{2,t}^\intercal\dot{W}_{2,t}\right\} \end{gathered} \tag{21}$$

Notice that $\Delta_t^\intercal P_1 A\Delta_t = \frac{1}{2}\Delta_t^\intercal\left[P_1A + A^\intercal P_1\right]\Delta_t$ and estimating the rest of the terms in (19) using the matrix inequality

$$XY^\intercal + YX^\intercal \le X\Lambda X^\intercal + Y\Lambda^{-1}Y^\intercal$$

valid for any $X, Y \in R^{r\times s}$ and any $0 < \Lambda = \Lambda^\intercal \in R^{s\times s}$ one gets:

$$\begin{gathered} 2\Delta_t^\intercal P_1\dot{\Delta}_t \le \Delta_t^\intercal\left(2\tilde{f}_1\Lambda_{\tilde{f}}\text{+}\lambda_{\max}(\Lambda_\sigma)\,l_\sigma\right)\Delta_t + \\ \Delta_t^\intercal\left(\delta^2\Lambda_1^{-1}\text{+}\delta^2\Lambda_2^{-1}\text{+}W_1^0\Lambda_\sigma^{-1}\left(W_1^0\right)^\intercal\right)\Delta_t + \\ \Delta_t^\intercal P_1\left(W_2^0\Lambda_\varphi^{-1}\left(W_2^0\right)^\intercal\text{+}\Lambda_{\tilde{f}}^{-1}\text{+}\Lambda_{\xi_1}^{-1}\right)P_1\Delta_t + \\ 2\Upsilon_2\text{+}\tilde{f}_0\text{+}\Upsilon_1\text{+}2\tilde{f}_1\|\hat{x}_t\|_{\Lambda_{\tilde{f}}}^2 + \\ \tilde{f}_2\|\hat{x}\|_{\Lambda_{\tilde{f}}}^2\text{+}e_t^\intercal CN_\delta P_1\tilde{W}_{1,t}\sigma(\hat{x}_t) + \\ \Delta_t^\intercal\left[P_1A\text{+}A^\intercal P_1\right]\Delta_t\text{+}\|\tilde{\varphi}(x,\hat{x}_t)u\|_{\Lambda_\varphi}^2 + \\ \sigma^\intercal(\hat{x}_t)\tilde{W}_{1,t}^\intercal P_1N_\delta\Xi_1N_\delta P_1\tilde{W}_{1,t}\sigma(\hat{x}_t) + \\ e_t^\intercal CN_\delta P_1\tilde{W}_{2,t}\varphi(\hat{x})u_t\text{-}\Delta_t^\intercal P_1K_1(y_t - \hat{y}_t) + \\ u_t^\intercal\varphi^\intercal(\hat{x})\tilde{W}_{2,t}^\intercal P_1N_\delta\Xi_2N_\delta P_1\tilde{W}_{2,t}\varphi(\hat{x})u_t \end{gathered}$$

By the same reason, it follows

$$\begin{gathered} 2\Delta_t^\intercal P_1K_1(y_t\text{-}\hat{y}_t) \le \Delta_t^\intercal\left[P_1K_1C\text{+}\,C^\intercal K_1^\intercal P_1\right]\Delta_t \\ +\Delta_t^\intercal\left[P_1K_1\Lambda_{\xi_2}^{-1}K_1^\intercal P_1\right]\Delta_t + \Upsilon_2 \end{gathered}$$

. Additionally,

$$\begin{gathered} 2\hat{x}_t^\intercal P_2\left(A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t)\right) + \\ 2\hat{x}_t^\intercal P_2\left(W_{2,t}\varphi(\hat{x}_t)u_t + K_1\left[y_t - C\hat{x}_t\right]\right) \\ \le \hat{x}_t^\intercal\left(P_2A + A^TP_2\right)\hat{x}_t - \\ \hat{x}_t^\intercal P_2\tilde{W}_{1,t}\sigma(\hat{x}_t) - 2\hat{x}_t^\intercal P_2\tilde{W}_{2,t}\varphi(\hat{x}_t)u_t \\ +\hat{x}_t^\intercal P_2K_1\left(C\Lambda_{K1}C^\intercal + \Lambda_{\xi_2}^{-1}\right)K_1^\intercal P_2\hat{x}_t + \\ \Delta_t^\intercal\Lambda_{K1}^{-1}\Delta_t + \Upsilon_2 + \|\tilde{\varphi}(x,\hat{x}_t)u\|_{\Lambda_\varphi}^2 + \\ \hat{x}_t^\intercal\left(W_1^0\Lambda_\sigma^{-1}\left(W_1^0\right)^\intercal + W_2^0\Lambda_\varphi^{-1}\left(W_2^0\right)^\intercal\right)\hat{x}_t + \\ \hat{x}_t^\intercal\left(\lambda_{\max}(\Lambda_\sigma)\,l_\sigma\right)\hat{x}_t \end{gathered}$$

Using the upper bound $u_t^\intercal\Lambda_u u_t \le v_0 + v_1\|\hat{x}\|_{\Lambda_u}^2$ (9), after the substitution of all of these inequalities in (21) one gets the following:

$$\begin{gathered}
\dot{V} \leq \Delta_t^{\intercal}\left[P_1\left(A+K_1 C\right)+\left(A+K_1 C\right)^{\intercal} P_1\right] \Delta_t \\
\Delta_t^{\intercal}\left(\delta^2 \Lambda_1^{-1}+\delta^2 \Lambda_2^{-1}+2 \tilde{f}_1 \Lambda_{\bar{f}}\right) \Delta_t+ \\
\Delta_t^{\intercal}\left(\lambda_{\max }\left(\Lambda_\sigma\right) l_\sigma+\Lambda_{K 1}^{-1}+Q_{01}\right) \Delta_t+ \\
\Delta_t^{\intercal} P_1\left(W_1^0 \Lambda_\sigma^{-1}\left(W_1^0\right)^{\intercal}+W_2^0 \Lambda_{\varphi}^{-1}\left(W_2^0\right)^{\intercal}\right) P_1 \Delta_t+ \\
\Delta_t^{\intercal} P_1\left(\Lambda_f^{-1}+\Lambda_{\xi_1}^{-1}+K_1 \Lambda_{\xi_2}^{-1} K_1^{\intercal}\right) P_1 \Delta_t+ \\
\hat{x}_t^{\intercal}\left(P_2 A+A^T P_2\right) \hat{x}_t+\operatorname{tr}\left\{k_1^{-1} \tilde{W}_{1, t}^{\intercal} \dot{W}_{1, t}+k_2^{-1} \tilde{W}_{2, t}^{\intercal} \dot{W}_{2, t}\right\} \\
\hat{x}_t^{\intercal} P_2\left[K_1\left(C \Lambda_{K 1} C^{\intercal}+\Lambda_{\xi_2}^{-1}\right) K_1^{\intercal}\right] P_2 \hat{x}_t+ \\
\hat{x}_t^{\intercal}\left(2 \tilde{f}_1 \lambda_{\max }\left(\Lambda_{\bar{f}}\right)+\tilde{f}_2 \lambda_{\max }\left(\Lambda_{\tilde{f}}\right)+2 \varphi^{+} v_1 \Lambda_u^{\prime}\right) \hat{x}_t \\
+\hat{x}_t^{\intercal}\left(W_1^0 \Lambda_\sigma^{-1}\left(W_1^0\right)^{\intercal}+W_2^0 \Lambda_{\varphi}^{-1}\left(W_2^0\right)^{\intercal}\right) \hat{x}_t+ \\
\hat{x}_t^{\intercal}\left(\lambda_{\max }\left(\Lambda_\sigma\right) l_\sigma+Q_{02}\right) \hat{x}_t+\left[4 \Upsilon_2+\tilde{f}_0+\Upsilon_1+2 \varphi^{+} v_0\right] \\
+\left(e_t^{\intercal} C+\sigma^{\intercal}\left(\hat{x}_t\right) \tilde{W}_{1, t}^{\intercal} P_1 N_\delta \Xi_1\right) N_\delta P_1 \tilde{W}_{1, t} \sigma\left(\hat{x}_t\right)+ \\
\left(e_t^{\intercal} C+u_t^{\intercal} \varphi^{\intercal}(\hat{x}) \tilde{W}_{2, t}^{\intercal} P_1 N_\delta \Xi_2\right) N_\delta P_1 \tilde{W}_{2, t} \varphi(\hat{x}) u_t- \\
2 \hat{x}_t^{\intercal} P_2 \tilde{W}_{1, t} \sigma\left(\hat{x}_t\right)-2 \hat{x}_t^{\intercal} P_2 \tilde{W}_{2, t} \varphi\left(\hat{x}_t\right) u_t+ \\
\alpha\left[2^{-1} \operatorname{tr}\left\{k_1^{-1} \tilde{W}_{1, t}^{\intercal} \tilde{W}_{1, t}+k_2^{-1} \tilde{W}_{2, t}^{\intercal} \tilde{W}_{2, t}\right\} \text{-} V\right]
\end{gathered}$$

where $\varphi^{+}=\lambda_{\max}\left[\tilde{\varphi}^{\intercal}\left(x, \hat{x}_t\right) \Lambda_{\varphi} \tilde{\varphi}\left(x, \hat{x}_t\right)\right]$. Since both Riccati equations (15) admit positive definite solutions, $P_1$ and $P_2$, and under the adaptive weights adjustment laws (14), one gets $\dot{V} \leq -\alpha V+\rho^{+}$. Here the main idea to add (into the learning laws (14)) the terms $\alpha\left(\operatorname{tr}\left\{\frac{\tilde{W}_{i, t}^{\intercal} \tilde{W}_{i, t}}{2 k_i}\right\}\right)$, $i=1,2$ and substracting them that (after simple arrangement) provides the appearance of the term $-\alpha V$ in the right-hand side of the last differential inequality. Then, using the Lemma 1 described in [16] the proof is completed.