Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Differential neural network identifier with composite learning laws for uncertain nonlinear systems

Alejandro Guarneros * **Iván Salgado** * **Manuel Mera** ** **Hafiz Ahmed** ***

* *CIDETC-Instituto Politécnico Nacional, Gustavo A. Madero, Mexico City (e-mail: isalgador@ipn.mx).*
** *ESIME Ticomán-Instituto Politécnico Nacional, Gustavo A. Madero, Mexico City, Mexico (e-mail: mlmerah@gmail.com)*
*** *Coventry University, Coventry CV1 5FB, United Kindgdom (e-mail: hafiz.h.ahmed@ieee.org)*

Abstract: This manuscript describes the design and numerical implementation of a novel composite differential neural network aimed to estimate nonlinear uncertain systems. A differential neural network (DNN) with a composite feedback matrix approximates the structure of non-linear uncertain systems. The feedback matrix is assumed to belong to a convex set as well as the free parameters of the DNN (weights) at any instant of time. Therefore, l-different DNN works in parallel. A composite Lyapunov function finds the convex hull approximation of the set of DNN working together to improve the approximation capabilities of classical neural networks. The main result of this study shows the practical stability of the estimation error. Numerical simulations demonstrate the approximation capabilities of the composite DNN implemented in a Van Der Pol oscillator where the presence of high-frequency components makes difficult a classical DNN approximation.

Keywords: Composite Lyapunov function, Differential neural networks, nonlinear systems, uncertain systems

1. INTRODUCTION

The artificial neural networks (ANNs) are complex parallel structures that emulate how the human brain processes a large number of data (Poznyak et al., 2019). The ANNs have been successfully applied to solve the problem of non-parametric identification. In the case of dynamic systems oriented to identification, estimation, and control, differential neural networks (DNNs) offer attractive features to deal with uncertain and perturbed systems. Among other characteristics, DNNs provide robustness and practical stability through the second Lyapunov's stability method (Poznyak et al., 2001). Indeed, the learning laws are a consequence of this analysis. This method guarantees the stability of the equilibrium point for the identification or estimation error as well as the boundedness of the DNN's weights. The outstanding results have brought about new structures with original learning processes. Some important applications of DNNs are found in delayed systems (Xu et al., 2019) and sliding mode based learning laws (Keighobadi et al., 2019), among others.
However, for systems with high-frequency components, the DNN have problems to reproduce accurately the system dynamics. The elements of the basis used in classical theory limit the identification properties of the DNNs (Poznyak et al., 2001). One possible solution is the implementation of several DNNs working in a parallel distribution (Hunter and Wilamowski, 2011). The index to select which DNN is in the "ON" state can be obtained by training an additional neural network or a fuzzy logic approach (Cervantes et al., 2017). In this last methodology, the defuzzification stage in the fuzzy logic algorithm yields an appropriate selection of the DNN to apply.
The concept of composite functions can expand the approximation properties of a DNN. Even when there already exists in literature the concept of composite neural network (Meng and Karniadakis, 2019), in this manuscript, the objective is to propose a set of learning laws derived from a composite Lyapunov function (CLF) while the composite network in (Meng and Karniadakis, 2019) implies a cascade structure to signal processing.
The CLF has the objective to make less conservative the estimations of invariant sets compared to ellipsoidal approximations (Hu and Lin, 2003). The CLFs have been applied for stability analysis, to estimate the regions of attraction for input saturated systems and to study the stability of piecewise linear and switched systems (Hu et al., 2008; Azhmyakov et al., 2019).
A classical Lyapunov function is commonly defined by a quadratic structure. A CLF is composed of the convex hull of a group of individual Lyapunov equations generally selected as ellipsoids. These are, in general, larger sets than those corresponding to each Lyapunov quadratic function. One of the main advantages of working with CLFs applied

⋆ Authors want to thank the financial support by SIP-Instituto Politécnico Nacional grant 20195253

to the DNN theory, is the facility to select the gains through the solution of a set of linear matrix inequalities. These gains ensure stability on larger sets than those obtained by conventional Lyapunov methods (Tingshu Hu and Zongli Lin, 2003).

The main idea behind this manuscript is the development of a so-called Composite Differential Neural Network (CDNN). The objective of this new structure is to obtain a better approximation despite the presence of high-frequency components. A set of individual networks run in parallel. Each DNN converge to a specific ellipsoidal set, the optimization of the convex hull employing the concept derived in (Hu and Lin, 2003) will produce a set of composite structure with composite learning laws for the CDNN. The classical learning laws derived in (Poznyak et al., 2001) will be improved with an optimization procedure over the CLF definition. Therefore the following contributions are

- The improvement of the identification theory based on DNN using enhanced composite learning laws derived from a novel controlled composite Lyapunov Function (CCLF).
- A numerical algorithm to implement the LMIs obtained from the stability analysis.
- The numerical implementation of this new structure in a nonlinear academic example.

2. NEURAL NETWORK IDENTIFIER WITH COMPOSITE LEARNING LAWS

2.1 Problem statement

Consider the following nonlinear system

$$\dot{x}(t) = f(x,t) + g(x)u(t) \tag{1}$$

where, $x \in \mathbb{X} \in \mathbb{R}^n$ is the state vector, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a feedback Lipschitz function that describes the system dynamics; $g : \mathbb{R}^n \to \mathbb{R}^m$ is the matrix function associated to the external input, which guarantees the controllability of system (1); u is the control input. This manuscripts considers the fulfilment of the following assumptions:

Assumption 1. There exists a control signal u which guarantee the close-loop stability of (1). Therefore,

$$\|x\| \le x^+, \quad \forall t \ge 0, \quad x^+ \in \mathbb{R}_+ \tag{2}$$

Assumption 2. The control action u belongs to the following admissible set

$$\begin{gathered} U_{adm} := \{u | \|u\| \le u_1\|x\| + u_0\}, \\ u_1, u_0 \in \mathbb{R}_+ \end{gathered} \tag{3}$$

Notice that assumption 2 includes many control designs, such as, sliding modes and classical feedback controllers. The following section describes the possible representation of system (1) as an adaptive structure based on CDNN.

2.2 Neural Network representation

Let us consider the representation of system (1) as

$$\dot{x}(t) = A^* x(t) + W_1^* \sigma(x) + W_2^* \phi(x)u(t) + \tilde{f}(x) \tag{4}$$

Where $A^* \in \mathbb{R}^{n\times n}$ is a unknown Hurwitz matrix. Let assume that matrix A^* can be represented by convex combination of l matrices, that is,

$$A^* \in \text{co}\{A_1, A_2, \ldots, A_l\} \tag{5}$$

W_i^* with $i = \overline{1,2}$ are the best fitted weights to adjust the adaptive structure in (4) to the nonlinear system in (1). They are unknown but bounded as

$$\begin{gathered} (W_i^*)^\top \Lambda_i W_i^* \le \bar{W}_i, \\ \bar{W}_i^\top = \bar{W}_i > 0, \quad \Lambda_i^\top = \Lambda_i > 0 \end{gathered} \tag{6}$$

$\sigma : \mathbb{R}^n \to \mathbb{R}^n$ and $\phi : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the activation functions. In this manuscript, sigmoid type functions constitute the basis σ and ϕ. Indeed, this selections satisfies the following assumption:

Assumption 4. The activation functions satisfies the sector condition given by

$$\begin{gathered} \|\sigma(y_2) - \sigma(y_1)\| \le L_\sigma \|y_2 - y_1\|, \\ \|\phi(y_2) - \phi(y_1)\|_M \le L_\sigma \|y_2 - y_1\| \end{gathered} \tag{7}$$

Where $\|\cdot\|$ and $\|\cdot\|_M$ represent any vector norm and any matrix norm respectively (Poznyak, 2008), $y_1, y_2 \in \mathbb{X} \subset \mathbb{R}^n$. Additionally, the selection of the activation functions ensure the fulfilment of the following inequalities

$$\begin{gathered} \|\sigma(y_1)\| \le \sigma_+, \quad \sigma^+ \in \mathbb{R}^+, \\ \|\phi(y_1)\|_M \le \phi_+, \quad \phi^+ \in \mathbb{R}^+, \end{gathered} \tag{8}$$

Assumption 5. The term $\tilde{f}$ describes the modelling error. This study assumes that $\tilde{f}$ is bounded by

$$\|\tilde{f}\| \le f^+, \quad f^+ \in \mathbb{R}_+ \tag{9}$$

Besides, the quadratic bound in (6) allow us to define o less restrictive representation, similar to matrix A^*, let represent W_i^* as a convex composition of a set of *ideal* weights

$$W_i^* \in \text{co}\left\{W_{i,1}^*, W_{i,2}^*, \ldots, W_{i,l}^*\right\}, \ i = \overline{1,2}. \tag{10}$$

2.3 Neural network identifier

Based on the representation given in (4), let us propose the following DNN identifier

$$\dot{\hat{x}}(t) = A(\lambda)\hat{x} + W_1(\lambda,t)\sigma(\hat{x}) + W_2(\lambda,t)\phi(\hat{x})u(t) \tag{11}$$

with

$$\begin{gathered} A(\lambda) = \sum_{j=1}^{l} \lambda_j A_j, \\ W_i(\lambda,t) = \sum_{j=1}^{l} \lambda_j W_i(t), \ \text{for } i = \overline{1,2}. \end{gathered} \tag{12}$$

where the scalars λ_j for $j = \overline{1,l}$ are the elements of a vector $\lambda \in \Gamma$. And

$$\Gamma := \left\{ \lambda \in \mathbb{R}^l : \sum_{j=1}^{l} \lambda_j = 1, \lambda_j \ge 0 \right\}, \tag{13}$$

The selection of the individual elements in the set Γ is obtained by the second method of Lyapunov. This method is described below in section 3.

2.4 Composite Learning laws

The free parameters of the CDNN are updated by the following learning laws

$$\begin{gathered} \dot{W}_i = -k_i P(\lambda) \Delta \Upsilon_i^\top, \\ \Upsilon_1 = \sigma(\hat{x}), \qquad \Upsilon_2 = \phi(\hat{x})u(t) \end{gathered} \tag{14}$$

Where $\Delta = \hat{x} - x$ defined as the identification error, $k_i \in \mathbb{R}_+$ are the learning coefficients of the CDNN, R_j^{-1} are the solutions of the following Riccati equations

$$\begin{gathered} A_j R_j^{-1} + R_j^{-1} A_j^\top + R_j^{-1} S R_j^{-1} + Q \leq 0 \\ S := \bar{W}_1 + \bar{W}_2, \\ Q := \left(L_\sigma \Lambda_1^{-1} + L_\phi u^+ \Lambda_2^{-1} \right) \\ 0 < \Lambda_1 = \Lambda_1^\top, \quad 0 < \Lambda_2 = \Lambda_2^\top. \end{gathered} \tag{15}$$

Additionally, defining the positive definite matrix

$$P(\lambda) := \left(\sum_{j=1}^{l} \lambda_j R_j^{-1} \right)^{-1}. \tag{16}$$

The learning laws are obtained by the Lyapunov's second method. The analysis is given in the following section.

3. MAIN RESULT

The development of these laws require in the analysis the concept of a composite Lyapunov function, which is given in the next definition.

Definition 1. (Composite Lyapunov function). Consider the elements in the set Γ described in (13) and the set of positive matrices R_j used to construct the composite set defined in (16). The V_c defined as

$$\begin{gathered} V_c(\Delta, \tilde{W}_i) = \min_{\lambda \in \Gamma} \Delta^\top P(\lambda) \Delta + \\ \min_{\lambda \in \Gamma} \sum_{i=1}^{2} k_j \operatorname{tr}\left\{ \tilde{W}_i^\top(\lambda) \tilde{W}_i(\lambda) \right\} \end{gathered} \tag{17}$$

With $\tilde{W}_i(\lambda) := W_i(\lambda) - W_i^*$. Clearly, V_c is positive. Therefore, if $\dot{V}_c$ is negative semi-definite, along the dynamics of Δ in any subset of the state space containing the origin, it is said that V_c is a composite quadratic Lyapunov function.

The next theorem summarizes the main result of this study

Theorem 1. Consider the approximation of system (1) by the time varying DNN in equation (4), the fulfilment of assumptions (1-5) and the DNN identifier in (11) updated with the learning laws (14). If the following LMI is feasible for a positive scalar α and for the symmetric positive definite matrix $P(\lambda^*)$,

$$\Omega_0 \leq 0, \tag{18}$$

with,

$$\Omega_0 = \begin{bmatrix} P^{-1}(\lambda^*) A(\lambda^*) + A^\top(\lambda^*) P^{-1}(\lambda^*) + S & I & I \\ I & -\alpha I & 0 \\ I & 0 & -Q^{-1} \end{bmatrix}$$

and

$$\lambda^* = \arg\min_{\lambda \in \Gamma} \max_i \|A_i x - A(\lambda) x\|. \tag{19}$$

Therefore, the equilibrium point of the estimation error Δ is ultimately bounded in neighbourhood around zero.

Proof. The composite Lyapunov function is

$$V_c := \Delta^\top P(\lambda) \Delta + \sum_{i=1}^{2} k_i \operatorname{tr}\left\{ \tilde{W}_i^\top(\lambda, t) \tilde{W}_i(\lambda, t) \right\} \tag{20}$$

The time derivative of (20) is

$$\dot{V}_c := 2\Delta^\top P(\lambda) \dot{\Delta} + 2 \sum_{i=1}^{2} k_i \operatorname{tr}\left\{ \tilde{W}_i^\top(t) \dot{W}_i \right\} \tag{21}$$

The dynamics of the identification error are the following

$$\begin{gathered} \dot{\Delta} = A(\lambda)\hat{x} - A^* x + \tilde{W}_1 \sigma(\hat{x}) + \tilde{W}_2 \phi(\hat{x}) u(t) + \\ W_1^* \tilde{\sigma} + W_1^* \tilde{\phi} u(t) - \tilde{f}(x) \end{gathered} \tag{22}$$

where

$$\tilde{\sigma} = \sigma(\hat{x}) - \sigma(x), \qquad \tilde{\phi} = \phi(\hat{x}) - \phi(x),$$
$$\tilde{W}_i := W_i(\lambda, t) - W_i^*, \quad i = \overline{1,2}$$

The substitution of (22) in the first term of (21) yields in

$$\begin{gathered} \Delta^\top P(\lambda) \dot{\Delta}(t) = \Delta^\top P(\lambda) A(\lambda) \hat{x} - A^* x - \\ \Delta^\top P(\lambda) \tilde{f}_i(x(t), t) + \\ \Delta^\top P(\lambda) \tilde{W}_1 \sigma(\hat{x}) + \Delta^\top P(\lambda) \tilde{W}_2 \phi(\hat{x}) u(t) + \\ \Delta^\top P(\lambda) W_1^* \tilde{\sigma} + \Delta^\top P(\lambda) W_1^* \tilde{\phi} u(t) \end{gathered}$$

Adding and subtracting $A(\lambda)x$ and rearranging the terms

$$\begin{gathered} \Delta^\top P(\lambda) \dot{\Delta}(t) = \Delta^\top P(\lambda) A(\lambda) \Delta - \\ \Delta^\top P(\lambda) (A^* - A(\lambda)) x + \Delta^\top P(\lambda) \tilde{f}_i(x(t), t) + \\ \Delta^\top P(\lambda) \tilde{W}_1 \sigma(\hat{x}) + \Delta^\top P(\lambda) \tilde{W}_2 \phi(\hat{x}) u(t) + \\ \Delta^\top P(\lambda) W_1^* \tilde{\sigma} + \Delta^\top P(\lambda) W_1^* \tilde{\phi} u(t) \end{gathered}$$

Applying the inequality $2X^\top Y \leq X^\top \Lambda X + Y^\top \Lambda^{-1} Y$ the following upper bounds are valid (Poznyak, 2008)

$$-2\Delta^\top P(\lambda) \tilde{f} \leq \Delta^\top P(\lambda) \Lambda_1 P(\lambda) \Delta(t) + \tilde{f}^\top \Lambda_1^{-1} \tilde{f} \tag{23}$$

For the terms containing W_1^* and W_2^* one has

$$\begin{gathered} \Delta^\top P(\lambda) W_1^* \tilde{\sigma} \leq \Delta^\top P(\lambda) \bar{W}_1 P(\lambda) \Delta + \\ L_\sigma \Delta^\top \Lambda_2^{-1} \Delta(t) \\ \Delta^\top P(\lambda) W_2^* \tilde{\phi} u(t) \leq \Delta^\top P(\lambda) \bar{W}_2 P(\lambda) \Delta + \\ L_\phi u^+ \Delta^\top \Lambda_3^{-1} \Delta(t) \end{gathered} \tag{24}$$

where $\bar{W}_1 := (W_1^*)^\top \Lambda_2 W_1^*$ and $\bar{W}_2 := (W_2^*)^\top \Lambda_2 W_2^*$. Taking into account that

$$\Delta^\top P(\lambda) \tilde{W}_1 \sigma(\hat{x}) = \operatorname{tr}\left\{ \tilde{W}_1^\top P(\lambda) \Delta \sigma^\top(\hat{x}) \right\}$$

$$\Delta^\top P(\lambda) \tilde{W}_2 \phi(\hat{x}) u(t) = \operatorname{tr}\left\{ \tilde{W}_2^\top P(\lambda) \Delta \left(\phi(\hat{x}) u(t) \right)^\top \right\} \tag{25}$$

Joining the results in (23)-(25) into the derivative of V_c in (21), the next result is obtained

$$\begin{gathered} \dot{V}_c \leq \Delta^\top \left(P(\lambda) A(\lambda) + A(\lambda)^\top P(\lambda) + \right. \\ \left. P(\lambda) S P(\lambda) + Q \right) \Delta - \\ 2\Delta^\top P(\lambda) \left(A^* - A(\lambda) \right) x + \\ 2\operatorname{tr}\left\{ \tilde{W}_1 \Theta_1 \right\} + 2\operatorname{tr}\left\{ \tilde{W}_2 \Theta_2 \right\} - \Delta^\top Q \Delta + \varsigma, \end{gathered} \tag{26}$$

where

$$\begin{gathered} \Theta_1 = k_1 \dot{W}_1 + P(\lambda) \Delta \sigma^\top(\hat{x}), \\ \Theta_2 = k_2 \dot{W}_2 + P(\lambda) \Delta \left(\phi(\hat{x}) u(t) \right)^\top. \end{gathered}$$

Although unknown, the term $\delta_x(\lambda) := (A^* - A(\lambda))x$ is always bounded for any $\lambda \in \Gamma$, by convexity and (5), as

$$\|A^* x - A(\lambda) x\| = \|\delta_x(\lambda)\| \leq \bar{\delta}_x(\lambda), \ \forall x \in \mathbb{R}^n,$$

for a positive scalar $\bar{\delta}_x \in \mathbb{R}$. This $\bar{\delta}_x(\lambda)$ can be obtained as follows.

Defining $z^* = A^* x$ and $z(\lambda) = A(\lambda) x$, and noting that $z^* \in \mathcal{C}$, $\mathcal{C} = co\{z_1, \ldots, z_l\}$, where $z_i = A_i x$ for $i = \overline{1, l}$. Then for a given $\lambda \in \Gamma$

$$\bar{\delta}_x(\lambda) = \max_i \|z_i - z(\lambda)\|.$$

Algorithm 1. Algorithm for numerical implementation.
Step 0. Select a suitable sampling period ($\tau > 0$) to implement an explicit Euler method. The sampling times $t_k = \tau k$, $k = 0, 1, \ldots$ are the iteration steps.
Step 1. Declare the variables of equation (18) according the number of DNNs to evaluate.
Step 2. Select the activation functions, the matrix A and other elements needed for each DNN
Step 3. Compose the set of matrices described in the main theorem and solved them to find the composite matrix $P(\lambda)$. An alternative option is to solved the Ricatti equation in (15) for each different DNN.
Step 4. Solve (19) to obtain the corresponding elements λ_j
Step 5. Feed the CDNN with the resulting learning laws.
Step 6. Let $k = k + 1$ and **Go To Step 4**.

So it is possible to select an optimal λ^* such that $\bar{\delta}_x(\lambda^*)$ is minimal, this is (19).

Taking this into consideration, and adding an subtracting $\alpha\frac{\|\delta_x(\lambda)\|^2}{\bar{\delta}_x^2(\lambda^*)}$, where α is a positive scalar, to (26)

$$\begin{aligned}\dot{V}_c \leq \Delta^\top \big(&P(\lambda)A(\lambda) + A(\lambda)^\top P(\lambda) + \\ &P(\lambda)SP(\lambda) + Q\big)\,\Delta - \\ 2\Delta^\top P(\lambda)\delta_x(\lambda) &\pm \alpha\frac{\|\delta_x(\lambda)\|^2}{\bar{\delta}_x^2(\lambda^*)} + \\ 2\mathrm{tr}\left\{\tilde{W}_1\Theta_1\right\} &+ 2\mathrm{tr}\left\{\tilde{W}_2\Theta_2\right\} - \Delta^\top Q\Delta + \varsigma.\end{aligned} \tag{27}$$

Defining the extended vector $\eta^\top := \left[\Delta^\top,\ \delta_x^\top(\lambda)\right]$, (27) can be rewritten as

$$\begin{aligned}\dot{V}_c \leq \eta^\top\Omega\eta + 2\mathrm{tr}\left\{\tilde{W}_1\Theta_1\right\} + \\ 2\mathrm{tr}\left\{\tilde{W}_2\Theta_2\right\} - \Delta^\top Q\Delta + \bar{\varsigma}(\lambda),\end{aligned}$$

with

$$\Omega = \begin{bmatrix} P(\lambda)A(\lambda) + A^\top(\lambda)P(\lambda) + P(\lambda)SP(\lambda) + Q & P(\lambda) \\ P(\lambda) & -\frac{\alpha}{\bar{\delta}_x^2}I \end{bmatrix}$$

and $\bar{\varsigma}(\lambda) = \varsigma + \alpha\frac{\|\delta_x(\lambda)\|^2}{\bar{\delta}_x^2(\lambda^*)}$.

Now, taking Θ_1 and Θ_1 equal to zero yields in the learning laws in (14), selecting $\lambda = \lambda^*$, and if $\Omega \leq 0$, or equivalently (18) is true, the time derivative of V_c is upper-bounded as

$$\dot{V}_c \leq -\Delta^\top Q\Delta + \varsigma(\lambda^*). \tag{28}$$

By the Barbalat's Lemma the identification error is practically stable in a neighbourhood of the equilibrium point in zero.

□

4. NUMERICAL RESULTS

4.1 Implementation procedure

To implement the composite approximation, the Algorithm 1 describes an iterative numerical solution to implement the optimization described in (19). The implementation of the CDNN seems to be complex for the iterative needed solution. However, notice that the set of LMI's described in (18) has to be solved once. The iterative requirement applies only in equation (19).

4.2 Numerical simulations

The model of the Van der Pol oscillator is given by the following couple of differential equations (Ahmed et al., 2018)

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + \varepsilon x_2(1 - x_1^2) + w(t) + u\end{aligned} \tag{29}$$

where $\varepsilon > 0$ is an specific parameter for the model, $x = \left[x_1\ x_2\right]^\top \in \mathbb{R}^2$ is the state vector, $w \in \mathbb{R}$ is an unknown but bounded and Lipschitz external disturbance, that is, $\|\dot{w}(t)\| \leq w^+$, for all $t \geq 0$ and with w^+ a known positive constant. For simulation proposes,$\varepsilon = 2$. For the model, the initial conditions were zero. A CDNN was designed with two different elements, that is, $l = 2$ in equation (12). Indeed, to compare the composition obtained, the two DNN were simulated. The first DNN had the following elements in its inner structure

$$A_1 = \begin{bmatrix} -50 & 0 \\ 0 & -50 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix},$$

$$Q_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

The parameter in the set (18) $\alpha_1 = 0.2$. This parameters yield in the following matrix R_1^{-1} as

$$R_1^{-1} = \begin{bmatrix} 0.1385 & 0.0615 \\ 0.0615 & 0.1385 \end{bmatrix}$$

The activation functions for the first network were selected as

$$\sigma_1(\hat{x}) = \begin{bmatrix} \dfrac{1}{1 + e^{(-0.10.1]^\top \hat{x}_i)}} \\ \dfrac{1}{1 + e^{([0.1-0.1]^\top \hat{x}_i)}} \end{bmatrix}$$

$$\phi_1(\hat{x}) = \frac{1}{1 + e^{(1-1]^\top \hat{x}_i)}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

The learning coefficients were $k_{11} = k_{12} = 25I_{2\times 2}$.
The second DNN had the following elements in its inner structure

$$A_2 = \begin{bmatrix} -3.5 & 0 \\ 0 & -5.1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.7 \end{bmatrix}$$

The parameter in the set (18) $\alpha_1 = 0.5$. This parameters yield in the following matrix R_2^{-1} as

$$R_2^{-1} = \begin{bmatrix} 0.2130 & 0.0115 \\ 0.0115 & 0.2500 \end{bmatrix}$$

The activation functions were

$$\sigma_2(\hat{x}_2) = \begin{bmatrix} \dfrac{1}{1 + e^{(-15-18]^\top \hat{x}_2)}} \\ \dfrac{1}{1 + e^{([1518]^\top \hat{x}_2)}} \end{bmatrix}$$

$$\phi_2(\hat{x}_2) = \frac{1}{1 + e^{(1-1]^\top \hat{x}_2)}} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

The learning coefficients ware

$$k_{21} = \begin{bmatrix} 2 & 0 \\ 0 & 2.4 \end{bmatrix} \qquad k_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The simulation was performed with an explicit Euler method with a sampling period of 0.001. Figure 1 shows the state identification with the two different DNNs. A continuous red line represents the real trajectories for the Van der Pol oscillator. The black dotted line correspond to the first DNN and the green one corresponds to the second DNN. Both of them have an acceptable identification performance. This issue has been addressed in many previous works (Poznyak et al., 2001, 2019). An additional improvement can be obtained with the increment of the activation functions and the size of the available parameters. However, there does not exist an explicit method to select the parameters (Lewis et al., 1998; Igelnik and Yoh-Han Pao, 1995) and this procedure sometimes make difficult to adjust approaches based on DNN. The CDNN presented in figure 1 is adjusted by means of an LMI optimization procedure. The approximation of the CDNN is depicted with a dashed blue line. Even when the three approximations remain in a small convergence region, the CDNN takes advantage when the derivative of the signal increased its value. In the closer view one can notice that the CDNN approximates better the nonlinear system. From the second four to the second five is possible to see how the first DNN is below the real system, and the second DNN upper estimate the nonlinear trajectories. The CDNN makes a weighing between them and makes a more exact identification.

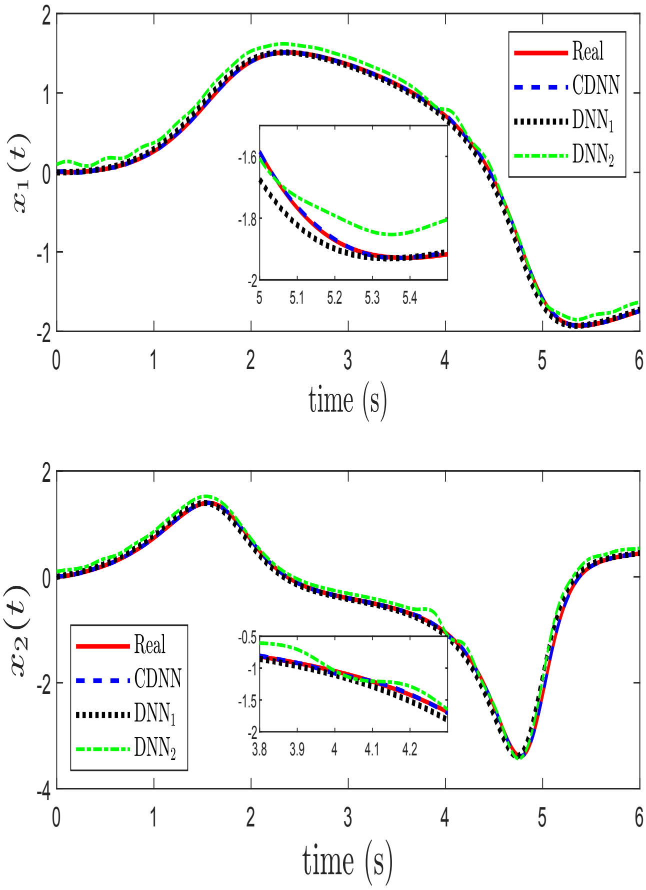

Fig. 1. State identification based on differential neural networks: comparison of the composite version against individual structures

Figure 2 shows the performance index of each network defined as the Euclidean norm of the identification error. Both single structures have a similar behaviour. However, the euclidean norm for the CDNN remains in a smaller neighbourhood almost all the time. Figure 2 describes with a red continuous line the Euclidean norm of the CDNN approximation.

The weights for each sampled time are shown in figure

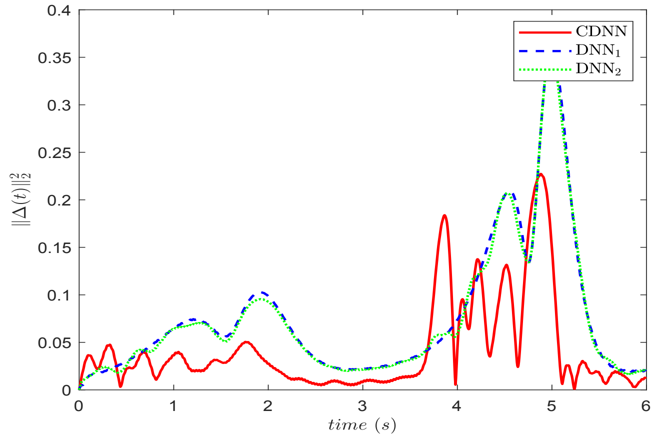

Fig. 2. Euclidean norm of the identification error for the two single structures and the CDNN. The CDNN Euclidean norm is smaller than the other two cases

3. The weights present some variations between 0 to 0.5 for the first CDNN. The second single DNN structure contributes with values from 0.5 to one. The variations are more significant when the derivative of the signal increased. As a consequence, as it was mentioned before, the CDNN obtains a better approximation.

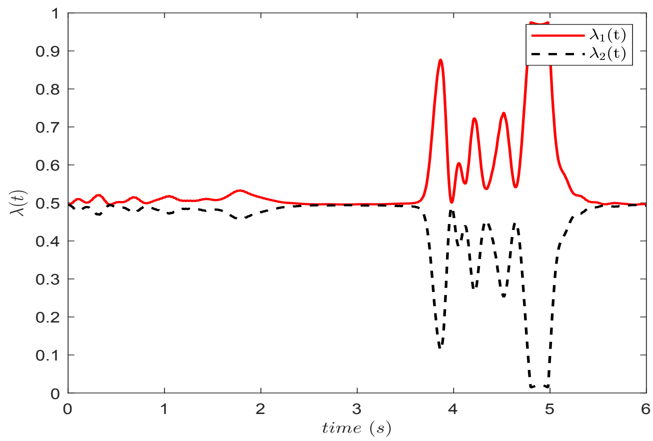

Fig. 3. Weights obtained through the optimization procedure in equation (16)

5. CONCLUSIONS

A CDNN was presented to improve the approximation characteristics of classical DNN. A new composite controlled Lyapunov function was presented to develop new learning laws for the CDNN. Even when this structure needs an on-line computation of the weights included in an optimization process, the complete procedure can be alleviated with only periodic calculations. Future research should be oriented to include noises in the available measurements.

REFERENCES

Ahmed, H., Ríos, H., Ayalew, B., and Wang, Y. (2018). Second-order sliding-mode differentiators: an experimental comparative analysis using van der pol oscillator. *International Journal of Control*, 91(9), 2100–2112. doi: 10.1080/00207179.2018.1442023.

Azhmyakov, V., Mera, M., and Juárez, R. (2019). Advances in attractive ellipsoid method for robust control design. *International Journal of Robust and Nonlinear Control*, 29(5), 1418–1436. doi:10.1002/rnc.4446.

Cervantes, J., Yu, W., Salazar, S., and Chairez, I. (2017). Takagi–sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 25(6), 1601–1615. doi: 10.1109/TFUZZ.2016.2612697.

Hu, T. and Lin, Z. (2003). Composite quadratic lyapunov functions for constrained control systems. *IEEE Transactions On Automatic Control*, 3, 3500–3505. doi: 10.1109/CDC.2002.1184417.

Hu, T., Ma, L., and Lin, Z. (2008). Stabilization of switched systems via composite quadratic functions. *IEEE Transactions on Automatic Control*, 53(11), 2571–2585.

Hunter, D. and Wilamowski, B. (2011). Parallel multilayer neural network architecture with improved efficiency. In *2011 4th International Conference on Human System Interactions, HSI 2011*, 299–304. doi: 10.1109/HSI.2011.5937382.

Igelnik, B. and Yoh-Han Pao (1995). Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6), 1320–1329. doi:10.1109/72.471375.

Keighobadi, J., Hosseini-Pishrobat, M., and Faraji, J. (2019). Adaptive neural dynamic surface control of mechanical systems using integral terminal sliding mode. *Neurocomputing*. doi:10.1016/j.neucom.2019.10.046.

Lewis, F.L., Yesildirak, A., and Jagannathan, S. (1998). Neural network control of robot manipulators and nonlinear systems.

Meng, X. and Karniadakis, G.E. (2019). A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 109020. doi:10.1016/j.jcp.2019.109020.

Poznyak, A. (2008). *Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques*. Elsevier, Amsterdam.

Poznyak, A., Chairez, I., and Poznyak, T. (2019). A survey on artificial neural networks application for identification and control in environmental engineering: Biological and chemical systems with uncertain models. *Annual Reviews in Control*. doi:10.1016/j.arcontrol.2019.07.003.

Poznyak, A.S., Sanchez, E.N., and Yu, W. (2001). *Differential Neural Networks for Robust Nonlinear Control*. World Scientific Publishing.

Tingshu Hu and Zongli Lin (2003). Properties of the composite quadratic lyapunov functions. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 5, 5197–5202 Vol.5. doi: 10.1109/CDC.2003.1272462.

Xu, Z., Peng, D., and Li, X. (2019). Synchronization of chaotic neural networks with time delay via distributed delayed impulsive control. *Neural Networks*, 118, 332 – 337. doi:10.1016/j.neunet.2019.07.002.