Neuroidentifier for a class of nonlinear systems: A Sliding Modes approach*

Alejandro Guarneros-Sandoval[1], Mariana Ballesteros[2], Isaac Chairez[3]

Abstract— **This article focuses on designing a neural identifier for non-linear systems using an Integral Sliding Mode (ISM) approach. A Differential Neural Network (DNN) structure identifies a non-linear system; meanwhile, the learning laws for stabilizing the identification error are obtained from the Lyapunov Stability analysis. An integral sliding surface is proposed to provide the convergence of the system identifier weights to the ideals to approximate the non-linear system. The results for identifying the Van Der Poll Oscillator and the Inverted Pendulum show better identification error reduction performance than a traditional DNN.**

Index Terms— **Differential Neural Networks,System identifier,Integral Sliding Modes**

I. Introduction

Most engineering tasks require the use of parametric and non-parametric models to achieve a desired goal [1], [2]. Although modeling of the plant or process is desirable, complex elements and accessible information require a trade-off between accuracy and practical implementations. Therefore, a common choice is to use non-parametric models to approximate the system dynamics [3].

One of the main non-parametric models is the neural network topology [4]. Neural networks can estimate almost any function with an arbitrary accuracy based on the Stone-Weistrass Theorem to approximate system dynamics dependent on the number of parameters used [5]. Neural networks can be divided according to their structure into static and dynamic, the former working as an input-output relation used for regression, classification, and approximation problems for time-independent tasks [6]. On the other hand, dynamic neural networks are suitable for problems involving time dependencies, such as approximating signals or state equations of a plant or process due to their recurrent nature [7].

Dynamic neural networks can be represented by differential equations, known as Differential Neural Networks (DNN) [8]. Analyzing the equilibrium point of a system identification error around the origin of dynamic systems in continuous time is possible. This procedure results in adaptive laws for the neural network weights. The design and implementation of DNN rely on the task to be solved and the system properties. A common approach is to use system identification to propose control schemes tackling system uncertainties [8], [9]. However, the identification process influences subsequent tasks such as control implementation or prediction. Thus, an important issue is the convergence of the identifier to the system with certain properties, such as finite-time convergence.

To assure a finite time convergence, a common approach is the Sliding Mode (SM) methodology to force a system into a desired sliding surface employing a control parameter [10].

ISM is a methodology that aims to eliminate the reaching phase by considering an integral function related to the analyzed system, improving the finite-time convergence [11]. The research field associated with the control of systems using ISM has been investigated for many years. However, the ISM application for system identification has not been addressed to the best knowledge of the authors. Therefore, implementing the ISM to design the input parameter that modifies DNN weights to ensure that the identification error reaches a region near a stability point in finite time is the main contribution of this work.

II. Problem definition

The main goal is to create a DNN structure based on ISM to identify a certain class of dynamical systems.

A. Plant description

Consider a class of dynamical systems as follows

$$\dot{x} = \bar{f}(x,u) + \xi(x,u),\ x(0) = x_0, \tag{1}$$

where $x \in \mathbf{X} \subset \mathbb{R}^n$ is the state vector belonging to all the possible trajectories of the system $\mathbf{X} = \{x \in \mathbb{R}^n \mid \|x\|^2 < x^+ < +\infty\}$; $\forall t \in \mathbb{R}^+ \cup \{0\}$, x^+ is a positive constant, and $x_0 \in \mathbf{X}$, is the initial condition. The limit for x is known in advance. The variable $u \in \mathbf{U}^{adm}$ is the control action belonging to an admissible set $\mathbf{U}^{adm} = \{u \in \mathbb{R}^m | \|u\|^2 \leq u^+\}$; $\forall t \in \mathbb{R}^+ \cup \{0\}$, and $u^+ \in \mathbb{R}^+$. The vector field $\bar{f} : \mathbf{X} \times \mathbf{U}^{adm} \to \mathbb{R}^n$ represents the evolution of the unperturbed non-linear dynamical system. The piecewise continuous and integrable term $\xi : \mathbf{X} \times \mathbf{U}^{adm} \times \mathbb{R}^+ \to \mathbb{R}^n$ represents perturbation effects on nominal dynamics from the internal uncertainties due to the imprecise modeling and from any integrable exogenous

*This work was partly supported by the Instituto Politécnico Nacional under Grant: SIP-20250253. Isaac Chairez thanks the Tecnologico de Monterrey under the grant associated with the research program Challenge-Based Research Funding Program 2022 project with ID IJXT070-22TE6000 and 2023 with ID and IJXT070-23EG60002. Alejandro Guarneros-Sandoval is sponsored by a Mexican scholarship from the SECIHTI, CVU: 966608

[1]Tecnologico de Monterrey, School of Engineering and Sciences, Zapopan, Jalisco, Mexico. email: a01337659@tec.mx

[2]Medical Robotics and Biosignals Lab, CIDETEC, Instituto Politécnico Nacional, Gustavo A. Madero, Mexico City. email: mballesterose@ipn.mx

[3]Tecnologico de Monterrey, Institute of Advanced Materials for Sustainable Manufacturing, Zapopan, Jalisco, Mexico. email: isaac.chairez@tec.mx

disturbances and is assumed to be continuous and bounded in its nature.

For the class of system representation in (1), the following assumptions are made:

Assumption 1: The function $\bar{f} : \mathbf{X} \times \mathbf{U}^{adm} \rightarrow \mathbb{R}^n$ is locally Lipschitz continuous w.r.t. x with Lipschitz constant L_f, i.e.,

$$\|\bar{f}(x^{(1)}, u) - \bar{f}(x^{(2)}, u)\|^2 \leq L_f \|x^{(1)} - x^{(2)}\|^2, \qquad (2)$$

where $x^{(1)},\ x^{(2)} \in \mathbf{X}$.

Assumption 2: The term $\xi : \mathbf{X} \times \mathbf{U}^{adm} \rightarrow \mathbb{R}^n$ belongs to the set Ξ^{adm} defined by

$$\Xi^{adm} := \{\xi(x, u) \in \mathbb{R}^n | \sup_{t \geq 0} \|\xi(x, u)\|^2 \leq \xi^+\}. \qquad (3)$$

B. Integral Sliding Mode Differential Neural Network (SMDNN)

Based on the results from the approximation theory proposed originally in [5], the dynamics in (1) can have the following DNN identifier realization:

$$\begin{aligned} \frac{d\hat{x}}{dt} &= h(x, \hat{x}, u|\hat{W}_i)\ , \\ \frac{d\hat{W}_i}{dt} &= \Phi_i(\hat{W}_i, x, \hat{x}, u)\ , \\ & \quad i = 1, 2, 3, \end{aligned} \qquad (4)$$

where $\hat{x} \in \mathbb{R}^n$ is the state of the identifier, $h(.) \colon \mathbf{X} \times \mathbb{R}^n \times \mathbf{U}^{adm} \rightarrow \mathbb{R}^n$ is the function parameterized with a set of weight matrices $\hat{W}_i$, $\hat{W}_i \in \mathbb{R}^{n \times r_i}$, for the neuron numbers r_1, r_2 and r_3. A learning law as given in the second equation of (4) is proposed to determine adaptively the elements of the weight matrix in a non-parametric manner. The adaptive learning law $\Phi_i(.)$ is designed such that the identification error $e_t = x - \hat{x}$ converges to the origin and is a practical, stable equilibrium point of the identification error space, i.e., to design an SMDNN identifier for (1). In this work, based on a specific structure introduced in [8], we propose a novel structure for the identifier function $h(.)$ with discontinuous elements depending on the identification error.

C. SMDNN Approximation Model

For the design of the identifier, we consider the approximation of the term $\bar{f}(x, u)$ in (1) using ANN approximation properties [5]. Such approximation can be represented by

$$\bar{f}(x, u) = f_{\mathbf{N}}(x, u) + \tilde{f}(x, u)\ . \qquad (5)$$

The term $f_{\mathbf{N}}(x, u)$ in (5) has the following structure

$$\begin{aligned} f_{\mathbf{N}}(x, u) := (A + W_1 \sigma_1(x))x + W_2 \sigma_2(x) \\ +(B + W_3 \sigma_3(x))u\ , \end{aligned} \qquad (6)$$

where $\tilde{f}(x, u)$ is the approximation error, $A \in \mathbb{R}^{n \times n}$ is a Hurwitz matrix, the matrix $W_1 \in \mathbb{R}^{n \times r_1}$ approximates nonlinear system state affine elements. The matrix $W_2 \in \mathbb{R}^{n \times r_2}$ allows the estimation of nonlinear elements. The matrix $B \in \mathbb{R}^{n \times m}$ corresponds to the lineal control affine element, while $W_3 \in \mathbb{R}^{n \times r_3}$ approximates non linear control affine elements.

Equation (6) represents the SMDNN like structure. The activation functions $\sigma_1 \colon \mathbf{X} \rightarrow \mathbb{R}^{r_1 \times n}$, $\sigma_2 \colon \mathbf{X} \rightarrow \mathbb{R}^{r_2}$ and $\sigma_3 \colon \mathbf{X} \rightarrow \mathbb{R}^{r_3 \times m}$ have the elements as shown below

$$\begin{aligned} \sigma_{1_{(i,j)}} &= \frac{1}{1 + e^{-c_{i,j}^\top x}} - d_{i,j}; \quad c_{i,j} \in \mathbb{R}^n;\ d_{i,j} \in \mathbb{R}; \\ & \quad i = 1, \ldots, r_1;\ j = 1, \ldots, n; \\ \sigma_{2_{(i)}} &= \frac{1}{1 + e^{-\overline{c}_i^\top x}} - \overline{d}_i; \\ \overline{c}_i &\in \mathbb{R}^n; \quad \overline{d}_i \in \mathbb{R}; i = 1, \ldots, r_2; \\ \sigma_{3_{(i,j)}} &= \frac{1}{1 + e^{-\breve{c}_{i,j}^\top x}} - \breve{d}_{i,j}; \quad \breve{c}_{i,j} \in \mathbb{R}^n;\ \breve{d}_{i,j} \in \mathbb{R}; \\ & \quad i = 1, \ldots, r_3;\ j = 1, \ldots, m. \end{aligned} \qquad (7)$$

Notice that the smooth structure of sigmoidal functional allows claiming that $\sigma_1(\cdot)$, $\sigma_2(\cdot)$ and $\sigma_3(\cdot)$ fulfill the sector condition [8],

$$\begin{aligned} \|\sigma_i(x_a) - \sigma_i(x_b)\|^2 \leq L_{\sigma_i} \|x_a - x_b\|^2;\ \ x_a, x_b \in \mathbb{R}^n \\ \|\sigma_i(\cdot)\|^2 \leq L_{\sigma_i}^+; \quad L_{\sigma_i}, L_{\sigma_i}^+ \in \mathbb{R}^+; \quad i = 1, \ldots, 3. \end{aligned} \qquad (8)$$

Other activation functions can be used as long as they satisfy the aforementioned sector condition.

Assumption 3: The approximation error is represented by a monotonically decreasing function $\tilde{f}$ with respect to the number of elements in the SMDNN approximation weights and the activation function parameters as

$$\begin{aligned} \left\|\tilde{f}(x, u)\right\|^2 \leq f_0; \quad f_0 \in \mathbb{R}^+; \\ \forall x \in \mathbf{X}; \quad \forall u \in \mathbf{U}^{adm}. \end{aligned} \qquad (9)$$

Taking into account the error function properties express in (9), and by the approximation capabilities demonstrated in [12], considering the kind of systems given by the right-hand side of equation (1), there exist a Hurwitz matrix $A \in \mathbb{R}^{n \times n}$ and input matrix $B \in \mathbb{R}^{n \times m}$ and weight matrices, $W_1 \in \mathbb{R}^{n \times r_1}$, $W_2 \in \mathbb{R}^{n \times r_2}$ and $W_3 \in \mathbb{R}^{n \times r_3}$ such that, for a given $\epsilon_i, i = \{0, 1, 2\}$, the following inequality is valid

$$f_0 \leq \epsilon_0 + \epsilon_1 x^+ + \epsilon_2 u^+, \qquad (10)$$

with the approximation structure given in (6), and the respective sigmoidal functions (7) with r_1, r_2 and r_3 number of elements.

D. SMDNN Identifier

Taking into consideration that the DNN structure (6) is a valid approximation of the systems considered in this manuscript (1), it is feasible to propose the following differential equation to define the suggested DNN identifier:

$$\begin{aligned} \frac{d\hat{x}}{dt} &= A\hat{x} + \hat{W}_1 \sigma_1(\hat{x}) x + \hat{W}_2 \sigma_2(\hat{x}) \\ &\quad +(B + \hat{W}_3 \sigma_3(\hat{x}))u + K\text{Sign}(S), \end{aligned} \qquad (11)$$

where $\hat{x} \in \mathbb{R}^n$ are the states of the system identifier, $\hat{W}_i$, $i = \{1, 2, 3\}$, are the weights that approximate to the best fitted W_i. The approximation error between the best-fit weights and the approximate is defined as $\tilde{W}_i = W_i - \hat{W}_i$, and $e = x - \hat{x}$ is the

identification error. $K \in \mathbb{R}^{n\times n}$ is a constant positive definite matrix selected in order to tackle the error approximation with the following equation

$$f_0 \leq \lambda_{min}\{K\}. \tag{12}$$

The vector-valued function Sign : $\mathbf{X} \rightarrow \mathbf{X}$; is formed as $\text{Sign}(e) = [\text{sign}(e_i)]_{i=\in[1,\dots n]}$ where

$$\text{sign}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ [-1,1] & \text{if } z = 0 \\ -1, & \text{if } z \leq 0 \end{cases} \tag{13}$$

and $z \in \mathbb{R}$. The variable S refers to the sliding surface to be designed to achieve the finite time convergence. The structure given above combines pseudolinear approximation techniques such as in [13] and the use of SM to tackle uncertainties such as [14].

Consider the learning laws for (4) as

$$\begin{aligned} \frac{d\hat{W}_1}{dt} &= k_1^{-1}S(\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})^\top x_i + x^\top\sigma_1^\top(\hat{x})), \\ \frac{d\hat{W}_2}{dt} &= k_2^{-1}S\big((D_{\sigma_2}e + \nu_{\sigma_2})^\top + \sigma_2^\top(\hat{x})\big), \\ \frac{d\hat{W}_3}{dt} &= k_3^{-1}S\big(\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})^\top u_i + u^\top\sigma_3(\hat{x})\big), \end{aligned} \tag{14}$$

where $\nu_{\sigma_{1,i}}$, ν_{σ_2} and $\nu_{\sigma_{3,i}}$ are the upper bound of the derivative errors. $D_{\sigma_{1,i}}$, D_{σ_2} and $D_{\sigma_{3,i}}$ are the derivative approximation from the Mean Value Theorem (MVT) and k_1^{-1}, k_2^{-1}, and $k_3^{-1} \in \mathbb{R}^{n\times n}$ positive definite symmetric matrices. Therefore, the sliding surface is designed as

$$\begin{aligned} S = e - e_0 - \int_{t_0}^{t_f}\Big(&Ae + \hat{W}_1\sigma_1(x)x - \hat{W}_1\sigma_1(\hat{x})x \\ &+\hat{W}_2\sigma_2(x) - \hat{W}_2\sigma_2(\hat{x}) + \hat{W}_3\sigma_3(x)u - \hat{W}_3\sigma_3(\hat{x})u\Big)d\tau, \end{aligned} \tag{15}$$

where e_0 is the initial identification error.

By the MVT, it is possible to approximate the difference of a function with a difference in their arguments as the derivative of the function multiplied by the difference of its inputs, i.e.,

$$\begin{aligned} \hat{W}_1\sigma_1(x)x - \hat{W}_1\sigma_1(\hat{x})x &= \hat{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i; \\ \hat{W}_2\sigma_2(x) - \hat{W}_2\sigma_2(\hat{x}) &= \hat{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}); \\ \hat{W}_3\sigma_3(x)u - \hat{W}_3\sigma_3(\hat{x})u &= \hat{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i. \end{aligned} \tag{16}$$

Then, the sliding surface in (15), based on (16), can be express as the following

$$\begin{aligned} S = e - e_0 - \int_{t_0}^{t_f}\Big(&Ae + \hat{W}_1\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\hat{W}_2\left(D_{\sigma_2}e + \nu_{\sigma_2}\right) + \hat{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i\Big)d\tau. \end{aligned} \tag{17}$$

The derivative of the sliding function is computed as

$$\begin{aligned} \dot{S} = &Ae + W_1\sigma_1(x)x - \hat{W}_1\sigma_1(\hat{x})x + W_2\sigma_2(x) \\ &-\hat{W}_2\sigma_2(\hat{x}) + W_3\sigma_3(x)u - \hat{W}_3\sigma_3(\hat{x})u \\ &-K\text{Sign}(S) + \tilde{f}(x,u) \\ &-\Big(Ae + \hat{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\hat{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}) \\ &+\hat{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i\Big), \end{aligned} \tag{18}$$

adding the terms $\pm W_1\sigma_1(\hat{x})x$, $\pm W_2\sigma_2(\hat{x})$ and $\pm W_3\sigma_3(\hat{x})u$ into (18) allows the representation of the sliding surface derivative as

$$\begin{aligned} \dot{S} = &Ae + W_1\tilde{\sigma}_1(x,\hat{x})x + \tilde{W}_1\sigma_1(\hat{x})x \\ &+W_2\tilde{\sigma}_2(x,\hat{x}) + \tilde{W}_2\sigma_2(\hat{x}) \\ &+W_3\tilde{\sigma}_3(x,\hat{x})u + \tilde{W}_3\sigma_3(\hat{x})u - K\text{Sign}(S) + \tilde{f}(x,u) \\ &-\Big(Ae + \hat{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\hat{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}) + \hat{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i\Big). \end{aligned} \tag{19}$$

Applying the MVT for the elements $W_1\tilde{\sigma}_1(x,\hat{x})x$, $W_2\tilde{\sigma}_2(x,\hat{x})$ and $W_3\tilde{\sigma}_3(x,\hat{x})u$, we can derive

$$\begin{aligned} \dot{S} = &Ae + W_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\tilde{W}_1\sigma_1(\hat{x})x + W_2(D_{\sigma_2}e + \nu_{\sigma_2}) + \tilde{W}_2\sigma_2(\hat{x}) \\ &+W_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i + \tilde{W}_3\sigma_3(\hat{x})u \\ &-K\text{Sign}(S) + \tilde{f}(x,u) \\ &-\Big(Ae + \hat{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\hat{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}) + \hat{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i\Big). \end{aligned} \tag{20}$$

Eliminating similar terms, and with the definition of $\tilde{W}_i = W_i - \hat{W}_i$, the sliding surface derivative results in

$$\begin{aligned} \dot{S} = &\tilde{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i + \tilde{W}_1\sigma_1(\hat{x})x \\ &+\tilde{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}) + \tilde{W}_2\sigma_2(\hat{x}) \\ &+\tilde{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i + \tilde{W}_3\sigma_3(\hat{x})u_i \\ &-K\text{Sign}(S) + \tilde{f}(x,u). \end{aligned} \tag{21}$$

Proposing the Lyapunov function as

$$V = S^\top S + \frac{k_1}{2}\|\tilde{W}_1\|_F^2 + \frac{k_2}{2}\|\tilde{W}_2\|_F^2 + \frac{k_3}{2}\|\tilde{W}_3\|_F^2, \tag{22}$$

its derivative, taking into account $\dfrac{d\tilde{W}_i}{dt} = -\dfrac{d\hat{W}_i}{dt}$, can be expressed as

$$\begin{aligned} \dot{V} = S^\top\dot{S} - k_1\text{tr}\{\tilde{W}_1^\top\tfrac{d\hat{W}_1}{dt}\} - k_2\text{tr}\{\tilde{W}_2^\top\tfrac{d\hat{W}_2}{dt}\} \\ -k_3\text{tr}\{\tilde{W}_3^\top\tfrac{d\hat{W}_3}{dt}\}. \end{aligned} \tag{23}$$

Substituting the sliding surface derivative

$$\begin{aligned} \dot{V} = S^\top\Big(&\tilde{W}_1\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})x_i \\ &+\tilde{W}_1\sigma_1(\hat{x}))x + \tilde{W}_2(D_{\sigma_2}e + \nu_{\sigma_2}) + \tilde{W}_2\sigma_2(\hat{x}) \\ &+\tilde{W}_3\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})u_i \\ &+\tilde{W}_3\sigma_3(\hat{x})u + \tilde{f}(x,u) - K\text{Sign}(S)\Big) - k_1\text{tr}\{\tilde{W}_1^\top\tfrac{d\hat{W}_1}{dt}\} \\ &-k_2\text{tr}\{\tilde{W}_2^\top\tfrac{d\hat{W}_2}{dt}\} - k_3\text{tr}\{\tilde{W}_3^\top\tfrac{d\hat{W}_3}{dt}\}. \end{aligned} \tag{24}$$

Grouping similar terms

$$\begin{aligned} \dot{V} \leq\ &S^\top\Big(\tilde{f}(x,u) - K\text{Sign}(S)\Big) \\ &-k_1\text{tr}\{\tilde{W}_1^\top(\tfrac{d\hat{W}_1}{dt} - k_1^{-1}S\Theta_1)\} \\ &-k_2\text{tr}\{\tilde{W}_2^\top(\tfrac{d\hat{W}_2}{dt} - k_2^{-1}S\big(\Theta_2\big)\} \\ &-k_3\text{tr}\{\tilde{W}_3^\top(\tfrac{d\hat{W}_3}{dt} - k_3^{-1}S\big(\Theta_3\big)\} \\ \Theta_1 = &\textstyle\sum_{i=1}^{n}(D_{\sigma_{1,i}}e + \nu_{\sigma_{1,i}})^\top x_i + x^\top\sigma_1^\top(\hat{x}) \\ \Theta_2 = &(D_{\sigma_2}e + \nu_{\sigma_2})^\top + \sigma_2^\top(\hat{x}) \\ \Theta_3 = &\textstyle\sum_{i=1}^{n}(D_{\sigma_{3,i}}e + \nu_{\sigma_{3,i}})^\top u_i + u^\top\sigma_3(\hat{x})). \end{aligned} \tag{25}$$

Then, taking into account the learning laws of the weights in (14), we obtain

$$\dot{V} \leq S^\top\Big(\tilde{f}(x,u) - K\text{Sign}(S)\Big), \tag{26}$$

then we can find the upper bound as the minimum eigenvalue of the matrix K, and separate the terms, such as

$$\dot{V} \leq -\lambda_{min}\{K\}|S| + S^{\top}\tilde{f}(x,u), \tag{27}$$

Using the Cauchy-Schwarz inequality, we can obtain the following upper bound

$$\dot{V} \leq -|S|\left(\lambda_{min}\{K\} - f_0\right), \tag{28}$$

selecting the gain K such that $f_0 \leq \lambda_{min}\{K\}$, the Lyapunov function converges asymptotically to the origin.

III. Results

To show the identification error convergence exhibited by the ISM identifier 11, this work compares the proposed approach against a traditional DNN [8] in two numerical examples, the Van Der Poll oscillator and the inverted pendulum [15], [16]. An essential remark for both numerical examples is the derivative approximation errors, i.e., $\nu_{\sigma_{1,i}}$, ν_{σ_2} and $\nu_{\sigma_{3,i}}$ are considered to be insignificant and set to zero.

A. Van Der Poll Oscillator

The Van Der Poll Oscillator dynamic is characterized by the following dynamics

$$\begin{aligned} \dot{x}_1 &= x_2 + \eta_1 \\ \dot{x}_2 &= -x_1 + x_2\left(1 - x_1^2\right) + \eta_2 + u \end{aligned} \tag{29}$$

where $[x_1 \quad x_2]^{\top} \in \mathbb{R}^n$ are related to the system states, and $u \in \mathbb{R}^m$ is the input to the system. The control is set as $u = 0.1\sin(t)$; the external disturbances η_1 and η_2 are computed as a combination of sine functions

$$\begin{aligned} \eta_1 &= -0.01\sin(0.1t) + 0.05\sin(4t) \\ &\quad +0.01\sin(100t) \\ \eta_2 &= 0.1\sin(10t) + 0.05\sin(0.1t) - 0.5\sin(50t). \end{aligned} \tag{30}$$

The weight matrices W_1, W_2 and W_3 have $r_1 = 2$, $r_2 = 3$ and $r_3 = 4$ neurons. Based on the universal approximation of neural networks with stochastic parameters [17], the sigmoidal parameters have a uniform distribution between -2.5 and 2.5 with MATLAB fixed seed on 2. The initial weights use a uniform distribution between -2.5 and 2.5. The remaining learning constants were set as

$$\begin{aligned} k_1 = k_2 = k_3 = 10 * I_n; \quad K = 20; \\ A = \begin{bmatrix} 0 & 10 \\ -10 & -10 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{31}$$

The traditional DNN [8] used to compare the proposed SMDNN has the following structure and corresponding learning laws

$$\begin{aligned} \dot{\hat{x}} &= A_D\hat{x} + \hat{W}_{n1}\sigma_2(\hat{x}) + \hat{W}_{n2}\sigma_3(\hat{x})u; \\ \hat{W}_{n1} &\in \mathbb{R}^{n\times q_1}; \quad \hat{W}_{n2} \in \mathbb{R}^{n\times q_2}; \\ \dot{\hat{W}}_{n1} &= k_{n1}P_n e\sigma_2(\hat{x})^{\top}; \quad \dot{\hat{W}}_{n2} = k_{n2}P_n e u^{\top}\sigma_3(\hat{x})^{\top} \end{aligned} \tag{32}$$

where $\sigma_2 : \mathbb{R}^n \to \mathbb{R}^{q_1}$ and $\sigma_3 : \mathbb{R}^n \to \mathbb{R}^{q_2\times m}$ are sigmoidal activation functions with the same structure than (7). For the numerical example, $q_1 = 5$ and $q_2 = 6$. The activation function parameters are selected with a uniform distribution between $[-0.5, 0.5]$ with the MATLAB fixed seed of 3. The learning laws parameters were selected as

$$\begin{aligned} k_{n1} = k_{n2} = 100 * I_n; \quad P_n = I_n \\ A_D = \begin{bmatrix} -5 & 0 \\ 0 & -5.5 \end{bmatrix}. \end{aligned} \tag{33}$$

The identification process of both traditional DNN and SMDNN for x_1 shows a similar convergence to the system dynamics in less than 2 seconds (Figure 1). On the other hand, the identification of x_2 is initially faster with the traditional DNN as it is practically the same as the system state (Figure 2), where the proposed method has an initial separation from the system dynamics. Nevertheless, the zoom in both Figures exhibits a closer convergence of the SMDNN to both system dynamics; meanwhile, the traditional DNN has a small divergence between 4 and 5 seconds for x_2 dynamics.

Figure 3 shows the difference for the identification error

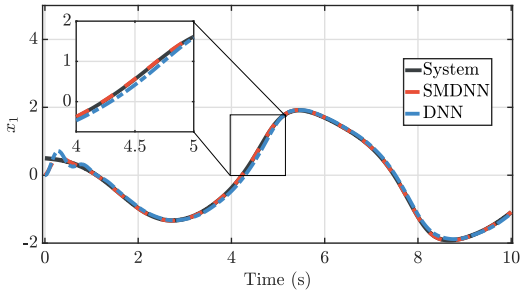

Fig. 1. System identification comparison for x_1. The zoom shows a closer convergence to the system by the proposed SMDNN.

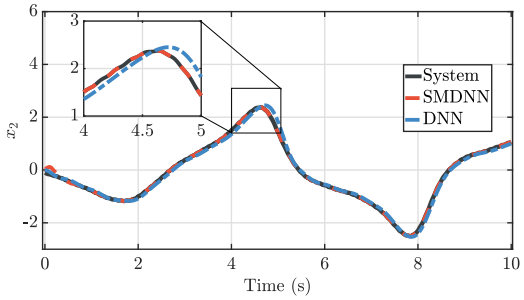

Fig. 2. System identification comparison for x_2. The traditional DNN diverges from the system between seconds 4 and 5.

convergence between the two DNNs. The proposed SMDNN converges faster to a region near the origin and with fewer oscillations than the traditional DNN, properties exhibited by the SM methodologies. Additionally, the traditional DNN identification error increases at the second 5, meanwhile the SMDNN remains close to the origin for the rest of the time.

Finally, Figure 4 presents W_1 dynamics by the SMDNN learning laws. It is worth remarking on the slow dynamics presented by the weights as the SMC forces the identifier to converge to the system trajectories and keep the identification error near the origin. This behavior allows the sigmoidal functions to represent the system dynamics in finite time.

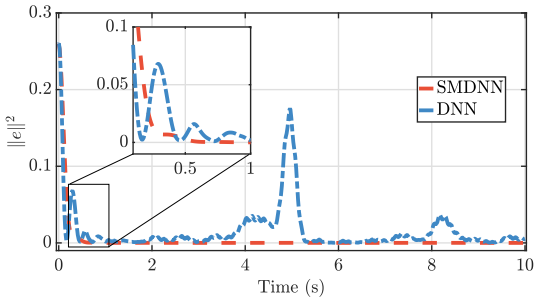

Fig. 3. The square of the identification error shows the characteristic convergence of the SM methodologies compared to the oscillations of the traditional DNN

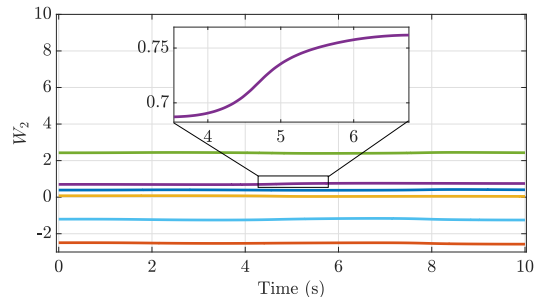

Fig. 4. W_1 dynamics, with an almost constant behaviour.

B. Inverted Pendulum

The second numerical example is the inverted pendulum [15], setting the moment of inertia $J = 1\text{kg} \cdot \text{m}^2$, mass of the inverted pendulum $M = \frac{1}{3}$ kg, gravitational acceleration $9.8\frac{\text{m}}{\text{s}^2}$ length of the inverted pendulum $h = 1.5\text{m}$ and friction factor $f_c = 0.2$. The inverted pendulum dynamics obeys the differential equation

$$\begin{aligned} \dot{\theta}_1 &= \theta_2 + \nu \\ \dot{\theta}_2 &= \tfrac{1}{J}u - \tfrac{\text{Mgh}}{\text{J}} \sin(\theta_1) - \tfrac{f_c}{\text{J}}\dot{\theta}_1 \\ \nu &= 5d_1\theta_1 \sin(d_2\theta_2), \end{aligned} \tag{34}$$

where $\theta_1 \in \mathbb{R}$ and $\theta_2 \in \mathbb{R}$ are the angle and angular velocity for the inverted pendulum. The control $u \in \mathbb{R}$ has an oscillation as with the Van Der Poll oscillator $u = 0.1\sin(t)$, and the perturbation parameters for ν are d_1 and d_2 are selected from a uniform distribution between $[-0.5, 0.5]$. The number of neurons for both neural networks is $r_1 = r_2 = r_3 = 2$ and $q_1 = q_2 = 3$. The matrix A for both networks is selected as

$$A = A_D = \begin{bmatrix} 0 & 5 \\ -20 & -20 \end{bmatrix}. \tag{35}$$

The remaining parameters for both DNN, the learning laws gains and the discontinuous element gain, were the same as for the Van Der Poll numerical example.

The system trajectory converges to the origin due to the presence of a periodic control input. Therefore, the identification processes for θ_1 (Figure 5) and θ_2 (Figure 6) show that the SMDNN converges faster and closer to the state trajectory compared to traditional DNN. Compared to the Van Der Poll oscillator, the traditional DNN shows more divergence from the system state trajectory, especially in θ_2.

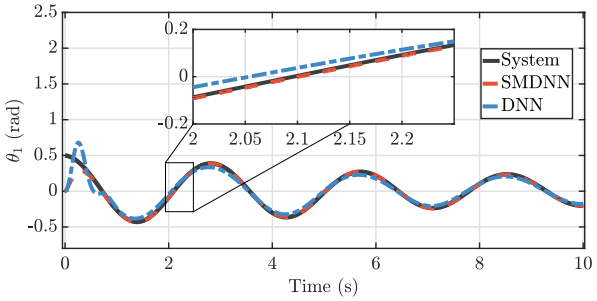

Fig. 5. System identification for θ_1. A highlight of the closer convergence to the system by the proposed method is shown.

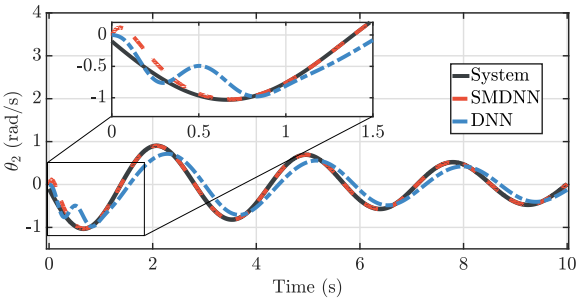

Fig. 6. System identification for θ_2. The traditional DNN needed more time for the converging to θ_2 trajectory.

The identification error stability for the inverted pendulum shows the finite time convergence to a region near the origin by the SMDNN. Once the network reaches this region, the discontinuous elements of the weights maintain it there, as can be seen in Figure (7). On the other hand, the traditional DNN presents its characteristic asymptotic convergence with vanishing oscillation as time increases. Finally, similar to the Van Der Poll numerical example, the SMDNN weights remain almost constant with small changes due to system behavior (Figure 8).

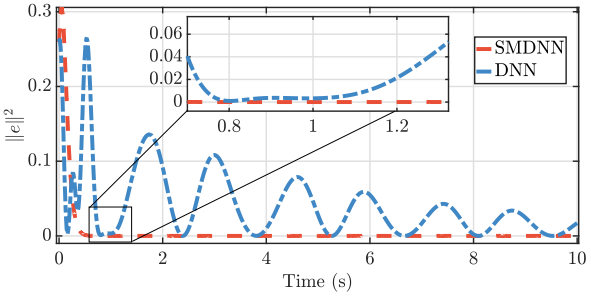

Fig. 7. Identification error convergence dynamics for both DNN.

IV. Conclusion

A novel DNN structure, the SMDNN is presented in this work. The system topology was selected to have a state-

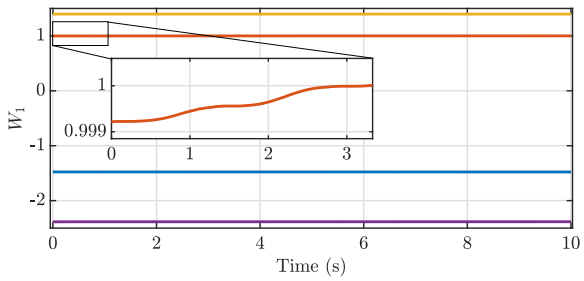

Fig. 8. SMDNN weights W_2 showing small variations during the simulation.

input affine structure based on successful implementations of previous works. The learning laws for the SMDNN identifier were designed using the Lyapunov stability analysis, including discontinuous elements. The discontinuous elements derive from the application of an ISM control-like structure, which enforces the identifier to reach the system trajectory in a finite time. The SMDNN was compared against a traditional DNN, simulating the identification process in the Van Der Poll Oscillator and an inverted pendulum, demonstrating a clear advantage over the asymptotic convergence in both scenarios. A remark for improvement of the DNN is the application of less restrictive discontinuous functions, such as the hyperbolic tangent, and the inclusion of the derivative approximation error resulting from the Mean Value theorem. Nevertheless, the findings enforce the application of the SMDNN to develop closed-loop controllers in future research work.

References

[1] R. Kumar, S. Srivastava, Externally Recurrent Neural Network based identification of dynamic systems using Lyapunov stability analysis, ISA Transactions 98 (2020) 292 – 308. doi:10.1016/j.isatra.2019.08.032.

[2] O. Nelles, Nonlinear system identification: from classical approaches to neural networks and fuzzy models, Springer Science & Business Media, 2013.

[3] H. V. H. Ayala, D. Habineza, M. Rakotondrabe, L. dos Santos Coelho, Nonlinear black-box system identification through coevolutionary algorithms and radial basis function artificial neural networks, Applied Soft Computing 87 (2020) 105990.

[4] D. Chen, S. Li, Q. Wu, L. Liao, Simultaneous identification, tracking control and disturbance rejection of uncertain nonlinear dynamics systems: A unified neural approach, Neurocomputing 381 (2020) 282 – 297.

[5] N. E. Cotter, The Stone-Weierstrass theorem and its application to neural networks, IEEE Transactions on Neural Networks 1 (4) (1990) 290–295.

[6] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[7] F. Lewis, S. Jagannathan, A. Yesildirak, Neural network control of robot manipulators and non-linear systems, CRC Press, 1998.

[8] A. S. Poznyak, E. N. Sanchez, W. Yu, Differential neural networks for robust nonlinear control: identification, state estimation and trajectory tracking, World Scientific, 2001.

[9] M. Ballesteros, I. Chairez, A. Poznyak, Robust optimal feedback control design for uncertain systems based on artificial neural network approximation of the Bellman's value function, Neurocomputing 413 (2020) 134–144. doi:10.1016/j.neucom.2020.06.085.

[10] V. Utkin, A. Poznyak, Y. V. Orlov, A. Polyakov, Road Map for Sliding Mode Control Design, SpringerBriefs in Mathematics, Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-41709-3.

[11] S. M.K., R. K. P., J. K.S., Control of Stewart platform using adaptive smooth integral sliding mode algorithm, ISA Transactions 156 (2025) 99–108. doi:10.1016/j.isatra.2024.11.009.

[12] M. Leshno, V. Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks 6 (6) (1993) 861–867. doi:10.1016/S0893-6080(05)80131-5.

[13] A. Guarneros-Sandoval, M. Ballesteros, R. Q. Fuentes-Aguilar, I. Chairez, State-input affine approximate modeling based on a differential neural network identifier, Applied Mathematical Modelling 125 (2024) 544–554. doi:10.1016/j.apm.2023.08.039.

[14] I. Chairez, Finite time convergent learning law for continuous neural networks, Neural networks 50 (2014) 175–182.

[15] S. Xue, B. Luo, D. Liu, Event-Triggered Adaptive Dynamic Programming for Unmatched Uncertain Nonlinear Continuous-Time Systems, IEEE Transactions on Neural Networks and Learning Systems 32 (7) (2021) 2939–2951. doi:10.1109/TNNLS.2020.3009015.

[16] M. Li, D. Wang, J. Ren, J. Qiao, Advanced optimal tracking integrating a neural critic technique for asymmetric constrained zero-sum games, Neural Networks 177 (2024) 106388. doi:10.1016/j.neunet.2024.106388.

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