

## Research Article

# About Extracting Dynamic Information of Unknown Complex Systems by Neural Networks

Eloy Irigoyen <sup>1</sup>, Antonio Javier Barragán,<sup>2</sup> Mikel Larrea <sup>1</sup> and José Manuel Andújar <sup>2</sup>

<sup>1</sup>UPV/EHU, Alda, Urquijo, s/n, Bizkaia, Spain

<sup>2</sup>UHU, ETSI, Campus de El Carmen, Huelva, Spain

Correspondence should be addressed to Eloy Irigoyen; [eloy.irigoyen@ehu.eus](mailto:eloy.irigoyen@ehu.eus)

Received 18 October 2017; Revised 25 April 2018; Accepted 16 May 2018; Published 8 July 2018

Academic Editor: Marek Reformat

Copyright © 2018 Eloy Irigoyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a straightforward methodology based on neural networks (NN) which allows to obtain relevant dynamic information of unknown nonlinear systems. It provides an approach for cases in which the complex task of analyzing the dynamic behaviour of nonlinear systems makes it excessively challenging to obtain an accurate mathematical model. After reviewing the suitability of multilayer perceptrons (MLPs) as universal approximators to replace a mathematical model, the first part of this work presents a system representation using a model formulated with state variables which can be exported to a NN structure. Considering the linearization of the NN model in a mesh of operating points, the second part of this work presents the study of equilibrium states in such points by calculating the Jacobian matrix of the system through the NN model. The results analyzed in three case studies provide representative examples of the strengths of the proposed method. Conclusively, it is feasible to study the system behaviour based on MLPs, which enables the analysis of the local stability of the equilibrium points, as well as the system dynamics in its environment, therefore obtaining valuable information of the system dynamic behaviour.

## 1. Introduction

The innumerable strategies and new proposals in the control system area are generally based on the knowledge of the system to be controlled. In some cases, its model is achieved by mathematical and analytical procedures, because such model is a simplified representation of the system, or the problem to study is highly restricted to a specific operating point. Although the modelling is often approached by the means of mathematical and analytical methods, with the growing complexity involved in the systems and the requirements of accurate representation, such approaches turn into an increasingly challenging solution.

Consequently, considering the maturity of computational intelligence (CI) techniques and new enabler technologies, CI methods represent an attractive alternative to develop accurate modelling and control solutions for highly complex models [1–4].

Taking into account the need to obtain precise models of complex systems which are applicable in a wide operating

range, the CI techniques are an ideal method to reproduce the behaviour of a complex system [2, 5, 6] or, as it will be presented in this work, to analyze the dynamics of systems.

When studying a process in engineering, in biomedical field, in natural sciences, and even in social systems, approaching the analysis from a dynamic point of view can be very attractive and convenient, depending on the focus of such study. A dynamic analysis of the system can provide wide and very rich information related to how the system will respond under certain inputs. Moreover, it can allow to study its dynamic behaviour through the analysis of the stability in open-loop, both locally and globally. In addition, it will be possible to study whether certain nonlinear phenomena affect the system, for example bifurcations, saddle, and limit cycles [7, 8].

The methods traditionally applied in control engineering are based on linear approximations around several operating points of the system. This is suitable when problems are studied and solved in a local domain. However, there is a

trend to approach bigger problems with a more abstracted and global perspective, leading to the use of nonlinear methods [9–12]. At some point of complexity, and certainly if the system involves unknown parts and other uncertainties, an entirely formal analysis of the system becomes unsustainable [13–15].

One of the main reasons for the use of nonlinear models is based on the dynamics of linear systems, since conventional mathematical formulations are not rich enough to reproduce a series of phenomena that usually appear in the real life [2, 16, 17]. The dynamic behaviour of a linear system, without considering its order, is basically governed by the eigenvalues of the corresponding state matrix [8]. On the contrary, nonlinear systems show a much richer behaviour, with self-excited oscillations (i.e., limit cycles), aperiodic behaviours and critically sensitive to the initial conditions [7], and chaos [18], as well as other dynamic phenomena exclusive to nonlinear systems, such as the existence of multiple states of equilibrium and bifurcations [19], among others.

The typically appropriate initial approach to analyze nonlinear systems is to use a representation of the system by means of a mathematical model, generally represented in state variables. This is possible assuming that sufficient information and knowledge of the system is available to generate its state equations, provided that the system dynamic is not extremely complex. In many applications, current research deals with the study of unknown complex systems, whether due to a complex dynamic, high dimensionality, or lack of information about the physical relationships that govern the behaviour of the system. In such situations, the techniques from the field of intelligent control can help to improve these studies, as Barragán et al. present in [20, 21] using Fuzzy logic to define a formal methodology for analyzing the dynamic behaviour of nonlinear systems, or Grande et al. in [11] to extract qualitative models of spatial evolution from a chemical system. In the same sense, neural networks (NN) become a powerful technique, since they are able to model highly complex nonlinear systems from input-output data. Proper selection and training of a basic structure such as a multilayer perceptron (MLP) can accurately reproduce the behaviour of a nonlinear system. This modelling technique can be used, both qualitatively and analytically [6, 22–25], taking into account that MLPs are universal approximators, either for a function [26–29] or its derivative [30, 31]. Thus, although the system might be unknown, it is possible to obtain a NN model of its behaviour, representing its dynamics in the workspace studied. In a formal sense, a NN model is a mathematical model. Hence, from this NN model, it is possible to study several aspects of the real dynamics of the system, conditioned only by the high precision of the model. This can be achieved with an exhaustive experimental stage where the topology of a NN is selected, capable of faithfully reproducing the behaviour of the system with the expected precision. This approach in solving this problem allows dealing with nonlinear systems, where modeling by traditional mathematical techniques can be challenging.

During the analysis and design of control solutions, knowing the equilibrium states of a system, as well as the

stability of such states, is an aspect of great interest. When the model of the system is completely unknown, this information could help to clarify how the system works, even to ease the design of an appropriate control. It should be noted that despite the existence of recent works that present formal analysis methodologies based on Fuzzy logic [17, 20, 21, 32] the authors have not currently found any work focused from the NN point of view, under a general approach as presented in this proposal.

This work presents a straightforward and easy to use methodology for extracting information from unknown systems using NNs. The main objective of this proposal is to develop a method that allows obtaining information on the dynamics of nonlinear systems, when there is no mathematical model, neither accurate nor approximate, to analyze them. In these situations, any additional information reached by new methods is significant, especially when this information is related to the analysis of the presence of equilibrium states and their local stability, as presented below. In this work, an MLP neural network is trained with a set of measured values of inputs and outputs of supposedly unknown systems, in order to reproduce the behaviour of these systems. For this purpose, taking into account that the dynamic study aims at analyzing the behaviour of each nonlinear system in their corresponding equilibrium states, the dataset of examples to train the NN will be obtained from its entire operating range. More specifically, the equilibrium states of three nonlinear systems will be studied through their NN models, which reproduce their corresponding state variable models. The equilibrium states are reached by a precise linearization in a grid of operating points extracted from the NN models, and subsequently performing a study of local stability. Using this information, the local stability of equilibrium states is obtained, as well as the system dynamics in the vicinity of the studied points, achieving valuable information about the dynamic behaviour of the nonlinear system.

This paper is organized as follows: Section 2 presents the problem and the formulation associated with it, explaining how it will be dealt with throughout the document. In Section 3, the procedures to obtain the linearization of a system and its extension to a NN model are explained. Section 4 presents three case studies to demonstrate the proposed approach, based on solid results. Finally, this work finalizes with the corresponding conclusions.

## 2. Problem Formulation

A generic continuous dynamic system will be considered, represented by state variables  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ , where  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$  and  $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T$  depict a  $m$  input system of order  $n$  with  $u_i(t)$  representing the inputs and  $x_i(t)$  the state variables.  $\mathbf{f}$  is a static nonlinear map defined as  $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  [33]. An equivalent NN model, based on a MLP structure, which can estimate both the state variables of the continuous system  $\hat{\mathbf{x}}(t)$  and the system output  $\hat{\mathbf{y}}(t)$ , is represented by the equations in (1) [34–36].

$$\begin{aligned}\widehat{\mathbf{x}}(t|\boldsymbol{\theta}) &= \mathbf{g}(\boldsymbol{\varphi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}), \\ \widehat{\mathbf{y}}(t|\boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta})\widehat{\mathbf{x}}(t|\boldsymbol{\theta}),\end{aligned}\quad (1)$$

$$\text{being } \boldsymbol{\varphi}(t, \boldsymbol{\theta}) = \begin{bmatrix} \mathbf{x}(t|\boldsymbol{\theta}) \\ \mathbf{u}(t) \end{bmatrix}, \quad (2)$$

where  $\boldsymbol{\varphi}(t, \boldsymbol{\theta})$  is the regression vector and  $\boldsymbol{\theta}$  the vector of parameters ( $\mathbf{w}, \mathbf{W}$ ) of the NN, the inner weights, and the biases.  $\mathbf{g}(\cdot)$  is the function realized by the MLP, defined as  $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ . To model the evolution of each  $i$ th state variable, a MLP structure has to be trained adapting the parameters mentioned above. In the training process, the needed information is provided for both the states  $x_i(t)$ , with  $i = 1, \dots, n$ , and the system inputs  $u_j(t)$ , with  $j = 1, \dots, m$ .

By selecting a simple MLP structure that consists of one hidden layer of  $h$  neurons, with sigmoidal activation functions, and a linear output layer, the NN output  $O_{MLP}$  can be calculated by the following general expression [35]:

$$O_{MLP} = \sum_{h=1}^{n_h} W_{k,h} F_h(act_h) + W_{k,0}, \quad (3)$$

where  $n_h$  is the number of hidden neurons,  $k$  represents the output neuron (for the case of several NN outputs),  $W_{k,h}$  are the output layer weights and biases,  $F_h$  is the activation function of each hidden neuron, and  $act_h$  is the sum of weighted inputs to each hidden neuron, as shown in

$$act_h = \sum_{l=1}^{n+m} \omega_{h,l} \varphi_l(t) + \omega_{h,0}, \quad (4)$$

where  $\omega_{h,l}$  are the weights and biases of the hidden layer and  $\varphi_l(t)$  is the input vector to the NN, being  $n+m$  the vector dimension.

Taking into account that a NN will reproduce the evolution of each state variable  $\dot{x}_i(t)$ , the neural model can be related to the state model [35] as

$$\dot{x}_i(t) = f_i(\mathbf{x}(t), \mathbf{u}(t)) \equiv \widehat{\dot{x}}_i(t|\boldsymbol{\theta}), \quad (5)$$

$$\widehat{\dot{x}}_i(t|\boldsymbol{\theta}) = \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=1}^{n+m} \omega_{h,l} \varphi_l(t) + \omega_{h,0} \right] + W_{i,0}. \quad (6)$$

From the above representation, in order to simplify the methodology in studying the obtained NN models, each state variable will be modeled by a different NN.

### 3. Information Obtained from the Neural Network Model

After obtaining an accurate model of a system, it is a fact that this model can be used to obtain system information through well-known techniques. In this section, a very important technique is presented to study nonlinear systems in two phases, as required by the methodology of this work. Firstly, the linearization of a neural state model will be exposed in detail. Secondly, the study of the equilibrium states of an unknown nonlinear system will be presented. This study is

carried out through a NN model that reproduces the behaviour of the aforementioned nonlinear system. The study of the equilibrium states from the NN model, together with the study of their local stability from the linearization, allows to analyze the operational behaviour of a system from a qualitative point of view.

**3.1. Linearization of a Neural Model.** Linearization is one of the most commonly used techniques in solving design problems in the field of nonlinear control systems, even though it is necessary to point out that this is a technique not ideal in many situations where the effects of nonlinearities are not negligible. It is a very convenient technique for the control of not excessively complex systems or in situations when the dynamics of the system is approximately known in regions where the system behaviour is close to a linear one, basically around equilibrium states.

Thereby, apart from being a method that aims at the control of systems, linearization could be a powerful resource to obtain information from a nonlinear system. It could be considered that, except in some situations, the behaviour of a nonlinear system around an equilibrium state is analogous to the one observed after linearization of the system in such state [19, 37, 38]. So, the study and calculation of equivalent linear systems from a nonlinear NN model can be a powerful technique to obtain information concerning the real nonlinear system analyzed.

The generic state model, obtained from a nonlinear system, is represented by

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \text{where } \mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m &\longrightarrow \mathbb{R}^n.\end{aligned}\quad (7)$$

The first-order simplification of the Taylor series of the nonlinear system, in the domain of the state  $(\mathbf{x}_0, \mathbf{u}_0)$ , can be determined an approximation as

$$\dot{\mathbf{x}}(t) \approx \mathbf{A}_0 + \mathbf{A}\bar{\mathbf{x}}(t) + \mathbf{B}\bar{\mathbf{u}}(t), \quad (8)$$

where  $\mathbf{A}_0$ ,  $\mathbf{A}$ , and  $\mathbf{B}$  are  $(n \times 1)$ ,  $(n \times n)$ , and  $(n \times m)$  matrices, respectively.

Being  $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_0$  and  $\bar{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_0$ , the matrices of the linearized system are obtained as

$$\begin{aligned}\mathbf{A}_0 &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0), \\ \mathbf{A} &= \left( \begin{array}{ccc} \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_1} & \dots & \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_n} \end{array} \right) \Bigg|_{(\mathbf{x}_0, \mathbf{u}_0)}, \\ \mathbf{B} &= \left( \begin{array}{ccc} \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_1} & \dots & \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_1} & \dots & \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_m} \end{array} \right) \Bigg|_{(\mathbf{x}_0, \mathbf{u}_0)}.\end{aligned}\quad (9)$$

For the rest of the presented work, the time ( $t$ ) dependence of state variables and system inputs will be suppressed in order to abbreviate the expressions.

If  $(\mathbf{x}_0, \mathbf{u}_0)$  is an equilibrium state of the system, the matrix  $\mathbf{A}_0$  will be zero, since by definition, an equilibrium state leads the state equation to zero.

When the system (7) is represented by a NN model, the equivalent mathematical model is shown as (6). Linearizing the (7) around the state  $(\mathbf{x}_0, \mathbf{u}_0)$ , the new equivalent mathematical model of a linearized neural model is represented by (10), being  $x_q$  and  $u_v$  the  $q$ th and  $v$ th vector components of  $\mathbf{x}_0$  and  $\mathbf{u}_0$ , respectively. These components are also the set of inputs ( $n+m$ ) of the NN.

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left. \frac{\partial f_i}{\partial x_q} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \sum_{v=n+1}^{n+m} \left. \frac{\partial f_i}{\partial u_v} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \quad (10)$$

By extending the previous (10) with (6),

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[ \frac{\partial}{\partial x_q} \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right. \right. \\ & \left. \left. + W_{i,0} \right) \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q \\ & + \sum_{v=n+1}^{n+m} \left[ \frac{\partial}{\partial u_v} \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right. \right. \\ & \left. \left. + W_{i,0} \right) \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v, \end{aligned} \quad (11)$$

and consequently, (12) is obtained as

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[ \frac{\partial \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial x_q} \right] \Big|_{\bar{x}_q(\mathbf{x}_0, \mathbf{u}_0)} \\ & + \sum_{v=n+1}^{n+m} \left[ \frac{\partial \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial u_v} \right] \Big|_{\bar{u}_v(\mathbf{x}_0, \mathbf{u}_0)}, \end{aligned} \quad (12)$$

in a reduced form to work with

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left[ \frac{\partial G_h}{\partial x_q} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \sum_{v=n+1}^{n+m} \left[ \frac{\partial H_h}{\partial u_v} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \quad (13)$$

Subsequently, based on the works of Pirabakaran and Becerra [23] and Larrea [39], where these derivatives are calculated through the internal connections of the NN, (13) can be calculated separately for each of its terms. For this purpose, it is necessary to define the activation function of the hidden layer neurons,  $F_j(act_h)$ . In this work, for the neurons of the hidden layer of the selected multilayer perceptron structure, the activation function hyperbolic tangent is chosen. The selection of a smooth activation function is performed to enable the calculation of the partial derivatives

shown in (16). Applying the chain rule, we decompose the first part in three partial derivatives.

$$\frac{\partial G_h}{\partial x_q} = \frac{\partial G_h}{\partial F_h} \frac{\partial F_h}{\partial act_h} \frac{\partial act_h}{\partial x_q}. \quad (14)$$

The first partial derivative of (14) is straightforward.

$$\frac{\partial G_h}{\partial F_h} = W_{i,h}. \quad (15)$$

As the activation function  $F_h(\cdot)$  is  $\tanh(\cdot)$ , we obtain that the second partial derivative of (14) results in

$$\frac{\partial F_h(act_h)}{\partial act_h} = \frac{\partial}{\partial act_h} \frac{\sinh(act_h)}{\cosh(act_h)} = 1 - \tanh^2(act_h) = 1 - o_h^2, \quad (16)$$

where  $o_h$  is precisely the value of the  $h$ th neuron output for the MLP neural network,  $\tanh(act_h) = o_h$ , whilst the third partial derivative in (14) results in

$$\frac{\partial act_h}{\partial x_q} = \frac{\partial}{\partial x_q} \sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} = \omega_{h,q}. \quad (17)$$

Then, the overall solution for  $\partial G_h / \partial x_q$  is

$$\frac{\partial \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial x_q} = \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,q}, \quad (18)$$

being  $q$  an integer value into the interval  $[1, n]$ .

For the second derivative  $\partial H_h / \partial u_v$ , the procedure is similar to the previous one, obtaining the expression

$$\frac{\partial}{\partial u_v} \left( \sum_{h=1}^{n_h} W_{i,h} F_h \left[ \sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right) = \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,v}, \quad (19)$$

where  $v$  has integer values into the interval  $[n+1, n+m]$ .

Substituting (18) and (19) in (12), the equivalent mathematical model of a nonlinear system based on NN and linearization around a state  $(\mathbf{x}_0, \mathbf{u}_0)$  is depicted by

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[ \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,q} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q \\ & + \sum_{v=n+1}^{n+m} \left[ \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,v} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \end{aligned} \quad (20)$$

**3.2. Equilibrium States and Local Stability.** In order to perform an exhaustive analysis of the nonlinear system, it is first necessary to obtain an appropriate neural state model of that system, as presented in (1). This could be done through some of the existing modelling techniques [35], either online [40, 41] or offline [1, 6]. Subsequently, it is important to locate the different equilibrium states of the system. The search and location of such equilibrium states

in a control system are one of the first problems that have to be solved in order to develop a study of the behaviour of the system. Before designing the control system, the identification and analysis of the equilibrium states provide valuable information about the behaviour of the system, especially in the case of a nonlinear system, since these states are the most relevant cases to study such systems through linearization techniques. In order to locate the equilibrium states of the system, (21) must be solved.

$$\left[ \widehat{\mathbf{x}}_1(t) \cdots \widehat{\mathbf{x}}_n(t) \right]^T = [0 \cdots 0]^T. \quad (21)$$

For nonlinear dynamics, the equilibrium states could be very difficult to solve analytically, so it is necessary to use numerical methods [26–28, 30]. Given the mathematical model of a NN system characterized by a MLP, see (3), the set of nonlinear equations to solve is the following:

$$\begin{aligned} \sum_{h=1}^{n_h} W_{1,h} F_h(act_h) + W_{1,0} &= 0 \\ \sum_{h=1}^{n_h} W_{2,h} F_h(act_h) + W_{2,0} &= 0 \\ &\vdots \\ \sum_{h=1}^{n_h} W_{n,h} F_h(act_h) + W_{n,0} &= 0. \end{aligned} \quad (22)$$

These equations represent the  $n$  MLPs that model the nonlinear system under study. When analyzing the dynamics of nonlinear systems, the calculation of their equilibrium states can be a notable problem. In contrast to linear systems, where one or infinite equilibrium states exist, a nonlinear system can contain one, none, a finite number, or infinite states of equilibrium. For the resolution of the set of equations in (22), both numerical or more complex methods can be utilized. Complex methods, as bioinspired algorithms (i.e., evolutionary computation techniques), can locate a large number of solutions, but its slower convergence is a clear disadvantage in comparison with numerical methods. Then, in order to solve the set of nonlinear equations in (22), the use of numerical methods will be proposed, since they can offer a rapid convergence and precision in the obtained results [42, 43]. In this sense, the Levenberg-Marquardt (L-M) method [44] with the extension proposed by Moré [45] will be performed. This algorithm needs initial conditions to initialize the search. Thus, taking into account that it will be necessary to maximize the probability of finding every existing equilibrium state, a thin grid of points in the ranges of inputs and state variables will be used. Although this is the methodology that has been used in this article, the important thing in this case is to obtain as many solutions as possible, regardless of the method used to find them. Therefore, any other algorithm to solve (22) would be perfectly valid.

The L-M algorithm requires the Jacobian matrix of the system, in order to accelerate its convergence. This matrix can be obtained, either with explicit calculation or with some

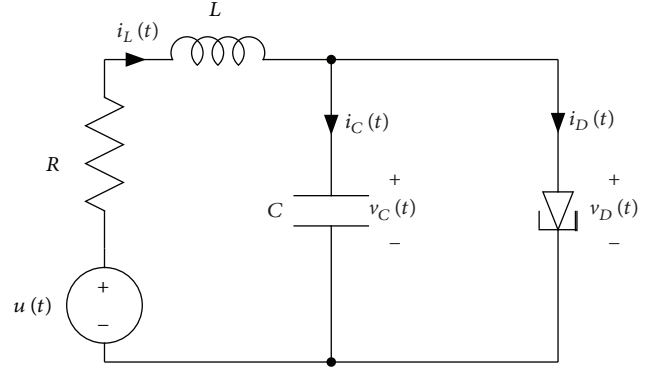


FIGURE 1: Tunnel-diode circuit.

technique to approximate it. In the previous Section 3.1, the calculation of the Jacobian matrix of a NN model has been solved under a general approach. Therefore, this matrix can be included into the numerical algorithm to enhance its precision and velocity of convergence.

Furthermore, the Jacobian matrix can be used, both for solving and finding the equilibrium states and for the linearization of the system in each of the solutions obtained. In this way, it is also possible to study the characteristics of the located equilibrium states, from the eigenvalues of the dynamic matrix of the linearized system. This analysis could improve the interpretation of system dynamics; it could help to study the local stability, even to observe more complex behaviours, such as bifurcations, saddle points, or limit cycles.

#### 4. Case Studies

In this section, three different examples are presented. These examples come from different areas, being nonlinear electrical, mechanical, and biological systems, which initially will be considered as unknown. The algorithms have been implemented with the tools of MATLAB R, both for the MLP neural network training and for the calculation of the NN model linearization.

**4.1. Equilibrium States of a Tunnel-Diode Circuit.** Let the tunnel-diode circuit shown in Figure 1, where  $R = 1, 5 \text{ k}\Omega$ ,  $C = 2 \text{ pF}$ , and  $L = 5 \mu\text{H}$ , with  $x_1(t) = v_C(t)$  and  $x_2(t) = i_L(t)$  as variables of the system, and  $h(v)$  the nonlinear relation between both. This is a case of study broadly used [20, 21], with the state model expressed by (23).

$$\begin{aligned} \dot{x}_1 &= 0.5[-h(x_1) + x_2], \\ \dot{x}_2 &= 0.2(-x_1 - 1.5x_2 + 1.2). \end{aligned} \quad (23)$$

Supposing that  $h(v)$  has a nonlinear relation characterized by (24) and shown in Figure 2,

$$\begin{aligned} h(x_1) &= 17.76x_1 - 103.79x_1^2 + 229.62x_1^3 \\ &\quad - 226.31x_1^4 + 83.72x_1^5. \end{aligned} \quad (24)$$

Considering that the system dynamics is unknown, the first stage in the work plan will be to obtain a NN model of

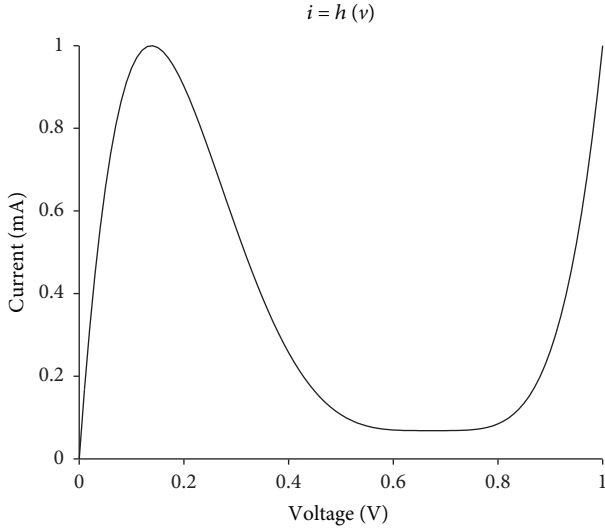


FIGURE 2: Phase portrait of the tunnel-diode circuit.

the system from input-output data. These data will come from an exhaustive selection of examples that will cover the whole universe of the discourse of operating points of the system. Subsequently, after analyzing the NN model and its Jacobian matrix, relevant conclusions will be drawn regarding the dynamics of the system.

In the case of the tunnel-diode, a set of 2000 points is created to model the behaviour of the nonlinear relation  $v_D - i_D$  which is uniformly distributed along their discourse universes, respectively.

$$\begin{aligned} X &\subseteq \{x|x_1(t) \in [0, 1], x_2(t) \in [0, 2]\}, \\ U &\subseteq \{u|u(t) \in [0, 2]\}, \end{aligned} \quad (25)$$

where 1600 data samples (80%) will be used in the modelling phase and 400 data samples (20%) in the validation phase. It must be noted that the mathematical model of the tunnel-diode has only been utilized to obtain the dataset, meaning that for the rest of the process it is considered unknown.

In order to keep an appropriate methodological procedure that provides the most adequate MLP neural network, a batch of training experiments has been prepared where the number of neurons in the hidden layer changes, being this the parameter to be studied in relation to the precise response of the NN output. All training experiments have been initialized with an extension of an advanced algorithm to assure the initial local stability of the NN [46]. The final structure selected has been the simplest one obtained that allows to estimate both outputs with enough precision approaching the real ones. This is a MLP with three inputs,  $x_1$ ,  $x_2$ , and  $u$ , a hidden layer of three neurons with hyperbolic tangent activation function, and two outputs,  $x_1$  and  $x_2$ . The training process was performed with a second order algorithm based on the L-M approach [44, 47], obtaining a validation MSE of 0.1147 V/ns and 3.03 mA/ns, respectively.

To solve the set of equations shown in (22), a minimization algorithm based on the same approach of L-M, adapted by Moré, has been used [44, 45]. With this purpose, a dense mesh has been created to initialize the calculation process whose limits are directly related to the universe of discourse of the studied variables, applying increments of 0.05 and 0.1. In Figure 3, solutions obtained by the numerical algorithm in the NN model are compared with the real system, neglecting the nonvalid solutions located outside the universe of discourse.

Comprehensively using the linearization proposed in this work, it has been possible to find linearized models in each of the equilibrium states shown in Figure 3. These have been characterized in relation to their behaviour, based on the eigenvalues of the dynamic matrix of the linearized system. As can be seen in Figure 3, the equilibrium states of the real system have been calculated accurately, where each type has correlation with the corresponding one obtained by the NN model. It must be noted that the numerical algorithm has found a set of equilibrium states in the NN model which do not exist in the real system. This error may be due to the intrinsic modeling error or to the minimization process with the L-M algorithm. When using numerical methods, tolerances have to be assumed to find the solutions. These tolerances, in addition to the minimization algorithm error, are considered as actual solutions. However, as they are not actual solutions, the study of the eigenvalues in these points reflects an erratic behaviour, since the types of detected eigenvalues are not coherent (at very close points, several types of eigenvalues are detected). This phenomenon does not happen in actual equilibrium states, so it is a simple way to discard the false states detected.

4.2. *The Case of an Inverted Pendulum with Friction.* The second example is an inverted pendulum with friction presented in Figure 4. The dynamics of this system is represented by (26).

$$ML^2\ddot{\theta}(t) + b\dot{\theta}(t) - MgL \sin(\theta(t)) = \tau(t), \quad (26)$$

where the coefficient of friction is  $b=0.5$  N/m/s,  $\theta$  is the pendulum angle with respect to the vertical,  $\tau$  is the applied pair in N/m,  $M=1$  Kg is the mass of the pendulum, the length of rod is  $L=0.5$  m, and the acceleration of gravity is  $g=9.81$  m/s<sup>2</sup>.

Let be the state vector given by

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \theta(t) \\ \dot{\theta}(t) \end{pmatrix}. \quad (27)$$

Then, the system of inverted pendulum can be defined by the state model as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= \frac{g}{L} \sin(x_1(t)) - \frac{b}{ML^2}x_2(t) + \frac{1}{ML^2}u(t). \end{aligned} \quad (28)$$

As a starting point, we consider that the system dynamics is unknown with the aim of validating the techniques

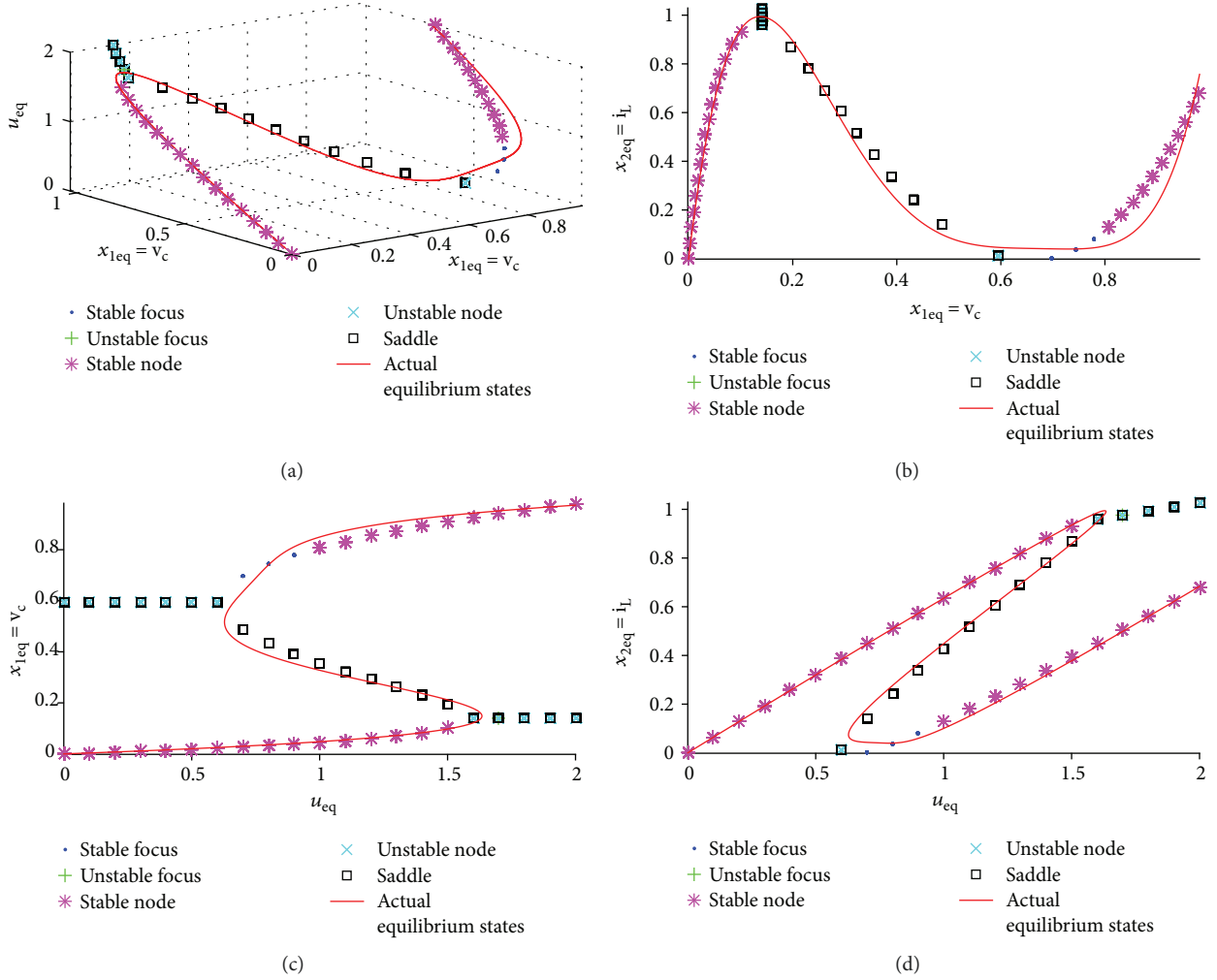


FIGURE 3: Equilibrium states of the tunnel-diode NN model.

developed in this paper. Moreover, we assume that it is possible to collect enough input-output data to obtain the neural model of the system. Thus, the first stage of this work will be to extract a MLP neural model of the pendulum system to be studied.

According to the previous example, a dataset has been prepared considering the universes of discourse of the state variables and input signal, respectively,

$$\begin{aligned} X &\subseteq \{ \mathbf{x} \mid x_1(t) \in [-\pi, \pi], x_2(t) \in [-10, 10] \}, \\ U &\subseteq \{ \tau \mid \tau(t) \in [-10^3, 10^3] \}. \end{aligned} \quad (29)$$

From these discourse universes, 2000 input-output data samples have been extracted, proportionally distributed, taking 1600 data samples (80%) for neural network training and 400 data samples (20%) for the validation. As was the case of the above study, the pendulum system will be considered as unknown.

Also as the previous case, the MLP neural network has been trained using the L-M algorithm [44, 47]. The most appropriate NN structure obtained has three inputs,  $x_1$ ,  $x_2$ , and  $\tau$ , a hidden layer of three neurons with hyperbolic tangent as activation function, and two outputs,  $\dot{x}_1$  and  $\dot{x}_2$ .

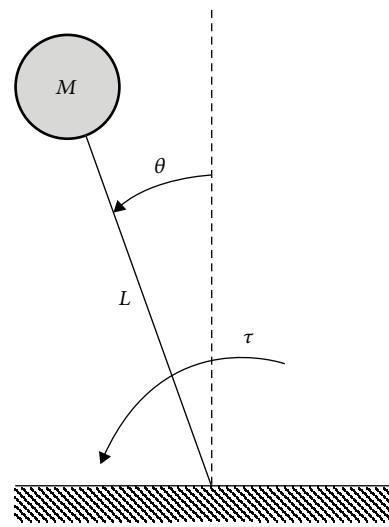


FIGURE 4: Inverted pendulum with friction.

The training process has been performed with the same algorithm, Levenberg-Marquardt, and the validation MSE errors have been  $6.334 \times 10^{-6}$  rad/s and 0.1857 rad.

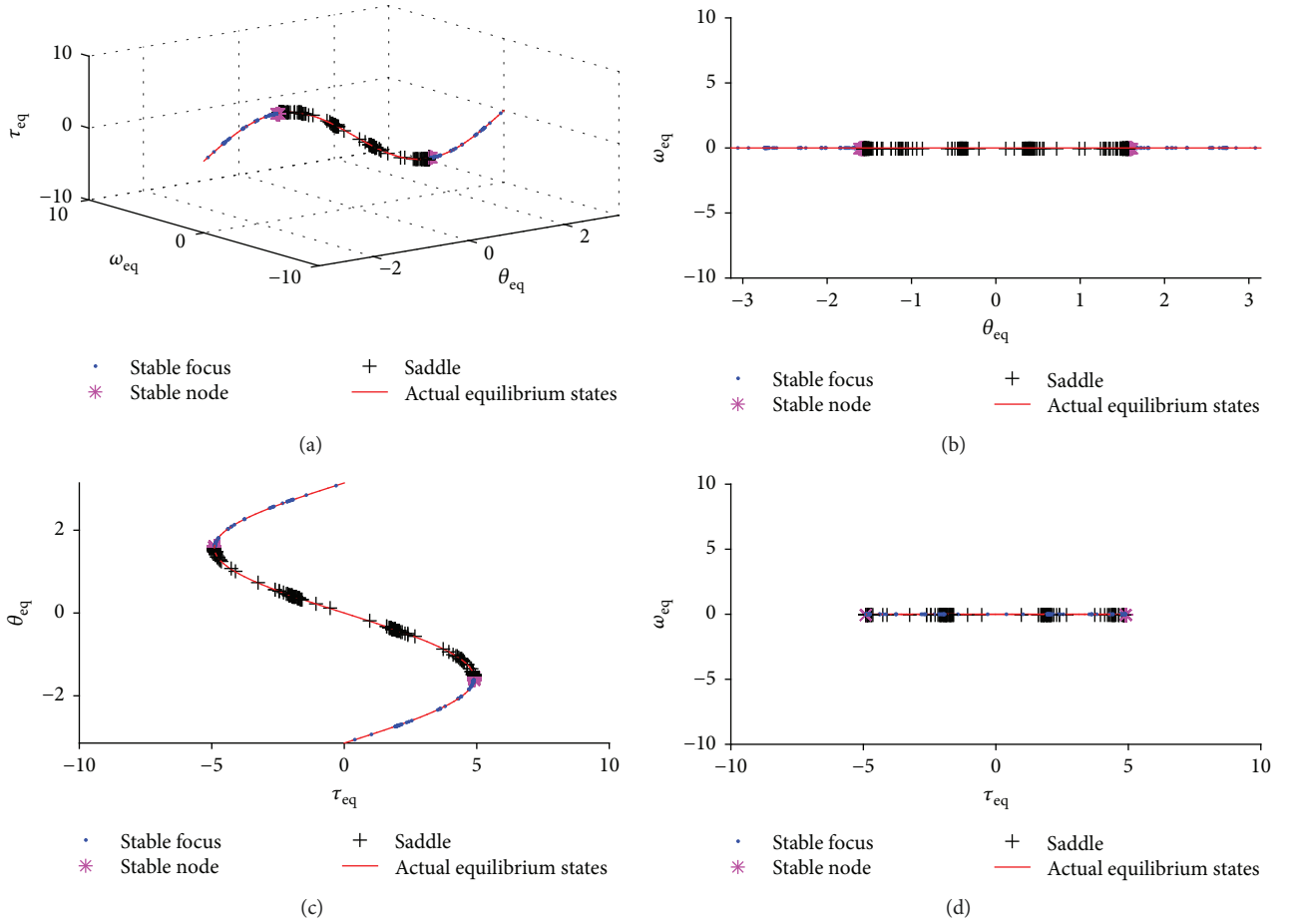


FIGURE 5: Equilibrium states of the inverted pendulum with friction.

Taking into account the obtained NN model of the system, the equilibrium states have been calculated as presented in Section 3.2 and explained at the above example. Thus, a grid of initial points has been created again, limiting each variable range into their universes of discourse, with increments of 0.1, 5, and 4, for the states and the input, respectively. All the equilibrium states located with the NN model are shown in Figure 5. As in the previous example, the points located outside of the universe of discourse have been discarded and removed.

For the real system, its equilibrium states are located on the line  $\tau = -MLg \sin(\theta)$ , being saddle points the equilibrium points on  $-\pi/2 < \theta < \pi/2$ , and stable focuses on the rest sited on  $[-\pi, \pi]$  [48]. Compared with the equilibrium points estimated by the NN model, it can be concluded that those points have been correctly obtained. Certain discrepancies can be noticed in the steady states obtained around  $|\tau| \approx 5$  and  $|\theta| \approx \pi/2$ , but the rest of equilibrium states estimated into the universe of discourse have been accurately located. The appearance of errors is inevitable as the NN model has an implicit error associated. Considering that such errors are within permissible limits, it is necessary to emphasize that the extracted information, with this neural approach, through the analysis of the

neural structure is a powerful technique to obtain information from an unknown system.

4.3. *A Prey-Predator Biological System.* Nature is capable of providing diverse real-life examples in which animal behaviour presents multiple stable equilibrium states. This is the case of a predator-prey system with adaptation of prey behaviour to changing environmental conditions [49, 50]. The classic Lotka-Volterra predator-prey model is given by

$$\begin{aligned} \dot{p}(t) &= rp(t)(1 - p(t)/K) - ap(t)q(t), \\ \dot{q}(t) &= ep(t)q(t) - mq(t), \end{aligned} \quad (30)$$

where  $p(t)$  and  $q(t)$  are sizes of the prey and the predator populations,  $r$  is per capita reproduction rate of the prey,  $K$  is the carrying capacity of environment,  $m$  is the per capita mortality rate of the predator in the prey deficiency,  $a$  is the attack rate, and  $e = ka$  shows the consumption efficiency after the attack. Let be the state vector given by

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} p(t) \\ q(t) \end{pmatrix}. \quad (31)$$



Then, the system of a biological prey-predator can be defined by the state model.

$$\begin{aligned}\dot{x}_1(t) &= rx_1(t)(1 - x_1(t)/K) - ax_1(t)x_2(t), \\ \dot{x}_2(t) &= ex_1(t)x_2(t) - mx_2(t).\end{aligned}\quad (32)$$

In response to an external challenge, each individual into the prey community may modify its behaviour between risky and safe conduct. On the basis of the vulnerability of individuals in risky mode, the first expression in (32), suppressing the time ( $t$ ) dependence, will be extended as

$$\begin{aligned}\dot{x}_{1R} &= r_R x_{1R} - c_R x_{1R}^2 - h_R x_{1R} x_{1S} - a_R x_{1R} x_2, \\ \dot{x}_{1S} &= r_S x_{1S} - c_S x_{1S}^2 - h_S x_{1R} x_{1S} - a_S x_{1S} x_2.\end{aligned}\quad (33)$$

In (33),  $x_{1R}$  and  $x_{1S}$  are the subpopulations of  $x_1$  in risky (R) and safe (S) modes. In addition,  $c_i = r_i/K_i$ ,  $h_R$ , and  $h_S$  are magnitudes of the intermode competition [50].

Taking into account that  $x_1 = x_{1R} + x_{1S}$ , and defining  $D_S$  the nonlinear function of the probability for a prey to be hunted by a predator as shown in Figure 6, with  $D_R + D_S = 1$  [50], the state model of the behavioural predator-prey system will be

$$\begin{aligned}\dot{x}_1 &= (r_S D_S + r_R(1 - D_S))x_1 - (a_S D_S + a_R(1 - D_S))x_1 x_2 \\ &\quad - (c_S D_S^2 + c_R(1 - D_S)^2 + (h_R + h_S)D_S(1 - D_S))x_1^2, \\ \dot{x}_2 &= (e_S D_S + e_R(1 - D_S))x_1 x_2 - mx_2.\end{aligned}\quad (34)$$

Moreover, to reject biologically unreachable equilibrium states, it is necessary to preserve the condition.

$$h_S + h_R - 2c_R \geq 0. \quad (35)$$

As Pimenov et al. argue [50], this condition implies that prey in both modes, risky and safe, is taking advance of the same environment and using the same resources. This basically differs from the concepts of multiple patches or refuge, when the prey in these habitats is assumed to have no access to resources in other patches, and hence all  $h_{ij} = 0$ .

Considering again that the system dynamics are unknown, an extensive set of input-output data have been obtained from the state model in (34). Subsequently, a batch of different structures of MLP neural networks has been trained in order to extract an appropriate NN that accurately reproduces the behaviour of the predator-prey system. This dataset contains 10,000 points uniformly extracted from the universes of discourse of the state variables,  $x_1 = p(t)$  and  $x_2 = q(t)$ , and input,  $u(t) = K_R(t)$ , respectively.

$$\begin{aligned}X &\subseteq \{\mathbf{x} \mid x_1(t) \in [0, 30], x_2(t) \in [0, 70]\}, \\ U &\subseteq \{u \mid u(t) \in [5, 100]\},\end{aligned}\quad (36)$$

where 8000 data (80%) have been used for training MLP neural networks, proceeding in previous examples, and 2000 data (20%) have been managed in the validation stage to select an appropriate neural model. After training and validating, for each of the state variables, a MLP structure

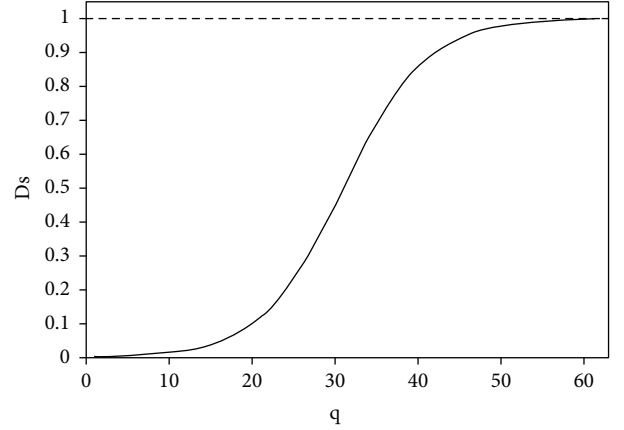


FIGURE 6: Functional response: probability of hunted prey  $D_S(q)$  versus size of predator population  $q$ .

with a different dimension has been selected, incorporating 15 neurons in the hidden layer for the estimation of  $\dot{x}_1$  and 6 neurons in the case of  $\dot{x}_2$ . The training process has been performed with the same algorithm as above, the second order L-M algorithm [44, 47], obtaining a validation MSE result of 17.797 and 0.1059 individuals, respectively.

Afterwards, the set of equations in (23) has been solved with the same approach as before, the L-M algorithm adapted by Moré [44, 45], through a dense mesh of points created into the universes of discourse of the studied variables,  $p(t)$ ,  $q(t)$ , and  $K_R(t)$ , applying increments of 1 individual. Figure 7 shows the comparison between the solutions obtained through the NN model and the real system.

After the linearization developed in this work, different equilibrium states have been found in the real system, as shown in Figure 7. These states are characterized by their eigenvalues of the corresponding dynamic matrix. In the same way, it can be seen how the equilibrium states calculated by the NN model have correlation with the states existing in the real system. All these results are directly related to the analysis of the real system that Pimenov shows at work [50], and they maintain a precise correspondence with them. As in previous examples, those points calculated outside of the universe of discourse can also be discarded.

## 5. Conclusions

This work has presented a straightforward methodological procedure based on neural network models, to analyze the dynamical behaviour of unknown nonlinear systems. This implies presenting the linearization of a neural model to be compared with the proper linearization of the real system, and finally explaining the calculation of the equilibrium states of the real system through the NN model and its relation with the study of the local stability.

This work initiates a research line where NNs are the basis of a nonlinear system model, allowing to calculate its corresponding Jacobian matrix to analyze the local stability of the equilibrium states. Since it is assumed that there is no approximate mathematical model of the nonlinear

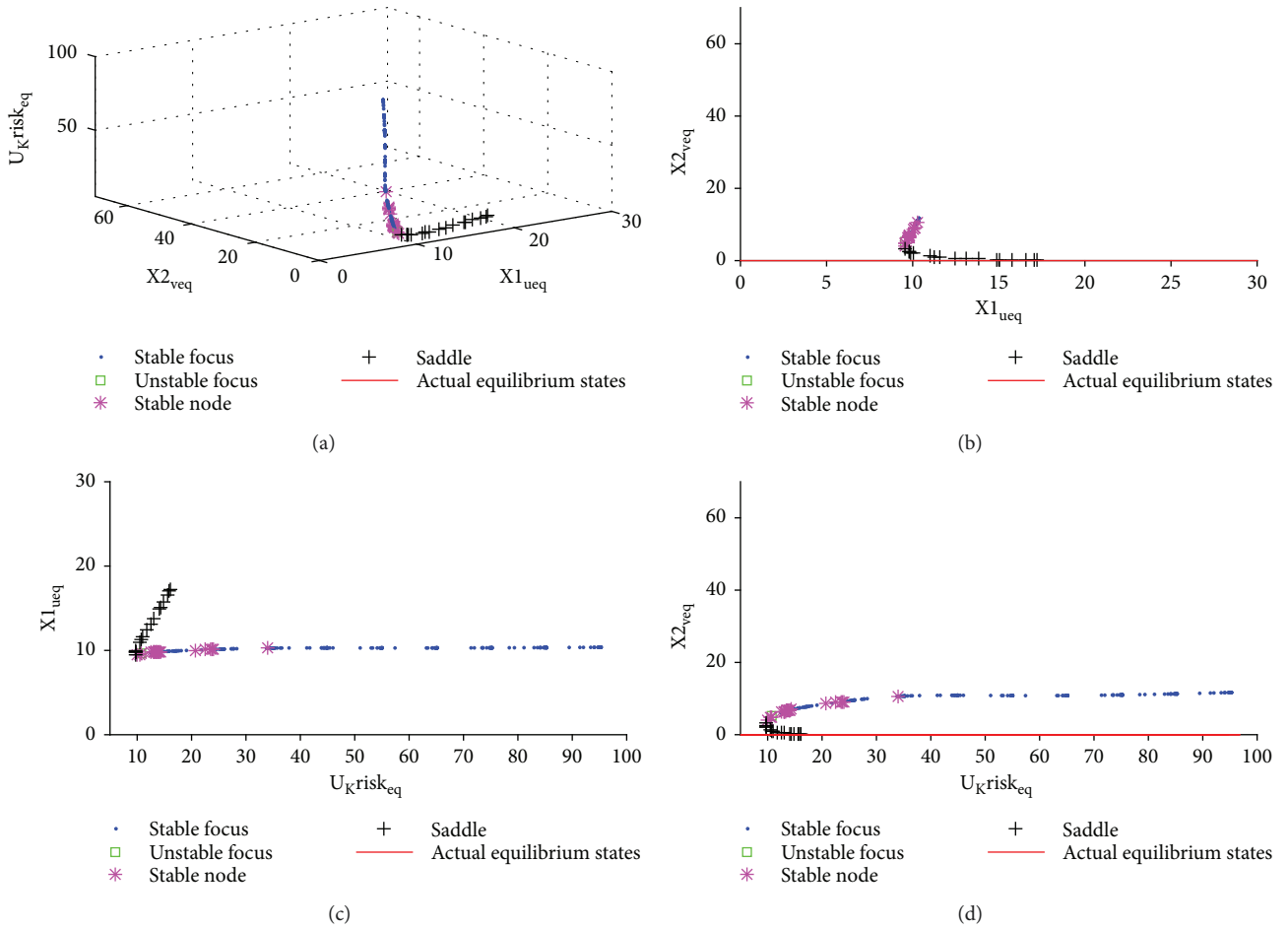


FIGURE 7: Equilibrium states of the predator-prey NN model.

system, the accuracy of the study of equilibrium states is related to the NN training process. In this process, many experiments have been developed to ensure the representativeness of the entire operating range that will be considered in the analysis of the nonlinear system. This has involved keeping the system under control and acquiring the appropriate amount of examples of variables and states.

The proposed method has been validated with three nonlinear systems coming from different fields (electrical, mechanical, and biological). This approach has been developed with multilayer perceptron neural network structures, one of the most simple and feasible in training and application. The training processes have been based on widely used Levenberg-Marquardt algorithm and developed with exhaustive input-output data of the above systems, guaranteeing the representativeness of each universe of discourse. Taking into account that after the training the errors have been within permissible limits, it is necessary to emphasize that the extracted information, with this neural approach, through the analysis of the neural structure is a powerful method to obtain information from an unknown system.

The results have shown that it is possible to obtain dynamical information of nonlinear systems uniquely using the corresponding NN model. This information is directly

related to the equilibrium states of the real system, enabling to study the local stability around each equilibrium state.

Due to the interest of this research line, more additional works could be developed in the future considering other NN structures different to that presented in this paper, the multilayer perceptron with sigmoidal activation function in its unique hidden layer. The study of nonlinear systems of more complicated dynamic features (including systems with memory and with noncontinuous operating ranges), together with the modification of the NN structures (recurrent, radial basis function, deep neural networks, etc.), could establish a promising and extensive field of study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank the Ministry of Economy, Industry and Competitiveness of Spain that has funded this work under the project DPI2017-85540-R (H2SMART-GRID).

## References

- [1] A. Zubizarreta, M. Larrea, E. Irigoyen, I. Cabanes, and E. Portillo, "Real time direct kinematic problem computation of the 3prs robot using neural networks," *Neurocomputing*, vol. 271, Supplement C, pp. 104–114, 2018.
- [2] J. M. Andújar and J. M. Bravo, "Multivariable fuzzy control applied to the physical–chemical treatment facility of a cellulose factory," *Fuzzy Sets and Systems*, vol. 150, no. 3, pp. 475–492, 2005.
- [3] M. E. Ortiz-Quisbert, M. A. Duarte-Mermoud, F. Milla, and R. Castro-Linares, "Control Adaptativo Fraccionario Optimizado por Algoritmos Genéticos, Aplicado a Reguladores Automáticos de Voltaje," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 13, no. 4, pp. 403–409, 2016.
- [4] E. Irigoyen, M. Larrea, J. Valera, V. Gómez, and F. Artaza, "A hybridized neurogenetic solution for controlling industrial r3 workspace," *Neural Network World*, vol. 20, no. 7, pp. 811–824, 2010.
- [5] J. G. G. Fontanet, A. L. Cervantes, and I. B. Ortiz, "Alternativas de control para un Péndulo de Furuta," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 13, no. 4, pp. 410–420, 2016.
- [6] E. Irigoyen and G. Miñano, "A narx neural network model for enhancing cardiovascular rehabilitation therapies," *Neurocomputing*, vol. 109, pp. 9–15, 2013.
- [7] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, 2000.
- [8] W. J. Rugh, *Linear System Theory*, Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1996.
- [9] T. Jing and F. Chen, "Finite-time lag synchronization of delayed neural networks via periodically intermittent control," *Complexity*, vol. 21, no. S1, pp. 211–219, 2016.
- [10] S. Lei and H. Shen, "Fault-tolerant mixed  $\mathcal{H}_\infty$ /passive synchronization for delayed chaotic neural networks with sampled-data control," *Complexity*, vol. 21, no. 6, pp. 246–259, 2016.
- [11] J. A. Grande, J. M. Andújar, J. Aroba et al., "Fuzzy modeling of the spatial evolution of the chemistry in the tinto river (SW Spain)," *Water Resources Management*, vol. 24, no. 12, pp. 3219–3235, 2010.
- [12] M. Larrea, E. Irigoyen, V. Gómez, and F. Artaza, "Nonlinear system control based on neural networks with adaptive predictive strategy," in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, pp. 1–7, Bilbao, Spain, September 2010.
- [13] M. J. López-Baldán, A. García-Cerezo, J. M. C. López, and A. R. Gallego, "Fuzzy modeling of a thermal solar plant," *International Journal of Intelligent Systems*, vol. 17, no. 4, pp. 369–379, 2002.
- [14] S. Chen and S. A. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *International Journal of Control*, vol. 56, no. 2, pp. 319–346, 1992.
- [15] J. J. V. García, V. G. Garay, E. I. Gordo, F. A. Fano, and M. L. Sukia, "Intelligent multi-objective nonlinear model predictive control (imo-nmpc): towards the 'on-line' optimization of highly complex control problems," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6527–6540, 2012.
- [16] H. J. Marquez, *Nonlinear Control Systems. Analysis and Design*, John Wiley & Sons, Inc., 2003.
- [17] J. M. Andújar, J. M. Bravo, and A. Peregrín, "Stability analysis and synthesis of multivariable fuzzy systems using interval arithmetic," *Fuzzy Sets and Systems*, vol. 148, no. 3, pp. 337–353, 2004.
- [18] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos. Texts in Applied Mathematics*, Springer, 2nd edition, 2003.
- [19] S. Sastry, *Nonlinear System: Analysis, Stability, and Control*, Springer, New York, NY, USA, 1999.
- [20] A. J. Barragán, B. M. Al-Hadithi, A. Jiménez, and J. M. Andújar, "A general methodology for online TS fuzzy modeling by the extended kalman filter," *Applied Soft Computing*, vol. 18, pp. 277–289, 2014.
- [21] A. J. Barragán, B. M. Al-Hadithi, J. M. Andújar, and A. Jiménez, "Metodología formal de análisis del comportamiento dinámico de sistemas no lineales mediante lógica borrosa," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 12, no. 4, pp. 434–445, 2015.
- [22] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 497–509, 2016.
- [23] K. Pirabakaran and V. M. Becerra, "Pid autotuning using neural networks and model reference adaptive control," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 451–456, 2002.
- [24] N. Kishor, R. P. Saini, and S. P. Singh, "Small hydro power plant identification using nnarx structure," *Neural Computing and Applications*, vol. 14, no. 3, pp. 212–222, 2005.
- [25] R. K. Al Seyab and Y. Cao, "Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation," *Journal of Process Control*, vol. 18, no. 6, pp. 568–581, 2008.
- [26] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [27] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [28] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.
- [29] V. Z. Marmarelis and X. Zhao, "Volterra models and three-layer perceptrons," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1421–1433, 1997.
- [30] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990.
- [31] A. R. Gallant and H. White, "On learning the derivatives of an unknown mapping with multilayer feedforward networks," *Neural Networks*, vol. 5, no. 1, pp. 129–138, 1992.
- [32] J. M. A. Marquez, A. J. B. Pina, and M. E. G. Arias, "A general and formal methodology to design stable nonlinear fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1081–1091, 2009.
- [33] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [34] S. Jagannathan, F. L. Lewis, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, 1999.

- [35] N. K. Poulsen, M. Nørgaard, O. Ravn, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*, Springer, 2000.
- [36] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*, Francis & Taylor, 2006.
- [37] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer Verlag, Berlin, Germany, 1990.
- [38] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1991.
- [39] M. Larrea, *Estrategia Adaptativa-Predictiva Basada en Redes Neuronales para el Control de Sistemas No Lineales*, [Ph.D. thesis], UPV/EHU, 2014, Editor: CreateSpace Independent Publishing Platform.
- [40] Q. Song, Y. Wu, and Y. C. Soh, "Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1841–1853, 2008.
- [41] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor uav using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.
- [42] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local NLP solvers: a multistart framework for global optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [43] F. Glover, "A template for scatter search and path relinking," in *Artificial Evolution. AE 1997*, vol. 1363 of Lecture Notes in Computer Science, pp. 1–51, Springer, Berlin Heidelberg, 1998.
- [44] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [45] J. J. Moré G. A. Watson, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical Analysis*, pp. 105–116, Springer, Berlin, Heidelberg, 1978.
- [46] E. Irigoyen and M. Pinzolas, "Numerical bounds to assure initial local stability of narx multilayer perceptrons and radial basis functions," *Neurocomputing*, vol. 72, no. 1–3, pp. 539–547, 2008.
- [47] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [48] E. I. Butikov, "On the dynamic stabilization of an inverted pendulum," *American Journal of Physics*, vol. 69, no. 7, pp. 755–768, 2001.
- [49] A. Korobeinikov, "Stability of ecosystem: global properties of a general predator-prey model," *Mathematical Medicine and Biology*, vol. 26, no. 4, pp. 309–321, 2009.
- [50] A. Pimenov, T. C. Kelly, A. Korobeinikov, M. J. A. O'Callaghan, and D. Rachinskii, "Adaptive behaviour and multiple equilibrium states in a predator-prey model," *Theoretical Population Biology*, vol. 101, pp. 24–30, 2015.

