

Design of a Neural Network Based Stable State Observer for MIMO Systems

Jean Gutenbert Kenfack Wamba^{1,2}, Eric Michel Deussom Djomadji^{1,2,3},
Jean Claude Lionel Ng'anyogo², Arsene Roger Bienvenu Fouba², Alain Tiedeu¹

¹Department of Electrical and Telecommunications Engineering, National Advanced School of Engineering of Yaoundé, University of Yaoundé I, Yaoundé, Cameroon

²Division of ICT, National Advanced School of Post, Telecommunication and ICT, University of Yaounde I, Yaoundé, Cameroon

³Department of Electrical and Electronic Engineering, College of Technology, University of Buea, Buea, Cameroon

Email: eric.deussom@gmail.com

How to cite this paper: Wamba, J.G.K., Djomadji, E.M.D., Ng'anyogo, J.C.L., Fouba, A.R.B. and Tiedeu, A. (2023) Design of a Neural Network Based Stable State Observer for MIMO Systems. *Journal of Computer and Communications*, 11, 87-110. <https://doi.org/10.4236/jcc.2023.1111006>

Received: October 6, 2023

Accepted: November 26, 2023

Published: November 29, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

MIMO (Multiple Input Multiple Output) is a key technology underpinning fourth generation or 4G networks. This technology allows 4G networks to increase throughput. However, the dynamics of the MIMO system are not under control due to the many uncertainties that destabilize the system. This work is therefore very relevant in the sense that an observer can be used to monitor the dynamics of such a system. This work presents a neuro-adaptive observer based on a radial basis function neural network for generic non-linear MIMO systems. Unlike most neuro-adaptive observers, the proposed observer uses a neural network that is non-linear in its parameters. It can therefore be applied to systems with high degrees of nonlinearity without any a priori knowledge of the system dynamics. Indeed, in addition to the fact that neural networks are very good nonlinear approximators, their adaptive behavior makes them powerful tools for observing the state without any a priori knowledge of the dynamics of the system. The learning rule of the neural network is an approach based on the modified backpropagation algorithm: A term has been added to guarantee the robustness of the observer. The proposed approach is not limited by a strong assumption. The stability of the neuro-adaptive observer is demonstrated by the direct Lyapunov method. Simulation results are presented in the context of MIMO signal transmission applied in LTE, to demonstrate the performance of our observer.

Keywords

Stability, Neural Network, MIMO, LTE Network, Lyapunov Function

1. Introduction

An observer is a “computer” measurement tool that allows all the states of an industrial system to be retrieved with the minimum of information about these states. This minimum information is very often obtained with the help of a sensor. However, exclusive use of sensors is not always possible for the following reasons:

- The prohibitive cost of the physical sensor(s);
- The sensor that does not fit the dynamics of the system (sensor too slow);
- The non-existence of a sensor for the quantity to be measured [1].

An observer is therefore responsible for estimating the state of a system while optimising the number of sensors in an industrial application: hence its economic interest.

Furthermore, the state of a process specifies its behaviour and many control schemes rely on the availability of states. However, in many practical systems, only the input and output of a system are measurable. Therefore, estimation of the system state plays a crucial role in monitoring the process, detecting and diagnosing faults and achieving better performance [2]. The presence of several inputs and outputs in a MIMO system causes a difficulty in modelling the dynamics and thus becomes a potential source of uncertainty that can degrade the performance of a MIMO system and even in some cases destabilise the system. Specifically in the case of the 4G LTE network at CAMTEL, no tool or platform is used to monitor the dynamics of the MIMO system: once the system is configured, it is “left” to its own devices and the maintenance teams’ only return when users report problems. The design of our observer is therefore very important, as it will not only serve as a software sensor to monitor the dynamics of the MIMO system but also to estimate the states of the MIMO system in the transmission of signals.

Problem

MIMO is the flagship technology used in the 4G network to increase throughput and spectral efficiency. However, when transmitting MIMO signals, many uncertainties arise, such as multipath propagation, fading and others. All these problems affect the performance of MIMO and have a direct impact on the end user. In short, the dynamics of the MIMO system at CAMTEL are not under control. In order to overcome this problem, we propose a neuro-adaptive observer that will be able to evaluate and monitor the dynamics of the MIMO system. The main question that is highlighted here is how to efficiently and reliably estimate the states of a MIMO system in signal transmission?

Objectives

The main objective of this work is to design a neural network based steady state observer for MIMO systems. To achieve this, we will specifically:

- Model and implement the proposed state observer with Simulink and MATLAB for state estimates of a MIMO system;
- Model the MIMO system through this observer;

- Evaluate the stability of our observer by the direct Lyapunov method.

2. Key Concepts and Literature Review

2.1. Key Concepts

1) Dynamic system

From the point of view of mathematics, a dynamical system presents a mathematical concept by which a guiding rule is established to define how a point propagates in time in a geometric space [3]. In other words, a dynamic system is a mechanical, physical, economic, environmental or any other domain whose state evolves with time [4].

There are two general components of a dynamic system, namely:

- States: these are referred to as the essential information required to define the output of the system at a given time or simply a set of quantities sufficient to qualify the system;
- Dynamics: this is referred to as a set of rules that define the evolution of the system's states over time.

Figure 1 illustrates the classification of dynamic systems.

In this work, we focus on nonlinear continuous-time dynamical systems because MIMO systems are similar to them. We call a continuous-time dynamical system on a set Ω a family of applications $\{\varphi_t; t \in \mathbb{R}^+\}$, where $\{\varphi_t; t \in \mathbb{R}\}$, parameterized either by the set \mathbb{R}^+ of positive or zero real numbers, or by the set of all real numbers \mathbb{R} , verifying the following properties:

- Each application φ_t is defined on a part U_t of Ω , and with values in Ω .
- The application φ_0 , defined on an integer Ω , is id_Ω .
- if $0 \leq t_1 \leq t_2$, then $U_{t_2} \subset U_{t_1}$.
- Are t and s be 02 elements of the set $(\mathbb{R}^+ \text{ or } \mathbb{R})$ which parameterises the family of applications under consideration. Let $x \in U_s$. Then $\varphi_s(x)$ is an element of U_t if and only if x is an element of U_{s+t} and, when this is the case, $\varphi_t(\varphi_s(x)) = \varphi_{s+t}(x)$ [5].

The set Ω is called the phase space of the dynamical system.

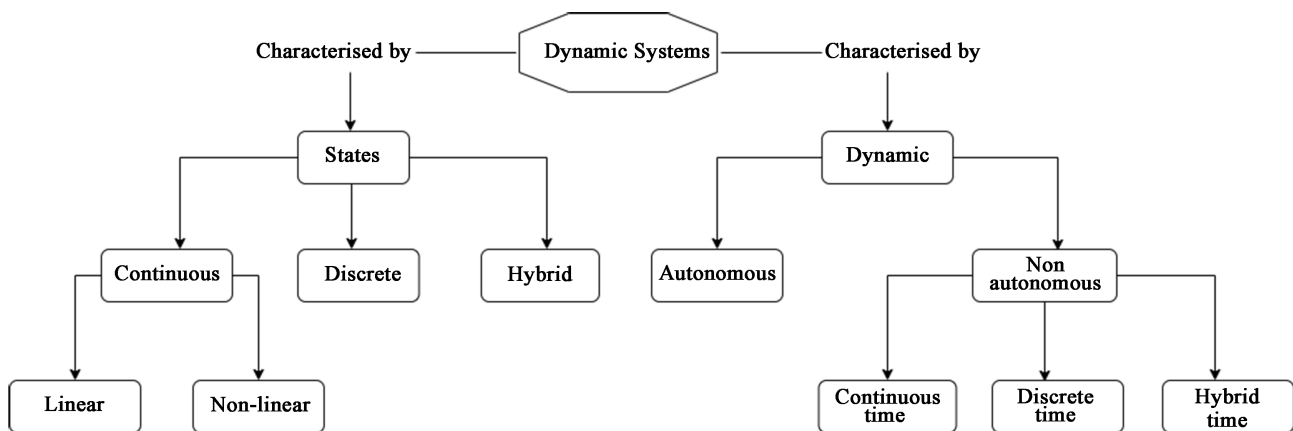


Figure 1. Classification of dynamic systems.

2) Principle of state estimation

An observer or state reconstructor is a software sensor that allows the reconstruction of the internal state variables of a system from the inputs and outputs of the real system. In other words, an observer is a “computer” measurement tool that allows the retrieval of all the states of an industrial system with the minimum of information about these states. The block diagram of a state observer is presented as follows in **Figure 2**:

For a system presented by the following system of equations:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \quad (1)$$

A state observer is shown in **Figure 3**.

This structure first reveals the presence of a state estimator operating in an open loop characterised by the same dynamics as the system. The dynamics desired in closed loop by this observer is obtained by introducing a vector (or matrix in the multivariable case) of gains L.

3) Lyapunov stability of dynamic systems

In general, Lyapunov stability theory can deal with an unforced system *i.e.* where $x \in \mathbb{R}^n$, $t \in \mathbb{R}^+$, and if $\dot{x}(t) = f(x(t), t)$ where $x \in \mathbb{R}^n$, $t \in \mathbb{R}^+$, and $f : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$.

If $f(x_e, t) = 0$, then x_e is an equilibrium point. Furthermore, it is assumed

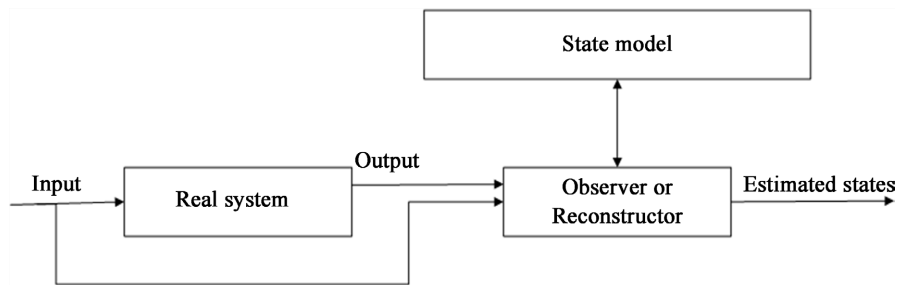


Figure 2. Block diagram of a state observer.

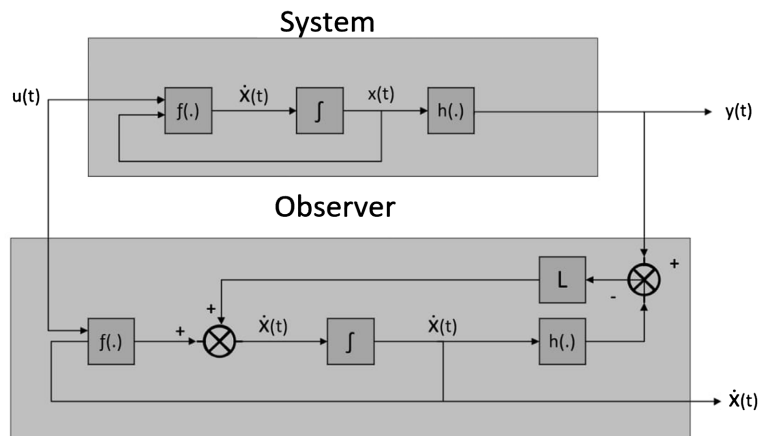


Figure 3. Principle of estimation of a state observer.

that $f(x(t), t)$ can satisfy standard conditions on the existence and uniqueness of the solution. For example, such a condition could state that $f(x(t), t)$ is continuous in the Lipschitz sense with respect to x , or it is uniformly and piecewise continuous at t .

There are two general methods consisting of the analysis of the stability of the equilibrium, namely the direct Lyapunov method or second Lyapunov method and the indirect Lyapunov method or first method. On the one hand, the direct Lyapunov method examines the existence of Lyapunov functions, *i.e.* scalar auxiliary functions of the state space, to determine the stability properties of the equilibrium. On the other hand, the indirect Lyapunov method explores the stability properties from the linearisation of a dynamical system described by non-linear differential equations in order to deduce the stability properties of the equilibrium [3].

- **The direct Lyapunov method for stability**

The direct Lyapunov method is the most essential approach for the design and analysis of linear and non-linear dynamic systems. It can be used directly for a non-linear system without linearisation to analyse the overall stability. The fundamental concept of the direct Lyapunov method is based on the notion that if the total energy of a system is continuously dissipating, the system will optimally reach an equilibrium and eventually remain there. The method aims to transform the problem of stability analysis into an analysis of the characteristics of a few specific Lyapunov functions, with the assumption that this analysis could be done without the need to integrate the original system. The first step consists of the formulation of an appropriate scalar function *i.e.* the Lyapunov function while the second step consists of evaluating the first order time derivative of the Lyapunov function along the trajectory of the system. The system will become stable if its energy dissipates and the derivative of the Lyapunov function decreases as time increases [3].

- **The indirect Lyapunov method for stability**

The indirect Lyapunov method explores the stability properties from the linearisation of a dynamical system described by non-linear differential equations in order to deduce the stability properties of the equilibrium. This approach assumes strong constraints that greatly limit the study of the stability of the said system.

➤ **Fundamental principles of Lyapunov stability theory**

Assume $V(x, t)$ is a non-negative function and its derivative lies along the trajectory of the system [3].

- The origin of the system is locally stable (in the Lyapunov sense) if $V(x, t)$ is locally positive definite and locally at x and for all t .
- The origin of the system is uniformly locally stable (in the Lyapunov sense) if $V(x, t)$ is locally positive and decreasing, and locally at x and for all t .
- The origin of the system is uniformly locally asymptotically stable (in the Lyapunov sense) if $V(x, t) \leq 0$ is locally positive definite and decreasing,

and is locally positive definite.

The origin of the system is globally uniformly asymptotically stable (in the Lyapunov sense) if $V(x, t) \leq 0$ is positive definite and decreasing, and $-\dot{V}(x, t) \leq 0$ is positive definite.

4) MIMO in 4G networks

The architecture of a 4G network can be presented as follows (Figure 4):

MIMO is used between the UE (User Equipment) and the E-UTRAN. It is one of the major technological breakthroughs of the 1990s in the field of signal processing. MIMO relies on the presence of multiple antennas at the transmitter and receiver to allow the transmission of multiple independent data streams over the same time-frequency resources. This is known as spatial multiplexing, where the spatial dimension is created by the multiple antennas. The most effective way to improve capacity (throughput) is to use multi-antenna technologies when the signal to noise ratio is high. Multi-antenna technology can make better use of space resources, it can improve the transmission capacity of a wireless communication system without increasing the transmitted power and bandwidth. MIMO advantages are presented below:

- Array gain: It increases the transmit power and can be used for beamforming.
- Diversity gain: It weakens the interference caused by channel fading.
- Spatial multiplexing gain: It doubles the rate within the same bandwidth after spatial orthogonal channels are constructed.

Figure 5 illustrates a MIMO system:

➤ **Classification of multi-antenna technologies**

On distingue quatre grandes techniques permises par la présence d’antennes multiples:

There are four main techniques enabled by the presence of multiple antennas:

- **Transmission diversity** consists of transmitting the same information from several antennas, possibly according to a specific coding of the information for each antenna;
- **beamforming** concentrates the signal energy in the direction of the receiver and thus increases the transmission rate;
- **Single User MIMO (SU-MIMO)** transmits several independent streams of

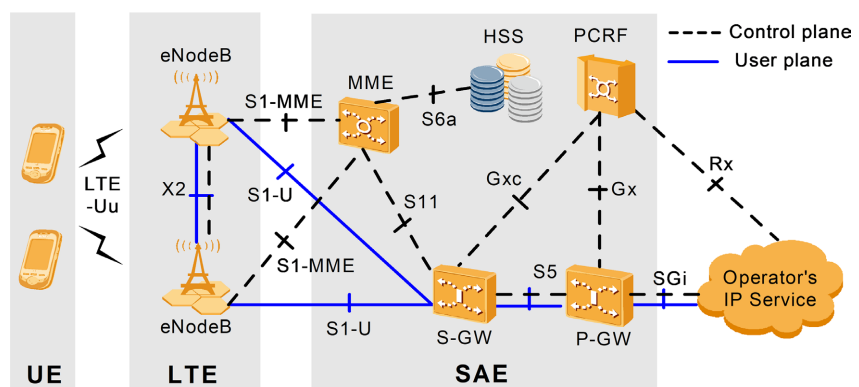


Figure 4. 4G network architecture.

information on the same time-frequency resources, separated in space;

- **Multi-User MIMO (MU-MIMO)** transmits spatially multiplexed streams to different receivers, which improves the overall system throughput.

➤ **Types of MIMO scenarios in LTE**

In LTE, we distinguish two main MIMO scenarios (illustrated in **Figure 6**) including:

- **SU-MIMO (Single User MIMO)**: which aims to increase the throughput of a single user, which would also improve the capacity of the cell. It can be used in both downlink and uplink channels;
- **MU-MIMO (Multi User MIMO)**: this is just implemented in the uplink nowadays. It improves the capacity gain of the cell.

Table 1 provides a summary comparison between SU-MIMO and MU-MIMO:

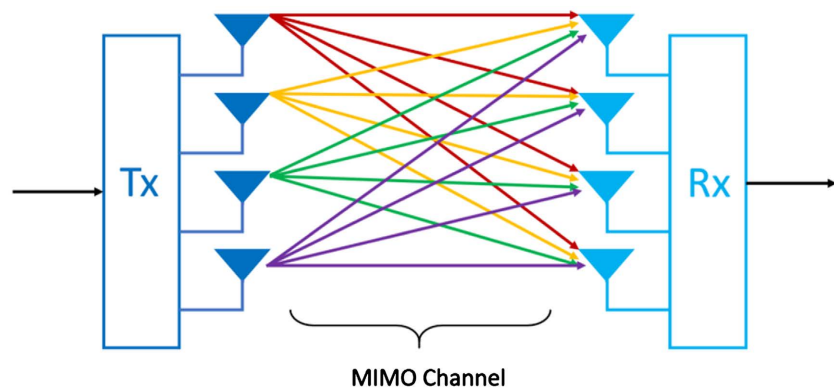


Figure 5. MIMO system.

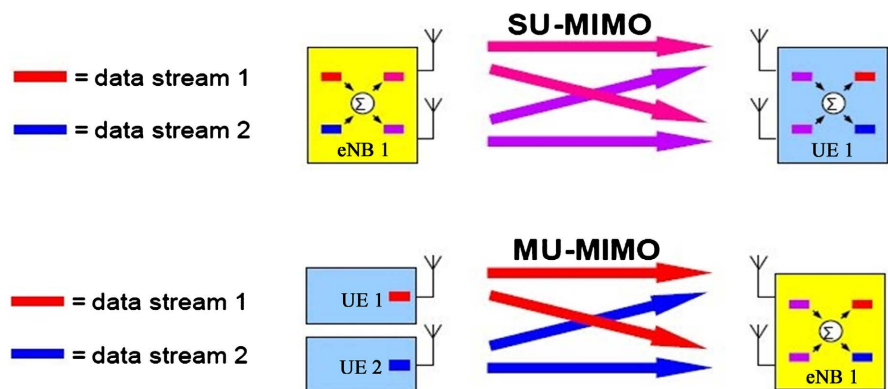


Figure 6. SU-MIMO and MU-MIMO.

Table 1. Comparison between SU-MIMO and MU-MIMO.

Characteristic	MU-MIMO	SU-MIMO
Main aspect	The cell communicates with several users	The cell communicates with a single user.
Objective	MIMO capacity gain	Increased user throughput for one user
Benefits	Multiplexing gain	No interference
Flow rate	Higher onset with high signal to noise ratio	Higher throughput with low signal to noise ratio

5) Radial basis function neural networks

Radial Basis Function Neural Networks (RBFNN) are neural networks that use radial basis functions as their activation function. A radial basis function is a real-valued function whose value depends only on the distance from its input parameter x to another given point, commonly called the origin or centre of the function. Any function Φ that satisfies the equality $\Phi(x) = \varphi(\|x\|)$ is a radial basis function. The norm $\|\cdot\|$ used corresponds to the Euclidean distance.

- **Architecture of a neural network with radial basis function**

Figure 7 shows a neural network with a radial basis function:

Based on this architecture, we can say that RBFNs are typically composed of three layers, namely

- **The input layer:** it simply transmits input features to the hidden layers. Therefore, the input layer and the data are of the same size. No calculations are performed in the input layers. As in the case of *feed-forward* networks, the input units are fully connected to the hidden units and transmit their data upstream;
- **The hidden layer:** the power of an RBFNN is based on the structure and calculations performed at this level. Indeed, the hidden layer performs a calculation on a comparison with a prototype vector. Each hidden unit contains a prototype vector of dimension d of the input data. Let the prototype vector of the i -th hidden unit be denoted by $\bar{\mu}_i$. In addition, the i th hidden unit contains a bandwidth denoted by σ_i . Although prototype vectors are always specific to particular units, the bandwidths of different units σ_i are often set to the same value σ . The prototype vectors and bandwidth(s) are usually learned either unsupervised or using a learning algorithm. Thus, for any

training point the activation function $h_i = \Phi_i(\bar{X}) = \exp\left(-\frac{\|\bar{X} - \bar{\mu}_i\|^2}{2 \cdot \sigma_i^2}\right)$

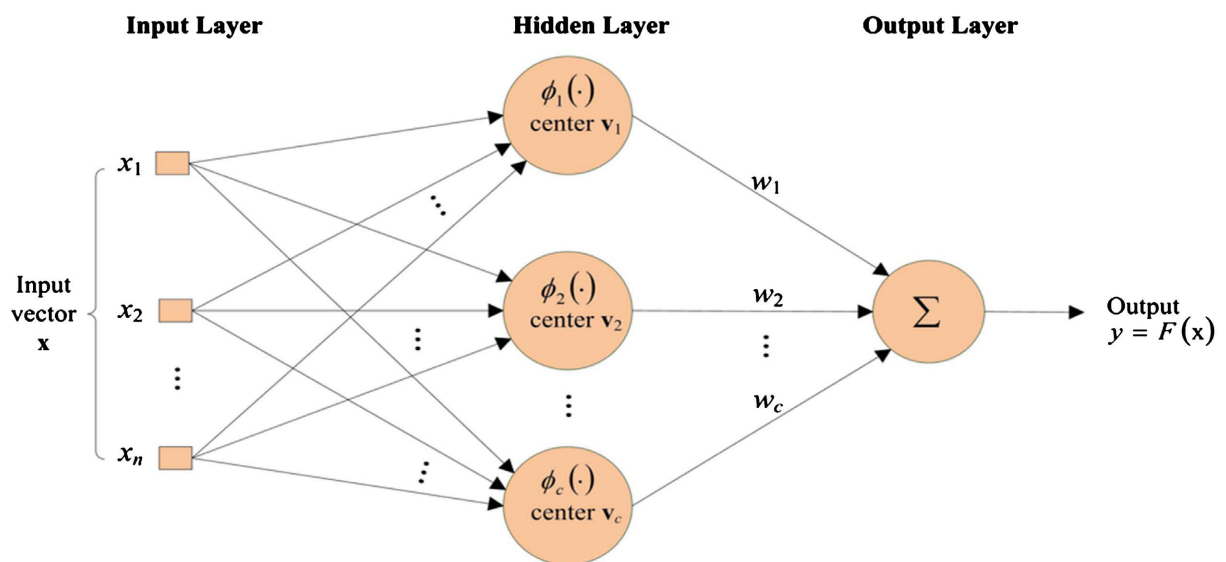


Figure 7. Basic architecture of a neural network with radial basis function.

$\forall i \in \{1, \dots, m\}$. The total number of hidden units is denoted m .

- **the output layer:** for any learning point \bar{X} , h_i , is the output of the i -th hidden unit as defined by equation (). The weights of the connections between the hidden nodes and the output nodes are set by. The prediction of the RBF network in the output layer is defined as follows:

$$\hat{y} = \sum_{i=1}^m w_i h_i = \sum_{i=1}^m w_i \Phi_i(\bar{X}) = \sum_{i=1}^m w_i \exp\left(-\frac{\|\bar{X} - \bar{\mu}_i\|^2}{2 \cdot \sigma_i^2}\right)$$

➤ Applications of RBFNs

A key point is that the hidden layer of an RBFNN is created in an unsupervised manner, which tends to make it robust to all types of noise. [6]

RBFNNs can be used in a number of areas including

- classification and regression problems;
- universal approximation of functions (linear and non-linear).

2.2. Literature Review

M.S. Ahmed and S.H. Riyaz in “*Dynamic observer—a neural net approach*” [7] work on a general nonlinear multiple-input multiple-output (MIMO) system that has been linearized and an extended Kalman filter has been used to estimate the system states. The gain of the proposed observer was computed by a multi-layer feedforward neural network.

A.S. Poznyak, E.N. Sanchez *et al.* in “*output trajectory tracking using dynamic neural networks*” [8] consider a general nonlinear model. It was stated that any general nonlinear model can be described by an affine model plus a bounded unmodeled dynamic term. Therefore, this affine model was used for the design of the observer. No clear method was suggested to decrease the amount of error in an arbitrary way.

J.A.R Vargas, E.M. Hemerly in “*Robust neural adaptive observer for MIMO nonlinear systems*” [9] propose an observer for a general nonlinear MIMO system using a neural network linear in its parameters. The strict real positive assumption has been relaxed. However, according to the authors, it is extremely difficult to choose appropriate values of design parameters such as the different gains and functional links of the neural networks. Moreover, the observer has an open-loop structure.

N. Hovakimyan, A.J Calise and V.K Madyastha in “*An adaptive observer design methodology for bounded nonlinear networks*” [5] propose a nonlinear observer based on a nonlinear neural network. However, a linear approximation was used using a Taylor series expansion to facilitate stability analysis.

A. Alessandri, C. Cervellera, F. Grassia *et al.* in “*Design of observers for continuous-time nonlinear systems using neural networks*” [10] propose a neural network based observer for a general nonlinear MIMO system. The weight updating mechanism is performed using the gradient descent method. The observer has been shown to be experimentally stable but no mathematical proof has

been provided.

H.A. Talebi, R.V. Patel *et al.* in “A neural network based observer for flexible joint manipulators” [11] propose a state observer based on a general model of nonlinear MIMO systems and was found to be experimentally stable, but no mathematical evidence was given to support the experiments.

F. Abdollahi, H.A. Talebi, R.V. Patel *et al.* in “A stable neural network observer with application to flexible-joint manipulators” [12] propose a recurrent neural network based observer has been used for general nonlinear MIMO systems. The strict positive real assumption was also relaxed. The weights of the neural network are updated by the backpropagation algorithm. Although the stability of the observer has been demonstrated by the Lyapunov method. However, like most neural network based observers, it has been linearly parameterised. This assumption greatly simplifies the analysis, but it is a strong constraint, as not all non-linear functions can be represented by such equations.

F. Abdollahi, H.A. Talebi, R.V. Patel. in “A stable neural network-based observer with application to flexible joint manipulators” [2] propose an adaptive observer based on a recurrent neural network for a general model of nonlinear MIMO systems. The neural network is non-linear. The weight updating mechanism is a modified version of the backpropagation algorithm with a simple structure and an e-modification term added for robustness. The strict positive real assumption has been relaxed. The stability of the observer was done using the direct Lyapunov method.

In this work, we propose a new approach for non-linear MIMO systems. We propose a state observer based on a neural network with radial basis function. No strict real positive assumption is imposed on the output error equation. We use the direct Lyapunov method to prove the stability of the observer and the neural network. The weight updating mechanism follows the results established in [2].

3. Method

3.1. The Proposed Neuro-Adaptive Observer

We would like to remind you that this methodology is based on the work of [2].

Consider the general model of a non-linear MIMO system:

$$\begin{aligned}\dot{x}(t) &= f(x, u) \\ y(t) &= Cx(t)\end{aligned}\tag{2}$$

where $u \in \mathbb{R}^{m_u}$ is the input, $y \in \mathbb{R}^{m_y}$ is the output, $x \in \mathbb{R}^n$ is the state vector of the system and f is a vector-valued non-linear function. It is assumed that the non-linear system () is **observable**. Another assumption made here is that the open loop system is stable. In other words, the states of the system are bounded in L_∞ . This is a common assumption in identification schemes [2].

Now, by adding and subtracting Ax , (2) becomes:

$$\begin{aligned}\dot{x}(t) &= Ax + g(x, u) \\ y(t) &= Cx(t)\end{aligned}\tag{3}$$

where A is a Hurwitz matrix, the pair (C, A) is observable and $g(x, u) = f(x) - Ax$

Now the observer model can be chosen:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{g}(\hat{x}, u) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{4}$$

where \hat{x} denotes the observer state, and the observer gain $G \in \mathbb{R}^{n \times m_y}$ is chosen such that $A - GC$ is a Hurwitz matrix. The existence of such a gain is guaranteed since it can be chosen such that the pair (C, A) is observable. The key to designing a neuro-observer is to use a neural network to identify the non-linearity and a conventional observer to estimate the states. It is well known that a three-layer neural network is capable of approximating non-linear systems with any degree of non-linearity [2]. Indeed, it has been shown by several researchers that for x restricted to a compact set of $x \in \mathbb{R}^n$ and for a sufficiently large number of hidden layer neurons, there will exist weights and weighting coefficients such that any continuous function on this compact set S can be represented by:

$$g(x, u) = W\sigma(V\bar{x}) + \epsilon(x)$$

where W and V are the weighting matrices of the output and hidden layers respectively, $\bar{x} = [xu]$; $\epsilon(x)$; is the bounded approximation error of the neural network and $\sigma(\cdot)$ is the transfer function of the hidden neurons and is generally considered as a sigmoid function.

$$\sigma_i(V_i\bar{x}) = \frac{2}{1 + \exp^{-2V_i\bar{x}}} - 1.$$

where V_i is the i -th row of V , and is the i -th element of and $\sigma_i(V_i\bar{x})$ is the i th element of $\sigma(V\bar{x})$.

We assume that the upper bound on the fixed ideal weights W and V exists such that:

$$\|W\|_F \leq W_M \tag{5}$$

$$\|V\|_F \leq V_M \tag{6}$$

We also assume that the sigmoidal function is bounded by:

$$\|\sigma(V\bar{x})\| \leq \sigma_m. \tag{7}$$

Thus, the function g can be approximated by:

$$\hat{g}(\hat{x}, u) = \hat{W}\sigma(\hat{V}\hat{x}). \tag{8}$$

The proposed observer is therefore given by:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{W}\sigma(\hat{V}\hat{x}) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{9}$$

Let us define the state estimation error as $\tilde{x} = x - \hat{x}$ and using Equations (3), (7) and (8), we can express the dynamics of the error as follows:

$$\begin{aligned} \dot{\hat{x}}(t) &= Ax + W\sigma(V\bar{x}) - A\hat{x} - \hat{W}\sigma(\hat{V}\hat{x}) - G(Cx - C\hat{x}) + \epsilon(x) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{10}$$

By adding and subtracting $\hat{W}\sigma(\hat{V}\hat{x})$ from (10), we can write:

$$\begin{aligned} \dot{\hat{x}}(t) &= A_c\tilde{x} + \tilde{W}\sigma(\hat{V}\hat{x}) + w(t) \\ \hat{y}(t) &= C\tilde{x}(t) \end{aligned} \tag{11}$$

où $\tilde{W} = W - \hat{W}$, $A_c = A - GC$, $w(t) = W[\sigma(V\hat{x}) - \sigma(\hat{V}\hat{x})] + \epsilon(x)$ est un terme de perturbation i.e. $\|w(t)\| \leq \bar{w}$ pour une certaine constante positive \bar{w} due à la fonction sigmoïdale et le fait que les poids idéaux du réseau neuronal soit bornés.

Where $\tilde{W} = W - \hat{W}$, $A_c = A - GC$, $w(t) = W[\sigma(V\hat{x}) - \sigma(\hat{V}\hat{x})] + \epsilon(x)$ is a disturbance term i.e. $\|w(t)\| \leq \bar{w}$ for a certain positive constant \bar{w} due to the sigmoidal function and the fact that the ideal weights of the neural network are bounded.

The structure of our observer is shown in **Figure 8**.

3.2. Stability Analysis

The stability of the observer is closely related to the stability of the proposed neural network. The stability of the neural network is evaluated here by the weight update mechanism.

Recall the Theorem established in [2] Consider the model defined in (2) and the observer in (9). If the weights of the neural network are updated according to the following:

$$\dot{\hat{W}} = -\eta_1 (\hat{y}^T C A_c^{-1})^T (\sigma(\hat{V}\hat{x}))^T - \rho_1 \|\hat{y}\| \hat{W} \tag{12}$$

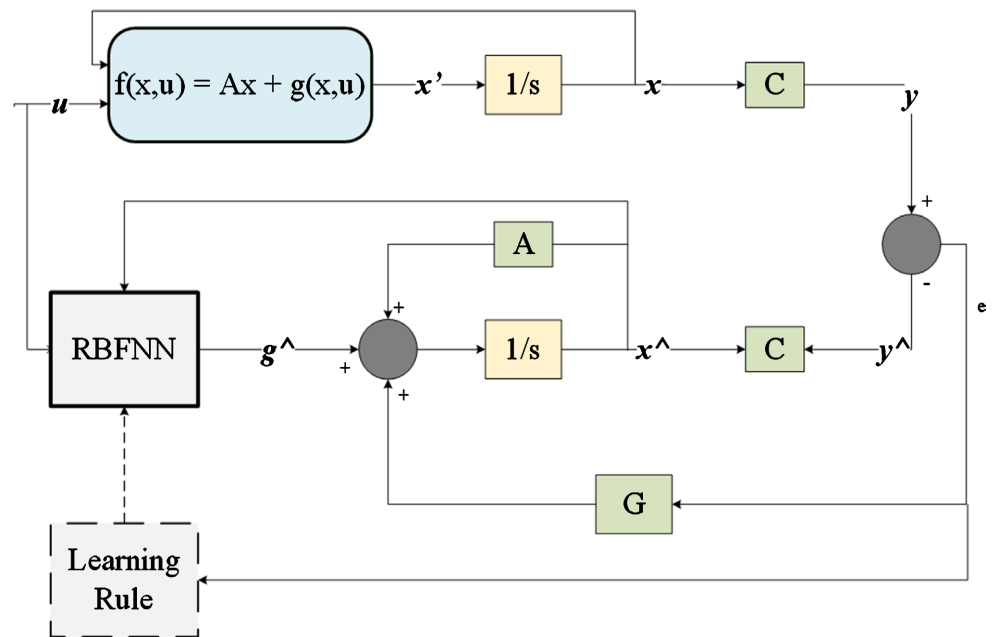


Figure 8. Structure of the state observer.

$$\dot{\hat{V}} = -\eta_2 \left(\tilde{y}^T C A_c^{-1} \hat{W} \left(I - \Lambda(\hat{V} \hat{x}) \right) \right)^T \text{sgn}(\hat{x})^T - \rho_2 \|\tilde{y}\| \hat{V} \tag{13}$$

where $\Lambda(\hat{V} \hat{x}) = \text{diag} \left\{ \sigma_i^2(\hat{V}_i \hat{x}) \right\}, i = 1, 2, \dots, m$ and $\text{sgn}(\hat{x})$ is the sign function defined by:

$$\text{sgn}(\hat{x}) = \begin{cases} 1 & \text{pour } \hat{x} > 0 \\ 0 & \text{pour } \hat{x} = 0 \\ -1 & \text{pour } \hat{x} < 0 \end{cases}$$

then, $\tilde{x}, \tilde{W}, \tilde{V}, \tilde{y} \in L_\infty$ i.e. the estimation error, the weight error and the output error are bounded. In these equations, η_1 and η_2 represent the learning rates, $J = \frac{1}{2}(\tilde{y}^T \tilde{y})$ is the objective function and ρ_1 and ρ_2 are small positive numbers.

The first terms in (12) and (13) represent the back-propagation terms and the second terms are the modification terms to materialize the damping in the equations i.e.:

$$\dot{\hat{W}} = -\eta_1 \left(\frac{\partial J}{\partial \hat{W}} \right) - \rho_1 \|\tilde{y}\| \hat{W} \tag{14}$$

$$\dot{\hat{V}} = -\eta_2 \left(\frac{\partial J}{\partial \hat{V}} \right) - \rho_2 \|\tilde{y}\| \hat{V} \tag{15}$$

This result was used to update the weights of the radial basis function neural network used.

For Lyapunov stability, [2] uses the following Lyapunov function:

$$L = \frac{1}{2} \tilde{x}^T P \tilde{x} + \frac{1}{2} \text{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T \tilde{V})$$

where $P = P^T$ where is a positive definite function satisfying the algebraic Lyapunov equation:

$$A_c^T P + P A_c = -Q$$

for the Hurwitz matrix A_c and for a certain positive definite matrix Q .

The time derivative of the Lyapunov function is defined by:

$$\dot{L} = \frac{1}{2} \dot{\tilde{x}}^T P \tilde{x} + \frac{1}{2} \tilde{x}^T P \dot{\tilde{x}} + \text{tr}(\tilde{W}^T \dot{\tilde{W}}) + \text{tr}(\tilde{V}^T \dot{\tilde{V}})$$

In fact, \dot{L} is defined negative outside the ball of radius described by: $\chi = \{ \tilde{x} \mid \|\tilde{x}\| > b \}$, and \tilde{x} is uniformly bounded with:

$$b = \frac{2 \left(\|P\| \bar{w} + (\rho_1 \|C\| - K_1^2) K_2^2 + (\rho_2 \|C\| - 1) K_3^2 \right)}{\lambda_{\min}(Q)}$$

$$K_1 = \frac{\|l_2\|}{2}$$

$$K_2 = \frac{\rho_1 W_M \|C\| + \sigma_m \|l_1\| + \|P\| \sigma_m}{2(\rho_1 \|C\| - K_1^2)}$$

$$K_3 = \frac{\rho_2 V_M \|C\| + W_M \|l_2\|}{2(\rho_2 \|C\| - 1)}$$

$$l_1 = \eta_1 A_c^{-T} C^T C, l_2 = \eta_2 A_c^{-T} C^T C$$

4. Results and Interpretations

4.1. Presentation of the State Observer

The state observer built using Simulink blocs is presented as follows (Figure 9):

Since the overall scheme is quite large, we thought of using subsystems to define our observer. Indeed, according to Figure 9, we can see that our observer consists mainly of five (05) blocks or subsystems.

- **The MIMO system**

The MIMO system is presented as follows in Figure 10:

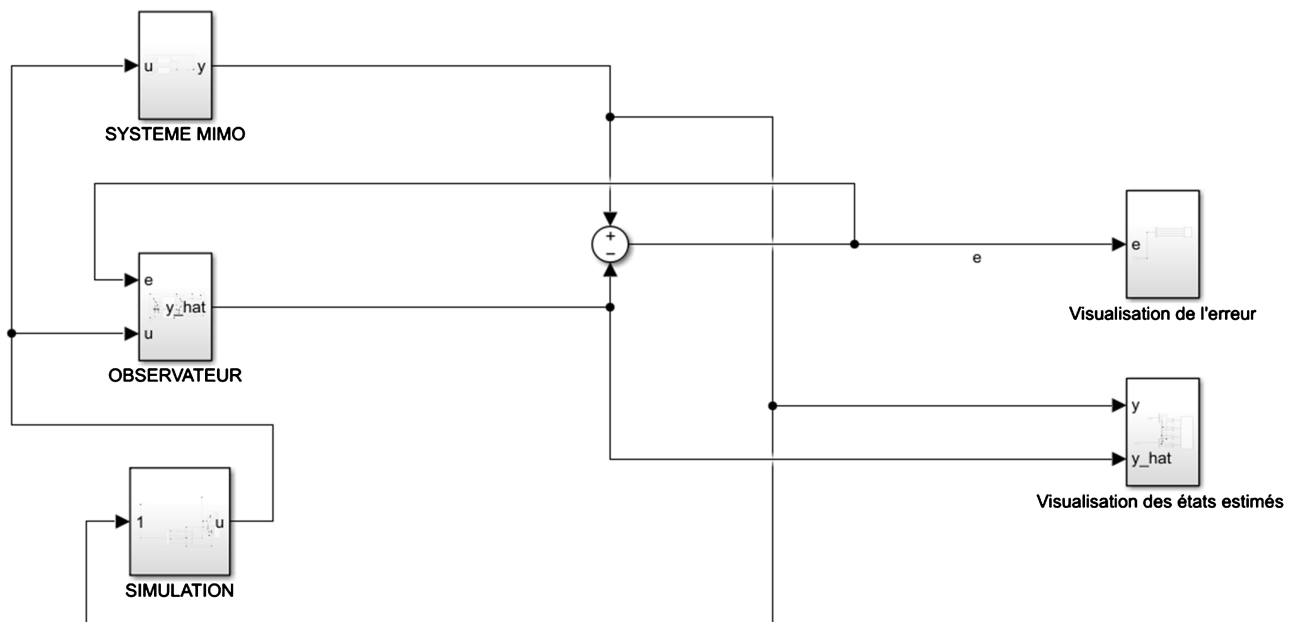


Figure 9. Presentation of the state observer.

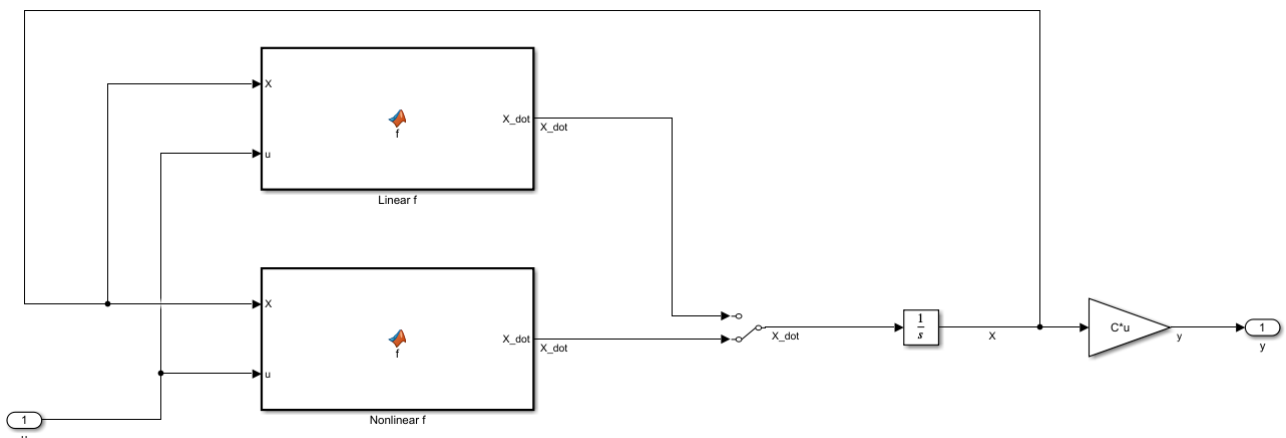


Figure 10. MIMO system simulation using simulink.

The MIMO system models the differential system as defined in (2). In order to generalize our solution, we have chosen to model in Simulink both nonlinear and linear MIMO systems. However, it is the non-linear case that interests us.

- **The Observer neuro-adaptive**

Observer’s diagram is presented in **Figure 11** as follows:

We can see that our observer is indeed a neural network with a radial basis function.

- **The simulation**

This block allows us to carry out the simulation for the transmission of signals, **Figure 12** shows the simulation block of the observer in SIMULINK.

- **Visualisation of estimated states**

Block allows us to check if our observer is able to perform a good estimation/reconstruction of states. It is presented as follows:

As can be seen in **Figure 13** this block takes as input the real states and the estimated states in order to visualize if the observer has been able to estimate the real states well.

- **Viewing the error**

This block allows us to visualise the output error in order to appreciate not only the performance but also the experimental stability of our observer. It is presented in **Figure 14**.

4.2. Tests and Interpretations

In order to perform our tests, we set the parameters of our neural network as presented in **Table 2**.

We have performed some simulations to test our observer. The simulations

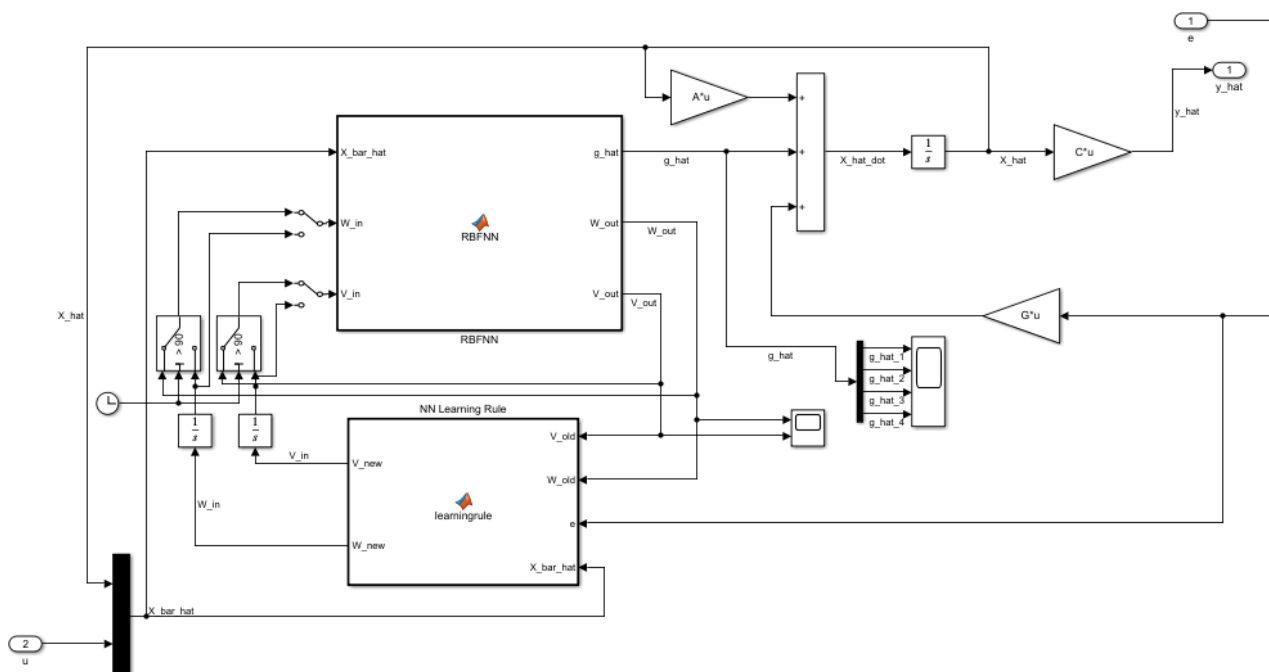


Figure 11. Structure of our observer in SIMULINK.

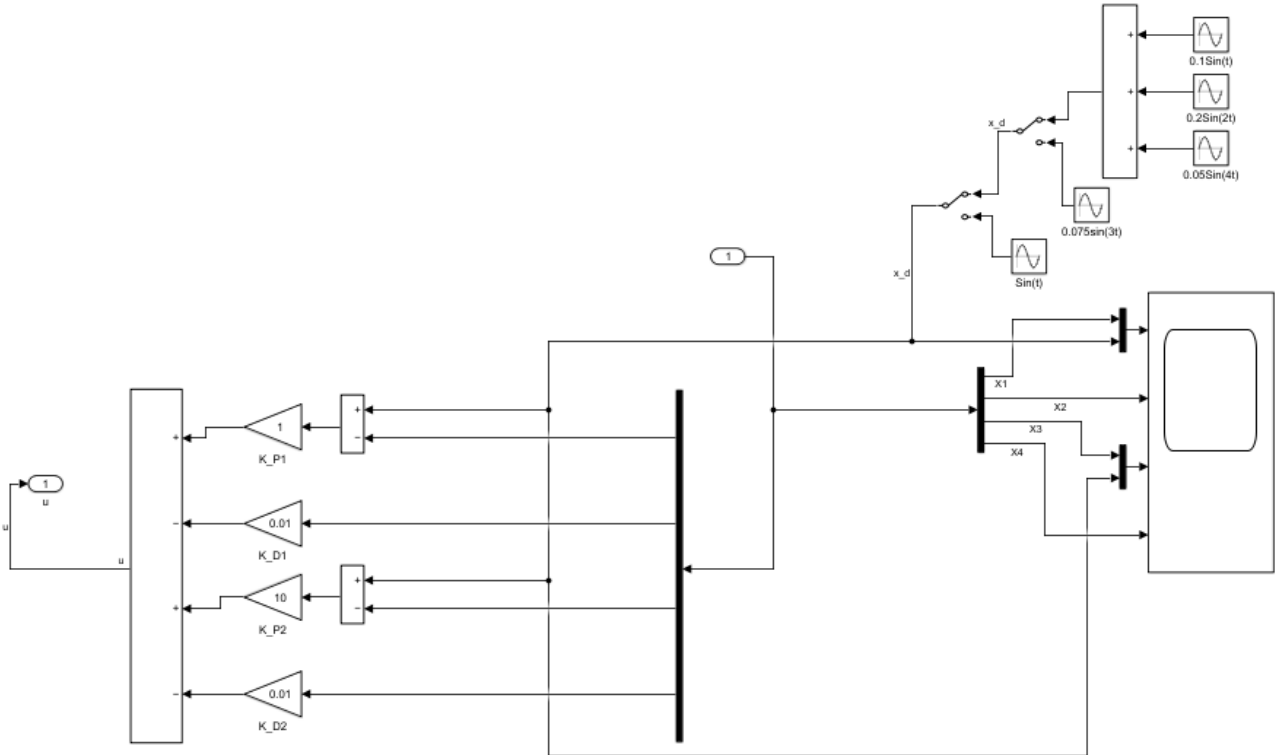


Figure 12. Simulation block for our observer in SIMULINK.

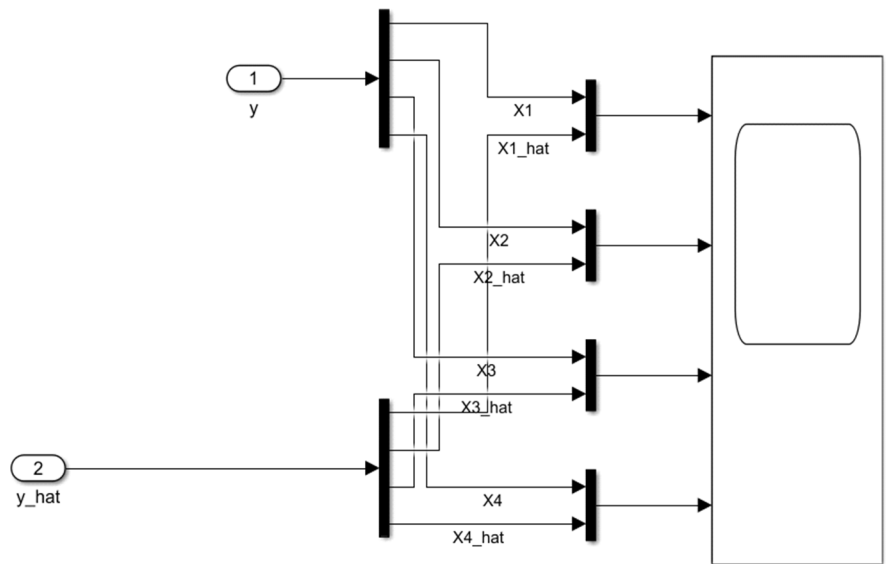


Figure 13. SIMULINK block for displaying estimated states.

are done for signals. We have obtained the following simulations:

According to Figure 15 we can see that there is almost no difference between the actual states and the states estimated by our observer. Our observer is therefore able to estimate the states for the aggregate signals $0.1\sin(t)$, $0.2\sin(2t)$ and $0.05\sin(4t)$ quite efficiently.

Figure 16 also shows that the error for the different states tends to cancel

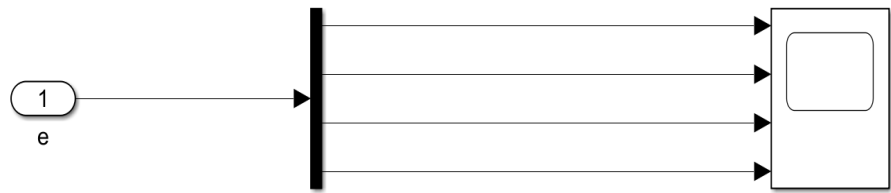


Figure 14. SIMULINK error display block.

Table 2. Simulation parameters for the neural network.

Hyperparameters	Values
η_1	5000
η_2	5000
ρ_1	0.5
ρ_2	1.5
Hidden Layer	5 (pour la première simulation) puis 10
Output layer	4
Input layer	1

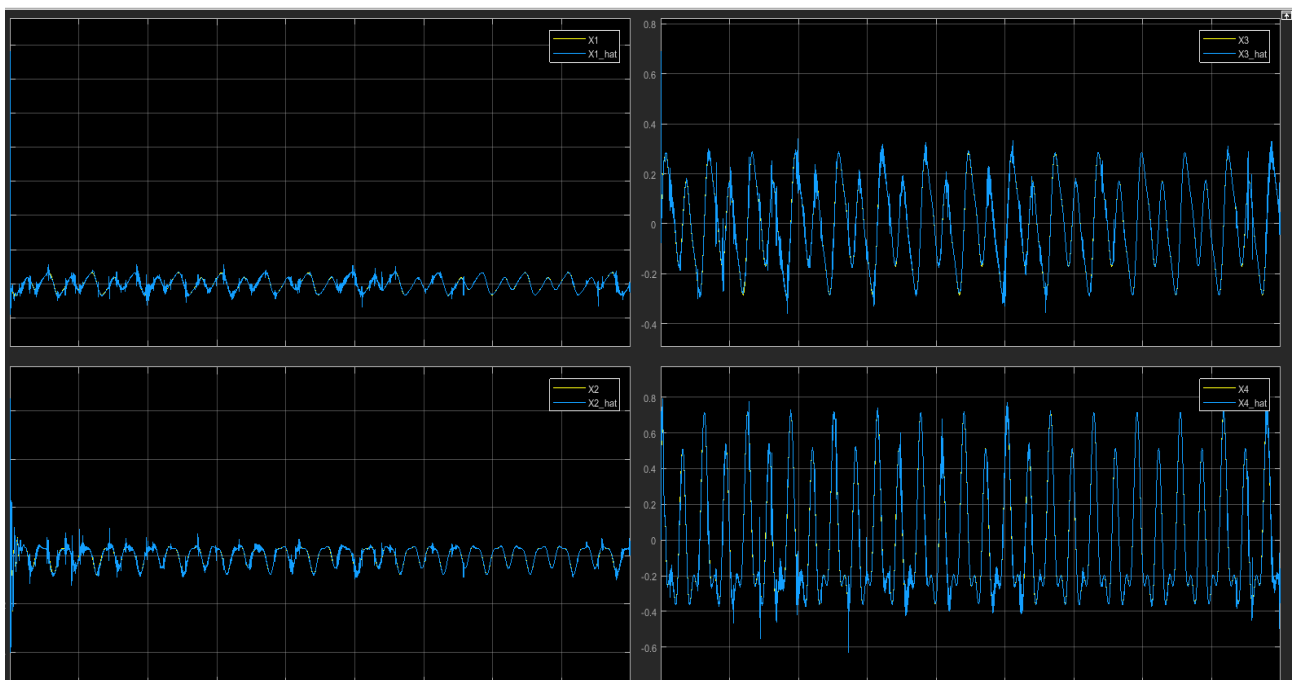


Figure 15. Estimated states for the aggregate signals $0.1\sin(t)$, $0.2\sin(2t)$ and $0.05\sin(4t)$.

out. The differences are not significant.

Figure 17 also shows us that for the signal $0.075\sin(3t)$, our observer manages to reproduce the estimated states. The small peaks observed are due to the processing by the hidden layers.

Figure 18 shows us that the error for the $0.075\sin(3t)$ signal tends to cancel out after a certain time (about 50 s).

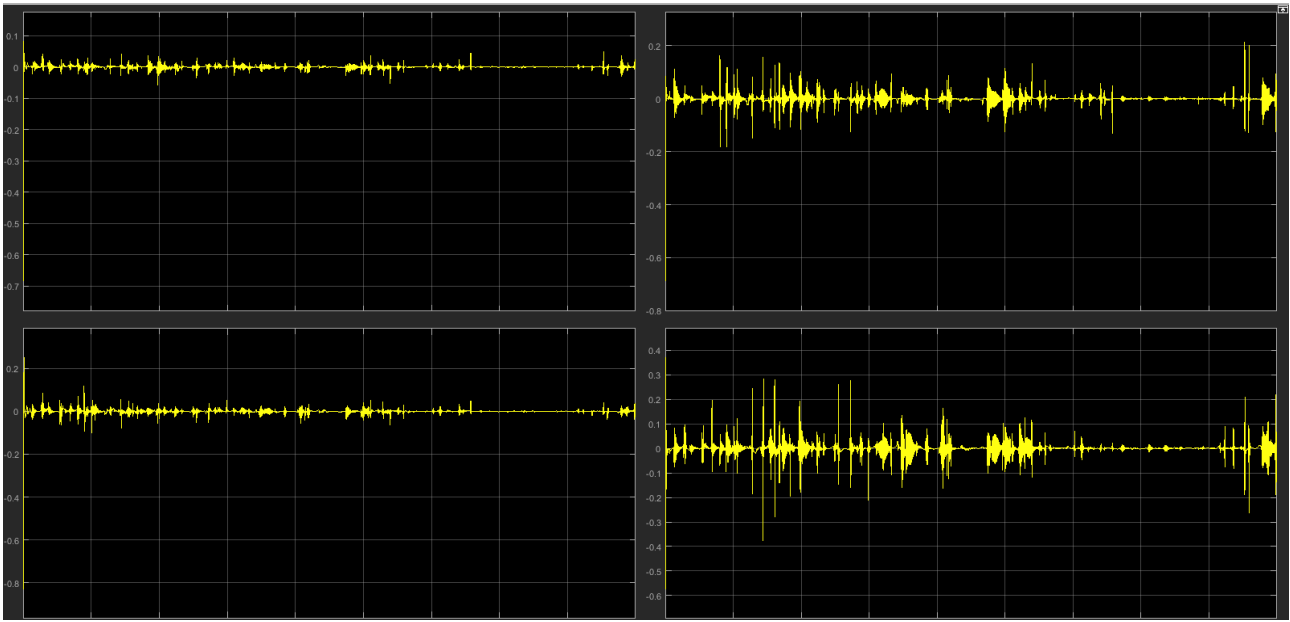


Figure 16. Visualisation of the error for the aggregated signals $0.1\sin(t)$, $0.2\sin(2t)$ and $0.05\sin(4t)$.

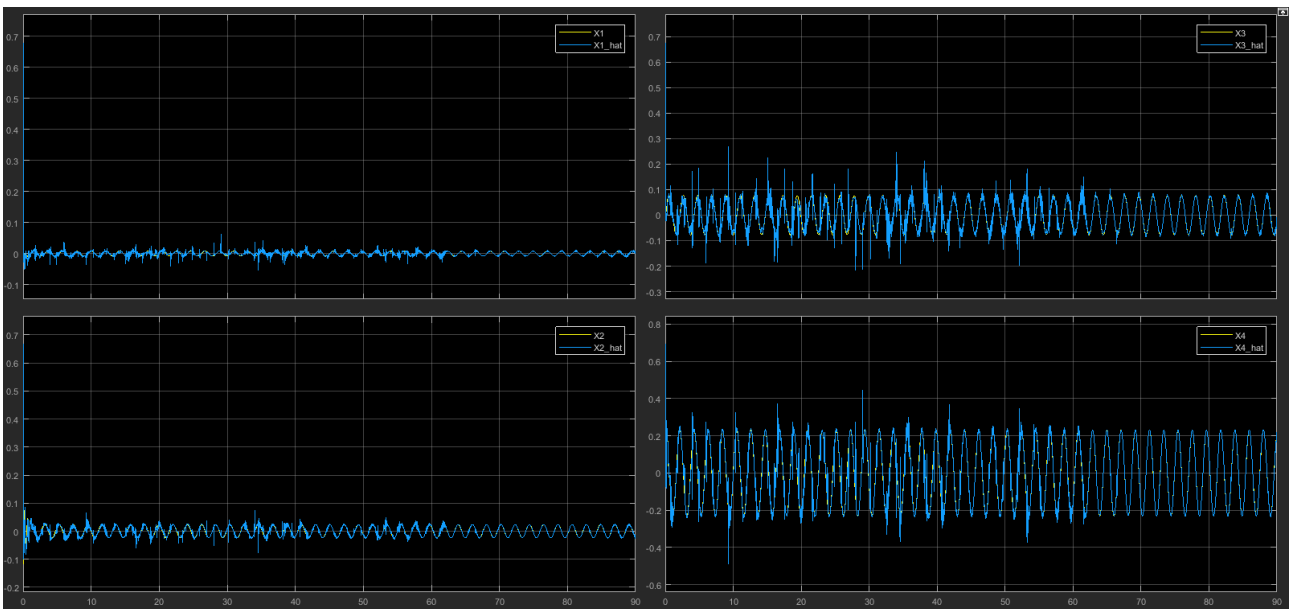


Figure 17. Visualisation of the estimated states for the $0.075\sin(3t)$ signal.

Figure 19 further demonstrates that our observer reconstructs the states almost perfectly.

Figure 20 shows us that the observed deviations are quite close to zero.

We can therefore state after the various simulations carried out that our observer is not sensitive to variations in the pulsation and amplitude of the transmitted signals. Indeed, the observer manages to estimate the states of our signals quite accurately by minimising the error in an optimal way. Subsequently, we simulated for a noisy signal with a non-zero phase. **Figure 21** shows the results.

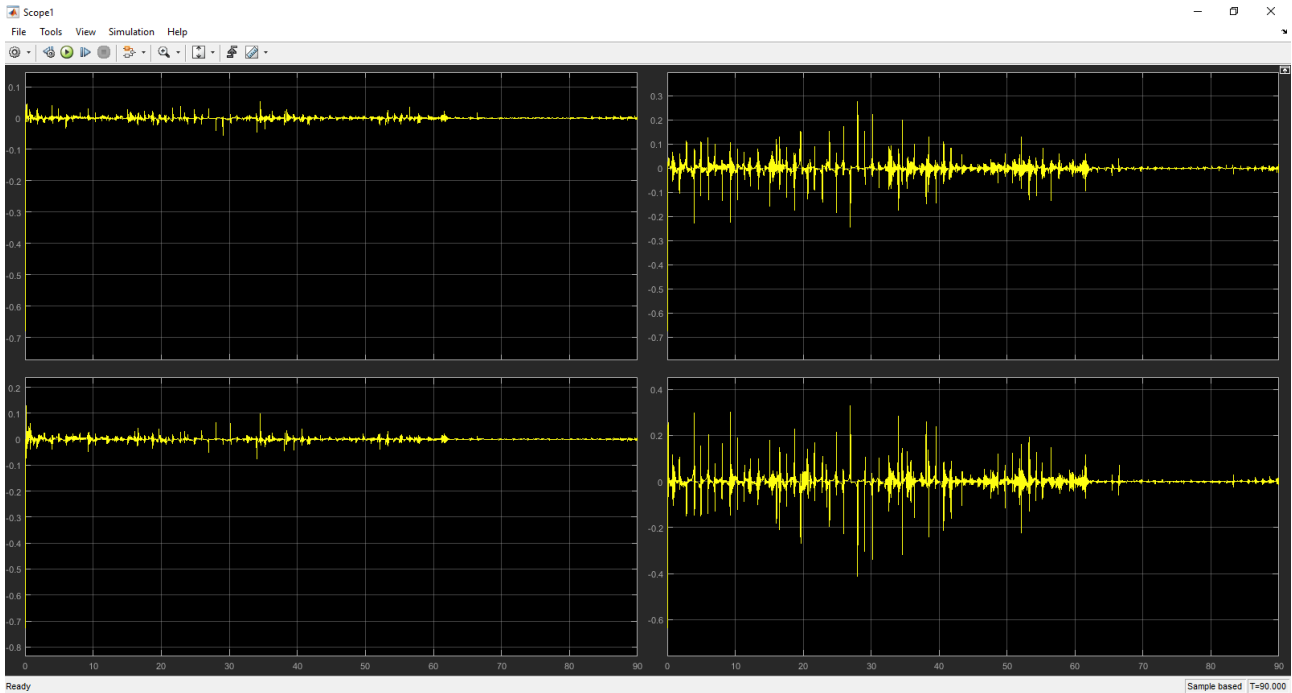


Figure 18. Errors for the $0.075\sin(3t)$ signal.

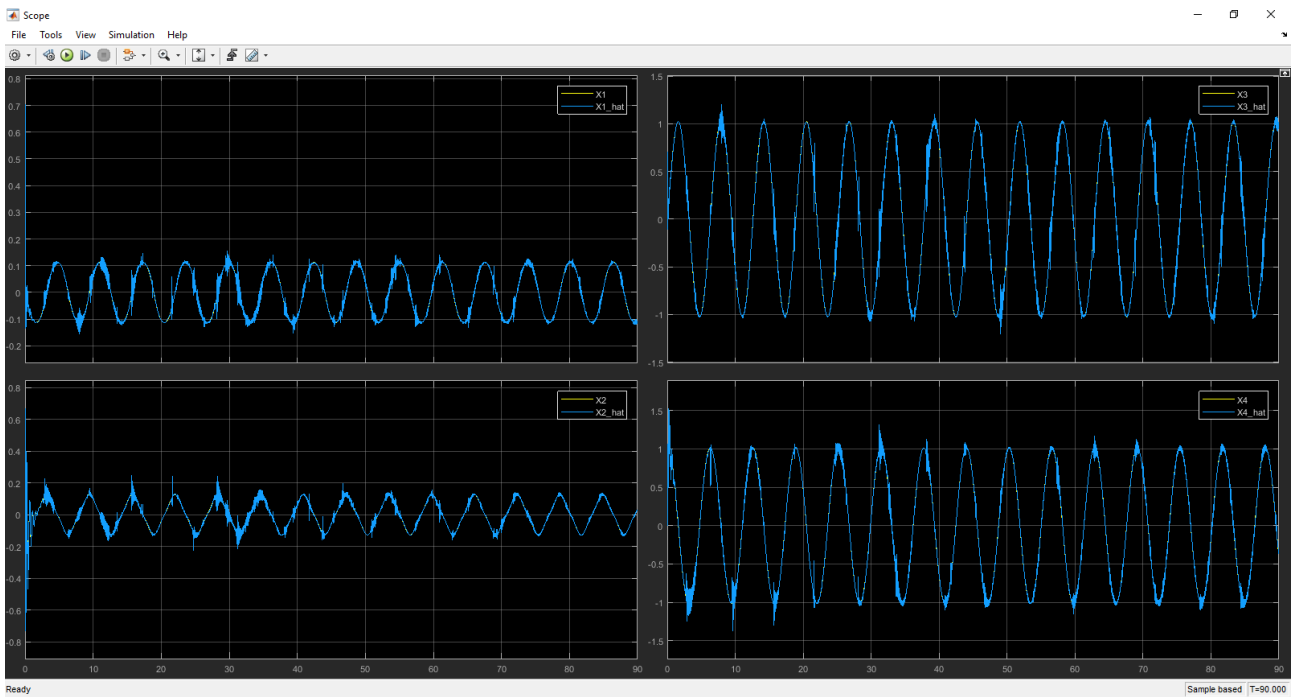


Figure 19. Visualisation of the estimated states for signal sint .

We see that the observer is able to reproduce the states but at some levels we see quite large spikes. This is due to the number of hidden layers in our neural network. Indeed, the current number of hidden layers in our neural network is not sufficient to reconstruct the signal “perfectly”.

We note that although the error is quite close to zero, **Figure 22** shows some

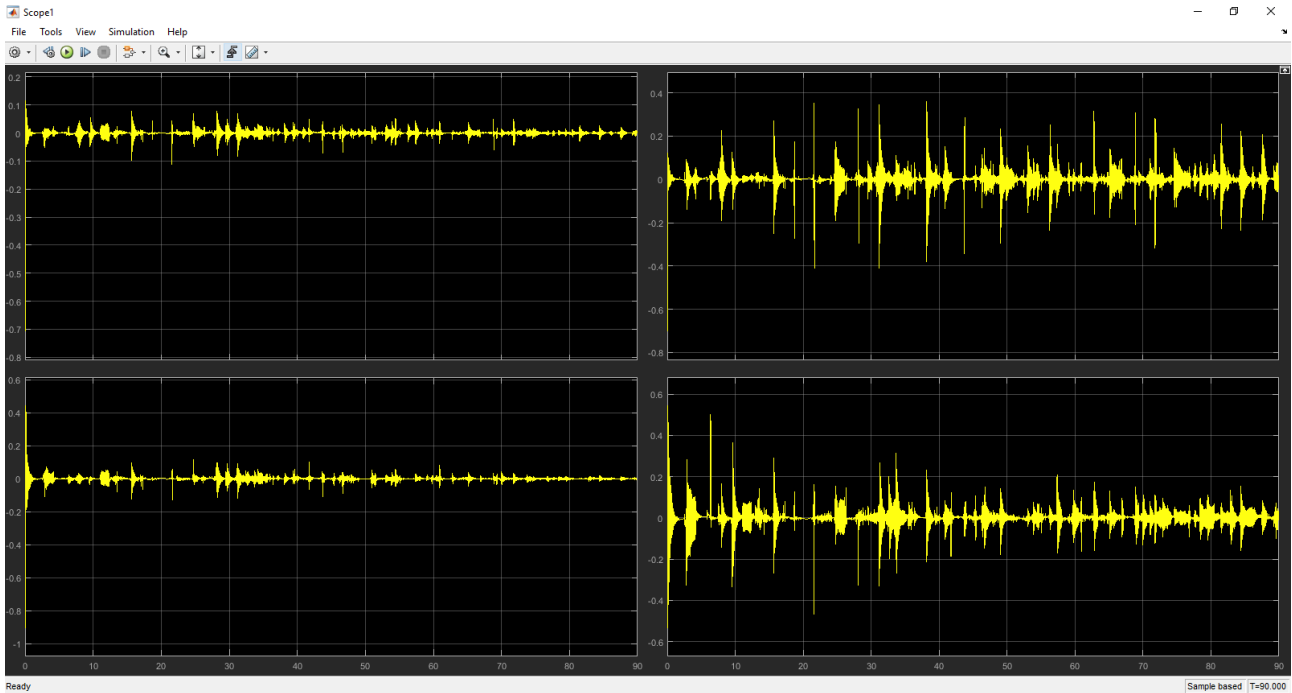


Figure 20. Error visualisation for the $\sin(t)$ signal.

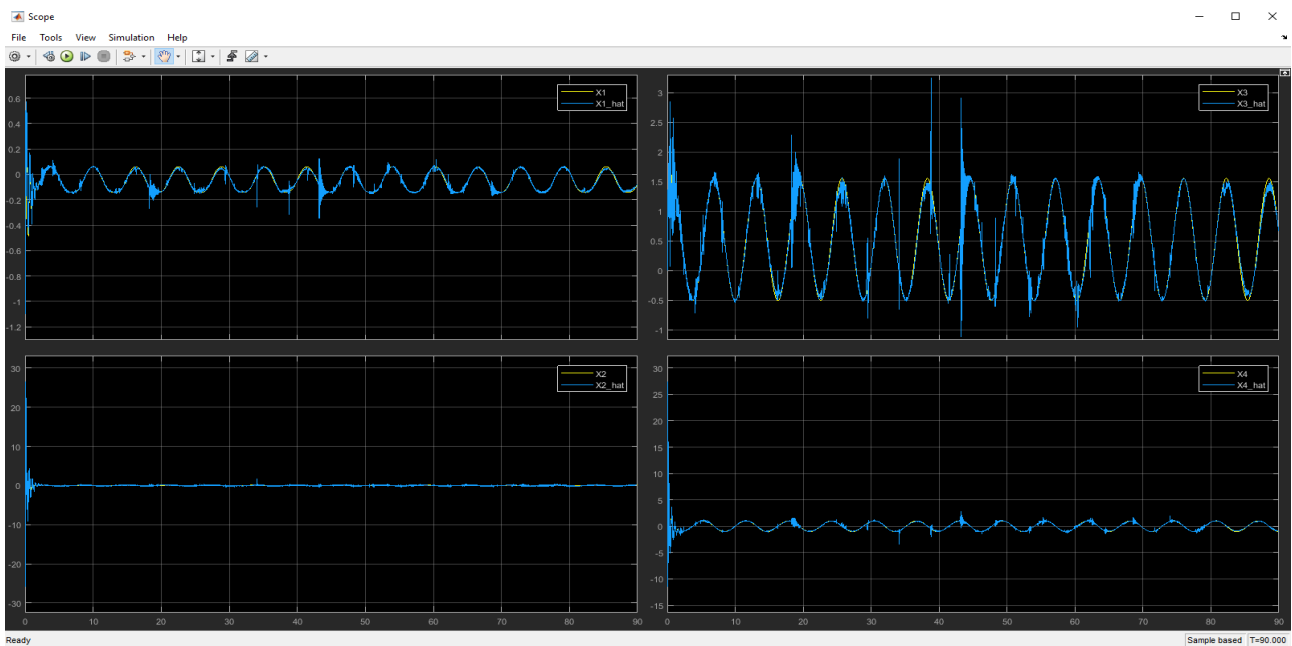


Figure 21. Visualization of the estimated states for $\sin(t + \pi/3) + 0.5$.

rather large peaks. Subsequently, we have seen that by increasing the number of hidden layers, the error has dropped considerably and the signal reconstruction is almost perfect.

Indeed, after increasing the number of hidden layers we got the results presented in **Figure 23** and **Figure 24**. We notice that the estimated states no longer show considerable peaks. Indeed, this allows us to state that our observer can

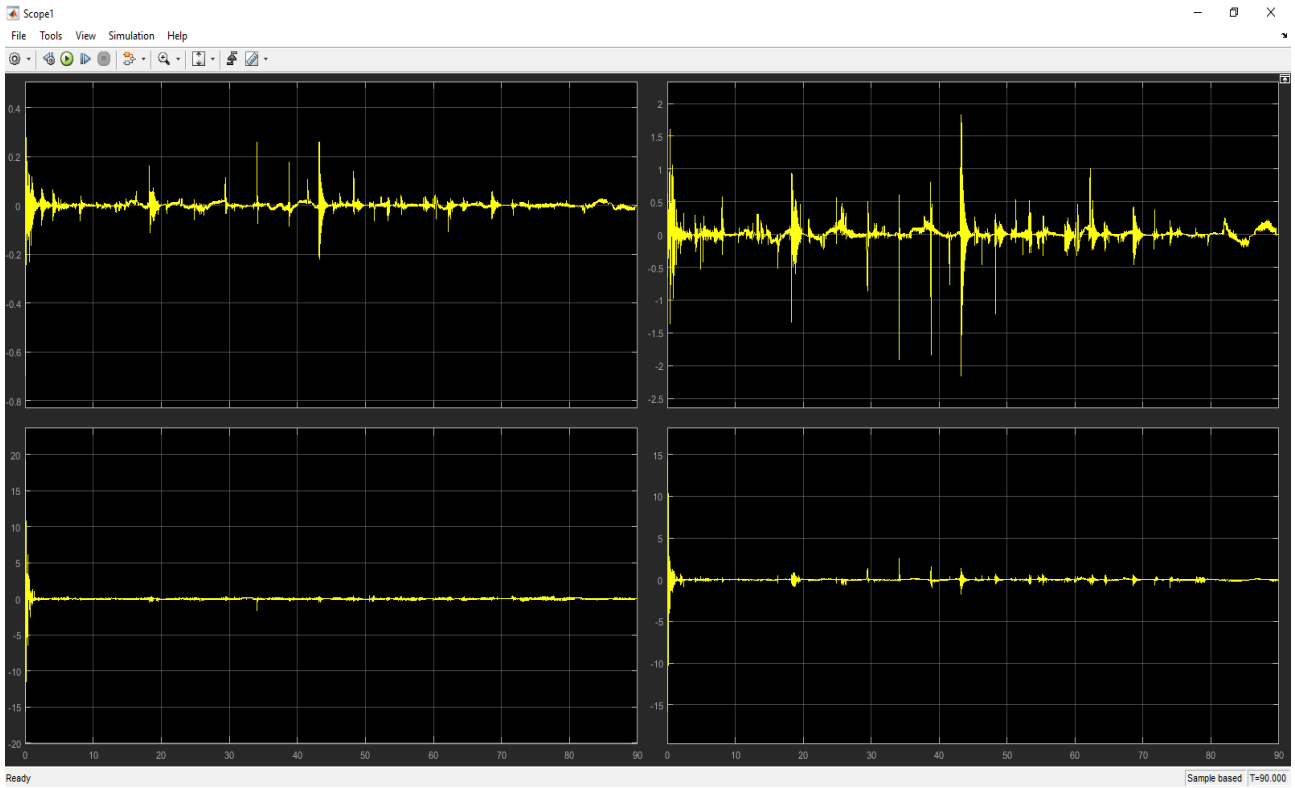


Figure 22. Visualization of the error for the signal $\sin(t + \pi/3) + 0.5$.

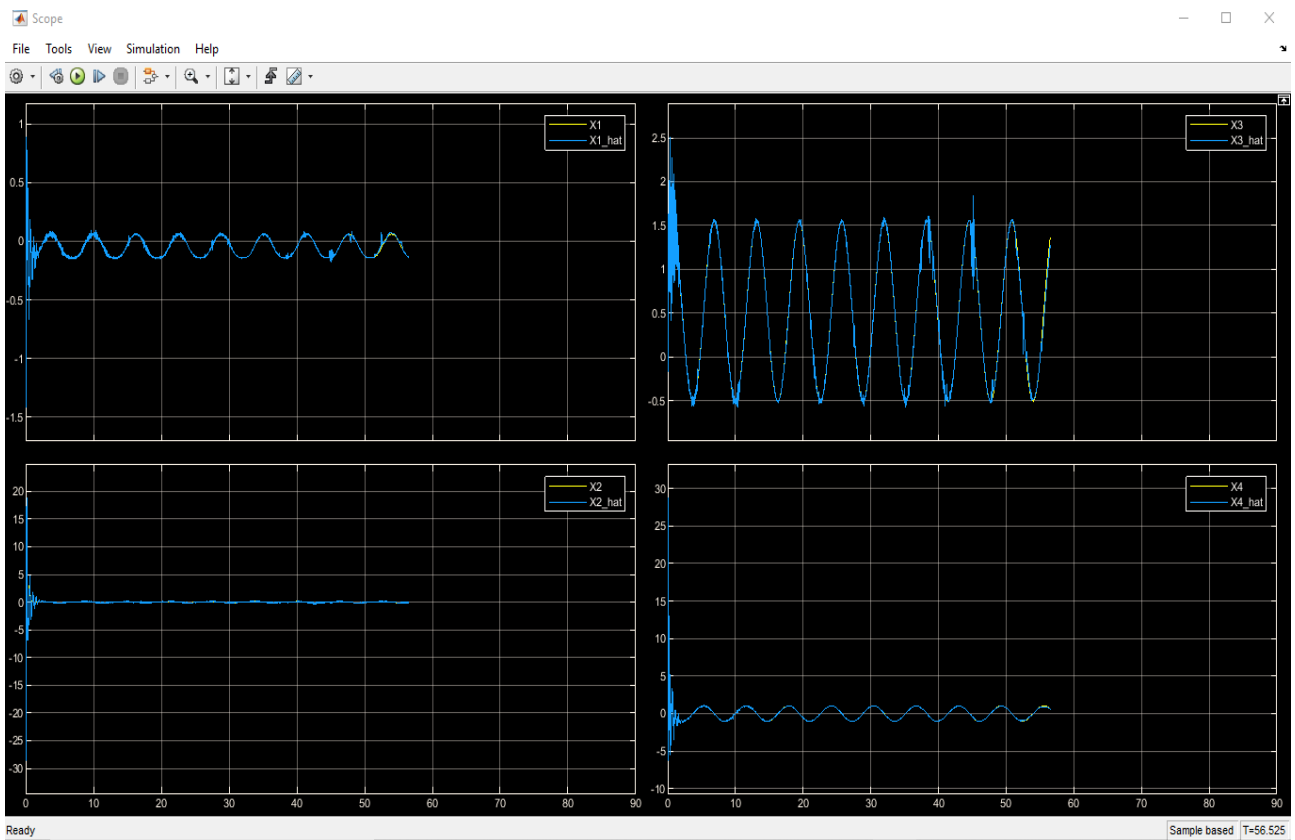


Figure 23. Visualization of the estimated states for the signal $\sin(t + \pi/3) + 0.5$ after adjustment of the number of hidden layers.

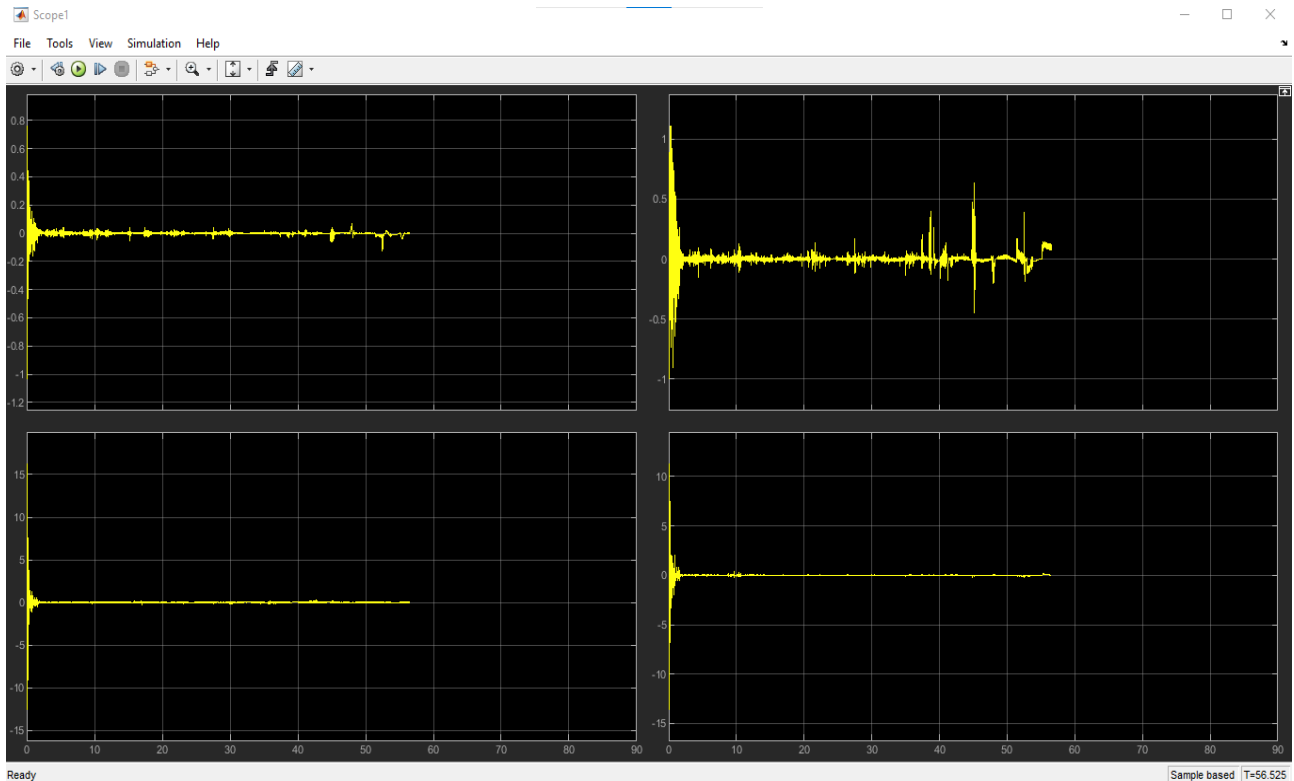


Figure 24. Visualization of the estimated errors for the signal $\sin(t + \pi/3) + 0.5$ after adjustment of the number of hidden layers.

also be used for noisy signals. The most important thing is to adjust the hidden layers of our neural network. The approximation of non-linearities is better when the number of hidden layers increases. Indeed, it is these layers that are responsible for the approximation. We also see that the error is reduced and converges to 0.

5. Conclusions

This work proposes a new approach for the design of state observers for MIMO systems. In this work we focus on the nature of the neural network used. The stability of our observer is also demonstrated experimentally. The key to the design of our observer lies in the radial basis function neural network used. Abdollahi *et al.* in [2] proposed a neuro-adaptive observer based on recurrent neural networks which have the following limitations:

- Recurrent neural networks are difficult to train and are most often subject to gradient fading and explosion;
- The computation time of the hidden layers is long.

On the other hand, the RBFNNs we use to design our observer are very good universal approximators, they are easy to implement and train, and the computational time of the hidden layers is relatively small compared to the RNNs used by Abdollahi *et al.* [2]. We have face some difficulties when doing the present work and we can cite:

- The performance of the machine used: neural networks are quite resource

intensive and the machine we had was not powerful enough to perform the approximation once we increased the number of hidden layers;

- In this work, we have not explicitly modelled the uncertainties associated with a MIMO system mathematically, we have assumed that such a system is subject to uncertainties.

For Future steps we can include

- Make a comparative study of state observers using other types of neural networks such as CNNs (Convolutional Neural Networks), auto-encoders and GANs (Generative Adversarial Networks);
- Explore other stability study techniques such as LMI (Linear Matrix Inequality);
- Mathematically model the dynamics of MIMO uncertainty;
- Explore alternative learning methods for neural networks such as genetic algorithms;

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Souad, A. (2015) Observateurs d'état pour les systèmes dynamiques non linéaires. Master's Thesis, Université Mouloud Mammeri, Tizi-Ouzou, 67 p.
- [2] Abdollahi, F., Talebi, H.A. and Patel, R.V. (2006) A Stable Neural Network-Based Observer with Application to Flexible-Joint Manipulators. *IEEE Transactions on Neural Networks*, **17**, 118-129. <https://doi.org/10.1109/TNN.2005.863458>
- [3] Rajchakit, G., Agarwal, P. and Ramalingam, S. (2021) Stability Analysis of Neural Networks. Springer Singapore, Singapore. <https://doi.org/10.1007/978-981-16-6534-9>
- [4] Pac, J.-L. (2016) Systèmes dynamiques. 2nd Edition, Dunod, Paris.
- [5] Hovakimyan, N., Calise, A.J. and Madyastha, V.K. (2002) An Adaptive Observer Design Methodology for Bounded Nonlinear Processes. *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, 10-13 December 2002, 4700-4705.
- [6] Aggarwal, C.C. (2018) Neural Networks and Deep Learning: A Textbook. Springer International Publishing, Cham. <https://doi.org/10.1007/978-3-319-94463-0>
- [7] Ahmed, M.S. and Riyaz, S.H. (2000) Dynamic Observer—A Neural Net Approach. *Journal of Intelligent and Fuzzy Systems*, **9**, 113-127.
- [8] Pozynak, A.S., Sanchez, E.N., Palma, O. and Yu, W. (2000) Output Trajectory Tracking Using Dynamic Neural Networks. *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, 12-15 December 2000, 949-954.
- [9] Vargas, J.A.R. and Hemerly, E.M. (1999) Robust Neural Adaptive Observer for MIMO Non Linear Systems. *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, 12-15 October 1999, 1084-1089.
- [10] Alessandri, A., Cervellera, C., Grassia, F. and Sanguineti, M. (2004) Design of Observers for Continuous-Time Non Linear Systems Using Neural Networks. *Pro-*

ceedings of the 2004 American Control Conference, Boston, 30 June-2 July 2004, 2433-2438. <https://doi.org/10.23919/ACC.2004.1383829>

- [11] Talebi, H.A., Patel, R.V. and Wong, M. (2002) A Neural Network Based Observer for Flexible-Joint Manipulators. *IFAC Proceedings Volumes*, **35**, 317-322. <https://doi.org/10.3182/20020721-6-ES-1901.00865>
- [12] Abdollahi, F., Talebi, H.A. and Patel, R.V. (2002) A Stable Neural Network Observer with Application to Flexible-Joint Manipulators. *IEEE Transactions on Neural Networks*, **17**, 118-129.