

Research Article

FPGA Realization and Lyapunov–Krasovskii Analysis for a Master-Slave Synchronization Scheme Involving Chaotic Systems and Time-Delay Neural Networks

J. Perez-Padron , C. Posadas-Castillo , J. Paz-Perez , E. Zambrano-Serrano ,
and M. A. Platas-Garza 

Universidad Autónoma de Nuevo León, Av. Pedro de Alba S/N, Cd. Universitaria, San Nicolás de los Garza
C.P. 66455, NL, Mexico

Correspondence should be addressed to M. A. Platas-Garza; miguel.platasgrz@uanl.edu.mx

Received 17 July 2021; Revised 5 September 2021; Accepted 8 September 2021; Published 23 September 2021

Academic Editor: Adel Ouannas

Copyright © 2021 J. Perez-Padron 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.

In this paper, the trajectory tracking control and the field programmable gate array (FPGA) implementation between a recurrent neural network with time delay and a chaotic system are presented. The tracking error is globally asymptotically stabilized by means of a control law generated from the Lyapunov–Krasovskii and Lur’e theory. The applicability of the approach is illustrated by considering two different chaotic systems: Liu chaotic system and Genesis–Tesi chaotic system. The numerical results have shown the effectiveness of obtained theoretical results. Finally, the theoretical results are implemented on an FPGA, confirming the feasibility of the synchronization scheme and showing that it is hardware realizable.

1. Introduction

Neural networks can be considered as nonlinear mathematical functions whose parameters are adjusted to describe the behavior of a static or dynamic system [1]. According to their structure, neural networks can be classified as static neural networks or dynamic neural networks. The static neural networks are capable of approximating any function using static mapping [2]. Meanwhile, dynamic neural networks, also called recurrent neural networks, have feedback connections that give them higher capability than static neural networks. For example, recurrent neural networks can reproduce the dynamic response of a dynamic system due to their feedback connection. They can also overcome many problems related to static neural networks, such as extreme global search, and consequently, have better approximation properties [3]. Recently, due to the richness of dynamic behaviors from recurrent neural networks, they have received much attention and been investigated in a wide variety of applications in diverse fields such as in associative memory [4], neurodynamic optimization problems

[5], pattern recognition [6], image processing [7], and so on [8, 9].

It is well known that the time-delay phenomenon is an inherent feature of many physical processes such as nuclear reactors, chemical processes, and biological systems. Usually, it is considered a source of oscillation and divergence, leading to system performance degradation or even generating instability [9]. A delayed neural network can generate complicated dynamic behaviors such as boundedness, periodicity, stability, chaos, and synchronization [10]. Synchronization is a fundamental behavior of recurrent neural networks. Hence, many researchers devote their effort to studying the delay effect on drive-response synchronization. Indeed, the drive-response synchronization of recurrent neural networks cannot be achieved individually, so external input signals should be considered in the neural network response. Many efficient control approaches have been designed to control or achieve the synchronized state in the literature, such as linear feedback control, active control, intermittent control, adaptive control, event-triggered control, pinning control, and impulsive control, among

others [11–13]. In particular, the Lyapunov–Krasovskii approach has demonstrated to be an efficient method to deal with the global asymptotic stability of recurrent neural networks with time delay. In [14], the global synchronization of multiple recurrent neural networks with time delays under a switching topology via impulsive interaction is given. An exponential H_∞ synchronization method for an uncertain drive and response neural networks with mixed time delays was presented based on a Lyapunov–Krasovskii functional and free-weighting matrices [15]. The synchronization of delayed neural networks with hybrid couplings by employing the Lyapunov functional method and the Kronecker product technique is used to guarantee the global synchronization in coupled networks [16]. In [17], the global asymptotic synchronization for a class of delayed neural networks is investigated. In [18], the pinning synchronization in an array of coupled delayed neural networks with both constant and delayed couplings is presented, showing that the network can be pinned to a homogenous state by applying adaptive feedback control. It is worth noting that the Lyapunov–Krasovskii approach has been demonstrated to be an efficient method to deal with the global asymptotic stability of a recurrent neural network with time delay [19, 20]. In [21], the stability of linear continuous-time systems with time delay by employing new Lyapunov–Krasovskii functionals is presented. New Lyapunov–Krasovskii functionals are proposed to achieve the synchronization behavior of delay neural networks with two time scales for the fixed and adaptive coupling schemes [22]. Some new Lyapunov–Krasovskii functionals are developed by nonuniformly dividing the delay interval into multiple segments and choosing proper functionals with different weighting matrices corresponding to different sections in the Lyapunov–Krasovskii functionals proposed in [23]. In [24], the global stability of a system composed by identical Hopfield neural networks with time-delayed connections is presented by constructing a Lyapunov–Krasovskii functional.

Inspired by all the above works, this paper focuses on the problem of tracking a given reference trajectory. It proposes a trajectory tracking scheme in which a time-delay neural network is forced to follow a chaotic system’s reference signal, achieving a drive-response synchronization state between the neural network and the chaotic system. The control law that guarantees the trajectory tracking problem solution is obtained by using the Lyapunov–Krasovskii and Lur’e approach. The pertinence of the approach is demonstrated by considering two numerical examples. In each example, synchronization with one chaotic oscillator is addressed, and the derived control laws are tested via numerical experimentation. The Genesis–Tesi and Liu systems are considered. Although the proposed scheme could be useful to follow other nonlinear systems, we decide to restrict this work using only these two chaotic systems because their tracking can be more challenging than other tracking problems such as the reproduction of the gait cycle in humanoids [25] or the trajectory tracking in different mechanical systems [26] and complex systems [27]. In all the former cases, the movement may be softer and periodic.

Additionally, the FPGA technology is used to implement the proposed scheme by using a NI c-RIO-9068 device with a Xilinx Zynq-7000 XC7Z020 FPGA. The hardware implementation results confirm the feasibility of the proposed scheme. We decided to implement the entire scheme on the same chip, i.e., the system to be tracked, the recurrent neural network with time delay, and the derived control law are implemented on the same FPGA chip.

This article is organized as follows. In Section 2, the construction of the mathematical model is given. The control law of delayed neural network and global asymptotic stability conditions are shown in Section 3. The numerical results showing the applicability of the theory are given in Section 4. Section 5 provides experimental results based on FPGA implementations. Finally, we conclude the paper in the last section.

2. Preliminaries

Consider the following time-delay neural network model:

$$\dot{x}(t) = Ax(t) + W\phi[x(t - \tau)] + u(t), \quad (1)$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in \mathbb{R}^n$ is the state vector associated with the neurons; $A = \text{diag}(-\lambda_1, -\lambda_2, \dots, -\lambda_n)$ with λ_i with $i = 1, \dots, n$ being positive constants representing the rate with at which the neural network will reset its potential to the resting state when disconnected from the external input; $W = [w_{ij}] \in \mathbb{R}^n \times \mathbb{R}^n$ represents the delayed connection weight matrix; $\phi(x) = [\phi_1(x_1), \phi_2(x_2), \dots, \phi_n(x_n)]^T \in \mathbb{R}^n$ is the activation function; $\tau > 0$ is the transmission delay; and $u(t) = (u_1(t), u_2(t), \dots, u_n(t))^T \in \mathbb{R}^n$ is an external input vector.

The following properties are assumed for the activation function [28, 29]:

- (1) $\phi(x)$ is bounded and monotonically nondecreasing on \mathbb{R} .
- (2) The activation function $\phi(x)$ is Lipschitz continuous, that is, there exists constant $L_\phi > 0$ such that

$$|\phi(x) - \phi(y)| \leq L_\phi |x - y|, \quad \forall x, y \in \mathbb{R}. \quad (2)$$

The system to be tracked by the recurrent neural network with time delay (driving system) is defined as an autonomous n -dimensional dynamical system given by

$$\dot{x}_r(t) = f(x_r(t)), \quad (3)$$

where $x_r(t) = [x_{r_1}(t), x_{r_2}(t), \dots, x_{r_n}(t)]^T \in \mathbb{R}^n$ is a n -dimensional state vector, with $f(x_r(t))$ defining a vector field $f(x_r(t)): \mathbb{R}^n \rightarrow \mathbb{R}^n$. In this paper, (3) is a nonlinear system that generates chaotic behavior.

We continue by presenting the Lyapunov–Krasovskii approach. This method is the natural extension of the second Lyapunov approach associated with the stability analysis of functional differential equations [30]. It consists of considering functionals of the form $V(t; x_t)$ that are positive definite and decreasing along the trajectories of system

$\dot{x}(t) = f(t, x(t), x_t)$. The Lyapunov–Krasovskii theorem is expressed as follows.

Theorem 1 (see [30]). *Let u, v , and $w: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be nondecreasing functions such that $u(\theta)$ and $v(\theta)$ are strictly positive for all $\theta > 0$. Let $h \in \mathbb{R}^+$ and $\beta: \mathcal{C}[-h; 0] \rightarrow \mathbb{R}^n$. Assume that the vector field $f(\cdot)$ of $\dot{x}(t) = f(t, x(t), x_t)$ is bounded for bounded values of its arguments. If there exists a continuous and differentiable functional $V: \mathbb{R} \times \mathcal{C}[-h; 0] \rightarrow \mathbb{R}^+$ such that*

$$\begin{aligned} (a) \quad & u(\|\beta(0)\|) \leq V(t, \beta) \leq v(\|\beta\|) \\ (b) \quad & \dot{V}(t, \beta) \leq -w(\|\beta(0)\|) \quad \text{for all trajectories of} \\ & x(t) = f(t, x(t), x_t) \end{aligned}$$

$$V(t + \theta, \beta(t + \theta)) \leq V(t, \beta(t)), \quad \forall \theta \in [-h, 0], \quad (4)$$

then the solution $x_t = 0$ is uniformly stable for $\dot{x}(t) = f(t, x(t), x_t)$.

Finally, we add two definitions that will be used to analyze equilibrium points.

Definition 1. An equilibrium E_* is called a saddle point of index 1 if E_* has one eigenvalue with nonnegative real part (unstable).

Definition 2. An equilibrium E_* is called a saddle point of index 2 if E_* has two unstable eigenvalues.

3. Trajectory Tracking for a Recurrent Neural Network with Time Delay

While the main objective of the work is to present the proposed tracking scheme and its hardware implementation, the main objective of this section is to develop a control law such that the delayed neural network (1) tracks the solution of the dynamical system (3).

3.1. Dynamical Analysis Error. Define the tracking error as $e(t) = x(t) - x_r(t)$, where the respective error dynamics are

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_r(t). \quad (5)$$

Substituting (1) and (3) in (5), the dynamics of the error are governed by the following equation:

$$\dot{e}(t) = Ax(t) + W\phi[x(t - \tau)] + u(t) - f(x_r(t)). \quad (6)$$

Adding and subtracting to (6) the terms $Ax_r(t)$, $W\phi[x_r(t - \tau)]$, and $\alpha(t)$, we have

$$\begin{aligned} \dot{e}(t) = & Ae(t) + W(\phi[x(t - \tau)] - \phi[x_r(t - \tau)]) \\ & + (u(t) - \alpha(t) + [Ax_r(t) + W\phi[x_r(t - \tau)] + \alpha(t)] \\ & - f(x_r(t)), \end{aligned} \quad (7)$$

where $\alpha(t)$ is the function to be determined. In order to guarantee that the neural network given in (1) can track the

reference trajectory of dynamical system (3), the following assumption has to be satisfied [31].

There exist functions $\rho(t)$ and $\alpha(t)$ such that

$$\begin{aligned} \dot{\rho}(t) &= A\rho(t) + W\phi[\rho(t - \tau)] + \alpha(t), \\ \rho(t) &= x_r(t). \end{aligned} \quad (8)$$

Therefore, from (3) and (8), we obtain that

$$Ax_r(t) + W\phi[x_r(t - \tau)] + \alpha(t) = f(x_r(t)), \quad (9)$$

and then

$$\alpha(t) = f(x_r(t)) - Ax_r(t) - W\phi[x_r(t - \tau)], \quad (10)$$

so that (7) becomes

$$\dot{e}(t) = Ae(t) + W\varphi_\phi(t - \tau) + \bar{u}(t), \quad (11)$$

with $W\varphi_\phi(t - \tau) = W(\phi[x(t - \tau)] - \phi[x_r(t - \tau)])$ and $\bar{u}(t) = (u(t) - \alpha(t))$. It is clear that $e(t) = 0$ is an equilibrium point of (11) when $\bar{u}(t) = 0$. In this way, the tracking problem can be expressed as a global asymptotic stabilization problem for system (11).

3.2. Error Stabilization and Control Design. In order to establish the convergence of (11) to $e(t) = 0$, a Lyapunov–Krasovskii functional is proposed ensuring the desired tracking. The Lyapunov–Krasovskii analysis is a Lyapunov-inspired method that consists of proposing a functional $V(t, x(t))$ of the state $x(t)$, which should be positive and decreasing definite along the trajectories of the system. This is essential for the design of a globally and asymptotically stabilizing control law for time-delay systems [20]. In this work, the following Lyapunov–Krasovskii functional is proposed as in [19, 32]:

$$V(e) = \sum_{i=1}^n \int_0^{e_i} \varphi(\xi, x_r) d\xi + \int_{t-\tau}^t (\varphi_\phi^T(s) W^T W \varphi_\phi(s)) ds, \quad (12)$$

where the expression $\sum_{i=1}^n \int_0^{e_i} \varphi(\xi, x_r) d\xi$ is proposed to analyze the stability related to error $\dot{e}(t)$, whereas the expression $\int_{t-\tau}^t (\varphi_\phi^T(s) W^T W \varphi_\phi(s)) ds$ is envisioned to analyze the effect of the delay τ ; note that this term vanishes as $\tau \rightarrow 0$.

In this manner, the time derivate of (12), along the trajectories of (11), was computed using the fundamental theorem of calculus as follows:

$$\begin{aligned} \dot{V}(e) = & \frac{\partial V(e)}{\partial e} \dot{e} = \varphi(e, x_r)^T \dot{e} + \varphi_\phi^T W^T W \varphi_\phi \\ & - \varphi_\phi^T(t - \tau) W^T W \varphi_\phi(t - \tau). \end{aligned} \quad (13)$$

Substituting (11) in (13), we get

$$\begin{aligned} \dot{V}(e) = & \varphi(e, x_r)^T Ae + \varphi(e, x_r)^T W(\phi[x(t - \tau)] \\ & - \phi[x_r(t - \tau)]) + \varphi(e, x_r)^T \bar{u} + \varphi_\phi^T W^T W \varphi_\phi \\ & - \varphi_\phi^T(t - \tau) W^T W \varphi_\phi(t - \tau). \end{aligned} \quad (14)$$

Regarding $\varphi_\phi(t - \tau) = (\phi[x(t - \tau)] - \phi[x_r(t - \tau)])$, equation (14) can be denoted as

$$\begin{aligned} \dot{V}(e) = & -\lambda\varphi(e, x_r)^T e + \varphi(e, x_r)^T W\varphi_\phi(t - \tau) + \varphi(e, x_r)^T \tilde{u} \\ & + \varphi_\phi^T W^T W\varphi_\phi - \varphi_\phi^T(t - \tau)W^T W\varphi_\phi(t - \tau). \end{aligned} \quad (15)$$

Consider the inequality [19]:

$$P^T Q + Q^T P \leq P^T \Lambda P + Q^T \Lambda^{-1} Q, \quad (16)$$

which holds for all matrices $P, Q \in \mathbb{R}^{n \times k}$ and a positive definite matrix $\Lambda \in \mathbb{R}^{n \times n}$. To prove the inequality (16), we consider $F \equiv P^T \Lambda P + Q^T \Lambda^{-1} Q - P^T Q - Q^T P \geq 0$. Denoting $v_1 = \Lambda^{(1/2)} P v$ and $v_2 = \Lambda^{-(1/2)} Q v$, we obtain $F = v_1^T v_1 - v_2^T v_2 - 2v_1^T v_2 = \|v_1 - v_2\|^2 \geq 0$.

Applying (16) with $\Lambda = I$ to the term $\varphi(e, x_r)^T W\varphi_\phi(t - \tau)$, we get

$$\begin{aligned} \dot{V}(e) \leq & -\lambda\varphi(e, x_r)^T e + \frac{1}{2}\varphi(e, x_r)^T \varphi(e, x_r) \\ & + \frac{1}{2}\varphi_\phi^T(t - \tau)W^T W\varphi_\phi(t - \tau) + \varphi(e, x_r)^T \tilde{u} \\ & + \varphi_\phi^T W^T W\varphi_\phi - \varphi_\phi^T(t - \tau)W^T W\varphi_\phi(t - \tau). \end{aligned} \quad (17)$$

Simplifying (17), the following is obtained:

$$\begin{aligned} \dot{V}(e) \leq & -\lambda\varphi(e, x_r)^T e + \frac{1}{2}\varphi(e, x_r)^T \varphi(e, x_r) \\ & - \frac{1}{2}\varphi_\phi^T(t - \tau)W^T W\varphi_\phi(t - \tau) + \varphi(e, x_r)^T \tilde{u} \\ & + \varphi_\phi^T W^T W\varphi_\phi. \end{aligned} \quad (18)$$

Because $\varphi(e, x_r)$ is a sector function with respect to $e(t)$, there exist positive constants k_1 and k_2 such that $k_1 \|e\|^2 \leq \varphi(e, x_r)^T \varphi(e, x_r) \leq k_2 \|e\|^2$, where $\|\cdot\|$ is used to denote the Euclidean norm. From $\varphi_\phi(t) = (\phi(x(t)) - \phi(x_r(t)))$,

$$\begin{aligned} \|\varphi_\phi(t)\| & \|\phi(x(t)) - \phi(x_r(t))\| \\ & \leq L_\phi \|x(t) - x_r(t)\| \\ & = L_\phi \|e(t)\|. \end{aligned} \quad (19)$$

Then, there exists a positive constant L_ϕ such that $\varphi(e, x_r)^T \varphi(e, x_r) \leq L_\phi^2 \|e\|^2$. Henceforth, (18) can be written as follows:

$$\begin{aligned} \dot{V}(e) \leq & -\left[\lambda k_1 - \frac{1}{2}L_\phi^2\right] \|e\|^2 - \frac{1}{2}\varphi_\phi^T(t - \tau)W^T W\varphi_\phi(t - \tau) \\ & + \varphi(e, x_r)^T \tilde{u} + \varphi_\phi^T W^T W\varphi_\phi. \end{aligned} \quad (20)$$

Simplifying (20), we have

$$\dot{V}(e) \leq -\left[\lambda k_1 - \frac{1}{2}L_\phi^2\right] \|e\|^2 + \varphi(e, x_r)^T \tilde{u} + \varphi_\phi^T W^T W\varphi_\phi. \quad (21)$$

Since φ_ϕ is Lipschitz, we consider (20) applied to $\varphi_\phi^T W^T W\varphi_\phi$, obtaining

$$\varphi_\phi^T W^T W\varphi_\phi \leq \|\varphi_\phi^T W^T W\varphi_\phi\| \leq L_\phi^2 \|W\|^2 \|e\|^2, \quad (22)$$

with L_ϕ^2 being the Lipschitz constant of $\phi(\cdot)$.

Replacing (22) in equation (21), the following expression is obtained:

$$\dot{V}(e) \leq -\left[\lambda k_1 - \frac{1}{2}L_\phi^2\right] \|e\|^2 + L_\phi^2 \|W\|^2 \|e\|^2 + \varphi(e, x_r)^T \tilde{u}. \quad (23)$$

Now we consider the control law given as follows:

$$\tilde{u} = -\left(2 + 2L_\phi^2 \|W\|^2\right) \varphi(e, x_r). \quad (24)$$

Then, substituting (24) in (23), we obtain

$$\dot{V}(e) \leq -\left[\lambda + L_\phi^2 + L_\phi^2 \|W\|^2\right] \|e\|^2. \quad (25)$$

Then, $\dot{V}(e) < 0$ for all $e \neq 0$. This means that the proposed control law can globally and asymptotically stabilize the error system, thus ensuring the tracking of (1) by (3) achieving the synchronization, which means that $\lim_{t \rightarrow \infty} \|x(t) - x_r(t)\| = 0$ holds; moreover, this implies that the both systems have reached the synchrony state (synchronization).

Finally, the control action driving the recurrent neural network is given by

$$u = -\left(2 + 2L_\phi^2 \|W\|^2\right) \varphi(e, x_r) + f(x_r) - Ax_r - W\phi[x_r(t - \tau)]. \quad (26)$$

We summarize the developed analysis in the following theorem.

Theorem 2. *The control law defined as u in (26) ensures that the time-delay neural network established in (1) tracks the dynamical behavior generated by the reference system (chaotic system) (3).*

4. Numerical Simulations

In this section, computer simulations are presented to confirm the applicability of the results proposed previously, which are applied to two dynamical systems that generate chaotic behavior.

4.1. Synchronization between Delayed Neural Network and Genesio-Tesi System. In 1992, Genesio and Tesi proposed a chaotic system known as the Genesio-Tesi system [33]. It is described by the following simple three-dimensional autonomous system with only one quadratic nonlinear term:

$$\begin{aligned} \dot{x}_{r_1} &= x_{r_2}, \\ \dot{x}_{r_2} &= x_{r_3}, \\ \dot{x}_{r_3} &= -a_1 x_{r_1} - a_2 x_{r_2} - a_3 x_{r_3} + a_4 x_{r_1}^2, \end{aligned} \quad (27)$$

where $x_r = [x_{r_1}, x_{r_2}, x_{r_3}] \in \mathbb{R}^3$ is the vector state and a_i with $(i = 1, \dots, 4)$ are positive real parameters. System (27) has two equilibrium points $E_1(0, 0, 0)$ and $E_2(a_1/a_4, 0, 0)$. The parameters and initial conditions of (27) are chosen as $a_1 = 1$, $a_2 = 1.1$, $a_3 = 0.44$, $a_4 = 1$, and $[x_{r_1}(0), x_{r_2}(0), x_{r_3}(0)]^T = [0.1, 0.1, 0.1]^T$. The corresponding eigenvalues of equilibria E_1 , E_2 and their stability are described in Table 1. The Lyapunov exponents for system (27) are $LE_1 = 0.0818$, $LE_2 = 0$, $LE_3 = -0.5227$ which according to [34] make the system chaotic. The algorithm

employed for determining Lyapunov exponents was proposed in [35]. Its Kaplan–Yorke dimension is $D_{KY} = 2.1564$, and the phase plane of the chaotic system is displayed in Figure 1. In addition, the bifurcation diagrams of system (27) in terms of the parameters a_1 and a_3 are plotted in Figures 2(a) and 2(b), respectively. Moreover, the bifurcation diagrams are obtained by plotting the local maximal of the state variable x_{r_1} denoted as \hat{x} in Figures 2(a) and 2(b).

By considering the following delayed neural network defined as in equation (1), we get

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.3 & \frac{4}{5} & 0 \\ 0.3 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tanh(x_1(t-\tau)) \\ \tanh(x_2(t-\tau)) \\ \tanh(x_2(t-\tau)) \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad (28)$$

where $u = [u_1, u_2, u_3]^T$ is the control action applied to the recurrent neural network defined as in (24) and defined by

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = -\psi \begin{bmatrix} x_1 - x_{r_1} \\ x_2 - x_{r_2} \\ x_3 - x_{r_3} \end{bmatrix} + \begin{bmatrix} x_{r_2} \\ x_{r_3} \\ -a_1 x_{r_1} - a_2 x_{r_2} - a_3 x_{r_3} + a_4 x_{r_1}^2 \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_{r_1} \\ x_{r_2} \\ x_{r_3} \end{bmatrix} - \begin{bmatrix} 0.3 & \frac{4}{5} & 0 \\ \frac{2}{5} & 0.3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tanh(x_{r_1}(t-\tau)) \\ \tanh(x_{r_2}(t-\tau)) \\ \tanh(x_{r_3}(t-\tau)) \end{bmatrix}, \quad (29)$$

where ψ is a real constant defined as follows:

$$\psi = 2 + 2 \left[\sqrt{\sum_{i=1}^n \sum_{j=1}^n w_{ij}^2} \right]^2. \quad (30)$$

The initial values of the delayed neural network are taken as $(2, -1, -2)$; also, we consider a delay $\tau = 15$ s. In Figure 3, the results are shown.

Figure 3(a) shows the time evolution of the state variables x_r (purple line) and x (olive line) that corresponds to the chaotic system and delayed neural network, respectively, and we observe that both solutions evolve independently until $\tau = 15$ s, that is, when the control law begins to activate. In Figure 3(b), the error between the state variables x_{r_i} and x_i with $i = 1, 2, 3$ are given, respectively.

4.2. Synchronization between Delayed Neural Network and Liu System. Now, in this instance, we demonstrate the applicability of the discussed results considering the delayed neural network (1) described as in the previous example, i.e., equation (28), where the reference model to be tracked is the Liu system.

Liu et al. introduced in 2009 a three-dimensional autonomous chaotic system which is based on two multipliers and one quadratic term to introduce the nonlinearity necessary for folding trajectories [36]. The Liu chaotic oscillator is defined as follows:

$$\begin{aligned} \dot{x}_{r_1} &= -ax_{r_1} - m_1 x_{r_2}^2, \\ \dot{x}_{r_2} &= bx_{r_2} - m_2 x_{r_1} x_{r_3}, \\ \dot{x}_{r_3} &= -cx_{r_3} + m_3 x_{r_1} x_{r_2}, \end{aligned} \quad (31)$$

where $a, b, c, m_1, m_2, m_3 \in \mathbb{R}$. Its parameters are chosen as $a = m_1 = 1$, $b = 2.5$, $c = 5$, and $m_2 = m_3 = 4$. The equilibrium points, eigenvalues, and stability of the system are represented in Table 1. By considering the following initial conditions $[x_{r_1}(0), x_{r_2}(0), x_{r_3}(0)]^T = [-1, -0.5, 0.5]^T$, the Lyapunov exponents are $LE_1 = 0.4155$, $LE_2 = 0$, $LE_3 = -3.8586$, which according to [34] confirm that the Liu system is a chaotic oscillator. The algorithm employed for determining Lyapunov exponents was proposed in [35]. Its Kaplan–Yorke dimension is $D_{KY} = 2.1077$. The phase planes of Liu chaotic oscillator are displayed in Figure 4. In addition, the bifurcation diagrams of system (31) with the parameters b and c are displayed in Figures 5(a) and 5(b), respectively. Moreover, the bifurcation diagrams are obtained by plotting the local maximum of the state variable x_{r_1} denoted as \hat{x} in Figures 5(a) and 5(b).

TABLE 1: Stability of equilibrium points from Genesio–Tesi and Liu chaotic systems.

Chaotic system	Equilibria	Eigenvalues	Stability
Genesio–Tesi	$E_1 = (0, 0, 0)$	$\lambda_1 = -0.7503$ $\lambda_{2,3} = 0.1551 \pm 1.1440j$	Saddle point of index 2
	$E_2 = (1, 0, 0)$	$\lambda_1 = 0.5871$ $\lambda_{2,3} = -0.5136 \pm 1.1997j$	Saddle point of index 1
Liu	$E_1 = (0, 0, 0)$	$\lambda_1 = 2.5$ $\lambda_2 = -1; \lambda_3 = -5$	Saddle point
	$E_2 = (-0.8838, -0.9401, 0.6647)$	$\lambda_1 = -4.3878$ $\lambda_{2,3} = 0.4439 \pm 3.3464j$	Saddle point of index 2
	$E_3 = (-0.8838, 0.9401, -0.6647)$	$\lambda_1 = -4.3878$ $\lambda_{2,3} = 0.4439 \pm 3.3464j$	Saddle point of index 2

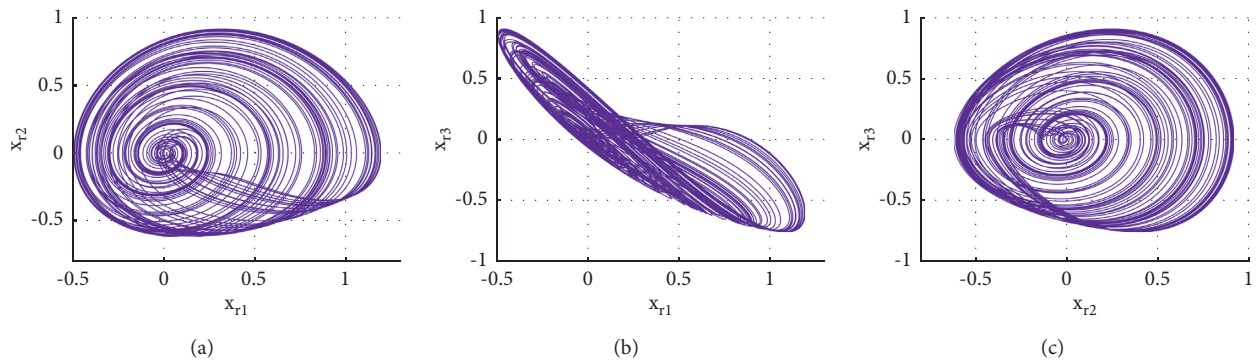


FIGURE 1: The Genesio–Tesi chaotic attractor in the phase plane with initial conditions $[x_{r_1}(0), x_{r_2}(0), x_{r_3}(0)]^T = [0.1, 0.1, 0.1]^T$. (a) $x_{r_1} - x_{r_2}$ phase plane. (b) $x_{r_1} - x_{r_3}$ phase plane. (c) $x_{r_2} - x_{r_3}$ phase plane.

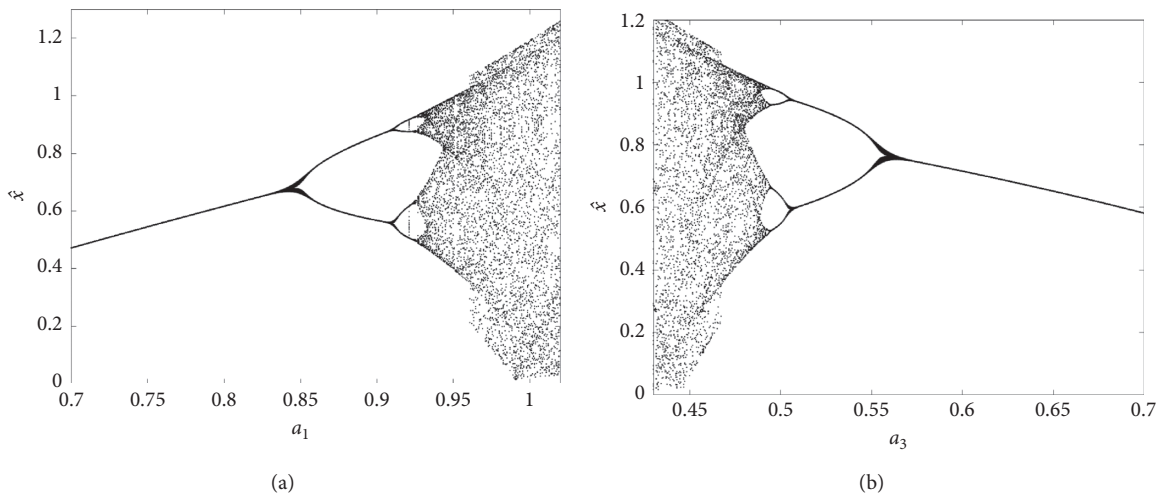


FIGURE 2: (a) Bifurcation diagram of Genesio–Tesi system with varying parameter a_1 . (b) Bifurcation diagram of Genesio–Tesi system with varying parameter a_3 .

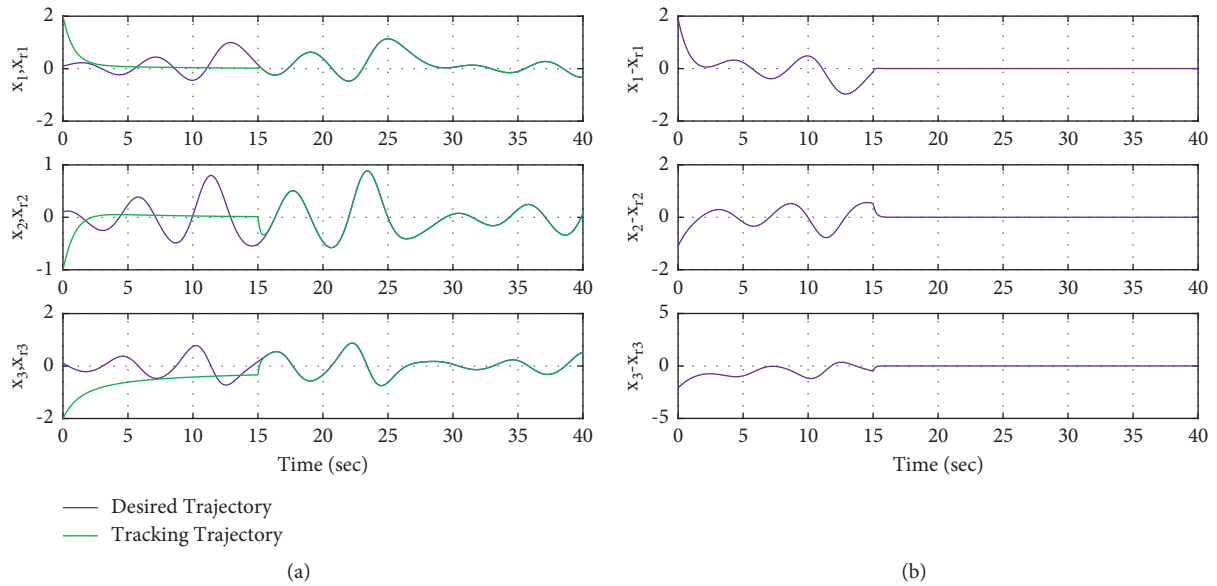


FIGURE 3: (a) Time evolution for the Genesio–Tesi reference states x_{r1} , x_{r2} , and x_{r3} with initial conditions $(0.1, 0.1, 0.1)$ (purple line) and time evolution for the neural network states x_1 , x_2 , and x_3 with initial conditions $(2, -1, -2)$ (green line). (b) Error between states $x_i - x_{r_i}$ with $i = 1, 2, 3$.

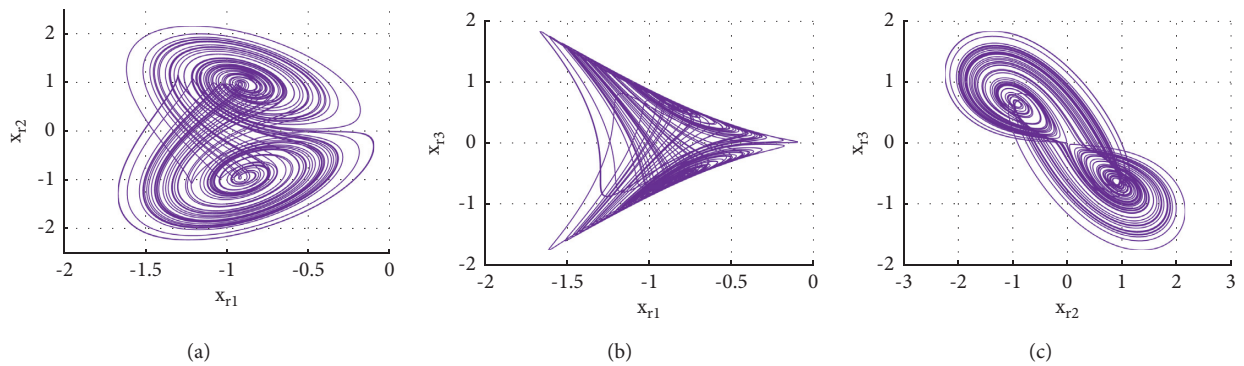


FIGURE 4: The Liu chaotic oscillator in the phase plane with initial conditions $[x_{r1}(0), x_{r2}(0), x_{r3}(0)]^T = [-1, -0.5, 0.5]^T$. (a) $x_{r1} - x_{r2}$ phase plane. (b) $x_{r1} - x_{r3}$ phase plane. (c) $x_{r2} - x_{r3}$ phase plane.

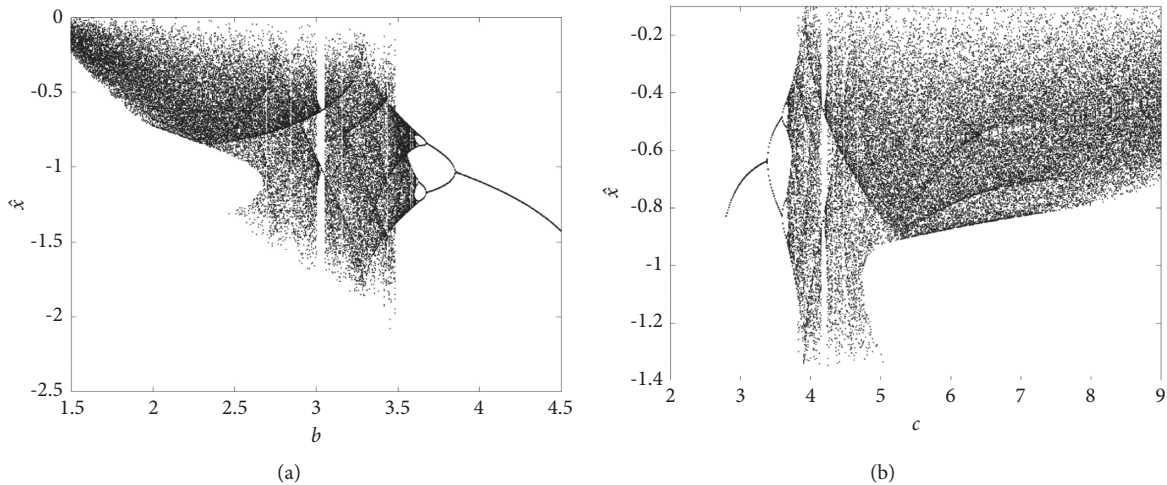


FIGURE 5: (a) Bifurcation diagram of Liu system with varying parameter b . (b) Bifurcation diagram of Liu system with varying parameter c .

The initial values of the delayed neural network are taken as $(0.1, 0.2, -0.1)$; also, we consider a delay $\tau = 15$ s. Simulation results are presented in Figure 6.

Here we consider the control law u defined as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = -\psi \begin{bmatrix} x_1 - x_{r_1} \\ x_2 - x_{r_2} \\ x_3 - x_{r_3} \end{bmatrix} + \begin{bmatrix} -ax_{r_1} - m_1x_{r_2}^2 \\ bx_{r_2} - m_2x_{r_1}x_{r_3} \\ -cx_{r_3} + m_3x_{r_1}x_{r_2} \end{bmatrix}$$

$$- \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_{r_1} \\ x_{r_2} \\ x_{r_3} \end{bmatrix} - \begin{bmatrix} 0.3 & \frac{4}{5} & 0 \\ \frac{2}{5} & 0.3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tanh(x_{r_1}(t - \tau)) \\ \tanh(x_{r_2}(t - \tau)) \\ \tanh(x_{r_3}(t - \tau)) \end{bmatrix}, \quad (32)$$

where ψ is a real constant given as in (30). Furthermore, Figure 6(a) presents the time-evolution of the delayed neural network (28) and the chaotic system (31), where they evolve independently until $\tau = 15$ seconds; at that time, the proposed control (32) law is incepted. Finally, in Figure 6(b), the error between $x_i - x_{r_i}$ states is displayed.

5. FPGA-Based Implementation

The digital implementations of chaotic systems have attracted increasing interest in the last few years [37, 38]. They provide certain advantages in comparison with analog-based systems, like accuracy and possible integration in embedded applications, especially in data encryption and secure communications, which exhibit various practical difficulties like the sensitivity of components to temperature, aging, etc. Recently, the digital-based chaotic implementations are realized considering different numerical circuit-based integrated systems like digital signal processors (DSPs), application specific integrated circuits (ASIC), and FPGA. But the FPGA implementation designs overcome the problems related to AISC and DSP-based applications; this is due to the excellent trade-off between computational power and the performance and reliability which it provides.

In this section, the FPGA realizations for the systems previously shown are addressed. We present two implementations: one of them implements the systems shown in Section 4.1 and the other implements those shown in Section 4.2. Both implementations were done into a Xilinx Zynq-7000 XC7Z020 FPGA chip. The LabVIEW FPGA compiler software and the National Instruments cRIO-9068 hardware were used. cRIO-9068 has attached the FPGA chip and the NI-9264 module. The NI-9264 module allows each design to have six analog outputs from 0 to 10 volts. Both designs use the states of the neural network and the chaotic system as outputs.

5.1. Description of the Designs. All the mathematical operations were done using fixed-point (FXP) arithmetic. FXP was chosen to reduce the hardware cost and achieve higher

speeds. We use 30 bits to represent each state of the neural network and the chaotic system; specifically, we use 7 bits for the integer part and 23 bits for the fractional part. In digital systems, the chaotic behavior may be affected by time discretization or low-resolution FXP arithmetic. In our case, we compute the maximum Lyapunov exponent (MLE) from the FXP sequences related to the implemented Liu and Genesio-Tesi systems by employing TISEAN package software [39]. The obtained values were as follows: MLE = 0.0664 for the Genesio-Tesi system and MLE = 0.0782 for the Liu system. The positive value for each MLE confirms chaotic behavior and validates the FXP resolution and the sampling time used.

5.1.1. Top Level. The top-level description of each implementation is shown in Figure 7, where k represents the sample index. In this work, the top-level description used allows to implement the tracker and the system to be tracked on the same chip, and is composed of the following modules: (a) the chaotic system to be tracked, (b) the delayed neural network, (c) the control law, and (e) the integration method. Note that the block diagram in Figure 7 applies to both designs.

5.1.2. Secondary Level. The dynamics of the chaotic system, which are represented in the dashed block labeled as (a) in Figure 7, are given by the time-discrete version of (27):

$$\begin{aligned} f_1[k] &= x_{r_2}[k], \\ f_2[k] &= x_{r_3}[k], \\ f_3[k] &= -x_{r_1}[k] - 1.1x_{r_2}[k] - 0.44x_{r_3}[k] + x_{r_1}^2[k], \end{aligned} \quad (33)$$

for the Genesio-Tesi case, and by the discrete version of (31):

$$\begin{aligned} f_1[k] &= -x_{r_1}[k] - x_{r_2}^2[k], \\ f_2[k] &= 2.5x_{r_2}[k] - 4x_{r_1}[k]x_{r_3}[k], \\ f_3[k] &= -5x_{r_3}[k] + 4x_{r_1}[k]x_{r_2}[k], \end{aligned} \quad (34)$$

for the Liu case. In both cases, FXP multipliers and adders were used to perform the arithmetic operations involved.

The dynamics of the delayed neural network, which are represented in the dashed block labeled as (b) in Figure 7, are given by the discrete version of (1):

$$\begin{aligned} \eta_1[k] &= -x_1[k] + \sum_{i=1}^3 w_{1,i} \tanh(x_i[k - m]) + u_1[k], \\ \eta_2[k] &= -x_2[k] + \sum_{i=1}^3 w_{2,i} \tanh(x_i[k - m]) + u_2[k], \\ \eta_3[k] &= -x_3[k] + \sum_{i=1}^3 w_{3,i} \tanh(x_i[k - m]) + u_3[k], \end{aligned} \quad (35)$$

where the $\tanh(\cdot)$ function was approximated by its fourth-order Taylor series, and FIFO buffers implemented on the block RAM of the device were used to perform the delay of m samples in the three states. Six FIFO registers were used,

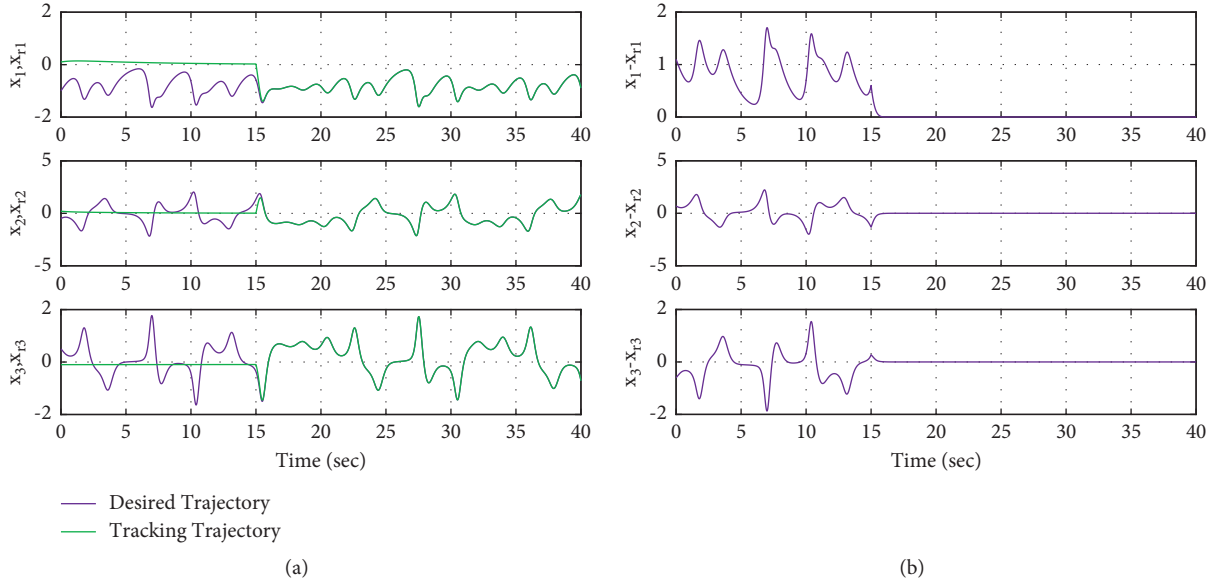


FIGURE 6: (a) Time evolution for the Liu reference states x_{r_1} , x_{r_2} , and x_{r_3} with initial values $(-1, -0.5, 0.5)$ (purple line) and time evolution for the neural network states x_1 , x_2 , and x_3 with initial condition $(0.1, 0.2, -0.1)$ (green line). (b) Error between states $x_i - x_{r_i}$ with $i = 1, 2, 3$.

each one with 16384 elements of 30 bits each. With this configuration, a delay of $m = 16384$ samples is generated in all the states.

The control law $\{u_1[k], u_2[k], u_3[k]\}$ applied to the neural network (35) is represented in the dashed block labeled as (c) in Figure 7. It was implemented as follows:

$$\begin{aligned} u_1[k] &= \psi e_1[k] + f_1[k] - x_{r_1}[k] - \sum_{i=1}^3 w_{1,i} \tanh(x_{r_i}[k - m]), \\ u_2[k] &= \psi e_2[k] + f_2[k] - x_{r_2}[k] - \sum_{i=1}^3 w_{2,i} \tanh(x_{r_i}[k - m]), \\ u_3[k] &= \psi e_3[k] + f_3[k] - x_{r_3}[k] - \sum_{i=1}^3 w_{3,i} \tanh(x_{r_i}[k - m]), \end{aligned} \quad (36)$$

where $e_j[k] = x_j[k] - x_{r_j}[k]$ for $j = 1, 2, 3$ denotes the error sequence.

Finally, shift registers were used to perform the numerical integration. In order to use an integration time of H seconds, a mod- M counter is inferred to derive an enable tick for all the registers each H seconds, where $M = 4560$ ticks. The Euler method is based on the definition of the derivative operator and was selected to perform the numerical integration in order to reduce the complexity of the design due to its maximum simplicity. The block description of this method is presented in the dashed block labeled as (d) in Figure 7, which is described as

$$\begin{aligned} x_{r_i}[k + 1] &= x_{r_i}[k] + H f_i[k], \\ x_i[k + 1] &= x_i[k] + H \eta_i[k], \end{aligned} \quad (37)$$

where $i = 1, 2, 3$.

5.2. Implementation Results. The resources used for each design are presented in Table 2. The total amount of resources employed was obtained from the NI-FPGA compiler report. The minimal amount of ticks $M = 4560$ needed to get a new valid sample was computed using the LabVIEW FPGA tick count function. Its value was the same for both designs addressed. The achieved latencies for both designs are expressed as $L = 114 \mu\text{s}$.

The states of the chaotic system and the neural network were scaled to six analog outputs of the NI-9264 module within the range of 0 to 10 volts. These outputs were monitored by a Tektronix TDS5104B digital oscilloscope. The results for the system implementation are given in Figures 8(a)–8(d). The phase portraits $\{x_{r_1}[k] - x_{r_3}[k]\}$, $\{x_{r_1}[k] - x_{r_2}[k]\}$, and $\{x_{r_2}[k] - x_{r_3}[k]\}$ are described in Figures 8(a)–8(c). The time evolution of $x_{r_1}[k]$, $x_{r_2}[k]$, and $x_{r_3}[k]$ is shown in Figure 8(d). Regarding the performance of the neural network, the phase portraits $\{x_{r_1}[k] - x_1[k]\}$, $\{x_{r_2}[k] - x_2[k]\}$, and $\{x_{r_3}[k] - x_3[k]\}$ are shown in Figures 9(a)–9(c). Note that each phase portrait evidences a 45° line, which is related to the correct tracking performed by the time-delay neural network. Additional evidence of the tracking is shown in Figure 9(d) where the time evolution of $x_{r_1}[k]$ and $x_1[k]$ and its error $e_1[k]$ is described.

In addition, similar results were achieved for the Liu system. The results for the Liu system are shown in Figures 10 and 11.

5.3. Comparison with Other Works. Synchronization is an emerging behavior in neural networks. It has received academic attention because of a wide variety of applications [3, 8]. Schemes that consider chaotic systems and delayed recurrent neural networks are of particular interest due to

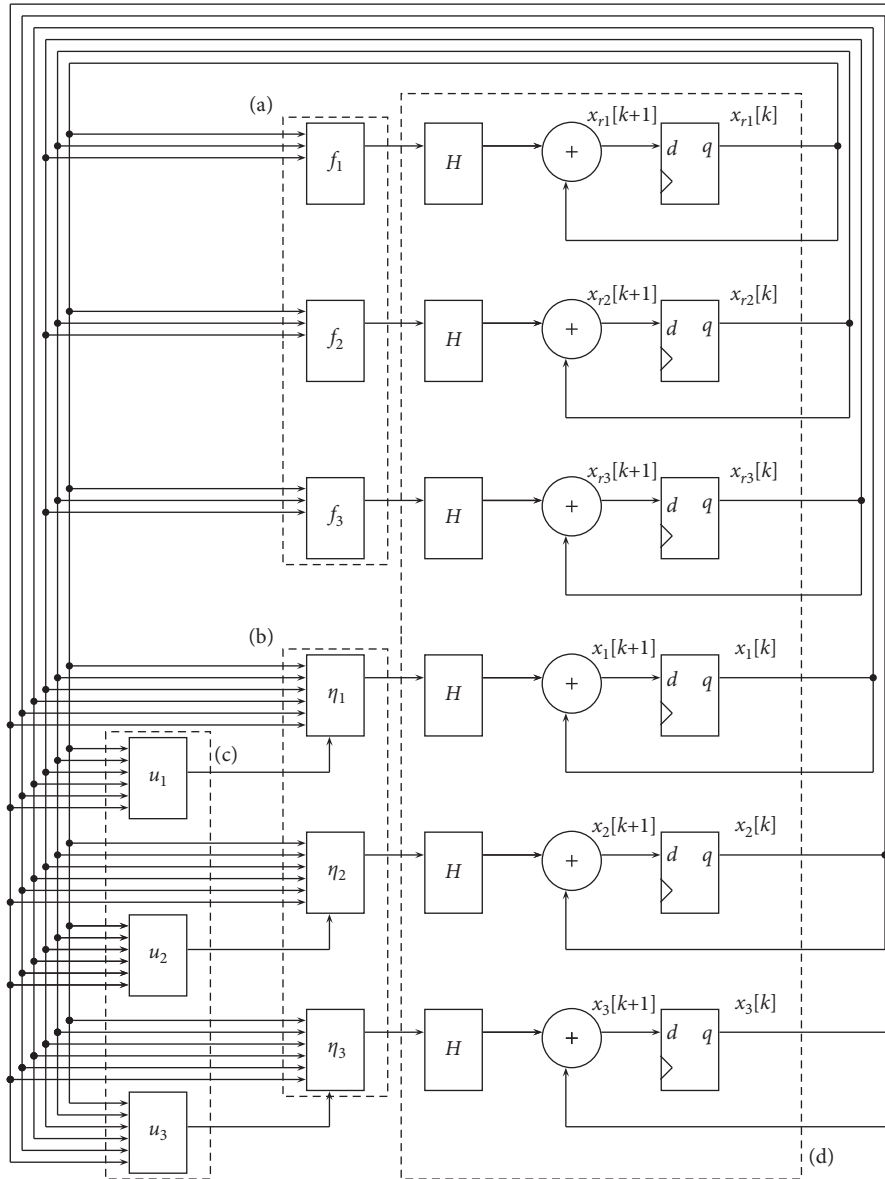


FIGURE 7: Block diagram describing the top-level description of the proposed implementation for the Genesis-Tesi and Liu trackers, where the dashed blocks represent (a) the chaotic system to be tracked given in (31) or (32), (b) the delayed neural network given in (33), (c) the control law given in (34), and (d) the Euler integration method (35).

TABLE 2: FPGA chip resource usage for the Xilinx Zynq-7000 XC7Z020 FPGA chip (the resources used for both implementations are shown in the table).

	Genesis-Tesi			Liu		
	Used	Total	Percent (%)	Used	Total	Percent (%)
Slice registers	24372	106400	22.9	24253	106400	22.8
Slice LUT	27543	53200	51.8	28550	53200	53.7
DSP48s	173	220	78.6	201	220	91.4
Block RAMs	103	140	73.6	103	140	73.6
Clock	40 MHz			40 MHz		

the high complexity depicted in their sensibility to initial conditions, parameters, and time delays. For that reason, synchronizing or controlling these schemes is challenging work; even more, the practical implementation is not a

trivial task. In this section, we perform a survey to demonstrate the pertinence of the proposed control method and the digital implementation. In Table 3, some synchronization schemes in a drive-response configuration are displayed

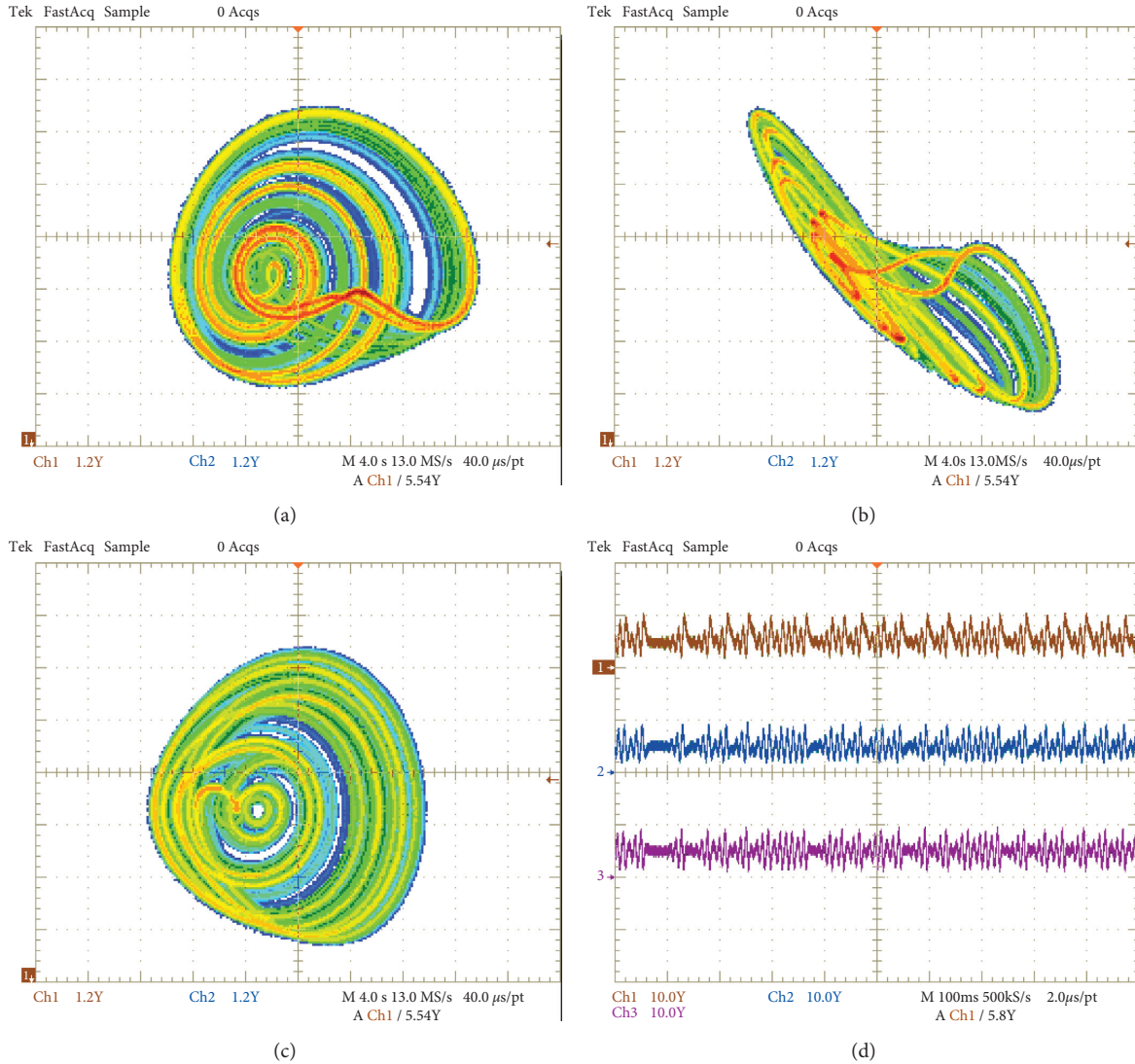


FIGURE 8: Response of the Genesio–Tesi chaotic oscillator implemented, with initial conditions $[x_{r_1}[0], x_{r_2}[0], x_{r_3}[0]]^T = [0.1, 0.1, 0.1]^T$. (a) $x_{r_1} - x_{r_2}$ phase plane. (b) $x_{r_1} - x_{r_3}$ phase plane. (c) $x_{r_2} - x_{r_3}$ phase plane. (d) Time evolution of x_{r_1} (brown line “CH1”), x_{r_2} (blue line “CH2”), and x_{r_3} (purple line “CH3”).

with their respective control approaches. From the information depicted in Table 3, reference [40] considers the implementation of a synchronization scheme on an Arduino chip while the rest of the references only consider numerical results. In this paper, we implement the chaotic system (27), the delayed recurrent neural network (28), and the control laws (29) and (30) in an FPGA.

Table 4 presents a brief review, including some FPGA-based implementations of chaotic systems over the last ten years. Table 4 does not include detailed characteristics like the chip resource usage and the achieved performance because these parameters are hardware dependent. Instead, it includes generalized details like a brief description of the application, the numerical representation, the integration method, and the hardware used.

From Table 4, it can be seen that the implementation of a single chaotic system is a common practice. However, recent works include a chaotic node as a module of a bigger scale

system, for example, a complex network in [38], an OFDM transmission system in [52], or a tracking scheme (this work).

It can also be seen that fixed-point (FXP) and floating-point (IEEE-754 FLP) implementations are equally used. The numerical representation selected in each work is related to a trade-off between accuracy and resource usage. It is evident that, in general, FXP representations require fewer resources and achieve better time performance, while an FLP representation has better numerical accuracy than FXP.

The integration method is used to implement a discrete version of the system. In Table 4, we found a prevalence of the following methods: fourth-order Runge–Kutta (RK4), forward Euler, and Heun. Note that the numerical integration method is not correlated to the date of the work, as we can see recent works using forward Euler and older works using a more complicated RK4 algorithm. Again, as the

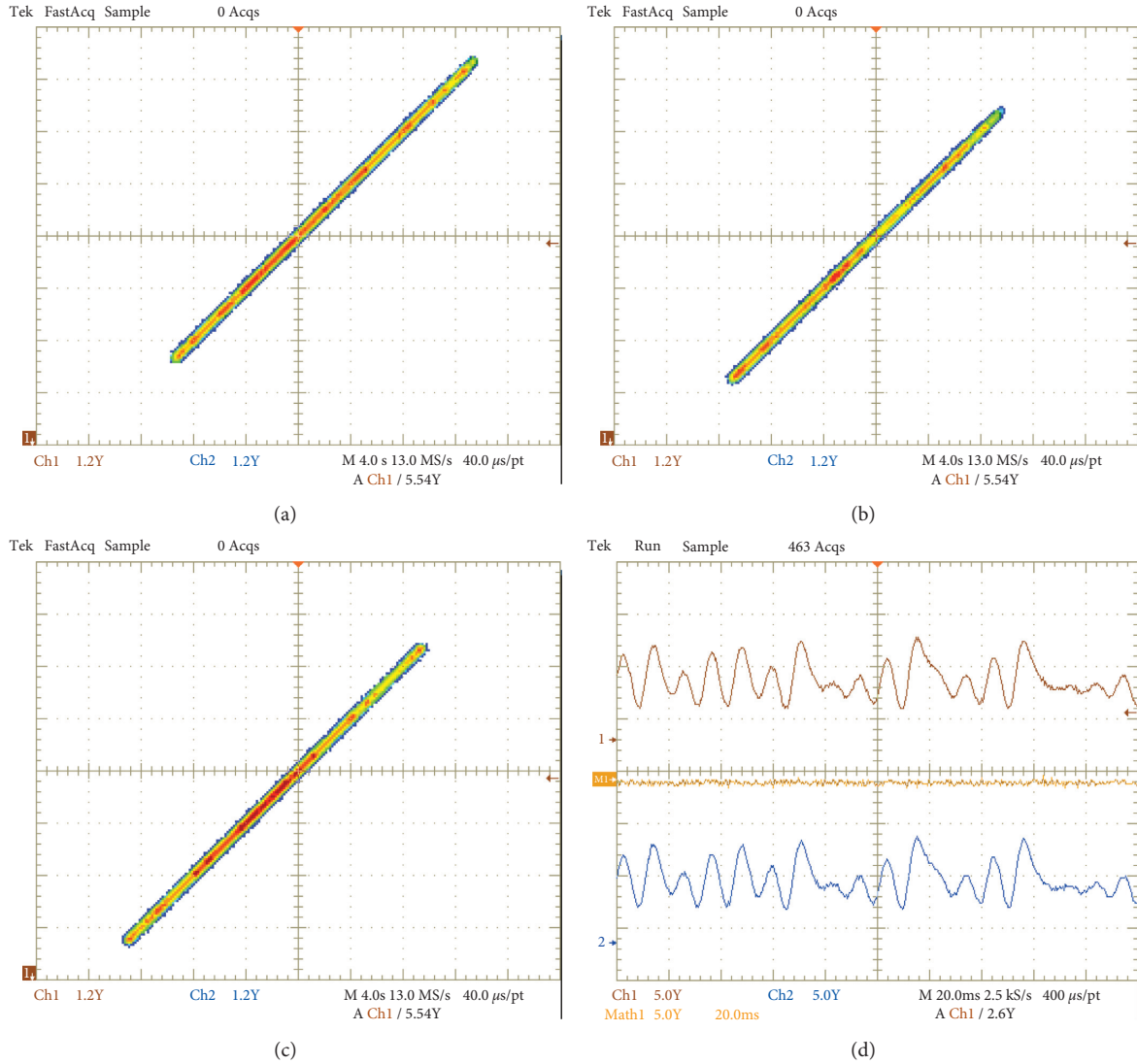


FIGURE 9: Synchronization of the time-delay neural network and the Genesio-Tesi oscillator, with initial conditions $[x_{r_1}[0], x_{r_2}[0], x_{r_3}[0]]^T = [0.1, 0.1, 0.1]^T$ and $[x_1[0], x_2[0], x_3[0]]^T = [1, 1, 1]^T$. (a) $x_{r_1} - x_1$ phase plane. (b) $x_{r_2} - x_2$ phase plane. (c) $x_{r_3} - x_3$ phase plane. (d) Time evolution of x_{r_1} (brown line "CH1"), x_1 (blue line "CH2"), and e_1 (orange line "Math1").

numerical representation, the integration algorithm selection is based on a trade-off between the accuracy and complexity of the design.

There is a diversity of devices and software platforms used for the FPGA fast prototyping. Some devices are displayed in the last column of Table 4, while some examples of

the software platforms are the Matlab HDL Coder (used in [48]), the LabVIEW FPGA software (used in [46]), and Xilinx design suites like ISE and Vivado (used in [49, 52]). In this work, we use LabVIEW FPGA environment. This programming environment is well suited for the system-level design of prototypes.

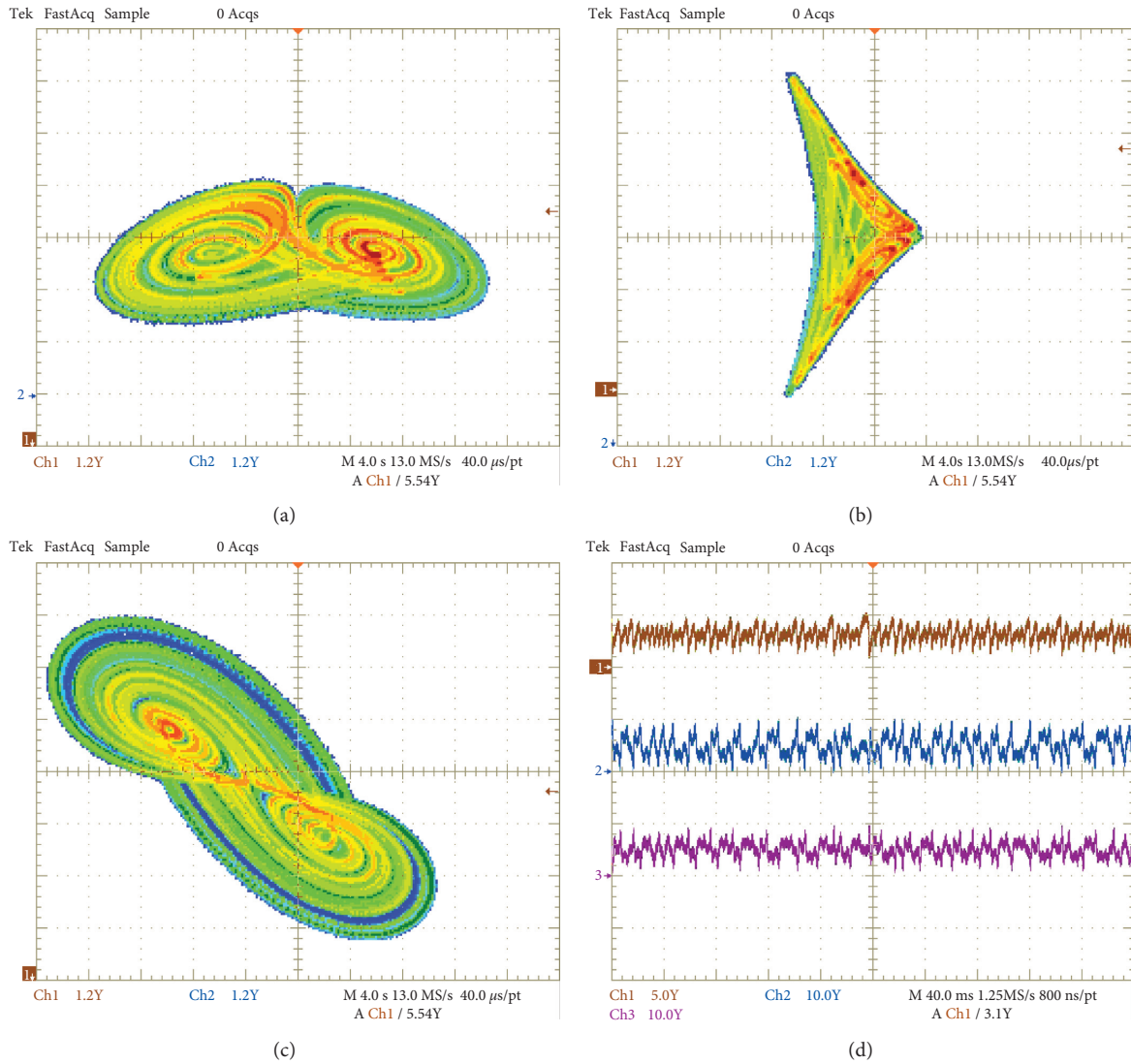


FIGURE 10: Response of the Liu chaotic oscillator implemented, with initial conditions $[x_{r_1}[0], x_{r_2}[0], x_{r_3}[0]]^T = [0.1, 0.1, 0.1]^T$. (a) $x_{r_2} - x_{r_1}$ phase plane. (b) $x_{r_1} - x_{r_3}$ phase plane. (c) $x_{r_2} - x_{r_3}$ phase plane. (d) Time evolution of x_{r_1} (brown line "CH1"), x_{r_2} (blue line "CH2"), and x_{r_3} (purple line "CH3").

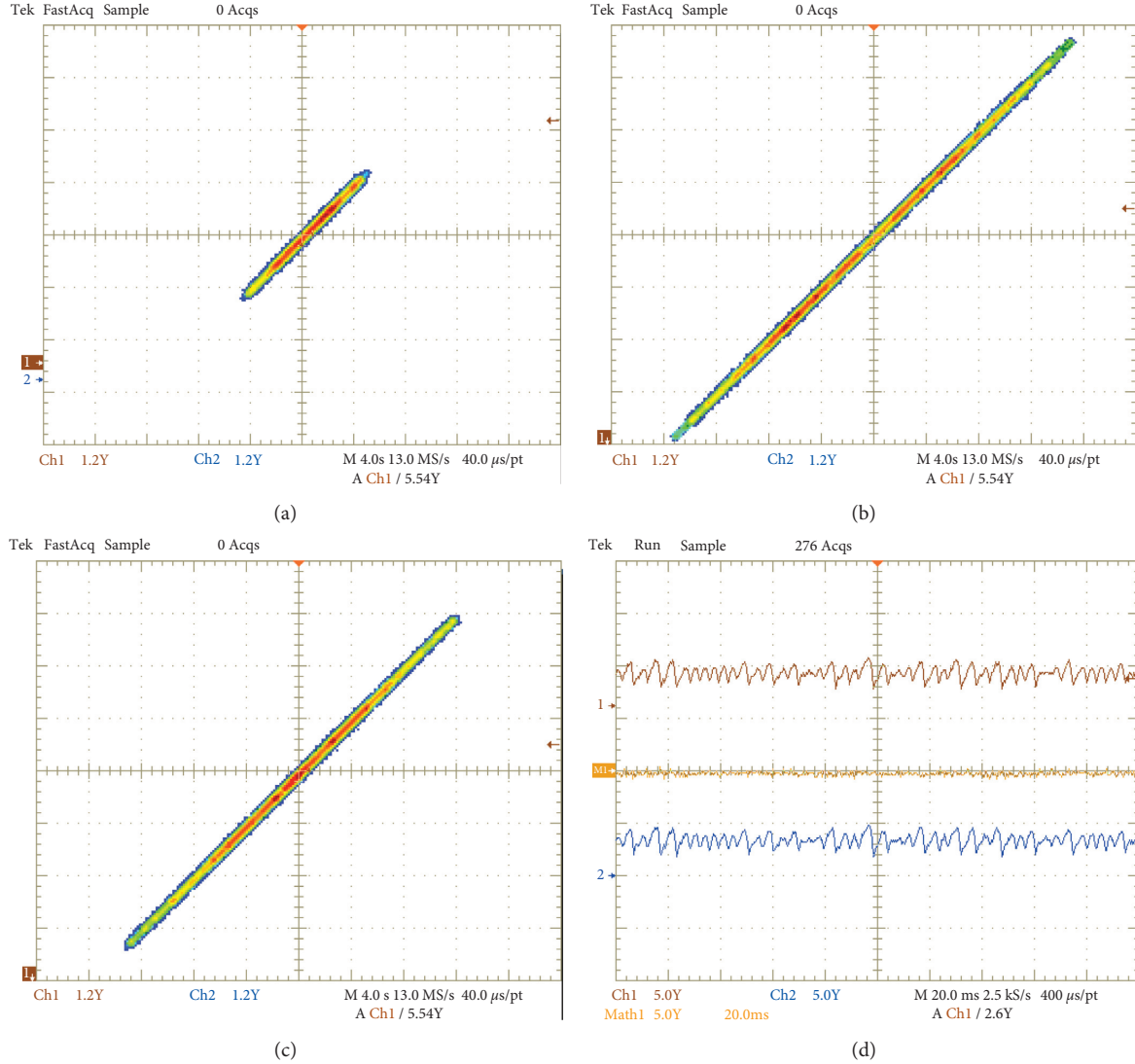


FIGURE 11: Synchronization of the time-delay neural network and the Liu oscillator, with initial conditions $[x_{r_1}[0], x_{r_2}[0], x_{r_3}[0]]^T = [0.1, 0.1, 0.1]^T$ and $[x_1[0], x_2[0], x_3[0]]^T = [1, 1, 1]^T$. (a) $x_{r_1} - x_1$ phase plane. (b) $x_{r_2} - x_{r_3}$ phase plane. (c) $x_{r_3} - x_3$ phase plane. (d) Time evolution of x_{r_1} (brown line “CH1”), x_1 (blue line “CH2”), and e_1 (orange line “Math1”).

TABLE 3: Some synchronization schemes proposed in the literature.

Reference	Approach	Synchronization scheme	Implementation
[40]	Extended Kalman filter	Drive-response between Genesio–Tesi multiscroll and recurrent neural network	Digital: Arduino uno microcontroller
[41]	Inverse optimality	Drive-response between chaotic time-delay recurrent neural networks	None
[19]	V-stability and Lyapunov theory	Drive-response between a delayed recurrent neural network and a complex dynamical network	None
[42]	Quantized sliding-mode control	Drive-response between delayed memristive neural networks	None
[43]	Adaptive intermittent control	Drive-response between chaotic systems with time-varying delay	None
Proposed	Lyapunov–Krasovskii and Lur’e theory	Drive-response between Liu-delayed recurrent neural network and Genesio–Tesi and delayed recurrent neural networks	Digital: FPGA Xilinx Zynq-7000 XC7Z020

TABLE 4: Some FPGA-based implementations of chaotic systems over the last decade.

Reference and year	Brief description	Numerical representation	Integration method	Device
[44], 2009	3D Chen chaotic system	IEEE-754 FLP 32 bits	RK4	Xilinx Virtex-II XCV1000FG456-4
[45], 2011	A new hyperchaotic 4D system	–	RK4	Xilinx Virtex-II Pro XC2VP30
[47], 2013	Switching 3D chaotic systems	FXP 32 bits 16I16Q	Euler	Xilinx Virtex-II FPGA 2VP30FFG896-7
[46], 2015	A 3D chaotic system	IEEE-754 FLP 32 bits	RK4	Xilinx Virtex-6 XC6VLX550T-2FF1759
[48], 2017	Fractional-order chaotic system	FXP 32 bits	Tustin	Altera Cyclone IV EP4CE11529C7N
[38], 2018	Fractional-order complex networks with chaotic nodes	FXP 24 bits 4I20Q	Grünwald–Letnikov	Xilinx Zynq-7000 XC7Z020
[49], 2019	Fractional-order chaotic system	FXP 32 bits	Grünwald–Letnikov	Altera Cyclone IV GX FPGA DE2I-150
[50], 2019	The Lü–Chen 2002 chaotic system	FXP 32 bits 16I16Q	Heun	Xilinx Virtex-6 XC6VLX75T-3FF484
[51], 2019	A new 3D chaotic system	FXP 32 bits 7I25Q	Euler and RK4	Xilinx Virtex-6 XC6VLX75T-3FF484
[52], 2019	OFDM system using chaos	FXP 128 bits	Euler	–
Proposed	A tracking scheme (master and slave)	FXP 30 bits 7I23Q	Euler	Xilinx Zynq-7000 XC7Z020

6. Conclusions

In this paper, the trajectory tracking for a nonlinear system with a time-delay neural network was obtained. In particular, neural networks are forced to follow reference signals generated by a chaotic system, thus achieving the synchronization state. The control law that guarantees trajectory tracking was obtained by considering the Lyapunov–Krasovskii and Lur’e theory. The applicability of the approach was illustrated by considering two different chaotic systems: Liu’s chaotic system and Genesio–Tesi chaotic system. Numerical simulations and FPGA implementations were presented to show the effectiveness of obtained theoretical results. The results obtained allowed us to demonstrate the successful implementation of these trajectory tracking examples on FPGA. Moreover, we confirm that the Lyapunov–Krasovskii theory is an efficient method to deal with the global asymptotic stability of a recurrent neural network with time delay. The synchronization of chaotic systems and the neural networks on different chips streamed through a physical channel could be addressed in a future continuation of this work. It should consider the channel and its properties, and it may be useful in the area of secure communications, where it is common to use the synchronization of chaotic systems as an encryption tool.

Data Availability

The data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no potential conflicts of interest.

Acknowledgments

J. Perez-Padron and J. Paz-Perez acknowledge the academic group “Sistemas dinámicos” of “Facultad de Ciencias Físico-Matemáticas” at UANL. C. Posadas-Castillo, E. Zambrano-Serrano, and M.A. Platas-Garza acknowledge “Departamento de Electrónica y Automatización” from “Facultad de Ingeniería Mecánica y Eléctrica” at UANL. C. Posadas-Castillo and E. Zambrano-Serrano acknowledge CONACYT/Mexico (research grants no. 166654 A1-S-31628 and 350385). C. Posadas-Castillo and M.A. Platas-Garza acknowledge UANL (research grants IT1784-21 and IT1824-21).

References

- [1] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*, Auerbach publications, Broken Sound Parkway, NW, USA, 2016.
- [2] R. Manivannan, S. Panda, K. T. Chong, and J. Cao, “An arcak-type state estimation design for time-delayed static neural networks with leakage term based on unified criteria,” *Neural Networks*, vol. 106, pp. 110–126, 2018.
- [3] Z. Xu, S. Li, X. Zhou, and T. Cheng, “Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties,” *Neurocomputing*, vol. 357, pp. 271–281, 2019.
- [4] T. Guo, L. Wang, M. Zhou, and S. Duan, “A multi-layer memristive recurrent neural network for solving static and dynamic image associative memory,” *Neurocomputing*, vol. 334, pp. 35–43, 2019.
- [5] Y. Jiang, C. Yang, J. Na, G. Li, Y. Li, and J. Zhong, “A brief review of neural networks based learning and control and their applications for robots,” *Complexity*, vol. 2017, Article ID 1895897, 14 pages, 2017.

- [6] Y. Li, W. Zheng, Z. Cui, and T. Zhang, "Face recognition based on recurrent regression neural network," *Neurocomputing*, vol. 297, pp. 50–58, 2018.
- [7] C. Qinetal, "Convolutional recurrent neural networks for dynamic MR image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 280–290, 2018.
- [8] K. Rajagopal, A. J. M. Khalaf, F. Parastesh, I. Moroz, A. Karthikeyan, and S. Jafari, "Dynamical behavior and network analysis of an extended Hindmarsh-Rose neuron model," *Nonlinear Dynamics*, vol. 98, no. 1, pp. 477–487, 2019.
- [9] J. Wang, F. Liu, and S. Qin, "Global exponential stability of uncertain memristor-based recurrent neural networks with mixed time delays," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 4, pp. 743–755, 2019.
- [10] G. Chen and Y. Yang, "New sufficient conditions for finite time stability of nonlinear time delay systems," *Asian Journal of Control*, vol. 21, no. 5, pp. 2321–2329, 2019.
- [11] E. Zambrano-Serrano and A. Anzo-Hernandez, "A novel antimonotonic hyperjerk system: analysis, synchronization and circuit design," *Physica D: Nonlinear Phenomena*, vol. 424, no. 1–8, p. 132927, 2021.
- [12] A. Taher Azar, F. E. Serrano, Q. Zhu et al., "Robust stabilization and synchronization of a novel chaotic system with input saturation constraints," *Entropy*, vol. 23, no. 3, p. 1110, 2021.
- [13] Y. Kan, J. Lu, J. Qiu, and J. Kurths, "Exponential synchronization of time-varying delayed complex-valued neural networks under hybrid impulsive controllers," *Neural Networks*, vol. 114, pp. 157–163, 2019.
- [14] S. Yang, Z. Guo, and J. Wang, "Global synchronization of multiple recurrent neural networks with time delays via impulsive interactions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1657–1667, 2016.
- [15] H. R. Karimi and H. Gao, "New delay-dependent exponential H_∞ synchronization for uncertain neural networks with mixed time delays," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 1, pp. 173–185, 2009.
- [16] B. Huang, H. Zhang, D. Gong, and J. Wang, "Synchronization analysis for static neural networks with hybrid couplings and time delays," *Neurocomputing*, vol. 148, pp. 288–293, 2015.
- [17] Z. Zhang and L. Ren, "New sufficient conditions on global asymptotic synchronization of inertial delayed neural networks by using integrating inequality techniques," *Nonlinear Dynamics*, vol. 95, no. 2, pp. 905–917, 2019.
- [18] Q. Song, J. Cao, and F. Liu, "Pinning synchronization of linearly coupled delayed neural networks," *Mathematics and Computers in Simulation*, vol. 86, pp. 39–51, 2012.
- [19] J. P. Perez, J. P. Padron, A. F. Hemandez, and S. Arroyo, "Complex dynamical network control for trajectory tracking using delayed recurrent neural networks," *Mathematical Problems in Engineering*, vol. 2014, Article ID 162610, 7 pages, 2014.
- [20] X.-M. Zhang, Q.-L. Han, X. Ge, and D. Ding, "An overview of recent developments in Lyapunov-Krasovskii functionals and stability criteria for recurrent neural networks with time-varying delays," *Neurocomputing*, vol. 313, pp. 392–401, 2018.
- [21] J. Chen, J. H. Park, and S. Xu, "Stability analysis of continuous-time systems with time-varying delay using new Lyapunov-Krasovskii functionals," *Journal of the Franklin Institute*, vol. 355, no. 13, pp. 5957–5967, 2018.
- [22] W. Yang, Y.-W. Wang, Y. Shen, and L. Pan, "Cluster synchronization of coupled delayed competitive neural networks with two time scales," *Nonlinear Dynamics*, vol. 90, no. 4, pp. 2767–2782, 2017.
- [23] X.-M. Qing-Long Han and Q.-L. Han, "New Lyapunov-Krasovskii functionals for global asymptotic stability of delayed neural networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 533–539, 2009.
- [24] A. Kundu, P. Das, and A. B. Roy, "Stability, bifurcations and synchronization in a delayed neural network model of n-identical neurons," *Mathematics and Computers in Simulation*, vol. 121, pp. 12–33, 2016.
- [25] J. Perez-Padron, J. P. Perez, C. F. Mendez-Barrios, and V. Ramírez-Rivera, "Trajectory tracking using adaptive fractional PID control of biped robots with time-delay feedback," in *Becoming Human with Humanoid from Physical Interaction to Social Intelligence*, A. Hoirul Basori, A. Leylavi Shoushtari, and A. V. T. IntechOpen, Eds., IntechOpen, London, UK, 2020.
- [26] J. Perez-Padron and J. P. Perez-Padron, "Trajectory tracking error between plant, reference and adaptive neural networks using two control law for two-link robot manipulator," in *Proceedings of the 2015 International Conference on Environmental Science and Sustainable Development (ICESSD)*, Bangkok, Thailand, October 2015.
- [27] J. Perez-Padron, J. P. Perez, C. F. Mendez Barrios, and E. J. González Galván, "Fractional complex dynamical systems for trajectory tracking using fractional neural network via fractional order PID control law," *Computación Y Sistemas*, vol. 24, no. 1, pp. 349–357, 2020.
- [28] J. Jun Wang and J. Wang, "Global asymptotic and robust stability of recurrent neural networks with time delays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 2, pp. 417–426, 2005.
- [29] J. Cao, Z. Wang, and Y. Sun, "Synchronization in an array of linearly stochastically coupled networks with time delays," *Physica A: Statistical Mechanics and Its Applications*, vol. 385, no. 2, pp. 718–728, 2007.
- [30] V. Kolmanovskii and A. Myshkis, *Applied Theory of Functional Differential Equations*, Springer Science & Business Media, Berlin, Germany, 1992.
- [31] M. Krstic and H. Deng, *Stabilization of Nonlinear Uncertain Systems*, Springer-Verlag, Berlin, Germany, 1998.
- [32] E. N. Sanchez, J. P. Perez, and G. Chen, "Using dynamic neural networks to generate chaos: an inverse optimal control approach," *International Journal of Bifurcation and Chaos*, vol. 11, no. 3, pp. 857–863, 2001.
- [33] R. Genesio and A. Tesi, "Harmonic balance methods for the analysis of chaotic dynamics in nonlinear systems," *Automatica*, vol. 28, no. 3, pp. 531–548, 1992.
- [34] J. C. Sprott, *Elegant Chaos: Algebraically Simple Chaotic Flows*, World Scientific, Singapore, 2010.
- [35] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
- [36] C. Liu, L. Liu, and T. Liu, "A novel three-dimensional autonomous chaos system, Chaos," *Solitons & Fractals*, vol. 39, no. 4, pp. 1950–1958, 2009.
- [37] J. M. Munoz-Pacheco, C. Posadas-Castillo, and E. Zambrano-Serrano, "The effect of a non-local fractional operator in an asymmetrical glucose-insulin regulatory system: analysis, synchronization and electronic implementation," *Symmetry*, vol. 12, no. 9, p. 1395, 2020.
- [38] A. G. Soriano-Sanchez, C. Posadas-Castillo, M. A. Platas-Garza, and A. Arellano-Delgado, "Synchronization and FPGA realization of complex networks with fractional-order Liu

- chaotic oscillators,” *Applied Mathematics and Computation*, vol. 332, pp. 250–262, 2018.
- [39] R. Hegger, H. Kantz, and T. Schreiber, “Practical implementation of nonlinear time series methods: the TISEAN package,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 9, no. 2, pp. 413–435, 1999.
- [40] C. E. Castañeda, D. López-Mancilla, R. Chiu et al., “Discrete-time neural synchronization between an arduino microcontroller and a compact development system using multi-scroll chaotic signals,” *Chaos, Solitons & Fractals*, vol. 119, pp. 269–275, 2019.
- [41] Z. Liu, “Design of nonlinear optimal control for chaotic synchronization of coupled stochastic neural networks via Hamilton-Jacobi-Bellman equation,” *Neural Networks*, vol. 99, pp. 166–177, 2018.
- [42] B. Sun, S. Wang, Y. Cao, Z. Guo, T. Huang, and S. Wena, “Exponential synchronization of memristive neural networks with time-varying delays via quantized sliding-mode control,” *Neural Networks*, vol. 126, pp. 163–169, 2020.
- [43] Y. Wang and L. Dong, “Adaptive synchronization of chaotic systems with time-varying delay via aperiodically intermittent control,” *Soft Computing*, vol. 24, pp. 12773–12780, 2020.
- [44] S. Sadoudi, M. Salah Azzaz, M. Djeddou, and M. Benssalah, “An FPGA real-time implementation of the Chen’s chaotic system for securing chaotic communications,” *International Journal of Nonlinear Science*, vol. 7, no. 4, pp. 467–474, 2009.
- [45] Z. Q. Chen, G. Q. Liang, and J. G. Tong, “The FPGA implementation of hyperchaotic system based upon VHDL design,” in *Proceedings of the 2011 Fourth International Workshop on Chaos-Fractals Theories and Applications*, pp. 47–51, Hangzhou, China, October 2011.
- [46] A. Akgul, H. Calgan, I. Koyuncu, I. Pehlivan, and A. Istanbulu, “Chaos-based engineering applications with a 3D chaotic system without equilibrium points,” *Nonlinear Dynamics*, vol. 84, no. 2, pp. 481–495, 2016.
- [47] M. Salah Azzaz, C. Tanougast, S. Sadoudi, R. Fellah, and A. Dandache, “A new auto-switched chaotic system and its FPGA implementation,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 7, pp. 1792–1804, 2013.
- [48] D. K. Shah, R. B. Chaurasiya, V. A. Vyawahare, K. Patil, and M. D. Patilb, “FPGA implementation of fractional-order chaotic systems,” *AEU-International Journal of Electronics and Communications*, vol. 78, pp. 245–257, 2017.
- [49] A. D. Pano-Azucena, B. Ovilla-Martinez, E. Tlelo-Cuautle, J. Manuel Muñoz-Pacheco, and L. G. de la Fraga, “FPGA-based implementation of different families of fractional-order chaotic oscillators applying Grünwald-Letnikov method,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 72, pp. 516–527, 2019.
- [50] M. Tuna, M. Alçın, İ. Koyuncu, C. B. Fidan, and İ. Pehlivan, “High speed FPGA-based chaotic oscillator design,” *Microprocessors and Microsystems*, vol. 66, pp. 72–80, 2019.
- [51] A. Sambas, S. Vaidyanathan, E. Tlelo-Cuautle et al., “A novel chaotic system with two circles of equilibrium points: multistability, electronic circuit and FPGA realization,” *Electronics*, vol. 8, no. 11, p. 1211, 2019.
- [52] Y. Wang, Y. Xuelin, and W. Hu, “FPGA implementation of real-time secure OFDM transmission using digital chaos,” in *Proceedings of the 20th International Conference on Transparent Optical Networks ICTON*, July 2018.