ISSN 2007-9737

Fractional Complex Dynamical Systems for Trajectory Tracking using Fractional Neural Network

Joel Perez P.

Universidad Autonoma de Nuevo Leon, Facultad de Ciencias Fisico Matematicas, Monterrey, Mexico

joelperezp@yahoo.com

Abstract. In this paper the problem of trajectory tracking is studied. Based on Lyapunov theory, a control law that achieves global asymptotic stability of the tracking error between a fractional recurrent neural network and the state of each single node of the fractional complex dynamical network is obtained. To illustrate the analytic results we present a tracking simulation of a simple network with four different nodes and five non-uniform links.

Keywords. Fractional complex dynamical systems, trajectory tracking, Lyapunov theory, control law.

1 Introduction

Trajectory tracking is a well studied problem in control theory. Its applications cover a wide range of topics from recognition of moving objects to synchronization, see, for example, [3, 9, 11]. Complex networks are dynamical systems interconnected by a function. Its behavior can be difficult to control and the dynamics of its nodes require a precise analysis, see for example [4]. In the classical case, one can track the nodes of the network by using Lyapunov theory and neural networks as in [10].

In recent years, there has been an increasing interest in studying fractional order systems, i.e. dynamical systems with differential equations of fractional order, see, for example, [1]. In these systems, the classical mathematical notion of derivative is changed to allow arbitrary orders. Fractional order neural network synchronization is studied, for instance, in [6]. In the case of complex dynamical networks of fractional order, cluster synchronization, stabilization, and partial synchronization has been studied, see [7, 8, 5].

In this paper we propose to use recurrent neural networks to track the nodes of a fractional order complex network. We use a Lyapunov function and the result in [2] to design a control law that tracks the system. We prove that tracking is guaranteed by showing that the error between the network and the neural network stabilizes if the control is applied. We show this in a rigorous mathematical form. The control law we obtain is of very general nature and applies to general networks. We provide an example to show how the control law applies to a specific situation.

2 Mathematical Models

2.1 Fractional General Complex Dynamical Network

In this work we use Caputo's fractional operator which is defined, for $0 < \alpha < 1$, by

$$x^{(\alpha)}(t) = {}_0^c D_t^\alpha x(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t x^{'}(\tau)(t-\tau)^{-\alpha} d\tau.$$

If $x(t) \in \mathbb{R}^n$, we consider that $x^{(\alpha)}(t)$ is the Caputo fractional operator applied to each entry:

$$x^{(\alpha)}(t) = ({}_0^c D_t^\alpha x_{i1}(t), ..., {}_0^c D_t^\alpha x_{in}(t))^T.$$

Consider a network consisting of N linearly and diffusively coupled nodes, with each node being an n-dimensional dynamical system, described by

$$x_i^{(\alpha)} = f_i(x_i) + \sum_{\substack{j=1 \\ j \neq i}}^{N} c_{ij} a_{ij} \Gamma(x_j - x_i), \quad i = 1, 2, \dots, N, \tag{1}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in \mathbb{R}^n$ are the state vectors of node i, $f_i : \mathbb{R}^n \longmapsto \mathbb{R}^n$ represents the self-dynamics of node i, constants $c_{ij} > 0$ are the coupling strengths between node i and node j, with $i, j = 1, 2, \ldots, N$. $\Gamma = (\tau_{ij}) \in \mathbb{R}^{n\times n}$ is a constant internal matrix that describes the way of linking the components in each pair of connected node vectors $(x_j - x_i)$: i.e. for some pairs (i, j) with $1 \le i, j \le n$ and $\tau_{ij} \ne 0$ the two coupled nodes are linked through their ith and jth sub-state variables, respectively, while the coupling matrix $A = (a_{ij}) \in \mathbb{R}^{N\times N}$ denotes the coupling configuration of the entire network: i.e. if there is a connection between node i and node $j(i \ne j)$, then $a_{ij} = a_{ji} = 1$; otherwise $a_{ij} = a_{ji} = 0$.

2.2 Fractional Recurrent Neural Network

Consider a fractional recurrent neural network in the following form:

$$\begin{aligned} x_{n_i}^{(\alpha)} &= A_{n_i} x_{n_i} + W_{n_i}\sigma(x_{in}) + u_{in} + \\ & \sum_{\substack{j=1\\ j\ne i}}^{N} c_{injn} a_{injn} \Gamma(x_{jn} - x_{in}), \\ i &= 1, 2, \ldots, N, \end{aligned} \tag{2}$$

where $x_{in} = (x_{in_1}, x_{in_2}, \ldots, x_{in_n})^T \in \mathbb{R}^n$ is the state vector of neural network i, $u_{in} \in \mathbb{R}^n$ is the input of neural network i, $A_{in} = -\lambda_{in} I_{n\times n}$, $i = 1, 2, \ldots, N$, is the state feedback matrix, with λ_{in} being a positive constant, $W_{in} \in \mathbb{R}^{n\times n}$ is the connection weight matrix with $i = 1, 2, \ldots, N$, and $\sigma(\cdot) \in \mathbb{R}^n$ is a Lipschitz sigmoid vector function [4], [5], such that $\sigma(x_{in}) = 0$ only at $x_{in} = 0$, with Lipschitz constant L_{σ_i}, $i = 1, 2, \ldots, N$ and neuron activation functions $\sigma_i(\cdot) = \tanh(\cdot)$, $i = 1, 2, \ldots, n$.

3 Trajectory Tracking

The objective is to develop a control law such that the ith fractional neural network (2) tracks the trajectory of the ith fractional dynamical system (1). We define the tracking error as $e_i = x_{in} - x_i$, $i = 1, 2, \ldots, N$ whose time derivative is

$$e_i^{(\alpha)} = x_{in_i}^{(\alpha)} - x_i^{(\alpha)}, \quad i = 1, 2, \ldots, N. \tag{3}$$

From (1), (2) and (3), we obtain

$$\begin{aligned} e_i^{(\alpha)} &= A_{in} x_{in} + W_{in}\sigma(x_{in}) + u_{in} - f_i(x_i) + \\ & \sum_{\substack{j=1\\ j\ne i}}^{N} c_{injn} a_{injn} \Gamma(x_{jn} - x_{in}) - \\ & \sum_{\substack{j=1\\ j\ne i}}^{N} c_{ij} a_{ij} \Gamma(x_j - x_i), \\ & i = 1, 2, \ldots, N. \end{aligned} \tag{4}$$

Adding and substrating $W_{in}\sigma(x_i)$, $\alpha_i(t)$, $i = 1, 2, \ldots, N$, to (4), where α_i is defined below, and considering that $x_{in} = e_i + x_i$, $i = 1, 2, \ldots, N$, then

$$\begin{aligned} e_i^{(\alpha)} &= A_{in} e_i + W_{in}(\sigma(e_i + x_i) - \sigma(x_i)) + \\ & (u_{in} - \alpha_i) + \\ & (A_{in} x_i + W_{in}\sigma(x_i) + \alpha_i) - f_i(x_i) + \\ & \sum_{\substack{j=1\\ j\ne i}}^{N} c_{injn} a_{injn} \Gamma(x_{jn} - x_{in}) - \\ & \sum_{\substack{j=1\\ j\ne i}}^{N} c_{ij} a_{ij} \Gamma(x_j - x_i), \\ & i = 1, 2, \ldots, N. \end{aligned} \tag{5}$$

In order to guarantee that the ith neural network (2) tracks the ith reference trajectory (1), the following assumption has to be satisfed:

Assumption 1. There exist functions $\rho_i(t)$ and $\alpha_i(t)$, $i = 1, 2, \ldots, N$, such that

$$\begin{aligned} \rho_i^{(\alpha)}(t) &= A_{in}\rho_i(t) + W_{in}\sigma(\rho_i(t)) + \alpha_i(t), \\ \rho_i(t) &= x_i(t), \\ i &= 1, 2, \ldots, N. \end{aligned} \tag{6}$$

Let's define

$$\begin{aligned} \widetilde{u}_{in} &= (u_{in} - \alpha_i), \\ \phi_\sigma(e_i, x_i) &= \sigma(e_i + x_i) - \sigma(x_i), \\ i &= 1, 2, \ldots, N. \end{aligned} \tag{7}$$

From (6) and (7), equation (5) is reduced to

$$
\begin{aligned}
e_i^{(\alpha)} = \quad & A_{in}e_i + W_{in}\phi_\sigma(e_i, x_i) + \widetilde{u}_{in} + \\
& \sum_{\substack{j=1\\j\neq i}}^{N} c_{injn}a_{injn}\Gamma(x_{jn} - x_{in}) - \\
& \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma(x_j - x_i), \qquad (8)\\
& i = 1, 2, ..., N.
\end{aligned}
$$

We can also write

$$
\begin{aligned}
& \sum_{\substack{j=1\\j\neq i}}^{N} c_{injn}a_{injn}\Gamma(x_{jn} - x_{in}) = \\
& \Gamma(\sum_{\substack{j=1\\j\neq i}}^{N} c_{injn_j}a_{injn}x_{jn} - x_{in}\sum_{\substack{j=1\\j\neq i}}^{N} c_{injn}a_{injn}) = \\
& \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma(x_j - x_i) = \qquad (9)\\
& \Gamma(\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}x_j - x_i\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}),\\
& i = 1, 2, \ldots, N,
\end{aligned}
$$

where we used $c_{injn} = c_{ij}$ and $a_{injn} = a_{ij}$. Then, with the above equation, equation (8) becomes

$$
\begin{aligned}
e_i^{(\alpha)} &= A_{in}e_i + W_{in}\phi_\sigma(e_i, x_i) + \widetilde{u}_{in} + \\
&\quad \Gamma(\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_j - e_i\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}) \\
&= A_{n_i}e_i + W_{in}\phi_\sigma(e_i, x_i) + \widetilde{u}_{in} + \qquad (10)\\
&\quad \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma(e_j - e_i),\\
&\quad i = 1, 2, \ldots, N.
\end{aligned}
$$

It is clear that $e_i = 0$, $i = 1, 2, \ldots, N$ is an equilibrium point of (10), when $\widetilde{u}_{in} = 0$, $i = 1, 2, \ldots, N$. Therefore, the tracking problem can be restated as a global asymptotic stabilization problem for the system (10).

4 Tracking Error Stabilization and Control Design

In order to establish the convergence of (10) to $e_i = 0$, $i = 1, 2, \ldots, N$, which ensures the desired tracking, first, we propose the following candidate Lyapunov function

$$
\begin{aligned}
V_N(e) &= \sum_{i=1}^{N} V(e_i) = \sum_{i=1}^{N} \tfrac{1}{2} \parallel e_i \parallel^2 \\
&= \frac{1}{2}\sum_{i=1}^{N} e_i^T e_i, \quad e = (e_1^T, \ldots, e_N^T)^T. \qquad (11)
\end{aligned}
$$

In fractional calculus, the product rule for the derivative is no longer valid. However, we still have an upper bound for the product that appears in (11). Specifically, from Lemma 1 in [2] the time derivative of (11), along the trajectories of (10), is

$$
\begin{aligned}
V_N^{(\alpha)}(e) &\leq \frac{\partial V_N(e)}{\partial e} D^\alpha e_i \\
&= \sum_{i=1}^{N} \frac{\partial V_N(e)}{\partial e_i} D^\alpha e_i \\
&= (e_1^T, \ldots, e_N^T)^T \times
\end{aligned}
$$

$$
\begin{pmatrix}
A_{1n}e_1 + W_{1n}\phi_\sigma(e_i, x_i) + \widetilde{u}_{1n} + \\
\sum_{\substack{j=1\\j\neq i}}^{N} c_{1j}a_{1j}\Gamma(e_j - e_1) \\
\vdots \\
A_{Nn}e_N + W_{Nn}\phi_\sigma(e_N, x_N) + \widetilde{u}_{Nn} + \\
\sum_{\substack{j=1\\j\neq i}}^{N} c_{Nj}a_{Nj}\Gamma(e_j - e_N)
\end{pmatrix}
$$

$$
\begin{aligned}
= \quad & \sum_{j=1}^{N} e_i^T (A_{in}e_i + W_{in}\phi_\sigma(e_i, x_i) + \widetilde{u}_{in} + \\
& \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma(e_j - e_j)). \qquad (12)
\end{aligned}
$$

We can then write

$$
\begin{aligned}
& V_N^{(\alpha)}(e) \leq \\
& \sum_{i=1}^{N} \left(-\lambda_{i_n} \|e_i\|^2 + e_i^{\mathsf{T}} W_{i_n}\phi_\sigma(e_i, x_i) + e_i^{\mathsf{T}}\widetilde{u}_{i_n}\right) +
\end{aligned}
$$

$$\sum_{i=1}^{N}\left(\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_i^{\intercal}\Gamma e_j - \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_i^{\intercal}\Gamma e_i\right). \quad (13)$$

Next, let's consider the following inequality, proved in [12]:

$$X^{\top}Y + Y^{\top}X \leq X^{\top}\Lambda X + Y^{\top}\Lambda^{-1}Y, \quad (14)$$

which holds for all matrices $X, Y \in \mathbb{R}^{n\times k}$ and $\Lambda \in \mathbb{R}^{n\times n}$ with $\Lambda = \Lambda^{\top} > 0$. Applying (14) with $\Lambda = I_{n\times n}$ to the term $e_i^{\intercal}W_{i_n}\phi_\sigma(e_i,x_i)$, $i = 1,2,...,N$, we get

$$\begin{aligned}
&e_i^{\intercal}W_{i_n}\phi_\sigma(e_i,x_i) \leq \\
&\frac{1}{2}e_i^{\intercal}e_i + \frac{1}{2}\phi_\sigma^{\top}(e_i,x_i)W_{i_n}^{\top}W_{i_n}\phi_\sigma(e_i,x_i) = \\
&\frac{1}{2}\left\|e_i\right\|^2 + \frac{1}{2}\phi_\sigma^{\top}(e_i,x_i)\times W_{i_n}^{\top}W_{i_n}\phi_\sigma(e_i,x_i), \\
&i = 1,2,...,N. \quad (15)
\end{aligned}$$

Since ϕ_σ is Lipschitz, then

$$\left\|\phi_\sigma(e_i,x_i)\right\| \leq L_{\phi_{\sigma 1}} \parallel e_i \parallel, i = 1,2,...,N \quad (16)$$

with Lipschitz constant $L_{\phi_{\sigma_i}}$. Applying (16) to $\frac{1}{2}\phi_\sigma^{\top}(e_i,x_i)W_{i_n}^{\top}W_{i_n}\phi_\sigma(e_i,x_i)$ we obtain

$$\begin{aligned}
&\frac{1}{2}\phi_\sigma^{\top}(e_i,x_i)W_{i_n}^{\top}W_{i_n}\phi_\sigma(e_i,x_i) \\
\leq\ &\frac{1}{2}\left\|\phi_\sigma^{\top}(e_i,x_i)W_{i_n}^{\top}W_{i_n}\phi_\sigma(e_i,x_i)\right\| \quad (17)\\
\leq\ &\frac{1}{2}\left(L_{\phi_{\sigma_i}}\right)^2\left\|W_{i_n}\right\|^2\left\|e_i\right\|^2, \ i = 1,2,...,N.
\end{aligned}$$

Next, (15) is reduced to

$$\begin{aligned}
&e_i^{\intercal}W_{i_n}\phi_\sigma(e_i,x_i) \\
\leq\ &\frac{1}{2}\left\|e_i\right\|^2 + \frac{1}{2}\left(L_{\phi_{\sigma_i}}\right)^2\left\|W_{i_n}\right\|^2\left\|e_i\right\|^2 \quad (18)\\
=\ &\frac{1}{2}\left(1 + L_{\phi_{\sigma_i}}^2\left\|W_{i_n}\right\|^2\right)\left\|e_i\right\|^2, \quad i = 1,2,...,N.
\end{aligned}$$

Then, we have that

$$\begin{aligned}
V_N^{(\alpha)}(e) \leq &-\sum_{i=1}^{N}\left(\lambda_{i_n}\left\|e_i\right\|^2 + \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_i^{\intercal}\Gamma e_i\right) + \\
&\frac{1}{2}\sum_{i=1}^{N}\left(\left(1 + L_{\phi_{\sigma_i}}^2\left\|W_{i_n}\right\|^2\right)\left\|e_i\right\|^2 + 2\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_i^{\intercal}\Gamma e_j\right) \\
&+ \sum_{i=1}^{N} e_i^{\intercal}\widetilde{u}_{i_n}. \quad (19)
\end{aligned}$$

We define $\widetilde{u}_{i_n} = \widetilde{\widetilde{u}}_i + \widetilde{\widetilde{u}}_{i_j}$, $i = 1,2,...,N$, and from (19) we get

$$\begin{aligned}
V_N^{(\alpha)}(e) \leq &-\sum_{i=1}^{N}\left(\lambda_{i_n}\left\|e_i\right\|^2 + \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}e_i^{\intercal}\Gamma e_i\right) + \\
&\frac{1}{2}\sum_{i=1}^{N}\left(e_i^{\intercal}\left(\left(1 + L_{\phi_{\sigma_i}}^2\left\|W_{i_n}\right\|^2\right)e_i + 2\widetilde{\widetilde{u}}_i\right)\right) + \quad (20)\\
&\sum_{i=1}^{N}\left(e_i^{\intercal}\left(\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma e_j + \widetilde{\widetilde{u}}_{i_j}\right)\right).
\end{aligned}$$

Now, we propose to use the following control law:

$$\begin{aligned}
\widetilde{u}_{i_n} &= \widetilde{\widetilde{u}}_i + \widetilde{\widetilde{u}}_{i_j} \\
&= -\frac{1}{2}\left(1 + L_{\phi_{\sigma_i}}^2\left\|W_{i_n}\right\|^2\right)e_i \\
&\quad -\sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma e_j, \quad (21)\\
i &= 1,2,...,N.
\end{aligned}$$

In this case, $V_N^{(\alpha)}(e) < 0, \forall\ e \neq 0$. This means that the proposed control law (21) can globally and asymptotically stabilize the ith error system (10), therefore ensuring the tracking of (1) by (2).

Finally, the control action of the recurrent neural networks is given by

$$\begin{aligned}
u_{i_n} &= \widetilde{u}_{i_n} + \alpha_i \\
&= -\frac{1}{2}\left(1 + L_{\phi_{\sigma_i}}^2\left\|W_{i_n}\right\|^2\right)e_i - \\
&\quad \sum_{\substack{j=1\\j\neq i}}^{N} c_{ij}a_{ij}\Gamma e_j + f_i\left(x_i\right) + \lambda_{i_n}x_i - W_{i_n}\sigma\left(x_i\right), \\
i &= 1,2,...,N. \quad (22)
\end{aligned}$$

5 Simulations

In order to illustrate the application of the discussed results, we consider a simple network with four different nodes and five non-uniform links, see Fig.1. The node self-dynamics are described by (see [13] for the origins of this example):

$$x_1^{(\alpha)} = -x_1^3,\ x_2^{(\alpha)} = -3x_2,$$
$$x_3^{(\alpha)} = \sin x_3,\ x_4^{(\alpha)} = -\mid x_4 \mid \qquad (23)$$

and the coupling strengths are $c_{12} = c_{21} = 1.3$, $c_{14} = c_{41} = 1.0$, $c_{13} = c_{31} = 2.7$, $c_{24} = c_{42} = 2.1$, $c_{34} = c_{43} = 1.5$.

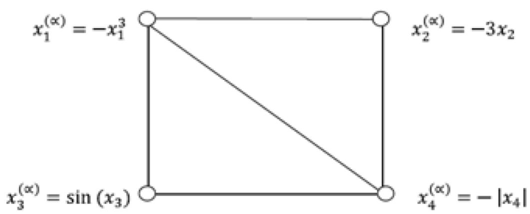

Fig. 1. Scheme of a simple network (23) with four different nodes and five non-uniform links

Fig. 2 shows the divergent phenomenon of network (23) with initial state $X(0) = (0, 0, 10, 0)^T$ and a three-time stronger coupling strength.

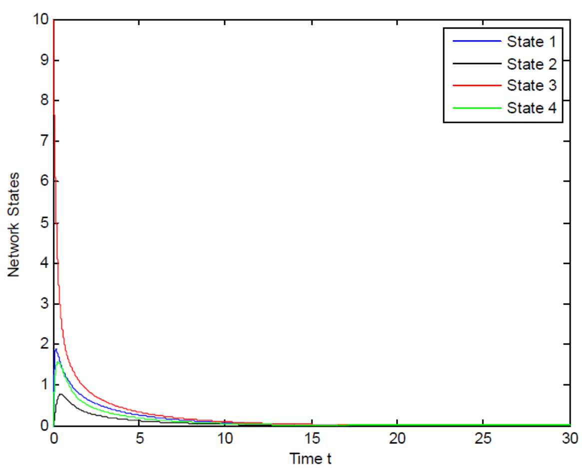

Fig. 2. The evolution of fractional network states with initial state $X(0) = (0, 0, 10, 0)^T$

The neural network was selected as

$$A_{i_n} = -I_{1\times1},\ W_{i_n} = (1)_{1\times1},\ \sigma(\cdot) = (\tanh x_{i_n})_{1\times1},$$

$$L_{\phi_{\sigma_i}} \triangleq n_i = 1,\ i = 1, 2, 3, 4, \qquad (24)$$

with initial state $X_n(0) = (0, 0, -10, 0)^T$ and $\Gamma = I_{1\times1}$.

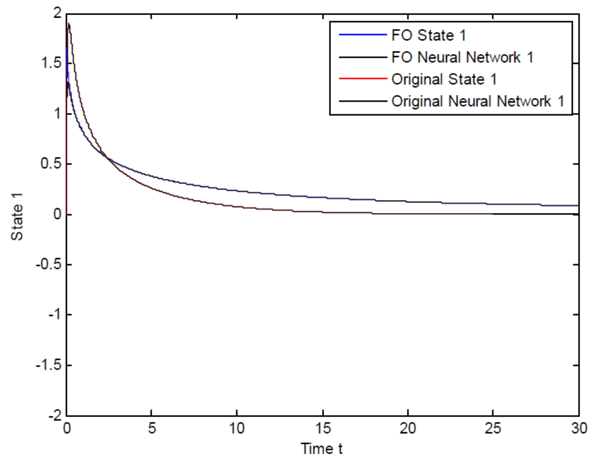

Fig. 3. Time evolution for state 1

The simulation was as follows: for the first 0.5 seconds, the two systems evolve by themselves; in this moment the control law (22) is applied.

Figure 3 represents numerical solutions of the following: 1) the dynamical system of integer order for the first state of the complex network (called original state in the figure), 2) the corresponding integer order neural network (called original state in the figure), 3) the dynamical system of fractional order for the first state, 3) the corresponding neural network.

Similar results for states 2, 3 and 4 are displayed in Fig. 4, Fig. 5, and Fig. 6, respectively. They show the time evolution for network states and the successful tracking as was expected from the general control law we obtained.

6 Conclusions

We have presented a controller design for trajectory tracking of a fractional general complex dynamical networks [15]. This framework is based on controlling dynamic neural networks using Lyapunov theory in the fractional case. We obtained a control law in a purely theoretical way, and it can be applied to a wide range of problems

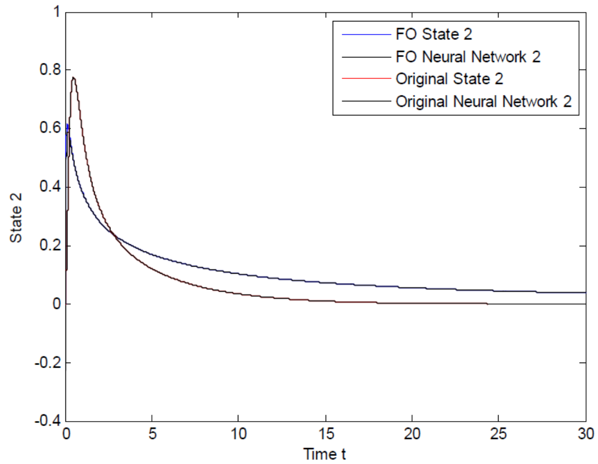

Fig. 4. Time evolution for state 2

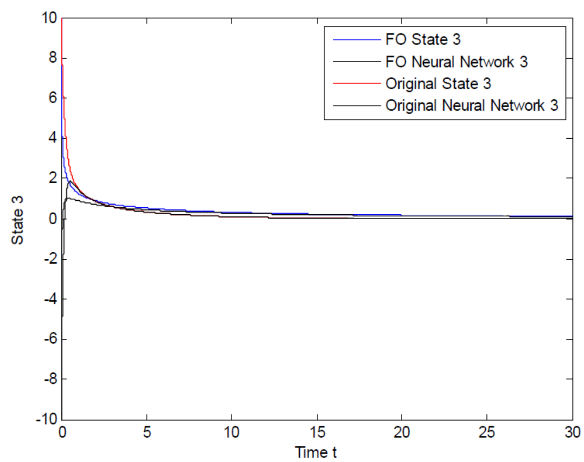

Fig. 5. Time evolution for state 3

in trajectory tracking. As an example, the proposed control is applied to a simple network with four different nodes and five non-uniform links. In future work, we will consider the stochastic case in fractional systems.

Acknowledgement

The author is thankful for the support of CONACYT and the Dynamical Systems Group of the Facultad de Ciencias Físico-Matemáticas-UANL, Mexico.

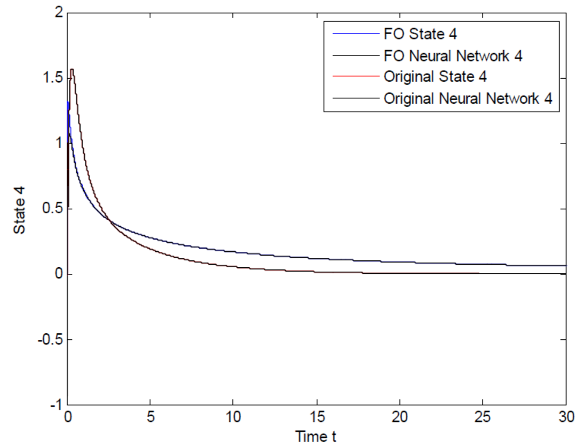

Fig. 6. Time evolution for state 4

References

1. **Abbas, S.** (**2011**). Existence of solutions to fractional order ordinary and delay diferential equations and applications. *Electronic Journal of Differential Equations*, Vol. 2011, No. 09, pp. 1–11.

2. **Aguila, N., Duarte, M., & Gallegos, J.** (**2014**). Lyapunov functions for fractional order systems. *Communications in Nonlinear Science and Numerical Simulation*, Vol. 19, No. 9.

3. **Choi, C.** (**2004**). *PID Trajectory Tracking Control for Mechanical Systems*. Springer.

4. **Li, X., Wang, X. F., & Chen, G.** (**2004**). Pinning a complex dynamical network to its equilibrium. *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 51, No. 10, pp. 2074–2087.

5. **Liping, C., Yi, C., Ranchao, W., Jian, S., & Ma, T.** (**2012**). Cluster synchronization in fractional-order complex dynamical networks. *Physics Letters*, Vol. 376, No. 35.

6. **Ma, W., Li, C., & Wu, Y.** (**2014**). Adaptive synchronization of fractional neural networks with unknown parameters and time delays. *Entropy*, Vol. 16, No. 12, pp. 6286–6299.

7. **Martínez, R., Jardón, H., Leon, J., & Fernandez, G.** (**2009**). Estabilización de redes complejas fraccionarias de sistemas de lorenz modificados y sistemas de chen. *Congreso Anual de la Asociacion de Mexico de Control Automatico*, Zacatecas, Mexico.

8. **Martínez, R., Leon, J., & Fernandez, G.** (**2010**). Sincronización parcial de redes complejas fraccionarias. *Congreso Anual de la Asociacion de Mexico de Control Automatico*, Puerto Vallarta, Jalisco, Mexico.

9. **Perez, J. & Perez, J.** (**2015**). Pid control law for trajectory tracking error using time-delay adaptive neural networks for chaos synchronization. *Computación y sistemas*, Vol. 19, No. 2.

10. **Perez, J., Perez, J., Flores, A., & Arroyo, S.** (**2014**). Complex dynamical network control for trajectory tracking using delayed recurrent neural networks. *Mathematical Problems in Engineering*, Vol. 2014.

11. **Perez, J., Perez, J., Rodriguez, F., & Flores, A.** (**2014**). Trajectory tracking for chaos synchronization via pi control law between roosler-chen. *Computación y sistemas*, Vol. 18, No. 2.

12. **Sanchez, E. & Perez, J.** (**1999**). Input-to-state stability analysis for dynamic neural networks. *IEEE Trans. on Circuits Syst. I*, Vol. 46, pp. 1395–1398.

13. **Xiang, J. & Chen, G.** (**2007**). On the v-stability of complex dynamical networks. *Automatica*, Vol. 43, pp. 1049–1057.

Joel Perez Padron obtained his B.Sc. degree in Mathematics in 1991, from the Faculty of Physical and Mathematical Sciences, a Master's degree in Electrical Engineering with specialization in Control in 2001, from the Faculty of Mechanical and Electrical Engineering, and a Ph.D. in Industrial Physics Engineering with specialization in Control in 2008, from the Faculty of Physical and Mathematical Sciences of Nuevo Leon Autonomous University.

Article received on 21/02/2015; accepted on 21/04/2016.
Corresponding author is Joel Perez Padron.