Neural Networks 57 (2014) 79–93

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Synchronization of memristor-based recurrent neural networks with two delay components based on second-order reciprocally convex approach

A. Chandrasekar [a], R. Rakkiyappan [a], Jinde Cao [b,c,*], S. Lakshmanan [d]

[a] Department of Mathematics, Bharathiar University, Coimbatore - 641 046, Tamilnadu, India
[b] Department of Mathematics, Southeast University, Nanjing 210096, China
[c] Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[d] Department of Mathematics, College of Science, UAE University, Al Ain 15551, United Arab Emirates

ARTICLE INFO

Article history:
Received 11 March 2014
Received in revised form 28 May 2014
Accepted 1 June 2014
Available online 6 June 2014

Keywords:
Synchronization
Memristor
Time-varying delays
Reciprocally convex approach
Lyapunov–Krasovskii functional

ABSTRACT

We extend the notion of Synchronization of memristor-based recurrent neural networks with two delay components based on second-order reciprocally convex approach. Some sufficient conditions are obtained to guarantee the synchronization of the memristor-based recurrent neural networks via delay-dependent output feedback controller in terms of linear matrix inequalities (LMIs). The activation functions are assumed to be of further common descriptions, which take a broad view and recover many of those existing methods. A Lyapunov–Krasovskii functional (LKF) with triple-integral terms is addressed in this paper to condense conservatism in the synchronization of systems with additive time-varying delays. Jensen's inequality is applied in partitioning the double integral terms in the derivation of LMIs and then a new kind of linear combination of positive functions weighted by the inverses of squared convex parameters has emerged. Meanwhile, this paper puts forward a well-organized method to manipulate such a combination by extending the lower bound lemma. The obtained conditions not only have less conservatism but also less decision variables than existing results. Finally, numerical results and its simulations are given to show the effectiveness of the proposed memristor-based synchronization control scheme.

1. Introduction

Pecora and Carroll declared a pioneering work on synchronization control of chaotic systems in the precedent years (Carroll & Pecora, 1991; Pecora & Carroll, 1990). They introduced a method to synchronize two identical chaotic systems with different initial conditions. Enormous efforts were guided to build up the new chaotic system (Pan, Zhou, & Fang, 2010) and propose assortment of methods to accomplish chaos control and synchronization for chaotic systems and chaotic neural networks (Liu, 2009) during the past two decades. Recently, the synchronization of chaotic neural networks has been intensively investigated due to their potential application in technological practice, such as secure communication (Liao & Tsai, 2001), image processing (Perez-Munuzuri, Perez-Villar, & Chua, 1993), harmonic oscillation generation and also exhibits synchronization in language emergence and development, which comes up with a common vocabulary, while agent's synchronization in association management will improve their work effectively. On the other hand, the synchronization in coupled identical delayed neural networks has been shown to have an important impact on the fundamental science (e.g., the self-organization behavior in the brain). In current years, the synchronization analysis for the complex networks which creates dissimilar kind of Neural Networks (NNs) has come into view as a research topic of primary significance. Research on synchronization control of chaotic systems has gained considerable attention in recent years due to their strong background in applications. For this reason, it has a wider importance to study the synchronization of chaotic NNs that has been proposed in the following literature: (see Balasubramaniam, Chandran, & Theesar, 2011; Cao, Alofi, Al-Mazrooei, & Elaiw, 2013; Cao, Chen, & Li, 2008; Cao & Li, 2009; Cao, Wang, & Sun, 2007; Gan, Xu, & Kang, 2011; Sun, Wang, Wang, & Cao, 2010; Zhu & Cao, 2012).

* Corresponding author at: Department of Mathematics, Southeast University, Nanjing 210096, China. Tel.: +86 25 83792315; fax: +86 25 83792316.
E-mail addresses: chandru7389@gmail.com (A. Chandrasekar), rakkigru@gmail.com (R. Rakkiyappan), jdcao@seu.edu.cn (J. Cao), lakshm85@gmail.com (S. Lakshmanan).

http://dx.doi.org/10.1016/j.neunet.2014.06.001
0893-6080/

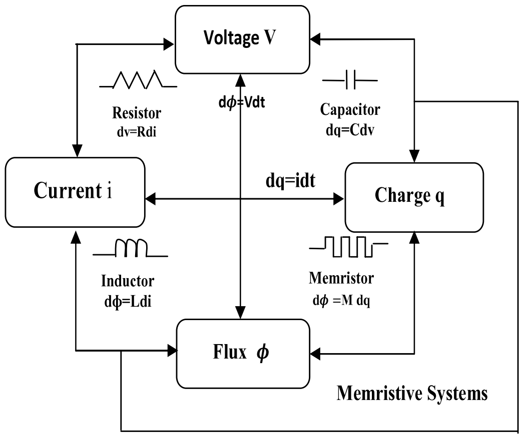

Fig. 1. The relation between four fundamental elements.

The utilization of time delay in the feedback loop eradicates the need for unambiguously determining some information about the fundamental dynamics other than the period of the desired orbit. In various applications, the fascinating problem is to design a memory less state-feedback controller $u(t) = K_1e(t)$, (see Cheng, Liao, & Hwang, 2005; Gao, Zhong, & Gao, 2009). Moreover, it has now been well realized that time delays always influence the dynamic properties of delayed chaotic NNs, which may cause periodic oscillations, bifurcation and chaotic attractors and so on. Therefore, time delays should be modeled in order to simulate more realistic networks. Thus if the information on the dimension of the time-varying delay $\tau(t)$ is available, then a delayed feedback controller of the form: $u(t) = K_1e(t) + K_2e(t - \tau(t))$ has been considered in the literature (see Li, Fei, Zhu, & Cong, 2008). In addition, an exceptional case where the information on the size of the time-varying delays $\tau(t)$, $\rho(t)$ is accessible, we also consider a delayed feedback controller of the form: $u(t) = K_1e(t) + K_2e(t - \tau(t)) + K_3e(t - \rho(t))$, (see Song, 2009). Therefore, the more common form of a delayed feedback controller is $u(t) = K_1e(t) + K_2\int_{t-\tau(t)}^{t} e(s)ds$ (see Park & Kwon, 2006; Yu & Cao, 2007). But in many real network papers only output signals can be measured (see Li & Bohner, 2010; Li, Ding, & Zhu, 2010) and it is of the form: $u(t) = K_1(f(y(t)) - f(x(t))) + K_2(f(y(t - \tau(t))) - f(x(t - \tau(t))))$. Motivated by the above discussion, in this paper we include distributed delay in the delayed output feedback controller and the controller takes the form $u(t) = K_1(f(y(t)) - f(x(t))) + K_2(f(y(t - \tau_1(t) - \tau_2(t))) - f(x(t - \tau_1(t) - \tau_2(t)))) + K_3\int_{t-\rho(t)}^{t}(f(y(s)) - f(x(s)))ds$ in the slave system, where $K_1, K_2, K_3 \in R^{n\times n}$ constant gain matrices. The survival of memristor which describes the relationship between electric charge and magnetic flux was predicted in 1971 by Chua (1971). The memristive functionality is not a distinctive property of two-terminal passive devices but essentially a memory effect related to internal state variable changes. This symmetry follows from the depiction of basic passive circuit elements as defined by a relation between two of the four fundamental circuit variables, namely voltage, current, charge and flux. A device linking charge and flux (themselves defined as time integrals of current and voltage), which would be the memristor, was still hypothetical at the time is clearly depicted in Fig. 1. Furthermore, their microscopically adapted internal state is simply measured as an external two-terminal resistance. Memristors were originally defined as components that relate charge and magnetic flux, but they can be further usefully described as devices with a pinched-hysteresis loop whose size is frequency dependent. This new circuit element will be helpful for low-power computation and storage to store information and data without the need of power (Ventra, Pershin, & Chua, 2009). In this way, the memristor remembers information. From the previous works (Bao & Zeng, 2013; Corinto, Ascoli, & Gilli, 2011; Itoh & Chua, 2008; Pershin & Di Ventra, 2010; Strukov, Snider, Stewart, & Williams, 2008; Wu & Zeng, 2012a; Wu, Zeng, Zhu, & Zhang, 2011; Yang, Cao, & Yu, 2014; Zhang, Shen, & Sun, 2012; Zhang, Shen, & Wang, 2013), we know that the potential applications of this device is in next generation computers and powerful brain-like neural computer. Additionally, the scrutiny of the memristor-based recurrent neural networks is able to disclose critical characteristics of the dynamics, such as the occurrence of sliding modes along switching surfaces, the chaos synchronization, and the ability to work out the accurate global minimum of the underlying energy function, which make the networks specially attractive for the solution of global optimization problems in real time.

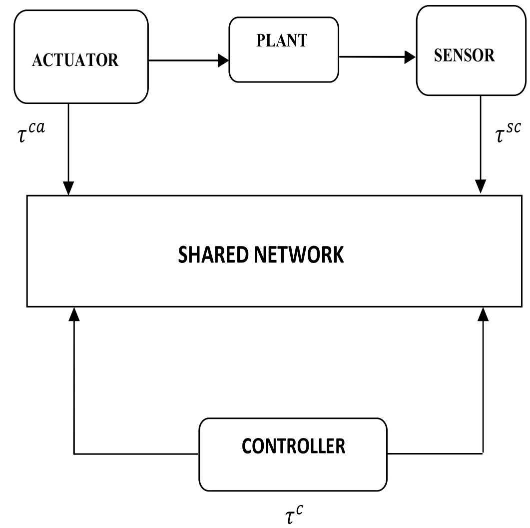

Fig. 2. Delays in networked control system.

The study of time-delay systems also called systems with after-effect or dead-time, hereditary systems, equations with deviating argument or differential-difference equations has received widespread consideration above the precedent years. While time-delays are the most important origin of oscillation, divergence or instability, substantial efforts have been made to stabilize the systems with time delays. In present years, a lot of efforts have been endowed in the analysis of time-delay systems, such as delayed stochastic system (Wang, Liu, & Liu, 2010), delayed BAM neural networks (Cao & Wan, 2014), delayed stochastic genetic regulatory networks (Wang, Wang, & Liang, 2010), and delayed stochastic complex networks (Wang, Wang, & Liang, 2009; Wang, Wang, & Liu, 2010). Motivated by recent efforts on time-delay systems, a new model for neural networks with two additive time-varying delays has been measured in Shao and Han (2011) and Zhao, Gao, and Mou (2008). We consider time-delays in the dynamical model as $\dot{x}(t) = Ax(t) + BKx(t - d_1(t) - d_2(t))$ where, $d_1(t)$ is the time-delay induced from sensor to controller and $d_2(t)$ is the delay induced from controller to the actuator. Then, from the mathematics point of view, $d_1(t)$ and $d_2(t)$ are lumped as one delay $d(t) = d_1(t)+d_2(t)$, then the system becomes $\dot{x}(t) = Ax(t)+BKx(t-d(t))$. One simple example is shown in Fig. 2, which can easily explain the concept of additive time-varying delays. In Fig. 2, it will be seen that there are basically three kinds of delays: τ^{sc} is used to represent the delay from sensor to controller, τ^{c} is the computational delay, and τ^{ca} is the delay from controller to actuator. Because of this feature, extensive potential applications of additive time-varying delays have been identified in the following literature (see Gao, Chen, & Lam, 2008; Lam, Gao, & Wang, 2007, and the references therein).

In this paper, we make great efforts to investigate the synchronization of memristor-based recurrent neural networks with two