Neural Networks 104 (2018) 15–25

Contents lists available at ScienceDirect

Neural Networks

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

Bio-inspired spiking neural network for nonlinear systems control

Javier Pérez *, Juan A. Cabrera, Juan J. Castillo, Juan M. Velasco

University of Malaga, C/ Ortiz Ramos s/n, 29071 Malaga, Spain

ARTICLE INFO

Article history:
Received 9 October 2017
Received in revised form 8 February 2018
Accepted 3 April 2018
Available online 12 April 2018

Keywords:
Spiking neural network
Supervised learning
Genetic algorithm
Nonlinear systems
Control

ABSTRACT

Spiking neural networks (SNN) are the third generation of artificial neural networks. SNN are the closest approximation to biological neural networks. SNNs make use of temporal spike trains to command inputs and outputs, allowing a faster and more complex computation. As demonstrated by biological organisms, they are a potentially good approach to designing controllers for highly nonlinear dynamic systems in which the performance of controllers developed by conventional techniques is not satisfactory or difficult to implement. SNN-based controllers exploit their ability for online learning and self-adaptation to evolve when transferred from simulations to the real world. SNN's inherent binary and temporary way of information codification facilitates their hardware implementation compared to analog neurons. Biological neural networks often require a lower number of neurons compared to other controllers based on artificial neural networks. In this work, these neuronal systems are imitated to perform the control of non-linear dynamic systems. For this purpose, a control structure based on spiking neural networks has been designed. Particular attention has been paid to optimizing the structure and size of the neural network. The proposed structure is able to control dynamic systems with a reduced number of neurons and connections. A supervised learning process using evolutionary algorithms has been carried out to perform controller training. The efficiency of the proposed network has been verified in two examples of dynamic systems control. Simulations show that the proposed control based on SNN exhibits superior performance compared to other approaches based on Neural Networks and SNNs.

1. Introduction

The use of systems based on Artificial Neural Networks (ANN) is increasingly spreading in the field of engineering. Their main applications are pattern recognition, data classification and prediction. A further use of ANN is the modeling and control of dynamic systems. Recently, research groups have begun to use ANN models inspired by the real behavior of biological neural networks. These types of networks have been called spiking neural networks (Gerstner & Kistler, 2002). This new approach offers a model closer to reality than previous generations of ANN.

The first generation of ANNs basically made use of McCulloch–Pitts threshold neurons, i.e. the neuron output only consisted of high or low levels. Second generation neurons use a continuous activation function to compute their output signals (a sigmoid function for example). The main difference between the 1st and 2nd generation of ANNs and SNN is the fact that the latter incorporate the concept of time into their operating model. In these networks, a spiking train between neurons, encodes and controls the system variables. Due to the time dependency of the variables, SNN is a suitable candidate for control of dynamic system (Meng, Wang, & Wang, 2017).

However, SNN has not been widely used in control schemes because of the complexity of the neuron model. The mathematical model of these neurons makes use of differential equations to obtain the state of the neuron at each time of the simulation. Therefore, a higher computational effort is required compared to ANN.

As a result, few works can be found in the literature in which SNN are applied to control system. For example, in Webb, Davies, and Lester (2011), the authors use spiking neurons to simulate a PID control. The authors claim that neurons approximate proportional, derivative and integral errors adequately. Despite that, they do not include a real application of control of a dynamic system using the PID neuron model. In Chadderdon, Neymotin, Kerr, and Lytton (2012) and Spüler, Nagel, and Rosenstiel (2015), the authors use a SNN to control an arm with 1 degree of freedom (DOF). To this purpose, the authors simulate the cortex, which is the region of our brain in charge of the control of voluntary movements, to control the arm. In Bouganis and Shanahan (2010), the authors present a SNN-based architecture to control a 4 DOF robotic arm. The structure is based on a large number of sensory neurons (1200 per signal) connected to the output or motor neurons (800 neurons per output). A similar application can be found in Carrillo, Ros,

* Corresponding author.
E-mail address: javierperez@uma.es (J. Pérez).

https://doi.org/10.1016/j.neunet.2018.04.002
0893-6080/

Boucheny, and Olivier (2008), where the authors aim to simulate the behavior of the cerebellum. As an application example, they use a SNN network to control a 2 DOF arm. Finally, another interesting work is a parallel architecture called SpiNNaker with more than one million neurons that can be used to control dynamic systems (Jin et al., 2010).

Contrary to classical control systems like PID or a fuzzy controller, in all these papers in which the control is carried out by means of a SNN, there is not a fixed control structure. In addition, they use complex neural networks structures with a high number of neurons and several layers. However, in nature there are examples of simple control with less than 10 neurons. In this case, a few input neurons are directly connected to other output neurons. For example, in Kandel (2001), the authors study the neural structure of a simple animal called Aplysia. They found that there are simple neural circuits to activate the gill. Furthermore, the connections between sensor and motor neurons are direct in these circuits. In Braitenberg (1984), a simple neural network is used to control vehicles. Vehicle behavior exhibits great dependence on the weights and type of connections between neurons. Another example of simple control neural networks can be found on the spinal cord, where a few neurons control the reflex acts of the human body (Balderas & Rojas, 2016), as happens with the patellar act. These neurons are completely isolated from the brain, performing a fast control action.

Evolution tends to reduce the size of neural networks that govern movements in living beings by optimizing their control function. Natural neuronal systems have been improving for millions of years which has enhanced their operation and reduced their energy demand. Nature shows that, in some cases, it is not necessary to increase the number of neurons to perform a complex control. SNNs take advantage of simulating natural systems to perform a faster and improved control compared to previous ANN approaches. Therefore, these systems are capable of carrying out the required control tasks with minimal resources and energy consumption.

Furthermore, the learning process is another important issue to be studied in this type of neural networks. Learning is a very important feature in biological neural networks since it allows a fast response to events that appear unexpectedly. Literature focuses on two types of learning processes: supervised and unsupervised. In unsupervised learning, biology-based learning rules are established, such as Spiking Timing Dependent Plasticity (STDP) (Froemke & Dan, 2002) or Bienenstock Cooper Munro (BCM) (Bienenstock, Cooper & Munro, 1982). These learning rules are based on the reinforcement or weakening of synaptic connections. On the contrary, in supervised learning, a set of training data consisting of pairs of input objects and the desired output values are required. The neural network must learn to predict the correct output for any valid input. There are several methods to perform this type of learning, such as Error Back Propagation (Bohtea, Koka, & La Poutr, 2002), Supervised Hebbain Learning (SHL) (Ruf & Schmitt, 1997), Remote Supervision (Ponulak & Kasiński, 2010) or the use of Evolutionary Algorithms (Belatreche, Maguire, McGinnity, & Wu, 2003). In this work, supervised learning based on genetic algorithms is employed. Despite a higher computational cost, the main advantage of using genetic algorithms in this application is that they can provide a nearly global optimal set of weights and neuron types in a reasonable time with simple programming.

Finally, the stability and robustness of the proposed control system, based on neural models of simple living beings and taught through supervised learning, must be verified. Two systems of different degrees of complexity, according to the number of differential equations that model them, have been chosen to test their performance. The first application is a model of a DC motor that includes the nonlinearities of the system. In the second example, the arm model described in Winters and Stark (1985) is controlled. In this case, arm movement is managed by different muscles coordinated by an activation signal (Chadderdon et al., 2012; Hulea & Caruntu, 2014).

The novelty of this work lies in the use of a reduced control structure based on spiking neural networks replicating biological control systems to control industrial applications. A further advantage of using a minimal number of neurons is that it enables knowing the function of each one of its components. This advantage makes it possible to adjust the parameters of the network (Arena, Fortuna, Frasca, & Patané, 2009) allowing to search for a concrete kind of action. The implementation in a real time control system is assured thanks to the simple execution code achieved and by the absence of filters and delays.

The remainder of the paper is organized as follows: Section 2 is devoted to the description of the neuron model. Next, the development of the control system based on SNN is included in Section 3. Section 4 presents the supervised learning process carried out using evolutionary algorithms. The performance of the whole system is verified through simulations included in Section 5. Finally, conclusions are drawn in Section 6.

2. Spiking neural network model

The main goal of this work is to implement a neural network with a reduced number of neurons in which the overall network performance can be known a priori. To do so, a description of the neuron model as well as connections between the neurons of the network, called synapses, is included next.

2.1. Neuron model

The choice of the neuron model is a subject of debate (Izhikevich, 2004). There are several models with different levels of complexity that reproduce biological neurons with lower or higher levels of accuracy. In this paper, the neuron presented by Izhikevich (2003) is utilized due to its good balance between computational cost and accuracy. Besides, its use is the most widespread within research groups in this area.

The Izhikevich neuron model is defined by two variables, u and v. Variable v is the potential of the membrane. It can be obtained from the following equation.

$$\frac{dv}{dt} = 0.94v^2 + 5v + 140 - u + I(t) \tag{1}$$

where I is the input current from the previous neuron synapses. Variable u is the recovery potential, which determines the response period. It can be obtained from Eq. (2).

$$\frac{du}{dt} = a(bv - u) \tag{2}$$

where a and b are parameters that define the neuron type. c defines the reset potential of the membrane after a spike. d is parameter to be added to the recovery variable after the spike takes place.

$$If\ v \geq 30\ then \begin{cases} v \leftarrow c \\ u \leftarrow u + d. \end{cases} \tag{3}$$

Parameters a, b, c and d fully define the spiking neuron model. This way, it is only necessary to describe the connections between them to have a complete neural network.

2.2. Synapse model

Synapses play a highly significant role in the neural network. They control the spike flow between neurons as well as the effect that they produce in the post-synaptic neuron. This way, an exciting action in the synapse can cause a fast trigger in the neuron, or,