Use of spiking neural networks for robot navigation

Rafael G. Mantovani¹, Pedro Paulo da S. Ayrosa²

¹Instituto de Ciências Matemáticas e Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos, SP – Brazil

²Departamento de Computação (DC)
Universidade Estadual de Londrina (UEL)
Londrina, PR – Brazil

rgmantov@icmc.usp.br, ayrosa@uel.br

Abstract. Detecting and avoiding possible collisions is one of the most important aspects in mobile robotics. Although this task seems easy when performed by animals (and humans), it presents difficulty when modeled and executed by autonomous robotic agents. With increasing focus on computational neuroscience in recent years, a new range of bioinspired models and algorithms have emerged in the specific literature, such as the Spiking Neural Networks (SNN). These models enhanced the biological realism of their computational units using 'individual spikes', allowing the incorporation of spatial and temporal information in the processes of communication and computing, like real neurons do. This paper deals with an application of spiking neural networks in autonomous mobile robot control. Physical experiments were done using Lego Mindstorms NXT, while prototyping of the model used Microsoft Robotics Developer Studio (MRDS) which, besides providing a virtual environment for simulation, has resources for the programming of a variety of robotics hardware, including the NXT.

keywords: Spiking neural network, autonomous robotics, trajectory control.

1. Introduction

Detecting and preventing collisions is an important aspect in mobile robotics. For an autonomous robot that performs a navigation task, detection and avoidance of obstacles is a key issue for its own security [Soumare et al. 2002]. Although the prediction of collisions is a relatively simple task when performed by living beings, it has some difficulty when performed by robots, being a well researched topic in robotics. The question of how biological systems transform all sensitive information in motion is a constant investigation for many neuroscientists and experts in robotics [Horiuchi 2009].

Different bioinspired mechanisms are currently used in robotics, taking advantage of the characteristics of neural cells such as ones found in the hippocampus and parahippocampal region of the brain, used to implement location and navigation capabilities in biological agents [Arena et al. 2009]. Neuroscientific research found that the hippocampus plays a key role in the control of navigation in animals, encoding spatial positions and acting as a mechanism of associative memory [Burgess and O’Keefe 1996], [Burgess et al. 2001] and [Jensen and Lisman 2005]. The researchers could observe that communication between neurons could be done using the timing of the spikes emitted...
by neurons of a synapse, a process subsequently formalized and named "Spike-timing-dependent plasticity" or STDP.

Spurred by advances in technology and neuroscience, many works that employ Spiking Neural Networks (SNN) [Mass and Bishop 2001] can be found in the literature. These neural models focus on the mathematical formalization of the computational properties of biological neurons and many successful applications have been found covering areas such as: computer vision [Christodoulou et al. 2002] [Escobar and AL 2009]; image segmentation [Rowcliffe et al. 2002] [Buhmann et al. 2005]; forecasting time series [Burgsteiner et al. 2007]; speech recognition [Maass et al. 2002] [Verstraeten et al. 2005]; control of robotic arms [Joshi and Maass 2005] [Rowcliffe and Feng 2008] and autonomous navigation [Horiuchi 2009], [Arena et al. 2009], [Stratton et al. 2011] e [Nichols et al. 2013]. This work aims to investigate the use of a spiking neural network to control the navigation of an autonomous robot during an obstacle avoidance task. As SNNs incorporate biological plausibility representing and manipulating all the information in the form of spikes, the use of such models can provide us a better understanding of existing biological models. This approach is interesting for a robotic community as an alternative approach for navigation of robots in known environment.

2. Spiking Neural Model

SNN are a machine learning paradigm that emerged during the 1990s. They focus on the use of biologically realistic neuron models and information coding schemes. SNN are closely related to computational neuroscience. Computational neuroscience aims at the realistic representation of the brain, and began a discipline when Alan Hodgkin and Andrew Huxley published their study of a squid giant axon [Hodgkin and Huxley 1952]. Since then, many more neuron models have been created, but the Hodgkin-Huxley neuron model is the model that most accurately reproduces the features of a real neuron. However, this model has a high computational cost, since it is described by four non-linear differential equations, which prevents its use in simulations on a network with a large number of neurons. On the other hand, the models of the type integrate-and-fire [Dayan and Abbott 2001] are computationally effective but too simple and unable to play certain types of dynamics exhibited by cortical neurons. Contextualized with such reality, Izhikevich [Izhikevich 2003] developed a spiking neuron model that merits in presenting the biological plausibility of Hodgkin-Huxley model, besides being computationally efficient as integrators models. Through a simple set of parameters the model can reproduce the potential activation of known types of cortical neurons.

2.1. Neuron Model

The Izhikevich’s model consists of a two-dimensional system of ordinary differential equations

\[
\begin{align*}
\dot{v} &= 0.04v^2 + 5v + 140 - u + I \\
\dot{u} &= a(bv - u)
\end{align*}
\]

with an after-reset-spike given by
if \( v \geq 30 \text{ mV} \), else \[
\begin{align*}
v &\leftarrow c \\
u &\leftarrow u + d.
\end{align*}
\] (3)

where the variables \( v \) and \( u \), and the parameters \( a, b, c \text{ e } d \) are dimensionless. This model is well studied in [Kampakis 2012].

![Diagram](image)

**Figure 1.** Excitable Class I neurons: encoding the distance between the robot and an obstacle through a sequence of spikes (Arena et al., 2009).

A neural dynamics that catches attention is the dynamics found in the Class I type neurons, located in the hippocampus and responsible for tasks of movement and spatial-localization [Dayan and Abbott 2001]. The main feature of these neurons is the fact that they have a characteristic response through spikes which is proportional to the amplitude of the input signal (electric current value) they receive. This is important because it acts as a coding mechanism, encoding any measure as a characteristic frequency of issued spikes, as depicted in fig.1. The Izhikevich model parameters used to represent the behavior of a Class I neuron are: \( a = 0.02, b = -0.1, c = -55 \text{ and } d = 6 \) [Izhikevich 2004], while the input variable \( I \) takes the values of both external stimuli (sensory stimuli) as the values of synaptic inputs. These types of neurons will be exploited to model the spatial navigation task of an autonomous agent.

### 2.2. Learning Rule - STPD

Recent experimental and theoretical studies indicate the existence of a mechanism of information processing based on the precise timing of the spike issued from brain neurons. A persistent change in synaptic efficacy depending on the relative timing of pre and postsynaptic spikes is a phenomenon known as *spike-timing-dependent plasticity*, or simply STDP [Florian 2007]. Through this mechanism external stimuli can be encoded using the relative time between spikes of pre and postsynaptic neurons [Hosaka et al. 2008]. This mechanism of synaptic learning is biologically plausible, and has been observed in biological neural systems, especially in CA1 pyramidal neurons of the hippocampus [Dan and Poo 2004]. This form of synaptic modification can automatically balance the synaptic connections to make postsynaptic neurons to fire irregularly, but more sensitive to the time of presynaptic spikes [Song et al. 2000]. This synaptic learning algorithm was adopted to make changes in SNN synapses.
Let \( w \) a variable that represents a synaptic weight. Pre- and postsynaptic action potentials change the weight \( w \) \((w \rightarrow w + \Delta w)\) by means of a function \( F(\Delta t) \). \( \Delta t \) represents the difference between the times of pre- and postsynaptic spikes:

\[
\Delta t = t_{pre} - t_{post}
\] (4)

Once defined the values of \( \Delta t \), we can calculate the synaptic adjustment by the STDP rule itself [Song et al. 2000]:

\[
\Delta w = \begin{cases} 
A_+ \exp^{\Delta t/\tau_+} & \text{if } \Delta t < 0 \\
A_- \exp^{-\Delta t/\tau_-} & \text{if } \Delta t \geq 0 
\end{cases}
\] (5)

The parameters \( \tau_+ \) and \( \tau_- \) determine the variation of the interval between spikes in which synaptic changes of strengthening and weakening occur. The other parameters, \( A_+ \) and \( A_- \), determine the maximum amount of synaptic modification (\( \Delta w \)) that occur when \( \Delta t \) is close to zero [Benuskova and Abraham 2007].

3. Structure Control and Simulation

Our proposed approach was designed to be used with the Lego Mindstorms NXT robotics kit (henceforth referred as NXT). The communication between our desktop computer and the NXT robot took place via Bluetooth technology, which reacted sufficiently fast for this implementation of neural network. This scenario enabled us to create applications for the NXT on a desktop computer while maintaining robot mobility, i.e. we could use computing and memory capacity of the desktop and made full use of robot features at the same time. Figure 2 depicts the communication between the NXT and the desktop computer. It represents the implementation of a SNN, which receives coded information from touch and ultrasonic sensors. The communication takes place in real time via Bluetooth. The transmission itself takes place during each sensory-control cycle of networks activity as the system receives current information from sensors and directs the movement of the robot accordingly. The wireless information transfer is shown as a dashed line. The applied type of communication does not limit mobility of the robot. When activated, the proposed navigation system initiates the communication between desktop computer and robot. Before becoming inactive, communication is terminated.

The robot used in the trials was a tribot (three wheels). The motor activity of the wheels is directly controlled by the output of the spiking neural network. All the implementations were carried out using two ultrasonic sensors and two touch sensors located on the front of the agent, taking a range of perception of \([0, +45 \text{ degrees}]\) (for the right sensors) or \([-45 \text{ degrees}, 0]\) (for the left ones). The distance between the robot and the nearest object \((d_0)\) is measured in ”robot units (r.u.)” [Arena et al. 2009], and are used to compute the input stimuli of each sensor.

The collision sensors are activated when this distance is equal to \( d_0 = 0.6 \text{ r.u.} \), in this case the input associated with the stimulus is defined as a constant value of \( I = 9 \). This value causes the generation of regular spikes, allowing the robot to avoid an obstacle by changing its movement in one direction. For the ultrasonic sensors used for predictions, the stimulus associated with the input of sensory neurons is described by an exponential function presented in [Arena et al. 2009]:

\[
I = 9 \cdot \exp^{-0.6 \cdot d_0} + 2.2.
\] (6)
Using this function the neurons start firing when the distance between the robot and the closest object is less than $11$ r.u.

The robot moves in an environment filled with randomly spaced obstacles. Each step of the network execution simulates a time window of $300$ ms. The robot moves are generated according to the number of spikes fired by motor neurons (from the output layer) during this time interval. The number of spikes emitted by motor neurons is combined to generate the control action of the robot. The signal that controls every engine depends on this number of spikes emitted by motor neurons associated to them.

The network structure and connections between neurons is inspired by Braitenberg vehicles [Braitenberg 1984]. Such models are simple reactive vehicles equipped with a bioinspired control mechanism based on direct coupling between sensors and motors. This type of model is biologically relevant as it has been used to model the nervous system of crickets in the task of recognition of sound sources (cricket phonotaxis) [Webb and Scutt 2000]. The effect of excitatory synapses is to depolarize the postsynaptic potential, while the inhibitory synapses one is the reverse, i.e., hyperpolarize this potential. When there is no external stimuli sensed by contact sensors, there is a constant (which can be seen in fig.2) that is inserted in the electric current variable of motor neurons and ensures the neural activity responsible for the robot’s movement.

4. Experiments and Results

This section presents a discussion of various aspects related to the simulations and physical experiments carried out and the results as well. We created a simulation arena for testing and prototyping. The simulated environment is a fully enclosed arena with obstacles placed randomly. The initial position of the robot in this space is also random (fig.3). Thus, the aim is to design a robot that navigates in this space avoiding obstacles and walls...
using a SNN.

4.1. Simulated Trials

To ease implementation process and better error handling on agent navigation, Microsoft Robotics Developer Studio (MRDS) [Hung et al. 2008] was used. Apart from allowing programming and control over the robotics kit used, the MRDS also provides an engine with physical environment for three-dimensional simulations. Model validation carried out 10 different scenarios of navigation along the simulations. In such scenarios the initial position of the agent is located randomly, and the number of obstacles is $4 \leq obs \leq 7$. The length of the sonar perception is $10 \ r.u.$ Each scenario was simulated for a period of time of $T = 25,000$ simulation steps. During the simulations and experiments the synaptic weights of nonconditioned synapses (touch sensors) were fixed at $-8$ (inhibitory synapses) and $+8$ (excitatory synapses). The initial values of the synapses of ultrasonic sensors are initialized at $0.05$ (excitatory synapses) and $-0.05$ (inhibitory synapses).

![Virtual simulation environment.](image)

Figure 3. Virtual simulation environment. (Left) Frontal view of the robot. (Right) Simulated agent positioned close to obstacles.

To illustrate the performance of the robot in the navigation task of obstacle avoidance we use two performance measures: the number of predictions and collisions during time, and the average distance when the robot turns to a direction. A motion turn is the robot’s response to an external stimulus triggered by a conditioned stimulus (sonar information) or by an unconditioned stimulus (contact sensor information). To test the behavior of the neural model we analyze the amount of turns generated during the simulation. A change of direction by the information of sonar feature is called a ”prediction”, and a turn as a result of information from contact sensors feature is called a ”collision”. If the neural model performs well during the simulation, the number of predictions will grow, while the number of collisions will decrease.

Figure 4 shows the behavior described by the robot in an observed simulation scenario. The left diagram shows the trajectory of the robot during the learning phase in which the robot avoids obstacles only through collisions. The right one shows that after adaptation of the synaptic weights the robot can avoid the presence of obstacles using the information from ultrasonic sensors. The paths [1-2], [3-6] and [7-9] in the right figure show the preventive behavior of the robot, using the stimuli received by sonar.
The chart depicted in figure 5 shows us a comparison between the number of collisions and predictions executed by the agent in all the simulated scenarios. These numbers are calculated every 1000 simulation steps during robot navigation. Over the simulation, the number of collisions decreases while the number of predictions grows, indicating that the robot learned the task of prediction using ultrasonic sensor information.

Another performance measure used in trials is the ”average evasive distance” which is the average distance that a preemptive move is performed by the robot. Every time a rotation is detected, it stores the distance of the robot to the nearest object, and average value is estimated every 1000 simulation steps. It is expected that through the robot navigation the average distance grows and converge to a value near the maximum range of sonar.

Figure 6 shows the behavior observed by the evasive average distance calculated over all simulated scenarios. It may be noted that when the robot starts moving, due to a larger number of collisions, the distance of the maneuvering is relatively small, with val-
ues between 0 e 0.4 m. From the moment that learning begins to hold its first predictions, the evasive distance observed grows to a value between 0.8 e 1.2 m. Finally, after stabilization of the synaptic weights, the distance value increases and approaches the length perception of sonar (2.0 m), with values between 1.6 e 1.9 m and thus characterizing a safe navigation.

4.2. Physical experimentations

Subsequent to the simulations physical experimentations of the neural model were performed. The navigation control agent through SNN was conducted in five different scenarios, where the number of obstacles ranged from 0 and 10, and the agent was placed in each corner of the arena, and once in the center, totaling five scenarios. Each experiment contains a total of 5000 simulated steps, where each step is equivalent to an action taken by the robot.

The figure 7 shows the behavior of the agent before and after synaptic learning. In the figure it is characterized the change in trajectory performed by the robot as a result of modifications of the synaptic weights of the network over the robot’s movement. A move initially characterized by changes of direction that are very close to obstacles (fig.7 (left)) becomes a movement driven by information obtained by the pair of sonars (fig.7 (right)).

The chart with the performance of robot navigation over the five tested scenarios is shown in figure 8. Among the steps [0000-2000] a high number of collisions and the lack of predictive movements can be noted. Approximately in the range of steps [2500 - 2600] the weights begin to adjust by increasing the number of evasive movements, and reducing slightly the number of collisions. In the interval [4000 - 5000] a sharp drop in the number of collisions can be distinguished, compared to previous times. At the same time, the distance at which avoidance occurs increases, characterizing ‘safer’ movements. As the number of predictions grows, this distance is increased approaching the maximum range of the sonar (as shown in Fig. 9).
4.3. Comparing results of simulations trials and physical experimentation

Although the behavior observed during physical experiments are within expectations, it was possible to notice a difference in the number of collisions recorded at the end of the trials in relation to data obtained at the end of simulations (figures 5 and 8). In simulated experiments the number of collisions decreases sharply and remained at a very low value, "featuring an almost ideal situation". Moreover, in physical experiments, although it has decreased, the number of collisions remains considerably high, suggesting the interference of some factor (or factors) linked directly to the experiment, such as: questions regarding the battery of the robot, which can wear out and interfere with the performance of sensors, actuators; interference from the environment used for physical experimentation; etc. Despite the data obtained by physical experiments are not "ideal", the behavior approaches what was seen in the simulations, and both sets of tests were able to decrease the number of collisions and increase the number of predictions.
5. Conclusion

In this paper we investigate the use of a SNN to control a robotic agent in a task of obstacle avoidance. It was applied the Izikevich neuron model, that besides having a non-expensive implementation can play a wide range of behavior of nerve cells using only a small set of parameters. The network model and the synaptic learning algorithm used (STDP) incorporate biological plausibility of the structure model, both for representation and manipulation of information to the network via spikes.

In order to validate the model, simulated and physical scenarios were created with robot and obstacles placed randomly within a maze completely surrounded. If the SNN could play his role well, the robot could avoid obstacles using the information of sonar instead of the information provided by the touch sensors. We have seen over the trials that after a learning period, which varied depending on the configuration of the scenario, the number of collisions detected by using touch sensors decreased, while the number of predictions generated by the sonar information grew, featuring obstacle avoidance. The relative distance between the robot and obstacles while turning in a direction has also increased. Initially this value was relatively low, since collisions were detected. While the synaptic weights were adjusted by the STDP algorithm the distance increased significantly as to avoid an obstacle. This is a strong indicative that the SNN learns well and use the information of sonar with a satisfactory accuracy to predict collisions.

During the experiments, in some situations the number of predictions generated was excessive. Further investigation and alternatives for solving this problem are also needed. An important next step for this work is the comparison with existing models in the literature, in order to compare and demonstrate the performance of the proposed model. Another future contributions include some changes and improvements in the model, like modify some learning aspects (insert delay between spikes, optimize parameters, etc) or explore other neuron models as the Kasabov probabilistic model [Kasabov 2010].
Acknowledgement

The authors would like to thank CAPES (Brazilian Agency) for their financial support.

References


