NEURAL NETWORKS FOR LOCOMOTION IN ROBOTS, AND DIGITAL COUNTERS

ADRIAN RENDON-NAVA

School of Engineering, Robert Gordon University, Aberdeen, UK
E-mail: a.rendon-nava@rgu.ac.uk

Abstract - This document presents the simulation of an Artificial Neural Network implementing a digital counter which could be used as a Central Pattern Generator (CPG) for locomotion in robots. The counter was designed by using an implementation of T flip-flops and NAND gates with neurons. Inhibitory and excitatory neurons were used for this simulation employing a leaky integrate-and-fire (LIF) neuronal model. The digital counter has an output of three bits, but it was designed to be scalable so that if more "muscles" (motors) need to be connected, the counter can be modified to have more output bits.

Keywords - Artificial neural networks, Central Pattern Generator, locomotion in robots, Python.

I. INTRODUCTION

The vast majority of animals that we know of have the ability to move in order to interact with the world whether it is for exploring, foraging, escaping for a predator, etc. The ability to move from one place to another is called locomotion [1] and most living organisms of the animal kingdom have a form of locomotion. For some vertebrate animals such as mammals, birds, reptiles, and amphibians, they are equipped with limbs that allow them to move around their environment. When endowing a robot with locomotion for dealing with uneven terrains, legs offer several advantages when compared to a wheeled device like a better ability to handle gaps and steps, and increased agility to deal with slopes [2].

Both vertebrate and invertebrate animals have dedicated neural networks that produce rhythmic patterns of neural activity to accomplish activities such as locomotion, chewing, breathing, and digesting. These neural circuits are also known as Central Pattern Generators (CPGs) [3]. One surprising feature of CPGs is that apparently, they do not require sensory feedback in order to produce such rhythms. Design of CPGs has been approached from different perspectives. Some authors have used oscillators to generate rhythmic patterns for Earthworm-like robots [4], for quadruped robots [5], or for robots in general [6].

Using oscillators for CPGs design has proven to be quite effective for producing rhythmic patterns for robots unfortunately, when trying to take a biological approach to explain how neural circuits within the CPG are connected, they offer little contribution. Generation of rhythmic patterns has also been proposed using neural networks as in [7–9]. More information about what research has been done in CPGs in the last couple of decades can be found in [10, 11].

II. METHODOLOGY

This paper proposes the use of a digital counter implemented with neural networks which could help to better understand how the neural circuits inside a CPG work and are connected. To demonstrate the latter, let us consider as an example the way human beings walk. When we master the art of walking, we barely pay attention to it while moving. But if we stop to think about it for a moment, we will realize that the activity is nothing but a sequence of movements that keep repeating over and over until we stop. So, once we have rose our right foot and we are taking it ahead towards the ground, when does our brain know when to begin lifting the left foot? Analysing the question, it becomes clear that the brain will send a signal to lift the left foot if and only if:

- The right foot is firmly set on the floor.
- The body is balanced.
- The whole weight of the body can be supported by the right foot.

The conditions mentioned previously will only be met if sensory feedback exists. The first condition will be satisfied if the brain receives signals from pressure sensors on the foot to know that contact with the floor has been made. For the remaining two conditions, the brain needs to receive feedback from the vestibular system within our ears to know that the body is balanced; and it also needs to know the position and tension in the muscles of the supporting leg so that the body won't fall. These are the reasons why for achieving locomotion in a robot, getting data from sensors and motors is of the utmost importance. The design proposed in this paper will be implemented on a Go1 Edu quadruped robot from Unitree so it will use feedback from sensors (pressure sensors at the bottom of each leg and current position of the servomotors) as the clock signal for the counter.

III. DESIGN

A. Neuron Model

The model used for this simulation was a leaky integrate-and-fire (LIF) neuronal model [12]. Although the LIF neuronal model is a highly simplified version of a biological neuron and neglects many aspects of neuronal dynamics, it is one of the most widely used models for analyzing the behavior of neural systems. A model was developed in Python for both an excitatory neuron (neuron that favours the activation of the neurons connected to it), and an inhibitory one (neuron that reduces the probability of neurons connected to it to generate an action potential).

The model for the neuron was designed to be as close as possible to a biological neuron (except for the resting potential which was set at 0 mV as a reference). The main parameters for the simulated neuron model can be found in Table I.

B. Logic Gates

Once the neuron model was ready, A design for various logic gates was done including NAND, NOR, NOT, and XOR. Neuronal circuits for an AND and OR logic gates were not designed since they can be done using a single neuron. A diagram for the NAND, NOR, and XOR logic gates is shown in Fig. 1. The blue circles correspond to excitatory neurons and the red ones correspond to inhibitory neurons.As can be seen from Fig. 1, the same neuronal configuration was used for both the NAND and NOR gates so to get a different truth table for each case the weights on the neurons were changed.

C. Flip-Flop T

The next step after having a design ready for the logic gates was to design a Flip-Flop T (Fig. 2) with neurons. Flip-Flop T is named like that because it toggles the output when the input T is enabled. Assuming a logic one is always applied to input T, then if the previous output Q was zero, it will change to one with the next clock signal. Conversely, if the previous output was a logic one, then with the next clock signal the output Q will change to zero. Q' is the complemented output for Q and as mentioned at the beginning of this section, the clock signal would come from sensory input of the robot to let it know that the next movement in the sequence needs to be executed.

D. 3-bit Counter

With the flip-flop T working, a synchronous 3-bit counter was designed using one flip-flop T for each bit (Fig 3). The counter will go from binary zero (000) with the first clock signal, up to binary 7 (111) with the eighth clock signal and start again. If more bits are needed for the counter, the design can be escalated to include more flip-flops but whether three or five bits are needed for the counter, a decoder stage must be used after the counter to reach all servomotors involved in the movement sequence of the robot.

III. RESULTS

A. Neuron Model

The neuron model was tested by sending impulses to both an excitatory and an inhibitory neuron which were in turn connected to each other. The graph in red corresponds to the inhibitory neuron and the blue one corresponds tothe excitatory neuron as seen from Fig. 4. Although both neuronswere fed with the same input, the excitatory neuron fired less than the inhibitory one. This was caused in part because of the interconnection between them. When the inhibitory neuron fired at times 20 and 50 approximately, it decreased the action potential being built in the excitatory neuron as seen for the same times in Fig. 4-b. The action potential occurring in the inhibitory neuron at time 80 had no effect on the excitatory neuron since both action potentials happened at approximately the same time.

Parameter	Value
Action Potential duration	5 ms
Excitatory Post-Synaptic Potential (EPSP) duration	15 ms
Inhibitory Post-Synaptic Potential (IPSP) duration	15 ms
Resting Potential	0 mV
Action Potential Threshold	14.9 mV
Action Potential Peak	86.9 mV

Table:-1 Parameters for simulated neuron

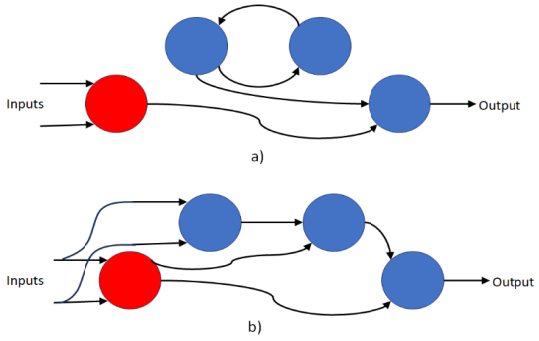


Fig:-1Neuron Diagram for a) NAND and NOR, and b) XOR Logic Gates.

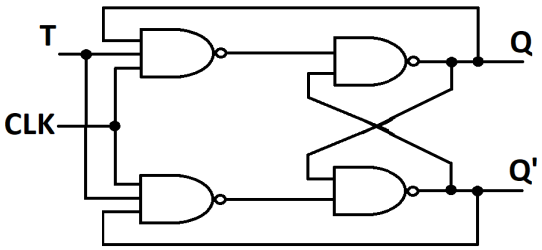


Fig:-2 Flip-Flop T.

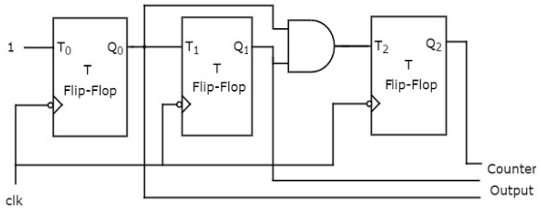


Fig:-3 3-bit counter.

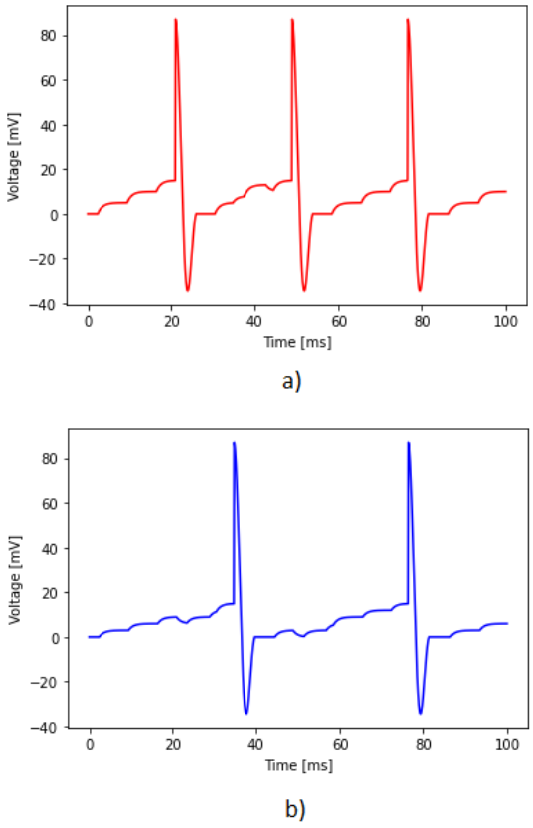


Fig:-4 Neuron model response: a) Inhibitory neuron, b) Excitatory neuron.

B. Logic Gates

We know that for the output of a NAND gate to be a logic zero, all its inputs must be one. Any other combination of inputs (00, 01, and 10 for a 2-input NAND gate) will result in an output of one. This means that the neuronal circuit for the NAND gate must have a way of generating a continuous trainof pulses until all the inputs are one. The two excitatory neurons at the top of Fig. 1-a have precisely that function. An initial pulse had to be applied to one of the neurons for the oscillation to start. An input of 0,1 was applied to the NAND gate at times 10 and 21 of 1,0 at times 55 and 66, and of 1,1 at times 100, 110, and 123 (Fig. 5-a). As it can be seen from Fig. 5-b, a sustained input of 1,1 is required if we wish the output to be a constant logic zero.

C. Flip-Flop T

Although the flip-flop from Fig. 2 seems symmetrical, building an electronic circuit such as the flip-flop T using a Complementary Metal-Oxide Semiconductor (CMOS) technology will result in a circuit that is not completely symmetrical. What is meant by this, is that when polarized, the electric current flowing within the flip-flop T will arrive at slightly different times to outputs Q and Q'. This small, but very important difference in time will allow the flip-flop to have a correct behavior and have a logic one on Q and a logic zero on Q' or vice versa.

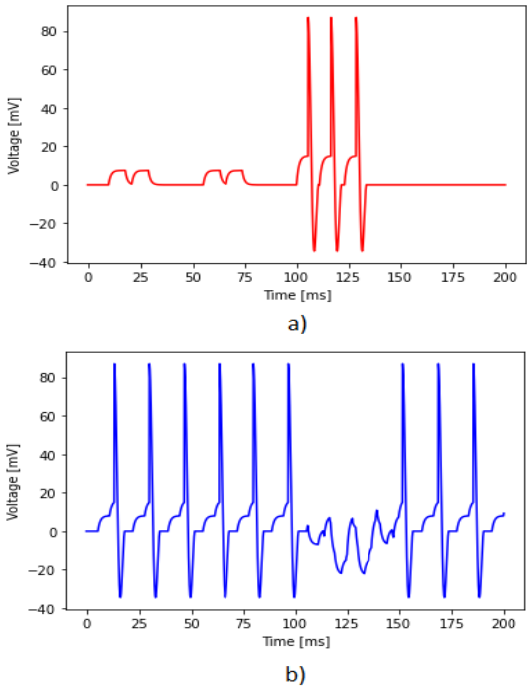


Fig 5:-I/O signals for the NAND gate: a) inputs (red) and b) output (blue).

The first time the flip-flop was simulated the output at both Q and Q' was the same because the signal was reaching both outputs at the same time. To solve this problem and provide a time delay in the neuronal circuit, an extra neuron had to be included between the path that goes from Q to the NAND gate producing Q'.

D. 3-bit Counter

Fig. 6 shows the output of the 3-bit counter going from the most significant bit (MSB) at the top, to the least significant bit (LSB) at the bottom. It can be seen that for each bit, there is an action potential at

the beginning of the graph (time zero). This is because the neural network has not yet felt the effect of the time delay introduced by the extra neuron discussed in the previous section (IV-C).

V. CONCLUSION AND FUTURE WORK

A 3-bit digital counter implemented with an Artificial Neural Network was presented. A further decoder stage would be needed if the counter were to be connected to the servomotors from the robot. The counter design can also be scaled in case more bits at the output are required. Future work involves testing the digital counter along with a decoder neural network on a quadruped robot.

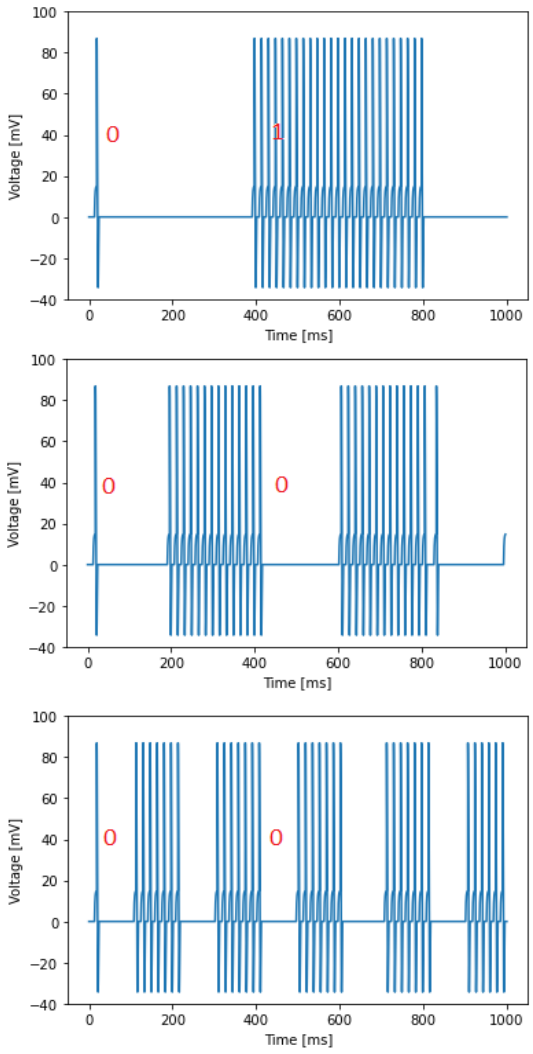


Fig 6: Output of the 3-bit counter. Top: Most significant bit. Bottom: Least significant bit.

★★★

REFERENCES

[1] I. Merriam-Webster, Merriam-Webster's Collegiate Dictionary. Merriam-Webster, 2004.
[2] M. A. Sharbafi, and A. Seyfarth, "Introduction," in Bioinspiered Legged Locomotion, M. A. Sharbafi and A. Seyfarth, Eds. Butterworth-Heinemann, 2017, pp. 1–4.
[3] D. Bucher, "Central pattern generators," in Encyclopedia of Neuroscience, L. R. Squire, Academic Press, 2009, pp. 691–700.
[4] Q. Zhou, J. Xu, and H. Fang, "A cpg-based versatile control framework for metameric earthworm-like robotic locomotion", Advanced Science, vol. 10:2206336, pp. 1–13, 2023.
[5] K. Saito, Y. Ikeda, M. Takato, Y. Sekine, and F. Uchikoba, "Development of quadruped robot with locomotion rhythm generator using pulse-type hardware neural networks", Artificial Life and Robotics, vol. 20, pp. 366–371, November 2015.
[6] T. Nachstedt, C. Tetzlaff, and P. Manoonpong, "Fast dynamical coupling enhances frequency adaptation of oscillators for robotic locomotion control", Frontiers in Neurorobotics, vol. 11:14, pp. 1–13, March 2017.
[7] C. J. Pretorius, M. C. du Plessis, and J. W. Gonsalves, "Evolutionary robotics applied to hexapod locomotion: a comparative study of simulation techniques", Journal of Intelligent & Robotic Systems, vol. 96, pp. 363–385, February 2019.
[8] E. Aljalbout, F. Walter, F. Röhrbein, and A. Knoll, "Task-independent spiking central pattern generator: a learning-based approach", Neural Processing Letters, vol. 51, pp. 2751–2764, March 2020.
[9] C. Maufroy, H. Kimura, and K. Takase, "Towards a general neural controller for quadrupedal locomotion", Neural Networks, vol. 21, pp. 667–681, 2008.
[10] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on CPG-inspired control models and system implementation", IEEE Transactions on Neural Networks and Learning Systems, vol. 25, pp. 441–456, March 2014.
[11] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review", Neural Networks, vol. 21, pp. 642–653, 2008.
[12] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, Neuronal Dynamics, Cambridge: Cambridge University Press, 2014, pp. 10–25.