Central Pattern Generator Utilizing the Izhikevich Neuron Model

Joseph Szlavik
Department of Electrical and Computer Engineering
University of Pittsburgh
jjs229@pitt.edu

Souvik Roy
Department of Bioengineering
University of Pittsburgh
souvikroy554@gmail.com

Gilgal Ansah
Department of Bioengineering
University of Pittsburgh
gia21@pitt.edu

***Abstract*—This project aims to develop a computer implementation of the Izhikevich neuron mathematical model that mimics the organic structure of the brain, and to use this model to create a Central Pattern Generator (CPG) that can generate positive linking bursts to different segments of a digitally modeled leg. The system is designed with parameterization in mind to allow for easy adjustment of the output to the leg. The first step of the project is to implement an Izhikevich Neuron model in Python, which is capable of outputting different firing patterns based on different model parameters. After fine-tuning and examining different patterns, four separate neurons will be integrated into a CPG, with adjustable linking parameters to allow for testing of different system responses. The final step is to bring the model to a simulation software to test its operation on a bipedal or quadruped model. The minimum viable deliverable is a functioning neuron model, with an adjustable operational CPG model also within reach. The stretch goal is to integrate the system design with simulation software to create a bipedal or quadruped model capable of walking on its own.**

A copy of our code can be found here: https://bit.ly/CPG_Izhi

Keywords—Izhikevich Neuron, Central Pattern Generator, NEST

I. Introduction

Central pattern generators (CPGs) are neural circuits that generate rhythmic patterns of motor activity, such as walking, swimming, or breathing, without the need for continuous input from higher brain centers [1]. CPGs are found in a wide range of animals, from simple invertebrates to complex vertebrates, and are thought to have evolved early in the history of animal life. In recent years, researchers have made significant progress in understanding the cellular and molecular mechanisms that underlie the generation and modulation of CPG activity. This has opened up exciting possibilities for developing new treatments for disorders of motor function, such as spinal cord injury (SCI), stroke, and Parkinson's disease that affect millions of people worldwide. This is particularly relevant in the field of neural engineering because well-understood circuits can be modulated by neural interfaces. As such recent work has been using spinal cord stimulation in patients with SCI to enable walking using CPGs in the lower spinal cord by simulating cortical input through a neural device.

Neuromodulation is an important part of CPG functionality in what allows them to be flexible systems with the ability for humans to do complex behaviors outside of simply rhythms such as walking over uneven terrain. As such simple neuron models such as the leaky integrated fire (LIF) model are not able to exhibit this neuromodulatory behavior at a single node. As such there are more complex biomimetic models that allow for this neuromodulation such as the Izhikevich or Hodgkin-Huxley models. These models, inspired by neurophysiology, better capture the biophysics of the single neuron but are based on non-intuitive and computationally expensive differential equations which would be difficult to utilize in a large network such as modern ANNs and SNNs. However these biomimetic models are a good fit for utilization in CPGs because they consist of a relatively small number of neurons the complex firing patterns of which can significantly alter network output to successfully represent true walking behaviors.

II. Background

The Izhikevich neuron model is a mathematical model of a neuron that was proposed by Eugene M. Izhikevich in 2003 [2] as a simplified alternative to the more complex Hodgkin-Huxley model. The Izhikevich model is based on a two-dimensional system of differential equations that describe the dynamics of the neuron's membrane potential and recovery variable. Despite its simplicity, the Izhikevich model is capable of replicating a wide range of neuronal behaviors, including regular spiking, bursting, and chaotic activity.

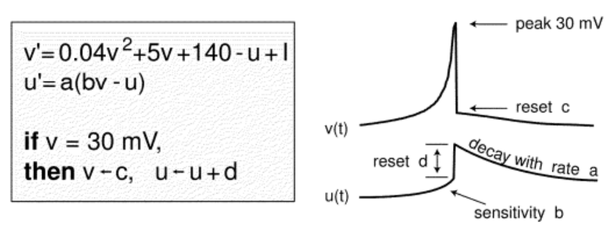

Fig. 1. The neural dynamics of the Izhikevich Neuron Model. v is the membrane voltage and u is the membrane recovery variable. [2]

Previous studies have implemented CPGs by using simple LIF neurons [3,4], however these models are unable to utilize neuromodulation to alter their rhythms in response to environmental challenges. This is why we seek to implement it for the first time in a CPG model in order to generate rhythmic behaviors. Ideally this system would be able to generate a variety of patterns to make it robust in response to environmental challenges.

The aim of our project is to mirror the organic structure of the brain by utilizing a mathematical model of a neuron. We will use these outputs along with mathematical models of synapses to create a Central Pattern Generator (CPG). From there, the CPG will output positive linking bursts to different segments of a digitally modeled leg. The system will be designed with parameterization in mind to allow for easy adjustment of the output to the leg.

To implement this system, the first step of the project is to implement an Izhikevich Neuron model in python. This model is capable of outputting different firing patterns based on different model parameters. We will fine tune and examine different patterns as we integrate the neuron into the CPG. Different neuron models will be evaluated later as we see how it affects our system.

After we have an operational mathematical model of a neuron, we will integrate four separate neurons into a central pattern generator. The linking parameters will be easily adjustable to allow for testing of different system responses. The ability to alter the neuron and CPG parameters allow for system configuration and refinement.

Lastly, after the system outputs a desired result, we will bring the model to a simulation software to test its operation on a bipedal or quadruped model. Now having the system parameterized and a model to rapid prototype, we will be able to quickly iterate our models for a better result.

By the end of this project, our minimum viable deliverable is a functioning neuron model. We believe that having an adjustable operational CPG model is also in doable scope for this project's deadline. Our stretching goal is to integrate this system design with simulation software. The primary concern is how well we can integrate the software to one another and the overall ability to use the simulation software. If we are able to complete this last step properly, we would have a bipedal or quadruped model fully capable of walking on its own.

III. Design And Implementation

A. Izhikevich Neuron Model

Most previously designed and simulated CPGs used the LIF neuron model. Therefore our goal was to create a model of the Izhikevich Neuron that would resemble the functions of a classic LIF neuron. This would allow us to easily create a CPG as well as implement the neuron into any simulation software. Looking at the basic function of other LIF neuron models and in conjunction with traditional black box coding methods, we decided that the minimal amount of code with the maximum amount of adjustability would be the best.

Using black box coding method, we have "set" and "get" functions that allow the user to access the variable easily without getting elbow deep in the code. In addition, a "setType" function would be implemented that allows the user to easily switch the firing type of the neuron. These would be preset, but it allows the user to see how that type of neuron interacts with their experiment and the "set" function would allow them to individually change and tune that specific neuron.

Lastly, to use the neurons, there are two separate functions. The first is the "fire" function. This function would only need a time, a time step, and a current. It would then be able to return a voltage time and an array of voltages. The other way to interact with the neurons is through the "plot" function. This function allows the user to plot the individual neuron using the same parameters as stated above. This feature would help the user diagnose and set up their individual neurons before integrating them into their CPG model.

B. CPG Network

The CPG network consisted of four neurons connected as a ring oscillator. Each neuron had an input step current that was delayed by some variable amount. The amplitude of all the currents were the same. At the synaptic junctions, the output of the pre-synaptic neuron has a synaptic weight (α), which is used to compute the synaptic conductance (g_{syn}), which is the sum of all input weights. The conductance, together with the decay constant of the neuron, (τ_{syn}), is then used to compute the synaptic current (I_{syn}) which is added to the input step current at each time step. I_{syn} is computed using the following equation:

$$I_{syn} = g_{syn} \times (c - v) \times e^{\frac{-dt}{\tau_{syn}}}$$

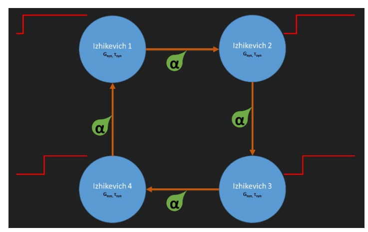

Fig. 2. Schematic of the ring oscillator. The red step functions by each neuron are the step currents. The delay increases from neuron 1 to neuron 4.

To simulate locomotion, each neuron was made analogous to one of the four legs of a quadruped. The intuition is that when a neuron spikes, the corresponding leg moves. The goal of the CPG is to generate different spike patterns to be able to produce different types of locomotion.

C. Implementation through NEST and NeuroRobotics Platform (NRP)

NEST and NRP were both designed as part of the European Human Brain Project and are useful tools for both neuroscientists and neural engineers. NEST is a python library which allows for simulation of neural networks and has built-in models such as the LIF neuron. It also has an open source language NEST-ML which can be used to write new models into the NEST system. NRP is a simulation software which allows users to write brains in NEST and using them to pilot robots in virtual spaces by using ROS programming and transfer functions. The design of this project will transfer the CPG network built using the Izhikevich model into NEST using

NEST-ML to code the Izhikevich object. This will be used to pilot a quadruped robot in the NRP simulated virtual space.

IV. Results And Challenges

A. Izhikevich Neuron Model

Initial parameterization of the neuron model was not very difficult. It took a few tries and several times reorganizing the code to the simplest and usable fashion to take it to its current state. It ended up being extremely lean code and we are very happy with its result.

Below are some of the results from the code, showing different spike patterns.

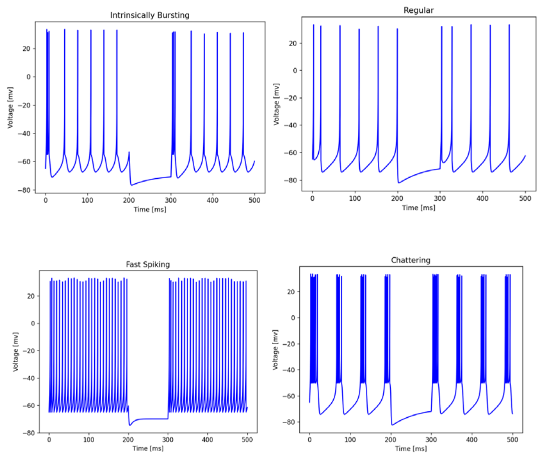

Fig. 3. Showing different spiking modes with the Izhikevich Neuron Model

Trying to get the code implemented into neuron simulation software was the difficult aspect. The issues were primarily with the simulation software. We attempted to use the software Nest for our simulations, but it required NestML to implement our model. Nest is a new company and has very little support. NestML is an even younger section of the company. Spending over two dozen hours and multiple virtual machine reinstalls resulted in no progress with the customizing software.

B. CPG Network

With a constant synaptic weight of $\alpha = 1$, and step current delay values of $0.01s, 0.06s, 0.11s$ *and* $0.16s$respectively, we were able to successfully simulate the walking pattern for the CPG. By changing the spiking mode, we are also able to generate more diverse walking pattern.

One of the main challenges we encountered in our project was the observation that the membrane voltage plots for the central pattern generator (CPG) exhibited dynamics characteristic of the Leaky-Integrate-Fire (LIF) neuron model. While a single instance of the Izhikevich python class produced the expected firing patterns, the shape of the voltage plots for the neurons in the network took on an LIF-like form. We hypothesize that this may be due to the influence of the network connections on the

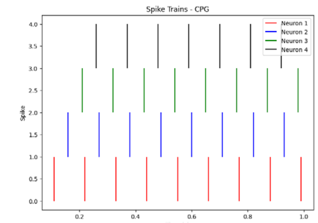

Fig. 4. Walking spike pattern for CPG using Regular Spiking

membrane voltage, although the underlying mechanism for this behavior remains unknown at present.

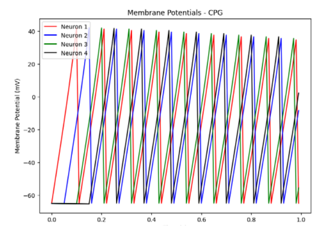

Fig. 5. LIF like voltage patterns

C. Implementation through NEST and NeuroRobotics Platform (NRP)

The CPG network built using LIF neurons was successfully written in NEST. Figure shows a visualization of the network using NEST desktop, a web-based GUI for NEST. The weights of the connections and the parameters of the neurons were empirically determined using other implementation of CPGs using other python libraries. Figure shows the activity of this network. After an initialization phase, there is consistent rhythmic firing of each of the neurons in the network in a mechanism that could drive locomotion using further downstream circuit to more finely control multiple actuations within a robotic leg. There were significant challenges with the remainder of the proposed implementations as much of this technology was made the in last few years and documentation is not extensive enough for successful trouble shooting. Ideally this network would be recreated using Izhikevich neurons written with NEST-ML and then ported to NRP for use as a brain, but this will have to future work.

V. Conclusion And Future Work

The next steps would be to optimally integrate all parts of this project. This would mean using NEST-ML to write the Izhikevich object and then placing that model in the CPG network and optimizing parameters to create rhythmic firing. Then that code could be used to create a brain in NRP which could make the Trigllio quadraped robot walk. This could then potentially be ported to a hardware model of the robot.

Acknowledgment

We would like to express our sincere gratitude to Rajkumar Kubendran, PhD and Vijay Shankaran Vivekanand from the ENIGMA lab at the University of Pittsburgh for their invaluable contributions to this work. Vijay Shankaran Vivekanand

provided valuable insights and expertise in the experimental setup and execution, while Dr. Kubendran offered helpful suggestions and feedback throughout the process. Their contributions were instrumental in the success of this project.

References

[1] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biology*, vol. 11, no. 23, pp. R986–R996, Nov. 2001, doi: https://doi.org/10.1016/s0960-9822(01)00581-4.

[2] E. M. Izhikevich, "Simple Model of Spiking Neurons", in IEEE Transactions On Neural Networks, Vol. 14, No. 6, November 2003 pp.1569–1572.

[3] P. Lopez-Osorio, A. Patiño-Saucedo, J. P. Dominguez-Morales, H. Rostro-Gonzalez, and F. Perez-Peña, "Neuromorphic adaptive spiking CPG towards bio-inspired locomotion," *Neurocomputing*, vol. 502, pp. 57–70, Sep. 2022, doi: https://doi.org/10.1016/j.neucom.2022.06.085.

[4] K. Smith, A. S. Lele, Y. Fang, "A Spiking Central Pattern Generator for Quadrupedal Gaits," unpublished.