Error Correction in CPG Gait Control of Cockroach Model by Characterizing PID Like Neural Responses

Jessica M. Hatch

Supervisor: Dr. Roger Quinn

Center for Biologically Inspired Robotics Research and Department of Physics
Case Western Reserve University, 10900 Euclid Ave. Cleveland, OH 44106

Date of submission: 05/02/2012
Abstract

The Center for Biologically Inspired Robotics Research is building a computational neural system for leg coordination into gaits that automatically adapt to irregular terrain for modeling animals and for control of legged robots. To construct this biologically-based simulation, a complex system of components modeling muscles, sensors, and neural-networks is used. To ensure a smooth walking pattern, there is a need for a sub network of neurons that acts as feedback loop for the Central Pattern Generator. There is biological evidence to suggest that there is a proportional-derivative type controller that acts as the check between joint motions and commands issued by the central nervous system. In order to build this network the response of single neurons and combinations of neurons must be explored and compared to proportional, derivative, and integral responses of the test inputs. Three different inputs were used: a tonic, burst, and repetitive current, and seven neurons were characterized. Proportional and Integrative responses were identified and further study is needed to identify derivative responses. The ultimate goal is to use these characterized responses to create a dynamic model that optimizes behavior and performance, and further research is needed to build and test the PID controller from the available neurons.
## Contents

1 Introduction .......................................................... 2
   1.1 Background ......................................................... 2
   1.2 CPG Model Overview and Current Network ....................... 4
   1.3 Neuron Equivalent Circuit ........................................ 6
   1.4 Goals and Objectives ............................................. 7

2 Methods and Techniques ............................................... 8
   2.1 Software Used ...................................................... 8
   2.2 Method of Exploration ............................................ 8
   2.3 Expected PID Response ........................................... 10

3 Results and Analysis .................................................. 11
   3.1 BiStable Firing Rate Neuron ....................................... 11
   3.2 Tonic Firing Rate Neuron ......................................... 15
   3.3 PaceMaker Neuron .................................................. 18
   3.4 Firing Rate Neuron ................................................ 21
   3.5 Summary of Results and Analysis ................................. 24

4 Conclusion ............................................................... 25
   4.1 Objectives Met ...................................................... 25
   4.2 Future Work ......................................................... 26

A Final Presentation Slides (5/1/2012) .................................. 30

B Additional Graphs from Test Series .................................... 34
   B.1 Bistable Firing Rate Neuron: .................................... 34
   B.2 Tonic Firing Rate Neuron: ......................................... 35
   B.3 Firing Rate Neuron .................................................. 36

C Mathematica Code–PID Response WorkBook ............................ 37
Chapter 1

Introduction

1.1 Background

Legged creatures are able to navigate complex terrain. The Center for Biologically Inspired Robotics Research at Case Western Reserve University has a goal to model legged insects to create a robot with applications that include search and rescue bots, surveillance, and swarm tactics. The Robotics Lab at CWRU has chosen a unique approach to this challenge–create a simulation that is one to one between the model and the known biology. The overall goal is to create a robot controller capable of gait control and adaption to dynamic environments. The neural architecture of this controller is to be constructed based on what is known of the actual biological structures. In partnership with Dr. Roy Ritzman’s lab at CWRU, which studies cockroaches, the cockroach will be used as the biological model and this controller must work in real time and be capable of various gaits.

A basic understanding of neural biology and a brief introduction to the neural networks involved in this research are necessary to understand the mechanics and physics of this project. The term neural network refers to a configuration of interconnecting neurons occurring in biological systems. These groups of neurons are organized in specific ways to achieve physiological functions, such as controlling breathing, heart-rate, or gait.

A neuron is composed of several parts: soma, axon, terminals, and dendrites (Figure 1.1). The soma is the cell body. The axon carries information out of the soma to other neurons. The terminals are the outputs of the cell and the dendrites are the inputs for information from other neurons [3]. Nerve cells use electrical signals to communicate and pass information from one cell to the next. These electrical currents are carried by ions within the neuron and two kinds of channels
direct the current, active and passive[3]. Passive channels are always open whereas active channels are closed by default and open in response to a voltage change or in the presence of a chemical receptor[3]. These active, or gated, channels have an “all or nothing” response—that is, there is a threshold voltage that must be attained for there to be a response from the channel.

Figure 1.1: A neuron is a specialized cell that is made up of dendrites, cell body (stoma), axon, myelin, and terminals. Electrical signals travel through the neuron and across a synaptic gap to other neurons. There are three types of neurons: sensory, motor, and interneurons. Image courtesy of The University of Waikato.

There are three categories of neurons: sensory, motor, and interneurons. Sensory neurons carry information into the nervous system. Specialized cells, called receptors, gather information (light, heat, pressure, etc.) and convert these signals into neural responses[3]. Motor neurons are nerve cells that connect to muscles and direct signals to control motion of joints and reflexes, and interneurons are any nerve cells that are between a sensory neuron and a motor neuron[3]. Electrical transmissions are quick and are used to communicate motion of rapid coordinated activity[3]. The speed of transmission is fast because the signal strength is fixed and uses the gaps between neurons to transmit the message from one nerve to another. Chemical transmission uses chemicals that diffuse across synapses and connect with receptors to trigger a response there. The signal strength is not fixed in this kind of transmission, and therefore larger signal strength triggers a larger response at the receptor[3].
1.2 CPG Model Overview and Current Network

Key to this research is the Central Pattern Generator (CPG). CPGs are groups of neurons that interact to control rhythmic motions of the organism [6]. While the exact biological principles are not fully known, several models of CPGs have been developed and tested. The half center model was proposed by Thomas G. Brown in 1914 [7] and then later developed by Anders Lundberg in 1981 [8, 9]. This model describes the CPG as two centers that are connected and can send signals back and forth through reciprocal inhibition[6].

Reciprocal inhibition means that if a signal is being received by a neuron it cannot transmit to the neuron that is sending the signal. Once that received signal decays the potential of the receiving neuron rises until it reaches a threshold and then can send a signal back to the other neuron while inhibiting the other neuron’s ability to transmit. This forms a cycle of inhibition with a decaying signal flowing in one direction, until the potential of the receiver is sufficient to reverse the inhibition and flow of information. The oscillation period between the two states and the phase of the signal are the two ways to characterize and control the CPG[6]. The signal outputs of CPGs are delivered to motor neurons; therefore, by controlling the CPG, one is indirectly controlling the movement of the organism or robot. Figure 1.2 demonstrates this oscillatory behavior: as neuron 3 spikes, neuron 4 is inhibited and vice versa.

![Figure 1.2: A Central Patter Generator (CPG) is a pair of neurons that are mutually inhibiting. As 3 spikes in the figure above, 4 is inhibited from spiking. 3’s signal decays and 4 is then able to reach a threshold value and spike, reversing the inhibition. This spike and inhibit cycle creates an oscillatory pattern.](image)

This cockroach model is unique in that the mechanics and networks are directly...
modeled from the known biology. A model was created using a neuron simulation tool, Animatlab. The current neural network is depicted in figure 1.3. Each red pentagon represents a leg joint. At each joint is a Central Pattern Generator Pair. This neuron is mutually inhibiting; as one half of the pair is firing the other cannot fire. As the first signal decays the second neuron in the pair is able to fire and inhibit the opposite neuron. The oscillatory relationship is responsible for the contraction and extension of each leg.

Figure 1.3: This is the neural network. Cockroaches have 6 legs. Each red unit represents a leg joint—a CPG pair that is responsible for leg contraction and extension which is the basis for a normal walking gait. The integrators (Left and Right) smooth signals entering the network so that the system is not thrown too far from equilibrium.

The simulation of the cockroach mimics the tripod gate of the insect. Figure 1.4 shows the process for this gait. As three neurons at the joints are able to fire, the “opposite” three are inhibited. This cycle of release and contract of three joints at a time enables the smooth tripod gait. This is a desirable walking pattern because this gait provides stability, the ability to climb and deal with uneven terrain, as well as easily transition between different walking speeds.
Figure 1.4: This is how CPGs are used to create a tripod gait. Each oval is a CPG, the lines represent synaptic connections, and the small circles indicate an inhibiting connection. The green coloring means that the neuron is firing and a black neuron indicates that it’s firing is being inhibited.

The current network design is functional and the simulation cockroach is able to move about the simulated environment. Two shortcomings of this model are one, the accuracy of the sensory data is taken on faith and two, the CPG response to the sensory input is assumed to be appropriate. Large and sudden inputs to this model can disrupt the tripod gait and effect negatively the equilibrium. An integrator is used to smooth the inputs and mitigate the likelihood of severely disrupting the walking pattern. These preventative measures help decrease the risk of instability, however, an error feedback loop should be used in conjunction with these measures to keep a smooth gait and adjust to a dynamic environment.

1.3 Neuron Equivalent Circuit

Neurons can be modeled as a resistor-capacitor circuit. The basis for this model comes from the fact that neurons have a membrane resting potential, the voltage changes when a current is applied, and the voltage change decays exponentially\cite{12}. The resting potential corresponds to an electromotive force, maintaining equilibrium in the system. A proportional relationship between voltage and current is described in Ohm’s Law, $V = I*R$, which governs the response of a resistor, and an exponential
decay can be modeled by a capacitor[12]. Thus a single neuron is a resistor and capacitor in parallel with a battery as shown in Figure 1.5.

![Image of equivalent circuit for a single neuron]

Figure 1.5: This is the equivalent circuit for a single neuron. The battery models the membrane potential, the resistor represents a proportional relationship between current and voltage, and the capacitor models the time-constant behavior of a neuron.

1.4 Goals and Objectives

The overall goal of this senior project was to characterize neurons to find proportional, integral and derivative responses to a variety of inputs so that these neurons could be put into a network to act as a controller for error correction. This controller will be connected to one of the CPG half pairs at a single joint in the cockroach model because monitoring and correcting the behavior of one will effect the behavior of all connected CPGs. This means that one controller will be sufficient and therefore insure that the model is not unnecessarily complicated.

Another objective is to have a qualitative way to determine how close to a proportional, integral, or derivative response the neuron provides. To accomplish this, a digital PID circuit was built using MultiSim. This simulation provides a control for the research and allows for a way to compare responses from the test neural network to the output of a PID circuit.
Chapter 2

Methods and Techniques

2.1 Software Used

All software and tests were run using Windows XP. AnimatLab, Mathematica, MultiSim, and Excel were used to model data, define theoretical results, analyze data and run simulations. AnimatLab is an open-source software package, available at http://www.animatlab.com/Download.htm. AnimatLab uses biological models for neurons and neural networks, employs Vortex as its high fidelity physics engine, and allows the users to build and test custom neural networks. The capabilities of AnimatLab align well with the needs of this research.

Mathematica is a computing software (available at http://www.wolfram.com/mathematica/). A Mathematica script was used to create visuals for a proportional, integral, and derivative responses. The full code is located in Appendix C.

MultiSim is a National Instruments software (http://www.ni.com/multisim/), where the user builds circuits in simulation and then runs the environment and can monitor the circuit and take measurements. Proportional, integral, and derivative modules are included in MultiSim, and the drag and drop interface is intuitive. Furthermore experiments can be run in real time. Lastly, MultiSim accepts csv files as input and animatlab outputs in csv format. Therefore both our simulators are compatible can can exchange data sets.

2.2 Method of Exploration

There are many variables involved in this research, the neuron type, properties of the neuron, the type of stimulus, and so on. In order to isolate causes and explore relations between these variables the research was broken into different pieces and
a variety of software applications used. In order to have a control, a PID loop was created using NI MultiSim. Also I wrote a script in Mathematica where the user can input a function and the proportional, integral, and derivative of that function are the output. These two processes provide a baseline with which to compare the collected data.

There are seven neurons available in AnimatLab: BiStable Firing Rate, Tonic Firing Rate, PaceMaker, Firing Rate, Spiking, NonSpiking, and Random Firing Rate Neuron. 7 separate simulations were created each with six of the same neuron as part of each simulation. Applied to each collection of neurons were six test patterns which were various combinations of the three stimuli available in AnimatLab: Tonic, Burst, and Repetative. The six test series are shown below in Figure 2.1. Varying amplitudes, durations, and negative and positive values were used. In total each type of neuron was exposed to 18 stimuli, and each simulation was 25 seconds long.

The data collected in Animatlab were organized in the graphing tool provided by the software as well as exported to an excel file. The data was organized by test series number and neuron type. Each trial displays the external current, intrinsic current, firing frequency, and membrane potential. Finally the graph was saved using image capture.

Figure 2.1: These are the 6 test series used in the experiment.
2.3 Expected PID Response

As mentioned above the following stimuli were used: tonic, burst, repetative. These stimuli were combined in six different test series. Figure 2.2 shows the theoretical proportional, integral, and derivative responses expected from these inputs.

![Figure 2.2: Input stimuli and the expected responses.](image)

The tonic pulses stay pulses in the proportional response, become a dirac delta and well pair for the derivative response are is a ramp for the integral response. The repetative and burst stimuli have similar PID responses. The proportional responses have the same signal but larger amplitudes, the derivative responses are dirac delta and well pairs that are consecutive, and the integral response is a step function. For the purposes of this experiment, all dirac deltas and dirac wells were adjusted to have finite amplitudes. This is acceptable because neurons have finite responses, so putting a ceiling and floor on the signal make sense. Also important to note that at this stage in the research, a visual expectation was used to determine which responses looked similar to the theoretical findings. The key features of the theoretical responses were used as the guide in determining what responses looked most like. The collected data was scanned for those features and the instances of similarity were documented.
Chapter 3

Results and Analysis

3.1 BiStable Firing Rate Neuron

Six bistable firing rate neurons were tested with different external currents. The graphs that follow summarize the results. On the graphs, red is the applied (external) current (nA), green is the firing frequency of the cell (Hz), blue is the intrinsic current (nA), and yellow is the membrane voltage (mV). In almost all cases the Bistable Neuron exhibited a proportional response to the external current. Each response is reviewed below. Test Series 5 and 6 for this neuron are located in Appendix B.
Figure 3.1: Test Series 1: BiStable Neuron. A proportional response is evident in the frequency and voltage responses for the repeat and burst inputs. The Tonic input response is almost like a ramp for the frequency and voltage responses. This response is more like an integral behavior. The Bistable Neuron’s current is constant and is turned on once the threshold value is met.
Figure 3.2: Test Series 2: Bistable Firing Rate Neuron. Important to note is that there is saturation in some of the signals for this test series. The firing frequency of the cell is saturated at a particular external current value. None of the responses are proportional, integrative, or derivative like for the repeat input when the external current oscillates between positive and negative. The membrane voltage is effected by this input, while the firing frequency and intrinsic current both at zero. The membrane voltage has integral like behavior for square wave inputs. The ramps are not ideal, though, because there is a long decay time. This may be remedied by shorting the time constant under the cell properties.
Figure 3.3: Test Series 3: Bistable Firing Rate Neuron. This test series clearly shows that the BiStable Firing Rate neuron will saturate in firing frequency and membrane potential. Also this series also shows that the intrinsic current has two states on and off. The intrinsic current is an all or nothing kind of response. If the applied current is above the threshold value, the intrinsic current is turned on. Once the applied current becomes negative, the intrinsic current flips to the off position.

Figure 3.4: Test Series 4: BiStable Firing Rate Neuron. This test series is all narrow pulses and finite dirac deltas. It is in this experiment that one can see variations in the frequency response rate. Most notably in the repeat section of the stimuli, as the peaks grow in the input, the frequency response also grows.
3.2 Tonic Firing Rate Neuron

The tonic firing rate neuron results are below. The tonic firing rate neuron demonstrated responses for tonic inputs and did not have a strong response in behavior to repetitive inputs and burst inputs. A sudden on-off did not have a great effect on the neuron. A sustained change is needed to achieve a noticeable response. Test Series 6 for this neuron are located in Appendix B.

Figure 3.5: Test Series 1: Firing Rate Neuron. There is a response for both negative and positive input currents in the membrane voltage. However, the response does not seem to be proportional, integral, or derivative like in behavior.
Figure 3.6: Test Series 2: Tonic Firing Rate Neuron. A proportional like behavior is observed for tonic inputs. One key feature to note is that the firing frequency of the neuron saturates and that aids in making the behavior appear more proportional like. Furthermore, the response is only proportional like when the applied current is being driven between positive and negative values, otherwise the frequency remains on or off at a single value. The membrane voltage is effected by the applied current, the response is not proportional or integral like in behavior though.

Figure 3.7: Test Series 3: Tonic Firing Rate Neuron. In this test series the saturation of the membrane potential is evident. There is some proportional like behavior as well. The response in membrane potential grows in amplitude as the tonic current increases in amplitude.
Figure 3.8: Test Series 4: Tonic Firing Rate Neuron. One of the drawbacks of the Tonic Neuron is that the response to narrow inputs is minimal. The threshold value for firing frequency is met so there is no noticeable change after the initial spike of current, and the membrane voltage is only slightly perturbed by the quickly changing input current.

Figure 3.9: Test Series 5: Tonic Firing Rate Neuron. The membrane potential for this test series is the most like a proportional response for the Tonic Neuron in this experiment. The middle stimuli grow in amplitude and so does the membrane voltage. There is a breakdown in this proportional pattern for the final stimuli, where the width of the pulse is very small and therefore the response of the cell is severely limited.
3.3 PaceMaker Neuron

The single pacemaker neuron also displayed proportional and integrative like behaviors in response to the six test series. There were a few unique qualities in these responses, especially in membrane voltage. Occasionally the voltage would spike and disrupt the otherwise smooth response. More investigation is required to find the cause of this behavior. The pacemaker showed some proportional behavior, although most of the responses look to be an all or nothing response. Additionally the integral like behavior was evident for narrow square wave inputs.

![Figure 3.10: Test Series 1: PaceMaker Neuron. The intrinsic current is proportional with the exception of the first stimuli—the small tonic input. The response for the repeat and burst inputs are proportional like in behavior. The time constant could be adjusted to give a better response.](image)
Figure 3.11: Test Series 2: PaceMaker Neuron. The proportional resone is not exact for the friring frequency behavior. The adjustment time is longer than a proportional reponse would be, and the dirac delta functions do not yeild any lasting or periodic response from the firing frequency. This is because the threshold value to fire was not met by the third stimulus input.

Figure 3.12: Test Series 3: PaceMaker Neuron. The firing frequency and intrinsic current reflect the same period and type of response to the input. However the amplitude of the resone does not change–these two characteristics of the cell have an all or nothing response. The membrane potential demostrates a proportional response for the large tonic inputs. The response to the repetative inputs looks like a different kind of response. The signal is decreased in amplitude and transformed to a tooth like pattern.
Figure 3.13: Test Series 4: PaceMaker Neuron. Test series 4 is almost exclusively dirac delta like inputs which the pacemaker neuron struggles to respond to such quick inputs. With such small inputs, there does not appear to be a clear proportional, integral, or derivative like response.

Figure 3.14: Test Series 5: PaceMaker Neuron. There are characteristics of a proportional response from test series 5. The membrane voltage looks similar to the tonic and burst inputs. The membrane volatage response does not, however, match the dirac delta like inputs at the very end of the trial.
3.4 Firing Rate Neuron

The firing rate neuron was also tested and response data collected from the 6 input series. The firing frequency and membrane voltage demonstrated a strong proportional response. The intrinsic current had less variation and did not demonstrate proportional, integral, or derivative like behavior.
Figure 3.16: Test Series 1: Firing Rate Neuron. The firing rate neuron succeeds where the pacemaker neuron fell short. For extremely narrow tonic inputs and for dirac delta like inputs, the firing frequency and membrane voltage responses are similar to a proportional response like behavior. The responses are not perfectly narrow, there is a time constant factor that contributes to the tail and the end of the signal.

Figure 3.17: Test Series 2: Firing Rate Neuron. The responses for negative values is does not appear to be as strongly proportional as other responses. The firing frequency cannot be negative, for one thing. Furthermore the responses time for the membrane voltage is significantly longer and the response does not look like tonic inputs, but rather a triangle wave.
Figure 3.18: Test Series 3: Firing Rate Neuron. When the square waves are wider, the membrane potential is able to saturate and have a flat top as well, making the response look proportional like. The delta-like inputs are handled well by firing rate neuron. In fact, the firing rate of the cell drops to zero.

Figure 3.19: Test Series 4: Firing Rate Neuron. This test is similar to test series 1. Once again the response is proportional like in behavior and as the amplitude changes so does the firing frequency and membrane potential.
Figure 3.20: Test Series 5: Firing Rate Neuron. The response to test series 5 makes a strong case that the firing rate neuron is most like a proportional response in behavior and is a clear candidate to use in a PID loop as the proportional controller. The amplitudes change in the response according to the changes in amplitude in the input current. The shape of the input remains relatively unchanges and is still recognizable in the outputs.

3.5 Summary of Results and Analysis

Strong proportional like behavior was observed in the BiStable Firing Rate Neuron, Tonic Firing Rate Neuron, PaceMaker Neuron, and the Firing Rate Neuron. The PaceMaker Neuron also demonstrated an integral like behavior for some kinds of inputs. A proportional response was expected from single neurons because one neuron in a network is the simplest configuration and it is logical to expect the most simple response—the same signal altered in amplitude. The demonstrated integral like behavior seems to stem from saturation and the delay in response time capability of the cell. The membrane voltage and the firing frequency have maximum values they can be driven to. The limits of the system work to our advantage in these circumstances and help the response look more like a proportional or integrative behavior. The derivative response was not observed in these trials. The derivative response is more complicated and will most likely require a larger network of neurons to manipulate the signal in such a way to bring about the desired behavior.
Chapter 4

Conclusion

4.1 Objectives Met

The objectives of this project were to:

1. Create a PID Controller in NI MultiSim
2. Characterize neurons in AnimatLab to find proportional, integral, and derivative like behaviors
3. Qualitatively determine how close to a proportional, integral, or derivative response the neuron provides.

The all of the first objective and parts of the second objective were met this year. The PID Controller in MultiSim is operational and calculates the proportional, integrative, and derivative responses to an error signal and outputs the error correction.

Four of the seven neurons in AnimatLab were analyzed for a PID-like response using 18 stimuli in total in six test series that were applied to single neuron neural networks. The remaining neurons, Random Firing Rate, Spiking, and NonSpiking Neuron, still need to be tested and analyzed for proportional, derivative, and integral responses. Also more complex networks need to be tested and analyzed for the desired responses.

The final objective is a natural continuation of the work completed this year. The observational data is important to have to provide an idea of the avenues of further investigation that look most promising. The pacemaker neuron, for example, showed a strong proportional response qualitatively. This makes this neuron a strong candidate for the proportional part of the controller. The Tonic Neuron also showed a proportional response. A quantitative way to determine which is the better fit is needed at this stage in the research to determine which neuron to investigate further.
4.2 Future Work

The remaining neurons need to be tested in animatlab with the 6 test series and the data needs to be quantitatively analyzed in addition to the qualitative results. Proportional and Integral responses were found through this research. Further investigation is needed to determine how proportional and how integral-like these responses are.

Missing in these results is the derivative response. One possibility is the Random Firing Rate neuron. This hypothesis is based on the fact that the derivative part of the controller is responsible for correcting for future error. In a way it mathematically predicts what the error will be in the next iteration. Much of the error in the simulation’s signal is due to noise. This noise is randomized and therefore using a random firing neuron (which fires at both random intervals and amplitudes) may be a way to predict what the error will be in the future for the simulation.

There are also several parameters that can be set for each neuron. Permutations of the testing series and parameter values should be investigated to map the relationship between input types, the parameter values, and the behavior of the response signal. Of particular interest will be the decay parameter for the cell because adjusting this variable will theoretically sharpen the response of the neuron which will aid in tuning the behavior to a proportional or integrative like response.
Acknowledgments

Thank you to Dr. Roger Quinn for offering me this research opportunity and for your help and guidance during the research. Thank you to Nicholas Szczecinski for discussing topics ranging from mathematical assumptions, brainstorming potential neurons to investigate, and interpreting results. Your advice and insights were valuable and enormously helpful in focusing the scope of the project and furthering the research. Thank you also to the other members of The A-Team: Victoria Webster, Alexander Hunt, and Alexander Lonsberry, for your advice, support, and creating a positive team environment in which to conduct this senior project. Finally, thank you to the members of the Senior Project Seminar 2012 for attending the presentations of my research, asking questions that helped clarify my project and methods of research, and providing feedback on my presentation skills.
Bibliography


Appendix A

Final Presentation Slides (5/1/2012)

Neural Model—Equivalent Circuit

- Membrane Potential
- Inject current—voltage changes proportionally
  \[ V = I \times R \]
- Voltage returns to potential exponentially

Presentation Topics

- Introduction
- Methods
- Results and Analysis
- Objectives Completed
- Future Work
- Questions
CENTRAL PATTERN GENERATOR

- Neural Network
- Mutual Inhibition
- Tripod Gate

Sensory data taken on faith
Assumed appropriate response

Neural Comparator Network

Proposed Algorithm

Tools
- Windows OS
- AnimatLab—Neural Simulation
- Multisim—Circuit Simulation
- Mathematica—Mathematical Aid

METHODS
Stimuli

- Pulse (tonic), Burst, and Repetitive Burst

6 Stimulation Patterns

7 Neuron Types

- Tonic Firing Rate
- Spiking Neuron
- Random Firing
- PaceMaker Firing
- NonSpiking
- Firing Rate
- Bistable Firing Rate

RESULTS
Results

- The BiStable Neuron: Both positive and negative external currents evoke a response. The proportion of the response is significantly less for negative values.

Results

- The BiStable Neuron: Both positive and negative external currents evoke a response. The proportion of the response is significantly less for negative values.

- The Pacemaker Neuron: The intrinsic current flips between a high and low. High value is not dependent upon the strength of the external current.

- The Pacemaker Neuron: The intrinsic current flips between a high and low. High value is not dependent upon the strength of the external current.

Objectives Met

✓ Create Digital Circuit in Multisim

✓ Characterize Neurons in AnimatLab

✓ Analyze data for Proportional, Integral, and Derivative like behavior

Objective Met

✓ Create Digital Circuit in Multisim

✓ Characterize Neurons in AnimatLab

✓ Analyze for P, I, and D response

Looking Forward

- Objectives met
  ✓ Create Digital Circuit in Multisim
  ✓ Characterize Neurons in AnimatLab
  ✓ Analyze for P, I, and D response

- Future Work
  ✓ Continue to analyze Neurons (Spiking, Nonspiking, Random)
  ✓ Qualitatively determine P, I, and D type responses
  ✓ Build Controller and test

Thank You

- Dr. Roger Quinn
- Nicholas Szczecinski
- Victoria Webster
- Alexander Hunt
- Alexander Lonsberry

QUESTIONS
Appendix B

Additional Graphs from Test Series

B.1 Bistable Firing Rate Neuron:

Figure B.1: Test Series 5: BiStable Firing Rate Neuron.
B.2 Tonic Firing Rate Neuron:

Figure B.2: Test Series 6

Figure B.3: Test Series 6: Tonic Firing Rate Neuron.
B.3 Firing Rate Neuron

Figure B.4: Test Series 6: Firing Rate Neuron. The response to test series 6 makes a strong case that the firing rate neuron is most like a proportional response in behavior and is a clear candidate to use in a PID loop as the proportional controller. The amplitudes change in the response according to the changes in amplitude in the input current. The shape of the input remains relatively unchanged and is still recognizable in the outputs.
Appendix C

Mathematica Code–PID Response WorkBook

(*Calculate Integral, Derivative, and Proportional Response for User Inputs
Jessica Hatch Senior Engineering Physics BSE Capstone Fall 2011 to Spring 2012*)

(*deletes previous session information*)Clear[F, x, varOfInterest, intStart, intEnd, Propor, Deriv, Integ, k]

F[x_] = UnitStep[x - 2] - UnitStep[x - 4]; (*define the input function*)
varOfInterest = x; (*define what variable to operate with*)
intStart = 0; (*defines start time*) intEnd = 2*Pi; (*defines end time*)
Propor = k*F[x] Deriv = D[F[x], varOfInterest] Integ = \[Integral]F[x] \[DifferentialD]varOfInterest

k = 5; (*Define the proportionality constant*)

(*This Plots the Original Function, a proportional function, the orignal function's derivative and the integral of the function*)Plot[{F[x], Propor, Deriv, Integ}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Proportional", "Derivative", "Integral"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large, GridLines -> Automatic, PlotLabel -> "Function and Calculated P, I, D Responses", Frame -> True]

(*use this command to export above graph to GIF file on your desktop change the folder path by adding to "Desktop/..."*)Export["Desktop/all.gif", Plot[{F[x], Propor, Deriv, Integ}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Proportional", "Derivative", "Integral"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large, GridLines -> Automatic, PlotLabel -> "Function and Calculated P, I, D Responses", Frame -> True]]

(*Plot the fnc and derivative*)Plot[{F[x], Deriv}, {x, intStart, intEnd}, PlotLe-
Clear[k, min, max]; min = 0; (*minimum proportionality constant*)
max = 5; (*maximum proportionality constant*)

(*allows an adjustment of the proportional response*)Manipulate[ Plot[{F[x], k*F[x]}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Proportional"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large, GridLines -> Automatic, PlotLabel -> "Function and Variable Proportional Response", Frame -> True], {k, min, max}]

(*this section exports the graphs to .gif files change the folder path by adding to "Desktop/..."*)

(*all responses*)Export["Desktop/all.gif", Plot[{F[x], Propor, Deriv, Integ}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Proportional", "Derivative", "Integral"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large]]


(*original function and calculated response*)Export["F[x]andPropor.gif", Plot[{F[x], Propor}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Proportional"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large]] Export["F[x]andDerivative.gif", Plot[{F[x], Deriv}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Derivative"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large]] Export["F[x]andIntegral.gif", Plot[{F[x], Integ}, {x, intStart, intEnd}, PlotLegend -> {"F[x]", "Integral"}, LegendPosition -> {1.1, -.4}, ImageSize -> Large]]