Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Autonomous Navigational Controller Inspired by the Hippocampus

Maizura Mokhtar, David M. Halliday, and Andy M. Tyrrell

Abstract—**This study uses models of pyramidal neurons in the hippocampus to design a hardware spiking neural network neuro-controller model for the purpose of navigation. The neural network model consists of individual neurons modeled using the two-dimensional bio-inspired Izhikevich algorithm. The network is connected according to the connectivity within the hippocampus region, as this region is one of the regions in the brain that is responsible for path navigation. The information processed by the model helps provide navigation and creates memories. The neural network model is intended to be implemented onto a Field Programmable Gate Array (FPGA) device. This eliminates the need of an operating system to run the network, thus achieving autonomy.**

Index Terms—**Neural network architecture, neural network hardware.**

I. Introduction

The aim of this study is to create a hardware spiking neural network (SNN) model that has the ability to compute positional information based on the theory of path navigation in the hippocampus. A hardware spiking neural network model allows for an autonomous navigational controller to be incorporated directly onto a robot as a plug-in module. This neural network model is intended to be fully implemented on a Field Programmable Gate Array (FPGA) device, thus eliminating the need for an operating system to run the neural network. Implementing the neural network on hardware can provide a faster and efficient method of achieving navigational autonomy.

The neural network model was inspired by the structure of the hippocampus. This is one of the regions in the brain, especially in mammals, that is believed to be responsible for path navigation. A study done on London Taxi drivers indicates the importance of the hippocampus in human path navigation [1].

There are two methods in achieving the neural network model; realistic/bio-physical modeling and bio-inspired modeling. Bio-physical is modeling all aspects of the neuron and the neural network whereas bio-inspired modeling uses simplified models to help reduce the complexity of a bio-physical model. Examples of bio-inspired model are the leaky-integrate-and-fire (LIF) model, the FitzHugh-Nagumo two-dimensional equation and the Izhikevich equations reviewed in [2]. Our neural network model uses the bio-inspired approach of the Izhikevich equation for the single neuron model. This model is more suited to hardware implementation, without losing any of the neuronal dynamics that are relevant [2].

Our hippocampus inspired neural network model emphasizes the ability of individual neurons to replicate the firing patterns of their biological counterpart, whilst they are operating within the network. This differs from other hippocampus neural network models, whereby:

1) Biophysical approach: The compartmental model used by [3] is too complex to be implemented onto a hardware device. A compartmental model of a neuron provides a detailed description of the neuron, consisting of its physical dendritic structure and its Hodgkin and Huxley conductance description within a compartment. If this approach was chosen, there is a possibility that only one neuron with limited number of compartments can be implemented onto an FPGA.

2) Place cell-Place field: The firing pattern assumed by [4] emerged from a population of neurons or place cells in the hippocampus. Their neural network pattern of activity does not encompass the behavior of individual pyramidal neurons found in the hippocampus. The pyramidal neurons are the principle neurons in the hippocampus responsible for neuronal behavior [5]. The firing behavior of neurons in [4] is determined by averaging the rate of the place cells activities. Incorporating individual neuron's behavior to determine the firing rate of the place cells (as in this study) may help provide additional neuronal dynamics to the place cells and place fields representation.

This paper briefly describes the steps taken in creating the hippocampus-inspired neural network model. Section II begins with the theory of the hippocampus neurons and the design of the single neuron model. This is followed by the investigation of neuron to neuron coupling that creates the neural network model. Section III describes the VHDL simulation results of the neural network model applied to an agent going through a maze. The final two sections conclude with the ideas and motivation that can help towards the development of the autonomous hardware system.

Manuscript received January 30, 2007.

M. Mokhtar is with the Intelligent Systems Group, Department of Electronics, The University of York, York YO10 5DD UK (phone: 44-1904-432830; e-mail: mm520@ ohm.york.ac.uk).

D. M. Halliday is with the Intelligent Systems Group, Department of Electronics, The University of York, York YO10 5DD UK (e-mail: dh20@ohm.york.ac.uk).

A. M. Tyrrell is with the Intelligent Systems Group, Department of Electronics, The University of York, York YO10 5DD UK (e-mail: amt@ohm.york.ac.uk).

II. Hippocampus-inspired Neural Network

The objective of the research is to create an autonomous hardware navigational module. The model is based on the hippocampus proper region or the Cornu Ammonis, CA region [5].

A. Hippocampus proper

The hippocampus proper consist of the Cornu Ammonis CA layers. The flow of information in the hippocampus proper is known as the trisynaptic pathway. This pathway is one of the processes within the hippocampus that is responsible for storing and retrieving information in memory [5]. The trisynaptic pathway starts with:

1) Perforant pathway: The propagation of the afferent input from the enthorhinal cortex, EC layer II/III, to the CA3 and CA1 region. These inputs carry information about the environment [5].

2) Recursive collaterals: The recursive collaterals are the recursive connections within the CA3 region. The recursive collaterals cycle inputs within the CA3 region to allow the hippocampus to form episiodic memory or memories of episodes. Episodic memory is how the hippocampus associates an event in time to an event in memory, thus allowing reconstruction when the event is recalled from memory. The recurrent collateral also refreshes memory to ensure long term memory [6].

3) Schafer collateral: Connection from the CA3 region to the CA1 region. The CA1 region also receives information from the EC_IN. The CA1 region acts as the short term memory buffer to store current responses [5].

4) Final Connection: The final connection is the projection of information both from the recurrent collateral of the CA3 region and the output of the CA1 region back to the enthorhinal cortex but to the EC layer IV/V so that this information can be passed to other regions of the brain [5], [6].

B. Hippocampus-inspired Neural Network

The hippocampus-inspired neural network model is based on the trisynaptic pathway described in Section *A*. The trisynaptic pathway is modeled with a small number of neurons in each region:

1) One neuron in the EC layer II/III region (EC_IN)
2) Recurrent collateral between three CA3 neurons.
3) One neuron in the CA1 region
4) One neuron in the EC layer IV/V region (EC_OUT)

The connections between these six neurons are shown in Fig. 1. This group of neurons is referred to as a Neuron Assembly (NA). The complete network consists of 8 identical NAs arranged at points on a compass. This representation is chosen because it allows the network to provide directional control to an agent. The correct direction is indicated by an increase in firing frequency of the neurons in the NA that represents that direction. Subsequently, this leads to mutual inhibition of the NA in the opposite point.

This representation was based on the concept of head direction cells [4], which stated that head direction cells are neurons whose activity is based upon orientation of a rat's head. Each head direction cells fire maximally when the rat is facing in a specific direction regardless of its behavior [4], [7].

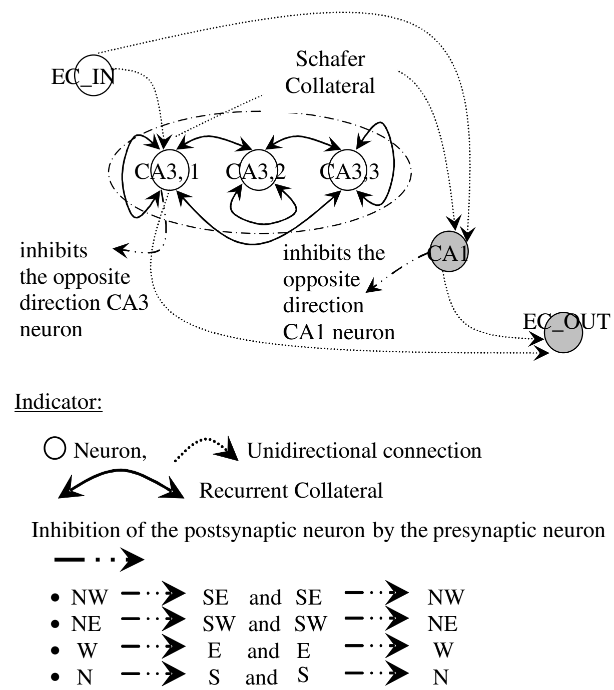

Fig. 1. The neural network within one Neuron Assembly (NA). One NA connects to its opposite direction NA via the inhibitory connection at CA3 and CA1 neuron. Summation of the firing rate of the CA1 and EC_OUT neuron (shaded neurons) helps indicate the next desired move.

C. Neuron Equations

The algorithms implemented to model the CA1 and CA3 hippocampus neuron behavior are formed from the equations for the membrane potential, *v* and the membrane recovery potential, *u*, introduced by Izhikevich [8]

$$v : {dv}/{dt} = 0.04v^2 + 5v + 140 - u + I \tag{1}$$

$$u : {du}/{dt} = a(b \cdot v - u) \tag{2}$$

$$\text{if } v \geq 30, \text{ then } v = c \text{ and } u = u + d. \tag{3}$$

Equation 1 was chosen by Izhikevich because this equation has the ability to fit spike initiation dynamics of cortical neurons [8]. In this project, values of the variables a,b,c and d in (2) and (3) are tuned so that the neuron models display the different firing patterns seen at the hippocampus pyramidal neuron, CA3 and CA1 [5], [10]. The values of the variable a,b,c and d of our model are:

1) CA3 neuron:

$$a = 0.001 * x/5 \tag{4}$$

$$b = 0.2 \tag{5}$$

$$c = -65 + \left(12.5 + \sqrt[3]{x}\right) * \left(0.5 + 1/\sqrt{x}\right) * \left(0.5 + 1/\sqrt{x}\right) \tag{6}$$

$$d = 8 - \left(6 + \sqrt[3]{x}\right) * \left(0.5 + 1/\sqrt{x}\right) * \left(0.5 + 1/\sqrt{x}\right) \tag{7}$$

If *Injected Current* $\leq$ *0.4nA* then *x = Injected Current*1.0*, else *x = Injected Current*2.0*

2) CA1 neuron:

$$a = 0.025 \tag{8}$$

$$b = 0.2 \tag{9}$$

$$c = -55 + \left(\text{InjectedCurrent}/2 * 0.55255^2\right) \tag{10}$$

$$d = 8 - \left(6 * 0.55255^2\right) \tag{11}$$

D. Membrane Potential of the CA3 and CA1 Single Neuron Model Results

The single neuron model was first simulated in Matlab®. As a verification stage the membrane potential of the single neuron models were compared against the membrane potential of the bio-physical model of the hippocampus neuron CA3 and CA1 model found in [10]. The characteristics of the two single neuron models described by [10] are:

1) CA3 neuron model: The CA3 neuron behavior is characterized by two frequency versus injected current curves shown in Fig. 10 of [10]. Results of our CA3 neuron model are :

TABLE I

CA3 : CURVE 1 BURST FREQUENCY, BF VS SOMATIC INJECTED CURRENT, IC

Somatic IC (nA)	Results of Matlab®: BF (Hz)	Results of VHDL: BF(Hz)
0.175	0.56	1.429
0.2	0.8	2.500

Rheobase of about 0.15nA

TABLE 2

CA3 : CURVE 2 ACTION POTENTIAL FREQUENCY, AF VS SOMATIC INJECTED CURRENT, IC

Somatic IC (nA)	Results of Matlab®: BF (Hz)	Results of VHDL: BF(Hz)
0.4	20	20
0.8	60	60

An important behavior of the CA3 neuron is the switching between burst firing to action potential firing (or repetitive firing) [10]. This was observed in both the Matlab® model and the VHDL model of the CA3 neuron. The differences in results shown in Table 1 resulted from the translation of the Matlab® model to VHDL. The differences in results are discussed in Section V.

2) CA1 neuron model: When the CA1 neuron is injected with current <1nA at the soma, rhythmic trains of action potentials are observed. Larger currents, ≥1.15nA causes the neuron to respond with bursts of action potential [10].

TABLE 3

CA1 : ACTION POTENTIAL FREQUENCY, AF VS SOMATIC INJECTED CURRENT, IC

Somatic IC (nA)	Results of Matlab®:	Results of VHDL:
0.25	17	20
0.4	26	33
0.8	Doublet of 30Hz	Doublet of 30Hz
1.2	3 spikes per burst (BF=30Hz)	3 spikes per burst (BF=30Hz)

Our results show that the Matlab® simulation of the bio-inspired CA3 and CA1 neuron model using the modified Izhikevich algorithm has similar behavior to the bio-physical neuron model described in [10]. These similarities give confidence to our bio-inspired approach to the single neuron model. The next section describes the design of the hippocampus-inspired neural network model.

E. Neural Network

To create a neural network, single neurons must communicate through synaptic transmission between the transmitting (presynaptic) neuron and the receiving (postsynaptic) neuron via a post-synaptic potential (PSP). The postsynaptic potential can either be excitatory or inhibitory. An excitatory postsynaptic potential (EPSP) is a positive influence on the postsynaptic neuron, providing an increase in the membrane potential of the postsynaptic neuron. Inhibitory postsynaptic potential (IPSP) inhibits the postsynaptic neuron [11]. In our neural network model, the EPSP is modeled as a small percentage (8%) of the presynaptic neuron membrane potential whilst IPSP is (-1 x EPSP).

A weight factor is included to control the strength of connection between the presynaptic neuron and the postsynaptic neuron. Two learning rules are used to determine the value of the weights. The two learning rules are the Hebbian Timing-based learning rule or the Spike Timing Dependant Plasticity (STDP) and the Hebbian Rate-based learning rule.

1) Spike Timing Dependent Plasticity, STDP: STDP is dependent on the time of spikes between the pre and postsynaptic neurons. If a presynaptic spike precedes a postsynaptic spike within the time constant, there will be an increase in weight. Otherwise, the weight is either unchanged or depletes [12], [13].

2) Hebbian Rate-based Learning: Hebbian Rate-based Learning is dependent on the mean firing rate of the neurons. If the mean firing rates of the presynaptic and postsynaptic neurons are both high, the connection between these neurons are strengthened, otherwise they are reduced [13].

Both of these learning rules are used in our neural network model. STDP is used to adjust the weights when the agent is in a particular moment or episode in the environment and is trying to make its decision based on the propagation of spikes between each region. If the propagation of spikes is consistent, the weights are consistently increased. If the propagation of spikes is weak, the weights will be reduced. The NA with the strongest connections or highest activation will provide a maximum mean firing rate of its output neurons (CA1 and ECOUT neuron), thus identifying the winning NA. The mean firing rate for each NA is calculated at the end of the episode. A Hebbian Rate-based Learning rule is used to reinforce the weights of the output neurons of the winning NA (the direction which the agent defines to be correct). The weights of the opposite direction of the winning NA are subsequently reduced. Implementation of these learning rules is described in [14].

F. *Results of the Hippocampus-Inspired Neural Network*

Results from our Matlab® single neuron models are similar to the bio-physical neuron model described in [10]. These similarities are important for the design of the hardware spiking neural network model. The complete neural network model is constructed by connecting together the four pairs of coupled NAs. Two NAs are coupled so that an excitation of one direction will inhibit the NA of the opposite direction. The four coupled NA pairs are North and South, East and West, South East and North West and South West and North East. If there is a path in the direction which the NA represents, the neurons in that particular NA will show an increase in its firing rate. If there is a wall instead, there will be either a decrease in its firing rate or no firing at all.

The performance of the hippocampus-inspired neural network model is judged by the navigational skill of the agent going through a maze. The test was done on three different sizes of maze and the agent encountered a different maze on each trial. The size of the maze is identified by the number of step in the x-axis, *x,* and the y-axis, *y*, of the maze, (*x* x *y*). The capability of the neural network was evaluated by how long it took for the agent to complete each maze by calculating the number of episodes per maze, Table 4. One episode is t = 500ms. Even with varying sizes of maze, the agent still managed to navigate its way from the start to the end of the maze within reasonable times.

TABLE 4
HOW LONG DID THE AGENT TOOK TO COMPLETE THE MAZE

Size of maze	Number of trials	Average number of episodes
10 x 10	20	967
15 x 15	4	2386
20 x 20	5	5066
30 x 30	1	8462

III. NEURAL ACTIVITY OF THE HIPPOCAMPUS-INSPIRED NEURAL NETWORK VHDL MODEL

As the motivation of this project is to create a hardware model, the Matlab® model was then translated to a hardware synthesizable language or VHDL (Very High-speed Integrated Circuit Hardware Description Language). The VHDL model was then emulated with the maze navigation task. As stated above, there are differences in the frequency of the CA3 neuron before it switches to repetitive firing behavior. The slope of the frequency is the same but the VHDL model results have three times the frequency of the Matlab® model. Because of this, the synaptic input to the CA3 neuron in the neural network model was reduced to ¼ of its original value.

Fig. 3 shows the output of the neurons in NA(N), (S), (E) and (W) of the neural network in one episode, indicated by the shaded box in Fig. 2, when the agent, represented by the black diamond goes through the path in a 5x5 maze in Fig. 2.

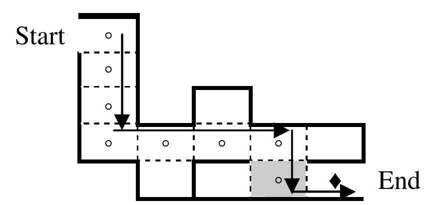

Fig. 2. A path in a 5x5 maze used to test and simulate the VHDL neural network model. The circles represent the previous episodes and the arrows show the path the agent (black diamond) took within the maze.

The summation of firing rates of the CA1 and EC_OUT neurons of the NAs are used to determine the next desired move at one episode. At this point in the maze, the agent has to choose either to move to the north, west or east. Fig. 3 shows that NA(E) has greater excitation compared to the others. Therefore, the agent chooses east as its next move. The CA1 and EC_OUT neurons of NA(N) and NA(W) shows small or little excitation because its prior moves to the south and east has caused the inhibition of weights within the NAs. Therefore, the input for the EC_IN could not substantially excite both the CA1 and EC_OUT neurons. Even though the CA1 and EC_OUT neurons are quiet, but there are still excitations of the CA3 neurons. CA3 neurons are excited because the recurrent collateral of the CA3 is trying to consolidate this event within its synapses.

The path the agent chose in the maze showed that the agent followed a one-way path without unnecessary backtracking or crossing over walls. One episode, t_{period} has 500 steps. If there is a path, EC_IN neuron will be excited otherwise the EC_IN neuron will be silent when there is wall. Walls are indicated by continuous thick lines and one episode is within the broken thin lines. NA(NW), (NE), (SW) and (SE) were simulated but the since these direction was not activated, the neurons within these NAs are inactive and not shown.

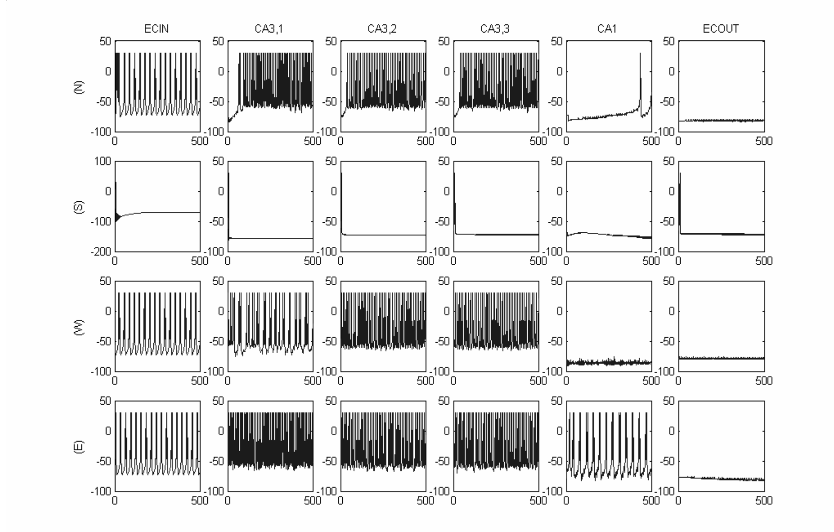

Fig. 3. VHDL simulation results of the neurons in the neural network at one episode (shaded box) applied to an agent going through a path in Fig. 2. Y-axis in all graphs represents the membrane potential in millivolts (mV). X-axis represents a sequential time step between 0 to 500 (steps), which is t_{period} = 1.

IV. Hardware Implementation of the Hippocampus-inspired Neural Network

Similar design methods are applied to the creation of the hardware model. These methods are:

1) Design of the single neuron model
2) Connecting the neurons to create the neuron assembly. There will be eight of these assemblies, each representing a pointer on a compass.
3) The eight NAs are assembled together to create the complete hippocampus-inspired neural network model
4) Synthesizing the neural network model onto an FPGA.

Output of the single neuron VHDL models are shown in Tables 1-3. Design methods number 2-4 are work in progress. The successful FPGA design will be incorporated into a robot to act as an on-board neuro-controller device that helps the robot navigates within its environment. The ability to implement the neural network onto a single FPGA device shall constitute the success of autonomy.

There are issues in translating the software model to hardware. The issues in realizing any artificial neural network model to hardware and its advantages have been discussed in [15]. For this project, the issues are:

1) Word Length and Precision: The word length chosen for the hardware spiking neural network model is 32-bit signed binary fixed point notation. This notation consists of 1 sign bit, 7 bits for decimal and 24 bits for fraction.

This notation was chosen because it is much easier to design the neural network in this notation instead of floating point notation. It is also believed to be unnecessary to require the precision of the floating point notation.

The fixed point notation provides a minimum integer value of $\pm(5.9605\text{x}10^{-8})$ or $\pm(2^{-24})$ milivolts and a maximum integer value of $\pm$ $(128-2^{-24})$ milivolts. These values are suitable to represent the membrane potential of a neuron because the membrane potential of a neuron is believed to be within the range of ±100 milivolts

2) Mathematical Operation: Not all mathematical operations can be directly synthesised onto the FPGA or they require large number of logic slices for each operation. Example of these function are exponential, *e*, cube roots, square roots and division.

3) Timing Constraints: The number of clock cycles to implement one neuron operation is important for the operation of the complete neural network. The clock cycle is dependent on the clock frequency. It is vital that one cycle of operation is completed in 1ms, as one time step of (1) and (2) is equivalent to 1ms [16].

4) Size and cost of the FPGA: Design of the neural network model within the constraints of a specific FPGA. The device under consideration is the Xilinx Virtex IIPro XC2VP30.

The output produced by ModelSim® VHDL simulation of the hippocampus-inspired neural network (Section III) is in 32-bit fixed point notation. The binary output values were converted in Matlab® for plotting against 500 time step in one episode. Time, *t* in Fig. 2 (b) is not equivalent to 1ms but is t = 130µs. This is because each mathematical operation in the neuron and the network algorithm operates with respect to the clock frequency. The clock frequency used for the VHDL simulation is the clock frequency of the intended device (100MHz).

Further work on trying to implement the VHDL model to the FPGA device has lead to the conclusion that the VHDL model in Section III will not be the final model used for implementation. This is because the intended device, the Xilinx Virtex IIPro XC2VP30 has insufficient number of logic slices to implement this design. The VHDL model requires almost 150% of the number of slices of the device. Changes will be made to the design so that it can accommodated on the FPGA. The Xilinx Virtex IIPro XC2VP30 has two embedded PowerPCs. In order for the network to fit onto this device, it is proposed that the single neuron model (CA3 neuron and CA1 neuron) and the network itself are implemented on hardware and the weight updates are implemented on the embedded PowerPC. Results from the implemented neural network model will be available in future publications.

V. Discussion

The key behavior of the hippocampus neuron is the ability to switch between burst firing and repetitive firing. The best translation of the software model to VHDL with respect to the limitations described above is the model that produced the results in Table 1-3. There are differences in the results for the CA3 neuron model, the VHDL results of the burst firing frequency are three time larger than the Matlab® results (Table 1). One possible explanation of this results is that x in (6) and (7) for the VHDL model are constantly kept at *Injected Current*2.0*. This is because the rheobase value for the VHDL CA3 neuron model is found to be twice the Matlab® model. However, there is a similarity in the slope of the frequency curve of the CA3 VHDL neuron model and the Matlab model®. Reducing the synaptic input to the CA3 neuron to ¼ of its original value can help preserve the neuron behavior within the network. The VHDL model also preserves the switching behavior between burst firing and repetitive firing at somatic injected current around 0.25nA.

The aim of this research is to model the autonomous navigational controller based on the biological nervous system. The region responsible for such behavior in the brain is the hippocampus [1], [4]. This region is also believed to be responsible for mammalian learning and memory [5].

There are two methods in designing a biological neural network model, either a bio-physical architecture or a simplified bio-inspired architecture. The bio-inspired modeling approach of the two-dimensional Izhikevich algorithm was chosen for the design of our single neuron model. This algorithm can reproduce a range of neuron behavior with simple algorithms [2], [8].

The hippocampus-inspired neural network model was first designed and tested in Matlab®. This software neural network model was then converted to a language that is suitable for hardware synthesis, VHDL. Methods to create the neural network model started with the single neuron model. The success of the single neuron model was measured by how close the model behaves to a previous bio-physical model [3], [10]. The CA3 neuron model fires in bursts whilst the CA1 neuron fires repetitively. Entorhinal cortex layer II/III (EC_IN) behaves similarly to the CA1 neuron and the entorhinal cortex layer IV/V (EC_OUT) is similar to the CA3 neuron. Successful single neuron models were then coupled to create the neural network.

The neural network is to be fully implemented onto a FPGA device, thus allowing the neural network to be independent of an operating system.

The future success of this research project will be measured by its capability to act as an autonomous hardware device that can independently navigate and learn about its environment in an unsupervised fashion. The learning of the neural network on the device is governed by the dynamics of the hippocampus region in the brain. It is hoped that in the future, the hippocampus-inspired neural network hardware device will be incorporated to a robot, enabling autonomous navigation.

References

[1] E. A. Maguire, D. G. Gadian, I. S. Johnsrude, C. D. Good, J. Ashburner, R. S. J. Frackowiak, and C. D. Frith, "Navigation-related structural change in the hippocampus of taxi drives", PNAS., vol. 97, pp 4398-4403, 2000.

[2] E. M. Izhikevich, *Dynamical systems in neuroscience: the geometry of excitability and bursting*. Cambridge, Mass: MIT Press, 2006, ch 4.

[3] R. D. Traub, D. Contreras, M. O. Cunningham, H. Murray, F. E. N. LeBeau, A. Roopun, A. Bibbig, W. B. Wilent W.B, M. J. Higley, and M. A. Whittington, "Single-Column Thalamocortical Network Model Exhibiting Gamma Oscillations, Sleep Spindles, and Epileptogenic Bursts", J. Neurophysiol., vol. 93, pp 2194-2232, 2005.

[4] A. Arleo, W. and Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," *Biological Cybernetics,* vol. 83, pp 287-299, 2000.

[5] J. Daniel and D. G. Amaral, "Hippocampus" in *The synaptic organization of the brain,* 4th ed., G.M. Shepherd, Ed. New York: Oxford University Press, 1998, pp 417-510.

[6] L. Wiskott, M. J. Rosch, and G. Kempermann. (2004, December 30).. "What is the Functional Role of Adult Neurogenesis in the Hippocampus?" in *Cognitive Science EPrint Archive (CogPrints).* [Online]. Available: http://cogprints.org/4012/wiskottetal2004.pdf.

[7] P. E. Sharp, H. T. Blair, and M. Brown, "Neural Network Modeling of the Hippocampal Formation Spatial Signals and Their Possible Role in Navigation: A Modular Approach," *Hippocampus,* vol 6, pp 720-734, 1996.

[8] E. M. Izhikevich, "Which model to use for cortical spiking neuron?," *IEEE Trans. Neural Networks,* vol 4, pp 1063-1070, September 2004.

[9] R. D. Traub, D. Schmitz, J. G. R. Jefferys, and A. Draguhn, "High-Frequency Population Oscillations are Predicted to Occur in Hippocampal Pyramidal Neuronal Network Interconnected by Axoaxonal Gap Junctions.," *Neuroscience,* vol. 92, pp. 407-426, 1999.

[10] R. D. Traub, R. K. S. Wong, R. Miles, and H. Michelson, "A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances," *J. Neurophysiol.,* vol. 66, pp. 635-650, 1991.

[11] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*, 1st ed, Cambridge: Cambridge University Press, 2002, pp 1-6.

[12] E. A. DiPaolo, "Spike-Timing Dependent Plasticity for Evolved Robots," *Adaptive Behavior*, vol. 10, no. 3-4, pp. 243-263, 2002.

[13] D. Floreano and J. Urzelai, "Evolutionary Robots with On-line Self-organization and Behavioral Fitness," *Neural Network,* vol. 13, pp 431-443, June 2000.

[14] M. Mokhtar, D.M. Halliday, A. M. Tyrell, "Hippocampus Neurons and Place Cells/Place Field Representation to Provide Path Navigation," in *Int. Joint Conf. on Neural Networks*, Florida, 2007.

[15] P. Moerland and E. Fiesler, "Neural Network Adaptations to Hardware Implementation" in *Handbook of Neural Computation*, E. Fiesler and R. Beale, Ed. New York: Institute of Physics Publishing and Oxford University Publishing, 1997, E1.2:1-13.

[16] M. A. D'Haene, "A Framework for Parallel Event Driven Simulation of Large Spiking Neural Networks" in *7e FirW Doctoraatssymposium,* pp. on CD.