TOWARDS AN "EARLY NEURAL CIRCUIT SIMULATOR": A FPGA IMPLEMENTATION OF PROCESSING IN THE RAT WHISKER SYSTEM

Brian Leung, Yan Pan, Chris Schroeder[†], Seda Ogrenci Memik, Gokhan Memik, Mitra JZ Hartmann[†*],

Department of Electrical Engineering and Computer Science
†Department of Biomedical Engineering, *Department of Mechanical Engineering,
Northwestern University
2145 Sheridan Road, Evanston, IL, 60208
email: [brian.leung, panyan, cshroeder, f-memik, g-memik, m-hartmann] @northwestern.edu

ABSTRACT
We have constructed a FPGA-based "early neural circuit simulator" to model the first two stages of stimulus encoding and processing in the rat whisker system. Rats use tactile input from their whiskers to extract object features such as size and shape. We use the simulator to examine the plausibility of the hypothesis that neural circuits in the rat's brain compute gradients of radial distance across the whisker array to make predictions about the environment. This prediction could be a component of a feed-forward signal that guides the navigation behavior of the rat. The use of a FPGA is highly suitable for such an application, because the computation involved in this system is a massively parallel problem. For our applications, we determined that a Cyclone II FPGA could simulate up to 14 neurons in parallel in just 265 ns achieving a 386-fold speedup over the software implementation of the same model.

1. INTRODUCTION
Animals use movements to acquire and refine incoming sensory data in order to construct meaningful representations of the environment. During exploratory behaviors, each movement an animal makes aids in the extraction of task-relevant sensory data. As of yet, however, neuroscientists have little understanding of how the body and brain work together to acquire, encode, and process the sensory data generated through movement. The dynamics of animal movement are difficult to simulate, and thus the dynamics of sensory acquisition behaviors have remained largely uncharacterized and unquantified.

It is becoming increasingly clear, however, that movement is as critical a part of sensing as the sensory receptors themselves. Some examples include eye movements for vision [1] and hand movements during touch [2]. The movements that occur during sensory data acquisition define the sensory input that the early stages of the nervous system must encode and constrain neural algorithms that can occur at higher levels of processing. Thus, development of robust computational models of the early stages of sensorimotor processing in active sensing behaviors is essential to further study sensory-motor relationships.

As an initial step towards this long-term goal, we have constructed a FPGA-based neural circuit simulator to model the early stages of stimulus encoding and processing in the rat whisker system, which is one particular model of active sensing. The purpose of this system is to examine the possibility that rats are using gradients of radial distance across the array to make predictions about the environment while moving. This prediction could be a component of a feed-forward signal to guide the rat's navigation behavior.

The use of a FPGA is perfect for such an application, because the computation involved in this system is a massively parallel problem. Furthermore, a FPGA-based system can be easily reconfigured to test other hypotheses about the rat whisker system or even other sensory modes such as vision or audition. Our main goal is to construct a versatile framework, where possible computational models of neural processing can be tested and evaluated efficiently. In this paper, we will particularly illustrate the application of this framework to test a specific whisker sensory input encoding, which we hypothesize to aid navigation of rats.

The present manuscript is organized as follows. Section 2 provides background on the rat whisker active sensing system model. Section 3 details the FPGA implementation of the early neural circuit simulator. We present the performance results in Section 4. Section 5 briefly reviews some related work and Section 6 concludes the paper.

2. WHISKERS AS AN ACTIVE SENSING SYSTEM
2.1 Neuroscience of the Whisker System
As shown in Fig. 1 (a) rats have approximately 30 whiskers on each side of their face [3]. When exploring their environment, rats sweep their whiskers back and forth against objects at frequencies between 5 and 25 Hz [4] in a behavior known as "whisking". Using only its whiskers, rats can extract accurate information about object position, orientation, size, shape, and texture (i.e., the object's spatial properties). Thus, the whisker system represents an efficient active-sensing system that can operate in complete darkness.
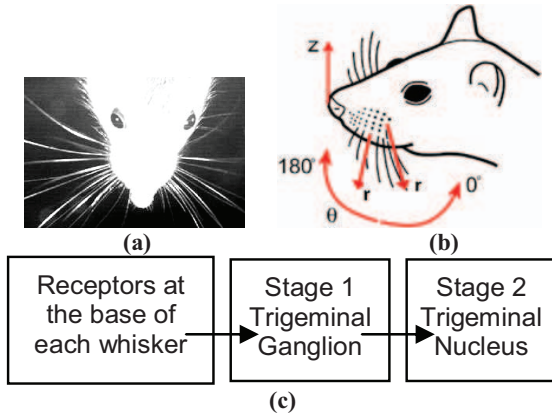
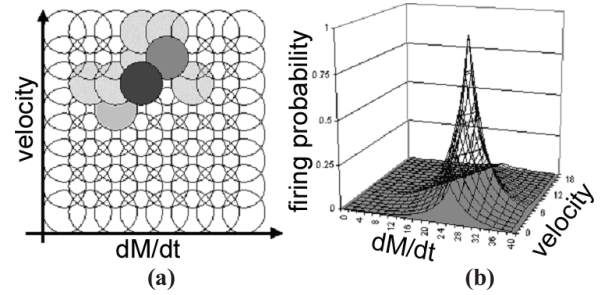**Fig. 1. Rat whisker array and associated neural pathway.**

Fig. 2. Probability density functions for state space encoding of whisker dynamics. (a) Overlapping 2D radial basis functions are used to completely cover the moment-dot/velocity portion of the state space. Shaded units represent neurons that fire (with a particular probability) in response to the whisker's presence in that particular region of the state space. (b) Typical firing probability of a neuron based on the combination of velocity and position variables. In this example, the firing probability for each variable axis exhibits a normal distribution.

Fig. 1 (b) illustrates that the whiskers are arranged in an orderly grid on the rat's face, and also shows the cylindrical coordinate system that is often used to describe whisker movements. θ is the angular position of each whisker, z is the height of each whisker, and r is the radial distance out along the whisker length. Rat whiskers do not have sensors along their length. Instead, mechanoreceptors are found only at the base of each whisker [5]. It is proposed that the mechanoreceptors provide information about angular position (θ) and angular velocity (dθ/dt), as well as the bending moment (M) and rate of change of bending moment (dM/dt) at the whisker base [6-8].

Given that whiskers do not have sensors along their length, how does the rat determine the radial distance r to an object? Recent work from our laboratory has shown that the radial distance from the base of the whisker to an object can be determined by monitoring the derivative of moment with respect to angular position [6, 9] as in Eq. 1:

$$r = C\frac{d\theta}{dM} = C\frac{d\theta}{dt}\frac{dt}{dM} \qquad (1)$$

where C is a constant related to the whisker stiffness.

Fig. 1 (c) illustrates how the mechanical information from receptors at the whisker base travels through the brain. The first brain stage is a group of neurons called the "trigeminal ganglion." The second brain stage is a group of neurons called the "trigeminal nucleus."

2.1.1. First processing stage: the trigeminal ganglion.
As described above, the mechanical information at the base of the whisker is thought to include θ, dθ/dt, M and dM/dt. However, many studies have already demonstrated that first-stage neurons (in the trigeminal ganglion) do not "purely" encode these variables. Instead, the first-stage neurons output a sharp voltage spike in response to combinations of these mechanical variables. To explain these earlier data, we have recently suggested that an individual neuron encodes a particular *dynamical state* of the whisker. For example, Fig. 2 (a) illustrates a first-stage

neuron that responds most strongly to the combination of dM/dt and dθ/dt in the circular black region, and less strongly to combinations in the neighboring gray regions. The diversity of neuronal response characteristics found in the trigeminal ganglion, spanning the full range of all the variables, is highly suggestive of a distributed state space representation of whisker movements. Each neuron will have a firing probability (that is, a probability of outputting a voltage spike) based on different combinations of variables, as shown in Fig. 2 (b). It is important to note that neurons in the first stage of processing produce a voltage spike in response to deflections of <u>one and only one</u> whisker [10].

2.1.2. Second processing stage: the trigeminal nucleus
Presently, very little is known about the computations that occur in the second stage of processing of whisker information, that is, in the trigeminal nucleus. However, it is well established that neurons of the second stage respond to <u>more than one</u> whisker [11]. This means that they will produce a voltage spike when any of several different whiskers is moved or deflected. Thus, the neurons in the trigeminal nuclei can be thought of as responding to combinations of different patterns of input across multiple whiskers. We have recently suggested that the neurons of the trigeminal nucleus may be computing spatial gradients of radial contact distance (r) across the array. This idea is based on the insight that as an animal moves through the environment, spatial and temporal gradients of sensory data are related through the velocity of the moving sensory surface [12]. These gradients, as expressed in the complete derivative, provide an inviolate mathematical description of information flow over moving sensory surfaces. Computing the complete derivative at multiple spatial scales would allow the animal to predict the stimulus that it will measure

in the next sensory instant, conferring tremendous survival advantage.

2.2. Computational Model of the Whisker System

The computational model of the whisker system is divided according to the two-stage model of the nervous system.

2.2.1 First stage (trigeminal ganglion) model

Each of the roughly 30 whiskers on the rat's face has associated with it between 1500 and 2000 ganglion neurons, If our *dynamical state* hypothesis is correct [6], these neurons must cover (in a pair-wise manner) the full state space of [θ, dθ/dt, M, dM/dt]. This is equivalent to approximately 50,000 whisker-responsive neurons. Calculating the probability distribution shown in Fig. 2 (b) for each of these 50,000 neurons is already a large computational task, requiring a minimum of 0.394 seconds in MATLAB for a single input set. This would not be able to meet the frequency requirement of 25 Hz. In this work, we use only one combination of the six possible state-space pairs to model responses in the trigeminal ganglion, namely, pair-wise combinations of dM/dt and dθ/dt (angular velocity). The rationale for this choice can be seen in Eq. 1. The radial distance of object contact out along the whisker is related to the ratio of dM/dt and the angular velocity. This means that in Fig. 2, each neuron that responds to a particular combination of dM/dt and velocity is in essence encoding a particular radial distance.

To reiterate, we have limited the computations in the present manuscript to 1/6 of those that are likely to occur in the real rat, modeling only neurons representing combinations of dM/dt and dθ/dt. In future work, the model will be expanded to include neuronal cell types covering all pair-wise combinations of state variables for the entire trigeminal ganglion.

2.2.2 Second stage (trigeminal nuclei) model

A ubiquitous feature of early neural circuits across modalities (vision, audition, somatosensation) is the presence of *lateral inhibition*. In simple terms, lateral inhibition has the effect of making a neuron respond to differences in input across the spatial extent of a receptor array, instead of to the absolute value of the input. This is accomplished because the neuron's response to a central receptor is enhanced, while responses to surrounding receptors are suppressed. The neural mechanisms of lateral inhibition are not relevant here. Instead, it is simply important to understand that there is considerable evidence that neurons are likely to respond most strongly to spatial gradients of input across a receptor sheet.

Our laboratory has recently suggested that the purpose of these gradients may be to aid the brain in computing the "complete derivative" [12]. In the case of the rat whisker array exploring a static environment, the evolution of the sensory flow field via the complete derivative can be written as a function of the radial distance *r*:

$$\frac{dr}{dt} = V \bullet \nabla r \qquad (2)$$

In Eq. 2, V is the velocity of the rat relative to the object, and ∇ is the gradient operator. Eq. 2 indicates that computing local spatial derivatives allows prediction of the sensory data that will be acquired in the next sensory instant. In other words, Eq. 2 would allow the rat to form a quantitative expectation value of the future against which it can match its sensory input.

In order for this computation to be of value, however, gradients will need to be computed across different length scales, and in different directions. In addition, sensory surfaces can move with different velocities, so computing the total derivative is likely to involve computing many local velocities. These requirements for massive parallelism to perform the computation at many different scales essentially mandates the use of FPGAs.
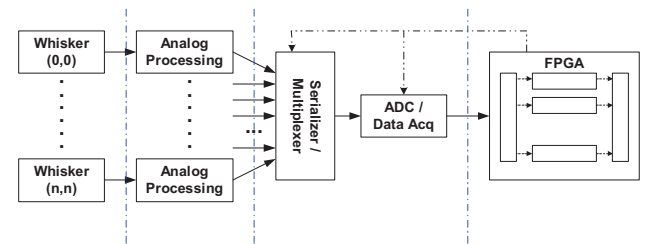

Fig.3. System architecture overview.

3. SYSTEM ARCHITECTURE

Our neural circuit simulator system consists of four stages—the robotic whisker matrix, the analog processing stage, the serializing/MUXing and ADC stage, and the FPGA processing stage. Fig. 3 contains a general overview of the system architecture.

3.1 The "Robotic" Whisker Matrix

We constructed a 5x5 whisker matrix as shown in Fig. 4. The whiskers are separated from one another by a distance of 1.17 inches. This matrix collects whisker data to be processed. For this application, we only considered rotation in one plane, so we are only analyzing one dimension of moment using a single row of the whiskers (1x5). Whiskers were rotated at a fixed speed of approximately 50 deg/sec, mimicking the brushing of rat whiskers on a nearby surface. Each whisker generates a voltage reading proportional to the bending moment (M). This reading is similar to a signal a neuron would send when a whisker is bent.

3.2 Analog Processing

We pass the incoming voltage signals through a low-pass filter to suppress noise. Then, we differentiate the smoothed voltage data using a differentiator. Analog differentiators are used because they provide better accuracy by operating on the analog signal while saving computation resources on the FPGA. In turn, we can dedicate the FPGA resources to other data processing operations. In addition, by adjusting the gain of the differentiator, we can apply the division operation required for the calculations in the same step.
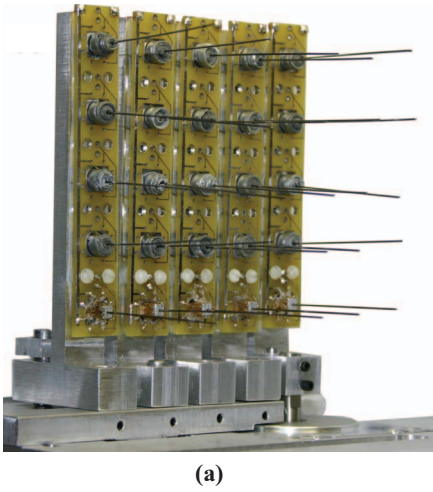
(a) **(b)**

Fig. 4. (a) Constructed Whisker Matrix and (b)one whisker.

3.3. Serializer/Multiplexer and ADC/Data Acquisition

A serializer and MUX are necessary to pass the data into the FPGA because there are not enough pins to pass all the data in parallel. Controlled by the FPGA, the MUX selects signals in sequence to be converted by the ADC. This makes efficient use of the ADC, but incurs extra latency in the data acquisition phase. However, as the ADC is generally fast with a sampling rate usually above 100K samples per second [13, 14], multiplexing is a valid solution.

3.4 FPGA Processing

We used the Cyclone II FPGA on an Altera DE2 board to implement our system to process the voltage readings from the whisker matrix. We utilized the 32-bit single precision subtraction, 32-bit single precision multiplication, 32-bit single precision addition, and 32-bit D flip-flop modules from the IP Megafunctions in the Quartus II software to construct the single processing unit shown in Fig. 5. All of the mathematical operations are floating point operations to ensure that there is minimal degradation in the precision of the data calculations.

The main function of the single processing unit is to predict and compare radial distances using simulated neural data provided by the robotic whisker matrix. The subtraction, multiplication, and addition processes are used to make a prediction of the next radial distance. The results are stored, and compared on the next clock cycle when the actual value is transmitted by the whisker matrix. The algorithm was implemented using the model described in Section 2.2.

In a system where more than one whisker is present, we construct a new processing unit by combining several single processing units. For example, in the case of making predictions for 5 whiskers, 5 of these single processing units would be connected in parallel.

As we expand the algorithm for more whiskers, more resources are used. In the event where we approach the
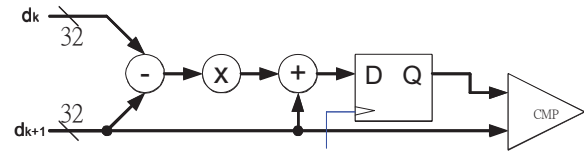

Fig. 5. Single Processing Unit

limit of the FPGA resources, we partially serialize the algorithm by reusing a module that performs the maximum number of parallel computations on the FPGA.

4. RESULTS

In the following, we first present the implementation results of our FPGA-based neural circuit simulator. Next, we present how this simulator has been used for performing the slope prediction based on the hypothesized computational model.

4.1 FPGA Performance

We used the Quartus II 6.1 software on a Pentium M 1.6 Ghz computer to implement the data processing algorithms of the neural circuit simulator. Table 1 contains the resource usage and the performance of the FPGA for our design used for predicting and comparing the radial distances. From the results, it is evident that the logic usage of the FPGA is strained when single precision data processing of 14 whiskers are in parallel. In the event that more than 14 whiskers are present, we partially serialize the computation by reusing the 14 parallel computation channels over multiple iterations.

The number of neurons processed and resources used on the FPGA are directly proportional. As we increase the number of processing units in the system, the cycle time remains about the same with an average of 94.41 ns and a standard deviation of 0.225 ns. Each processing unit takes 25 clock cycles to complete the computation for one neuron.

We decided to compare our results with a MATLAB implementation, because researchers in neuroscience almost exclusively utilize MATLAB for their data processing. A Pentium M 1.6 GHz computer running MATLAB 7 is compared with the Cyclone II FPGA running our algorithm. The FPGA time to calculate a prediction is obtained by multiplying the number of total cycles (25 cycles) by the clock periods shown in Table 1. The overall performance improvement for 32-bit single precision calculations for the whiskers is shown in Table 2.

As is evident from the results, the FPGA-based solution offers much better performance. In the current work, we only simulate one neuron for each whisker. However, future work may require a much larger number of neurons simulated for each whisker. From Fig. 6 (a), it can be seen that the MATLAB solution will not be able to support calculations for more than 50,000 neurons and still meet the sampling frequency requirement of 25 Hz.

Table 1. Resource Usage and Performance for Calculations.

| Neurons	Logic Usage			Total Memory Bits	Clock/ Cycle Time
	Total Logic Elements	Total Combinational Functions	Dedicated Logic Registers		
5	11,488 / 33,216 (35%)	11,111 / 33,216 (33%)	5,394 / 33,216 (16%)	536 / 483,840 (<1%)	94.36 MHz (10.598 ns)
10	22,964 / 33,216 (69%)	22,211 / 33,216 (67%)	5,394 / 33,216 (32%)	1,072 / 483,840 (<1%)	94.92 MHz (10.535 ns)
14	31,547 / 33,216 (95%)	30,508 / 33,216 (92%)	14,693 / 33,216 (44%)	1,459 / 483,840 (<1%)	94.32 MHz (10.602 ns)
25	31,547 / 33,216 (95%)	30,508 / 33,216 (92%)	14,693 / 33,216 (44%)	1,459 / 483,840 (<1%)	94.32 MHz (10.602 ns)
100	31,547 / 33,216 (95%)	30,508 / 33,216 (92%)	14,693 / 33,216 (44%)	1,459 / 483,840 (<1%)	94.32 MHz (10.602 ns)
500	31,547 / 33,216 (95%)	30,508 / 33,216 (92%)	14,693 / 33,216 (44%)	1,459 / 483,840 (<1%)	94.32 MHz (10.602 ns)
50K	31,547 / 33,216 (95%)	30,508 / 33,216 (92%)	14,693 / 33,216 (44%)	1,459 / 483,840 (<1%)	94.32 MHz (10.602 ns)
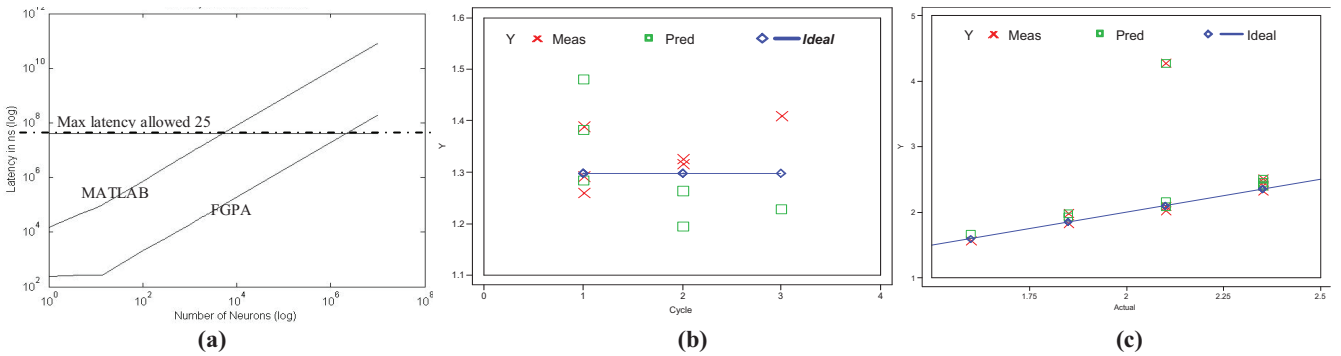
Fig. 6. (a) Latency vs. number of neurons. (b) Flat wall scenario. (c) 20-Degree angled wall scenario.

Table 2. MATLAB vs. FPGA calculation times.

Number of Neurons	MATLAB Calculation Time	FPGA Calculation	Speedup
5	49.1 μs	265 ns	185
10	75.1 μs	263 ns	286
14	102.1 μs	265 ns	386
25	186.3 μs	530 ns	368
100	706.0 μs	2.12 μs	333
500	3.350 ms	9.54 μs	351
50K	0.394 s	.947 ms	416

We have demonstrated that our FPGA implementation has made successful calculations using real data generated with the whisker matrix in shown in Fig. 4.

4.2. Successful Slope Prediction Using the Neural Circuit Simulator

Using our FPGA-based neural circuit simulator, we compared actual radial distances with measured radial distances and with predicted radial distances for a 1x5 whisker array. We considered two different test cases—a flat wall and a 20-degree angled wall.

For the flat wall scenario, the 5 whiskers touched a smooth wall perpendicularly at a fixed distance of 1.29 inches and the whiskers were moved relative to the wall to generate the data. As for the 20-degree angled wall, and the whiskers were moved at a 20-degree angle relative to the wall to generate the data.

4.2.1 Predicting Slope Using the 1x5 Array

Fig. 6 (b) shows a comparison between the actual values, measured values, and the predicted values for a flat wall.

The percent error of the prediction is 6.31%. Fig. 6 (c) shows a comparison between the actual values, measured values, and predicted values for a 20-degree angled wall with a percent error of 13.7%.

This high error is due to outliers in the data measurements and has to do with the calibration of the whiskers in the whisker matrix. If the outliers are eliminated, the percent error reduces to 3.84% for the 20-degree sloped wall. Outliers in the data measurements caused by malfunctioning whiskers result in some significant differences between measured and predicted values. However, with a larger number of whiskers, these outliers can be easily filtered out, and the accuracy of the system can be improved. Our current implementation of the physical system is limited to a single row of 1X5 whiskers. In the next section, we present experiments based on simulated data for a higher number of whiskers.

4.2.2 Simulated Data

For a 5x5 whisker matrix for the ideal suface shown in Fig. 7 (a). Fig. 7 (b) shows the surface after Gaussian noise was added to simulate the data from the robotic whisker matrix. Fig. 7 (c) shows the predicted surface using our FPGA-based simulator. Although the prediction is clearly noisier than the ideal surface, the figure demonstrates the ability of the the FPGA-system to reconstruct the 3D surface from a non-ideal input.
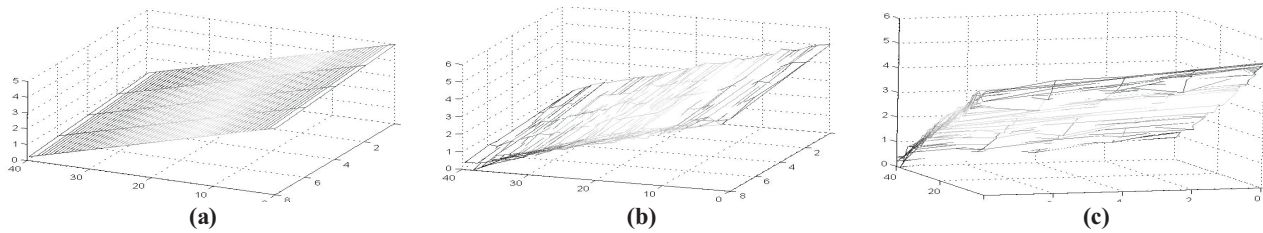
Fig. 7. (a) Simulated ideal angled wall (b) Simulated angled wall measured by whisker matrix (c) Predicted angled wall.

5. RELATED WORK

Over the years, there has been an increasing interest in biologically inspired FPGA applications. Ghani et al [14] developed an area efficient multiplier-less hardware architecture for implementing the integrate-and-fire spiking neural networks model. C. Torres-Huitzil [15] developed an area-efficient implementation of the pulse-mode neuron on a FPGA. This optimized neuron model is able to conserve FPGA resources without sacrificing desired model characteristics. A particularly related work is by M.J. Pearson et al. [16]. They developed a FPGA-based system to implement spiking neural networks of the rat brainstem. However, their work focused primarily on modeling the directionally sensitive properties of the mechanoreceptors in the follicle, and generating realistic models of spiking neurons. This work targets a specific model. Our work is different from theirs in that our main goal is to develop a FPGA-based platform that allows neuroscientists to evaluate different candidate neural algorithms. Our work contains a complete system for neuroscientists to gather and process the data and can be adapted to test possible hypotheses for neural computation.

6. CONCLUSIONS

In conclusion, we have demonstrated that FPGAs can be used to construct efficient models of massively-parallel neural circuits, without requiring the level of detail associated with spiking neurons. It has to be noted that in our pilot experiment, we designed a simplified prediction model, which can be implemented by both a PC-based system and a FPGA-based system. Specifically, the present manuscript has addressed only a much reduced model of the rat whisker system, in which the rat moves in a single direction relative to a fixed wall. When extended across multiple spatial and temporal scales, with the rat approaching an arbitrarily-shaped wall from an arbitrary direction, the proposed algorithm will require a highly parallel and efficient system such as FPGA.

7. ACKNOWLEDGEMENTS

8. REFERENCES

[1] Martinez-Conde, S., et al., "Microsaccades counteract fading during fixation." Neuron, 2006. **49**: p. 297–305.

[2] Klatzky, R.L., et al. Identifying objects by touch: An "expert system". Perception and & Psychophysics, 1985, 37(4), 299-302.

[3] Welker, W.I. "Analysis of sniffing of the albino rat". Behaviour, 1964. **22**: p. 223-244.

[4] Carvell, G.E. and D.J. Simons, "Biometric analyses of vibrissal tactile discrimination in the rat". J. Neurosci., 1990.

[5] Ebara, S., et al., "Similarities and differences in the innervation of mystacial vibrissal follicle-sinus complexes in the rat and cat: A confocal microscopic study". Journal of Comparative Neurology, 2002. **449**(2): p. 103-119.

[6] Birdwell, J.A. et al., "Biomechanical models for radial distance determination by the rat vibrissal system". Journal of Neurophysiology, 2007. **98**(4): p. 2439-2455.

[7] Shoykhet, M. et al., "Coding of deflection velocity and amplitude by whisker primary afferent neurons: Implications for higher level processing". Somatosensory and Motor Research, 2000. **17**(2): p. 171-180.

[8] Szwed, M., K. Bagdasarian, and E. Ahissar, "Encoding of vibrissal active touch". Neuron, 2003. **40**: p. 621-630.

[9] Solomon, J.H. and M.J. Hartmann, "Robotic whiskers used to sense features". Nature, 2006. **443**(7111): p. 525-525.

[10] Zucker, E. and W.I. Welker, "Coding of somatic sensory input by vibrissae neurons in the rat's trigeminal ganglion". Brain Res, 1969. **12**: p. 138-156.

[11] Furuta, T. et al., "Inhibitory gating of vibrissal inputs in the brainstem". Journal of Neuroscience, 2008.

[12] Gopal, V. and M. Hartmann, "Using hardware models to quantify sensory data acquisition across the rat vibrissal array". The Journal of Bioinspiration and Biomimetics, 2007.

[13] *Ti data converters.* [cited March 25, 2008]; Available from: http://www.ti.com/hdr_p_dc.

[14] Ghani, A. "Area Efficient Architecture for Large Scale Implementation of Biologically Plausible Spiking Neural Networks on Reconfigurable Hardware", in International Conference on Field Programmable Logic and Applications. 2006: Madrid, Spain.

[15] Torres-Huitzil, C. "Area-Efficient Implementation of a Pulse-Mode Neuron Model", in International Conference on Field Programmable Logic and Applications. 2006: Madrid, Spain.

[16] Pearson, M.J. et al., "Implementing spiking neural networks for real-time signal-processing and control applications: A model-validated FPGA approach". IEEE Transactions on Neural Networks, 2007.