2015 28th International Conference on VLSI Design and 2015 14th International Conference on Embedded Systems

Robot navigation using neuro-electronic hybrid systems

Jude Baby George
Center for Nanosicence and Engineering
Indian Institute of Science
Bangalore, 560012
Email: jude@cense.iisc.ernet.in

Grace Mathew Abraham
Center for Nanosicence and Engineering
Indian Institute of Science
Bangalore, 560012
Email: gracema@cense.iisc.ernet.in

Bharadwaj Amrutur
Department of
Electrical Communications Engineering
Indian Institute of Science
Bangalore, 560012
Email: amrutur@ece.iisc.ernet.in

Sujit Kumar Sikdar
Molecular Biophysics Unit
Indian Institute of Science
Bangalore, 560012
Email: sks@mbu.ernet.in

***Abstract*—Neuro-electronic hybrid systems have been gaining interest of researchers as a possible architecture for computing. This aims to exploit the strengths of biological neuronal systems with their immense parallel processing and learning capabilities along with that of VLSI systems. Towards this end, we have set up a system which demonstrates the use of a live neuronal culture to solve a real world problem of controlling a robot doing the task of obstacle avoidance. We show that a neuronal culture can look at the sensor inputs to the robot and generate motor commands to allow it to explore an arena while avoiding obstacles.**

I. Introduction

The computing capabilities of neuronal systems have continued to fascinate researchers for a long time. Several problem solving strategies have been developed over time being inspired by such systems. Artificial neural networks, fuzzy systems and Neuromorphic circuits (analog, digital) are some examples. These attempt to capture the architecture, parallel processing capabilities and learning strategies of the neuronal networks which allow such networks to solve complex problems using simple elements. These use the conventional hardware based on VLSI to implement such systems(software, fpga, custom digital circuits [1] , analog circuits [2]). There is also active research involved to develop hardware more suitable to mimic neuronal systems (for example, using phase change memory to implement synapses [3]). However, several challenges(versatility, connectivity, power) limit the scaling of such systems to the level of neuronal systems like a brain.

In this scenario, Neuro electronic hybrid systems are being explored as a possible architecture for computing. Here, it is envisioned that the capabilities of neuronal systems can be harnessed by connecting them appropriately with the real world. This opens up the possibilities to use the biological hardware as a platform for computation. Neurons, which form the basic computational units in the brain are very versatile in terms of their behavior and adaptability. This allows them to implement a wide variety of functions and gives them learning capabilities. Neuronal networks formed by them can have a high degree of connectivity which lends them to parallel processing and learning capabilities. Hybrid systems attempt to leverage these capabilities to solve real world problems.

Among the different alternatives, dissociated cultures of neurons grown on multi electrode arrays have been promising for studying computational properties of neuronal networks. [4] had shown that it is possible to interface a neuronal culture to a flight simulator. Simple robotic control have been demonstrated by various groups ([5] and [6]). Training of neuronal networks have been demonstrated by [5] and [7].

Several challenges need to be addressed before such systems can be used for real world applications. The appropriate methods to give inputs to a neuronal system and decode its outputs based on the expected task is an open problem. The methods that could be used to harness the learning capabilities of such networks is also not well understood. This requires a better understanding of the functioning of such systems.

Towards this end, we have set up a system which could be used to study various aspects of this problem. We have built a closed loop system where a robot, equipped with IR sensors to detect the obstacles is connected to a live neuronal culture. The signals from the sensors are given to the culture with appropriate encoding and the output from the culture is decoded to drive its actuators(motors). The robot is able to explore an arena while avoiding various obstacles that it encounters. This demonstrates the ability of the culture to control a real world object in real time. We have addressed the problem of input coding and output decoding for this application. Section II describes the system including the culture, the robot and their interconnection. This is followed by a discussion of neuronal control of the robot, input coding and output decoding schemes used to achieve this goal. Then the results are summarized and the scope for further studies using the system are outlined.

1063-9667/15 $31.00 © 2015 IEEE
DOI 10.1109/VLSID.2015.21

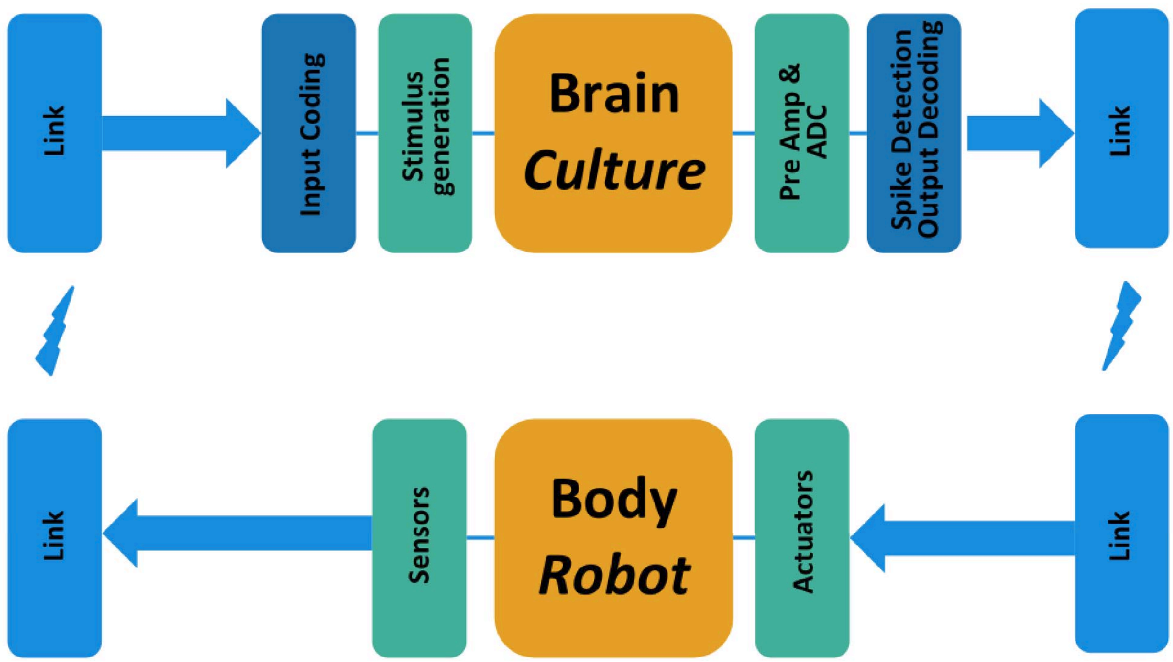

Fig. 1. System

II. System

The system used for the study comprises of three major components: the neuronal culture, the interface between the culture and other hardware components and the robot. There are also boards to enable wifi connectivity between various components of the system. A block diagram of this system is shown in figure 1.

Neurons from rats' hippocampus are grown on multi-electrode arrays(MEA) to study their activity. Active neurons generate electrical activity. Multi-electrode arrays have an array of electrodes on which neurons are grown. These electrodes pick up electrical activity of the neurons in the region around it. The array of electrodes allows us to pick up electrical activity from the neurons at various spatial locations of the network. It is also possible to apply voltage pulses to these electrodes and thus give input to the network. Thus a bidirectional interface between the neuronal network and external world can be formed using MEA's. These details are discussed below.

A. Neuronal Culture and Long term maintenance

Neuronal culture growth and maintenance was done similar to the procedures described by [8].

Dissociated neuronal cell cultures were prepared by papain digestion of whole hippocampus of 0-2 day old rat pups. 120 electrode MEA from *MultiChannel Systems©, Germany* for reusing, were soaked overnight with Tergazyme detergent (Sigma-Aldrich, USA) thoroughly rinsed with milliQ water and allowed to dry under laminar hood, sterilized with 70% ethanol and UV light. Sterilized MEA were coated with 0.05% (w/v) polyethylenimine solution in borate buffer, rinsed thoroughly with milliQ water allowed to dry and kept under laminar hood until cell seeding.

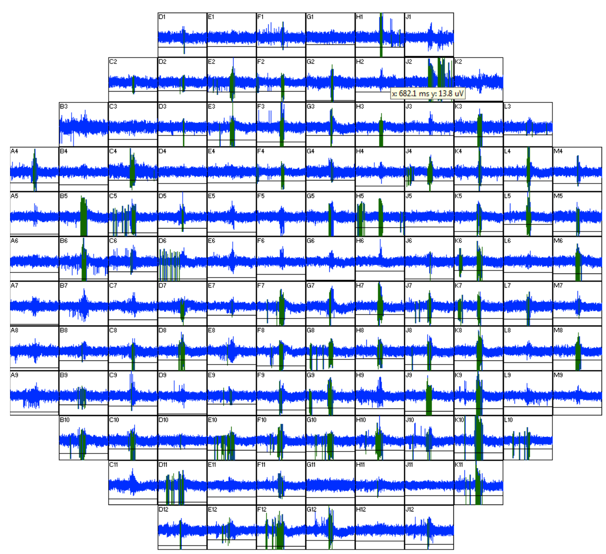

Fig. 2. Spontaneous activity recorded from neuronal culture activity after DIV25.A network wide burst is shown. Each box represents the electrical activity from an electrode in the MEA. Scale bar: y-axis, $200\mu V$; x-axis, $1000ms$.

Wistar rats were decapitated, according to approved protocols by the Animal Ethics and Welfare Committee of Indian Institute of Science, Bangalore, India, and these were followed in all the experiments. The brain was removed, chilled with ice PBS(Phosphate buffer saline), and the hippocampus was micro-dissected under sterile conditions. Papain solution was prepared according to [9] and aliquoted into 1.5 ml and stored at $-20\,^{\circ}\mathrm{C}$, and thawed at $37\,^{\circ}\mathrm{C}$ just before use. Hippocampus was digested in 2 ml papain solution for 20 min at $37\,^{\circ}\mathrm{C}$ stirring manually. The papain solution was aspirated and the pieces were triturated three times, three passes each with 1 ml of medium, using a P-1000 Pipetman. 50 000 to 2 00 000 cells were plated in a 20 μl droplet covering the 2.4x2.4 mm^2 electrode region of the 120MEA200/30iR-Ti *(Multichannel Systems©, Germany)*, forming a dense monolayer. The MEA's were coated with laminin and incubated for $\frac{1}{2}$ hr just before seeding. The dishes were flooded with 1 ml of medium after the cells had adhered to the substrate (45 min), and stored with ethylene-propylene membrane lids *(MEA-MEM membranes, ALA Scientific Instruments Inc., USA)* in a 65% RH incubator at $37\,^{\circ}\mathrm{C}$, 5% CO_2. The medium, adapted from [10], was Dulbeccos modified Eagles medium (DMEM, Irvine Scientific) with 10% FBS serum (Gibco) and was stored in the incubator to equilibrate the pH and temperature before feeding. We used antibiotic/antimycotic drugs to control contamination. Feedings consisted of 50% medium replacement twice per week. The medium was used with glial conditioning (ara-C) after 7 days.

The culture dish was placed in a separate incubator which maintained an ambient of 5% CO_2 at $37\,^{\circ}\mathrm{C}$ while doing recordings and stimulations. Figure 2 shows the activity recorded from the culture after 25 days in vitro. The cultures have been maintained with sufficient activity for about 100 days.

B. Interface Board

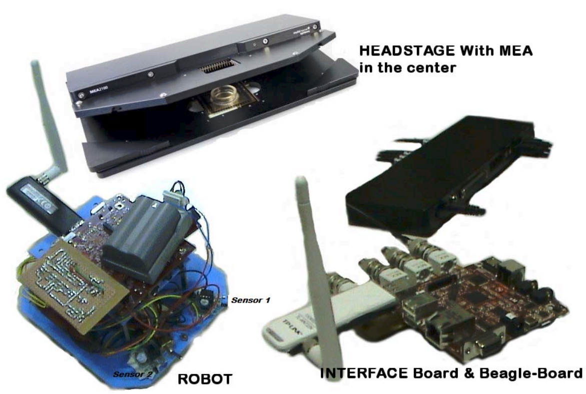

Fig. 3. Headstage, Interface board and the robot

The activity of the neuronal network is picked up by the electrodes on the MEA's recorded using MEA-2100 system from *MultiChannel Systems©, Germany*. This hardware has ADC's which allows us to record voltages at 120 electrodes simultaneously at a sampling rate of 50kHz. The hardware also has DAC's which allows us to apply voltage or current pulses at these electrodes. It has 2_{nd} digital filters which can be programmed to filter the digitized samples from each channel. It also has an on-board DSP(TMS320C6454) which allows real-time processing of the signals from the culture and apply stimulus inputs. We have connected this DSP to a beagle board(xM) using its I/O pins. The beagle board runs a Linux Angstrom distribution and allows an adhoc wifi network to be set up using a standard wifi dongle(TP-LINK TL WN722N). This allows a wireless communication channel to be set up between the robot and the DSP(and hence the culture).

C. Robot

The robot is a simple platform with two obstacle sensors to detect the obstacles and two motors to provide locomotion capabilities. It is designed such that with appropriate control inputs, it can move around in an arena while avoiding obstacles. The obstacle sensor consists of an IR led which emits IR light and a photo-diode which detects IR light. Obstacle is detected based on the amount of reflected light received by the photo-diode. The motors are driven from a $11V$ Lithum Polymer(LiPo) battery source using standard motor driver IC $L293D$. A beagle board (xM) is used to control the motor drivers. A pulse width modulation(PWM) function is implemented in software to allow the control of speed of motors. The beagle board also allows the robot to communicate with the culture over the wifi link.

D. The Closed Loop System

The beagle boards on the robot(BB1) and the one connected to the DSP(BB2) are linked to each other via an adhoc wifi network. A server program running on BB2 connects to a client program on BB1. These programs are written using C programming language. The client program reads the sensor inputs which are connected to BB1's GPIO pins and sends it to the server program. The GPIO's indicate a '1' when an obstacle is present and a '0' otherwise. Thus the two sensors together can be in 4 different states, {00,01,10,11}. The DSP then receives this sensor code and stimulates the culture in a manner pre-defined for the this code. It then waits for some time to gather the response from the culture. This is then translated to an appropriate movement command for the robot and is then sent to the client program on the robot. The movement command can be one of the following {FRONT, BACK, LEFT, RIGHT}. The client program then runs the motors accordingly. The basis and details of the encoding and decoding schemes are described in the following sections.

E. Stimulation and Spike Detection

1) Stimulation details : Stimulus pulses are applied to the culture using appropriate electrodes based on the inputs received from the sensors. The translation of sensor values to stimulus pulse timing is defined as input coding and is described in the following section. Each stimulus pulse is a biphasic voltage pulse with each phase having an amplitude of $500mV$ and duration of $500\mu s$. Such pulses have been shown to be suitable to cause network activity in the neuronal cultures by [11].

2) Spike detection : Extra cellular field potentials are recorded by the electrodes of the MEA's. To do further processing we need to detect useful events in these recordings. Occurrence and timing of action potentials are important events in neuronal processing. These appear as voltage spikes in extracellular recordings. In order to detect these spikes, the digitized signal is first filtered using a 2^{nd} order high pass butterworth filter with a cut off frequency of 500 Hz. Then the spikes are detected by setting an appropriate threshold on the voltage levels(5x the standard deviation of noise over a 10s interval) on each channel. Once the spikes are identified for each channel, further processing is done to translate this to motor commands for the robot. This is called output decoding and is described in the following section.

III. Neuronal Control

A. Basis of neuronal control

Neuronal control is implemented based on the finding that the probability of occurrence of a spike at an electrode in response to a sequence of stimulus pulses at some input electrodes depends on the electrodes that are being stimulated, the order in which they are stimulated and the timing between various pulses [12]. Consider Figure 4. This figure captures the effect of the selection, order and timing of stimulus pulses on the response of the network. Each circle represents an MEA and each dot inside it represents the location of an electrode. Each column corresponds to stimulation of two electrodes in different sequence and with different timings. The intensity of the dot indicates the probability that a spike is observed at that electrode when a pair of stimulus pulses are applied in a manner specified by the row and column labels. It can be observed that the pattern of response is dependant on the stimulus conditions. Thus it can be concluded that the network is able to distinguish between various inputs based on the electrodes that are stimulated, their order and timing. This can be used as a basis of encoding the inputs to the culture and decoding the output responses.

B. Selection of input electrodes

Each of the 120 Electrodes is stimulated one at a time. This is repeated thrice and the 8 electrodes that evoke maximum network activity are selected for further trials.

C. Input Pattern

Input pattern P with r electrodes is defined as a sequence of firing of the input electrodes with some time delay between each stimulation.

$$\mathbf{I_j} = \begin{bmatrix} t_1 \\ t_2 \\ .. \\ t_r \end{bmatrix} \tag{1}$$

where t_i indicates the time of firing for input electrode i for the j^{th} input pattern. This time is an offset from the start of presentation of the stimulus.

$$t_i = nt_d \tag{2}$$

The time of firing of each electrode w.r.t a fixed time is an integer multiple of a time delay t_d ms which can be varied from 0.5ms to 1sec in steps of $20\mu s$.

D. Output Pattern

The output pattern is a binary vector which indicates whether there was a spike at an electrode following an input stimulus pattern. It is defined as follows.

$$\mathbf{X_k^j} = \begin{bmatrix} s1_k^j \\ s2_k^j \\ .. \\ s120_k^j \end{bmatrix} \tag{3}$$

X_k^j is the output pattern for the culture for the k^{th} presentation of input pattern j . Here sM_k^j is the spike occurrence indicator for electrode M and is defined as sM_k^j = 1 if at least one spike occurs in the time window 5ms to 100ms after the j^{th} input pattern is presented to the culture k^{th} time.

E. Output Decoding

The decoder is an array of parallel perceptrons that can be trained to distinguish between each input from the rest of the inputs or a set of inputs from the remaining ones. A perceptron is a simple processing element which does a weighted sum of its inputs and generates a binary(1/0) output if the sum is greater than a threshold value. It can be described by the following expression.

$$O_k^j = \begin{cases} 1 & \mathbf{W_j} \cdot \mathbf{X_k^j} - Threshold > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Here O_k^j is defined as the output of the perceptron j with a weight vector $\mathbf{W_j}$ for the k^{th} persentation of input pattern j.

The weight vector describes a hyperplane which separates the set of outputs the perceptron is trained to decode from the rest of the outputs. These set of weights are learned using the perceptron training algorithm, the delta rule [13] [12].

F. Input Coding and Output Decoding for Robotic control

- 8 electrodes are selected as candidates for defining input patterns.
- 56 patterns are created by pairing 2 of 8 electrodes at a time with a time delay of 0.5ms. These are applied to the culture in a random order 45 times.
- 56 perceptrons are trained to decode output patterns corresponding to 56 inputs. 4 input patterns for which the perceptrons are able to decode the output patterns with least error are selected for robotic control.
- Each of the 4 input patterns is then mapped to a sensor code {00,01,10,11} and the corresponding output perceptron is mapped to an action { FRONT, LEFT, RIGHT, BACK}.
- When the system is working in a closed loop, the input pattern corresponding to the sensor code is applied to the culture. The output binary vector is formed after 100ms. Each perceptron does a weighted sum of this vector. The perceptron with the maximum result is selected as the winner(Winner Take All/WTA

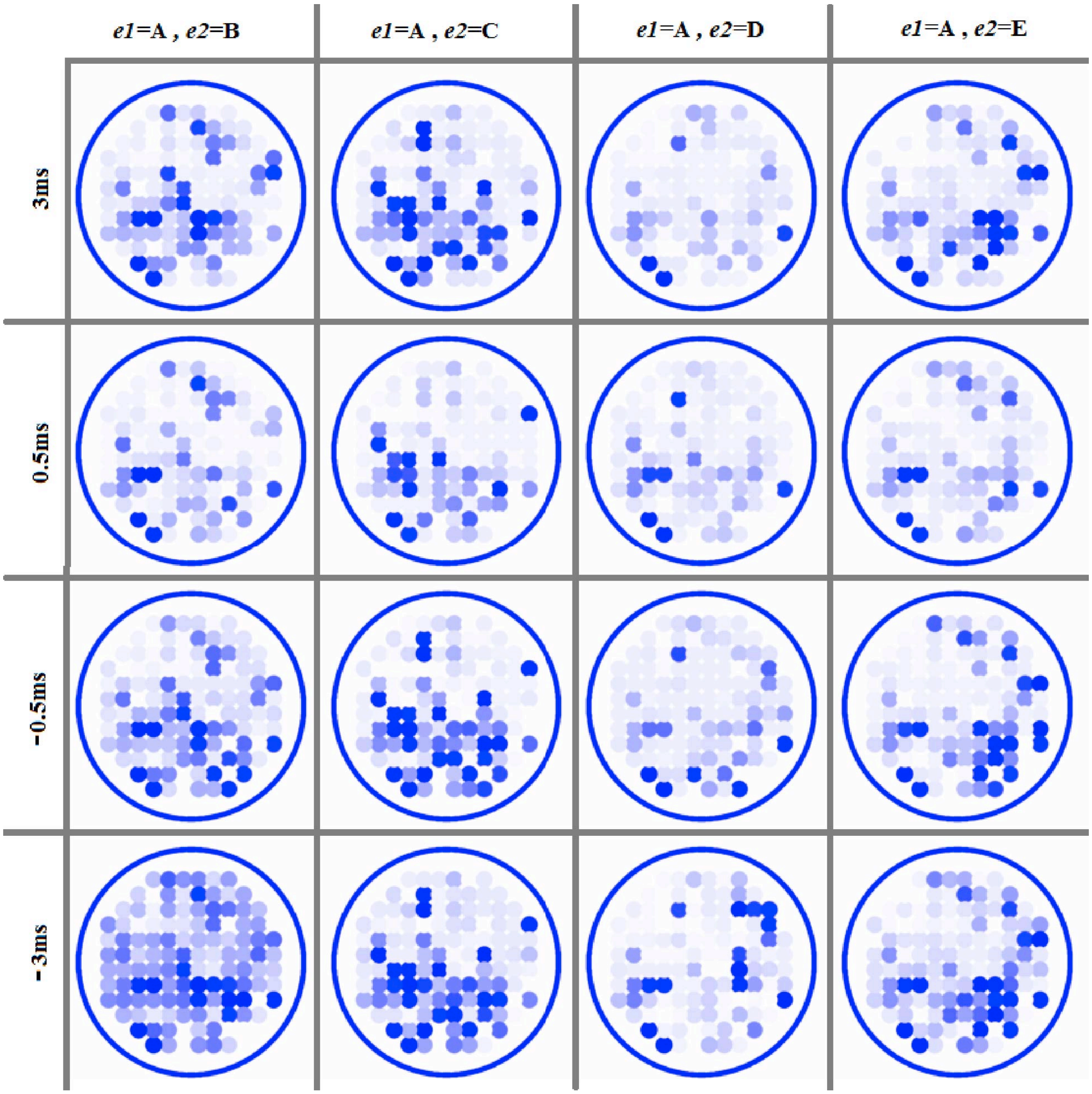

Fig. 4. Network response depends on the pair of inputs stimulated and the relative timing between them. This figure shows a few of the samples. Here, each column corresponds to an electrode pair shown in the top of the column chosen from the set of 8 electrodes identified initially for stimulation $\{A, B, C, D, E\}$. Each row corresponds to a particular order and timing between the stimulation of these electrodes as labelled on the left side of the row. Each dot corresponds to an electrode and the intensity of the color is proportional to the probability of a spike at that electrode for the stimulus pattern. Different activity patterns can be observed on the electrodes for various combination of sequences and timing. This shows that culture response is depends on selection of electrodes and order and timing of input pulses

strategy). This is to avoid a deadlock for the cases if none of the perceptrons identify the pattern or more than one perceptron detects the output code. The output action mapped to the winning perceptron is send to the robot as a movement command.

G. Details of the implementation

Various steps discussed above are implemented using custom software developed in MATLAB and C programming language. The DSP code handles the control of the hardware for acquisition and stimulation. MATLAB code that runs on a data aquisition computer performs the tasks of setting up the system with the correct parameters. The details of the various steps are discussed below.

The software for the control of DAQ hardware runs on a TMS320C6454 DSP. It is programmed using C programming language. The various modules in the software handle spike detection, forming output vectors and perceptron decoding based on programmed weights. It also has a module to configure the DAC's and play different stimulation patterns on the electrodes. These patterns have to be pre-loaded into the DSP from the DAQ computer. When the code is running after begin loaded with the required parameters, it plays the patterns on the electrodes and sends out the acquired data to the computer. If it is set up to control the robot, it would periodically gather the sensor inputs from the connected Beagle Board, stimulate the culture based on the sensor value, decode the spikes, do output decoding and send the motor commands to the robot.

The MATLAB code handles the higher level control of the protocols involved in conducting the experiments. For every experiment, it forms the stimulation patterns to be applied and loads them into the DSP. It then receives the data-streams (which has raw data as recorded from the electrodes as well as the stimulus information) and saves it to the hard-drive for later processing. It implements the modules for threshold estimation on the raw data and also has the algorithm for determining the appropriate weight vectors for the decoders.

The code on Beagle Boards(BB) connected to the DSP (BB2) as well as running the robot(BB1) is also developed in C programming language. The BB2 exchanges data with the DSP using a simple bit-banging protocol defined for exchange of data between the two. The two BB's communicate with each other over an adhoc wifi link using a TCP protocol. The BB1 controls the robot using its GPIO's.

The sensor values are read and the motor directions are updated every 250 ms. Here, 3 ms is required to exchange data between the two BB's over the wifi-link. Once the input coded pattern is applied to the culture, the response from the culture is gathered for next 100 ms. The perceptron decoders take about $1\mu s$ to do output decoding.

IV. RESULT

With the closed loop system in action, the robot was able to navigate exploring the arena while avoiding the obstacles. The accuracy in generating the correct command inputs for the robot was greater than 98%. In each trial, we run the robot for 10min over which the accuracy was maintained. The following clip shows the robot exploring an arena while avoiding the obstacles (http://youtu.be/j4asE9l7OsI).

However, it is observed that culture outputs for a fixed input pattern changes over time. So we have to determine the parameters for the output decoders(the weight vectors for the perceptrons)every time we start a run of the robot.

V. CONCLUSION

We have demonstrated that with appropriate input coding and output decoding schemes the neuronal network is able to interact with real world objects in real time with sufficient accuracy for control. The experiment shows that input patterns can be defined such that the network is able to distinguish them. The network is able to produce consistent outputs for the input patterns which makes the control of the robot possible. Simple decoding schemes allow real time implementation of the system.

There are several challenges that need to be addressed before such systems can be put to use for more complicated tasks. We need to identify input coding schemes that map the inputs more naturally to the stimulus pulses and outputs to desired actions. We also need to find out methods such that the network can learn to do the task without pre-assignment of input and output patterns. This requires a better understanding of network architecture. We hope that our current system would allow us to study and address these problems.

ACKNOWLEDGMENT

The authors would like to thank Nirosha Sigarum and Tanumoy Saha for their help in the construction of the robot.This work was supported by Department of Electronics and Information Technology, Ministry of Communications and Information Technology, Government of India.

REFERENCES

[1] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Hfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. SAGHI, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, no. 73, 2011. [Online]. Available: http://www.frontiersin.org/

[2] N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using event-driven qdi circuits," in *Asynchronous Circuits and Systems (ASYNC), 2012 18th IEEE International Symposium on*. IEEE, 2012, pp. 25–32.

[3] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *Electron Devices Meeting (IEDM), 2011 IEEE International*. IEEE, 2011, pp. 4–4.

[4] T. B. Demarse and K. P. Dockendorf, "Adaptive Flight Control With Living Neuronal Networks on Microelectrode Arrays." *IEEE Joint Conference on Neural Networks*, 2005.

[5] D. J. Bakkum, Z. C. Chao, and S. M. Potter, "Spatio-temporal electrical stimuli shape behavior of an embodied cortical network in a goal-directed learning task," *Journal of neural engineering*, vol. 5, no. 3, p. 310, 2008.

[6] A. Novellino, P. Angelo, L. Cozzi, M. Chiappalone, V. Sanguineti, and S. Martinoia, "Connecting neurons to a mobile robot: An in vitro bidirectional neural interface," *Computational Intelligence and Neuroscience*, 2007.

[7] M. Ruaro, P. Bonifazi, and V. Torre, "Toward the neurocomputer: image processing and pattern recognition with neuronal cultures," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 3, pp. 371–383, 2005.

[8] S. M. Potter and T. B. Demarse, "A new approach to neural cell culture for long-term studies," *Journal of Neuroscience Methods*, vol. 110, pp. 17–24, 2001.

[9] M. Segal, R. Baughman, K. Jones, and J. Huettner, "Mass cultures and microislands of neurons from postnatal rat brain," *Culturing Nerve Cells*, vol. 2, p. 30938, 1998.

[10] Y. Jimbo, T. Tateno, and H. P. . C. Robinson, "Simultaneous Induction of Pathway-Specific Potentiation and Depression in Networks of Cortical Neurons," *Biophys. J.*, vol. 76, no. 2, pp. 670–678, Feb. 1999. [Online]. Available: http://www.biophysj.org/cgi/content/abstract/76/2/670

[11] D. A. Wagenaar, J. Pine, and S. M. Potter, "Effective parameters for stimulation of dissociated cultures using multi-electrode arrays," *Journal of Neuroscience Methods*, vol. 138, no. 12, pp. 27 – 37, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165027004001141

[12] J. B. George, G. M. Abraham, K. Singh, S. M. Ankolekar, B. Amrutur, and S. K. Sikdar, "Input coding for neuro-electronics hybrid systems," *BioSystems(to be published)*, 2014.

[13] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," in *1960 IRE WESCON Convention Record, Part 4*. New York: IRE, 1960, pp. 96–104. [Online]. Available: http://isl-www.stanford.edu/\~{}widrow/papers/c1960adaptiveswitching.pdf