Implementing a bio-inspired navigation system for a robot using spiking neural network

Jayesh Choudhary
Department of Electrical Engineering
IIT Bombay
Mumbai, India
jayeshchoudhary2000@gmail.com

Preetam Pinnada
Department of Electrical Engineering
IIT Bombay
Mumbai, India
preetamp2000@gmail.com

Srisht Fateh Singh
Department of Electrical Engineering
IIT Bombay
Mumbai, India
fateh321@gmail.com

***Abstract*—This paper discusses and presents previous work done in navigation of a robot implemented using a spiking neural network. The navigation system consists of bio-mimetic Head Direction (HD) cells which calculate the direction of the robot. This is done by integration of angular velocity that is realized by a continuous attractor network modeling the interaction between the lateral mammillary nucleus (LMN) and the dorsal tegmental nucleus (DTN). We then present a new method to identify the position of the robot in space. Specifically, a method to distinguish between 4 corners of a square based firing patterns of 4 distinct neurons.**

Keywords—Navigation, Head Direction cells, spiking neural network

I. Introduction

Sense of direction is crucial for both biological and artificial systems involving navigation. Extracellular recordings from freely-moving rats show the presence of head-direction (HD) cells, limbic neurons whose firing activity is correlated with the current direction of the head of the animal [1]. Not only to infer the location, but also for proper functioning of other systems responsible for the allocentric location representation of the hippocampal place cell. .An HD cell has a unique preferred direction for which it discharges maximally. The directional coding of the HD cells is independent of the animal location, and provides, therefore, an allocentric directional representation.

The properties of the HD cell system resemble those of a compass, but the allocentric coding of HD cells does not depend on geomagnetic fields. Rather, the preferred directions are anchored to visual cues in the environment. HD cells have been observed in several regions centered on the brain limbic system, in particular the postsubiculum (PSC), the anterodorsal thalamic nucleus (ADN), and the lateral mammillary nucleus (LMN) [1]. Inertial self-motion signals are likely to converge onto the HD system via the dorsal tegmental nucleus (DTN) [2] which projects inhibitory connections directly to LMN.

The first part of this work focuses on implementing the continuous attractor network which models the interaction between the LMN and DTN. This is done along with the integration of angular velocity which is also implemented in the overall SNN model [4].

The second part of the paper presents a new idea which aims at providing a navigation system consisting of the spatial coordinates of the robot. The inputs to this robot are supposed to come from an accelerometer as well as a gyroscope. Thus, linear acceleration and angular velocity are taken as inputs. These inputs are chosen based on the C.elegans worm robot implemented by [4]. Other than this for calibration purposes from time to time, allothetic information (visual input) can be used to modify the robot's dynamics.

In this work, [4] solved the contour tracking of the C.elegans worm inspired by the chemotaxis network. This solution was based on an end-to-end spiking neural network. The proposed navigation system is done to complement the C.elegans model by providing an efficient and compact navigation system implemented using a spiking neural network.

II. Methods

A. Implementation of Head Direction cells

The HD cells are modelled by means of a modular artificial neural-network as shown in Fig.1. The architecture consists of an attractor-integrator network composed by a population LMN of excitatory directional units and two populations DTN_{cw} and DTN_{ccw} of inhibitory directional units each. The HD units within each network have evenly distributed preferred directions.

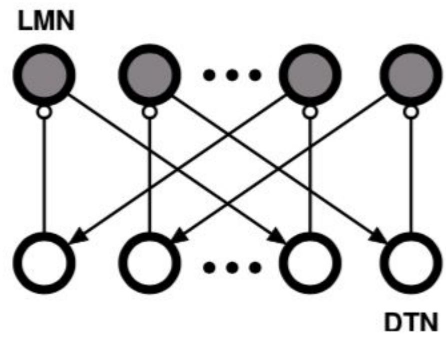


Fig. 1 Neural network between LMN and DTN

LMN projects excitatory connections to DTN. The weight w_{kn} of these connections is such that a neuron n in LMN with a preferred direction θ_n projects to a cell k in DTN with a preferred direction θ_k according to a Gaussian weight distribution:

$$w_{kn} = W_e^{max}.exp(\frac{-(\theta_n-\theta_k+\pi+\Delta)^2}{2\sigma^2}).$$

DTN projects inhibitory connections to LMN. The weight w_{nk} of these connections is distributed according to:

$$w_{nk} = W_i^{max}.exp(\frac{-(\theta_n-\theta_k)^2}{2\sigma^2} .$$

Here, the constants such as Δ and σ are defined for each cluster.

Also, there are lateral connections between DTN_{cw} and DTN_{ccw} to induce unbalanced cell activity between DTN_{cw} and DTN_{ccw}.

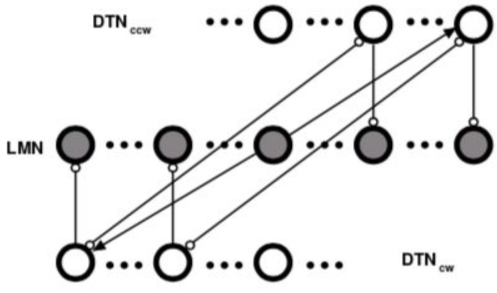


Fig. 2 Lateral connections within DTN

These connections are distributed such that a neuron i with a preferred direction θ_i projects to a cell l with a preferred direction θ_l according to :

$$w_{li} = W_{li}^{max}.exp(\frac{-(\theta_n-\theta_k+\Delta)^2}{2\sigma^2}) .$$

The neuron models used are simple Leaky Integrate and Fire neurons. The angular velocity is given as a current to the DTN cells.

For the implementation purposes, we had 1000 neurons in LMN and DTN each. In LMN, each neuron has a preferred direction which is taken from a uniform distribution. As for DTN, the first 500 corresponds to the counterclockwise direction and the latter 500 to clockwise direction. The weights for the network is as described above. The value used are as follow: For the excitatory connections, W_e^{max} = 63.3, σ_e = 67.2, Δ_e = 50 degrees if the postsynaptic neuron $\in DTN_{cw}$ and -50 degrees if the postsynaptic neuron $\in DTN_{ccw}$. For the inhibitory connections, W_i^{max} = 59.5, σ_i = 147.3. For the lateral connections, connections, W_l^{max} = 25.6, σ_e = 35.1, Δ_e = -30 degrees if the postsynaptic neuron $\in DTN_{cw}$ and 30 degrees if the postsynaptic neuron $\in DTN_{ccw}$.

Neuron parameters are as follows: Resting membrane potential V_l = -70 mV, threshold voltage V_t = -50 mV and reset potential V_r = -55 mV. C = 0.5 μF for excitatory cells, and C = 0.25 μF for inhibitory cells. Leak inductance g_m = 25 nS for excitatory cells and g_m = 20 nS for inhibitory cells. Refractory period is taken as 2 ms for excitatory and 1 ms for inhibitory neurons. The total current comprises the synaptic current I(t) and background current given by I_e = $rnd_{0,1}$. K(t) where K(t) is 1200 for excitatory neurons and K(t) = 100 + 0.44* $\omega(t)$ for inhibitory neurons where $\omega(t)$ is the angular velocity.

The dynamics of LIF neuron is given by:

$$C.\frac{dV}{dt} =- g(V(t) - V_l) + I(t) + I_e(t)$$

The synaptic current I(t) for a neuron is given by: I(t) = 0.01s(t) where the dynamics of s(t) is given by:

$s(t+\Delta t) = 0.99 * s(t) + \sum_{k=0}^{N} \delta(t-t_k).w_{nk}$ where Δt = 1 ms and $\sum_{k=0}^{N} \delta(t-t_k).w_{nk}$ is the weighted sum over the spikes emitted by the presynaptic neurons k and w_{nk} is the strength of the connection from a presynaptic unit k to the postsynaptic neuron n.

III. Intuitive understanding of the paper

The attractor network works like a converging network. It converges the firing of LMN neurons. Let us understand this using an example: suppose there is no input (ω=0), then the LMN neurons (which are excitatory) receive random current inputs. These LMN neurons (each with a prefered direction) excite the DTN neurons (which are inhibitory). This excitation is according to the gaussian profile defined above. The DTN neurons inhibit the LMN neurons again according to the above gaussian profile. Thus, the LMN neurons via DTN are giving negative feedback to itself. The LMN neurons with sufficient current survive while the rest dies down i.e. a self-sustained state in which only a subpopulation of cells with similar preferred directions discharges tonically . Fig (3) shows this transient behaviour of the LMN neurons.

Once this is established, if some current (which represents angular velocity) is given to DTN neurons, the DTN_{cw} neurons inhibit DTN_{ccw} according to the gaussian profile which in turn inhibit DTN_{cw}. The peak firing starts to move constantly which changes the peak firing of LMN proportional to ω. This phenomenon can be observed in the simulated fig(4). Thus, the direction of the bot is encoded in the firing rate of LMN neurons.

IV. Key challenges and questions

There are a couple of challenges that one can observe in this approach. One challenge is the reconfiguration using visual stimuli to tackle noise. According to the biological inferences [1], the brain constantly receives visual stimulus of the direction in order to calibrate the direction. This is done to deal with the noise interference. Thus on its own, this system is not strongly robust to noise.

Also, the integration of angular velocity is also not exact. This can lead to wrong results if the range of angular velocity is high.

Adding to above, this model is very heavy to run on a real-time robot. The paper [3] drops some weights whose magnitude is less than a threshold. But when we tried to implement such weights, the code runs somewhat faster but at the expense of output accuracy. Thus an ideal navigation system for a real-time robot should be concise, fast and robust to noise.

V. Replication of published results and challenges faced

The paper was implemented on MATLAB. The paper provides germane details when it comes to implementation. The structure of the network has been described in the methods section. The following important results were replicated:

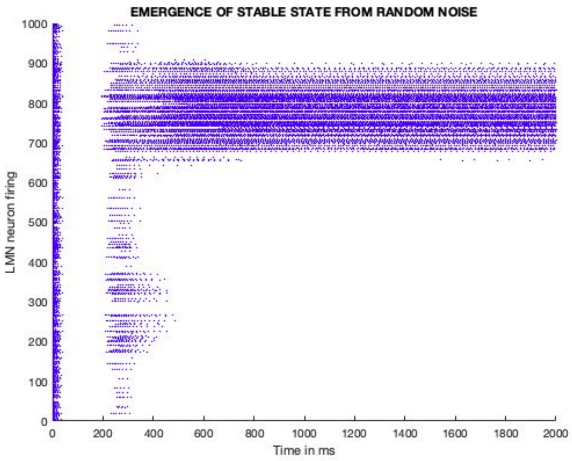


Fig. 3 Raster plot for LMN neurons when no angular velocity is provided

A. Convergence of the angle when there is no input velocity

Fig(3) shows the spike raster plot for the LMN neurons when no input angular velocity is applied. This figure has also been discussed in the understanding section. Since the input current to the LMN neurons is random, only the neighbouring group of neurons which received dominated input currents survived. In this case, only the attractor network constitutes the underpinning.

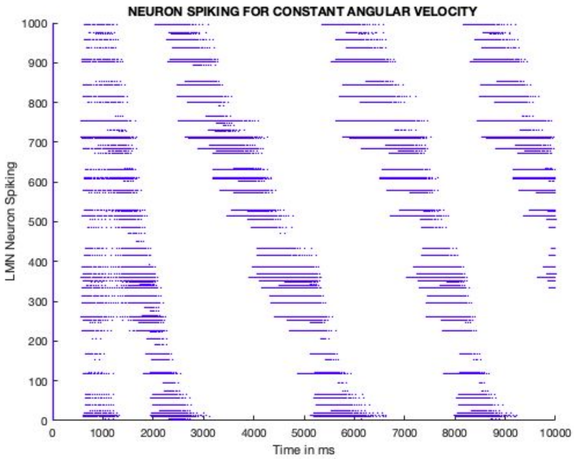


Fig. 4 Raster plot for LMN neurons when angular velocity is provided

B. Output when input angular velocity is given:

Fig(4) shows the spike raster plot of LMN neurons when it receives input angular velocity as well. We can see the shift in the firing of the neuron as time increases. This is due to the fact that as time progresses, due to angular velocity, the net angle changes. So the neurons with that particular angle as preferred angle along with some neighbouring neurons fire.

C. Challenges faced

As described before, lucid implementation details were provided in the paper. However there were two challenges that we came across: The numerical value for weights associated with input current were provided. However, the unit of input current was nowhere mentioned in the paper. Thus, we iterated over the standard units of currents and finally went with nA. Because of this, certain values of the constants given in the paper didn't yield results. But we had to tweak the values until we got satisfactory results.

Because of its size and complexity, we were unable to run the code with all gaussian weights taken non-zero. Thus, we had to ignore the weights which were below a certain threshold.

VI. Our proposed method

The crux of every navigation system is in its ability to determine the position in space. In our method we divide the 2D space into a square grid, and identify at which of these lattice points is the bot located at any given instance. For our implementation purposes, we limited these positions in space to four corners of a square. This idea can be scaled using symmetry to encompass a larger 2D space.

A. Network Architecture

The neural network contains the following 9 neurons -

1. 4 position neurons (viz. A,B,C,D) each corresponding to one of four corners of a square. These neurons fire in a mutually exclusive manner. At any instance the position of the robot is inferred from the neuron that's firing.
2. 4 direction neurons (viz. 0, 90, 180, 270) each indicating the direction of the robot w.r.t. a predetermined baseline. These neurons again fire in a mutually exclusive manner. At any instance the direction of the robot is inferred from the neuron that's firing.
3. 1 motor neurons (M). M fires when the robot makes a move. A move is defined as a transition from one corner of a square to another. Movement only along the edges of the square are permitted.

All the neurons are modelled as LIF neurons and all synapses in the network are of the same type. The architecture is described in Fig. 5. The motor neuron excites every position neuron. Since one can arrive at a particular corner of a square only in two directions, the angle neurons excite position neurons accordingly. For example, one can reach B from A in 0^0 direction and from C in 270^0 direction, therefore only 0^0 and 270^0 direction neurons excite the position neuron B. If this is to be extended to a larger space with greater number of lattice points, in such a case just like the motor neuron, every direction neuron would excite every other position neuron.

B. Neuronal Dynamics

Neuronal dynamics during navigation from a corner to the adjacent would play out as follows. Without any loss of generality, consider the robot starting at corner A and facing corner B, i.e. neurons A & 0^0 are firing at the moment. A change in position from A to B is seen as (A AND 0^0 AND M) together charging neuron B. When the B fires, it indicates the change in robots position. The firing of neuron B is sustained through a self-excitation indicated in amber colored arrows in Fig. 5. When B is being charged, simultaneously (A AND 0^0 AND M) inhibits the neuron A.

The notion of "AND" is taken from [4]. This way mutually exclusiveness is achieved in position neuron firing patterns.

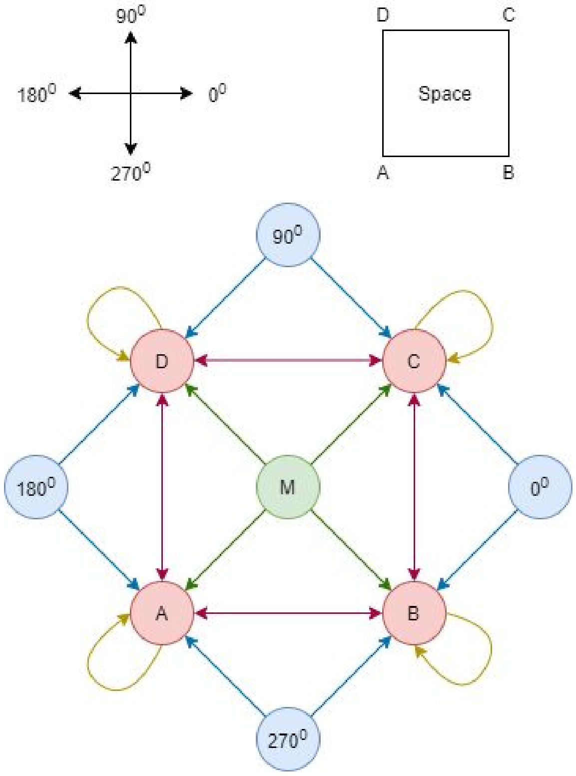


Fig. 5 Network architecture. Red - Position neurons; Green - Motor neurons ; and Blue - Direction neurons

C. *Simulation*

In our experiment, we ran the bot around the square once, starting from A viz. the path is A -> B -> C-> D-> A -> B. Commands to the bot are given as step currents to motor neurons. Direction neurons with desired functional capabilities are assumed to be given. This assumption is valid because their implementation can be done in a similar fashion with angular motion neurons in place. At the start of the experiment, the external stimulus is given to neuron A for 10 ms to bring the robot to life, and then onwards the bot is on its own. Total duration of the run is 0.5s. Refractory period of every neuron is 2 ms. Time constant of the nerve membrane is 15 ms and the synaptic time constant is 3.75 ms. Weights of all synapses are taken to be 15×10^3.

The results of this experiment are shown in Fig. 6 and in Fig. 7. It is evident from the firing patterns of position neurons A(red), B(green), C(blue), D(yellow) in these figures, that the bot is able to correctly identify it's position as it visits the four corners of a square.

D. *Challenges*

A key challenge in this method is control the minimum membrane potential while inhibiting a position neuron. In our implementation it was capped at -0.1V. Secondly, implementing mutual exclusiveness was tricky. We had to tweak membrane time constants to achieve it.

E. *Future work*

The future work intends to cover the entire surface area with multiple lattice points. In addition, instead of mandating mutual exclusiveness of position neurons, inferring position from neuron with highest firing frequency at any instant is more desirable and biologically plausible. Which could be achieved using a winner take all circuit.

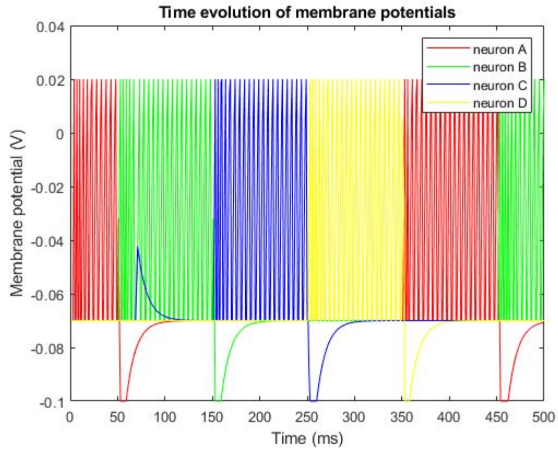
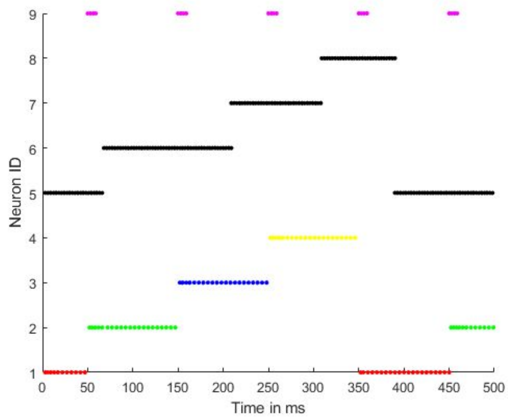


Fig. 6 Time evolution of membrane potentials of position neurons A (red),



B (green), C (blue), D(yellow)

Fig. 7 Firing pattern of all the neurons in the network. Neuron IDs - **1**:A, **2**:B, **3**:C, **4**:D, **5**:0^0, **6**:90^0, **7**:180^0, **8**:270^0, **9**:M

References

[1] J. S. Taube, Head direction cells and the neurophysiological basis for a sense of direction, Progress in Neurobiology, vol.55, issue.3, pp.225-256, 1998.

[2] Bassett, J P, and J S Taube. "Neural correlates for angular head velocity in the rat dorsal tegmental nucleus." *The Journal of neuroscience : the official journal of the Society for Neuroscience* vol. 21,15 (2001): 5740-51..

[3] Thomas Degris, Loïc Lachèze, Christian Boucheny, Angelo Arleo. A Spiking Neuron Model of Head-Direction Cells for Robot Orientation. *8th International Conference on Simulation of Adaptive Behavior*, Jul 2004, Los Angeles, CA, United States.

[4] Shukla S., Dutta S., Ganguly U. (2018) Design of Spiking Rate Coded Logic Gates for C. elegans Inspired Contour Tracking. In: Kůrková V., Manolopoulos Y., Hammer B., Iliadis L., Maglogiannis I. (eds) Artificial Neural Networks and Machine Learning – ICANN 2018. ICANN 2018. Lecture Notes in Computer Science, vol 11139. Springer, Cham