

Worminator: A platform to enable bio-inspired (*C. elegans*) robotics

Raphael Norman-Tenazas Jordan K. Matelsky Kapil D. Katyal Erik C. Johnson William R. Gray-Roncal
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road; Laurel, Maryland 20723

Abstract

Recent years have seen a renaissance in artificial intelligence (AI) technology and its applications, including robotics. Many of these solutions focus on solving a particular problem in a particular domain or environment. Creating robust and generalizable AI solutions is an area of great interest with applications to many different problem spaces. Biological organisms and biological nervous systems serve as an existence proof that such a generalized intelligence solution is possible. We develop and explore a framework for the simulation of biological networks and extend these simulations to a real-world robotic platform. We focus our initial exploration on a simple, well-defined and highly stereotyped biological neural network (i.e., connectome) derived from the *Caenorhabditis elegans* nematode. We implement a reference anatomical connectome on a robotic platform, then perturb the network to study the influence of the network parameters on output behavior. Target perturbations can be derived from neuroscience, robotics, or machine learning domains. This platform is useful for exploring the relationship between learning and behavior for biological organisms and robots. To foster further discovery, we share an open source, easy to use framework with visualization and simulation capabilities and provide an interface to a TurtleBot using ROS.

Keywords: connectomics; robotics; *C. elegans*

Introduction

There has been great progress in the previous decades developing Artificial Intelligence techniques for robotics applications, but there are still critical limitations in terms of generalizability, learning, and robustness. One approach to overcoming these limitations is the study of biological systems (Lake, Ullman, Tenenbaum, & Gershman, 2017). Understanding the relationship between behavior and information processing in neural circuits is one of the core challenges in neuroscience, and may lead to neurally-inspired algorithms for robotic applications. Even organisms with small nervous systems, such as the nematode *C. elegans* (White, Southgate, Thomson, & Brenner, 1986), show robust behaviors such as avoidance and exploration. Simulating neural circuit activity during closed-loop behavioral tasks provides a means to study the computational principles underlying behavior, which might then be applied to robotics.

Simulating a complete nervous system requires that system's connectome, the structural or functional "wiring diagram." A connectome is conventionally represented as a

graph, a mathematical object with vertices, directed edges, and attributes. By generating connectomes from electron microscopy, anatomical graphs are derived where vertices represent individual neurons and edges are the directed synaptic connections (Lichtman & Sanes, 2008). Simulation of neural circuits also requires an understanding of neuronal dynamics and neuromodulation (Bargmann & Marder, 2013), as different neural time dynamics can generate different responses from the same underlying connectome.

As a model organism, *C. elegans* offers significant promise for closed-loop simulation. The *C. elegans* connectome is well-characterized; the hermaphrodite nematode contains approximately 300 neurons and 8000 chemical and electrical synapses (White et al., 1986). While the neuronal dynamics and role of neuromodulation are not fully understood, neural circuits have been proposed for many behaviors such as touch-withdrawal (Chalfie et al., 1985) and exploration of environments (Gray, Hill, & Bargmann, 2005). New patch clamping results also give insight into the dynamics of individual neurons (Lockery & Goodman, 2009).

Due to the promise of *C. elegans* for closed-loop simulation of behavior, the OpenWorm project developed detailed simulation of all the cells in the worm relevant to sensing and motor activity (Szigeti et al., 2014), using models of channel dynamics. Other recent simulation efforts compared the worm's tap-withdraw response to optimal control policies (Lechner, Grosu, & Hasani, 2017) and proposed a Simulink model of the worm using single-compartmental models (Hasani, Beder, Fuchs, Lung, & Grosu, 2017).

An alternative approach is to implement dynamic simulations of the *C. elegans* connectome on physical, robotic platforms. These studies apply the neural computational principles underlying *C. elegans* behavior to robot control, and enable comparison between biologically-inspired algorithms and traditional approaches. One such robotic platform implemented a six-neuron simulation of *C. elegans* chemotaxis (Morse, Lockery, & Ferrée, 1998), but did not simulate the entire worm nervous system. More recently, a robotic system simulating the entire *C. elegans* connectome was demonstrated (Busbice, 2014). This robot exhibited the tap-withdrawal behavior of the nematode, but had technical limitations preventing easy modification of the connectome or introduction of new models of neural dynamics.

Methods

As a next step in exploring neurally-inspired robotic control and the relationship between neural computation and behavior in *C. elegans*, we present a novel method to translate an arbitrary graph (either biologically derived or synthetic) to a

ROS-compatible robotic platform. We provide example implementations for neuron simulation, sensory input, and motor output, in an extensible, flexible approach.

Network Simulation and Visualization

To enable the simulation and analysis of highly recurrent, time-dependent, biofidelic spiking neurons, we propose *Electrode*, a platform that can closely model electrophysiological processes at variable levels of simulation resolution. Unlike similar frameworks that require specialized neuromorphic hardware, such as SpiNNaker (Knight, Tully, Kaplan, Lansner, & Furber, 2016) and TrueNorth (Merolla et al., 2014), *Electrode* runs on consumer hardware, enabling high-speed neurophysiological simulation on hardware platforms ranging from personal laptops to high-performance computing clusters. Although comparable software simulators exist (Stimberg, Goodman, Benichoux, & Brette, 2014), *Electrode* allows arbitrary input and output by placement of simulated electrodes. Simulated electrodes can be inserted at any point in the network to represent input (i.e., stimulation), output (i.e., recording), or both (as in biological experiments). In this way, we aim to make it possible to replicate biological experiments with *Electrode*'s "in-silico" neurons.

By using a novel, flexible abstraction for a neuron, we can enable simultaneous simulation of different levels of detail for different parts of a single model. Namely, we may use electrophysiological neuron models in critical portions of the processing unit, and use simplified neuron models for surrounding neuron populations, as has been previously demonstrated in similar simulation efforts (Hawrylycz et al., 2016). By changing the paradigm of bio-fidelic neuron simulation at scale, we believe that *Electrode* could open the door for defining innovations in the world of neuro-inspired machine learning. For larger networks, we are exploring an implementation of these ideas using distributed processing in highly parallel languages to simulate large neuron populations at faster speeds than are possible with existing implementations.

We also provide a visualization tool that enables qualitative insight into the neural activity underlying behavior. Our visualization engine, written in p5.js (p5js.org), provides a real-time visual representation of the neural activity of a simulated network (Figure 1).

Our simulation approach can be used to explore learning and neuroscience functionality toward improving robotic functionality while tailoring the simulation fidelity to the application domain.

Graphs to Robots

Our Python implementation simulates an arbitrary weighted directed graph from disk as a neuronal network, allowing virtual 'electrodes' to be attached at network inputs and outputs.

We leverage the Robot Operating System (ROS) (Quigley et al., 2009), a collection of software that provides tools for robotic software development. A main concept in ROS is the use of *topics*, which facilitate the passing of data from different processes. Processes, or nodes, can subscribe to or publish

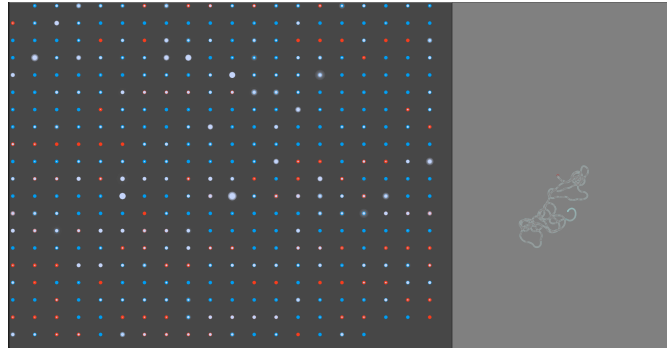


Figure 1: Our visualization engine; Neurons are represented on the left side in arbitrary order, with motor neurons and muscles colored red (Busbice, 2014). Depolarization is represented by the expansion of the pale blue highlights. On the right side, a cyan worm leaves a gray trail to enable the study of behavior of a simple simulated organism side by side with its neural activity.

to topics. In addition, ROS is supported by a large community of roboticists contributing significant functionality including perception, mapping, navigation and obstacle avoidance. We are exploring how these capabilities can be combined with biological networks to improve learning and performance.

Results

We demonstrate our generalized connectome-to-robot pipeline on a TurtleBot robot (<https://www.turtlebot.com/>).

For our neuronal network, we chose a *C. elegans* connectome described in the NeuroML (Gleeson et al., 2010) format. This format allows application-specific metadata, which in the case of our simple connectome would represent the synaptic weight between neurons. Our method supports loading any weighted directed graph that can be loaded by NetworkX (Hagberg, Swart, & S Chult, 2008). We parsed the *C. elegans* connectome file created from earlier robotic experiments (Busbice, 2014), which defines the synaptic weights based on the number of connections each pair of neurons shared. In the case of our *C. elegans* simulation, our inputs are sensory neurons and the outputs are motor neurons, although these can be user-specified.

We chose the mechanosensory neurons on the nose of the worm to be connected to the frontal 60° of a laser distance scanner on a TurtleBot. A distance closer than 0.5m will cause all of the mechanosensory neurons to fire. In order to control the movements of the robot, we followed an earlier implementation (Busbice, 2014), wherein the voltage ratio between all left and right motor neurons is used to determine direction of locomotion.

For the purposes of demonstration and fast implementation on a robot, we use the *Integrate and Fire* simple spiking neuron model, in which neurons accumulate voltage based on weighted inputs and fire when that accumulation reaches a

certain value. The firing of a neuron resets its voltage.

For simulation of the robot, we used Gazebo (Koenig & Howard, 2004). This software integrates closely with ROS and provides nodes that give the position, orientation and other state properties of the robot. The simulated world is an open space within a barrier ring with radius 9.5m. The robot was placed in the center of the ring with the same orientation at every iteration. While the robot was not receiving sensory input (i.e when it was not "touching" anything), the connectome received "food" signals by firing the gustatory neurons. This is necessary in order to encourage the robot to explore.

Connectome perturbation

Scientists have identified the GABAergic neurons expressing the gene *unc-25* (Jin, Jorgensen, Hartwig, & Horvitz, 1999), which is the same gene targeted by (Xu & Chisholm, 2016) in their ablation study, which led to inhibited locomotion. We further identified the dopaminergic and serotonergic neurons and ablated each of these groups, validating that we can change robot behavior through a targeted approach. These groups of neurons can be seen in Table 1.

Table 1: Sensory Neurons

Neuron	Neurotransmitter
AVL, DD, DVB, RIS, RME, VD	GABA
NSM, RIH, ADF, AIM	Serotonin
ADE	Dopamine

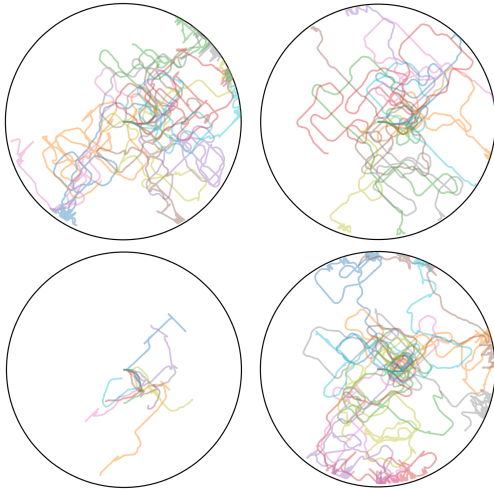


Figure 2: Movement of simulated robot traces, each initialized in scenario center. **Upper left:** Unmodified; **Upper right:** No GABAergic neurons; **Lower left:** No serotonergic neurons; **Lower right:** No dopaminergic neurons.

The results of this experiment can be seen in Fig 2. Each image represents the paths of 10 simulated robots, each of which was run for 10 real-time minutes. The robots' positions were recorded continuously and plotted. Removing the

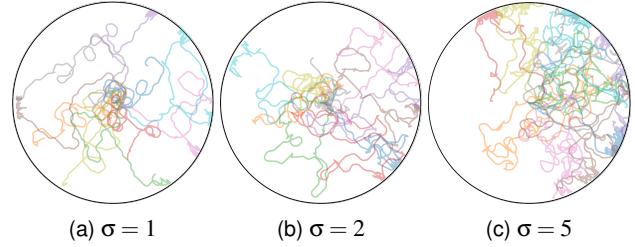


Figure 3: Simulated movement with randomized weights.

GABAergic neurons has prevented the worm from reversing when it came into contact with the barrier. In addition, it caused decreased overall locomotion: The unmodified model attained an average speed of 0.12 m/s (SEM 0.0038) and the modified model attained an average speed of 0.078 (SEM 0.0094). This is consistent with the outputs (though not necessarily the mechanism) obtained previously (Xu & Chisholm, 2016).

Finally, in order to determine the stability of neuronal net, we varied the weights by normally distributing them around the original weights with varying standard deviations. These results can be seen in Fig. 3. This shows that increasing the randomness of the network increases the exploratory nature of the robot and the randomness of its movements. Now that we have shown the ability to modify the network, we next plan to modify weights using learning algorithms to promote task specific behavior.

Discussion

In this work, we provide an easy-to-use framework to explore connectome function and robotic applications.

We note that many aspects of our model are not bio-fidelic, but that our framework is modular and can be extended to increase model fidelity if desired; the fidelity required may vary depending on the application. We briefly outline a few limitations of our model: (1) Neuron model: It is not well known how every *C. elegans* neuron operates, and while there have been developments to show that some *C. elegans* neurons use a plateau-modeled neuron (Lockery & Goodman, 2009), we chose to use this simple spiking model in order to implement it on lower level hardware. (2) Synaptic weight choice: Although we recognize that the weights of our neuronal network are derived from connection count – a measure that does not fully describe synaptic weight, we chose to use these weights because it is the best estimate we have of all the weights of the *C. elegans* connectome. (3) Robotic control: Our control model does not represent the movement of the *C. elegans* well. However, it was chosen for mathematical simplicity and reproducibility. Due to the abstraction of our method, we allow more complicated or application-specific locomotion models to be implemented.

As previously mentioned, our choice of the robotic simulation is centered around the ROS and Gazebo tools. This simulation environment offers the ability to quickly simulate

new environments, robots and sensors with varying complexity and available degrees of control. As we expand our work in this space, our goal is to evaluate the performance of different sensing modalities, ability to transfer knowledge to new robotic platforms, and create increasingly complex environments to assess the biological system's meta-learning capabilities.

In this work we aim for reproducibility and mathematical simplicity, toward creating a bio-inspired platform to explore connectome reconstructions and assess their applications, opportunities and limitations. We provide tools and abstraction layers to allow others to write and implement more complex models toward more bio-fidelic behavior.

We extend previous work (Busbice, 2014) by designing and implementing a generalized connectome-to-robot pipeline. We leverage this pipeline to simulate the effects of network changes toward exploring bio-inspired robotic applications. All of our work and data is open source, available at <https://github.com/aplbrain/>.

We hope to extend this initial work to more detailed simulation of neuronal dynamics and more biophysically plausible learning rules, with the ultimate goal of developing robust exploration strategies for simple robots. Biological wiring diagrams of the brain (i.e., connectomes) offer great promise to inform artificial intelligence applications such as robotics.

Acknowledgments

Our research was supported by JHU/APL internal funding.

References

- Bargmann, C. I., & Marder, E. (2013). From the connectome to brain function. *Nature Methods*, 10(6), 483.
- Busbice, T. (2014). Extending the *C. elegans* connectome to robotics. *SOLID*.
- Chalfie, M., Sulston, J. E., White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1985). The neural circuit for touch sensitivity in *Caenorhabditis elegans*. *Journal of Neuroscience*, 5(4), 956–964.
- Gleeson, P., Crook, S., Cannon, R. C., Hines, M. L., Billings, G. O., Farinella, M., ... Silver, R. A. (2010, 10). NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLOS Computational Biology*, 6(6).
- Gray, J. M., Hill, J. J., & Bargmann, C. I. (2005). A circuit for navigation in *Caenorhabditis elegans*. *Proc. of the National Academy of Sciences of the United States of America*, 102(9), 3184–3191.
- Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using NetworkX* (Tech. Rep.). Los Alamos National Lab.
- Hasani, R. M., Beneder, V., Fuchs, M., Lung, D., & Grosu, R. (2017). Sim-ce: An advanced simulink platform for studying the brain of *Caenorhabditis elegans*. *arXiv preprint arXiv:1703.06270*.
- Hawrylycz, M., Anastassiou, C., Arhipov, A., Berg, J., Buice, M., Cain, N., ... et al. (2016). Inferring cortical function in the mouse visual system through large-scale systems neuroscience. *Proc. of the National Academy of Sciences*.
- Jin, Y., Jorgensen, E., Hartwig, E., & Horvitz, H. R. (1999). The *Caenorhabditis elegans* gene *unc-25* encodes glutamic acid decarboxylase and is required for synaptic transmission but not synaptic development. *Journal of Neuroscience*.
- Knight, J. C., Tully, P. J., Kaplan, B. A., Lansner, A., & Furber, S. B. (2016). Large-scale simulations of plastic neural networks on neuromorphic hardware. *Frontiers in Neuroanatomy*.
- Koenig, N., & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent robots and systems, 2004. IEEE/RSJ international conference on* (Vol. 3, pp. 2149–2154).
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.
- Lechner, M., Grosu, R., & Hasani, R. M. (2017). Worm-level control through search-based reinforcement learning. *arXiv preprint arXiv:1711.03467*.
- Lichtman, J. W., & Sanes, J. R. (2008). Ome sweet ome: what can the genome tell us about the connectome? *Current Opinion in Neurobiology*, 18(3), 346–353.
- Lockery, S. R., & Goodman, M. B. (2009). The quest for action potentials in *C. elegans* neurons hits a plateau. *Nature Neuroscience*, 12.
- Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... others (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*.
- Morse, T. M., Lockery, S. R., & Ferrée, T. C. (1998). Robust spatial navigation in a robot inspired by chemotaxis in *Caenorhabditis elegans*. *Adaptive Behavior*.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., ... Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop*.
- Stimberg, M., Goodman, D., Benichoux, V., & Brette, R. (2014). Equation-oriented specification of neural models for simulations. *Frontiers in Neuroinformatics*, 8, 6.
- Szigeti, B., Gleeson, P., Vella, M., Khayrulin, S., Palyanov, A., Hokanson, J., ... Larson, S. (2014). Openworm: an open-science approach to modeling *Caenorhabditis elegans*. *Frontiers in Computational Neuroscience*, 8, 137.
- White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos Trans R Soc Lond B Biol Sci*, 314(1165), 1–340.
- Xu, S., & Chisholm, A. D. (2016). Highly efficient optogenetic cell ablation in *C. elegans* using membrane-targeted minisog. *Scientific Reports*, 6.