



A Dynamic Simulation of a Compliant Worm Robot Amenable to Neural Control

Shane Riddle^(✉) , Clayton Jackson , Kathryn A. Daltorio ,
and Roger D. Quinn 

Department of Mechanical and Aerospace Engineering, Case Western Reserve
University, Cleveland, OH 44106, USA
shane.riddle@case.edu

Abstract. This paper details the development and validation of a dynamic 3D compliant worm-like robot model controlled by a Synthetic Nervous System (SNS). The model was built and simulated in the physics engine Mujoco which is able to approximate soft bodied dynamics and generate contact, gravitational, frictional, and internal forces. These capabilities allow the model to realistically simulate the movements and dynamic behavior of a physical soft-bodied worm-robot. For validation, the results of this simulation were compared to data gathered from a physical worm robot and found to closely match key behaviors such as deformation propagation along the compliant structure and actuator efficiency losses in the middle segments. The SNS controller was previously developed for a simple 2D kinematic model and has been successfully implemented on this 3D model with little alteration. It uses coupled oscillators to generate coordinated actuator control signals and induce peristaltic locomotion. This model will be useful for analyzing dynamic effects during peristaltic locomotion like contact forces and slip as well as developing and improving control algorithms that avoid unwanted slip.

Keywords: 3D Model · Compliant Structure · Mujoco · Synthetic Nervous System · Peristalsis · Worm Robot

1 Introduction

Soft robots are desirable for their efficacy in environments and tasks unsuitable for rigid body robots. Earthworm-inspired robots are a prime example of this. Their soft bodied locomotion lends itself well to squeezing through constrained spaces such as pipes and tunnels that are impassable for rigid bodied robots [3, 13]. They can also be useful in medical procedures, like endoscopy, for which rigid structures are undesirable [15]. The compliance that makes these worm-like robots so useful also makes them difficult to model accurately. While materials engineering and manufacturing methods continue to advance, easing the design

This work was supported by NSF PIRE Award 1743475 and NSF DBI 2015317 as part of the SF/CIHR/DFG/FRQ/UKRI-MRC Next Generation Networks for Neuroscience Program.

and fabrication process, so too have the modeling methods used to simulate these increasingly complex soft bodied structures and mechanisms [20].

There are many approaches to modeling and simulating soft robots each with their own set of advantages and trade-offs. Some focus on modeling flexible deformation analytically [21]. Others use finite element methods (FEM) to characterize the behavior of more complex mechanisms [4]. To perform dynamics analysis and controller development, the model in this paper uses a physics engine. Physics engines are frequently employed for rigid body robot modeling but some can also model soft body dynamics by approximating compliant structures using many discrete rigid bodies [27]. The rigid bodies are connected via joints with flexural and torsional stiffness and allow for bending and twisting movements within the structure. These approximations allow for dynamic computational efficiency but with less precision than FEM models. Reduced precision is not necessarily a detriment so long as the accuracy of the results can be validated externally. For example, we validate the model presented in this paper by comparing our results to those from a physical robot.

Peristaltic locomotion relies on geometric coupling between segment diameter and segment length [7, 26]. As a segment's diameter contracts, its length increases and vice versa. Earthworms take advantage of their hydrostatic skeletons to accomplish this. Their bodies maintain a constant volume so a change in one dimension necessitates a change in another. Many worm-like robots [13, 22, 28] and models [3, 8, 19] utilize a two dimensional geometric approximation consisting of rhombus structures with coupled height and length. None of these models, however, accurately portray both the soft bodied dynamics and three dimensional range of motion that the physical robots exhibit. While the previous models have been used to analyze behaviors like slip, turning, and reaction to perturbations and contact surface irregularities, further refinements of the model and expansion to 3D space will improve understanding of these phenomena.

This paper details the development of a three dimensional compliant worm-like robot model compatible with a Synthetic Nervous System (SNS) controller. The SNS used to control this model was developed for and implemented on a two dimensional simulated worm robot we introduced previously [19]. An SNS is a dynamical neural network comprised of computational models of neurons and synapses which have been implemented as controllers for many biologically inspired robots [5, 8, 9, 13, 24, 25]. This SNS uses coupled central-pattern generators (CPGs), oscillators thought to control many rhythmic behaviors in animals [16], to produce a peristaltic wave-form. The wave coordinates segment muscle contraction cycles, facilitating locomotion [10, 31].

The model reported in this paper is designed to emulate the worm-like robot presented by Wang [28], which uses the rhombus segment approximation. The model is built in the physics engine Mujoco, a platform oriented towards robotics and biomechanics research. Mujoco was selected for three reasons: it easily interfaces with the SNS bio-inspired control signals, it is capable of approximating soft bodied dynamics, and it can simulate ground reaction, gravitational, elastic, and friction forces [27]. The resulting simulation captures more realistic robot

motions, including deformation propagation whereby actuating segments can deform nearby segments. Deformation propagation is modeled more accurately here than in previous models [3, 8, 12, 19].

2 Methods

2.1 2D Worm Models

Many worm-like robot models use a two dimensional segment approximation. As found in Daltorio et al. [3] (Fig. 1b) and later in our previous work [19] (Fig. 1a) these models can exist in a 2D plane with segments consisting of rigid side lengths with hinge jointed vertices. Contraction of the segment occurs when an actuator pulls two opposing vertices inward, thus pushing the remaining vertices outwards. A spring connecting the outward displaced vertices provides tension to return them to the original position, thus re-expanding the segment. The geometric relationship governing this coupling is $l^2 + w^2 = 4l_s^2$ where (w) and (l) are rhombus height and length and (l_s) is the given side length.

While both models use rigid side lengths for the rhombi, the Daltorio model incorporates torsional springs at the joints to resist differences in neighboring segment heights. This mimics behavior exhibited in continuous mesh worm-like robots where structural compliance causes the deformation of one segment to propagate to its surrounding elements. A similar model presented in Boxerbaum et al. [1] (Fig. 1c) also implements torsion springs but places them in the middle of the otherwise rigid rhombus side lengths to mimic the compliance within the segment structure rather than between segments.

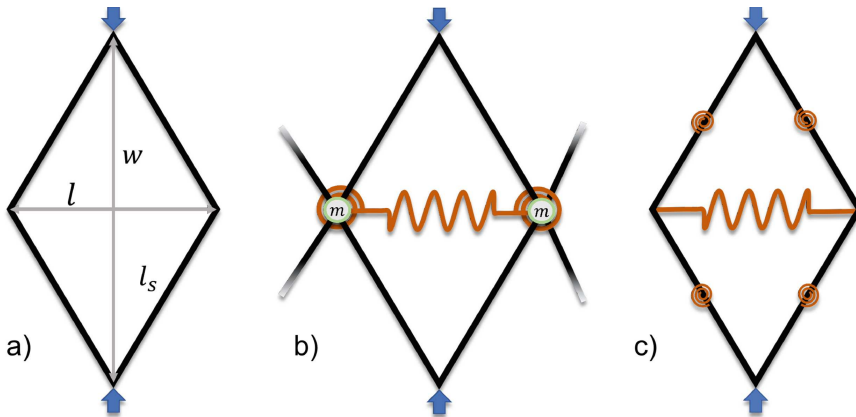


Fig. 1. Three previously reported models for 2D worm segments: a) Riddle et al. [19], b) Daltorio et al. [3], c) Boxerbaum et al. [1]. The blue arrows represent contractile actuator forces and the orange objects represent linear and torsional springs.

While useful in their own right all of these models, to varying degrees, suffer from oversimplification. Our previous model [19] was capable of peristalsis but

was purely kinematic, neglecting all forces as its purpose was only to validate the control system presented in that work. The Daltorio model simulates sagittal plane dynamics but neglects gravity, focusing instead on the larger forces provided by contact with pipe walls which limits it to such environments [14]. The Boxerbaum model also simulates 2D dynamics but neglects ground contact forces. Importantly, none of these captures the behavior of the compliant structure between actuated segments. Some worm-like robots such as MIT's mesh-worm [22] are made of a continuous mesh. They do not actuate every adjacent rhombus but instead rely on the mechanical advantage of deformation propagation along the structure (Fig. 2). To model this behavior and more accurately simulate gravitational and contact force effects, a 3D model is needed.

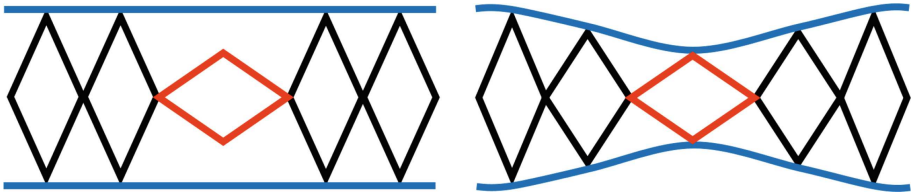


Fig. 2. Visualization of segment contraction with deformation propagation (right) and without (left). The contracting segment is orange and unactuated segments are black. (Color figure online)

2.2 3D Compliant Worm Model

Modeling a 3D robot using a physics engine has many advantages. A 2D approximation is most often implemented in physical worm robots by wrapping a mesh of rhombuses into a cylindrical tube, expanding it to 3D space [1, 13, 22, 28]. Huang et al. [8] analytically modeled the kinematics of a 3D worm-like robot in this fashion previously but utilized rigid side lengths and did not incorporate forces. Analytically deriving these forces is a difficult task, however, a physics engine can predict these. Assuming the model closely behaves like the robot, a physics engine is capable of simulating the forces without requiring an analytical derivation. This is useful for examining dynamic phenomena in complex three dimensional structures like sagging due to gravity and slip which is an important factor in peristaltic locomotion and has been the subject of many studies [3, 12, 28, 30, 32].

The 3D model described in this paper was designed to mimic the worm-like robot presented in Wang's dissertation [28] (Fig. 3). The physical robot's rhombus mesh was constructed from 12 polyethylene tubes wrapped helically in both clockwise and counterclockwise directions and pinned at the intersection points to create rotating joints. The joints serve as anchor points both for the elastic springs and for eyelets through which actuator cables were threaded.

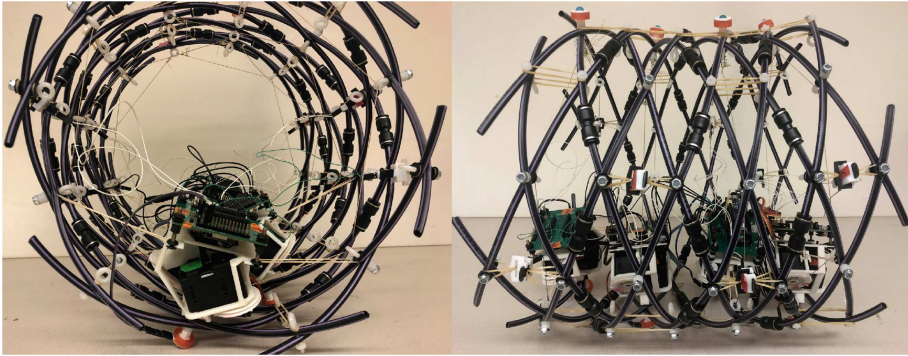


Fig. 3. The physical robot [28] the 3D model is based on from the a) axial view and b) sagittal view. Robot images provided by Wang.

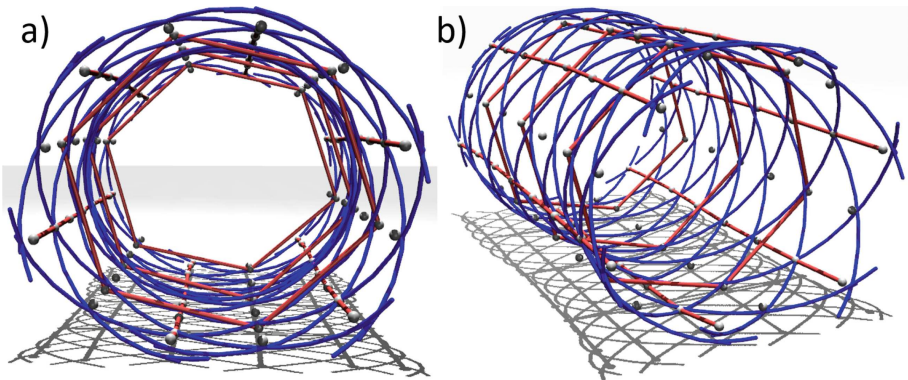


Fig. 4. Rendering of the model in Mujoco from the a) axial view and b) isometric view. The blue lines are the flexible cables used for the robot's structure. The circumferential red rings represent contractile muscle actuators. The axially oriented red lines are tendon springs. The spheres are visual markers for the joints holding everything together. (Color figure online)

The cables attach to spool wheels mounted to servo motors. When the actuators rotate, the cables are spooled in or out to shrink or expand the segment diameter, akin to the circumferential muscles found in a real worm. The number of actuated segments is defined by the number of actuator cables. The Wang robot is capable of turning so it has two motors per segment but for straight line locomotion only one motor per segment is required.

Soft Structure: The physics engine Mujoco was chosen for this task as it was designed specifically to facilitate robotics and biomechanics research [27]. The model was built by generating twelve sets of coordinates forming helical paths which follow the shapes of the polyethylene tubing in the physical robot's resting

state. Flexible cable composite objects were then defined in Mujoco overlayed on these paths. As in the physical robot, half of these twist clockwise and half twist counterclockwise with joints defined at the intersection points, thus forming the cylindrical rhombus mesh seen in Fig. 4. Mujoco models composite cable objects by discretizing them into smaller rigid body capsules connected end-to-end with flexible joints that allow for deformation of the structure. There are three such capsule discretizations per each side length of the rhombuses in this model. The coordinates and properties defining the structure and its compliant behavior are assembled in an XML file which Mujoco is able to read into the physics engine for both computation and rendering. A MATLAB code, available at <https://github.com/sriddle97/SNS-Controlled-Peristalsis.git>, was used to automate the XML generation given the desired dimensions (m), number of helixes (must be even and ≥ 6), and number of discretizations per rhombus side length.

We used a number of methods to set material and stiffness properties in the Mujoco model. The “bend” and “twist” properties of the cable object correlate to the Young’s and shear Moduli of the cable material, respectively, and dictate the degree of flexibility the structure exhibits. As such we set the “bend” and “twist” properties to their respective moduli for polyethelyene (approximately 0.3 GPa and 0.2 GPa). To ensure the model and the physical robot have the same mass, the density was also set to that of polyethylene (940 kg m^{-3}) and the cable thickness was set to match the cross-sectional area of the tubing, a necessary adjustment as Mujoco does not have an option to model hollow tubes directly. The cross sectional area for the tubing was found to be $1.7 \times 10^{-5} \text{ m}^2$ which equates to a radius of about 0.002 m for a solid circular cross section. It should be noted that Mujoco does not inherently define default units of measurement. Instead, it allows the user to define the units through the scale of their inputs. To ensure a consistent unit system for the simulated environment we define all physical properties using standard SI units (kg, m, s, N, Pa) and verified that resulting units of force were in Newtons (N).

Actuation: As a biomechanics oriented physics engine, Mujoco is capable of dynamically modeling muscles and tendons. Passive tendon objects, serving the same purpose as the elastic springs in the Wang robot, were attached to the model structure at joints running in the axial direction. These tendons are initialized such that their unstretched lengths coincide with the fully expanded resting state of the structure. Since the tendons only provide force in tension, this prevents them from expanding the model diameter beyond the physical robot’s mechanical limitations. The spring stiffness coefficient for these tendons was set to 19.9N/m to match those of the physical robot.

Rather than directly modeling servo motors, we took advantage of Mujoco’s built-in muscle actuators to provide contraction. Mujoco models these muscles as abstract force generators attached to fixed-length tendons. The muscle actuator behavior is dictated by a force-length-velocity function commonly reported in biomechanics literature [27]. The exact function used by Mujoco can be found in the FLV.m MATLAB file provided in Mujoco’s documentation and is visualized

in Fig. 5. The function requires the muscle's maximum and minimum lengths ($l_{\max}$, $l_{\min}$), the maximum shortening velocity at which the muscle force drops to zero ($v_{\max}$), the passive force generated at the muscle's maximum length ($f_{p,\max}$), and the active force generated at saturated lengthening velocity ($f_{v,\max}$). The muscles in our model use the default values Mujoco sets for these parameters except for the minimum length ($l_{\max} = 1.6$, $v_{\max} = 1.5$, $f_{p,\max} = 1.3$, $f_{v,\max} = 1.2$). $l_{\min}$ was increased from the default 0.5 to 0.58 to avoid over-contraction of the mesh.

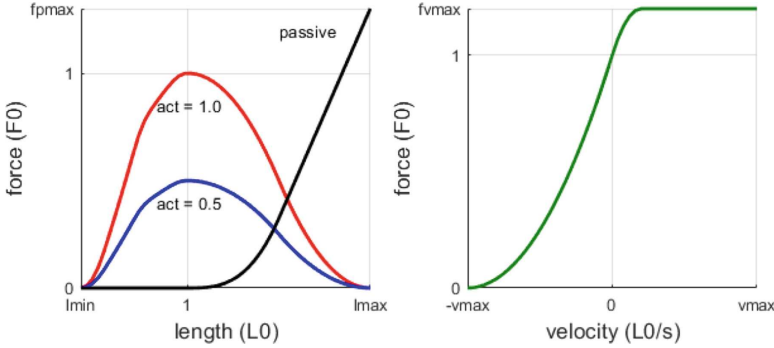


Fig. 5. Visual of the Force-Length-Velocity relationship (Eq. 1) Mujoco uses to model muscle actuator forces [27].

The forces are relative to the peak muscle activation force at zero velocity (F_0) and the lengths are relative to l_0 , the length where F_0 is generated. Both l_0 and F_0 are calculated automatically by the model compiler using the parameter values defined above. The compiler also has a scaling factor with which the user can manually adjust F_0 if needed. The actuator forces generated within the physics engine are determined using the function in Eq. 1. Here, the muscle activation signal $\text{act}(U)$ is a function of the corresponding CPG neuron membrane potentials. The muscle activation varies from 0 (deactivated) to 1 (fully activated) via a monotonic and saturating sigmoid as detailed in Sect. 2.3.

$$\text{actuator_force} = -\text{FLV}(l, v, \text{act}(U)) * F_0 \quad (1)$$

The automatically generated F_0 did not provide enough force to contract the segment so a scaling factor of 550,000 was included. This may seem high but the actuators generated forces up to 2.5 N at this scale. This is in line with the amount of force required to displace six 19.9 N/m tendon springs (the number of tendons per segment) approximately 2–3 cm, which is sufficient for appreciable segment contraction in this model.

Sensing: Mechanoreceptors and sensory neurons provide animal nervous systems with proprioceptive feedback to mediate locomotion. Worms have sensory

neurons that perceive muscle stretch and are used to control their muscle contractions during peristalsis and steering [29]. Here, these sensory neurons are approximated with mechanical stretch sensors. The stretch sensors feed the segment length data into the controller where it is used to coordinate actuation and propagate the peristaltic wave down the body. This approach is used on both the Wang robot and in our previous work where they give the SNS positional feedback from the 2D model. The data packet Mujoco outputs at each timestep of the simulation contains the tendon lengths which can be exported to the SNS controller where they are used to generate input signals for the next time step.

2.3 Synthetic Nervous System Controller

The SNS controller used in this work, seen in Fig. 6, is nearly identical to that presented in our previous paper [19].

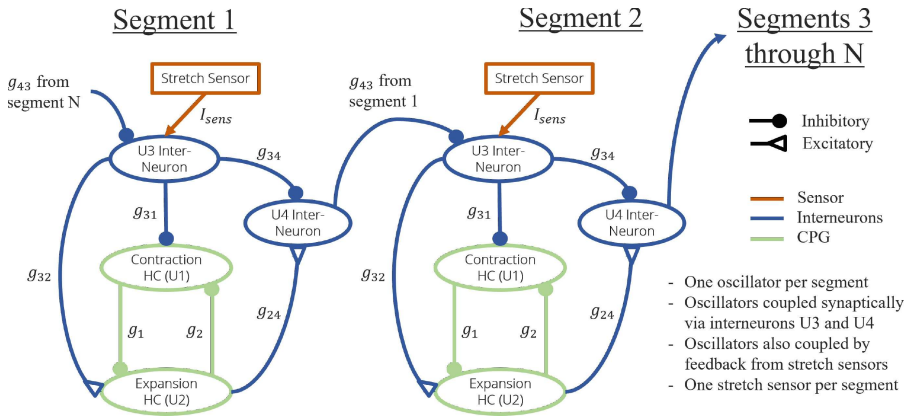


Fig. 6. Worm robot Synthetic Nervous System control network diagram (Fig. 2 of [19]).

The only substantial difference is that the network code was ported from MATLAB to python for Mujoco compatibility and reformatted using the SNS-Toolbox package developed by Nourse et al. [18]. SNS-Toolbox simplified the network building and Mujoco interfacing processes and will reduce the effort required to alter the controller for future work. As before, each segment's contraction cycle is controlled by a half-center oscillator CPG [2, 23]. The time dynamics enabled by the Hodgkin-Huxley fast transient sodium currents and mutual inhibition cause the neurons to flip between excited and inhibited states in a catch-and-release type manner [6]. The membrane potential of each half-center neuron is fed into the function shown in Eq. (2) to generate the muscle activation signal used in Eq. (1):

$$act = 1/(1 + e^{-0.35*(U_{exp} - U_{con})}) \quad (2)$$

Our previous 2D model had linear actuators that operated with position control. A piecewise linear sigmoid function, not unlike Eq. (2), mapped the

neuron potentials to position targets and a simple proportional controller dictated the actuator speed accordingly. The 3D model employs muscle actuators controlled by the force-length-velocity function in Eq. (1), so while the actuation command signal is similar, the actuation itself is much more bio-plausible than the position control. Regardless the end result is the same. When the contraction neuron potential is higher, the signal sent to the actuator tells it to contract the segment. Likewise, a higher expansion neuron potential signals for segment expansion. Interneurons and stretch sensors couple the CPGs from one segment to the next to coordinate the peristaltic wave propagation. For more details on the SNS controller, please refer to our previous publication [19].

3 Results

For a direct comparison to the previous 2D simulation results, the Mujoco model presented here is composed of 3 segments ($N=3$). In practice this can

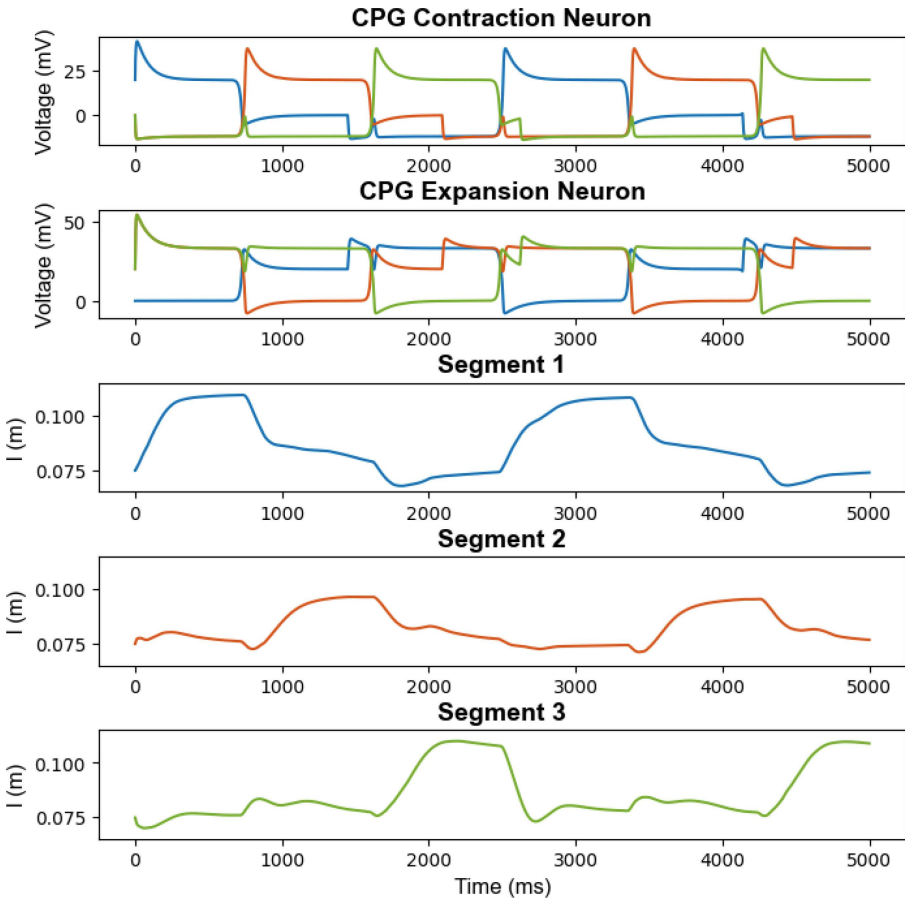


Fig. 7. Plots of the CPG neuron membrane potentials, color coded by segment number, and stretch sensor length readings for the 3 segment 3D model.

be expanded to any number $N \geq 3$ but does become more computationally expensive the larger the model gets. For reference, running this model for 5000 time steps took 3.5 min on a high-end, consumer grade computer. Rather than having actuators defined at every axially adjacent rhombus, one rhombus was left between each muscle ring. This was done to demonstrate that the structural deformation propagation can extend beyond directly neighboring rhombuses. Running the simulation for 5 s with 1 ms time steps produced the data plotted in Fig. 7. A video of this simulation as well as the code required to run it are available at <https://github.com/sriddle97/SNS-Controlled-Peristalsis.git>.

The results show that the controller generates peristaltic behavior as it did in the 2D model (see Fig. 4 in [19] for comparative data). The stretch sensor readings indicate lengthening of the segment when the contraction neuron is excited and shortening when the CPG flips and the expansion neuron is excited, as expected. Segments 1 and 3 experienced a maximum elongation of 4.20 cm and 4.02 cm respectively, a mere 4.45% difference. Segment 2, however, experienced a maximum elongation of just 2.55 cm, 39.4% less than that of Segment 1.

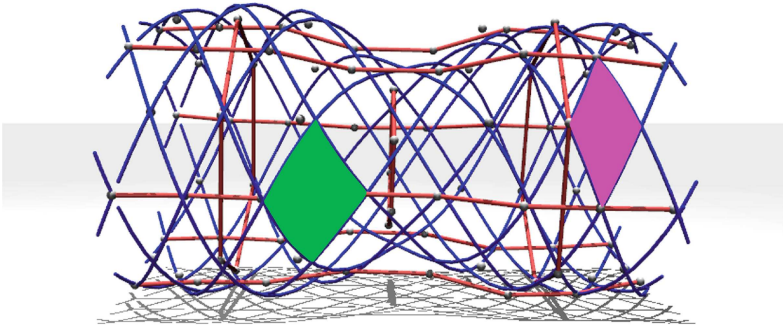


Fig. 8. The compliant worm model with the second segment contracted at $t = 1500$ ms. The green rhombus shows deformation propagating to a nearby unactuated segment. The pink rhombus indicates a segment sufficiently far from the contracted segment to remain relatively unaffected.

4 Discussion

The key difference between the 2D model data [19] and the 3D model data here lies in the stretch sensor length readings. The purely kinematic 2D model produced very smooth, idealized length changes that do not capture deformation caused by nearby segment actuation. On the other hand, the 3D model does capture the deformation propagation of the mesh, even when actuated rhombuses are not directly adjacent. This can be seen in the length plot data of Fig. 7 and is visualized with a rendering of the model during contraction in Fig. 8. When Segment 1 contracts, the length of Segment 2 also increases slightly and both

Segments 1 and 3 have elevated length readings when Segment 2 contracts since it neighbors both. This behavior closely emulates that exhibited by the Wang robot in their length readings seen in Fig. 9 (data provided by the authors [28]). The physical robot has two stretch sensors per segment to measure asymmetric contraction during turning.

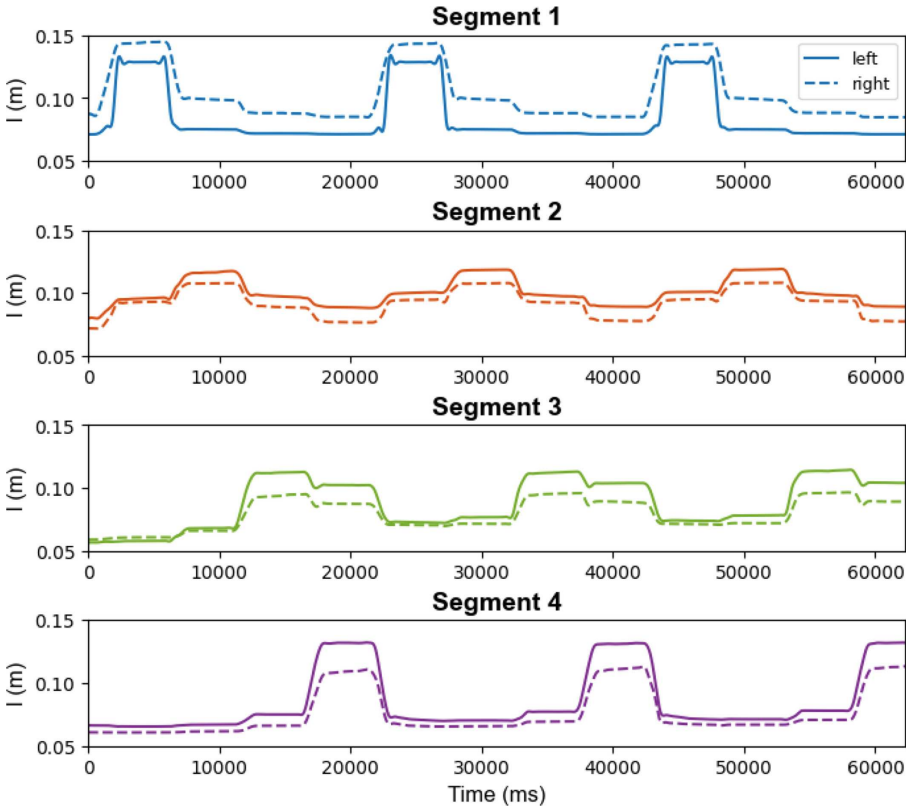


Fig. 9. Stretch sensor length readings from the physical robot during straight-line locomotion, provided by Wang [28]. The solid and dotted lines are for sensors on the left and right side of the robot, respectively.

It should be noted that Segment 2's smaller length change is due to contact frictional forces and its central location along the worm. While all segments experience losses from friction in the joints, Segment 2, unlike Segments 1 and 3, must push against the rest of the worm body in both directions in order to contract. Thus, it is reasonable to conclude that this energy loss is the result of the surrounding segments slipping along the ground as Segment 2 pushes them outward. In this simulation the robot model moves along flat ground, not inside a pipe, so ground contact forces are produced by gravitational effects.

If the model contained more segments the reduced efficiency would be present in all interior segments (2, $N - 1$). These observations are all supported by the Wang robot length data in Fig. 9 which also exhibits decreased lengthening in Segments 2 and 3 of their four segment robot ($N = 4$) while locomoting across flat ground with a non-zero coefficient of friction. The averages of the left and right sensor readings for each segment were used to compare maximum length changes to those of the 3D model. The average maximum length change of the physical robot's Segments 1–4, respectively, were 6.15 cm, 3.82 cm, 4.78 cm, and 5.98 cm. Just as in the model, the first segment elongates the most, there is a small difference between the first and last segment elongations (2.75%), and the middle segments elongate much less than Segment 1 (37.8% decrease for Segment 2, 22.2% for Segment 3).

5 Conclusions and Future Work

The primary purpose of this paper is to report the development of a 3D compliant worm-like robot model and show it is capable of accurately simulating soft bodied dynamics. The quantitative results in Fig. 7 substantiate these claims as our modeled data is comparable to that recorded from the Wang physical robot shown in Fig. 9 [28]. The qualitative evidence also supports this, as the snapshot in Fig. 8 and the video found at the linked Github page show behavior that visually matches the physical robot motion. A secondary purpose of this work was to show that the SNS developed in our previous publication could be applied to a more realistic robot model with little to no changes as we claimed [19]. Since the controller instigated peristaltic behavior in this much more realistic model, this work confirms that hypothesis.

Now that we have a model that simulates soft worm robot dynamics more realistically, we can use it to develop improved control systems. While peristaltic wave-forms were achieved, the parameters were not fine-tuned to produce appreciable directional locomotion in the simulated environment. As such the model, as presented in this paper, was only capable of achieving negligibly small speeds. This is largely due to unoptimized friction coefficients (Mujoco's default friction coefficient $\mu = 1$ was used) and a small number of segments, both of which impact locomotive efficiency [13]. As mentioned earlier, friction and slip play a large roll in peristaltic locomotion and the SNS controller, in its current state, was not designed to account for these since the initial 2D kinematic model did not simulate them. Furthermore, the tendon length data used for the approximated stretch sensor readings were somewhat idealized in that they did not include noise. Noise is present in nearly all sensor readings and must be mitigated when applying any control algorithm to a physical robot. Accounting for noise should be as simple as adjusting the stretch sensor signal threshold in the SNS controller but this remains to be validated.

Future work will focus on adjusting the controller to accommodate sensor noise and improve straight-line motion. This will likely involve adding contact pressure sensors to sense the normal forces along the underside of the worm

model. The combination of pressure and stretch sensors has been shown by Wang to provide reasonably reliable slip detection in soft worm robots [28]. We also plan to add more functionality to the SNS control system, namely adaptive peristaltic gaits. Nourse et al. [17] proposed an adaptive frequency CPG model which can increase and decrease oscillation rates in response to manual, neural, or sensory stimuli. In another paper, Ijspeert et al. [11] present a network of connected descending CPGs which exhibited the ability to transition a salamander robot's gait from walking to swimming by changing only the drive input. These works indicate that it should be feasible to implement an adaptive CPG in the SNS to enable adaptive locomotion in a worm robot. Such functionality would allow it to react to changes in its environment, such as the narrowing of a pipe, by increasing the peristaltic wave speed or even changing waveforms altogether.

Acknowledgements. We would like to thank Yifan Wang, Mingyi Wang, and Natasha Rouse for providing the sensor data from their physical worm robot and allowing us to use it in this publication for comparison to our 3D model [28].

References

1. Boxerbaum, A.S., Shaw, K.M., Chiel, H.J., Quinn, R.D.: Continuous wave peristaltic motion in a robot. *Int. J. Robot. Res.* **31**(3), 302–318 (2012)
2. Brown, T.G.: On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *J. Physiol.* **48**(1), 18–46 (1914)
3. Daltorio, K.A., Boxerbaum, A.S., Horschler, A.D., Shaw, K.M., Chiel, H.J., Quinn, R.D.: Efficient worm-like locomotion: slip and control of soft-bodied peristaltic robots. *Bioinspiration Biomimetics* **8**(3), 035003 (2013)
4. Duriez, C.: Control of elastic soft robots based on real-time finite element method. In: 2013 IEEE International Conference on Robotics and Automation, pp. 3982–3987 (2013)
5. Goldsmith, C.A., Szczecinski, N.S., Quinn, R.D.: Neurodynamic modeling of the fruit fly *drosophila melanogaster*. *Bioinspiration Biomimetics* **15**(6), 065003 (2020)
6. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **117**(4), 500–544 (1952)
7. Horschler, A.D., et al.: Peristaltic locomotion of a modular mesh-based worm robot: precision, compliance, and friction. *Soft Rob.* **2**(4), 135–145 (2015)
8. Huang, Y., Kandhari, A., Chiel, H.J., Quinn, R.D., Daltorio, K.A.: Mathematical modeling to improve control of mesh body for peristaltic locomotion. In: Mangan, M., Cutkosky, M., Mura, A., Verschure, P.F.M.J., Prescott, T., Lepora, N. (eds.) *Living Machines 2017. LNCS (LNAI)*, vol. 10384, pp. 193–203. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63537-8_17
9. Hunt, A., Szczecinski, N., Quinn, R.: Development and training of a neural controller for hind leg walking in a dog robot. *Front. Neurobot.* **11**, 18 (2017)
10. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* **21**(4), 642–653 (2008). *Robotics and Neuroscience*

11. Ijspeert, A.J., Crespi, A., Ryczko, D., Cabelguen, J.M.: From swimming to walking with a salamander robot driven by a spinal cord model. *Science* **315**(5817), 1416–1420 (2007)
12. Kandhari, A., Wang, Y., Chiel, H.J., Daltorio, K.A.: Turning in worm-like robots: the geometry of slip elimination suggests nonperiodic waves. *Soft Rob.* **6**(4), 560–577 (2019)
13. Kandhari, A., Wang, Y., Chiel, H.J., Quinn, R.D., Daltorio, K.A.: An analysis of peristaltic locomotion for maximizing velocity or minimizing cost of transport of earthworm-like robots. *Soft Rob.* **8**(4), 485–505 (2021)
14. Keudel, M., Schrader, S.: Axial and radial pressure exerted by earthworms of different ecological groups. *Biol. Fertil. Soils* **29**(3), 262–269 (1999)
15. Kim, B., Young, H., Hyeon, J., Park, J.O.: Inchworm-like colonoscopic robot with hollow body and steering device. *JSME Int. J. Ser. C Mech. Syst. Mach. Elements Manuf.* **49**, 205–212 (2006)
16. Marder, E., Bucher, D.: Central pattern generators and the control of rhythmic movements. *Curr. Biol.* **11**(23), R986–R996 (2001)
17. Nourse, W., Quinn, R.D., Szczecinski, N.S.: An adaptive frequency central pattern generator for synthetic nervous systems. In: Vouloutsis, V., et al. (eds.) *Living Machines 2018. LNCS (LNAI)*, vol. 10928, pp. 361–364. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95972-6_38
18. Nourse, W.R.P., Szczecinski, N.S., Quinn, R.D.: SNS-toolbox: a tool for efficient simulation of synthetic nervous systems. In: Hunt, A., et al. (eds.) *Biomimetic and Biohybrid Systems. Living Machines 2022. LNCS*, vol. 13548, pp. 32–43. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20470-8_4
19. Riddle, S., Nourse, W.R.P., Yu, Z., Thomas, P.J., Quinn, R.D.: A synthetic nervous system with coupled oscillators controls peristaltic locomotion. In: Hunt, A., et al. (eds.) *Biomimetic and Biohybrid Systems. Living Machines 2022. LNCS*, vol. 13548, pp. 249–261. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20470-8_25
20. Schmitt, F., Piccin, O., Barbé, L., Bayle, B.: Soft robots manufacturing: a review. *Front. Robot. AI* **5**, 84 (2018)
21. Sen, S., Awtar, S.: A closed-form nonlinear model for the constraint characteristics of symmetric spatial beams. *J. Mech. Des.* **135**(3) (2013)
22. Seok, S., Onal, C.D., Cho, K.J., Wood, R.J., Rus, D., Kim, S.: Meshworm: a peristaltic soft robot with antagonistic nickel titanium coil actuators. *IEEE/ASME Trans. Mechatron.* **18**(5), 1485–1497 (2013)
23. Szczecinski, N.S., Hunt, A.J., Quinn, R.D.: Design process and tools for dynamic neuromechanical models and robot controllers. *Biol. Cybern.* **111**(1), 105–127 (2017)
24. Szczecinski, N.S., Hunt, A.J., Quinn, R.D.: A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion. *Front. Neurobot.* **11**, 37 (2017)
25. Szczecinski, N.S., Quinn, R.D.: Template for the neural control of directed stepping generalized to all legs of MantisBot. *Bioinspiration Biomimetics* **12**(4), 045001 (2017)
26. Tanaka, Y., Ito, K., Nakagaki, T., Kobayashi, R.: Mechanics of peristaltic locomotion and role of anchoring. *J. R. Soc. Interface* **9**(67), 222–233 (2012)
27. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE (2012)

28. Wang, Y.: Preparing worm-like robots for unknown environments: perception and path planning. In: CWRU Dissertation (2022)
29. Yeon, J., et al.: A sensory-motor neuron type mediates proprioceptive coordination of steering in *C. elegans* via two TRPC channels. *PLoS Biol.* **16**(6) (2018)
30. Zarrouk, D., Sharf, I., Shoham, M.: Analysis of earthworm-like robotic locomotion on compliant surfaces. In: 2010 IEEE International Conference on Robotics and Automation, pp. 1574–1579 (2010)
31. Zhou, Q., Xu, J., Fang, H.: A CPG-based versatile control framework for metameric earthworm-like robotic locomotion. *Adv. Sci.* **n/a**(n/a), 2206336 (2023)
32. Zimmermann, K., Zeidis, I.: Worm-like locomotion as a problem of nonlinear dynamics. *J. Theor. Appl. Mech.* **45**, 179–187 (2007)