

A Geometric Kinematic Model for Flexible Voxel-Based Robots

Maryam Tebyani^{1,*}, Alex Spaeth^{1,2}, Nicholas Cramer³, and Mircea Teodorescu^{1,2}

1 Department of Electrical and Computer Engineering, University of California, Santa Cruz, Santa Cruz, California, United States

2 Genomics Institute, University of California, Santa Cruz, Santa Cruz, California, United States

3 NASA Ames Research Center, Moffett Field, California, United States

* mtebyani@ucsc.edu

Abstract

Voxel-based structures provide a modular, mechanically flexible periodic lattice which can be used as a soft robot through internal deformations. To engage these structures for robotic tasks, we use a finite element method to characterize the motion caused by deforming single degrees of freedom and develop a reduced kinematic model. We find that node translations propagate periodically along geometric planes within the lattice, and briefly show that translational modes dominate the energy usage of the actuators. The resulting kinematic model frames the structural deformations in terms of user-defined control and end effector nodes, which further reduces the model size. The derived Planes of Motion (POM) model can be equivalently used for forward and inverse kinematics, as demonstrated by the design of a tripod stable gait for a locomotive voxel robot and validation of the quasi-static model through physical experiments.

Keywords: Modular Robots, Flexible Locomotion, Kinematic Modeling

1. Introduction

Soft robots leverage the deformation of their body to create intelligent motion. The changes to the state of their body are a function of the robot's material properties and geometry.¹

Previously, Jenett et al. proposed a new class

of soft continuum robots where cellular composite structures with low density and high specific stiffness form modular soft robots.^{2,3} Discrete lattice-based building materials can combine large numbers of identical discrete volumetric pixels, or “voxels,” to form a metamaterial⁴ with several promising applications, allowing advances in aircraft where wings can passively and actively morph to increase aerodynamic efficiency and control authority,⁵ large-scale spacecraft can be autonomously assembled and monitored in outer space,^{6,7} and modular flexible robots can generate undulatory locomotion using linear actuation.⁸ Such modular robotic structures benefit from reconfigurability, element replacement, scalability, and decreased production costs.^{9,10}

However, the design and modeling of voxel-based continuum robots is limited by the complexity of available modeling approaches. A review of model-based and model-free static and dynamic controllers for soft robotic manipulators concludes that hybrid modeling is a promising area of research.¹¹ Recently, machine learning has been employed to develop differential models for closed-loop soft robotic control.¹² Model-based methods to predict the movement of soft robots often segment the body into constituent elements which range in computational cost from finite element models, which can take hours to simulate, to piecewise constant curvature models^{13–15} which can be used for real-time control.

Similar methods have been used to characterize a beam-shaped robotic swimmer constructed from voxels.¹⁶ However, more general robots cannot be modeled this way because of their three-dimensional lattice structure. Voxel structures have been characterized as materials,^{4,17} but limited work has been done to show how these structures can be employed for robotic tasks. Previ-

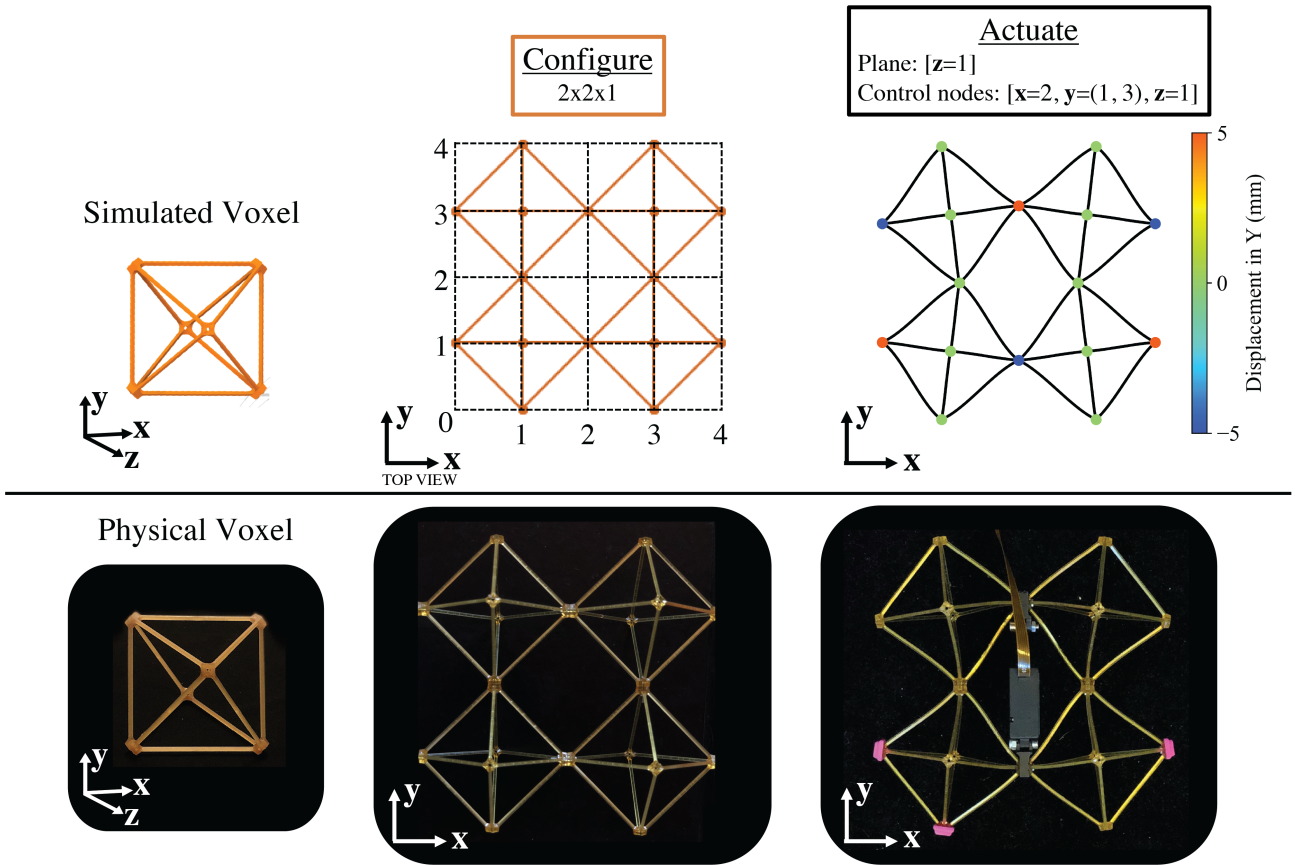


Figure 1. Voxel robots are designed by specifying the configuration and actuation of the voxel structure. The images on the left show a single voxel element as a 3D model on top and its physical counterpart on bottom. The middle images show a structure of 4 voxels. The images on the right show the resulting displacement of the voxel body when the actuation is applied via a linear actuator between two control nodes. Beam bending is approximated using a cubic spline for visualization purposes.

ously, Cramer et al. modeled quasi-static locomotion of a voxel robot using a linear finite element method, but finite element approaches are not suitable for real-time applications.⁸

In this paper, we attempt to address this deficiency by introducing a simple geometric model of deformation in voxel structures which enables a computationally cheap approximation of the position of nodes of interest as a function of actuator-induced structural deformation. This model can be used for the design of locomotive gaits, as well as for embedded control with modest resource requirements, highlighting how actuation of compliant structures can produce constrained motion which still benefits from the structure’s embedded intelligence. First, the motion produced by actuating voxel structures is studied. Emergent patterns in the voxel movement are used to define general relations for the motion of nodes under linear actuation, allowing the development of a geometric model which we call the Planes of Motion (POM)

model. Next, a voxel-based quadrupedal robot is designed using the derived simulation.

Experimental results show that the simplified model produces a kinematic template of voxel node displacements.

2. Methodology

The voxels used in this work are regular cubic octahedra structured as 12 beams joined at 6 nodes. This geometry can be understood as three orthogonally intersecting squares. Voxels are assembled into flexible structures which can be deformed by embedded linear actuators.

To illustrate the deformation behavior of our voxel structures, we use the open-source Python library PFEA⁸ to simulate the displacements of individual nodes. PFEA is a finite-element solver optimized for voxel structures, which calculates node positions under the assumption that each node has zero extent and all beams are linear. Ad-

ditionally, both PFEA and our geometric model to be introduced later make a quasi-static assumption; the low inertia of the voxels and high viscous friction of the actuators result in a slow-moving system which remains close to equilibrium.

PFEA results are compared to physical voxels in figure 1. Four voxels are arranged in a two-by-two square formation, and a linear actuator is placed between them. As the actuator expands, it forces the positions of opposite nodes apart, deforming the full structure. In the figure, nodes are colored according to their displacement in the vertical direction as calculated by PFEA.

2.1. Geometric Relations

Studying the configuration and actuation shown in figure 1, it is notable that constraints and loading conditions in simulation and the actuator mounts on the physical structure inhibit the rotational degrees of freedom of the nodes that they connect; the actuation strictly displaces a single, linear degree of freedom of the selected pair of control nodes.

For example, the actuator in figure 1 connects neighboring nodes which share an x and z coordinate. Voxels are made of narrow plastic beams which can bend easily but retain their shape and length, so lattice deformations effectively rotate voxels in place. This displaces other nodes which fall on the $z = 1$ plane, in a direction that accommodates the inciting node motion. Thus, a displacement pattern emerges and propagates throughout the actuated plane. This pattern is shown in figure 2, where the loads and constraints from figure 1 are applied on a larger voxel structure. Although the same force is applied, the displacements are smaller in figure 2 because more voxels must be deformed. This pattern does not decay with distance, as demonstrated by the Displacement Dissipation plot in figure 2. This implies that node translation dominates energy usage, as only up to 4.9% of displacement is lost to other modes, such as node rotation or beam bending.

Figure 3 further demonstrates that significant displacement within the structure is constrained to specific nodes as a function of the voxel geometry. Nodes outside of the actuated plane do not displace translationally. The rotation of the voxel

within a plane does rotate its out-of-plane nodes, but this rotation seems to dissipate in the beams without rotating the next layer of voxels. For instance, for this actuator in the $z = 1$ plane, other nodes on the $z = 0$ and $z = 2$ planes will rotate to remain in line with their voxels, but without leading to any movement of the adjacent nodes on the $z = -1$ and $z = 3$ planes, which would correspond to the rotation of other voxel layers.

With this in mind, the observation underlying our model is that in a voxel lattice, deformations amounting to a rotation of one of the squares making up each voxel require the least energy to perform, and propagate through the lattice within a single plane by rotating the corresponding square in the opposite direction in each adjacent voxel. The result is that nodes which lie on the same plane experience displacements of the same magnitude, with direction alternating along each axis.

Figure 4 illustrates this alternation as well as our numerical conventions for numbering of voxel nodes and actuation dimensions. Each actuation dimension rotates a corresponding square of the voxel, highlighted in blue on the voxel image on the left. This leads to displacements of the nodes located on that square, shown with arrows emanating from those nodes on the right. To approximate how voxel nodes will move based on the plane of motion that is activated by the actuated nodes, we consolidate this information into three matrices A_d corresponding to the three actuation directions d , and indexed by actuation dimension and voxel node index, as follows:

$$A_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The columns of A_d specify how a control node displacing along direction d will affect the position of each node of any voxels in the corresponding plane of motion, with direction and node indices according to the convention of figure 4. In

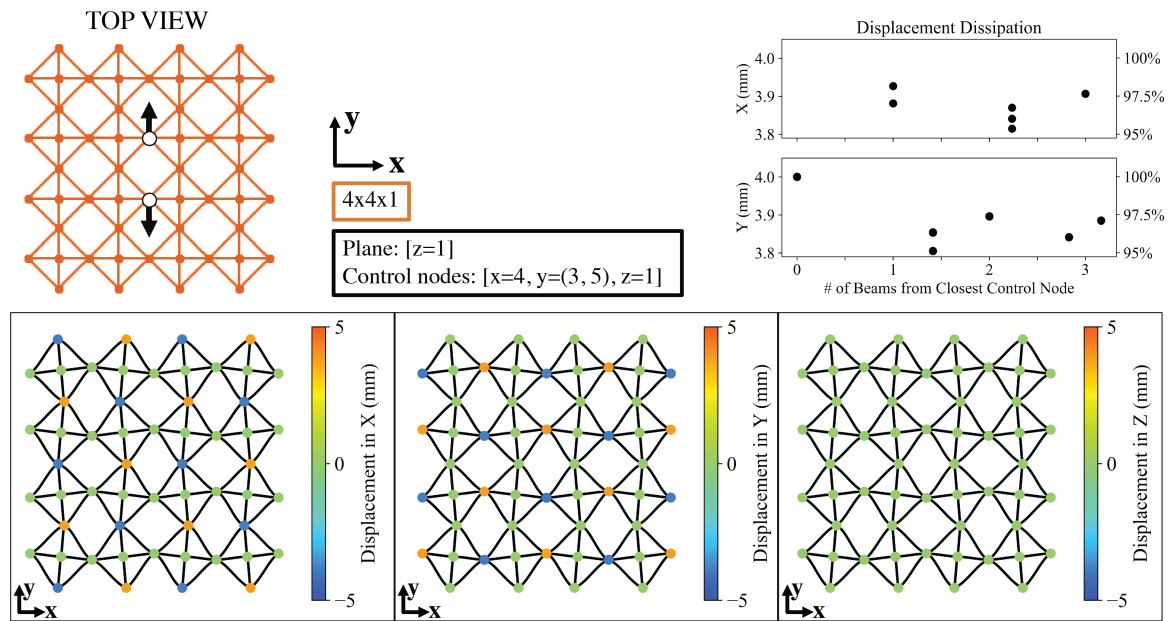


Figure 2. A 4x4x1 structure is actuated to demonstrate the emergent displacement pattern arising from the constrained degrees of freedom. The node coloring of the left bottom image shows the displacement magnitudes in the X dimension, the middle shows the displacement magnitudes in the Y, and the right in the Z. Finally, the Displacement Dissipation plot in the top right shows how energy absorbed by other modes, like node rotation and beam bending, diminishes node displacements.

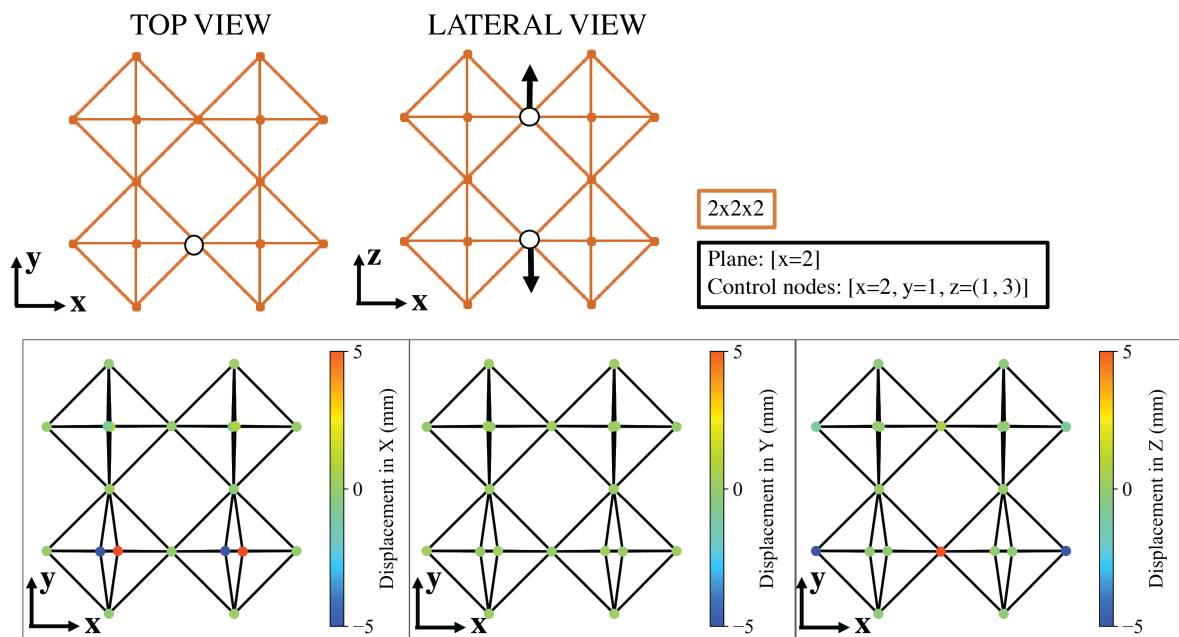


Figure 3. A 2x2x1 voxel configuration is actuated to demonstrate the minimal displacement of neighboring planes. Beams are plotted here as straight lines to make the three-dimensional structure clearer.

Actuation Dimension (0-2) Based on Voxel Node Index (n0-n5)

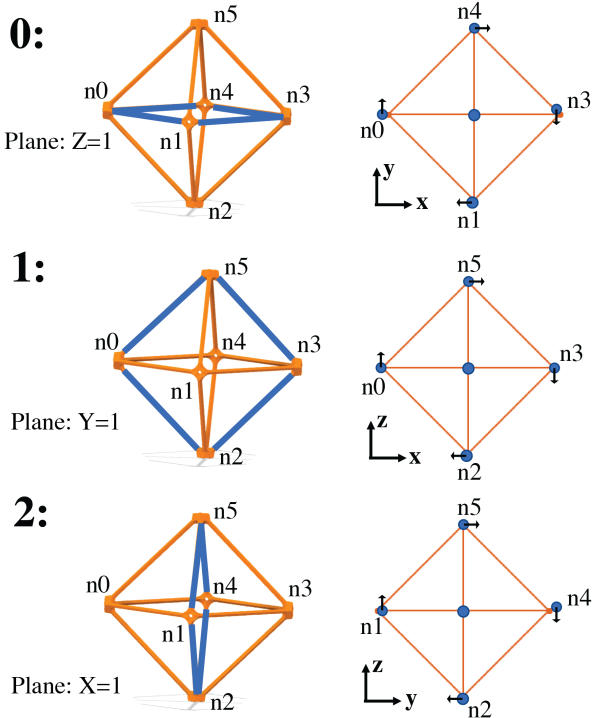


Figure 4. The numerical convention for voxel node and actuation dimension indexing. Actuation along a given plane, highlighted in blue (left), will translate voxel nodes in that plane about the corresponding voxel center, as indicated with the black arrows (right).

the model, it is useful to refer to these columns as vectors in space, for which we introduce the notation $\mathbf{a}_{d,n}$ to refer to the n th column of the matrix A_d , i.e. the displacement produced on node n of each voxel in the implicated plane by a unit control node displacement in direction d .

2.2. Planes of Motion (POM) Model

The arrays A_d are then used to define a reduced-dimension kinematic model of actuated voxel structures by considering a subset of the voxel nodes. At time t , the 3D displacement of the end effector nodes is defined by the current displacement of the control nodes. The vector $\mathbf{q}(t)$ in configuration space describes the displacement of the control nodes, and $\mathbf{x}_j(t)$ is the 3D displacement of the j th end effector. The connectivity of the voxel structure is defined by the array c_{ij} , which specifies the effect of control node i on end effector j . These entries are zero if the two nodes do not share a plane of motion; otherwise, be-

cause of the alternating nodal displacement pattern, they are set to -1 or $+1$ depending on the parity of the position difference. Together with the actuation dimension d_i of the i th control node and the node index n_j defining which node of its corresponding voxel the end effector occupies, this map fully specifies how displacements of a given control node deform the voxel structure. These geometric constants, specified numerically according to the conventions of figure 4, are calculated in software based on more physically intuitive quantities: the grid position of the voxel and corresponding within-voxel node ID of each control or end effector node, and the actuation direction of any control nodes.

Finally, the kinematic motion is simulated by calculating the contributions of each control node to each end effector node's position. So long as no two control nodes share a plane of motion, they will specify orthogonal displacements for any affected end effectors, which can simply be summed. For each control node i , a column \mathbf{a}_{d_i,n_j} corresponding to node index n_j is selected from the actuation matrix A_{d_i} for the actuation dimension d_i . The result is multiplied by the masked control node displacement, then summed for all control nodes.

$$\mathbf{x}_j(t) = \sum_{i=0}^{M-1} (\mathbf{a}_{d_i,n_j})(c_{ij}q_i(t)) \quad (1)$$

In summary, we reduce the space of nodes to focus on the nodes of interest: the actuated nodes, and those intended to interact with the environment. It is important to note that the POM model defines the relationship between two sets of node displacements, regardless of their robotic function. So far, we have designated a control node which is connected directly to the actuator and analyzed the movement of chosen "end effectors". Instead, we could just as well have defined a desired displacement for an end effector node and computed the corresponding actuator displacement. Essentially, the forward and inverse kinematic models of the robot are derived in the same way and take the same form. In fact, the same simulation is used to compute both the forward and inverse kinematics for the robotic locomotor presented in the next section.

3. Results

Previous work showed that rectilinear locomotion can be achieved by periodically deforming a voxel structure using bi-directional friction to promote forward motion.⁸ The feet of these robots slid along the ground, but real walking gaits are generated by lifting and lowering feet. As a step towards this type of gait, we designed an intermediate locomotive robot where the front feet raise and lower while the rear feet play a supporting role, remaining planted on the ground while shifting back and forth to adjust the robot’s balance. The core of this robot consists of the $2 \times 2 \times 2$ voxel structure shown in figure 3, to which two additional voxels with liftable front feet are attached; this configuration is detailed in figure 5.

This robotic configuration was previously used to demonstrate how spiking neural state machines can construct central pattern generators and account for the voxel’s elastic properties through proprioceptive feedback and frequency entrainment.^{18,19} Here, we provide the geometric intuition and modeling required to develop the desired gait and foot fall pattern, which highlight the benefits of reduced state kinematic models of flexible robots.

3.1. Gait Design

The robot is controlled by four linear actuators placed vertically between the bottom and top layer of its core. Each actuator is connected between two nodes in the voxel structure, shown as colored circles in figure 5; the associated POM are indicated with colored lines. In a forward kinematic view, where actuators are defined as control nodes, voxel nodes which fall on these planes can be controlled along a single dimension according to the general matrix relations given in Section 2. For instance, the red actuator controls nodes that fall along the red line in either the x or z dimension, depending on the node ID.

Nodes which fall on the intersection of two actuators’ POM can be actuated independently in two directions. Four such nodes are designated as end effectors and indicated by purple markers in the figure. The four feet of the robot are indicated by the black dotted circles. The two back feet move identically to markers 1 and 4 due to the

symmetry of the robot. For example, to move the back left foot “forward”, the red actuator needs to displace the end effector in positive x , and the yellow actuator needs to displace the end effector in positive y . The motion of the front feet is complicated by the irregularity of the voxel structure; the POM model requires at least two antagonistic voxels. Instead, the front voxels are modeled as independent geometric extensions of the nodes to which they are attached.

The desired node displacements of the feet presented in figure 5 describe a gait wherein the robot takes a forward step with the right front foot and then the left front foot. At alternating points throughout the actuation sequence, the front feet take turns lifting while the back feet shift between the heel and toe of the foot and displace in the forward direction. Backward steps can be achieved by running the foot positions in reverse.¹⁸

A highlight of the POM approach is that in a single pass it can both compute the inverse kinematics to designate the actuation sequence as well as a sort of “lateral kinematics”, equivalent to a composition of inverse and forward kinematics. This process predicts the motion of nodes which can be experimentally validated.

3.2. Robotic Experiment

A physical realization of the designed voxel robot with 3D printed back feet and “sticky” front feet performs the specified gait on a smooth surface as shown in figure 5. Actuation is performed by four linear actuators which vary the total distance between attached nodes by ± 10 mm, displacing each by ± 5 mm. Details about the design and fabrication of the feet, voxel material properties, and linear actuators are provided in the supplementary information (SI).

To track the position of the robot through time as it carries out its gait sequence, a 2D motion capture (mocap) system was set up. Four purple markers were placed on the nodes specified as end effectors to track. In addition to mocap data, actuator positions were also recorded via on-robot datalogging for comparison with model results.

To compare the observed mocap data to the predictions of the POM model, a preprocessing algorithm is employed which transforms the recorded coordinates $m_i^L(t)$ of the mocap markers in the

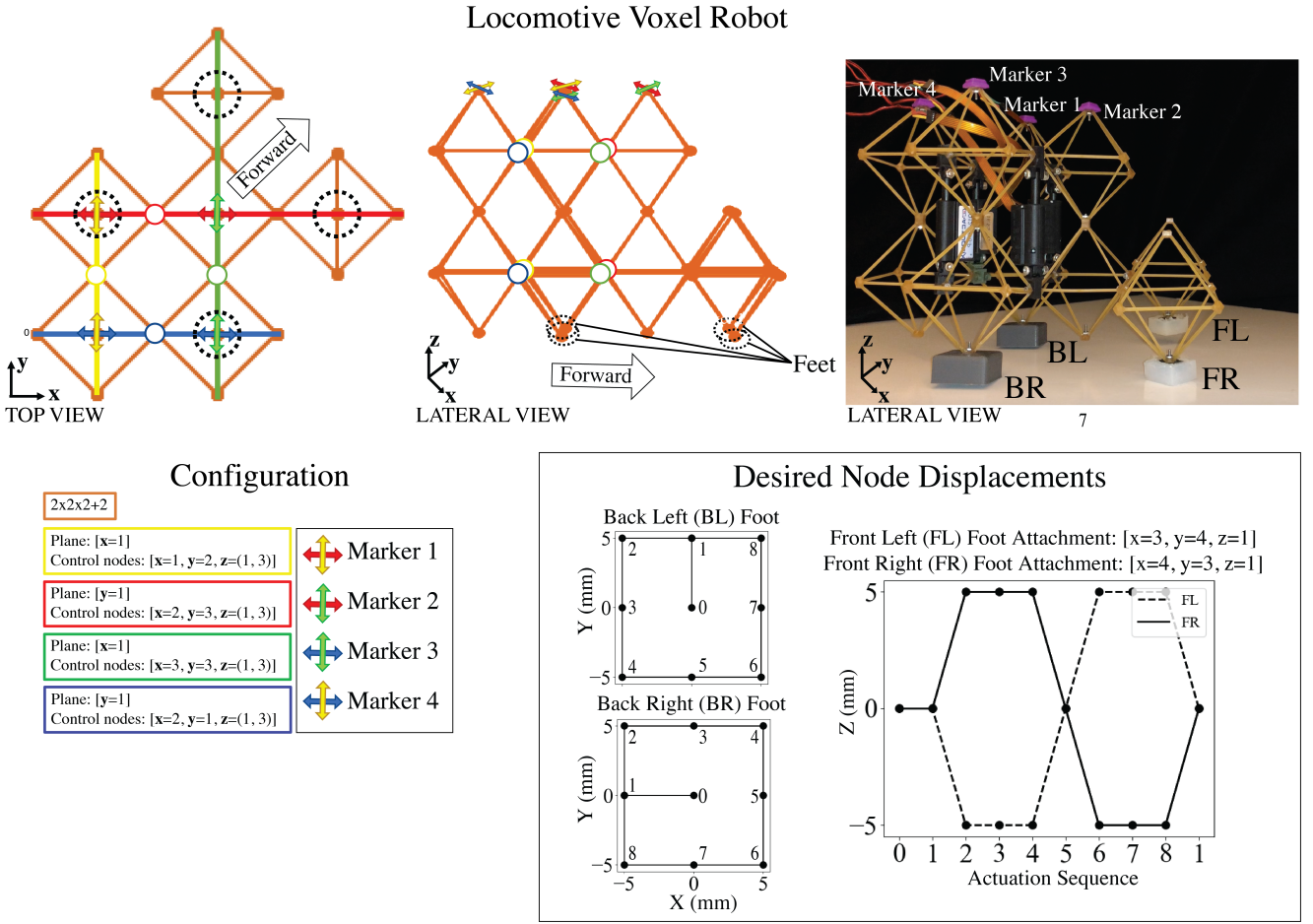


Figure 5. Top: The voxel configuration of the proposed locomotive robot. The two left schematics show a top view and a lateral view of the configuration, where 4 actuators are depicted with colored dots, 4 feet nodes are indicated by black dotted circles, and 4 markers are indicated by crossing arrows. The right image shows the physical prototype with the 4 physical actuators, 4 feet, and 4 (purple) markers. Bottom: The voxel configuration and actuator placement is detailed using the previously defined global coordinate system and a legend is given for the 4 markers shown in the schematics. On the right, the desired node displacements of the 4 feet to achieve the proposed locomotive gait are shown. The back feet nodes travel along the ground in the x and y direction while the front feet nodes travel along the vertical z dimension. These node displacements and robot configuration are fed into the POM model to produce the actuator positions required to achieve the feet displacements shown.

stationary lab frame to the corresponding coordinates $\mathbf{m}_i^R(t)$ in a robot-centered frame consistent with the model output. This transformation is necessary because the POM model does not address the contact dynamics that would be required to simulate the gait directly in the global frame. First, the mean position of the four markers $\bar{\mathbf{m}}^L(t) = \frac{1}{4} \sum_{i=1}^4 \mathbf{m}_i^L(t)$ is subtracted, as this represents motion of the center of mass; then, the marker positions are transformed to polar coordinates $(r_i(t), \theta_i(t))$, and their common rotation $\bar{\theta}(t)$ is likewise subtracted to account for twisting and turning of the body. Then, the positions are converted back to Cartesian coordinates $\mathbf{m}_i^R(t)$, and the amplitude is rescaled according to the variance

in marker position before and after the transformation. Each step of this correction is illustrated in the SI.

Figure 7 shows the data from the actuation logging and from the mocap system tracking the position of the markers in time. Here, the POM model is used for forward kinematics. The actuator data is plugged in as the actuation sequence to the POM model, along with the control and end effector node parameters, and produces the same node displacements as predicted in figure 6, but with slight noise. The mocap data of the purple markers show that the robot turns while locomoting, which could be due to systematic asymmetries like voxel fatigue, actuator extension and

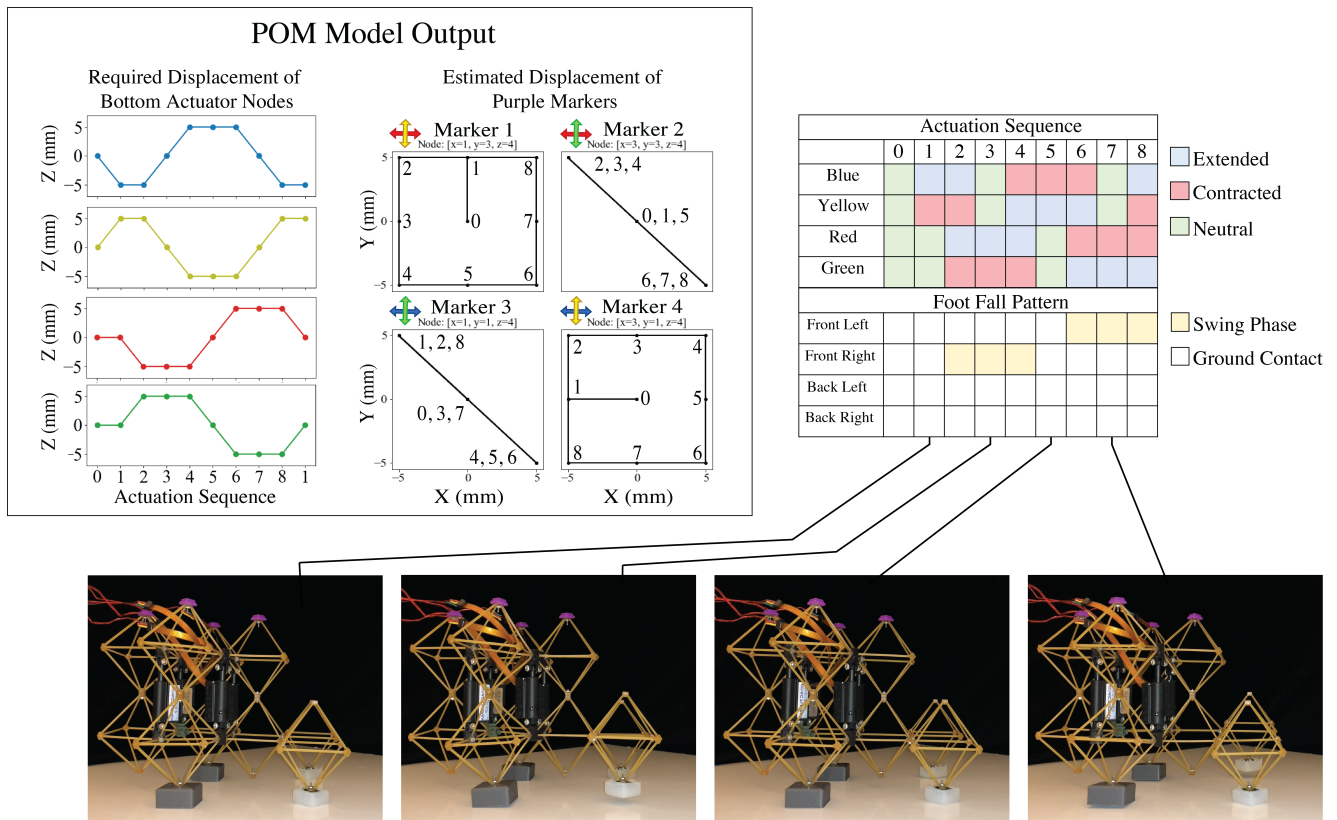


Figure 6. The POM Model Output plots show the required node displacements of the bottom actuator connection sites to achieve the desired gait, and the expected displacement of the purple markers obtained from plugging in the voxel configuration and desired feet motions to the POM model. This output defines the actuation sequence as a function of the foot fall pattern. The bottom row shows the physical robot in key stages throughout the sequence.

contact dynamics. After transformation into the body frame, the markers follow the same shape as the predicted kinematic template.

4. Conclusion

We have introduced a modeling method for flexible voxel robots which directly relates the positions of “control” (input) and “end-effector” (output) nodes, significantly reducing model space. The resulting kinematic templates can be used to design the robot and its movement. A locomotive voxel robot was designed using this framework and the approach was validated with physical experiments. End effector trajectories are predicted with surprising accuracy considering the simplicity of the model. The validity of the POM model on voxel-based structures has exciting implications for modeling flexible robots. This POM model can be used to quickly design end effector trajectories for locomotive gaits over specific terrain or to achieve other robotic tasks, like grasping or object manipulation. To improve the robot’s de-

sign and motion, an optimization algorithm could minimize the number of voxels and actuators in a structure, similar to work that has been done to computationally design mechanical characters whose end effectors follow user-defined trajectories.²⁰ To correct for systematic asymmetries, like turning or voxel fatigue, a calibration step could be included, where data collected from the movement of a voxel robot is fed into a trajectory optimization based on locomotive modeling, as proposed by Bittner et al. for planar serpent-like swimmers.²¹

Acknowledgements

The authors would like to acknowledge the technical support of the Braingeneers research group as well as a donation made possible by Eric and Wendy Schmidt by recommendation of the Schmidt Futures program.

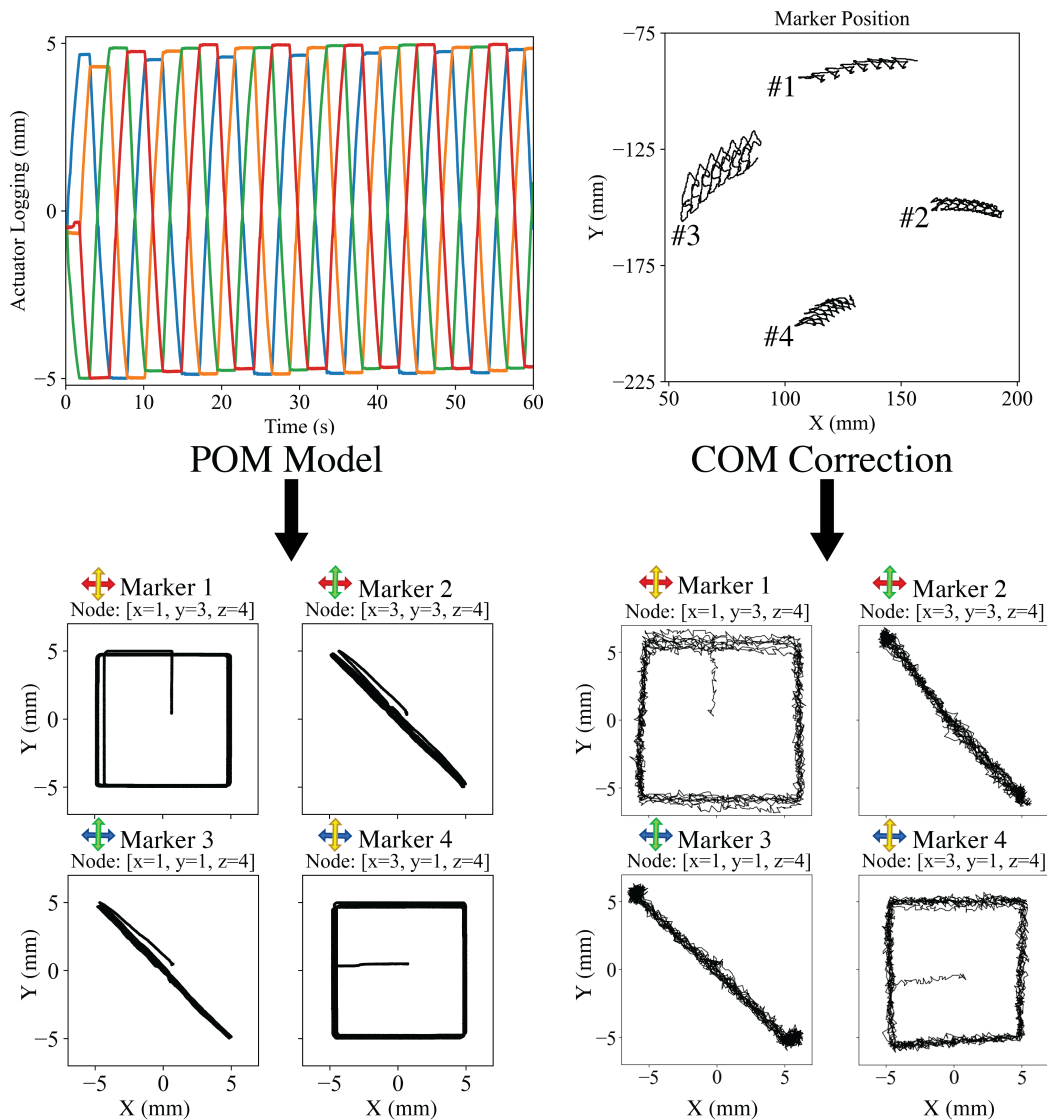


Figure 7. The actuator logging data is run through the POM model to estimate the displacement of the markers. The marker positions from the motion capture data are corrected for dynamic effects to estimate the displacements of the markers.

Author Disclosure Statement

The authors declare no conflicts of interest.

References

- ¹ D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, pp. 467–475, May 2015. Number: 7553 Publisher: Nature Publishing Group.
- ² B. Jenett, S. Calisch, D. Cellucci, N. Cramer, N. Gershenfeld, S. Swei, and K. C. Cheung, “Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures,” *Soft Robotics*, vol. 4, pp. 33–48, Oct. 2016. Publisher: Mary Ann Liebert, Inc., publishers.
- ³ B. Jenett, C. Cameron, F. Tourlomousis, A. P. Rubio, M. Ochalek, and N. Gershenfeld, “Discretely assembled mechanical metamaterials,” *Science Advances*, vol. 6, p. eabc9943, Nov. 2020.
- ⁴ A. Rafsanjani, K. Bertoldi, and A. R. Studart, “Programming soft robots with flexible mechanical metamaterials,” *Science Robotics*, vol. 4, Apr. 2019. Publisher: Science Robotics Section: Focus.
- ⁵ N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, and K. C. Cheung, “Elastic shape morphing of ultralight structures by programmable

- assembly,” *Smart Materials and Structures*, vol. 28, p. 055006, May 2019.
- ⁶ G. Trinh, G. Copplestone, M. O’Connor, S. Hu, S. Nowak, K. Cheung, B. Jenett, and D. Cellucci, “Robotically assembled aerospace structures: Digital material assembly using a gantry-type assembler,” in *2017 IEEE Aerospace Conference*, pp. 1–7, Mar. 2017.
- ⁷ B. Jenett, C. Gregg, D. Cellucci, and K. Cheung, “Design of multifunctional hierarchical space structures,” in *2017 IEEE Aerospace Conference*, (Big Sky, MT, USA), pp. 1–10, IEEE, Mar. 2017.
- ⁸ N. Cramer, M. Tebyani, K. Stone, D. Cellucci, K. C. Cheung, S. Swei, and M. Teodorescu, “Design and testing of FERVOR: Flexible and reconfigurable voxel-based robot,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2730–2735, Sept. 2017. ISSN: 2153-0866.
- ⁹ A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, “Current trends in reconfigurable modular robots design,” *International Journal of Advanced Robotic Systems*, vol. 14, p. 1729881417710457, May 2017. Publisher: SAGE Publications.
- ¹⁰ I.-M. Chen and M. Yim, “Modular Robots,” in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 531–542, Cham: Springer International Publishing, 2016.
- ¹¹ T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, “Control Strategies for Soft Robotic Manipulators: A Survey,” *Soft Robotics*, vol. 5, pp. 149–163, Apr. 2018.
- ¹² K. Chin, T. Hellebrekers, and C. Majidi, “Machine Learning for Soft Robotic Sensing and Control,” *Advanced Intelligent Systems*, vol. 2, no. 6, p. 1900171, 2020.
- ¹³ R. J. Webster and B. A. Jones, “Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review,” *The International Journal of Robotics Research*, vol. 29, pp. 1661–1683, Nov. 2010.
- ¹⁴ R. K. Katzschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, “Dynamic Motion Control of Multi-Segment Soft Robots Using Piecewise Constant Curvature Matched with an Augmented Rigid Body Model,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, (Seoul, Korea (South)), pp. 454–461, IEEE, Apr. 2019.
- ¹⁵ N. Kuppuswamy and J.-P. Carbajal, “Learning a Curvature Dynamic Model of an Octopus-inspired Soft Robot Arm Using Flexure Sensors,” *Procedia Computer Science*, vol. 7, pp. 294–296, Jan. 2011.
- ¹⁶ B. E. Jenett, *Discrete mechanical metamaterials*. Thesis, Massachusetts Institute of Technology, 2020. Accepted: 2021-05-14T16:29:23Z.
- ¹⁷ A. M. Popescu, *Vibrations Analysis of Discretely Assembled Ultra-Light Aero Structures*. PhD thesis, UC Santa Cruz, 2019.
- ¹⁸ A. Spaeth, M. Tebyani, D. Haussler, and M. Teodorescu, “Neuromorphic Closed-Loop Control of a Flexible Modular Robot by a Simulated Spiking Central Pattern Generator,” in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 46–51, May 2020.
- ¹⁹ A. Spaeth, M. Tebyani, D. Haussler, and M. Teodorescu, “Spiking neural state machine for gait frequency entrainment in a flexible modular robot,” *PLOS ONE*, vol. 15, p. e0240267, Oct. 2020. Publisher: Public Library of Science.
- ²⁰ S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, “Computational design of mechanical characters,” *ACM Transactions on Graphics*, vol. 32, pp. 1–12, July 2013.
- ²¹ B. Bittner, R. L. Hatton, and S. Revzen, “Geometrically optimal gaits: a data-driven approach,” *Nonlinear Dynamics*, vol. 94, pp. 1933–1948, Nov. 2018.