Any-axis Tensegrity Rolling via Symmetry-Reduced Reinforcement Learning

David Surovik, Jonathan Bruce, Kun Wang, Massimo Vespignani, Kostas E. Bekris

Abstract Tensegrity rovers incorporate design principles that give rise to many desirable properties, such as adaptability and robustness, while also creating challenges in terms of locomotion control. A recent milestone in this area combined reinforcement learning and optimal control to effect fixed-axis rolling of NASA's 6-bar spherical tensegrity rover prototype, SUPERball, with use of 12 actuators. The new 24-actuator version of SUPERball presents the potential for greatly increased locomotive abilities, but at a drastic nominal increase in the size of the data-driven control problem. This paper is focused upon unlocking those abilities while crucially moderating data requirements by incorporating symmetry reduction into the controller design pipeline, along with other new considerations. Experiments in simulation and on the hardware prototype demonstrate the resulting capability for any-axis rolling on the 24-actuator version of SUPERball, such that it may utilize diverse ground-contact patterns to smoothly locomote in arbitrary directions.

1 Introduction

Tensegrity rovers are dynamic truss structures that can deform passively, so as to avoid local stress accumulation, and actively, to locomote [13, 6]. Their passive properties aid active control through morphological computation [4], highlighting a close kinship with soft robots. Among other favorable traits, the adaptability and resilience of tensegrity structures makes them appealing for robotic exploration of extreme environments. This has recently led to the second hardware iteration of NASA's self-landing rover prototype, SUPERball (`SBv2`) [17], which comprises 6 rigid bars and 24 length-actuated elastic cables, shown in Fig. 1.

The same structural properties that motivate use of mobile tensegrities also make them challenging to control due to high system dimensionality, nonlinearity and coupling. Model-based approaches have been applied effectively for structural shape control [14] and can be adapted for locomotion [8]. Nonetheless, an impractical degree of state knowledge is typically required, and the neglect of the dynamical subtleties of a frequently changing contact state may hinder performance. This is increasingly leading to the application of data-driven approaches, such as evolutionary algorithms and machine learning, for controlling mobile tensegrities [5, 2, 9, 12, 3].

David Surovik, Kun Wang, and Kostas Bekris are with the Computer Science Department of Rutgers University, NJ, USA. e-mail: {ds1417,kw423,kostas.bekris}@cs.rutgers.edu.
Jonathan Bruce and Massimo Vespignani are with the NASA Ames Intelligent Robotics Group, CA, USA. e-mail: {jonathan.bruce,massimo.vespignani}@nasa.gov.

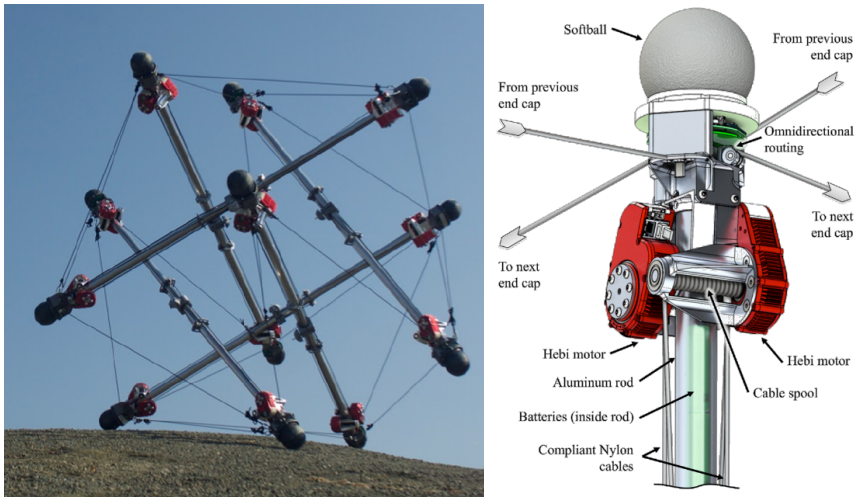

Fig. 1 Left: SUPERball version 2 at NASA Ames Research Center. Right: at either end of a rigid bar (in compression), motorized spools actuate the rest-lengths of elastic cables (in tension) [17].

1.1 Background

The context and contributions of this paper are better understood by first considering the topology of SUPERball, shown in Fig. 2. The edges of its convex hull consist of the 24 cables (solid black lines) and 6 "virtual" edges (dotted lines). Triangular faces of the hull are bordered by either three cables (Δ faces, light blue) or two (Λ faces, not shaded). The dotted gray box in 3D defines a cut that corresponds to the gray borders in 2D. Thick beige lines are bars, interior to the convex hull.

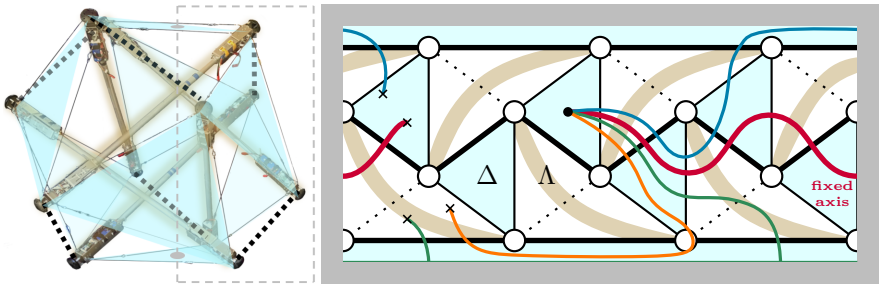

Fig. 2 Topology of SUPERball in 3D and 2D. Colored lines represent paths of the center-of-mass projected downward onto the convex hull of the vehicle during locomotion.

The previous version of SUPERball (`SBv1`) featured an identical structural topology to `SBv2` but could only actuate half of its cables, shown by the thicker black lines in Fig. 2. This arrangement constrained locomotion to follow one repeating pattern of contact states progressing horizontally on the diagram, i.e., periodic

fixed-axis rolling illustrated by the red trajectory. A control policy for generating this motion was obtained for SBv1 using Guided Policy Search (GPS) [18], a reinforcement learning method that will be described in Sec. 2.1. Ultimately, computing this policy required use of GPS individually for each of six sub-sequences of the contact pattern before finally merging the corresponding neural networks.

The fully-actuated nature of SBv2, along with an increased range of motion of each cable, permits a much broader degree of shape control [17]. This gives rise to the new possibility of any-axis locomotion, i.e., the ability move in arbitrary directions by using other contact patterns and transition geometries as shown by the green, blue, and orange paths in Fig. 2. Achieving such behaviors under the approach of [18] would require a drastic increase in the number of policy search instances and thus experimentation costs, among other potential issues. This work incorporates additional features into the policy search pipeline to produce a deployable control policy for any-axis locomotion of SBv2 while avoiding any increase in sample data requirements.

2 Approach

2.1 Prior Tools

Guided Policy Search (GPS) is a reinforcement learning method that leverages optimal control principles to reduce the amount of sampled experimental data required to compute a control policy [7]. This technique fits linear time-varying dynamics $\mathbf{x}_{t+1} = f_i(t, \mathbf{x}_t, \mathbf{u}_t)$ in numerous local regions i and computes optimal linear feedback policies $\mathbf{u}_t = p_i(t, \mathbf{x}_t)$. Then, a single neural network policy $\mathbf{u}_t = \pi(\mathbf{y}_t(\mathbf{x}_t))$ is trained to match the set of local policies given sensor data $\mathbf{y}_t$. Subject to considerations of observability, this allows the comprehensive state knowledge available within a simulator to be leveraged when optimizing a policy that can be deployed under limited sensing. The process, sketched in Fig. 3, is iterated using new samples generated by the neural network in order to converge upon a high-performing global policy.

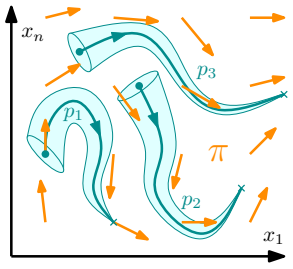

Fig. 3 Closed-loop dynamics; GPS trains a neural network π to match local optimal control laws p_i.

Model-Based Control faces some limitations when applied tensegrity motion, due to the influence of evolving contact states and the complications of gradient-based operations on a high-dimensional state. Nonetheless, certain simplifications can be used to generate useful controls with reasonable compute effort. Under full knowledge of the vehicle's current shape and assumptions about its contact state, it is possible to solve for cable length changes that displace the center-of-mass enough

to cause locomotion via instability [8]. An apriori policy such as this can be used to reduce the required number of GPS iterations relative to an initial null policy.

Exploitation of Vehicle Symmetry furthermore reduces the amount of data necessary per iteration of GPS. Due to the highly regular topology of SBv2, there exist 24 operations H_j that permute the labels of individual structural elements and transform the spatial reference frame without altering the intrinsic dynamics of the system. The policy may then take the form $\mathbf{u} = H_j^{-1}\pi(H_j\mathbf{y})$. With proper selection of a ruleset $j = j(\mathbf{y})$, the required coverage of the observation space $\mathscr{Y}$ is condensed by a factor of 24 to the subset $\mathscr{Y}_R$, as shown in Fig. 4 [15].

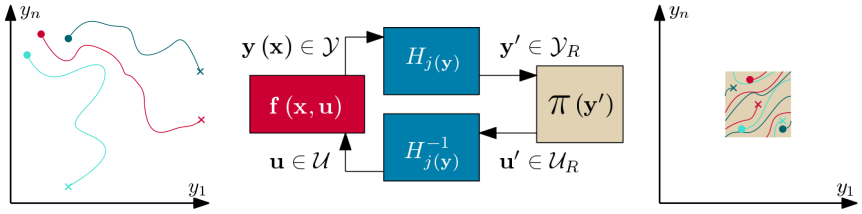

Fig. 4 Trajectories in raw (left) and symmetry-reduced (right) observation spaces. Center: the data-driven feedback policy controls the true state using only symmetry-reduced knowledge.

2.2 Policy Search Pipeline

Overview: The basic procedure for reinforcement learning is to obtain sample data by executing a control policy under various conditions, update the control policy based on how well each sample performed, and repeat. As in prior work [18], NASA's Tensegrity Robotics Toolkit [1] is used as a simulation testbed for generating samples. To bootstrap the learning process, the simple model-based controller [8] is used as the initial policy π_0. Fig. 5 gives the pipeline for improving the policy over one iteration of GPS. Use of sample segmentation and of dynamics fitting within a feature space, rather than the full state space, are primary new aspects.

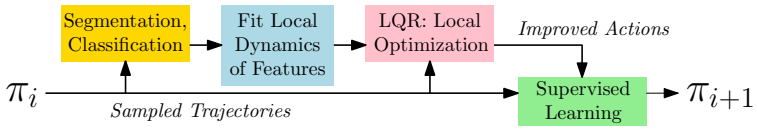

Fig. 5 The flow from sample data to policy improvement over a single iteration of GPS.

Segmentation and Classification of sample data is necessary for coping with the broad variety of motion under any-axis rolling, where nearby trajectories can rapidly diverge. Trajectories are thus segmented whenever the ground-projection of the center-of-mass crosses an edge of the convex hull (i.e., when a black line is crossed in Fig. 2). Segment categories are then assigned by the type of bottom triangle, Δ or Λ, as well as the relationship between the edge crossed *onto* it and the edge crossed *off of* it.

This process interplays with symmetry reduction in order to enable gradient-based optimization despite the use of a non-differentiable black-box dynamics testbed. For example, left-turning segments on Δ faces exhibit relatively similar dynamics and therefore can contribute to a common local model to be used for computing improvements to the corresponding actions. The prior fixed-axis rolling controller had relied upon short, similar motion samples of a single category to ensure this apriori [11].

Dynamical Feature Space: Even when sample data are well-organized, high dimensionality impedes the fitting of local dynamics models. Lower-dimensional features are instead selected based the intuition about the system dynamics: the primary external influences are normal and friction forces on contact nodes and gravity acting upon the center-of-mass. The center-of-mass-relative positions of the bottom triangle nodes, (ρ_A, ρ_B, ρ_C) are thus chosen as elements of the feature state χ.

Cost Function: The objective is simply to match the center-of-mass velocity $\bar{\mathbf{v}}$ to a reference $\bar{\mathbf{v}}^*$, giving linear quadratic cost $J = (\bar{\mathbf{v}} - \bar{\mathbf{v}}^*)^T W (\bar{\mathbf{v}} - \bar{\mathbf{v}}^*)$. The weight matrix W penalizes ground-plane components equally and de-weights the vertical. Thus, fitting dynamics of $\chi = (\rho_A, \rho_B, \rho_C, \bar{\mathbf{v}})$ provides enough information to improve the control time series while ignoring less well-behaved state variables.

Observation Space: the supervised learning phase trains on tuples $(\mathbf{y}', \mathbf{u}'')$, where $\mathbf{y}'$ are sensor-derived inputs and $\mathbf{u}''$ are improved associated actions. The observation vector contains the 24 cable rest lengths, the 6 bars' angular velocity vectors, the ID of the bottom triangle, and the commanded direction of motion.

3 Simulated Results

3.1 Single-Shape Controller

Preliminary efforts to train a neural network policy with a model-based supervisor yielded a noteworthy result: an undertrained network produced a constant output, i.e., the policy $\pi_C(H_{j(\mathbf{y})}\mathbf{y}) = \mathbf{u}^*$. This represents single shape that is a broad average of all shapes experienced by the locomoting model-based controller. Embodying this shape causes a transition between support faces, triggering a change in the raw control $\mathbf{u} = H_j^{-1}\mathbf{u}^*$ via the symmetry-based relabeling alone. The resulting sustained motion, seen in Fig. 6, resembles a simple "step-wise" paradigm [16] directable along any ground contact edge-normal, or the periodic fixed-axis motion of [18] and the red trajectory of Fig. 2. This *Single-Shape* controller thus serves as a useful point of comparison for the desired any-axis control.

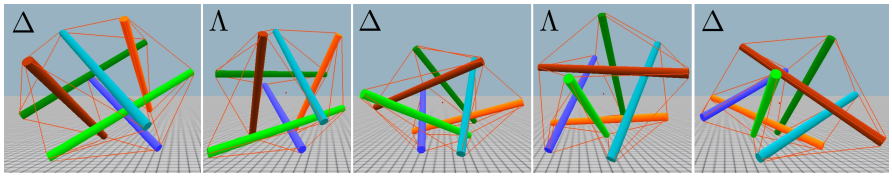

Fig. 6 A sequence of states as the Single-Shape control policy locomotes `SBv2` to the right.

3.2 Any-Axis Performance

After basic tuning of the neural network training phase, the application of the policy search update step described by Fig. 5 was repeatedly applied. This iteratively reduces the average sample cost J, corresponding to steady and effective directed locomotion with minimal side-to-side motion. Fig. 7 illustrates this improvement with top-down views of typical center-of-mass trajectories for three controllers attempting to visit a sequence of waypoints. A diagram of the robot marks its starting position and size; its center-of-mass must reach the interior of each gray waypoint before proceeding to the next.

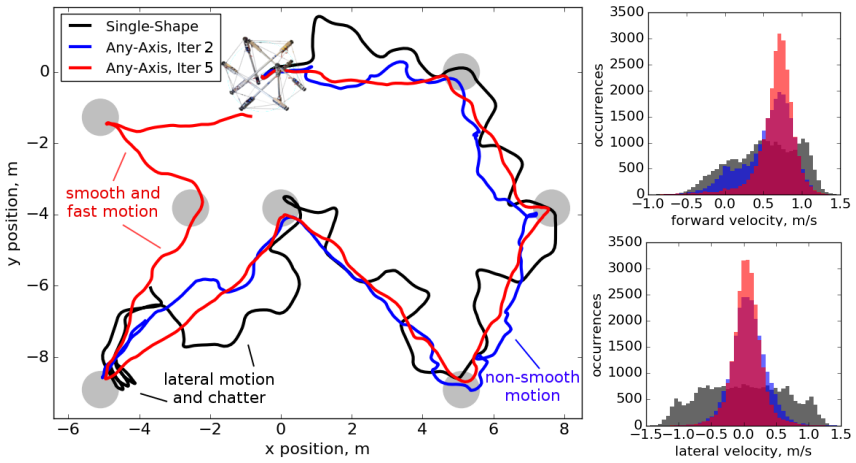

Fig. 7 Iterative performance improvement under policy search. Left: Top-down view of center-of-mass trajectories visiting a waypoint sequence. Right: histograms of forward and lateral velocities.

The Single-Shape controller shows zig-zag type motion characteristic of stepwise controllers that move the center-of-mass perpendicular to the forward edge. In some circumstances, such as the bottom-left of the figure, chatter results from its inability to transition over virtual edges. The Any-Axis controller generated by round 2 of policy search locomotes successfully, though it exhibits some irregularity that indicates halting progress. By the round 5 of policy search the controller exhibits relatively smooth paths with few irregularities.

Also shown are velocity distributions aggregated across many trials of each controller in the waypoint scenario. The velocity component toward the next waypoint should ideally be a narrow peak at the target velocity of 0.8 m/s. The Any-Axis-5 controller most closely resembles this distribution, while the Single-Shape controller has a much broader distribution and Any-Axis-2 has a high peak at zero caused by stuck states. Lateral velocity shows similar findings, with the exception that zero is the desired value and so target and stuck states are coincident.

3.3 Any-Axis Characteristics

Further illustration of the nature of the Any-Axis controller can help to reveal the capabilities of this policy search pipeline and the possibilities for future results. Fig. 8 shows a ground-track path of the Any-Axis controller with locations of the three nodes of the bottom face. Modest movement of these nodes is visible as they either slide along the ground or drift in the air during a ground-edge pivot. Some crossings of edge centers resemble prior $\Delta \rightarrow \Lambda \rightarrow \Delta$ motions; other crossings occur very close to nodes, while a crossing of a dotted edge indicates a $\Lambda \rightarrow \Lambda$ transition.

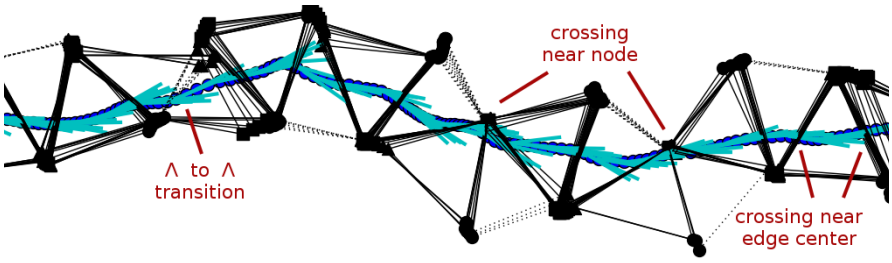

Fig. 8 Center-of-mass and bottom face geometries as the controller locomotes to the right. Light blue lines stem from the center-of-mass location and point in the velocity direction.

Figure 9 more broadly demonstrates this point using two simple geometric measures of the center-of-mass crossing an edge. The crossing *position* describes the center-of-mass location along the edge as a percentage of its total length, while the crossing *angle* relates the center-of-mass velocity direction to the edge direction. Scatter plots of these two values reveal a far wider distribution of behavior for the Any-Axis controller than for the step-wise Single-Shape controller. It is noted, however, that the high-frequency execution of the Single-Shape feedback law in scenarios involving turns does produce some notable variation due to dynamics.

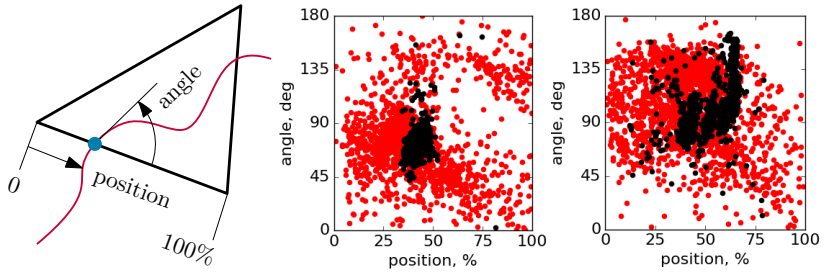

Fig. 9 Left: definition of two quantities describing the geometry of the center-of-mass path over an edge. Center and right: distributions for crossings onto Δ faces and onto Λ faces. Black points: Single-Shape controller; red points: Any-Axis controller.

4 Hardware Experiments

Due to the complex interplay of friction, elasticity, and high dimensionality, the "reality gap" between simulated and hardware behavior of mobile tensegrities is often substantial. Resolving this gap is a deep and profound challenge [10], with limited first steps lying within the scope of this investigation.

Calibration of control inputs (in the form of cable rest lengths) was found to be the most direct route to correcting errors, as the vehicle's shape has a sensitive "tipping-point" relationship with the resulting motion. This was implemented with a choice of two parameters relating hardware and software lengths: the reference length $\hat{u}$, and a scale factor c, such that $\mathbf{u}^*_{HW} = c\left(\mathbf{u}^*_{SW} - \hat{u}\right) + \hat{u}$. Small-scale testing with the Single-Shape controller determined values of $\hat{u} = 0.97m$ and $c = 1.2$ to best approximate intended behavior on hardware (see Fig. 10).

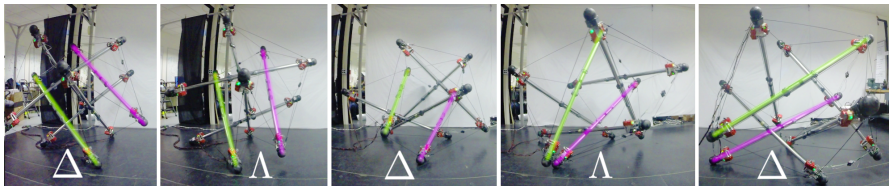

Fig. 10 Execution of the Single-Shape policy on the hardware prototype `SBv2`.

Hardware execution of the Any-Axis neural network policy introduced sensor-related considerations. Locomotion in an arbitrary direction requires knowledge of the vehicle's world frame orientation, or heading, defined via the bottom face. As localization data is not presently available, a simple scheme was introduced to approximate changes in heading based on the sequence of bottom face transitions.

In simulation, this bottom face is computed from node positions, while on hardware it is determined by a classifier that was trained on raw sensor data. As a result, the moment of transition can differ between these two testbeds. Combined with the discontinuous nature of the symmetry-reduced observations, this frequently caused forward/backward chatter of the hardware platform at transition points. Adding a time-delay on the transition detection minimized the occurrence of this issue.

Figure 11 provides the ground contact pattern of a successful hardware trial. Although center-of-mass data could not be obtained, the contact sequence nonetheless reveals directed any-axis locomotion similar to that of the simulated result of Fig. 8. While numerous hardware trials were similarly successful, occasional issues remained. Some transitions resulted in a significantly different heading change than was approximated, causing unintended turns in the trajectory. Some configurations also resulted in "stuck" states on the verge of an intended forward transition.

Footage of the hardware trial is available in the accompanying video. As it reveals, the speed of motion was also found to be significantly less than in simulation. The bar angular rates, important for moderating speed in simulation, could thus be omitted on hardware since the vehicle was not capable of exceeding the desired speed. Friction behavior also differed, with less smooth sliding of bottom nodes.

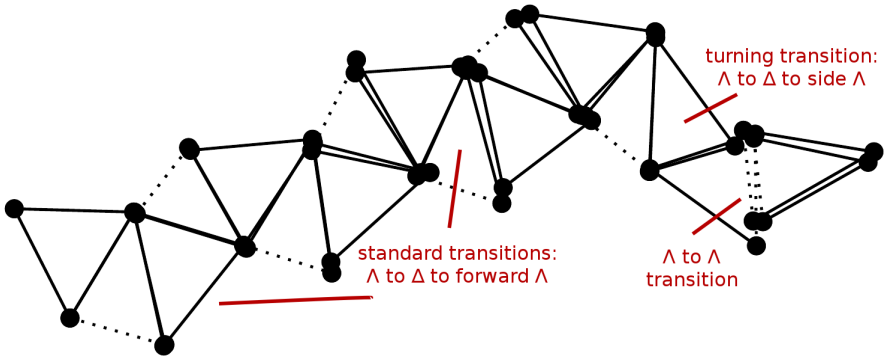

Fig. 11 Bottom face path of Any-Axis locomotion by the hardware prototype `SBv2`.

5 Discussion

Main Progress: The synthesis of techniques within this paper has produced the first feedback controller to smoothly locomote a 6-bar tensegrity rover in arbitrary directions, independent of orientation. This Any-Axis controller essentially guides state evolution in a variable direction along a 2D+ manifold of the state-space, rather than tracking a periodic trajectory or discrete points.

Symmetry-reduced optimization and training required a total of 125,000 time steps sampled at 10Hz. Previously, without symmetry reduction, 120,000 were needed for training fixed-axis motion with half as many actuators [18]. The broader control manifold is thus learned without significant increase in sample complexity.

Future Directions: Beyond potentially aiding obstacle avoidance, the modified policy search approach may have profound implications for locomotion on unstructured landscapes, a primary motivation for tensegrity use. Traversing rugged features could require both nuanced direction control as well as adaptiveness to non-flat contact geometry, which implies further broadening of the state-space volume navigated by the controller. This would require a corresponding increase in the sample complexity of dynamics fitting and supervised learning, making the applied symmetry reduction and segment classification all the more essential.

Reality Gap: As was seen, many significant differences between the simulation and hardware testbeds resulted in an imperfect transfer of control policies. Control calibrations proved sufficient to enable an initial demonstration of any-axis motion on hardware; however, some occurrences of stuck states and less smooth motion indicate room for more in-depth model identification. Faster and smoother motion, if feasible on hardware actuators, would require more accurate parameter values such as cable elasticity and coefficients of friction. These values might ideally be determined within the simulator testbed via Bayesian optimization, using a loss function of trajectory error relative to physical experiments under identical commands [19]. Finally, additional robustness in the bottom face detection might be obtainable by careful introduction of simulated noise into the symmetry-reduced training pipeline.

ACKNOWLEDGMENT: Supported by NASA ECF grant NNX15AU47G and NSF awards 1734492 and 1723868.

References

1. NASA Tensegrity Robotics Toolkit. ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/ntrt
2. Bliss, T., Iwasaki, T., Bart-Smith, H.: Central Pattern Generator Control of a Tensegrity Swimmer. Trans. on Mech. **18**(2) (2013)
3. Iscen, A., Caluwaerts, K., Bruce, J., Agogino, A., SunSpiral, V., Tumer, K.: Learning tensegrity locomotion using open-loop control signals and coevolutionary algorithms. Artificial Life **21**(2), 119–140 (2015)
4. Khazanov, M., Jocque, J., Rieffel, J.: Developing Morphological Computation in Tensegrity Robots for Controllable Actuation. In: 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 1049–1052. ACM, New York, NY, USA (2014). DOI 10.1145/2598394.2605680
5. Kim, K., Agogino, A.K., Toghyan, A., Moon, D., Taneja, L., Agogino, A.M.: Robust learning of tensegrity robot control for locomotion through form-finding. In: IROS (2015)
6. Koizumi, Y., Shibata, M., Hirai, S.: Rolling tensegrity driven by pneumatic soft actuators. In: ICRA, pp. 1988–1993 (2012). DOI 10.1109/ICRA.2012.6224834
7. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research **17**(1), 1334–1373 (2016)
8. Littlefield, Z., Surovik, D., Wang, W., Bekris, K.E.: From quasi-static to kinodynamic planning for spherical tensegrity locomotion. In: International Symosium on Robotics Research (ISRR). Puerto Varas, Chile (2017)
9. Mirletz, B., Bhandal, P., Adams, R.D., Agogino, A.K., Quinn, R.D., SunSpiral, V.: Goal directed CPG based control for high DOF tensegrity spines traversing irregular terrain. Soft Robotics (2015)
10. Mirletz, B.T., Park, I.W., Quinn, R.D., SunSpiral, V.: Towards bridging the reality gap between tensegrity simulation and robotic hardware. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 5357–5363. IEEE (2015)
11. Montgomery, W., Ajay, A., Finn, C., Abbeel, P., Levine, S.: Reset-free guided policy search: Efficient deep reinforcement learning with stochastic initial states. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3373–3380 (2017). DOI 10.1109/ICRA.2017.7989383
12. Paul, C., Valero-Cuevas, F.J., Lipson, H.: Design and control of tensegrity robots for locomotion. TRO **22**(5) (2006). DOI 10.1109/TRO.2006.878980
13. Rovira, A.G., Mirats Tur, J.M.: Control and Simulation of a Tensegrity-based Mobile Robot. RAS **57**(5), 526–535 (2009)
14. Sultan, C., Skelton, R.: Deployment of tensegrity structures. International Journal of Solids and Structures **40**(18), 4637–4657 (2003). DOI 10.1016/S0020-7683(03)00267-1
15. Surovik, D.A., Bekris, K.E.: Symmetric reduction of tensegrity rover dynamics for efficient data-driven control. In: ASCE International Conference on Engineering, Science, Construction and Operations in Challenging Environments (2018)
16. Vespignani, M., Ercolani, C., Friesen, J.M., Bruce, J.: Steerable locomotion controller for six-strut icosahedral tensegrity robots. In: IROS (2018)
17. Vespignani, M., Friesen, J.M., SunSpiral, V., Bruce, J.: Design of superball v2, a compliant tensegrity robot for absorbing large impacts. In: IROS (2018)
18. Zhang, M., Geng, X., Bruce, J., Caluwaerts, K., Vespignani, M., SunSpiral, V., Abbeel, P., Levine, S.: Deep reinforcement learning for tensegrity robot locomotion. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 634–641 (2017). DOI 10.1109/ICRA.2017.7989079
19. Zhu, S., Kimmel, A., Bekris, K.E., Boularias, A.: Fast model identification via physics engines for data-efficient policy search. In: International Joint Conference on Artificial Intelligence (IJCAI). Stockholm, Sweden (2018)