6N-DoF Pose Tracking for Tensegrity Robots

Shiyang Lu[1], William R. Johnson III[2], Kun Wang[1], Xiaonan Huang[2], Joran Booth[2], Rebecca Kramer-Bottiglio[2], Kostas Bekris[1]

[1] Rutgers University, New Brunswick, NJ, USA, ? ? ?
[2] Yale University, New Haven, CT, USA y

Abstract. Tensegrity robots, which are composed of compressive elements (rods) and flexible tensile elements (e.g., cables), have a variety of advantages, including flexibility, low weight, and resistance to mechanical impact. Nevertheless, the hybrid soft-rigid nature of these robots also complicates the ability to localize and track their state. This work aims to address what has been recognized as a grand challenge in this domain, i.e., the state estimation of tensegrity robots through a marker-less, vision-based method, as well as novel, on-board sensors that can measure the length of the robot's cables. In particular, an iterative optimization process is proposed to track the 6-DoF pose of each rigid element of a tensegrity robot from an RGB-D video as well as endcap distance measurements from the cable sensors. To ensure that the pose estimates of rigid elements are physically feasible, i.e., they are not resulting in collisions between rods or with the environment, physical constraints are introduced during the optimization. Real-world experiments are performed with a 3-bar tensegrity robot, which performs locomotion gaits. Given ground truth data from a motion capture system, the proposed method achieves less than 1 cm translation error and 3 degrees rotation error, which significantly outperforms alternatives. At the same time, the approach can provide accurate pose estimation throughout the robot's motion, while motion capture often fails due to occlusions.

Keywords: Robot Perception, Soft Robotics

1 Introduction

Tensegrity robots are lightweight, deformable platforms that are composed of compressive elements (rods) and flexible tensile elements (e.g., cables [29], springs [14], elastic membranes [1], etc.). This enables tensegrity robots to distribute external forces through deformation and thus prevent damage. Furthermore, by controlling its tensile elements, a tensegrity robot can reshape itself and move in complex environments [29]. An illustration of a 3-bar tensegrity robot, used for the experiments presented in this work, is shown in Fig. 1. Tensegrity robots' adaptability and compliance have motivated their use in many applications, such as manipulation [16], locomotion [22], morphing airfoil [8] and spacecraft landers [5]. While their versatility is exciting, tensegrity robots are difficult to track, model and control because of their complex dynamics [24].

In particular, state estimation and tracking are critical for control algorithms that require the current robot state. For tasks, such as bringing the tensegrity to a desired goal configuration, it is necessary to identify the 6D pose of its rigid elements. Reconstructing the trajectory of a tensegrity robot can also be used for sim2real, where the

? ? ? fshiyang.lu, kun.wang2012, kostas.bekrisg@rutgers.edu
y fwill.johnson, xiaonan.huang, joran.booth, rebecca.kramerg@yale.edu

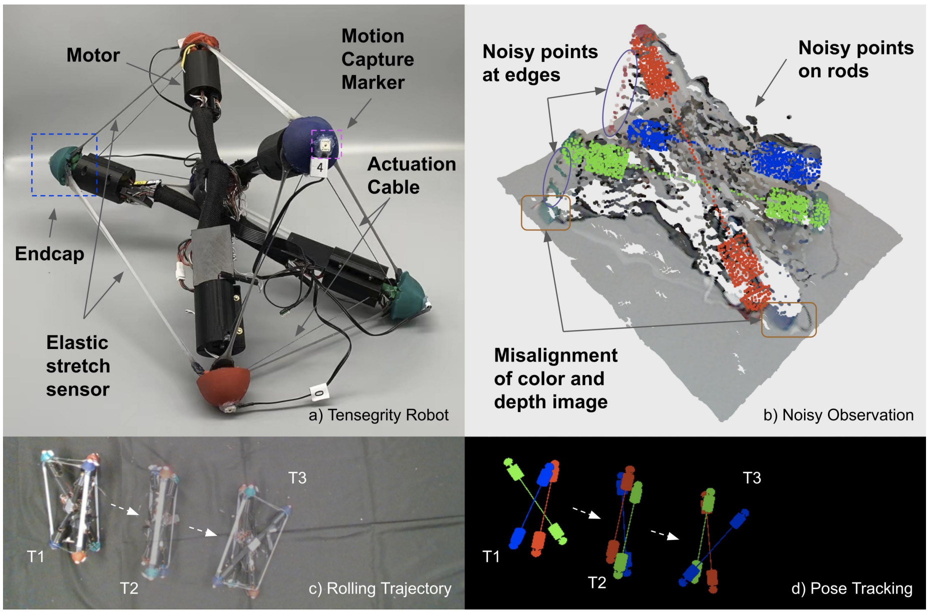

Fig. 1: (a) The 3-bar tensegrity robot used in the accompanying experiments. Each bar has a length of 36 cm. The robot can reshape itself and perform locomotion by changing the length of its actuation cables. Distances between endcaps are measured by on-board elastic stretch sen-sors. Motion capture markers are used to generate ground truth poses only for the purpose of evaluation. (b) Noisy observations from a commodity RGB-D camera pose challenges to vision-based pose estimation. (c) The tensegrity robot rolling on a carpet. The shape of this robot keeps changing during the motion and generates significant self-occlusions. (d) 6-DoF pose estimates for each bar at the corresponding time steps.

parameters of a simulated model are identified to match real-world trajectories. Such simulated models [19, 30–32] can then be used to generate locomotion policies via RL [25, 26] without the need for many real world experiments.

Accurately estimating the state of a tensegrity robot, i.e., the 6-DoF pose of each rod, remains a grand challenge in this domain [24]. Vision-based pose estimation is especially challenging due to noisy observations, self-occlusions, and a large number of degrees of freedom as shown in Fig. 1. This drives the need for improved sensor fusion, physics-aware state estimation algorithms, and more stable, on-board sensors [24].

This work proposes a robust approach for markerless, vision-based pose tracking of an N-bar tensegrity robot given as input: 1) RGB-D images from a single, external commodity camera, and 2) measured endcap distances from recently developed, on-board, elastic stretch sensors. The proposed approach has the following key features:

An ICP-like tracking method is used to fuse the multi-modal inputs in an iterative optimization process. An objective function is designed to maximize both unary and binary potentials. Unary potentials reflect how well the mesh models of the rods' endcaps fit with the observed point cloud. Binary potentials reflect the fitness of predicted endcap distances and the measured ones from on-board sensors. Adaptive weights are designed to balance these two potentials.

- A variety of physical constraints between rigid elements of tensegrity robots are introduced in the optimization. These constraints ensure that the estimated poses are physically feasible without relying on a sophisticated physics engine and further improve pose estimation accuracy.

The proposed method is evaluated qualitatively and quantitatively on real-world experiments discussed in section 5, using a 3-bar tensegrity performing locomotion gaits. The method achieves low translation and rotation error when compared against alternatives as well as ground truth data from motion capture. More critically, the approach can provide pose estimates throughout the robot's motion, while motion capture often fails due to occlusions. A link[3] points to a website that highlights the available experiments and shares an open-source dataset with the corresponding trajectories. The hope is that the methodology will prove useful for additional platforms beyond tensegrities that can make use of on-board distance sensors for soft elements and an external RGB-D camera.

2 Related Work

The majority of state estimation work on tensegrity robots is based on multiple on-board sensors, such as sensors for contact sensing [7], cable length measurement [28], strut extension [10], and inertial measurement units (IMUs) [4, 7]. However, with on-board sensors only, it is hard to reconstruct the state of tensegrity robots in the global frame. To estimate states in the global frame, Caluwaerts et.al [6] proposed a variant of the Unscented Kalman Filter (UKF) algorithm by fusing the measurements from IMU sensors and ranging sensors that are placed at the endcaps of rods and fixed anchors (landmarks) at the testing area. Similarly, a motion capture system could be setup and track the markers placed on endcaps of rods [3] to get their global coordinates. Despite being accurate when all the markers are visible, tracking often fails when markers are occluded. Moreover, since a motion capture system is fixed once calibrated, it constrains the movement of a robot to a small region.

More recently, Moldagalieva et al. [17] proposed the first vision-based method to estimate the pose of a tensegrity robot using fiducial markers. In particular, a hemispherical camera module was attached to the base of their robot, so as to track the fiducial markers printed on a triangle-shaped plate fixed at the upper part of the robot. The pose estimates could be further fine-tuned by fusing the cable length measurement using a neural network [15]. However, the requirement for fiducial markers to always be visible to the camera prevents the robot from doing actions like rolling. It also requires a large amount of data for training a deep model. In contrast, the method proposed in this paper detects endcaps of rods based on color images instead of relying on fiducial markers.

ICP algorithms [2, 9] have been widely used in the industry for point cloud registration and model matching. A large number of variants [20, 21, 23] have been proposed since the advent of this algorithm for better accuracy and faster convergence. Some extensions of the ICP algorithm can even be applied to track articulated objects, such as human bodies [18] and hands [27]. This proposed work is in line with the direc-tion of those methods but with an objective function and physical constraints that are specifically designed for tensegrity robots.

[3] https://sites.google.com/view/tensegrity-robot-perception

3 Problem Formulation

This work aims to track the pose of an N-bar tensegrity robot, i.e. 6-DoF pose of each rigid element (rod) of that robot, which are connected with flexible tensile elements (cables). An illustration of the robot used for the experimental tests of this paper is shown in Fig. 1. Two types of sensors are considered: 1. An overhead RGB-D camera records the motion of the tensegrity robot; 2. on-board strain sensors are publishing the cable length measurements during the motion. The images and cable length measurements are synchronized. The mesh model M of the rods and endcaps is assumed available, and the structure of the robot (i.e., the connectivity between endcaps with cables) is also provided. The endcaps of different rods are painted with different colors, and the ranges of their HSV values are known. Then, the objective is to compute the set of poses $P_t = fp^1_t; :::; p^N_t g 2 SE(3)$ of each rod at the current time step t, given:

known parameters for the setup, i.e., the camera's intrinsic matrix C, the tensegrity's rod length l_{rod} and diameter d_{rod},

previous rod pose estimates at step t 1, i.e., $P_{t\ 1} = fp^1_{t\ 1}; :::; p^N_{t\ 1} g 2 SE(3)$,

current measurements at time step t:

– the input of an RGB-D image I_t observing the tensegrity,
– cable length measurements $L_t = fl^{ij}_t j(i; j) 2 Eg$ where $i; j$ are endcap indices and $(i; j) 2 E$, if endcap i and endcap j are connected by a cable.

To initialize the process, a 2D Region of Interest (RoI), in the form of a bounding box, is manually defined for each endcap at the initial frame I_0 of the video sequence.

4 Method and Implementation

The following discussion describes step by step how the poses P_t are estimated at each time step. In terms of the visual data and given that the depth measurements of rods can often be very noisy, as shown in Fig. 1b, only the observations of endcaps are used in the proposed state estimation process.

4.1 Pose Initialization

The proposed tracking algorithm first establishes an initial pose estimate P_{init} given the manually defined image RoIs. This initial estimate does not have to be very accurate since it will be optimized by consecutive steps of the iterative process. The foreground pixels in each RoI are segmented based on the range of HSV values for that endcap, and then projected to 3D points given the camera intrinsic matrix C and the depth information. The centroid of the observed point cloud of each endcap is computed and used as a rough estimate of the initial endcap position. The initial position of each rod is then estimated by simply using the center of its two endcaps. The initial rotation is estimated using axis-angle representation where the axis is aligned with the direction from one endcap to the other and the angle is arbitrarily set due to rotational symmetry.

4.2 Iterative Pose Tracking

The proposed method addresses the pose tracking problem through an iterative optimization process where each iteration is composed of a transition step and a correction step. Those two steps are performed alternately until convergence or a preset maximum number of iterations is reached (an empirical number is 6). In the transition step, only the RGB-D observation is used to estimate a local transformation from the previous time step. While in the correction step, a joint constrained optimization is performed given the input of estimated endcap positions and cable length measurements. Those two steps are detailed below.

1) Transition Step

The transition model, also known as the action model, aims to provide a 6-DoF local transformation P^i_t for rods $i = 1; 2; \ldots; N$, such that the new pose estimate for each rod to be $\hat{P}^i_t = P^i_t \hat{P}^i_{t-1}$. It is, however, not easy to compute this transformation directly from control gaits, due to the complex dynamics of tensegrity robots. Instead, the local transformation is estimated via scan matching, i.e., the registration of endcap model points with the observed points in the RGB-D data. To prevent unobserved model points (e.g., those on the backside of an endcap) from being registered with the point cloud, this work adopts a hidden point removal method [13] so that only visible points on the model will be used for registration and thus improve accuracy.

Correspondence between endcap model points and observed points is established based on nearest neighbors as long as they are within a maximum distance threshold in the 3D space and also within an HSV range in the color space. The maximum dis-tance threshold d_{max} is designed to be a dynamic value that decreases with the number of iterations. The intuition behind this is that in the first few iterations the region for correspondence search should be large in case of significant movement, while in later iterations it should be small for fine-grained estimates. With known correspondences, a local transformation for each rod can be estimated with the Kabsch algorithm [12], which minimizes the weighted root mean square error (RMSE) between point pairs. The weight of each point pair is computed using the following equation, where d is the distance between a pair of points. This weight is monotonically decreasing when the distance increases and clipped to zero if the distance goes beyond d_{max}.

$$w(d) = \begin{cases} 1 - (\frac{d}{d_{max}})^2; & d \leq d_{max} \\ 0; & d > d_{max} \end{cases} \quad (1)$$

It is often the case, however, that the corresponding points of one endcap are too few for registration, mainly due to occlusions. To handle this problem, a certain number of dummy points (an empirical number is 50) are randomly sampled from observed and model points at the previous time step and pasted to the points in the current time step. This step is effective in preventing arbitrary poses being generated and results in more stable state estimates. It is also arguably better than alternative motion models, such as assuming the rod is not moving at the current time step since the observation of the other endcap can provide meaningful information. The weights of those dummy points are half the value of observed points.

2) Correction Step

In the correction step, a joint constrained optimization is performed. The intuition behind this optimization is that the estimated endcap positions are expected to be close to the output of the transition model, and the distance between two connected endcaps should be close to the cable length measurement. Meanwhile, endcaps and rods should satisfy a variety of physical constraints to ensure that the estimated 6-DoF poses are valid. Given the current pose estimates $\hat{P}_t = \{\hat{p}^1_t; \ldots; \hat{p}^N_t\}$ from the transition step and cable length measurements L_t, the objective function shown below is aiming to find a set of optimal endcap positions $Q = \{q_t^1, \dots, q_t^{2N}\}$:

$$L(Q) = \sum_{i}^{2N} w_i ||q_t^i - \hat{q}_t^i||_2^2 + \sum_{(i,j)\in E} w_{ij}(||q_t^i - q_t^j|| - l_t^{ij})^2$$
$$Q_{optim} = \underset{Q}{\operatorname{argmin}}\, L(Q) \tag{2}$$

where $\hat{Q} = \{\hat{q}_t^1, \dots, \hat{q}_t^{2N}\}$ are estimated endcap positions from the transition model and w_i, w_{ij} are adaptive weights for the unary and binary loss terms respectively. The constraints that are accompanying this objective function and computation of those dynamic weights are elaborated upon in subsections 4.3 and 4.4 respectively.

Once the positions of endcaps are optimized, the current pose estimates are updated for the next iteration such that the position of the rod is set to be the center of its endcaps and the rotation axis is set to be aligned with the direction from one endcap the other. Different from the initialization step in subsection 4.1, the rotation angle is no longer set to be arbitrary but rather to a value that minimizes the geodesic distance in $SO(3)$ from the previously estimated rotation. Despite rotational symmetry, an arbitrary rotation angle can affect downstream tasks that need to compute angular velocity, such as feedback control and system identification. This constraint optimization is performed with sequential least squares programming (SLSQP).

4.3 Physical Constraints

1) Constraints between endcaps

Rods are rigid elements of tensegrity robots. The distance between the estimated endcap positions of a rod should be a constant, i.e., the rod length l_{rod}, regardless of time. Assuming that i, j are the two endcaps of a rod, an equality constraint is applied:

$$||q_i - q_j|| = l_{rod}$$

2) Constraints between rods

The estimated poses of rods should not make them collide or even penetrate each other between consecutive frames. Thanks to the simple geometry of the rods, the physical constraints between them can be added to the optimization problem without using a sophisticated simulation engine.

Each rod is approximated as a cylinder and its endcaps as two hemispheres. With this approximation, a rod can be parameterized with only two variables: its length l_{rod} and its diameter d_{rod}. A collision between rod i and rod j can be easily detected by comparing the rod diameter d_{rod} with the rod distance $d(c_t^i, c_t^j)$ where (c_t^i, c_t^j) is the closest pair of points on the main axis of rod i and rod j at time step t. Rod i and rod j are colliding with each other if $d(c_t^i, c_t^j) < d_{rod}$. The closest pair of points can be computed given the rod pose estimates at timestep t. An inequality constraint can be added, i.e.

$$|c_t^i c_t^j| \geq d_{rod} \tag{a}$$

Meanwhile, the relative poses between two rods are not expected to change dramatically, and it is impossible for one rod to suddenly cross through another between two consecutive frames. An illustration of penetration between two rods is shown in Fig. 7.

Such penetration can be detected if the angle between $c_t^i c_t^j$ and $c_{t-1}^i c_{t-1}^j$ is greater than

90 degrees, which can be written as $\overset{!}{c^i_t c^j_t}\;\; c^i_{t}\;\; \overset{!}{c^j_{t\;\;1}} < 0$. Thus, another inequality constraint can be added, i.e.,

$$\overset{!}{c^i_t c^j_t}\;\; c^i_{t\;\;1} c^j_{t\;\;1}\;\; 0 \tag{b}$$

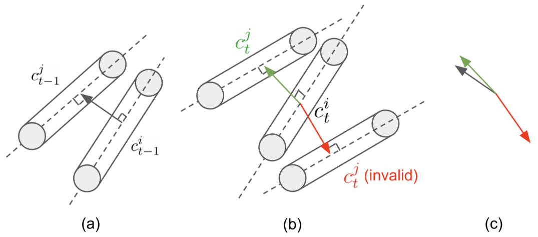

Fig. 2: Illustration of the physical constraints between rods. (a) Rod i and rod j at time step t 1, where the ($c^i_{t\;\;1}$; $c^j_{t\;\;1}$) is the closest pair of points. (b) Rod i and rod j with two pose hypotheses at time step t. Their distance should satisfy the constraint $j\overset{!}{c^i_t c^j_t}j\;\; d_{rod}$. (c) Despite that both hypotheses for rod j satisfy the distance constraint, the one in red is physically infeasible as the angle between $\overset{!}{c^i_t c^j_t}$ and $\overset{!}{c^i_{t\;\;1} c^j_{t\;\;1}}$ is greater than 90 degrees, which means rod j crosses through rod i between two consecutive frames.

Nevertheless, it is difficulty to directly apply constraints (a) and (b) despite their simple mathematical formulas. This is because the closest pair of points (c^i_t; c^j_t) depends on the relative pose of two rods, and their positions are non-linear with respect to the endcap positions. This problem is addressed by introducing a linear approximation of the constraints based on the assumption that the change in the closest points on the two rods is negligible in the rod's local frame between two consecutive frames, and the angle between $\overset{!}{c^i_t c^j_t}$ and $c^i_{t\;\;1} c^j_{t\;\;1}$ is small, i.e. cos() 1. With this assumption, constraints (a) and (b) can be approximated with a single constraint linear to the current endcap positions (c), i.e.,

$$\overset{!}{\hat{c}^i_t \hat{c}^j_t}\;\; \frac{\overset{!}{c^i_{t\;1} c^j_{t\;1}}}{j\overset{!}{c^i_{t\;1} c^j_{t\;1}}j}\; d_{rod} \tag{c}$$

where $\hat{c}^i_t$ and $\hat{c}^j_t$ are the same points as $c^i_{t\;\;1}$ and $c^j_{t\;\;1}$ in the local frame of rod i and rod j. This linear approximation makes it easier to optimize, and it is shown to be effective in experiments. If ($c^i_{t\;\;1}$; $c^j_{t\;\;1}$) are the endpoints of rod i and rod j, then this inequality constraint is not applied.

3) Constraints between rods and the supporting plane

If given the prior knowledge that the tensegrity robot is not moving in a complex environment but instead on the ground, an additional physical constraint can be applied between the rods and the ground to prevent invalid poses that make rods penetrate the ground. A extrinsic transformation from the camera frame to the world frame, which is composed of a rotation R and a translation t, can be computed using plane detection by RANSAC, where the ground is represented as z = 0 in the world frame. The inequality constraint for endcap positions at position q^i_t is:

$$z(Rq^i_t + t)\;\; d_{rod}=2; \quad i = 1; 2; ::; 2N$$

4.4 Adaptive Weights

Adaptive weights are applied to balance the unary loss and binary loss in the objective function in the correction step. Let the ratio r_i for the endcap i be the number of corresponded points over the total number of rendered points for endcap i assuming no occlusion. The unary weight w_i and binary weights w_{ij} are computed as follows:

$$w_i = \max(r_i;\ 0:1)$$

$$w_{ij} = \begin{cases} 0; & r_i > 0:5 \text{ and } r_j > 0:5 \\ 0:25; & r_i < 0:2 \text{ or } r_j < 0:2 \\ 0:25\ (1 \quad 0:5(r_i + r_j)); & \text{otherwise} \end{cases}$$

The design logic is that endcap distance measurements by on-board elastic stretch sensors are often not accurate enough and may enlarge error when both the end caps of a rod are visible, despite that these measurements are rather helpful when endcaps are occluded. When the number of corresponding points for both endcaps of a rod is large, i.e. $r_i > 0:5$ and $r_j > 0:5$, point registration is likely to give an accurate estimate, and the loss introduced by the distance measurement is not considered. However, when either of the endcaps on a rod is heavily occluded, i.e. $r_i < 0:2$ or $r_j < 0:2$, then the distance measurement must be taken into account and is set to be a constant value (0.25). The binary weights are otherwise linear to the sum of the ratio r_i. These coefficients and thresholds are empirical, but they work sufficiently well in the experiments.

4.5 Noise Filtering

Sensor noise in modern Lidar-based RGB-D cameras is typically small. Nevertheless, it is observed in real experiments that misalignment of the color and depth images does occur, and the depth quality is bad at the object edges. An example is shown in Fig. 1(b). This noise is usually not a big issue for target objects with a large observable region, but it turns out to be rather problematic for tensegrity robots as the endcaps are quite small (radius = 1.75 cm in the experiment). Although this noise can be partially removed by careful tuning of the camera setup, it is sensitive to many uncontrollable factors, such as lighting conditions in real environments. To minimize the sensor noise, observed points for each endcap are filtered given the prior knowledge of the endcap size. To be more specific, observed points are first filtered by the HSV range in color space and max correspondence distance in 3D space. Noisy points are further removed if their depths are greater than the minimum depth of the remaining points plus the endcap radius.

5 Experiments

5.1 Hardware Setup

The real-world data are collected on a 3-bar tensegrity robot with nine elastic stretch sensors connecting the nodes. The length of of each bar is 36 cm. Six of the sensors run parallel to the six cables that actuate the robot while the other three longer sensors are passive. The sensors are made from silicone elastomers (Eco-Flex 30 or DragonSkin 20; Smooth-On, Inc.) encapsulating two layers of a liquid metal paste made from eutectic gallium indium. The two layers of liquid metal compose a parallel plate capacitor whose capacitance varies linearly as the sensor is stretched [11]. The capacitance of each sensor is measured on-board with a commercial capacitive sensing breakout board

(MPR121; Adafruit). The sensors are calibrated individually to obtain a linear model mapping the sensor's capacitance to its length. Then, the sensors are used for feedback control during actuation trajectories, and the length of each sensor is published at each time step. The color and depth image sequences are collected by an RGB-D camera (Intel Realsense L515) at 720p resolution. In order to get the ground truth poses for quantitative evaluation, a commercial motion capture system (Phasespace, Inc.) is used. Markers are placed on the endcaps of rods to measure their positions. Note that measurements from the motion capture system also contain some small error, but this error is difficult to quantify. Fig. 3 shows the hardware setup and the environment where the robot experiments are performed.

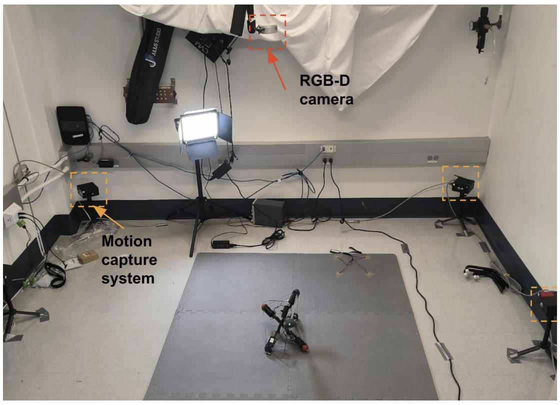

Fig. 3: Hardware setup of the experiments. An RGB-D camera (RealSense L515) is placed overhead to capture the whole region where the 3-bar tensegrity robot performs locomotion. A motion capture system (Phasespace, Inc.) with six cameras is placed around this region to capture endcap markers and generate ground truth for the purpose of evaluation.

5.2 Data Collection

The proposed algorithm is evaluated on a set of self-collected robot trajectories. In total, 16 unique trajectories are collected where the tensegrity robot is placed at different starting positions and a different set of controls are performed. For each trajectory, a sequence of RGB-D images and their corresponding cable length measurements are aligned and recorded at a frame rate around 10Hz. The shortest sequence contains 41 frames while the longest contains 130 frames. In total, this dataset contains 1524 frames, and about 95 frames on average for each trajectory. Note that markers placed on endcaps are not always visible to the motion capture system. The visibility of a marker from the camera and from the motion capture system is, however, different. An endcap occluded from the camera may have its marker visible to the motion capture system and vice versa. In total, the number of ground truth poses for all rods in all videos is 2786. Detailed statistics of the dataset are shown in Fig. 4.

5.3 Evaluation Metric

For quantitative evaluation of pose estimation, the mean and standard deviation of both translation and rotation errors are reported. Due to rotational symmetry along the main axis of rods, the rotation error is evaluated using the angle between the estimated and ground truth main axis, i.e. the pose errors of rods are evaluated on five degrees of

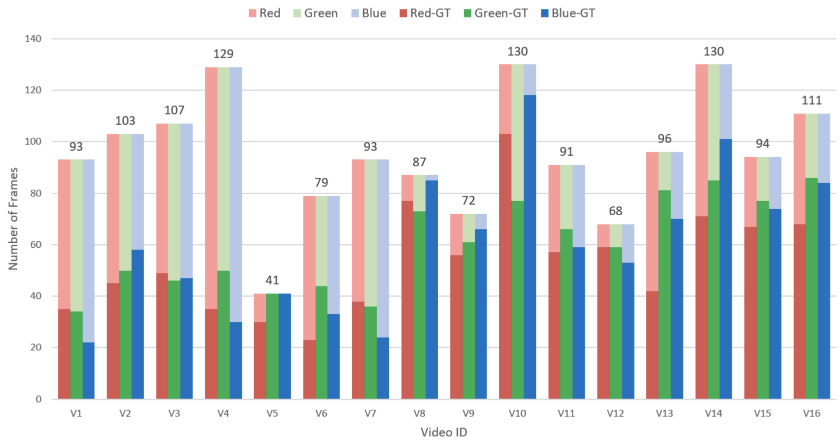

Fig. 4: Statistics of the dataset. The dataset contains 16 videos that are captured around 10Hz. This plot shows the number of frames in each video, and the number of available ground truth poses for rods with R/G/B endcaps respectively. The ground truth pose of a rod is only available when both of its endcap markers are visible to the motion capture system, which is not often the case in real experiments. This, however, motivates the proposed work.

freedom. Pose estimation results are also evaluated using the “2cm-5deg” metric, which reports the percentage of pose estimates that are within both 2cm translation error and 5 degree rotation error. The translation error of the center of mass is also evaluated, where the center of mass of the robot is computed by averaging the center positions of all rods. Apart from the pose estimation result in the global frame, the local shape reconstruction result is also evaluated. This result is correlated to the pose estimation result, but it is independent of coordinate frames. It is evaluated by comparing predicted distances between endcaps with measured distances from the motion capture system. Smaller distance error indicates a better shape reconstruction in the local frame.

5.4 Baselines and Ablation Studies

The first baseline method is naive ICP performed on each rod. Dummy points are added to improve stability because this baseline can easily lose track without them. However, cable length measurements are not used, and no joint optimization is performed. The second baseline first estimates the shape of the tensegrity robot using the endcap distance measurements from on-board sensors. A constrained optimization is done which minimizes the mean squared error (MSE) between the predicted distances between endcaps and the measured ones from on-board sensors, while it constrains the distance between the two endcaps of each rod as the rod length l_{rod}. The reconstructed robot is then considered as a rigid body and ICP is used for registration. The third baseline considers all the constraints in section 4.3, and performs a correction step similar to the proposed method. The correction step, however, is not integrated in the iterative process. Instead, it first performs ICP for a few iterations and then does one correction step at the end.

In addition, a set of ablation studies are done to verify the effectiveness of the proposed physical constraints and adaptive weights. The first ablation study removes the ground constraint and physical constraints between rods. The second ablation study only removes the physical constraints. The third ablation study keeps all constraints, but uses a static weight (0.25) instead of dynamic weights in the correction step. In addition to those ablation studies, the error of distance between endcaps, which is measured by the elastic stretch sensors, is reported. Note that all the hyperparameters, such as max correspondence distance thresholds, remain the same for all baselines and all testing trajectories for a fair comparison.

Method	Trans. Err (cm)	Rot. Err (deg)	2cm-5deg (%)	CoM Err (cm)	Shape Err (cm)
Baseline 1	1.56 / 1.89	4.55 / 7.96	73.1	1.07 / 0.76	1.30 / 1.81
Baseline 2	1.83 / 1.64	5.59 / 6.62	59.2	1.59 / 1.17	1.18 / 1.67
Baseline 3	1.10 / 0.75	3.23 / 2.91	79.2	0.81 / 0.44	0.92 / 0.91
Ablation 1	1.22 / 0.96	3.81 / 9.28	79.2	1.01 / 0.60	0.89 / 1.00
Ablation 2	1.04 / 0.69	3.46 / 9.10	82.3	0.79 / 0.43	0.85 / 0.80
Ablation 3	1.02 / 0.60	2.88 / 2.16	85.3	0.79 / 0.42	0.84 / 0.68
Measured	N/A	N/A	N/A	N/A	1.09 / 1.19
Proposed	0.99 / 0.58	2.84 / 2.12	85.5	0.77 / 0.44	0.80 / 0.64

Table 1: Quantitative results of pose tracking and local shape reconstruction.

5.5 Quantitative Results

Quantitative results of pose estimation and local shape reconstruction are shown in Table 1. The first baseline, naive ICP, has the worst performance in terms of local shape reconstruction, and the second largest tracking error in the global frame. Even though dummy points are added, it still loses tracking in multiple trials and is not able to recover. The second baseline has an even larger tracking error than the naive ICP, which may seem a little surprising. This is because the on-board sensors are often noisy, and the robot being treated as a rigid body after erroneous shape reconstruction further deteriorates the tracking performance. The third baseline performs much better by considering all the physical constraints and performing a correction step. However, ICP and the correction step are performed separately rather than in an interleaving way. Thus, it has an inferior performance than the proposed approach.

The first two ablation studies shows the benefits of the ground constraints and constraints between rods. Without both constraints, the result of the first ablation is not satisfactory despite being better than the naive ICP. With ground constraints, the translation error in the second ablation is much reduced but the rotation error remains large. The third ablation introduces constraints between rods, and significantly decreases the rotation error. The proposed method performs even better than the third ablation using dynamic weights in the joint optimization. Overall, the proposed method has a mean translation error within 1 cm, mean rotation error within 3 degrees, 85.5% of the rod pose estimates within 2cm-5deg error range, and mean center of mass error of 0.77 cm for the robot as a whole. It also has the smallest shape reconstruction error in the local frame, which is reflected by the mean distance error of 0.8 cm between endcaps.

5.6 Qualitative Results

An exemplary trial is shown in Fig. 5, where a robot is moving on a lawn, and the motion capture system is not available. Despite noisy background and self-occlusions, the proposed method is able to give highly accurate pose estimates, while the estimates from the naive ICP baseline have notable errors. In Fig. 6, distances between endcaps

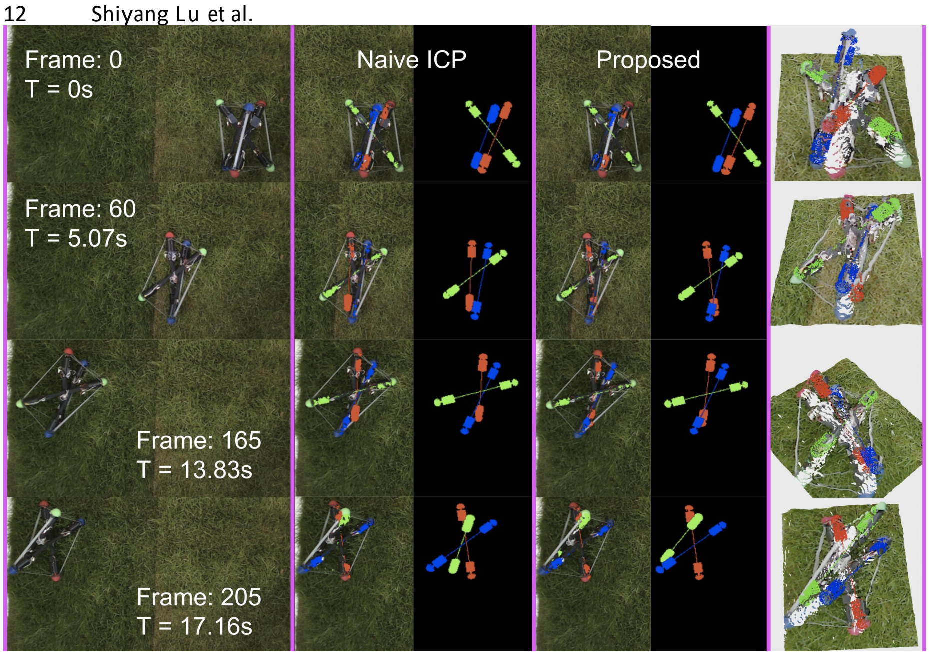

Fig. 5: Qualitative results. In this exemplary trial, the tensegrity robot moves from the bottom right to the top left. The left most column shows the raw RGB observation. The second and third columns show the rendered robot at the estimated poses by the naive ICP baseline and the proposed method respectively. The last column show the colored point cloud and robot estimated by the proposed method from some other viewpoints. The proposed method has superior performance over the naive ICP baseline especially when notable occlusion occurs.

at each frame is plotted for another exemplary trial. This is to show how the local shape reconstruction matches with the ground truth. Those predicted distances are compared with the measured distances from the elastic stretch sensors and the motion capture system (ground truth). The predicted results have smaller mean distance error (0.8cm) than those directly measured by the elastic stretch sensors (1.2cm). Some regions are highlighted in purple boxes to show the differences. The regions where the elastic stretch sensors have large errors are typically due to the sensors going slack. The proposed method, however, is robust to those noisy measurements. Fig. 7 shows the effectiveness of physical constraints via another example. Without physical constraint, rods can freely cross through each other. In this example, the red rod is above the green rod and blue rod in the first frame, but it's estimated to be underneath them in the next frame, which is physically infeasible. The blue and green rods are also colliding with each other in this estimate. The proposed physical constraint can effectively prevent such unrealistic estimates from being generated, further improving the accuracy.

5.7 Running Time

The running time is computed on a desktop with an AMD Ryzen 5900 CPU. For each iteration, the transition step takes 2ms, and the correction step takes 4ms. The method can achieve real-time performance (30Hz) with 6 iterations at each time step. Note that the accompanying implementation is written in Python, and a faster speed is expected with a more sophisticated implementation in C++.

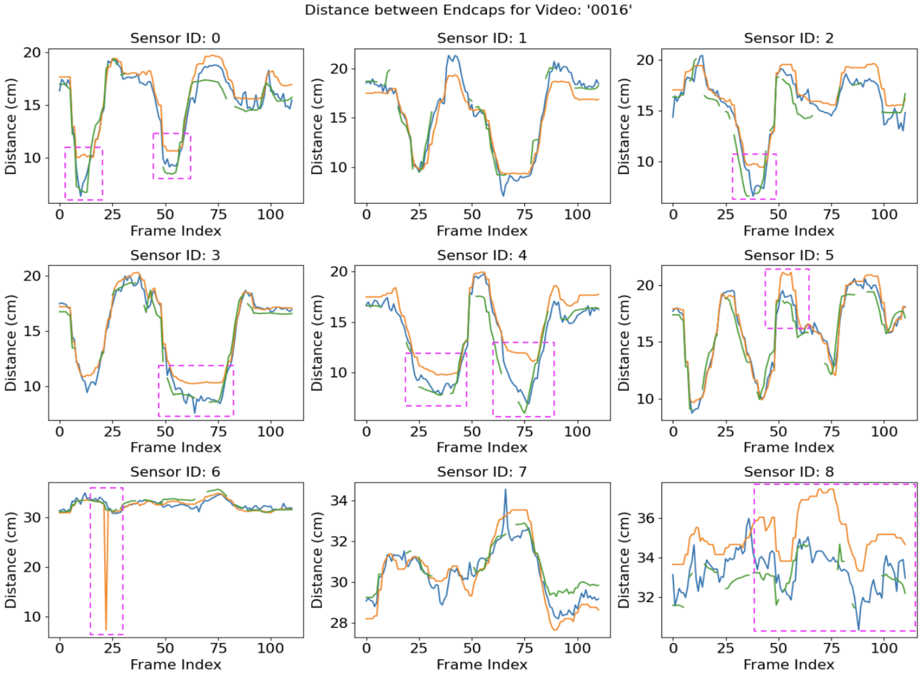

Fig. 6: Plots of distances between endcaps. These plots reflect the quality of local shape reconstruction in this exemplary trial. The predicted distances between endcaps by the proposed method (blue lines) are compared with those measured by on-board sensors (yellow lines) and by the motion capture system (green lines, used as ground truth). The predicted distances have even smaller mean error (0.8cm) than that from on-board sensors (1.2cm), which reflects a better shape reconstruction in the local frame. Some notable differences are highlighted in purpose boxes.

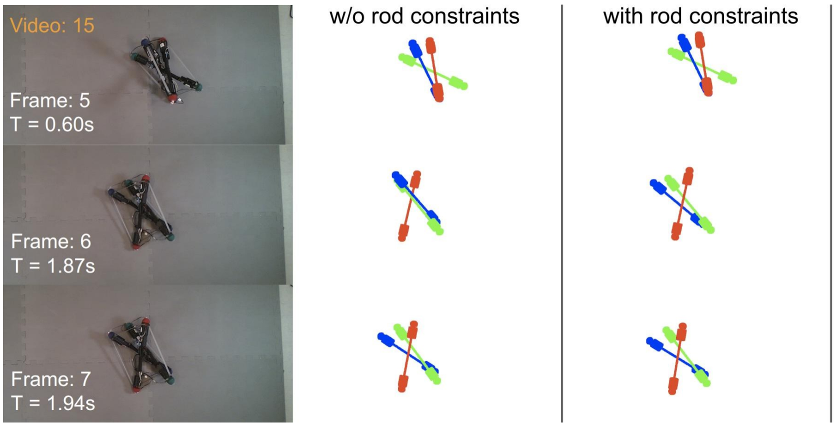

Fig. 7: Effectiveness of constraints between rods. This figure compares the pose estimates with and without rod constraints at certain frames. Large error occurs at frame 6 without using rod constraints. The rod with red endcaps crosses through the other two from the previous frame, which is physically infeasible. The blue and green ones are also colliding with each other. As a comparison, the estimates with rod constraints are physical valid and are more accurate.

6 Limitations

One limitation of this work is that manual annotation of the 2D region of interest is required for the first frame in order to compute an initial pose. Besides, tracking failure does happen at some time step despite being robust for most of the time. A failure case is shown in Fig. 8 where the movement of the robot is significant between two consecutive frames. Although the tensegrity robot is not able to move at a high speed, data transmission via Bluetooth can be unstable sometimes. After investigation, it turns out that not enough iterations were performed before the max correspondence distance d_{max} decreases to a value that excludes the observed points. This could be resolved by running more iterations. It is, however, tricky to set such hyperparameters, and there is a trade-off between the number of iterations and the running speed. This work is not data-driven and some manual tuning of these hyperparameters is still required.

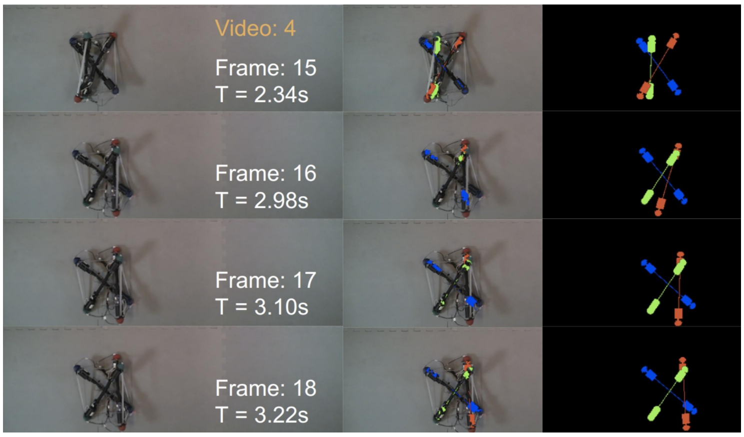

Fig. 8: Failure case. A large tracking error occurs at the frame 16 of this video. The robot has a significant movement between two consecutive frames and not enough iterations were performed to make the estimates converge. Tracking is fully recovered later at frame 18.

7 Conclusion

This work proposes a novel iterative algorithm for 6N-DoF pose tracking of N-bar tensegrity robots. Compared to the standard ICP algorithm for rigid objects, the proposed method fuses the cable length measurement as a soft constraint in the joint optimization process. Meanwhile, a set of physical constraints are introduced to prevent physically invalid pose estimates from being generated. A number of techniques are further applied to improve the robustness of pose estimation against sensor noise. Real-world experiments are performed with an exemplary three-bar tensegrity robot to showcase the robustness and effectiveness of the proposed tracking method, which has a rather small pose estimation error of 1cm and 3 degrees for each rod on average. One possible future work is global registration of tensegrity robots, so that the requirement of manual annotation at the initial frame can be removed, and tracking failures can be easily recovered.

References

1. R. L. Baines, J. W. Booth, and R. Kramer-Bottiglio. Rolling soft membrane-driven tensegrity robots. IEEE Robotics and Automation Letters, 5(4):6567–6574, 2020.
2. P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. Spie, 1992.
3. J. W. Booth, O. Cyr-Choiniere, J. C. Case, D. Shah, M. C. Yuen, and R. Kramer-Bottiglio. Surface actuation and sensing of a tensegrity structure using robotic skins. Soft Robotics, 8(5):531–541, 2021.
4. J. Bruce, K. Caluwaerts, A. Iscen, A. P. Sabelhaus, and V. SunSpiral. Design and evolution of a modular tensegrity robot platform. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 3483–3489. IEEE, 2014.
5. J. Bruce, A. P. Sabelhaus, Y. Chen, D. Lu, K. Morse, S. Milam, K. Caluwaerts, A. M. Agogino, and V. SunSpiral. Superball: Exploring tensegrities for planetary probes. 12th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), 2014.
6. K. Caluwaerts, J. Bruce, J. M. Friesen, and V. SunSpiral. State estimation for tensegrity robots. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1860–1865. IEEE, 2016.
7. K. Caluwaerts, J. Despraz, A. Işçen, A. P. Sabelhaus, J. Bruce, B. Schrauwen, and V. SunSpiral. Design and control of compliant tensegrity robots through simulation and hardware validation. Journal of the royal society interface, 11(98):20140520, 2014.
8. M. Chen, J. Liu, and R. E. Skelton. Design and control of tensegrity morphing airfoils. Mechanics Research Communications, 103:103480, 2020.
9. Y. Chen and G. Medioni. Object modelling by registration of multiple range images. Image and vision computing, 10(3):145–155, 1992.
10. J. Friesen, A. Pogue, T. Bewley, M. de Oliveira, R. Skelton, and V. Sunspiral. Ductt: A tensegrity robot for exploring duct systems. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 4222–4228. IEEE, 2014.
11. W. R. Johnson, J. Booth, and R. Kramer-Bottiglio. Integrated sensing in robotic skin modules. In 2021 IEEE Sensors, pages 1–4. IEEE, 2021.
12. W. Kabsch. A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography, 32(5):922–923, 1976.
13. S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. In ACM SIGGRAPH 2007 papers, pages 24–es. 2007.
14. K. Kim, D. Moon, J. Y. Bin, and A. M. Agogino. Design of a spherical tensegrity robot for dynamic locomotion. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 450–455. IEEE, 2017.
15. A. Kuzdeuov, M. Rubagotti, and H. A. Varol. Neural network augmented sensor fusion for pose estimation of tensegrity manipulators. IEEE Sensors Journal, 20(7):3655–3666, 2020.
16. S. Lessard, D. Castro, W. Asper, S. D. Chopra, L. B. Baltaxe-Admony, M. Teodorescu, V. SunSpiral, and A. Agogino. A bio-inspired tensegrity manipulator with multi-dof, structurally compliant joints. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5515–5520. IEEE, 2016.
17. A. Moldagalieva, D. Fadeyev, A. Kuzdeuov, V. Khan, B. Alimzhanov, and H. A. Varol. Computer vision-based pose estimation of tensegrity robots using fiducial markers. In 2019 IEEE/SICE International Symposium on System Integration (SII), pages 478–483, 2019.
18. L. Mundermann, S. Corazza, and T. P. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In 2007 IEEE conference on computer vision and pattern recognition, pages 1–6. IEEE, 2007.

19. NASA. NASA Tensegrity Robotics Toolkit, Accessed 2020. https://github.com/NASA-Tensegrity-Robotics-Toolkit/NTRTsim.
20. J. Park, Q.-Y. Zhou, and V. Koltun. Colored point cloud registration revisited. In Proceedings of the IEEE international conference on computer vision, pages 143–152, 2017.
21. S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In Proceedings third international conference on 3-D digital imaging and modeling, pages 145–152. IEEE, 2001.
22. A. P. Sabelhaus, L. J. van Vuuren, A. Joshi, E. Zhu, H. J. Garnier, K. A. Sover, J. Navarro, A. K. Agogino, and A. M. Agogino. Design, simulation, and testing of a flexible actuated spine for quadruped robots. arXiv preprint arXiv:1804.06527, 2018.
23. A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In Robotics: science and systems, volume 2, page 435. Seattle, WA, 2009.
24. D. S. Shah, J. W. Booth, R. L. Baines, K. Wang, M. Vespignani, K. Bekris, and R. Kramer-Bottiglio. Tensegrity robotics. Soft robotics, 2021.
25. D. Surovik, J. Bruce, K. Wang, M. Vespignani, and K. E. Bekris. Any-axis tensegrity rolling via bootstrapped learning and symmetry reduction. In International Symposium on Experimental Robotics (ISER), Buenos Aires, Argentina, 11/2018 2018.
26. D. Surovik, K. Wang, M. Vespignani, J. Bruce, and K. E. Bekris. Adaptive Tensegrity Locomotion: Controlling a Compliant Icosahedron with Symmetry-Reduced Reinforcement Learning. International Journal of Robotics Research (IJRR), 2019.
27. A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. In Computer graphics forum, volume 34, pages 101–114. Wiley Online Library, 2015.
28. B. R. Tietz, R. W. Carnahan, R. J. Bachmann, R. D. Quinn, and V. SunSpiral. Tetraspine: Robust terrain handling on a tensegrity robot using central pattern generators. In 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 261–267. IEEE, 2013.
29. M. Vespignani, J. M. Friesen, V. SunSpiral, and J. Bruce. Design of superball v2, a compliant tensegrity robot for absorbing large impacts. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2865–2871. IEEE, 2018.
30. K. Wang, M. Aanjaneya, and K. Bekris. A first principles approach for data-efficient system identification of spring-rod systems via differentiable physics engines. In A. M. Bayen, A. Jadbabaie, G. Pappas, P. A. Parrilo, B. Recht, C. Tomlin, and M. Zeilinger, editors, Proceedings of the 2nd Conference on Learning for Dynamics and Control, volume 120 of Proceedings of Machine Learning Research, pages 651–665. PMLR, 10–11 Jun 2020.
31. K. Wang, M. Aanjaneya, and K. Bekris. Sim2sim evaluation of a novel data-efficient differentiable physics engine for tensegrity robots. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1694–1701. IEEE, 2021.
32. K. Wang, M. Aanjaneya, and K. Bekris. A recurrent differentiable engine for modeling tensegrity robots trainable with low-frequency data. In 2022 IEEE International Conference on Robotics and Automation (ICRA), 2022.