Tensegrity Morphing: Machine Learning-Based Tensegrity Deformation Predictor for Traversing Cluttered Environments

Eduard Zalyaev, Sergei Savin

▶ **To cite this version:**

Eduard Zalyaev, Sergei Savin. Tensegrity Morphing: Machine Learning-Based Tensegrity Deformation Predictor for Traversing Cluttered Environments. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.473-480, 10.1007/978-3-030-85910-7_50 . hal-03806478

HAL Id: hal-03806478

https://inria.hal.science/hal-03806478v1

Submitted on 7 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Tensegrity Morphing: Machine Learning-based Tensegrity Deformation Predictor for Traversing Cluttered Environments*

Eduard Zalyaev[1][0000−0002−6590−0996] and Sergei Savin[1][0000−0001−7954−3144]

Innopolis University, Innopolis, 420500, Russia {e.zalyaev,s.savin}@innopolis.ru

Abstract. In this paper we introduce a neural network-based approach to tensegrity morphing: the task of actively changing the shape of a tensegrity structure to "fit" between obstacles in a cluttered environment. We specifically focus on a class of forming tasks, when the robot is required to pass between two parallel plate-like obstacles, and develop a robust solution both for generating dataset and for training predictor. Proposed predictor is able to predict both the shape of the tensegrity structure and the desired rest lengths of the actuated elastic elements, which can serve as motor commands when a quasi-static configuration-space trajectory tracking is used. We demonstrate high accuracy on validation dataset, and show the conditions when predictor overfits.

Keywords: Deformation of a tensegrity structure · Supervised learning · Dataset generation

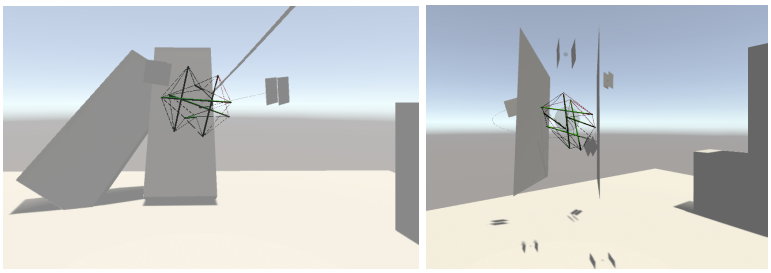
1 Introduction

Tensegrity structures are based on a a connection of a discontinuous set of compressive structural elements (struts) and a continuous set of tensile elements (cables); actuated tensegrity structures, used in robotics applications, are actuated by motors controlling rest lengths of the elastic elements. Such structures are relatively cheap t manufacture, foldable and resilient to impact which makes them useful for exploration and other impact-rich tasks.

Control of such structures poses a new set of challenges when compared to conventional robots, as even basic control approaches for tensegrity mechanisms are still being actively developed. Many of the commonly used control strategies involve the manipulation of cables in order to generate gaits. Control policies are often found by involving a such methods as genetic algorithms [5], reinforcement learning-based policy search for motor actions [3, 11] and neural network-based and Bayesian optimization of central pattern generators (CPG) [2, 6]. These methods cover the case when environment does not require a deformation of the structure to "fit in" or "fit through", which would often be the case when a

* The research is supported by grant of the Russian Science Foundation (project No:19-79-10246)

relatively large and highly deformable structure moves through a cluttered environment. Here we propose a method for automatically predicting correct shape of the robot and high-level motor commands, indicating desired rest lengths of the actuated elastic elements, in order to fit through a particular gap between the obstacles. The method is based on collecting a specially formatted dataset and training a neural network to predict configurations of the tensegrity structure based on the type of a gap between the obstacles that the robot encounters. To



(a) 4 layers (b) 5 layers

Fig. 1: Visualization of a flying tensegrity structure, fitting between narrow plate-like obstacles

the best of our knowledge, this is the first time a neural network-based predictor of the desired form and desired rest lengths of the actuated elastic elements was proposed. Our experiments (for building a dataset and validating the results in simulation) are performed with a a six-bar tensegrity structure.

The rest of the paper is organized as follows. Chapter 2 describes how tensegrity structure is modelled, including the questions of finding static equilibrium configurations under different constraints, and with different actuation patterns. Chapter 3 outlines the dataset generation procedure and the assumptions made for the simulation-based dataset generation. Chapter 4 presents our regression strategy, and chapter 5 gives an analysis of the effects different types of dense neural networks have on the training dynamics of the proposed dataset.

2 Tensegrity model

Tensegrity robot can be described as a graph consisting of n nodes. Connectivity between nodes is characterized by the type of forces acting between the nodes (tensile or compressive). The type of forces defines the physical implementation of the connection, which is a cable for tensile forces, and a strut for compressive ones.

Connections between the nodes can be represented using symmetric binary-valued adjacency matrix $\mathcal{C}$. This matrix describes which node is connected to

which: if element $\mathcal{C}_{ij} = 1$, then the i-th node is connected to the j-th node; $\mathcal{C}_{ij} = 0$, they are not connected.

Elastic properties of the connections can be described using stiffness coefficient matrix $\mathcal{M}$ and rest length matrix $\mathcal{R}$. Elements of $\mathcal{M}$ are denoted as μ_{ij} and define scalar coefficients of the linear elastic force generated between the two nodes. Elements of $\mathcal{R}$ are denoted as ρ_{ij} and define the rest length of the elastic element between i-th and j-th nodes. Note that:

$$\mathcal{M} = \mathcal{M}^\top \ \mathcal{R} = \mathcal{R}^\top \ \mu_{ii} = 0 \ \rho_{ii} = 0 \tag{1}$$

Also $\mathcal{C}_{ij} = 0$ implies $\mu_{ij} = 0$, i.e. $(1 - \mathcal{C}_{ij})\mu_{ij} = 0$. Assuming linear elastic force model, we can define the force $\mathbf{f}_{ij}$ acting between i-th and j-th nodes as follows:

$$\mathbf{f}_{ij} = \mu_{ij}(||\mathbf{r}_i - \mathbf{r}_j|| - \rho_{ij})\frac{\mathbf{r}_i - \mathbf{r}_j}{||\mathbf{r}_i - \mathbf{r}_j||}. \tag{2}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of the nodes i and j in Cartesian space.

Tensegrity robot achieves an *equilibrium position* when the resulting elastic forces acting on each node are equal to zero:

$$\sum_{j=1}^{n} \mathbf{f}_{ij}(\mathbf{r}_i, \mathbf{r}_j, \rho_{ij}) = 0, \ \forall i \tag{3}$$

We define *forward kinematics* as a problem of finding such $\mathbf{r}_i$, that for given ρ_{ij} condition (3) holds. Let us define node position matrix $\mathbf{R}$ given as a horizontal concatenation of the column-vectors $\mathbf{r}_i$, $\forall i$. Then we can define *output vector* for the tensegrity structure as:

$$\mathbf{y} = \mathbf{C}\mathbf{r}; \quad \mathbf{r} = \mathrm{vec}(\mathbf{R}) \tag{4}$$

where $\mathrm{vec}(\cdot)$ is a vectorization of a matrix (performed by stacking its columns to obtain a column-vector), and $\mathbf{C}$ is an output matrix. Assuming that we are given desired value of the output vector denoted as $\mathbf{y}^*$, we can define *inverse kinematics* as a problem of finding such ρ_{ij} and $\mathbf{r}_i$ that both (3) and (4) hold.

In general, only a subset of the cables are actuated, i.e., capable of changing their rest lengths. We can capture this fact by defining *tensegrity control input* as follows:

$$\mathbf{u} = \mathbf{B}\rho; \quad \rho = \mathrm{vec}(\mathcal{R}) \tag{5}$$

where $\mathbf{B}$ is the control matrix, determining which cables are actuated. This implies that

$$\mathbf{N}\rho = \mathrm{const}; \quad \mathbf{N} = \mathrm{null}(\mathbf{B}) \tag{6}$$

where $\mathrm{null}(\cdot)$ is an operator returning an orthonormal basis in the null space of the input matrix. Then inverse kinematics becomes a problem of finding such control input $\mathbf{u}$ that output vector $\mathbf{y}$ approaches its desired value $\mathbf{y}^*$, and conditions (1), (3), (4), (5) and (6) hold.

As a rule, it is computationally easy to generate a dataset of a stable tensegrity configurations by solving forward kinematics for a randomly generated matrix $\mathcal{R}$. Even though forward kinematics is not a convex problem, its geometry corresponds to the geometry of the potential energy of the structure, and its local minima represent stable configurations, i.e. viable solutions. Methods for solving forward kinematics for tensegrity structures are presented in [8, 9].

3 Tensegrity deformation as a data collection problem

Tensegrity structures can deform by changing values of $\rho = \text{vec}(\mathcal{R})$. Once a value of ρ is chosen, the corresponding values of $\mathbf{r} = \text{vec}(\mathbf{R})$ are found via forward kinematics. We will describe a configuration of a tensegrity structure as $\langle \mathbf{r}, \rho \rangle$. Having that description, we can formulate questions regarding the shape of the structure.

3.1 Direct fitting

Let us consider a problem of fitting a tensegrity structure into an allocated area. Assuming the area is described an an H-polytope (inequality representation of a polytope) $H = \{\mathbf{x} : \ \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, with matrix $\mathbf{A}$ and vector $\mathbf{b}$ representing the inequalities, we can define the problem as follows:

$$\begin{aligned} &\text{find} && \mathbf{x}, \mathbf{T} \\ &\text{s.t.} && \begin{cases} \mathbf{A}\mathbf{T}(\mathbf{r}_i + \mathbf{x}) \leq \mathbf{b}, \forall i \\ \mathbf{T}\mathbf{T}^\top = \mathbf{T}^\top \mathbf{T} = \mathbf{I} \end{cases} \end{aligned} \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^3$ is a displacement vector and $\mathbf{T} \in SO(3)$ is a rotation matrix. Searching over the space of rotation matrices is computationally difficult; here we propose to freeze the orientation of the structure, making the computations simple, but introducing additional challenges and limitations that are discussed further in this section. Without the search over possible orientations of the structure, the optimization problem becomes convex:

$$\begin{aligned} &\text{find} && \mathbf{x} \\ &\text{s.t.} && \mathbf{A}(\mathbf{r}_i + \mathbf{x}) \leq \mathbf{b}, \forall i \end{aligned} \tag{8}$$

If (8) has a solution, tensegrity configuration $\langle \mathbf{r}, \rho \rangle$ can fit into a given polytope H without rotations.

There is a special case of this problem that can be solved without the use of quadratic programming. Assume that the allocated area is restricted by two parallel planes, given by their normal vector $\mathbf{a}$, the distance between the planes is h. We will call such area an *opening*. Openings can be used to describe tall narrow doors and windows, spaces under wheeled vehicles, space between trees, buildings, cables and other structural element in an industrial sites, bridges, power lines, etc.

Configuration $\langle \mathbf{r}, \rho \rangle$ can fit into an opening described by a unit vector $\mathbf{a}$ and width h can formulated as:

$$\begin{aligned} w &= \max(\mathbf{a}^\top \mathbf{R}) - \min(\mathbf{a}^\top \mathbf{R}) \\ w &\leq h \end{aligned} \tag{9}$$

Where w is a measure of the width of the configuration with respect to the opening's direction. Condition (9) is can be checked very fast. If we have a dataset of stable configurations, we can pose questions such as which configuration is the "best fit" for the given opening. This can lead to an optimization problem:

$$\begin{aligned} \underset{\mathbf{k}}{\text{minimize}} \quad & ||\mathbf{R}_k - \mathbf{R}^*|| \\ \text{s.t.} \quad & \max(\mathbf{a}^\top \mathbf{R}_k) - \min(\mathbf{a}^\top \mathbf{R}_k) \leq h \end{aligned} \tag{10}$$

where $\mathbf{R}^*$ is a preferred configuration. Solving the problem (10) we can match an opening $(\mathbf{a}, h)$ with the best-fit configuration $\langle \mathbf{r}, \rho \rangle$. This allows us to build a labeled dataset, which can later be used to train a predictor, which could propose a configuration suitable for a given opening.

3.2 Inverse fitting problem

Let us consider the following problem: given configuration $\langle \mathbf{r}, \rho \rangle$, find the narrowest opening in which it can fit; we will call it an *inverse fitting problem*.

This problem can be approximately solved using SVD decomposition. Let matrix $\mathbf{R}$ have the following decomposition: $\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^\top$, where Σ is a matrix with singular values of $\mathbf{R}$ on its diagonal, sorted in descending order, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$. Then $\mathbf{u}_3$ gives us an approximation of the direction in which the structure has the least width. Then we can construct the opening $(\mathbf{a}, h)$, which current configuration would fit in as:

$$\begin{aligned} \mathbf{a} &= \mathbf{u}_3 \\ h &= \max(\mathbf{u}_3^\top \mathbf{R}) - \min(\mathbf{u}_3^\top \mathbf{R}) \end{aligned} \tag{11}$$

This inverse approach to pairing opening with configurations also allows us to generate a dataset to train a predictor. The major difference between the direct and inverse fitting is that the first allows us to focus on typical openings geometries, where as the second allows us to pair each configuration with an opening that fits it "tightly".

In the following section we discuss the use of the proposed dataset in training a stable configuration predictor, given an opening as an input.

4 Predictor structure and regression strategy

Our prior studies of ML-based forward kinematics for tensegrity robots [10] have shown that dense neural networks (DNN) demonstrated the best possible results

Table 1: Mean error and accuracy on validation dataset

Layer sizes	MSE	Accuracy(%)
[10]	2.652e-3 ± 2.0e-7	72.9 ± 0.072
[20]	2.155e-3 ± 2.6e-7	76.4 ± 0.14
[30]	1.730e-3 ± 1.3e-7	78.7 ± 0.093
[40]	1.726e-3 ± 1.2e-7	79.0 ± 0.11
[80]	1.291e-3 ± 2.9e-8	81.6 ± 0.034
[10 10]	1.250e-3 ± 2.4e-7	83.0 ± 0.22
[20 20]	6.570e-4 ± 4.8e-8	87.7 ± 0.13
[30 30]	5.949e-4 ± 7.1e-8	88.4 ± 0.11
[40 40]	5.203e-4 ± 4.5e-8	88.5 ± 0.21
[80 80]	4.113e-4 ± 1.0e-8	89.9 ± 0.16
[10 10 10]	8.097e-4 ± 1.365e-7	86.5 ± 0.15
[20 20 20]	5.475e-4 ± 1.1e-7	89.3 ± 0.15
[80 40 40]	3.695e-4 ± 1.257e-8	92.2 ± 0.035
[80 80 60 50]	3.477e-4 ± 1.865e-8	92.3 ± 0.031
[80 80 80 80]	4.194e-4 ± 1.207e-7	91.1 ± 0.092
[80 80 80 80 80]	4.585e-4 ± 8.812e-8	92.2 ± 0.033

compared to the linear models such as Linear Regression (LR) and Support Vector Machines (SVM) as well as kernel-based SVM. With that in mind, this study is focused exclusively on application of NN to a regression problem.

To determine which NN configuration fits our dataset best, we had to test it in the fashion of increasing complexity for both hidden layer sizes as well as for the number of such layers. Each network structure had ReLU [4] activation functions to prevent vanishing gradient problem from occurring on deeper model structures. The training was performed on 1500 epochs, batch size of 512 and learning rate of 10^{-3} with RMSprop Gradient Descent algorithm. Each model configuration was tested 20 times with the aim of providing more descriptive results that would include ranges for the appropriate metrics.

4.1 Dataset description

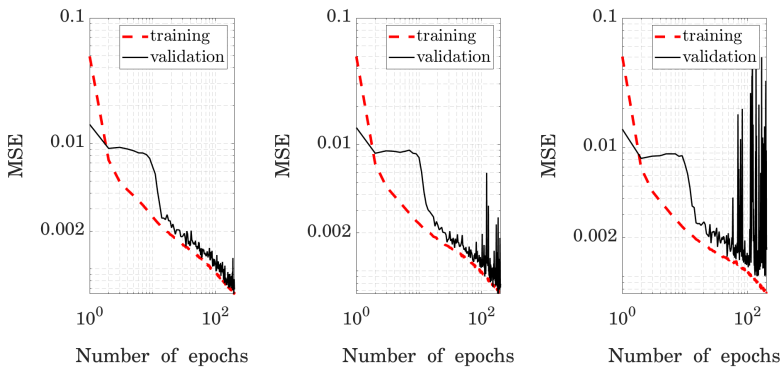
The dataset used in this study is generated for a six-bar tensegrity structure, well known for its use in NASA's SUPERball [1, 7]. The structure includes six rods, giving us total of 12 nodes, with $\mathbf{R} \in \mathbb{R}^{3\times 12}$. Forward kinematics is solved with the algorithm proposed in [8] with a non-convex optimization solver based on interior-point method.

The dataset included of $5 \cdot 10^4$ samples collected with the direct fitting method. Training and validation sets were divided in a ratio of 1 to 5.

5 Evaluation of different DNN structures

To evaluate the performance of the regression model we used mean squared error (MSE). However, the error rates may not always be indicative of the accuracy of the model since the amount of possible configurations that can fit into the opening is not fixed to a single rest length vector. So we additionally test the accuracy of the regression by checking whether or not out estimated rest lengths indeed produce robot configurations that will fit into the desired opening. In order to do that we have take our predicted $\hat{\rho}$ and solve forward kinematics to find corresponding for $\hat{\mathbf{r}}$. Thus found $\hat{\mathbf{r}}$ is then checked against the 9 criterion; if the check succeeds, the prediction is deemed accurate. The results of the evaluation can be seen in the table 1.

It can be seen that the accuracy and error rates are improving with the increase of the model's complexity. To better understand the training dynamics, we plot MSE for three different cases: network with 4, 5 and 6 layers. Fig. 2a shows low variances in error rates as well as error convergence, indicating that the training did not lead the model to overfit. Fig. 2b suggests no improvement with the increased number of layers, and finally Fig. 2c shows overfitting. In



(a) 4 layers (b) 5 layers (c) 6 layers

Fig. 2: Training and validation error curves for a regressor

order to visualize the performance of the designed method we built an obstacle course with way points, tasking the six-bar structure to pass through a number of parallel plates (assuming the structure has enough control authority to move along the given path, as a tensegrity drone would, for example), which were given to the robot as openings, described by their normal vector and width. The robot used pre-trained predictor to find appropriate stable configuration that fits through the opening.

6 Conclusions

In this paper we introduced an approach to formalize the cluttered environment traversal scenario for a tensegrity robot by introducing trajectory consisting of a set of openings that a robot has to fit in. A neural network model for robot stable state prediction from opening parameters was proposed and tested using a six strut tensegrity robot dynamics. Said model achieved 92% accuracy on a validation set while not showing any signs of overfitting. Unfortunately, our method is not universal and requires extensive testing on each individual robot configuration with accuracy validation that may take up a lot of time.

References

1. Bruce, J., Sabelhaus, A.P., Chen, Y., Lu, D., Morse, K., Milam, S., Caluwaerts, K., Agogino, A.M., SunSpiral, V.: Superball: Exploring tensegrities for planetary probes. In: 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS) (2014)
2. Hustig-Schultz, D., SunSpiral, V., Teodorescu, M.: Morphological design for controlled tensegrity quadruped locomotion. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4714–4719. IEEE (2016)
3. Luo, J., Edmunds, R., Rice, F., Agogino, A.M.: Tensegrity robot locomotion under limited sensory inputs via deep reinforcement learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6260–6267. IEEE (2018)
4. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning (ICML-10). pp. 807–814 (2010)
5. Paul, C., Roberts, J.W., Lipson, H., Cuevas, F.V.: Gait production in a tensegrity based robot. In: ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005. pp. 216–222. IEEE (2005)
6. Rennie, C.M.: Designing and learning CPG gaits for spherical tensegrity robots using Bayesian optimization. Ph.D. thesis, Rutgers University-School of Graduate Studies (2018)
7. Sabelhaus, A.P., Bruce, J., Caluwaerts, K., Manovi, P., Firoozi, R.F., Dobi, S., Agogino, A.M., SunSpiral, V.: System design and locomotion of superball, an untethered tensegrity robot. In: 2015 IEEE international conference on robotics and automation (ICRA). pp. 2867–2873. IEEE (2015)
8. Savin, S., Balakhnov, O., Klimchik, A.: Energy-based local forward and inverse kinematics methods for tensegrity robots. In: 2020 Fourth IEEE International Conference on Robotic Computing (IRC). pp. 280–284. IEEE (2020)
9. Savin, S., Balakhnov, O., Maloletov, A.: Linearization-based forward kinematic algorithm for tensegrity structures with compressible struts. In: Proceedings of 15th International Conference on Electromechanics and Robotics" Zavalishin's Readings". pp. 293–303. Springer (2021)
10. Zalyaev, E., Savin, S., Vorochaeva, L.: Machine learning approach for tensegrity form finding: Feature extraction problem. In: 2020 4th Scientific School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR). pp. 265–268. IEEE (2020)

11. 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. IEEE (2017)