Eurographics Symposium on Geometry Processing (2018)
T. Ju and A. Vaxman (Editors)

Local Models for Data Driven Inverse Kinematics of Soft Robots

F. Holsten[1], S. Darkner[1], M. P. Engell-Nørregård[2], and K. Erleben[1]

[1]University of Copenhagen, Denmark
[2]Evil Hippie Design, Denmark

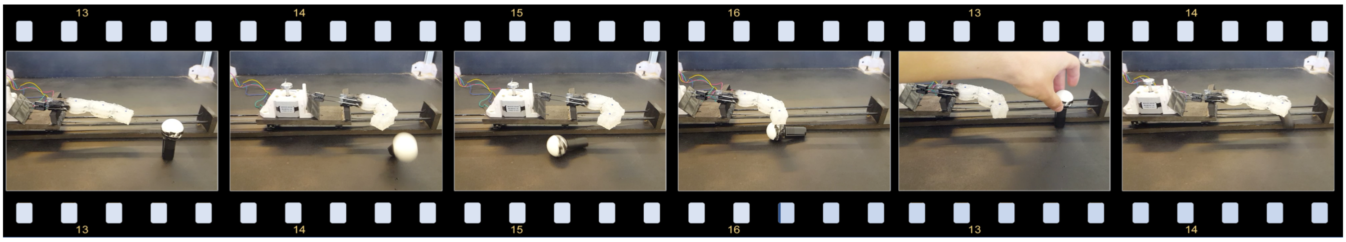

Figure 1: *Vision system is used to detect the ball position and an inverse kinematics (IK) model of a soft robotic finger is used to find control parameters that will hit the ball causing the ball to move on impact.*

Abstract

Soft robots are attractive because they have the potential of being safer, faster and cheaper than traditional rigid robots. If we can predict the shape of a soft robot for a given set of control parameters, then we can solve the inverse problem: to find an optimal set of control parameters for a given shape. This work takes a data-driven approach to create multiple local inverse models. This has two benefits: (1) We overcome the reality gap and (2) we gain performance and naive parallelism from using local models. Furthermore, we empirically prove that our approach outperforms a higher order global model.

CCS Concepts

• ***Computer systems organization*** $\rightarrow$ ***Robotic control;*** • ***Computing methodologies*** $\rightarrow$ *Physical simulation;*

1. Introduction

Current state of the art in controlling soft robotics and animatronics use digital twins. These create forward dynamics simulations that combined with optimization can solve inverse problems [CED17, ZKBT17]. The simulation models are created in a direct approach by describing the viscoelastic properties of the soft robot and specialized actuator models are created for using lines and/or air pressure etc for moving the robots. The advantage of the simulation models is that they generalizes easy. However, they suffer from the reality gap. It is quite hard to get a highly accurate model of a real world soft robot due to imperfections in manufacturing or unknown factors from the real-world. Computational fabrication has started to measure the viscoelastic properties using computer vision and map these to models by numerical coarsening [CLMK17]. Thereby tying simulations models closer to direct data. Our work is based on data obtained from depth cameras. Figure 1 shows motion clips from an inverse kinematics test using our models.

By combining point clouds with fiducial makers from multiple depth cameras we can easily create "shape functions" $\vec{s} \in \mathbf{R}^N$ that describes deformations of the soft robots as a function of the control parameters $\vec{\alpha} \in \mathbf{R}^P$. Our objective is to push for an agile and in-expensive approach to create data-driven models of soft robots. We do this by exploiting a Intel RealSense depth sensors combined with Arduino motor controls. Figure 2 shows our platform. We have applied our approach to several soft robots that we have build ourselves. Examples are shown in Figure 3.

2. Method and Results

The shape $\vec{s}$ of a robot can be described as a function of the control parameters $\vec{\alpha}$ and can be approximated by a Taylor series approximation

$$\vec{s}(\alpha) \approx \vec{s}(\vec{0}) + \mathbf{J}\vec{\alpha} + \cdots \tag{1}$$

We generate training data using a grid-search like technique exploring the $\vec{\alpha}$-space exhaustively. For a given α_k sample we use

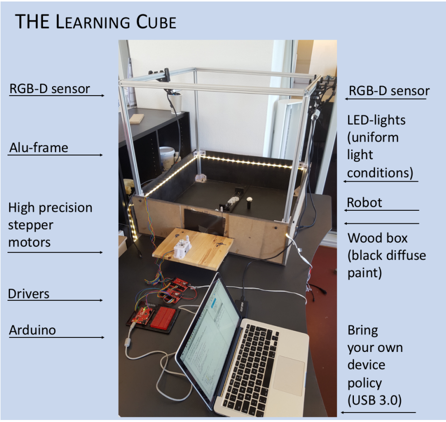

Figure 2: *The Learning Cube: RGB-D platform using Arduino based motor controls to obtain shape deformations of DIY soft robots. The technology is easy accessible and in-expensive. Further, the APIs for using the technology are quite mature and robust.*

Figure 3: *Different robots we have tested our approach on.*

RGBD data to acquire the current shape of the robot $\vec{s}_k$. We denote the rest position by $\vec{s}(\vec{0}) = \vec{s}_0$, and define the displacement field $\vec{u}_k = \vec{s}_k - \vec{s}_0$. We collect all such K data into matrices,

$$\mathbf{A} \equiv \begin{bmatrix} \vec{\alpha}_1 & \vec{\alpha}_2 & \cdot & \vec{\alpha}_K \end{bmatrix} \quad \text{and} \quad \mathbf{U} \equiv \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdot & \vec{u}_K \end{bmatrix} \tag{2}$$

This notation allows us to write up (1) simultaneously for all samples. Without loss of generality, a first order approximation can be written as

$$\mathbf{U} = \mathbf{J}^T \mathbf{A} \quad \text{and} \Rightarrow \mathbf{J} = \mathbf{U}\mathbf{A}^{\dagger}, \tag{3}$$

where $\mathbf{A}^{\dagger}$ is an appropriate pseudo-inverse. A direct approach for computing $\mathbf{J}$ is applicable due to relatively low dimension of controls. This is the main idea of our training and it generalizes to higher order models too. The Taylor approximation can be used to model non-linear shape deformations for any number of actuators.

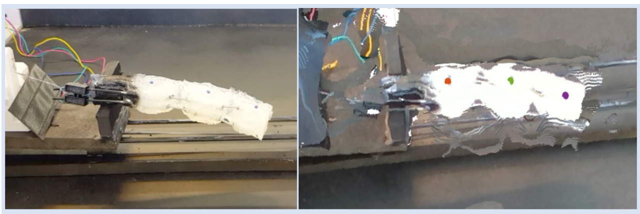

Figure 4: *Left shows a silicone finger and right shows point cloud data. This input is used to capture the shape of the robot.*

Figure 5: *Many low ordered, local models show promising results in terms of run-time complexity and validation loss (accuracy). Observe that 43 models of 1st order are just as "accurate" as using 1 model of 5th order.*

The issue is that the shape vectors we extract from the depth data contains noise, which higher order approximations tend to overfit to. To decrease the complexity of the model, while still being able to learn the non-linear deformations, we split the control parameter space into several disjoint regions. For each region, a lower order model can be fitted to the data as described above. We use optimization when solving the inverse kinematics (IK) problem

$$\vec{\alpha}_{\text{goal}} \equiv \arg\min_{\vec{\alpha}} \frac{1}{2} \,||\, \vec{s}(\alpha) - \vec{s}_{\text{goal}} \,||^2 \tag{4}$$

where $\vec{s}_{\text{goal}}$ is the desired goal shape and $\vec{\alpha}_{\text{goal}}$ is the control parameter solution. For 1st order models the problem reduces to a low dimensional QP method with linear constraints that is very efficient to solve. The IK problem is solved in parallel for each region and the solution with smallest objective value is chosen. For higher order models, we find the best local model using decision trees, which makes the time complexity $\mathcal{O}(1)$ with respect to the number of models. We performed a 5-fold cross-validation for optimal approximation order vs number of local models using two controls. Our results are shown in Figure 5.

3. Discussion and Conclusion

Our early results show that using a spatial decomposition of the configration space and a low-order local model for each subdomain outperforms using one higher order global model. This indicates that simplicity in models can be exploited and hence is very promising for future work on a full dynamic model. Our current approach is limited to static models. However, the RGBD sensors permits us to model low frequencies of motion. We aim to incorporate path and task planning into the models for more complex soft robots in the future.

References

[CED17] COEVOET E., ESCANDE A., DURIEZ C.: Optimization-based inverse model of soft robots with contact handling. *IEEE Robotics and Automation Letters 2*, 3 (July 2017), 1413–1419. 1

[CLMK17] CHEN D., LEVIN D. I. W., MATUSIK W., KAUFMAN D. M.: Dynamics-aware numerical coarsening for fabrication design. *ACM Trans. Graph. 36*, 4 (July 2017), 84:1–84:15. 1

[ZKBT17] ZEHNDER J., KNOOP E., BÄCHER M., THOMASZEWSKI B.: Metasilicone: Design and fabrication of composite silicone with desired mechanical properties. *ACM Trans. Graph. 36*, 6 (Nov. 2017), 240:1–240:13. 1