Learning Latent Space Dynamics for Tactile Servoing

Giovanni Sutanto[1,2,3], Balakumar Sundaralingam[1,4], Yevgen Chebotar[1,3], Zhe Su[2,3], Ankur Handa[1], Nathan Ratliff[1] and Dieter Fox[1,5]

***Abstract*— In order to accomplish an agile robotic manipulation, we need to endow our robot with tactile feedback capability, *i.e.* the ability to drive action based on tactile sensing. In this paper, we specifically address the challenge of tactile servoing, *i.e.* given the current tactile sensing and a target/goal tactile sensing — memorized from a successful task execution in the past — what is the action that will bring the current tactile sensing to move closer towards the target tactile sensing at the next time step. We develop a data-driven approach to acquire a dynamics model for tactile servoing by learning from demonstration. Moreover, our method represents the tactile sensing information as to lie on a surface — or a 2D manifold — and perform a manifold learning, making it applicable to any tactile skin geometry. We evaluate our method on a contact point tracking task using a robot equipped with a tactile finger.**

I. Introduction

Tactile sensing holds an important role in guiding object manipulation, as has been shown in a previous experiment [1]. To this end, recently a variety of tactile sensors [2], [3] have been developed and used in robotics research community, and researchers have designed several tactile-driven control — or popularly termed as *tactile servoing* — algorithms. However, many tactile servoing algorithms were designed for specific kinds of tactile sensor geometry, such as a planar surface [4] or a spherical surface [5], hence they do not apply to the broad class of tactile sensors in general.

In this paper, we present our work on a learning-based tactile servoing algorithm that does not assume a specific sensor geometry. For this purpose, we train a latent space dynamics model [6], [7] –that takes the latent representation of the current tactile sensing and the applied action, and predicts the latent representation of the next tactile sensing– which is termed as *forward dynamics*. Since we use the model for control, *i.e.* tactile servoing, it is also essential that we can compute actions, given both the current and target tactile sensing — termed as *inverse dynamics*. Our contribution is twofold: First, we employ a manifold learning technique to impose a Euclidean structure in the latent space representation of tactile sensing, such that the control in this space becomes straightforward. Second, we train a single model that is able to do both forward dynamics and inverse dynamics prediction for tactile servoing using the same demonstration dataset, which is more data-efficient than training separate models each [8]. All components in our model are trained in a supervised manner, without reinforcement learning.

[1]NVIDIA, USA.
[2]Autonomous Motion Department, MPI-IS, Tübingen, Germany.
[3]University of Southern California, Los Angeles, USA.
[4]University of Utah, Salt Lake City, USA.
[5]University of Washington, Seattle, WA, USA.

This research was supported in part by NVIDIA Research, National Science Foundation grants IIS-1205249, IIS-1017134, EECS-0926052, the Office of Naval Research, the Okawa Foundation, and the Max-Planck-Society.

II. Data-Driven Tactile Servoing Model

A. Tactile Servoing Problem Formulation

Given the current tactile sensing $\mathbf{s}_t$ and the target tactile sensing $\mathbf{s}_T$, the objective is to find the action $\mathbf{a}_t$ which will bring the next tactile sensing $\mathbf{s}_{t+1} = \mathbf{f}(\mathbf{s}_t, \mathbf{a}_t)$ closer to $\mathbf{s}_T$, which in the optimal case can be written as:

$$\mathbf{a}_t^* = \arg\min_{\mathbf{a}_t} d(\mathbf{f}(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_T) \tag{1}$$

B. Latent Space Representation

If the distance metric d is a squared $\mathcal{L}_2$ distance of two states, which lie on a Euclidean space and if $\mathbf{f}$ is smooth, then inverse dynamics $\mathbf{a}_t^*$ can be computed as proportional to $-\frac{\partial d}{\partial \mathbf{a}_t}$. Moreover, for some $\mathbf{f}$, the $\mathbf{a}_t^*$ in Eq. 1 can be computed analytically, at the condition $\frac{\partial d}{\partial \mathbf{a}_t} = \mathbf{0}$. Unfortunately, both $\mathbf{s}_{t+1}$ and $\mathbf{s}_T$ may not lie on a Euclidean space.

However, there seems to be natural characterizations of tactile sensing, such as contact points. A contact point is a 3D coordinate which lies on the skin surface. The skin surface is a 2D manifold (or space) –which may be bent on several parts– existing inside a 3D Euclidean space $\mathbb{R}^3$. If we are able to flatten the skin surface into a 2D Euclidean space $\mathbb{R}^2$, then the inverse dynamics $\mathbf{a}_t^*$ in Eq. 1 can be computed in a straightforward manner. Fortunately, this flattening can be done by a manifold learning technique called Multi-Dimensional Scaling (MDS) [9]. We obtain the latent state $\mathbf{z}$ via the embedding function (at time t) $\mathbf{z}_t = \mathbf{f}_{enc}(\mathbf{s}_t)$ which is represented by the encoder part of an auto-encoder.

C. Latent Space Forward Dynamics (LFD) and Inverse Dynamics (ID)

We assume the LFD model as follows:

$$\dot{\mathbf{z}}_t = \mathbf{f}_{fd}(\mathbf{z}_t, \mathbf{a}_t; \theta_d) \tag{2}$$

where θ_d is the set of trainable parameters of the dynamics model. Numerical integration gives us the discretization:

$$\mathbf{z}_{t+1} = \mathbf{f}_{dfd}(\mathbf{z}_t, \mathbf{a}_t, \Delta t; \theta_d) = \mathbf{z}_t + \mathbf{f}_{fd}(\mathbf{z}_t, \mathbf{a}_t; \theta_d)\Delta t \tag{3}$$

We explored a variety of combinations of LFD and ID models as can be seen in our full paper [10]. Our best result is achieved with a non-linear (NL) LFD model as follows:

$$\dot{\mathbf{z}}_t = \mathbf{f}_{fd}(\mathbf{z}_t, \mathbf{a}_t; \theta_d) = \mathbf{h}_{fcnnNL}(\begin{bmatrix}\mathbf{z}_t \\ \mathbf{a}_t\end{bmatrix}; \theta_d) \tag{4}$$

$\mathbf{h}_{fcnnNL}$ is a fully-connected neural network (fcnn) which represents the NL LFD model. We would like to be able to predict the forward dynamics in the latent space, so we define the following loss function:

$$L_{LFD} = \sum_{t=1}^{H} \left\| \mathbf{f}_{dfd}(\mathbf{z}_t, \mathbf{a}_t, \Delta t; \theta_d) - \mathbf{f}_{enc}(\mathbf{s}_{t+1}) \right\|^2 \tag{5}$$

with $\mathbf{z}_{t+1}$ is computed from Eq. 2 and 3. Based on the NL LFD model, we can derive the following from Eq. 4 (dropping time index t for a moment):

$$\ddot{\mathbf{z}} = \begin{bmatrix}\mathbf{J}_z & \mathbf{J}_a\end{bmatrix}\begin{bmatrix}\dot{\mathbf{z}}\\ \dot{\mathbf{a}}\end{bmatrix} = \mathbf{J}_z\dot{\mathbf{z}} + \mathbf{J}_a\dot{\mathbf{a}} \tag{6}$$

which can be discretized into:

$$\begin{aligned}\mathbf{z}_{t+1} =& \mathbf{z}_t + \left(\frac{1}{\Delta t}\mathbf{I} + \mathbf{J}_{z_{t-1}}\right)\Delta t\,(\mathbf{z}_t - \mathbf{z}_{t-1}) \\ &+ \mathbf{J}_{a_{t-1}}\Delta t\,(\mathbf{a}_t - \mathbf{a}_{t-1})\end{aligned} \tag{7}$$

with $\mathbf{J}_{z_{t-1}}$ and $\mathbf{J}_{a_{t-1}}$ are the Jacobians of $\mathbf{h}_{fcnnNL}$[1] w.r.t. previous latent state $\mathbf{z}_{t-1}$ and previous action $\mathbf{a}_{t-1}$, respectively. Let us define $\bar{\mathbf{A}}_{t-1} = \left(\frac{1}{\Delta t}\mathbf{I} + \mathbf{J}_{z_{t-1}}\right)$, $\bar{\mathbf{B}}_{t-1} = \mathbf{J}_{a_{t-1}}$, and $\bar{\mathbf{c}}_{t-1} = -\bar{\mathbf{A}}_{t-1}\mathbf{z}_{t-1} - \bar{\mathbf{B}}_{t-1}\mathbf{a}_{t-1}$, then Eq. 7 can be written as $\mathbf{z}_{t+1} = \mathbf{z}_t + (\bar{\mathbf{A}}_{t-1}\mathbf{z}_t + \bar{\mathbf{B}}_{t-1}\mathbf{a}_t + \bar{\mathbf{c}}_{t-1})\Delta t$. We can setup a constrained optimal control problem:

$$\begin{aligned}\min_{\mathbf{a}_t, \mathbf{z}_{t+1}} \quad & \frac{1}{2}\left\|\mathbf{z}_T - \mathbf{z}_{t+1}\right\|^2 + \frac{\beta}{2}\left\|\mathbf{a}_t\right\|^2 \\ \text{s.t.} \quad & \mathbf{z}_{t+1} = \mathbf{z}_t + (\bar{\mathbf{A}}_{t-1}\mathbf{z}_t + \bar{\mathbf{B}}_{t-1}\mathbf{a}_t + \bar{\mathbf{c}}_{t-1})\Delta t\end{aligned} \tag{8}$$

whose solution is:

$$\begin{aligned}\mathbf{a}_{t,ID} = \bar{\mathbf{B}}_{t-1}^{\mathrm{T}}\left(\bar{\mathbf{B}}_{t-1}\bar{\mathbf{B}}_{t-1}^{\mathrm{T}} + \frac{\beta}{\Delta t^2}\mathbf{I}\right)^{-1}\Big(\frac{\mathbf{z}_T - \mathbf{z}_t}{\Delta t} - \bar{\mathbf{A}}_{t-1}\mathbf{z}_t \\ - \bar{\mathbf{c}}_{t-1}\Big)\end{aligned} \tag{9}$$

Derivations of Eq. 9 from Eq. 8, can be seen in the Appendix of [10]. From Eq. 9, we can write:

$$\mathbf{a}_{t,ID} = \mathbf{f}_{id}(\mathbf{z}_T, \mathbf{z}_t, \mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \Delta t; \theta_d) \tag{10}$$

For our purpose, it is mostly important that the inferred inverse dynamics action points to the right direction. Therefore, we can leverage the demonstration dataset to also optimize the following inverse dynamics loss:

$$L_{ID} = \sum_{t=1}^{H}\left\|\frac{\mathbf{f}_{id}(\mathbf{z}_T = \mathbf{z}_{t+1}, \mathbf{z}_t, \mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \Delta t; \theta_d)}{\|\mathbf{f}_{id}(\mathbf{z}_T = \mathbf{z}_{t+1}, \mathbf{z}_t, \mathbf{z}_{t-1}, \mathbf{a}_{t-1}, \Delta t; \theta_d)\|} - \frac{\mathbf{a}_t}{\|\mathbf{a}_t\|}\right\|^2 \tag{11}$$

The training losses L_{LFD} and L_{ID} are minimized with respect to θ_d on the human demonstrations' trajectory dataset $\{(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})\}_{t\in\{1,\ldots,H\}}$.

III. Experiments

We evaluate our approach on the right arm of a bi-manual anthropomorphic robot system equipped with a biomimetic tactile sensor BioTac [2] on the tip of the middle finger of the right hand. We setup the end-effector frame to coincide with the BioTac finger frame, and choose the end-effector velocity expressed with respect to the end-effector frame as the action/policy representation for a good policy generalization.

In terms of Multi-Dimensional Scaling (MDS) performance, we plot the x-y coordinates of the latent space embedding of all tactile sensing $\mathbf{s}$ data points in the demonstration, in Figure 1. Each data point is colored and labeled based on the BioTac electrode index with maximum activation.

In Fig. 2, we provide snapshots of robot executions with real-time tactile sensing from the BioTac finger. We see that

[1]These Jacobians and its derivatives (for neural network training) exist because we choose smooth activation functions for $\mathbf{h}_{fcnnNL}$, such as hyperbolic tangent (tanh).

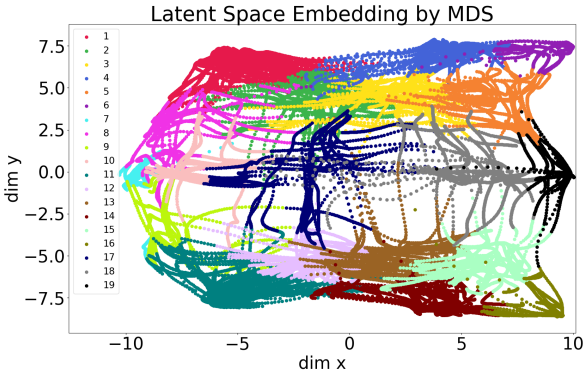
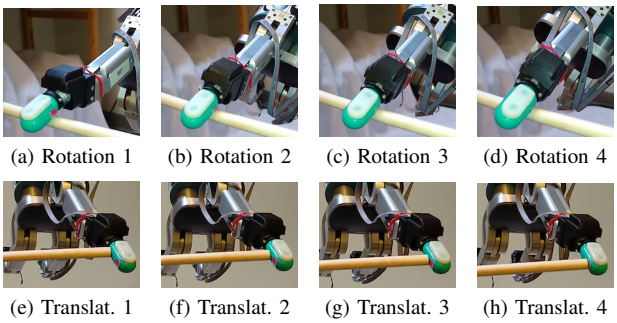


Fig. 1. x-y dimensions of latent space embedding by MDS



(a) Rotation 1 (b) Rotation 2 (c) Rotation 3 (d) Rotation 4

(e) Translat. 1 (f) Translat. 2 (g) Translat. 3 (h) Translat. 4

Fig. 2. Snapshots of our experiments on tactile servoing with the learned model on a robot. Red sticker indicates the target contact point. Figures (a)-(d) are for a target contact point whose achievement requires rotational change of pose of the BioTac finger. Figures (e)-(h) are for a target contact point whose achievement requires translational change of pose.

the system is able to produce the required rotational motions (Fig. 2 (a)-(d)) and translational motions (Fig. 2 (e)-(h)) needed to achieve the specified target contact point. The full pipeline of the experiment can be seen in the video `https://youtu.be/0QK0-Vx7WkI`.

IV. Future Work

We would like to extend this method by integrating vision and tactile information for robust robotic manipulation.

References

[1] R. Johansson, "Light a match: Normal, pre-anesthetization performance vs post-anesthetization performance," https://www.youtube.com/watch?v=0LfJ3M3Kn80, 2018, accessed: 2018-08-04.

[2] N. Wettels, V. Santos, R. Johansson, and G. Loeb, "Biomimetic tactile sensor array." *Advanced Robotics*, vol. 22, no. 8, pp. 829–849, 2008.

[3] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017.

[4] Q. Li, C. Schürmann, R. Haschke, and H. J. Ritter, "A control framework for tactile servoing," in *RSS*, 2013.

[5] N. F. Lepora, K. Aquilina, and L. Cramphorn, "Exploratory tactile servoing with active touch," *IEEE RA-L*, vol. 2, no. 2, April 2017.

[6] A. Byravan, F. Leeb, F. Meier, and D. Fox, "Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control," *CoRR*, vol. abs/1710.00489, 2017.

[7] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *NIPS*, 2015.

[8] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *CoRR*, vol. abs/1606.07419, 2016.

[9] G. Pai, R. Talmon, and R. Kimmel, "Parametric manifold learning via sparse multidimensional scaling," *CoRR*, vol. abs/1711.06011, 2017.

[10] G. Sutanto, N. D. Ratliff, B. Sundaralingam, Y. Chebotar, Z. Su, A. Handa, and D. Fox, "Learning latent space dynamics for tactile servoing," *CoRR*, vol. abs/1811.03704, 2018. [Online]. Available: http://arxiv.org/abs/1811.03704