ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Imitation learning for a continuum trunk robot

Milad S. Malekzadeh[1], J. F. Queißer[2] and J. J. Steil[1] *

1- Institute for Robotics and Process Control, Technische Universität Braunschweig
Muehlenpfordtstr. 23, D-38106 Braunschweig - Germany

2- Research Institute of Cognition and Robotics - Bielefeld University
Universitätsstr. 25, 33615 Bielefeld - Germany

Abstract. The paper applies learning from demonstration (LfD) for high-level trajectory planning and movement control of the Bionic Handling Assistant (BHA) robot. For such soft continuum robot with mechanical elasticity and complex dynamics it is difficult to use kinesthetic teaching to collect demonstration data. We propose to use an active compliant controller to this aim and record both position and orientation of the BHA's end-effector. Subsequently, this data is then encoded with a state-of-the-art task-parameterized probabilistic Gaussian mixture model and its performance and generalization is experimentally evaluated.

1 INTRODUCTION

An increasing number of soft robots have been developed recently, inspired from biological soft structures like the octopus arm [1] or the elephant trunk [2]. In spite of all the well-known benefits of soft manipulators such as hyper-redundancy, flexibility and safe interaction with the environment, the control of such robots remains challenging [11] and calls for biomimetic control as well [3–5].

One widely used bio-inspired approach for rigid robots is Learning from Demonstration (LfD). It is inspired by the way the knowledge is transferred between human beings while performing a task. It typically considers non-expert human users teaching or demonstrating to the robot by kinesthetic teaching, observation or teleoperation and subsequent encoding this data through machine learning, usually followed by trial-and-error learning for refinement.

Among the demonstration methods, kinesthetic teaching appears easiest and safest to implement since it taps into the user's intuitive understanding of the task and is directly performed on the target platform. In addition, the recorded demonstrations are easier to modify and re-implement on the same agent. However, this is not the case for soft robots in general, because of their elastic properties, very complex dynamics and most often the lack of suitable controllers.

Thus there are only few applications of LfD for soft robots. In [3], a context-dependent reward-weighted learning approach is proposed that extracts from demonstrations the weightings of some predefined objective functions. It implements skill transfer through demonstration on a rigid robot to a target soft

*M. Malekzadeh is funded by H2020 under GA 644727 - CogIMon. J. Q. received funding from the Cluster of Excellence 277 – Cognitive Interaction Technology and has been supported by the CODEFROR project (FP7-PIRSES-2013-612555).

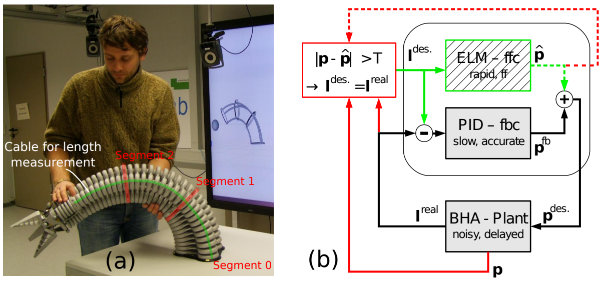

Fig. 1: (a) BHA robot actively follows externally introduced posture changes thereby implementing kinesthetic teaching. (b) Control loop of the BHA robot utilizing feedback control (fbc) and feed forward estimates (ffc), for details see 2.

robot, by defining the reward function in a reinforcement learning algorithm and thereby avoids kinesthetic teaching directly on a soft robot.

In this paper, we utilize an active compliant control that was introduced in [5] to record demonstrations directly with the soft continuum trunk robot Bionic Handling Assistant (BHA, [2]). After recording the demonstrations, we exploit Task-Parametrized Gaussian Mixture Models (TP-GMM) [6] to encode both the end-effector position and orientation. The contribution is to demonstrate feasibility of kinesthetic teaching on a soft robot to learn the full 6D-pose with a task-parameterized Gaussian mixture. The performance and generalization are evaluated in time-dependent and time-invariant scenarios.

2 Kinesthetic teaching on the BHA

The BHA [2] has been designed by Festo as a robotic pendant to an elephant trunk. It is pneumatically actuated and comprises several continuous parallel air chambers operated at low pressures. This makes the BHA inherently safe for physical interaction with humans. The BHA is separated into three segments as shown in Fig. 1a, each of which consists of three air chambers in a triangular arrangement. The air chambers extend their length in relation to the pressure in those chambers. A fourth end-effector segment is also available for grasping. The robot has no fixed joint angles and each segment bends if the three chambers have different lengths. Beside pressure sensors that are included in the air valves, the BHA is equipped with cable potentiometers that allow to measure the outer length of the air chambers providing information about the robot's geometric configuration.

The BHA is controlled in a component based software framework that we developed earlier [7]. In principle, length control can be accomplished with standard proportional integral derivative (PID) schemes, however, with low gains due to the slow plant dynamics, which results in very slow movements. In Fig. 1b, the black loop depicts this classic PID controller: $\boldsymbol{l}^{real}$ and $\boldsymbol{p}^{des}$ are the current segment's lengths and the desired pressure. To overcome this issue our BHA low level controller utilizes a learned equilibrium model to generate an additional feed forward signal that predicts required pressures $\hat{\boldsymbol{p}}$ (the green loop) for postures with zero velocity and acceleration. The combination of a slow PID controller and the feed forward signal of the equilibrium model leads to a significant improvement of length control [5]. For estimation of end-effector positions, we refer to an approximate kinematic model ignoring pressures and solely operating on the lengths of virtual air chambers [8].

The earned equilibrium model can also be used to implement a kinesthetic teaching mode to record position and orientation trajectory of the end-effector, first described for positions only in [5]. As shown in Fig. 1b, outer red loop, a compliance mode is enabled through sensing deviation of the actual pressures $\boldsymbol{p}$ from the predicted pressures $\hat{\boldsymbol{p}}$ of the pneumatic chambers for the current posture $\boldsymbol{l}^{real}$. Due to the elastic material of the robot, $\boldsymbol{p}$ can deviate from $\hat{\boldsymbol{p}}$ when the demonstrating user is reforming BHA during the kinesthetic teaching. In case this mismatch exceeds a threshold T, a posture update is initiated in the robot controller to comply with the deformed robot configuration i.e., pressures follow the kinesthetically taught postures. An interaction using this active compliance control is shown in Fig. 1a.

3 Dynamical system for position and orientation

The recorded data consists of 6D-poses of the robot's end-effector represented by $\boldsymbol{x} = [\boldsymbol{x}^{p\top}\ \boldsymbol{x}^{o\top}]^\top$, where $\boldsymbol{x}^p$ and $\boldsymbol{x}^o$ are the position and orientation (axis-angle representation). To increase robustness in task-space control when facing perturbations and to enable compliancy of the task by introducing tracking gains for the task, we first encode the data into the virtual attractor space following [9]. I.e. we assume a virtual unit mass at the end-effector of the robot, where two separate dynamical systems control the position and orientation of this unit mass by weighted superposition of virtual spring-damper systems. This separation is necessary due to the incommensurability of modalities.

Also the velocity and acceleration $\dot{\boldsymbol{x}}^p$, $\ddot{\boldsymbol{x}}^p$ are obtained from demonstration through numerical differentiation. After preprocessing, the Cartesian position is transformed into the movement of virtual unit-mass attractor points $\hat{\boldsymbol{x}}^p$.

$$\ddot{\boldsymbol{x}}^{\boldsymbol{p}} = \boldsymbol{K}^{\mathcal{P}}(\hat{\boldsymbol{x}}^p - \boldsymbol{x}^p) - \boldsymbol{K}^{\mathcal{V}}\dot{\boldsymbol{x}}^p, \tag{1}$$

where $\boldsymbol{K}^{\mathcal{P}}$, $\boldsymbol{K}^{\mathcal{V}} \in \mathbb{R}^{3\times 3}$ are the stiffness and damping matrices, set to have a critically damped system.

A second dynamical system with different tracking gains is used to convert the orientation of the end-effector expressed in a unit quaternion into the orientation of another virtual attractor $\hat{\boldsymbol{\mathfrak{x}}}^o$ in the unit quaternion space [10].

$$\ddot{\boldsymbol{x}}^{\boldsymbol{o}} = 2\boldsymbol{K}^{\mathcal{O}}\log(\hat{\boldsymbol{\mathfrak{x}}}^o * \bar{\boldsymbol{\mathfrak{x}}}^o) - \boldsymbol{K}^{\mathcal{W}}\dot{\boldsymbol{x}}^o, \tag{2}$$

where $\boldsymbol{K}^{\mathcal{O}}$, $\boldsymbol{K}^{\mathcal{W}} \in \mathbb{R}^{3\times 3}$ are the angular stiffness and damping and $\dot{\boldsymbol{x}}^o$ and $\ddot{\boldsymbol{x}}^{\boldsymbol{o}}$ are the angular velocity and acceleration. The quaternion equivalence of the axis-angle representation of the orientation $\boldsymbol{x}^o$, is represented by $\boldsymbol{\mathfrak{x}}^o$. Note that here $\bar{\boldsymbol{\mathfrak{x}}}^o$ is the quaternion conjugate of $\boldsymbol{\mathfrak{x}}^o$ and $*$ denotes the quaternion product.

4 Task-parametrized Gaussian Mixture Model for full pose

Consider a set of task-parameters in the form of frames of references, that matter for each demonstration, along with a set of demonstrations that depend on these. In the demonstration phase of TP-GMM, along with the demonstration trajectories, a set of task parameters are also recorded in the form of coordinate frames (local frames). In the encoding phase, all the recorded position and orientation trajectories are transformed into the local frames. A separate GMM is fitted over the data at each frame of reference. This is actually the TP-GMM encoding: a set of local GMMs. The resulted model encompasses the properties of all the local frames. In the reproduction phase, a new set of local frames are given. The TP-GMM is first transformed back into the world frame using the new frames of references. The component-wise time-respected Gaussian product of these GMMs, results in a single GMM which can be used with Gaussian Mixture Regression (GMR) to reproduce a new trajectory corresponding to the new set of frames (see [6, 9] for more details and experimental results).

The dataset $\boldsymbol{\xi}_n = [\boldsymbol{\xi}_n^{\mathcal{IN}},\ \boldsymbol{\xi}_n^{\mathcal{OUT}}]^\top$ includes both position and orientation attractors, where $\boldsymbol{\xi}_n^{\mathcal{IN}}$ and $\boldsymbol{\xi}_n^{\mathcal{OUT}}$ are the input and output part of $\boldsymbol{\xi}_n$ at time step n and $\boldsymbol{\xi}_n^{\mathcal{OUT}} = [\hat{\boldsymbol{x}}_n^p,\ \hat{\mathfrak{r}}_n^o]^\top$. For the time-based movement in 3D Cartesian space, $D = 8$ and $\boldsymbol{\xi}_n$ aggregates time as input ($\boldsymbol{\xi}^{\mathcal{IN}}$, one dimension) and Cartesian position (3 dimensions) and a unit quaternion orientation (4 dimensions).

The task parameters are P frames, represented by tuples $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^P$ at each time step n. For position data, they parametrize the origin and rotation matrix of the j-th coordinate system w.r. to a global frame of reference. For the quaternion orientation data, $\boldsymbol{b}_{n,j} = \boldsymbol{0}$ and $\boldsymbol{A}_{n,j} \in \mathbb{R}^{4\times 4}$ is the matrix representation of the j-th quaternion orientation of the j^{th} frame at time step n.

A task-space attractor trajectory $\boldsymbol{\xi} \in \mathbb{R}^{D\times N}$ with N samples in the global frame of reference can be observed from the viewpoint of each of the P coordinate systems (task-parameters) to obtain P different trajectories $\{\boldsymbol{X}^{(j)}\}_{j=1}^P \in \mathbb{R}^{D\times N}$. At each time step n, this change of perspective can be obtained by a linear transformation as $\boldsymbol{X}_n^{(j)} = \boldsymbol{A}_{n,j}^{-1}(\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j})$.

The transformed position attractor yields from $\boldsymbol{R}_{n,j}^{-1}(\hat{\boldsymbol{x}}_n^p - \boldsymbol{o}_{n,j})$, where $\boldsymbol{R}$ is the corresponding rotation matrix and $\boldsymbol{o}$ is the origin of the coordinate frame and the projected orientation attractor will be $\boldsymbol{A}_{n,j}^{-1}(\hat{\mathfrak{r}}_n^o - \boldsymbol{b}_{n,j}) = \boldsymbol{\mathcal{Q}}_{n,j}^{-1}\,\hat{\mathfrak{r}}_n^o$, where $\boldsymbol{\mathcal{Q}}$ is the quaternion matrix representation of $\boldsymbol{R}$. Learning then consists of iteratively updating the model parameters defined by $\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^P\}_{i=1}^K$, for a model with K components, where π_i is the mixing coefficient for the i^{th} Gaussian component and $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are center and covariance matrix of the i^{th} Gaussian component at frame j.

Given a set of task-parameters $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^P$, the learned model can be used to reproduce the demonstrations or generalize to trajectories for new frame of references. At each time step n, the model first retrieves a temporary GMM as a product of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) \ \propto\ \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j},\ \boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^\top\Big). \tag{3}$$

Gaussian Mixture Regression (GMR) is then used to retrieve the trajectory from this temporary model by estimating the conditional probability $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{OUT}} | \boldsymbol{\xi}_n^{\mathcal{IN}})$ relying on the joint probability $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{IN}}, \boldsymbol{\xi}_n^{\mathcal{OUT}})$ (more details in [6]).

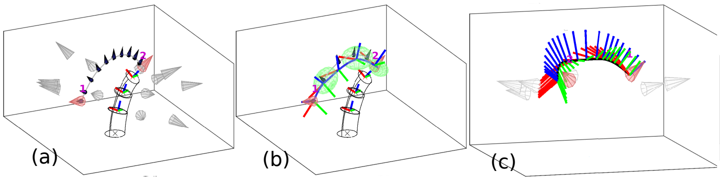

Fig. 2: (a) Demonstration, (b) Reproduction, (c) Reproduction for new set of frames.

By using the dynamical system of Eq. (1) and (2), it is straightforward to reproduce the position and orientation from their attractors, where stiffness and damping gains can be adjusted to implement desired task compliance.

5 Experiments

Experiment 1: teaching a point to point end-effector movement. For eight demonstrations, start and end Cartesian poses are chosen as fixed frames of reference (gray cones in Fig. 2a: going from 1 to 2). The user moves the robot from the start to the end while recording

$$\boldsymbol{\xi}_n = \begin{bmatrix} t_n \\ \hat{\boldsymbol{x}}^p_n \\ \hat{\boldsymbol{\mathfrak{r}}}^o_n \end{bmatrix}, \boldsymbol{b}_{n,j} = \begin{bmatrix} 0 \\ \boldsymbol{o}^{(j)}_n \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{A}_{n,j} = \begin{bmatrix} 1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^{(j)}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{Q}}^{(j)}_n \end{bmatrix}, \tag{4}$$

where $\boldsymbol{o}^{(j)}_n$ is the Cartesian position of the origin of j^{th} frame and $\boldsymbol{R}^{(j)}_n$ and $\boldsymbol{\mathcal{Q}}^{(j)}_n$ are respectively the rotation matrix and quaternion matrix representation of the orientation of j^{th} frame. The model has three Gaussian components and the stiffness and damping gains are $k^{\mathcal{P}} = 500$, $k^{\mathcal{V}} = 50$ and $k^{\mathcal{O}} = 250$, $k^{\mathcal{W}} = 25$.

TP-GMM is used to encode the recorded pose trajectories wrt these fixed frame of references. Fig. 2b shows the reproduction of the end-effector poses along the trajectory, given the same pair of frames as Fig. 2a. The retrieved GMM (green ellipsoids) are overlayed and constant over time as the reference frames are fixed. The generalization of TP-GMM is demonstrated in Fig. 2c though providing different poses for start and end points. The blue arrow is the robot's end-effector orientation retrieved by the model. The model is able to produce a suitable and smooth movement.

Experiment 2: tracking the position and orientation of a flying object (a cup, shown in Fig. 3 by blue cones) in the work-space by the end-effector. The position of the cup in the 3D Cartesian space is the input to learn a time-invariant task based on two frames of references. The first frame is the moving cup (blue cones) while the second frame is the fixed base of the robot. For $j = 1$, the first frame, we replace in Eq(4) $t_n \rightarrow \boldsymbol{o}^{cup}_n$, $\boldsymbol{o}^{(j)}_n \rightarrow \boldsymbol{o}^{cup}_n$, $\boldsymbol{R}^{(j)}_n \rightarrow \boldsymbol{R}^{cup}_n$ and $\boldsymbol{\mathcal{Q}}^{(j)}_n \rightarrow \boldsymbol{\mathcal{Q}}^{cup}_n$ with time-independent cup position and orientation (rotation matrix and quaternion matrix representation). Here, $D = 10$ ($\boldsymbol{o}^{cup}_n$ is 3D position).

Fig. 3a, shows one sample demonstration and frames. The demonstrations are recorded in the active compliant control, where the user physically moves the robot to follow the pose of the red cup which was moved randomly in space (by the experimenter). We collect 6 demonstrations. Since the first frame of reference is always in the vicinity of the robot's end-effector, the model learned the importance of this frame. Note that this implicit information is in the

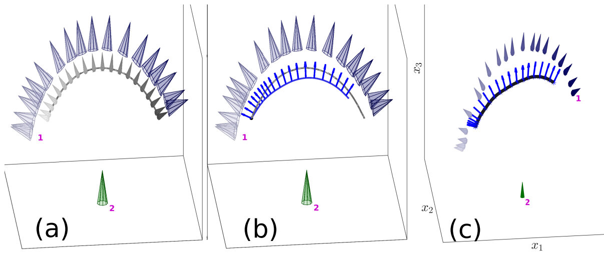

Fig. 3: (a) Demonstration, (b) Reproduction, (c) Reproduction for new pose of frames.

demonstration data and not given to the model explicitly. Fig. 3b depicts the reproduced pose for the movement of Fig. 3a. The trajectory and corresponding orientation are shown by blue lines and arrows for some of the time instances. We examined the generalization of the learned model by providing different poses for the moving frame (the cup in the real experiment). Fig. 3c, shows a new situation in which the robot successfully follows the pose of the second frame. The poses of the moving frame have been shown by gray cones that get darker towards the end of the movement. The reproduced trajectory of end-effector is shown by the black line on which the blue arrows are the reproduced orientations. This experiment is provided to show the capabilities of the algorithm in case the experiment is not time dependent.

References

[1] M. Cianchetti, T. Ranzani, G. Gerboni, T. Nanayakkara, K. Althoefer, P. Dasgupta, and A. Menciassi. Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach. *Soft Robotics*, 1(2):122–131, 2014.

[2] A. Grzesiak, R. Becker, and A. Verl. The Bionic Handling Assistant - A Success Story of Additive Manufacturing. *Assembly Automation*, 31(4):329–333, 2011.

[3] S. Calinon, D. Bruno, M. S. Malekzadeh, T. Nanayakkara, and D. G. Caldwell. Human-robot skills transfer interfaces for a flexible surgical robot. *Computer Methods and Programs in Biomedicine*, 116(2):81–96, September 2014. Special issue

[4] M. S. Malekzadeh, S. Calinon, D. Bruno, and D. G. Caldwell. Learning by imitation with the STIFF-FLOP surgical robot: A biomimetic approach inspired by octopus movements. *Robotics and Biomimetics, Special Issue on Medical Robotics*, 1(13):1–15, October 2014.

[5] J. F. Queißer, K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil. An active compliant control mode for interaction with a pneumatic soft robot. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 573–579. IEEE, 2014.

[6] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.

[7] M. Rolf, K. Neumann, J. Queißer, F. Reinhart, A. Nordmann, and J.J. Steil. A Multi-Level Control Architecture for the Bionic Handling Assistant. *Advanced Robotics*, 2015.

[8] M. Rolf and J.J. Steil. Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. *Proc. IEEE/RSJ IROS* pages 3440–3446, 2012.

[9] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell. Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems. *Proc. IEEE/RSJ IROS*, Hamburg, Sept.-Oct. 2015.

[10] A. Ude, B. Nemec, T. Petrić, and J. Morimoto. Orientation in cartesian space dynamic movement primitives. *Proc. IEEE ICRA*, pages 2997–3004, 2014.

[11] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann and O. Sawodny. Dynamic modeling of constant curvature continuum robots using the Euler-Lagrange formalism. *Proc. IEEE/RSJ IROS* pages 2428–2433, 2014.