Learning the end-effector pose from demonstration for Bionic Handling Assistant robot

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

***Abstract*— For most of the rigid manipulators, it is possible to apply a gravity compensation mode, by which the user is able to easily reconfigure the arm and record the necessary data. However, due to the specific characteristics of soft robots such as elastic properties and complex dynamics, it is usually very difficult to implement kinesthetic teaching for Learning from Demonstration (LfD) scenarios. This paper tackles this problem on a soft continuum robot named Bionic Handling Assistant (BHA). We propose to use an active compliant controller that facilitates the kinesthetic teaching for the user while recording the position and orientation of the robot's end-effector. The recorded demonstrations are then encoded with a task-parameterized probabilistic model through two separate dynamical systems (one for the position and one for the orientation). The approach was evaluated by conducting two experiments on the BHA robot.**

I. INTRODUCTION

An increasing number of soft robots have been developed recently, inspired from the soft structures in nature like octopus arm [1] or elephant trunk [2], based on the belief that biological structures have been efficiently established and evolved. 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 [3], [4].

One well-known control approach that is widely used for rigid robots, is Learning from Demonstration (LfD). LfD is based on imitation and exploits machine learning techniques. It is inspired by the way the knowledge is transferred between human beings while performing a task, usually with subsequent trial-and-error learning. In LfD, the goal is to extract important features of the task and its reproduction in new situations characterized by robustness with respect to possible perturbations. It typically considers non-expert human users teaching or demonstrating to the robot by kinesthetic teaching, observation or teleoperation and replaces the need to program the robot by an expert user for every specific task.

Although the LfD is a well-known method for the rigid robots, there are only a few applications of it in soft-robotics. For instance in [3], Calinon *et.al* proposed a context-dependent reward-weighted learning approach that is able to extract from demonstrations (fulfilled on a very different rigid robot), the weights for some predefined objective functions. The weighted sum of these objective functions is then used for the target soft robot, as the reward function in a reinforcement learning algorithm [5]. This can be seen as a skill transfer approach from a rigid robot to a soft robot. This is important since providing demonstrations is not usually easy for a soft robot, while performing kinesthetic teaching on a rigit robot is usually trivial.

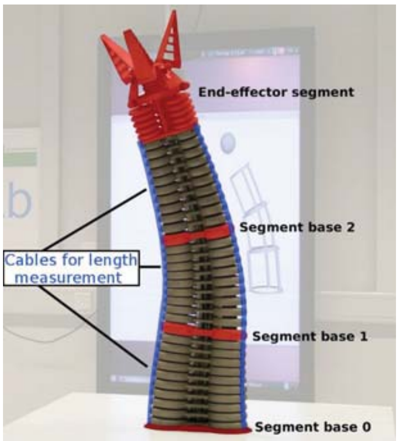


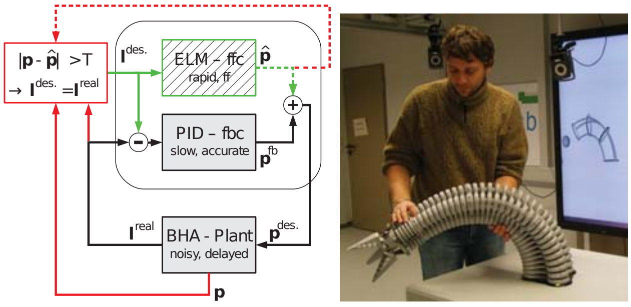
Fig. 1: BHA robot platform: three segments are connected (orange structure) in serial. Each segment's shape is defined by three air chambers that provide length information by cable sensors (blue).

Among all the methods to collect the demonstrations, kinesthetic teaching seems easier, faster and safer to implement since the user often tries to reconfigure the robot by his/her hands. In addition, the recorded demonstrations are easier to modify and re-implement on the same agent. However, it is not the case while we are dealing with soft robots in general, mainly because of specific characteristics of them such as elastic properties, very complex dynamics and lack of suitable controller.

In this paper we utilize an active compliant control mode already introduced in [2], to record kinesthetic demonstrations directly with a soft continuum robot termed as Bionic Handling Assistant (BHA). For most of the rigid manipulators, the gravity compensation modes can be set for recording the demonstrations during kinesthetic teaching. We propose to use a controller that acts like the gravity compensation mode and helps the user to record demonstrations (section II).

After recording the demonstrations, we exploit Task-Parametrized Gaussian Mixture Models (TP-GMM) [6] to encode the end-effector pose. Among several available LfD tools including Gaussian mixture regression [7], hidden Markov models [8], dynamic movement primitives [9], Gaussian process regression [10], we exploit the TP-GMM

[1] Milad S. Malekzadeh, J. F. Queißer, and J. J. Steil are with the Research Institute of Cognition and Robotics (CoR-Lab), Faculty of Technology at Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany, Email: mmalekzadeh, jqueisse, jsteil@cor-lab.uni-bielefeld.de



(a)

(b)

Fig. 2: (a) Low-Level control loop of the BHA robot utilizing feedback control (fbc) and feed forward estimates (ffc) of the equilibrium model, leaned by extreme learning machine (ELM), in addition to the compliant control mode highlighted in red. (b) Use case: BHA robot is following external force of interaction and allows for kinesthetic teaching.

approach that has better generalization properties for learning of position and orientation, compared to other similar methods.

We are interested in learning both the position and orientation for the robot's end-effector to suitably perform the experiments, we exploited the method described in [11] (section IV). A different pan-tilt parameterization of orientation was employed in [12] which is more suitable to encode the initial and final orientations of the end-effector.

The contribution of this paper is twofold: 1) A new controller permits the user to apply the kinesthetic teaching on a soft robot (the BHA robot) for the first time. The same controller can be used on similar soft robots (and even rigid ones); 2) The recorded demonstrations were used to learn the full pose of a soft-continuum robot end-effector with a task-parameterized version of Gaussian mixtures. The generalization capability of the proposed method is then evaluated in time-based and time-invariant scenarios.

II. The BHA robot

The Bionic Handling assistant (BHA, [13], [14]) has been designed by Festo as a robotic pendant to an elephant trunk. It is pneumatically actuated and comprises several continuous parallel components operated at low pressures, which makes the BHA inherently safe for physical interaction with humans and an interesting platform for collaborative robotics tasks. A further key aspect is the low-priced and rapid 3D manufacturing process with polyamide, resulting in application fields of small and medium sized enterprises like pick-and-place tasks.

The structure of the BHA is separated into three segments as shown in Fig. 1. Each segment consists of three triangular arranged air chambers. Therefore the main flexibility of the BHA is based on 9 air chambers that extend their length in relation to the pressure in those chambers. A fourth end-effector segment is also available but was neglected for this work. An active depression of the pressure of the chambers is not possible, solely the tension of the extended body reforms the structure back to the home position. The robot has no fixed joint angles and each robot segment starts to bend in the case that the three chambers reach 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 geometric information about the robot shape. Unfortunately, the hardware design of the BHA bothers analytical modeling. This includes elastic properties, complex dynamics of continuum deformation, hysteresis effects, long control delays and changing material properties. The whole control architecture is embedded in a component based software framework as shown in [15].

In principle, the length control can be accomplished with standard proportional integral derivative (PID) schemes. The fundamental problem is that these feedback control approaches can be applied only with low gains due to the slow plant dynamics, which consequently results in very slow movements. To overcome this issue the BHA low level controller refers to an equilibrium model to generate an additional feed forward signal. The equilibrium model predicts required pressures 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 [2].

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 [16]. Additionally it has been shown that the model error can be reduced by machine learning techniques [17]. A constant-curvature approach that is based on torus segments allows for kinematic simulation of continuous deformations. For each segment, the related three measured lengths of the actuators can be used to estimate the coordinate transformation between two platforms, which can then be chained in order to get the complete forward kinematics from base to end effector.

Active Compliant Control Mode for Kinesthetic Teaching

We refer to the utilization of a learned equilibrium model of the robot to implement a kinesthetic teaching mode (described in [2]). The low level control loop is shown in Fig. 2a. In the compliant mode, the deviation of the sensed pressures $\boldsymbol{p}$ and the predicted pressures $\hat{\boldsymbol{p}}$ of the pneumatic chambers for the current posture $\boldsymbol{l}^{real}$ is observed. Due to the elastic material, a deformation of the robot while keeping the chamber pressures constant, is possible. So a deformation of the robot will result in a mismatch between predicted and observed pressures. In case this mismatch exceeds threshold T, a posture update is initiated to comply with the deformed robot configuration. The threshold T has to be selected concerning sensory noise and uncertainty of the equilibrium model. A sample interaction with the BHA robot is shown in Fig. 2b.

III. Dynamical system for position and orientation

The recorded kinesthetic teaching dataset using the active compliant control, consists of position and orientation of the robot's end-effector. The pose of the end-effector needs to be modeled during the encoding phase and reproduced during the reproduction phase. However due to the multiple advantages of using dynamical systems in task-space control, we first encode the data into the virtual attractor space [11]. E.g., robustness when facing perturbations and control over the compliancy of the task execution by tuning the tracking gains, are some of the benefits of exploiting such approach. To do this, we assume a virtual unit mass at the end-effector of the robot, where two dynamical systems, control the position and orientation of this unit mass by weighted superposition of virtual spring-damper systems. The dynamical systems responsible for controlling the position and orientation are separated, due to the independence of modalities. In this paper this method is used to control the pose of a soft robot (BHA). However in [11], this approach has been used to control the pose of two Barrett WAMs (rigid robots) in a bi-manual task. The end-effector full pose is 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).

A. Dynamical System for position

During demonstration, the position of the robot's end-effector $\boldsymbol{x}^p$ is recorded along with its velocity and acceleration $\dot{\boldsymbol{x}}^p$, $\ddot{\boldsymbol{x}}^p$. After preprocessing, the Cartesian position is transformed into the movement of virtual unit-mass attractor points. The dynamical system is a second order linear differential equation, given by

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

where $\boldsymbol{K}^{\mathcal{P}}$, $\boldsymbol{K}^{\mathcal{V}} \in \mathbb{R}^{3\times3}$ are the stiffness and damping matrices, set to have critically damped system. In our application, $\boldsymbol{K}^{\mathcal{P}} = k^{\mathcal{P}}\boldsymbol{I}$ and $\boldsymbol{K}^{\mathcal{V}} = k^{\mathcal{V}}\boldsymbol{I}$, where $k^{\mathcal{V}} = 2\sqrt{k^{\mathcal{P}}}$. The trajectory of the virtual attractor $\hat{\boldsymbol{x}}^p$, is then computed by

$$\hat{\boldsymbol{x}}^p = (\boldsymbol{K}^{\mathcal{P}})^{-1}\ddot{\boldsymbol{x}}^{\boldsymbol{p}} + (\boldsymbol{K}^{\mathcal{P}})^{-1}\boldsymbol{K}^{\mathcal{V}}\dot{\boldsymbol{x}}^p + \boldsymbol{x}^p. \tag{2}$$

B. Dynamical System for orientation

Based on the formulation proposed in [11], [18], a second dynamical system with different tracking gains is used to convert the orientation of the end-effector expressed in unit quaternion, into the orientation of another virtual attractor. The equivalence of Eq. (1) in the unit quaternion space will be

$$\ddot{\boldsymbol{x}}^{\boldsymbol{o}} = 2\boldsymbol{K}^{\mathcal{O}}\log(\hat{\mathfrak{x}}^o * \bar{\mathfrak{x}}^o) - \boldsymbol{K}^{\mathcal{W}}\dot{\boldsymbol{x}}^o, \tag{3}$$

where $\boldsymbol{K}^{\mathcal{O}}$, $\boldsymbol{K}^{\mathcal{W}} \in \mathbb{R}^{3\times3}$ are the angular stiffness and damping matrices and $\dot{\boldsymbol{x}}^o$ and $\ddot{\boldsymbol{x}}^{\boldsymbol{o}}$ are the angular velocity and acceleration ($\boldsymbol{K}^{\mathcal{O}} = k^{\mathcal{O}}\boldsymbol{I}$ and $\boldsymbol{K}^{\mathcal{W}} = k^{\mathcal{W}}\boldsymbol{I}$, where $k^{\mathcal{W}} = 2\sqrt{k^{\mathcal{O}}}$). The quaternion equivalence of the axis-angle representation of the orientation $\boldsymbol{x}^o$, is represented by $\mathfrak{x}^o$. Similarly, $\hat{\mathfrak{x}}^o$ represents the orientation attractor. Note that here $\bar{\mathfrak{x}}^o$ is the quaternion conjugate of $\mathfrak{x}^o$ and $*$ denotes the quaternion product.

Eq. (1) and (3) are similar except for the tracking error term on the right side of the equations. In (1), the term $(\hat{\boldsymbol{x}}^p - \boldsymbol{x}^p)$ represents the error of the positions in the Cartesian space while the quaternion product $(\hat{\mathfrak{x}}^o * \bar{\mathfrak{x}}^o)$ in (3), gives the orientation error in unit quaternion space[1].

Based on the definition, the quaternion representation $\mathfrak{q}$ for a vector of axis-angle orientation $\boldsymbol{\theta} \in \mathbb{R}^{3\times1}$ is

$$\mathfrak{q} = \exp(\boldsymbol{\theta}) = \begin{cases} [\cos(\|\boldsymbol{\theta}\|) \quad \sin(\|\boldsymbol{\theta}\|)\frac{\boldsymbol{\theta}^\top}{\|\boldsymbol{\theta}\|}]^\top & , \ \boldsymbol{\theta} \neq 0 \\ [1\ 0\ 0\ 0]^\top & , \ \text{otherwise} \end{cases}.$$

A logarithmic map will inverse the mapping

$$\log(\mathfrak{q}) = \log\left(\begin{bmatrix} v \\ \boldsymbol{u} \end{bmatrix}\right) = \begin{cases} \arccos(v)\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|} & , \ \boldsymbol{u} \neq 0 \\ [0\ 0\ 0]^\top & , \ \text{otherwise} \end{cases}.$$

This mapping is one-to-one correspondent for $\|\boldsymbol{\theta}\| < \pi$. From Eq. (3), we can compute $\hat{\mathfrak{x}}_o$ using the above definition

$$\hat{\mathfrak{x}}_o = \exp\left(\tfrac{1}{2}(\boldsymbol{K}^{\mathcal{O}})^{-1}\ddot{\boldsymbol{x}}_{\boldsymbol{o}} + \tfrac{1}{2}(\boldsymbol{K}^{\mathcal{O}})^{-1}\boldsymbol{K}^{\mathcal{W}}\dot{\boldsymbol{x}}_o\right) * \mathfrak{x}_o \ , \tag{4}$$

by which we can retrieve the quaternion attractor through another dynamical system and suitable choice of corresponding stiffness and damping gains.

We therefore have computed the position attractor $\hat{\boldsymbol{x}}_p$ and the orientation attractor (in unit quaternion space) $\hat{\mathfrak{x}}_o$, which will be used through out the next section as the position and orientation references.

IV. Task-parametrized Gaussian mixture model for full pose

Consider a set of task-parameters represented as coordinate systems along with a set of demonstrations that depend on the task-parameters. We use TP-GMM as a statistical approach in combination with dynamical systems to encode different demonstrated actions in an abstract form of mixture of Gaussian components. The task-parameters are the frame of references that matter for each demonstration. The *model parameters* are iteratively estimated with expectation-maximization procedure using the *recorded trajectories* and the corresponding *task-parameters*. Then, during reproduction, the product of linearly transformed *model parameters* given new *task-parameters*, is used to estimate the *new trajectory* through Gaussian mixture regression [6].

In this section, we briefly describe the approach for the situations in which full pose (including position and orientation) can be encoded suitably by using TP-GMM (for more details see [6], [11]).

A. Learning the Model Parameters

The dataset includes both position and orientation attractors (extracted in the previous sections)

$$\boldsymbol{\xi}_n = \begin{bmatrix} \boldsymbol{\xi}_n^{\mathcal{IN}} \\ \boldsymbol{\xi}_n^{\mathcal{OUT}} \end{bmatrix} \ , \ \boldsymbol{\xi}_n^{\mathcal{OUT}} = \begin{bmatrix} \hat{\boldsymbol{x}}_n^p \\ \hat{\mathfrak{x}}_n^o \end{bmatrix},$$

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. For example, in a time-based movement in 3D Cartesian space, $D = 8$ corresponds to aggregation of

[1] Similar to the case of rotation matrices, if we represent two different orientations in quaternion space with $\mathcal{X}_1^o$ and $\mathcal{X}_2^o$, the quaternion that rotates $\mathcal{X}_1^o$ into $\mathcal{X}_2^o$, is given by $\mathcal{X}_2^o * \bar{\mathcal{X}}_1^o$.

time variable ($\boldsymbol{\xi}^{\mathcal{IN}}$, 1 dimension) and Cartesian position (3 dimensions) and unit quaternion orientation (4 dimensions).

The task parameters are P coordinate systems, represented by $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^{P}$ at each time step n. For position data, they correspond to the origin and rotation matrix of the coordinate system i.e. $\boldsymbol{b}_{n,j} \in \mathbb{R}^{3\times1}$ is the origin and $\boldsymbol{A}_{n,j} \in \mathbb{R}^{3\times3}$ is a set of basis vectors. For the quaternion orientation data, $\boldsymbol{b}_{n,j} = \boldsymbol{0}$ is 4×1 zero vector and $\boldsymbol{A}_{n,j} \in \mathbb{R}^{4\times4}$ is the matrix representation of the quaternion orientation (quaternion matrix[2]) 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 P coordinate systems (task-parameters) which forms different trajectories $\{\boldsymbol{X}^{(j)}\}_{j=1}^{P} \in \mathbb{R}^{D\times N}$. At each time step n, this projection can be obtained by a linear transformation as

$$\boldsymbol{X}_n^{(j)} = \boldsymbol{A}_{n,j}^{-1}(\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j}). \tag{5}$$

Based on Eq. (5), the projected position attractor yields from $\boldsymbol{R}_{n,j}^{-1}(\hat{\boldsymbol{x}}_n^p - \boldsymbol{o}_{n,j})$, in which $\boldsymbol{R}$ is the corresponding rotation matrix and $\boldsymbol{o}$ is the origin of the coordinate frame and the projected orientation attractor will be $\boldsymbol{A}_{n,j}^{-1}(\hat{\mathfrak{x}}_n^o - \boldsymbol{b}_{n,j}) = \boldsymbol{\mathcal{Q}}_{n,j}^{-1}\,\hat{\mathfrak{x}}_n^o$, where $\boldsymbol{\mathcal{Q}}$ is the quaternion matrix representation of $\boldsymbol{R}$.

Intuitively speaking, the idea is to observe the movement and the corresponding Gaussian Mixture Model (GMM) in the global frame, from every single frame of reference. The TP-GMM parameters are then, a set of GMMs projected into all of the frames. The learning process 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. These parameters are achieved with an Expectation Maximization (EM) process that iteratively updates the model parameters until convergence.

B. Reproduction

Given a set of task-parameters $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^{P}$ in the reproduction phase, the learned model is used to reproduce the movements in the previous situations (reproducing the demonstrations) or different trajectories with new frame of references. At each time step n, the model first retrieves a temporary GMM by a product of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) \;\propto\; \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j},\; \boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^{\top}\Big). \tag{6}$$

This product of Gaussians can be achieved by the following equations

$$\begin{aligned}
\boldsymbol{\Sigma}_{n,i} &= \Big(\sum_{j=1}^{P} \big(\boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^{\top}\big)^{-1}\Big)^{-1}, \\
\boldsymbol{\mu}_{n,i} &= \boldsymbol{\Sigma}_{n,i} \sum_{j=1}^{P} \big(\boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^{\top}\big)^{-1} (\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j}).
\end{aligned}$$

Based on Eq. (6) in the reproduction phase, the model parameters are first transformed using the new given frames and then the product of them forms the temporary GMM at each time step n.

Given the temporary GMM parameters, Gaussian Mixture Regression (GMR) is then used to retrieve the trajectory. GMR estimates the conditional probability $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{OUT}} | \boldsymbol{\xi}_n^{\mathcal{IN}})$ relying on the joint probability $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{IN}}, \boldsymbol{\xi}_n^{\mathcal{OUT}})$, encoded with GMM parameters (see details in [6]).

By using the dynamical system of Eq. (1) and (3), it is straightforward to reproduce the position and orientation from their attractors, either with the same stiffness and damping gains or different ones.

V. Experiments

The flexible BHA robot with 9 DoF is used in two experiments. The aim of the first experiment is to teach the robot, a time-based point to point end-effector movement. In the second experiment the robot learns to follow an object (a red cup) with its end-effector. This experiment is time-invariant.

The experiments are designed to show the capability of our proposed LfD approach. There are 3 phases for both of them: demonstration, model learning and reproduction phase. The demonstrations are recorded by kinesthetic teaching while the robot is in active compliant control model. During the demonstration, we record the full pose of the BHA end-effector along with the position and orientation of each frame of reference. The recorded end-effector data is smoothed out and re-sampled through out a preprocessing step. Eventually we will have 3 dimensions for the position and 4 dimensions for the orientation expressed in unit quaternion space.

A. Time based point to point movement

The aim of the first experiment is to teach the robot to move its end-effector from a start to an end pose. For each movement, the start and end Cartesian poses are chosen as 2 fixed frame of references. During demonstration, the user is able to slowly move the robot from the start to the end while recording the Cartesian positions and orientation of the end-effector. We recorded an appropriate number of demonstrations (here 8), by using a Vicon system. The Vicon system also collects the complete pose of the start and end frames that are fixed in this experiment. Fig. 2b shows one example of the experiment's setup by which we recorded the demonstrations.

[2] Quaternion matrix consists of the quaternion vector elements i.e. if $\mathfrak{q}_1$ and $\mathfrak{q}_2$ are two quaternions then the quaternion matrix $\boldsymbol{\mathcal{Q}}_1$ is built from the elements of $\mathfrak{q}_1$ so that $\boldsymbol{\mathcal{Q}}_1\mathfrak{q}_2 \equiv \mathfrak{q}_1 * \mathfrak{q}_2$ ($*$ is quaternion product).

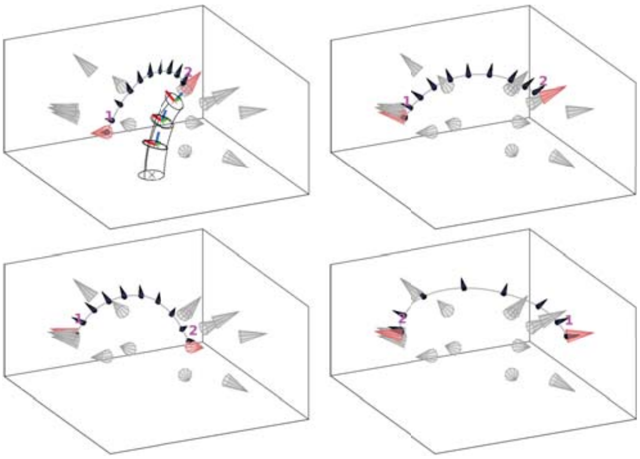


Fig. 3: Demonstrations of point-to-point movements. Here only 4 demonstrations out of 8 were depicted. The gray cones represent the pose of starts and ends. For each demonstration the starts and ends can be distinguished respectively by the pink 1 and 2 and red cones. The gray trajectory and the black cones on it show the pose of the end-effector in some selected intervals. A sample BHA robot is also shown in the top-left figure.

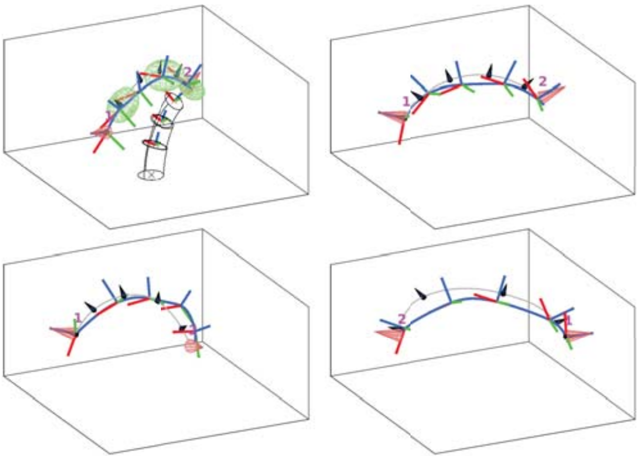


Fig. 4: Reproduction of 4 sample demonstrations in Fig. 3. The TP-GMM model is learned by considering the full pose of the start and end cones as the fixed frame of references (task parameters). The blue line and the frames on them are the reproduced end-effector position trajectories and samples of its orientation. The model is successfully learning and reproducing the full pose of the robot's end-effector. In the top-left figure the green ellipsoids are the retrieved GMM corresponding to the position attractors.

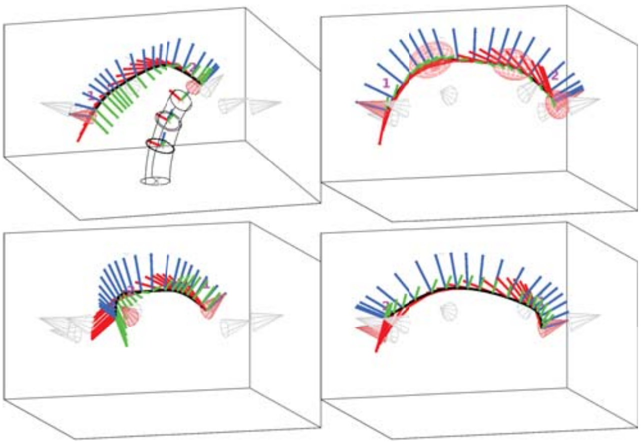


Fig. 5: Sample reproduced position and orientation of the end-effector given new start and end poses. The colored depicted frames show the orientation of the end-effector during the reproduced position trajectory. We have not shown all the frames for all the time steps. In the top-right figure, the red ellipsoids are the retrieved GMM corresponding to the position attractors.

In the time-based recorded dataset, we have 8 dimensions for $\boldsymbol{\xi}$ as

$$\boldsymbol{\xi}_n = \begin{bmatrix} t_n \\ \hat{\boldsymbol{x}}_n^p \\ \hat{\boldsymbol{\mathfrak{r}}}_n^o \end{bmatrix}, \boldsymbol{b}_{n,j} = \begin{bmatrix} 0 \\ \boldsymbol{o}_n^{(j)} \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{A}_{n,j} = \begin{bmatrix} 1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_n^{(j)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{Q}}_n^{(j)} \end{bmatrix}, \quad (7)$$

where $\boldsymbol{o}_n^{(j)}$ is the Cartesian position of the origin of j^{th} frame, $\boldsymbol{0}$ is a 4×1 zero vector. $\boldsymbol{R}_n^{(j)}$ and $\boldsymbol{\mathcal{Q}}_n^{(j)}$ are respectively the rotation matrix and quaternion matrix representation of the orientation of j^{th} frame.

We empirically chose 3 Gaussian components in the model. The stiffness and damping gains in Eq. (1) and (3) were set to $k^{\mathcal{P}} = 500$, $k^{\mathcal{V}} = 50$ and $k^{\mathcal{O}} = 250$, $k^{\mathcal{W}} = 25$ respectively, to keep the dynamical system close to an over-damped situation.

Fig. 3 shows 4 sample demonstrations and the corresponding fixed start and end poses by gray cones. A TP-GMM is used to encode the recorded pose trajectories given the poses of the start and end points as fixed frame of references. Fig. 4 shows how the learned TP-GMM model reproduces the end-effector poses along the trajectory, given the same pair of frames. The retrieved GMM has been plotted in the top-left figure as well. Note that since the references are fixed though out the experiment, the retrieved GMMs are the same for every time step.

The generalization capability of TP-GMM approach was examined successfully by providing different poses of start and end points. In our experiment the proposed model was able to produce suitable and smooth movements between the points. Fig. 5 shows the qualitative results of 4 sample reproduction, given new pairs of frames of references.

Here we provided the results only in simulation but in the real experiment, to move the BHA robot, a previously developed inverse kinematics model based on constant-curvature model [19] can be used to transform the Cartesian poses into the robot's joint space trajectory. We modified the iterative inverse kinematics model to have more tendency to fit the position of the end-effector rather than the orientation by exploiting null-space of the robot. This is done by searching more in the null-space of the robot to satisfy the orientation. The real robot experiment can be carried out as a ball in the basket or fruit picking scenario.

B. Time-invariant movement

In the second experiment, we want to demonstrate to the robot to follow the position and orientation of a flying object (a cup, shown in Fig. 6 by blue cones) in the work-space by its end-effector. Time was used as the input in the previous experiment. However using the TP-GMM, any other type

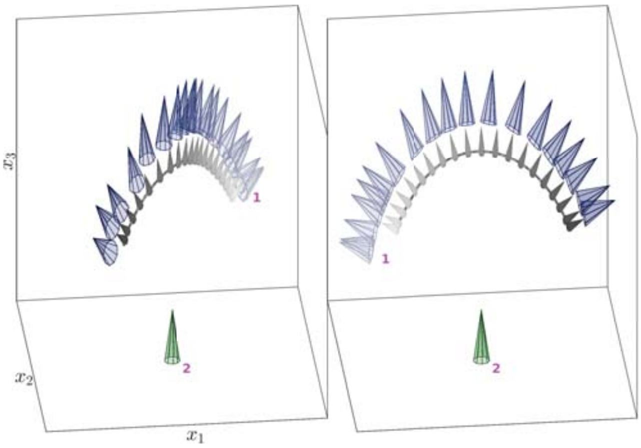


Fig. 6: Two samples of moving frame during demonstration: Frame 1 (blue cones) is a moving frame that its position and orientation should be followed by the pose of the robot's end-effector. The moving direction is shown by dark cones. Frame 2 (green cone) has a fixed pose at the base of the robot. The gray small cones show how the end-effector pose follows the pose of the second frame.

of inputs can be used to derive the movement [6]. Since the robot should follow the pose of the cup, we choose the position of the cup in the 3D Cartesian space as the input by which we can learn a time-invariant task. In this experiment, we defined 2 frames of references. The first frame is the moving cup (blue cones) and the second frame is the fixed base of the robot. We define the dataset and the first frame as follow

$$\boldsymbol{\xi}_n = \begin{bmatrix} \boldsymbol{o}_n^{rc} \\ \hat{\boldsymbol{x}}_n^{p} \\ \hat{\boldsymbol{\mathfrak{x}}}_n^{o} \end{bmatrix}, \boldsymbol{b}_{n,1} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{o}_n^{rc} \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{A}_{n,1} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_n^{rc} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mathcal{Q}}_n^{rc} \end{bmatrix}, \tag{8}$$

where, the position of the cup $\boldsymbol{o}_n^{rc}$ has been replaced by time. $\boldsymbol{R}_n^{rc}$ and $\boldsymbol{\mathcal{Q}}_n^{rc}$ are respectively, the rotation matrix and quaternion matrix representation of the orientation of the first frame (here the moving frame). In the above equation, $\boldsymbol{\xi}_n \in \mathbb{R}^{10\times1}$, $\boldsymbol{R}_n^{rc} \in \mathbb{R}^{3\times3}$, $\boldsymbol{\mathcal{Q}}_n^{rc} \in \mathbb{R}^{4\times4}$ and $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. The vectors $\boldsymbol{0}$ have suitable dimension. The second frame is always fixed at the base of the robot, $\boldsymbol{b}_{n,2} = \boldsymbol{0} \in \mathbb{R}^{10\times1}$ and $\boldsymbol{A}_{n,2} = \boldsymbol{I} \in \mathbb{R}^{10\times10}$.

Fig. 6 shows 2 sample demonstrations and frames. The demonstrations were recorded by kinesthetic teaching while the BHA robot was in the active compliant control mode i.e. it is easy for the user to reconfigure the robot and follow the pose of the red cup. During the demonstration, the cup was moving randomly in the space with one user and another user tries to keep the robot's end-effector always close to the cup. In Fig. 6 the blue cones show the movement of the cup in the robot's workspace. We collected 6 demonstrations for this experiment. The second fixed frame is visualized by the green cone fixed at the origin.

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 demonstration data and not given to the model explicitly. During reproduction, given the position and orientation of the two frames, the model retrieves the full pose of the end-effector. The inverse kinematic model of the BHA robot is then used to obtain the joint variables (lengths of the 3 modules).

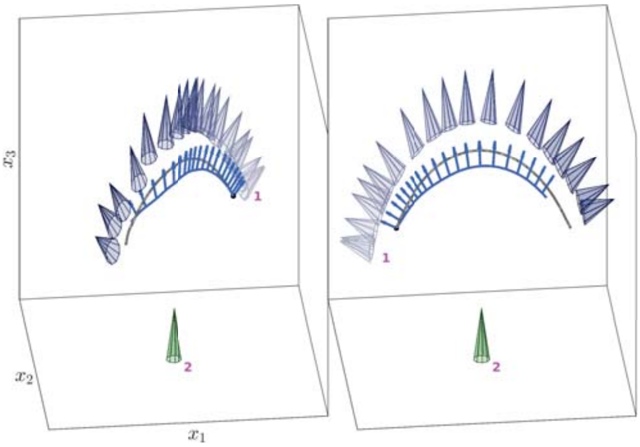


Fig. 7: Reproduction of the sample demonstrations by using TP-GMM. Here, the movement of the first frame is presented to the robot and the encoded model retrieves the pose of the end-effector (blue line and arrows).

Fig. 7 depicts the reproduced pose for the end-effector using the learned model. Here, for the moving frame, the same movement as Fig. 6 has been considered. The trajectory and the corresponding orientation are shown by blue lines and arrows for some of the time instances. The difference between the pose of the end-effector and the moving frame at the end of the movement is due to the fact that we have plotted only the first 100 time instances. The dynamical system makes the movement more compliant but also a bit delayed i.e. the positions and orientations follow their attractors slowly according to the stiffness and damping gains.

We examined the generalization capability of the learned model by proving different poses for the moving frame (the cup in the real experiment). Fig. 8, shows 2 new situations 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 the end-effector is shown by the black line on which the reproduced orientations have been shown by blue arrows.

VI. Conclusion and Future Works

In this paper, a practical approach that enables us to apply kinesthetic teaching on a soft pneumatic robot was suggested and tested. To the best of the author's knowledge, nobody has tried to do any task-related teaching on a real soft/continuum robot. The method is based on an active compliant controller that has been developed for this robot. Similar controller can be exploited on other soft robots such as STIFF-FLOP robot [1] while providing demonstrations is necessary like in [3].

A task-parametrized probabilistic model was used as a learning from demonstration algorithm to encode and retrieve both the position and the orientation of the end-effector.

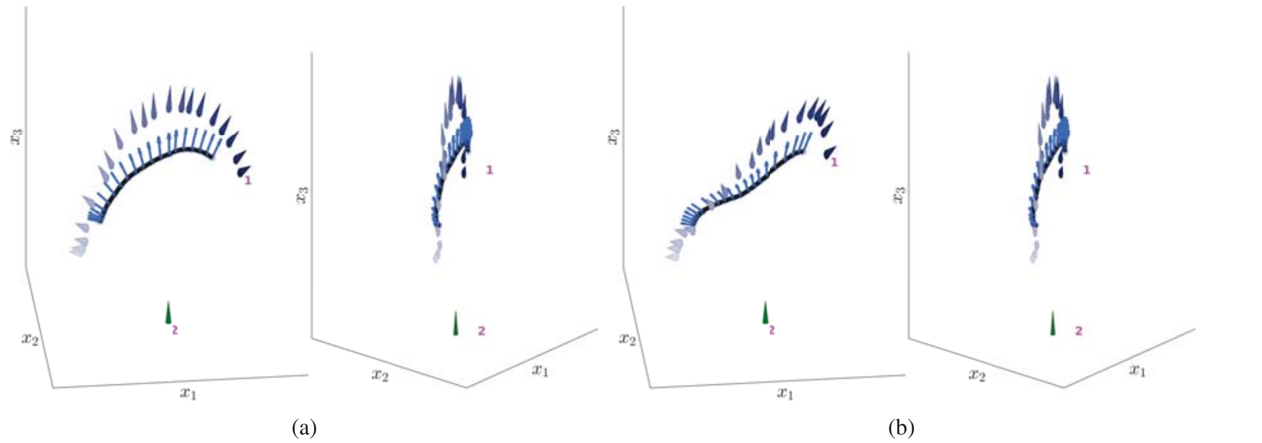


Fig. 8: Two different movements, each one has been shown from 2 angles. The gray cones are the position and orientation of the moving frame that the robot's end-effector (the blue line and arrow) is able to follow.

Two time-dependent and time-independent experiments were conducted to test the ability of the proposed method. The demonstrations were recorded by using a real BHA robot, whilst the reproduction results were shown in simulation. However in the future, we aim to design more complex real-world experiments.

Also we want to evaluate the approach more quantitatively, specially by analyzing the retrieved covariance matrices during the reproduction phase. The demonstrations were formulated by using dynamical systems. We are also interested in applying different stiffness and damping gains in the reproduction phase.

VII. Acknowledgements

This work is funded by the European Community's Horizon 2020 robotics program ICT-23-2014 under grant agreement 644727 - CogIMon. J. Queißer received funding from the Cluster of Excellence 277 Cognitive Interaction Technology and has been supported by the CODEFROR project (FP7-PIRSES-2013-612555).

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*, vol. 1, no. 2, pp. 122–131, 2014.

[2] 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)*, pp. 573–579, IEEE, 2014.

[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*, vol. 116, pp. 81–96, September 2014. Special issue on new methods of human-robot interaction in medical practice.

[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*, vol. 1, pp. 1–15, October 2014.

[5] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. Intl Conf. on Machine Learning (ICML)*, 2004.

[6] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[7] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems* (J. D. Cowan, G. Tesauro, and J. Alspector, eds.), vol. 6, (Burlington, MA, USA), pp. 120–127, Morgan Kaufmann Publishers, Inc., 1994.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77:2, pp. 257–285, February 1989.

[9] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.

[11] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, (Hamburg, Germany), Sept.-Oct. 2015.

[12] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, (Hong Kong, China), pp. 3339–3344, May-June 2014.

[13] K. J. Korane, "Robot imitates nature," *Machine Design*, vol. 82, no. 18, pp. 68–70, 2010.

[14] A. Grzesiak, R. Becker, and A. Verl, "The Bionic Handling Assistant - A Success Story of Additive Manufacturing," *Assembly Automation*, vol. 31, no. 4, pp. 329–333, 2011.

[15] M. Rolf, K. Neumann, J. Queier, F. Reinhart, A. Nordmann, and J. J. Steil, "A Multi-Level Control Architecture for the Bionic Handling Assistant," *Advanced Robotics*, vol. 29, no. 13: SI, pp. 847–859, 2015.

[16] M. Rolf and J. Steil, "Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 3440–3446, 2012.

[17] F. Reinhart and J. J. Steil, "Hybrid Mechanical and Data-driven Modeling Improves Inverse Kinematic Control of a Soft Robot," 2016.

[18] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pp. 2997–3004, IEEE, 2014.

[19] M. Rolf and J. Steil, "Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2012.