Learning the stiffness of a continuous soft manipulator from multiple demonstrations ⋆

Danilo Bruno[1], Sylvain Calinon[1,2], Milad S. Malekzadeh[1], and Darwin G. Caldwell[1]

[1] Department of Advanced Robotics - Istituto Italiano di Tecnologia (IIT)
Via Morego, 30 - 16163 Genova
[2] Idiap Research Institute - Rue Marconi 19 - CH-1920 Martigny, Switzerland

Abstract. Continuous soft robots are becoming more and more widespread in applications, due to their increased safety and flexibility in critical applications. The possibility of having soft robots that are able to change their stiffness in selected parts can help in situations where higher forces need to be applied. This paper describes a theoretical framework for learning the desired stiffness characteristics of the robot from multiple demonstrations. The framework is based on a statistical mathematical model for encoding the motion of a continuous manipulator, coupled with an optimal control strategy for learning the best impedance parameters of the manipulator.

1 INTRODUCTION

The use of soft robots is becoming crucial to perform tasks where the contact with a delicate environment is needed. This is happening for surgical robots as well as manipulators that need to be put in contact with humans [1–4].

The possibility of changing the stiffness of soft robots can become a desired feature where the softness of the robot prevents it from performing some task, such as moving objects or carrying weights. In this case, the robot should be aware of the stiffness that is needed to perform the task and apply the correct values where needed.

Moreover, the choice of using soft materials usually comes together with a different kind of embodiment, that departs from rigid kinematic chains. Usually soft robots are described as continuous robots and their kinematic/dynamic description needs to take this characteristics into account.

In this paper we will present a mathematical framework that can be employed to learn the impedance characteristics of a soft continuous manipulator from demonstrations. The paper is based on the main idea that the robot needs to increase its stiffness whenever the task needs a higher precision. This is true, for instance, in transportation tasks, where the relative position between the carried

⋆ This work was partially supported by the STIFF-FLOP European project under contract FP7-ICT-287728.

object and the end-effector of the robot is invariant during the crucial part of the task [5].

Within the learning from demonstrations scenario, the user providing demonstrations shows the task to perform several times: the consistency of the demonstrations is encoded into a statistical model and the information extracted back at reproduction time allows us to select the appropriate stiffness behaviour [6, 7].

In this paper, this feature is further enhanced by the use of a mathematical model handling the movement of a continuous robot. In this way, the model can be used to learn the correct stiffness behaviour of different parts along the continuous robot at different times.

The selection of the correct values of stiffness is performed by using optimal control techniques, that allow us to learn a positional controller along the manipulator that makes a trade off between minimizing the forces applied on the manipulator and tracking the demonstrations within the demonstrated variability [8].

The current paper is aimed at showing the general theoretical framework for learning the stiffness behaviour of a continuous robot and is organized as follows. In section 2 we present a generative model to learn motion skills from demonstrations. In particular, we show how a tracking controller can be learnt from demonstrations by using optimal control technique. Section 3 extends the model developed for Section 2 for encoding the motion of a continuum robot. Section 4 shows how those results can be effectively employed to learn the desired stiffness of the manipulator from demonstrations.

2 LEARNING MOTION SKILLS FROM DEMONSTRATIONS

2.1 GAUSSIAN MIXTURE MODELS

The setup for learning motion skills from demonstrations presented in this section is based on the construction of a statistical model of observed movements (demonstrations) given by a user.

The observations $\{\boldsymbol{\xi}_n\}_{n=1}^{N}$ representing the points of the demonstrations are assumed to be independent realizations of a random vector and is assumed to be distributed as a linear combination of Normal distributions as:

$$\mathcal{P}(\boldsymbol{\xi_n}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi_n}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

with

$$\mathcal{N}(\boldsymbol{\xi_n}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{\xi_n}-\boldsymbol{\mu}_k)^{\top}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\xi_n}-\boldsymbol{\mu}_k)\right].$$

The parameters of a *Gaussian mixture model* (GMM) with K components are thus defined by $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where π_k is the prior (mixing coefficient), $\boldsymbol{\mu}_k$ is the center, and $\boldsymbol{\Sigma}_k$ is the covariance matrix of the k-th mixture component.

The estimation of mixture parameters can be performed by maximizing the log-likelihood of the above distribution of the given dataset. This leads to an *expectation-maximization* (EM) process iteratively refining the model parameters to converge to a local optimum of the likelihood. These two steps are iteratively applied until a stopping criterion is satisfied. The two steps are described below.

E-step:

$$h_{n,i} = \frac{\pi_i \, \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}.$$

M-step:

$$\begin{aligned}
\pi_i &= \frac{\sum_{n=1}^{N} h_{n,i}}{N}, \\
\boldsymbol{\mu}_i &= \frac{\sum_{n=1}^{N} h_{n,i} \boldsymbol{\xi}_n}{\sum_{n=1}^{N} h_{n,i}}, \\
\boldsymbol{\Sigma}_i &= \frac{\sum_{n=1}^{N} h_{n,i} (\boldsymbol{\xi}_n - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_n - \boldsymbol{\mu}_i)^\top}{\sum_{n=1}^{N} h_{n,i}}.
\end{aligned}$$

The reproduction of an average movement or skill behavior can be formalized as a statistical regression problem. We demonstrated in previous work that Gaussian Mixture Models in combination with *Gaussian mixture regression* (GMR) offers a simple and elegant solution to handle encoding, recognition, prediction and reproduction in robot learning [5, 9]. It provides a probabilistic representation of the movement, where the model can retrieve actions in real-time, within a computation time that is independent of the number of datapoints in the training set.

By defining which variables span for input and output parts (noted respectively by $\mathcal{I}$ and $\mathcal{O}$ superscripts), a block decomposition of the datapoint $\boldsymbol{\xi}_n$, vectors $\boldsymbol{\mu}_i$ and matrices $\boldsymbol{\Sigma}_i$ can be written as

$$\boldsymbol{\xi}_n = \begin{bmatrix} \boldsymbol{\xi}_n^I \\ \boldsymbol{\xi}_n^O \end{bmatrix}, \quad \boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^I \\ \boldsymbol{\mu}_i^O \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^I & \boldsymbol{\Sigma}_i^{IO} \\ \boldsymbol{\Sigma}_i^{OI} & \boldsymbol{\Sigma}_i^O \end{bmatrix}.$$

The GMM thus encodes the joint distribution $\mathcal{P}(\boldsymbol{\xi}^I, \boldsymbol{\xi}^O) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ of the data $\boldsymbol{\xi}$. At each reproduction step n, $\mathcal{P}(\boldsymbol{\xi}_n^O | \boldsymbol{\xi}_n^I)$ is computed as the condi-

tional distribution

$$\mathcal{P}(\boldsymbol{\xi}_n^O|\boldsymbol{\xi}_n^I) \sim \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\, \mathcal{N}\Big(\hat{\boldsymbol{\mu}}_i^O(\boldsymbol{\xi}_n^I), \hat{\boldsymbol{\Sigma}}_i^O\Big), \tag{1}$$

$$\text{with} \quad \hat{\boldsymbol{\mu}}_i^O(\boldsymbol{\xi}_n^I) = \boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI}\boldsymbol{\Sigma}_i^{I^{-1}}(\boldsymbol{\xi}_n^I - \boldsymbol{\mu}_i^I), \tag{2}$$

$$\hat{\boldsymbol{\Sigma}}_i^O = \boldsymbol{\Sigma}_i^O - \boldsymbol{\Sigma}_i^{OI}\boldsymbol{\Sigma}_i^{I^{-1}}\boldsymbol{\Sigma}_i^{IO},$$

$$\text{and} \quad h_i(\boldsymbol{\xi}_n^I) = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_n^I|\, \boldsymbol{\mu}_i^I, \boldsymbol{\Sigma}_i^I)}{\sum_k^K \pi_k \mathcal{N}(\boldsymbol{\xi}_n^I|\, \boldsymbol{\mu}_k^I, \boldsymbol{\Sigma}_k^I)}. \tag{3}$$

In the general case, eq. (1) represents a multimodal distribution. In problems where a single output is expected (single peaked distribution), eq. (1) can be approximated by a single normal distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_n^O, \hat{\boldsymbol{\Sigma}}_n^O)$ with parameters

$$\hat{\boldsymbol{\mu}}_n^O = \sum_i h_i(\boldsymbol{\xi}_n^I)\left[\boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI}\boldsymbol{\Sigma}_i^{I^{-1}}(\boldsymbol{\xi}_n^I - \boldsymbol{\mu}_i^I)\right],$$

$$\hat{\boldsymbol{\Sigma}}_n^O = \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\boldsymbol{\Sigma}_i^O + \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\boldsymbol{\mu}_i^O(\boldsymbol{\mu}_i^O)^\top - \hat{\boldsymbol{\mu}}_n^O\hat{\boldsymbol{\mu}}_n^{O^\top}. \tag{4}$$

Eq. (4) is computed in real-time from the model parameters. The retrieved signal encapsulates variation and correlation information in the form of a probabilistic flow tube, see e.g., [10].

2.2 LQR REPRESENTATION FOR MOTION SKILLS

One typical scenario consists in collecting time and position values of the demonstrations and then extract the resulting movement by choosing time as input. This allows us to reproduce the average behaviour extracted from the demonstrations. This approach implements an open-loop controller which is not robust to perturbations.

In order to cope with this drawback, the above approach can be complemented by coupling the reproduction step with a second order linear dynamical system, which improves robustness and smoothness of the reproductions [7]. Moreover the stiffness and damping parameters characterizing the second order system can also be estimated from the observation by following a reasoning similar to the development of linear quadratic regulators [11].

The approach exploits Gaussian conditioning to retrieve a reference trajectory in the form of a full distribution $\mathcal{N}(\hat{\boldsymbol{\xi}}_t^O, \hat{\boldsymbol{\Sigma}}_t^O)$ varying at each time step t, see [8] for an experiment with a 7 DOFs manipulator.

Similarly as the solution proposed by Medina *et al.* in the context of risk-sensitive control for haptic assistance [12], the predicted variability can be exploited to form a minimal intervention controller [13]. An acceleration command

$$\boldsymbol{u}_t = \hat{\boldsymbol{K}}_t^P(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) - \hat{\boldsymbol{K}}_t^V\dot{\boldsymbol{x}}_t \tag{5}$$

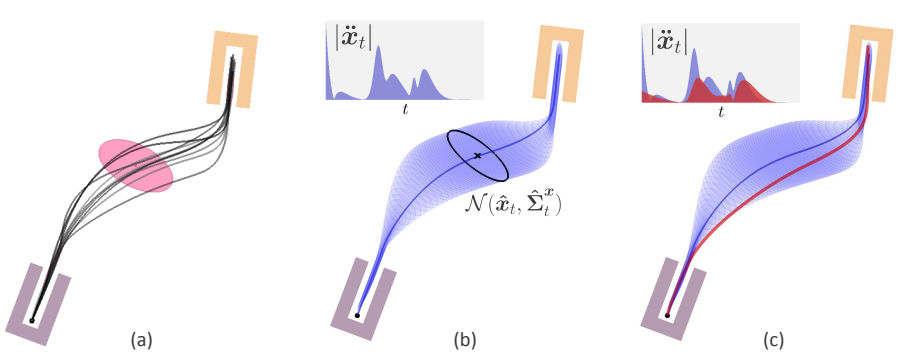

Fig. 1. Minimal intervention controller based on a linear quadratic regulator (LQR). *(a)* Multiple demonstrations. *(b)* Reproduction of the reference path defined as the centers of the Gaussians retrieved by GMR, with the corresponding acceleration profile. *(c)* Reproduction with a linear quadratic regulator, reducing the accelerations and jerks while keeping the movement within a boundary defined by the demonstrations.

is used to control the robot, with $\hat{\boldsymbol{x}}_t$ estimated by GMR. $\hat{\boldsymbol{K}}_t^P$ and $\hat{\boldsymbol{K}}_t^V$ are full stiffness and damping matrices estimated by a *linear quadratic regulator* (LQR) with time-varying weights. For a *finite horizon LQR*, this is achieved by minimizing the cost function

$$c^{(1)} = \sum_{t=1}^{T} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t)^\top \boldsymbol{Q}_t (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) \; + \; \boldsymbol{u}_t^\top \boldsymbol{R}_t \; \boldsymbol{u}_t, \tag{6}$$

subject to the constraints of a double integrator system. This cost function aims at finding an optimal feedback controller minimizing simultaneously the tracking errors and the control commands in proportion defined by weighting matrices $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$.

The solution can be computed by backward integration of a *Riccati ordinary differential equation* with varying full weighting matrix $\boldsymbol{Q}_t = (\hat{\boldsymbol{\Sigma}}_t^{\boldsymbol{x}})^{-1}$ estimated by GMR and by setting $\boldsymbol{R}_t$ as a constant diagonal matrix. It provides a time-varying feedback control law in the form of Eq. (5) with full stiffness and damping matrices $\hat{\boldsymbol{K}}_t^P$ and $\hat{\boldsymbol{K}}_t^V$.

To solve the above minimization problem, a boundary condition needs to be set on the final feedback term, which can for example be set to zero. It is in this case assumed that the robot comes back to a compliant state after the task is fulfilled.

In some situations, it might be computationally expensive to recompute at each iteration t a prediction on the remaining movement. An approximation that we exploited in [8] can in this case be locally computed by considering an *infinite horizon LQR* formulation to estimate a feedback term at iteration t by considering only the current estimate $\hat{\boldsymbol{\Sigma}}_t^{\boldsymbol{x}}$. This corresponds to the estimation

of a feedback controller that does not know in advance whether the precision at which it should track a target will vary. The corresponding cost function at iteration t corresponds to

$$c_t^{(2)} = \sum_{n=t}^{\infty} (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_n)^\top \boldsymbol{Q}_t (\hat{\boldsymbol{x}}_t - \boldsymbol{x}_n) \; + \; \boldsymbol{u}_n^\top \boldsymbol{R}_t \, \boldsymbol{u}_n, \quad \forall t \in \{1, \dots, T\} \tag{7}$$

which can be solved iteratively through the *algebraic Riccati equation*, providing an optimal feedback controller in the form of Eq. (5) with full stiffness and damping matrices $\hat{\boldsymbol{K}}_t^P$ and $\hat{\boldsymbol{K}}_t^V$.

3 REPRESENTATION OF MOTION IN A CONTINUUM ROBOT

In order to extend the above strategy to a continuous robot, a mathematical model of the latter needs to be developed. The model that we exploit in this paper was inspired by the biological studies carried out on the motion of the octopus. In fact, biologists use a similar formulation to describe the evolution of the curvature and elongation of the Octopus arm during the movement [14, 15].

Following the aforementioned studies, a generalization of the previous approach that is more suitable for robotics applications was proposed in [16]. The setup exploits the biologically inspired representation within a statistical framework scenario and allows us to exploit the instruments of Machine Learning to build a statistical model that can be effectively used to reproduce skills on continuous robotics arms.

For this purpose, the continuum robotic arm is approximated as a robot with a high number of links: it can be described as a kinematic chain with a high number of revolute joints alternated with prismatic joints describing the local elongation. The joint space of the robot is described by a collection of 3 Euler angles and a scalar offset for each link, where the forward kinematics can be evaluated by using standard robotics techniques.

The motion of the system can be represented by assigning the set of 4 joint variables for each discrete position along the arm. This is performed by defining a continuous *arm-index* $s \in [0, 1]$, representing the position of any point along an arm, with $s = 0$ and $s = 1$ representing respectively the base and the tip. We will rescale the duration of a movement so that it can be represented by a continuous *time index* $t \in [0, 1]$. We thus have 3 Euler angles $\theta_x, \theta_y, \theta_z$ and an offset ΔL for each t and s, that can be represented as a set of surfaces as shown in Fig. 2.

The original raw data consist of noisy Cartesian positions of selected points along the arm. A preprocessing step is performed by resampling and smoothing through a two-dimensional polynomial fitting (surface fitting) with a 7-degree polynomial. The degree for the polynomial is set experimentally by testing different orders.

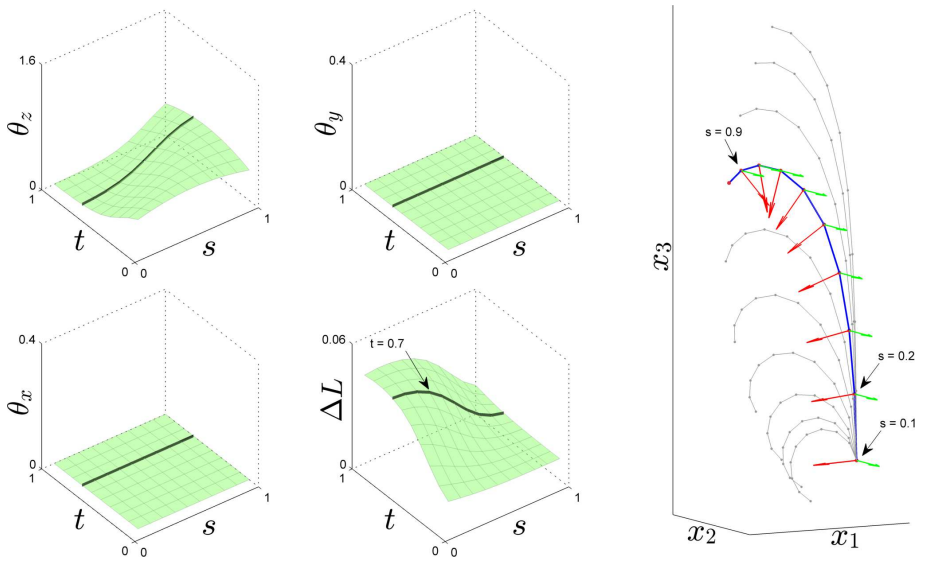

Fig. 2. Spatiotemporal representation. *Left:* The black lines represent slices in the Euler angles surfaces $\boldsymbol{\theta}$ and offset surface $\boldsymbol{\Delta L}$, corresponding to a static pose described by all the links ($0 \leq s \leq 1$) at $t = 0.7$. *Right:* For the same time frame, some of the corresponding Frenet frames along the arm are depicted in a 3D Cartesian space.

In order to represent the motion in the form of a statistical model, the pre-processed data are encoded into a Gaussian Mixture Model, representing the distribution of the joint variables for each value of s and t as shown in Fig.3.

Now, we can extend the dynamical system scenario to the continuous robot, by extending the concept of trajectory attractor to a surface attractor (generalization of the generic spring-damper system to a spatiotemporal dynamical system). We use both arm-index s and time t (rather than only time) as input variables, which enables the approach to encode the movement of the whole arm with a compact model (namely, by encoding the movement of all points along the arm).

A dynamical system can then be derived for the current situation in two different ways, either by integrating along the t index or along the s index. In the current scenario, the evolution is only semi bi-dimensional, in the sense that the surface is constructed by integrating either in the t direction from a given curve at $t = 0$, representing the initial shape or, alternatively, by integrating in the s direction from a given curve at $s = 0$, representing the evolution of a single point (e.g. the base of the continuous robot).

The difference is illustrated in Fig.4. The surface attractor allows us to evaluate at each time step the desired pose of the continuum robot. The dynamical system in time describes the motion of each single link along with the impedance parameters of the positional controller. On the other hand, the dynamical system in s allows us to evaluate at each time step the static pose of the robot.

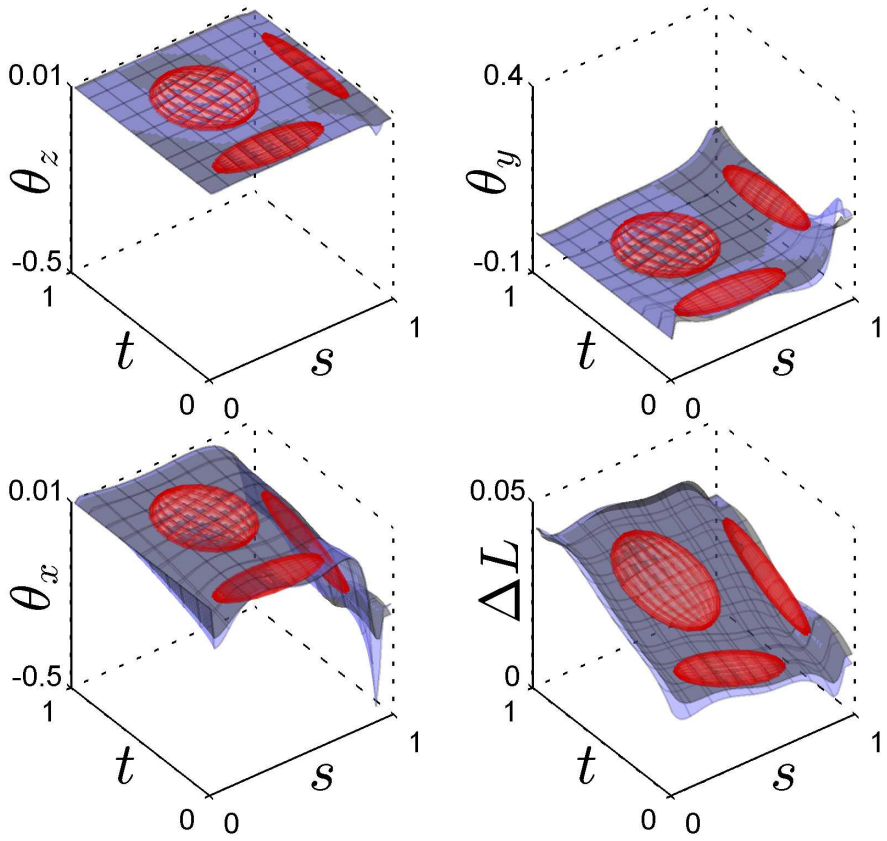

Fig. 3. The Gaussian Mixture Model encoding a movement for the joint variables representing the continuum robot. The number of components is chosen heuristically.

The picture shows the learnt stiffness values with colors from white to black depicting increasing levels of stiffness

4 LEARNING IMPEDANCE PARAMETERS

In the following section we will concentrate on the use of the above described scenario for learning impedance parameters from demonstrations. First of all a distinction should be made between the t-directed and the s-directed dynamical system. In the first case, the dynamical system implemented by eq.(5) represents the acceleration applied to the points of the continuous robot and can be interpreted as the impedance parameters of a positional controller implemented on the system.

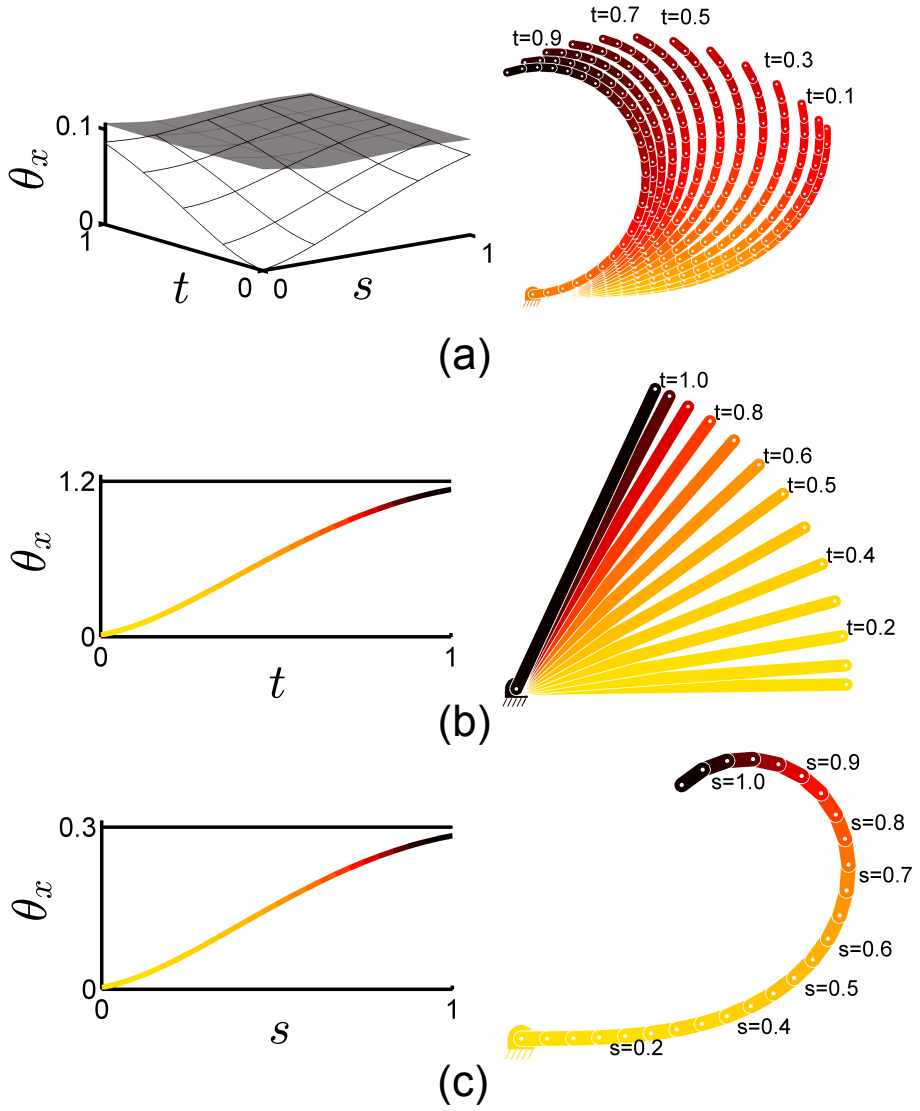

Fig. 4. A dynamical system with surface attractor (evolution over time and arm-index). *(a)* The grey surface on the left represents the attractor surface corresponding to the observed motion of the continuum arm, while the white surface is the reproduced motion of the arm. The right figure shows the arm configurations in 2D Cartesian space at different time steps. *(b)* The left figure shows the evolution in time of a link with a given arm-index s along the kinematic chain and the corresponding configurations (on the right). *(c)* The figures show the pose of the continuum arm for a given time step t. The evaluated stiffness values are represented by different colors (black means high stiffness).

On the other hand, the s-directed dynamical system has a different meaning. In this case, eq.(5) represents the second derivative with respect to a spatial variable and can be related to the bending and elongation properties of the system. In particular the stiffness matrix represents the resistance to bending and elongation of the continuum robot. For this reason, the described scenario can be effectively employed to learn from demonstrations what is the optimal stiffness of the different parts of the manipulator along the task.

As an example of scenario, we show how this mechanism works for a simple motion. The demonstration phase consists of moving the end effector of the manipulator to reach a target in front of it. The end-effector of the manipulator is moved around in the first part of the movement with a high variability (exploration phase). In the second part of the task, the robot is instead moved directly towards the target. As a result, the robot learns a different controller for the different parts of the task, as shown in Fig.5.

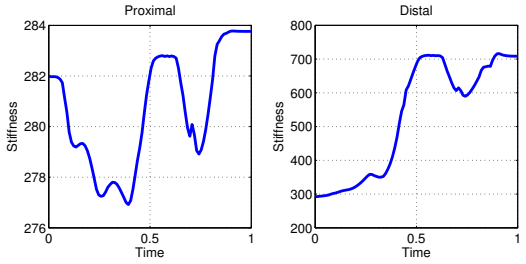

Fig. 5. Desired stiffness of the robot during the task. The desired stiffness for the proximal and distal parts of the manipulator are shown on the left and right respectively. As time increases, the manipulator moves towards a target placed in front of it.

In the first part of the task, where the demonstrated variability is higher, the robot learns that a lower stiffness is needed to complete the task. The value of the stiffness is instead increasing when the distal part of the manipulator enters in the area of low variability, towards the end of the movement.

5 CONCLUSION AND FURTHER WORK

In this paper we presented a theoretical framework for learning stiffness properties of continuous soft robots. The presented framework is based on a compact statistical representation of the continuous arm movement encoding the full motion of all the points of the arm. An optimal control strategy is employed to exploit the statistical model to learn a dynamical system representation of the motion of the system, whose impedance parameters are exploited to single out the stiffness of selected parts of the continuous manipulator.

One drawback of the current encoding and regression process is that it requires the multiple demonstrations samples to be rescaled and aligned along the

temporal index and along the arm index. Further work will investigate the combination of model predictive control with mixture of Gaussians as an approach that could generate a controller for the continuum robot while by-passing this data pre-processing phase.

References

1. A. Jiang, G. Xynogalas, P. Dasgupta, K. Althoefer, and T. Nanayakkara, "Design of a variable stiffness flexible manipulator with composite granular jamming and membrane coupling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2922–2927, 2012.
2. A. Jiang, A. Ataollahi, K. Althoefer, P. Dasgupta, and T. Nanayakkara, "A variable stiffness joint by granular jamming," in *ASME Intl Design Engineering Technical Conf. & Computers and Information in Engineering Conf. (IDETC/CIE)*, pp. 267–275, 2012.
3. M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, C. Laschi, and A. Menciassi, "STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3567–3581, 2013.
4. 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.
5. S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, pp. 44–54, June 2010.
6. 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.
7. S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, (Osaka, Japan), pp. 323–329, 2012.
8. 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.
9. S. Calinon and A. G. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proc. Intl Conf. on Machine Learning (ICML)*, (Bonn, Germany), pp. 105–112, August 2005.
10. D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2, pp. 115–131, 2011.
11. K. J. Astrom and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton University Press, 2008.
12. J. R. Medina, D. Lee, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *IEEE Intl Conf. on Robotics and Automation (ICRA)*, pp. 1025–1031, May 2012.
13. E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature Neuroscience*, vol. 5, pp. 1226–1235, 2002.

14. T. Flash and B. Hochner, "Motor primitives in vertebrates and invertebrates," *Current opinion in neurobiology*, vol. 15, no. 6, pp. 660–666, 2005.
15. I. Zelman, M. Titon, Y. Yekutieli, S. Hanassy, B. Hochner, and T. Flash, "Kinematic decomposition and classification of octopus arm movements," *Frontiers in Computational Neuroscience*, vol. 7, no. 60, 2013.
16. 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.