Reproduction of Human Demonstrations with a Soft-Robotic Arm based on a Library of Learned Probabilistic Movement Primitives

Paris Oikonomou, Athanasios Dometios, Mehdi Khamassi, Costas Tzafestas

Reproduction of Human Demonstrations with a Soft-Robotic Arm based on a Library of Learned Probabilistic Movement Primitives

Paris Oikonomou[1], Athanasios Dometios[1], Mehdi Khamassi[1,2] and Costas S. Tzafestas[1]

***Abstract*— In this paper we introduce a novel technique that aims to control a two-module bio-inspired soft-robotic arm in order to qualitatively reproduce human demonstrations. The main idea behind the proposed methodology is based on the assumption that a complex trajectory can be derived from the composition and asynchronous activation of learned parameterizable simple movements constituting a knowledge base. The present work capitalises on recent research progress in Movement Primitive (MP) theory in order to initially build a library of Probabilistic MPs (ProMPs), and subsequently to compute on the fly their proper combination in the task space resulting in the requested trajectory. At the same time, a model learning method is assigned with the task to approximate the inverse kinematics, while a replanning procedure handles the sequential and/or parallel ProMPs' asynchronous activation. Taking advantage of the mapping at the primitive-level that the ProMP framework provides, the composition is transferred into the actuation space for execution. The proposed control architecture is experimentally evaluated on a real soft-robotic arm, where its capability to simplify the trajectory control task for robots of complex unmodeled dynamics is exhibited.**

I. Introduction

Elderly population tends to increase according to World Health Organization's research on health and aging [1]. In the frames of the I-SUPPORT project (EU H2020 grant agreement no. 643666), a modular robotic system based on a soft-robotic arm (Fig. 1) was developed to support non-autonomous elderlies to independently complete various bathing tasks. Such an interactive bathing application is demanding in terms of safety since it involves human-robot interaction. Hence, research effort was focused on soft robots [2] with inherent or structural compliance, which gives them the ability to actively interact with the environment with drastically reduced risks of injuries. Many continuum manipulators have already been presented with tendon [3] or pneumatic actuation [4], or a combination of those [5]. Most of these robots have a complex mechanical and actuation structure, and require sophisticated kinematic analysis and control schemes. To address these issues, analytic kinematic models based on constant curvature assumption have been established [6], and powerful control strategies for continuum manipulators are still being developed [7].

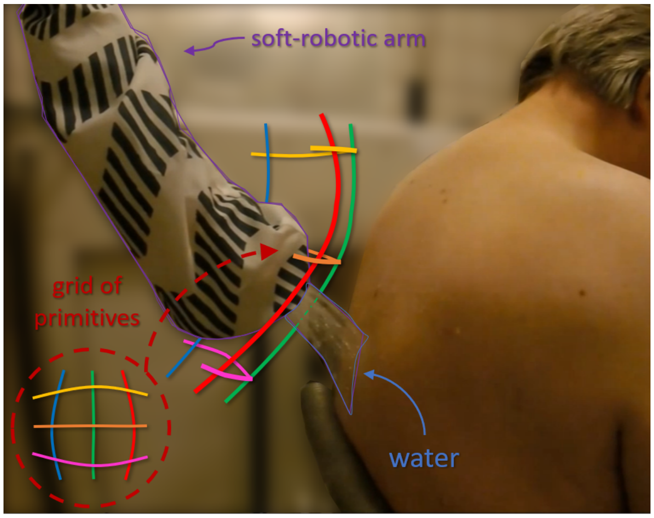

Fig. 1: The soft-robotic arm covered with nylon ensuring water resistance, while performing water pouring (part of the showering task) in a clinical environment. The colored lines depict the mean value of the learned ProMPs in the task space, covering the desired subspace of the robot's workspace. A projection of the grid of primitives on the human back is also illustrated at the bottom left.

In the context of an assistive bathing robot, learning of bathing motions from expert's demonstration is required to ensure the execution of proper washing actions in a human-friendly way. Complicated movements could be seen as a composition of fundamental building blocks called primitives, which is executed either in sequence, or with partial or complete overlap. The fusion of such primitive actions with different parameters (e.g. duration, amplitude etc.) can reproduce more delicate and human-friendly actions. An interactive version of Dynamic Movement Primitives was proposed in [8], combined with a vision-based controller for the adaptation of demonstrated washing actions on the user's body parts. However, this work addresses the problem only in the task space and does not take into account the behaviour of the soft-robotic arm.

Recent model-based approaches [9] have been specifically developed to perform dynamic motion control with continuum robots. Similarly, [10] presents suitable models that combine feed-forward control and decoupled PD-controllers, applied to a pneumatically actuated manipulator. A different approach based on open-loop predictive controllers is proposed in [11], using machine learning derived dynamic models directly from the actuation to the task space. The work presented in [12] is based on a different set of techniques, in which novel spatial dynamics are applied to variable length multi-section continuum arms under the assumption of circular arc deformation of continuum sections without torsion. A relevant approach is presented in [13] where the authors use a feed-forward neural network component to compensate for dynamic uncertainties.

[1] All authors are with the School of Electrical and Computer Engineering, National Technical University of Athens, Greece.

[2] Mehdi Khamassi is also with Sorbonne Université, CNRS, Institute of Intelligent Systems and Robotics, Paris, France.

Email: oikonpar@mail.ntua.gr

However it is evident in the literature that the control schemes proposed for soft robots are highly dependent on the hardware set-up and actuation. Thus, attempting a fair performance comparison, we focus on dynamic control strategies applied onto the same soft robot. One of our previous works [14] presents a model-free neurodynamic controller based on Central Pattern Generators (CPGs) for the generation and tracking of periodic motions by the end-effector (EE) of a single module version of the soft-robotic arm. In this work, we focus on a more robust implementation that extends the capabilities of the soft arm by activating two modules and thus allowing for tracking of more generic and complex motions demonstrated by humans. In [15] a closed-loop predictive controller was implemented using trajectory optimization for training of a model-based policy learning algorithm. The focus of this approach was to achieve trajectory tracking accuracy at each time step, a requirement rarely set for soft robots. Such control schemes require large amount of data and many iterations for a single trajectory reproduction, making training convergence time-consuming. In our latest work [16] the proposed controller used ProMPs to create a mapping at the primitive level between task and actuation-space, whose proper combination aims for the reproduction of a trajectory defined by sparse way-points with time-constraints. However, the requested trajectories are limited to be similar with the derived demonstrations, e.g. in terms of trajectory direction/flow.

In this paper, we present an architecture that aims to control a two-module bio-inspired soft-robotic arm. The designed controller exploits the enhanced parameterizability of ProMPs, focusing on the qualitative reproduction of human demonstrations, assuming that any trajectory might be defined as the proper composition of individual primitives obtained by a learned MP library. Apart from the common ProMP framework, this work includes several novelties: a path segmentation process is used to properly divide the human demonstration into small linear segments; a learning-based module with the ability to incrementally update its model approximates the robot's inverse kinematics; and an auxiliary process handles the primitive's asynchronous activation through replanning. The efficiency of the proposed methodology is evaluated experimentally on the same two-module soft manipulator used in [16], and described in [17], [18]. Each module is actuated by three radially symmetric tendons driven by three servomotors (Hitec HS-422 Super Sport - Supermodified Servo by 01^{TM} Mechatronics) which change the configuration of the module after modifying their cable's length. The experimental results highlight its capability to perform trajectory control using a complex robotic system with unknown dynamics, in applications where high-precision is not required.

II. Problem Statement

A soft robot like the one examined in this work is not often assigned with the task of path following, especially when high-precision is required; the mechanical properties of its design are mostly exploited in tasks where safety must be ensured, such as those involving human-robot interaction or manipulation of fragile materials. Hence, our focus lies on the qualitative reproduction of human motions defined in a subspace of the robot's workspace. In some approaches like in [19], the desired motion's mapping from the task to the actuation space is accomplished through kinaesthetic teaching, however this is not applicable here due to the mechanical structure of the soft robot.

The static mapping between the task and joint space is usually provided by a mathematical model based on the known geometry of a rigid manipulator, like in [20]. However, in cases where the complexity of the robot's dynamics prevents the use of such approaches, alternative solutions are recommended, like the ones reviewed in [21], [22] focusing on model learning. On the other hand, most of these methods lack the ability to adjust their behavior on-the-fly, providing only offline training; this is a serious flaw since changes in robot's dynamics constitute a usual phenomenon in bio-inspired systems. The learning-based approximation of inverse kinematics designed in [16] does not sufficiently exploit the available information, and rather focuses only on the points of high interest - the so-called conditioning points - which are sparse, hence it is rarely updated.

III. Control Architecture

The proposed methodology assumes that a requested trajectory could be described as the composition of primitives obtained by a learned MP library, and formed properly in the task space by exploiting the ProMP's properties, while subsequently the composition's parameters are transferred unchanged to the actuation space, and applied to the corresponding primitives provided by a learned mapping. A block diagram briefly describing the control flow is illustrated in Fig. 2.

A. Demonstration generation and MP training

In the scenario of washing the human back and its involved sub-processes such as the water pouring task (Fig. 1), the motion of the robot is limited to the quasi-plane defined by the human back's surface (assuming xy-plane, without loss of generality), while the movement on the perpendicular direction is considered to be negligible. Accordingly, a grid of primitives built across the xy-plane is required, providing the capability to plan and reproduce trajectories as a result of primitives' composition.

The process through which the demonstrations are generated is quite similar to the one described in [16]. Concretely, a subspace W_{Sub} within the robot's workspace is initially defined as a region of interest for the requested trajectories, where the demonstrations should lie in. Subsequently, all motors are fed with actuation that result in the EE's movement from a starting point lying on the border of W_{Sub} towards another one, defining a direction of motion. In contrast to [16] where the MP library is formed by demonstrations of a single direction, here the aforementioned process is executed four times so that the resulting MPs cover all directions of xy-plane - $S_D = \{x+, x-, y+, y-\}$. These trajectories

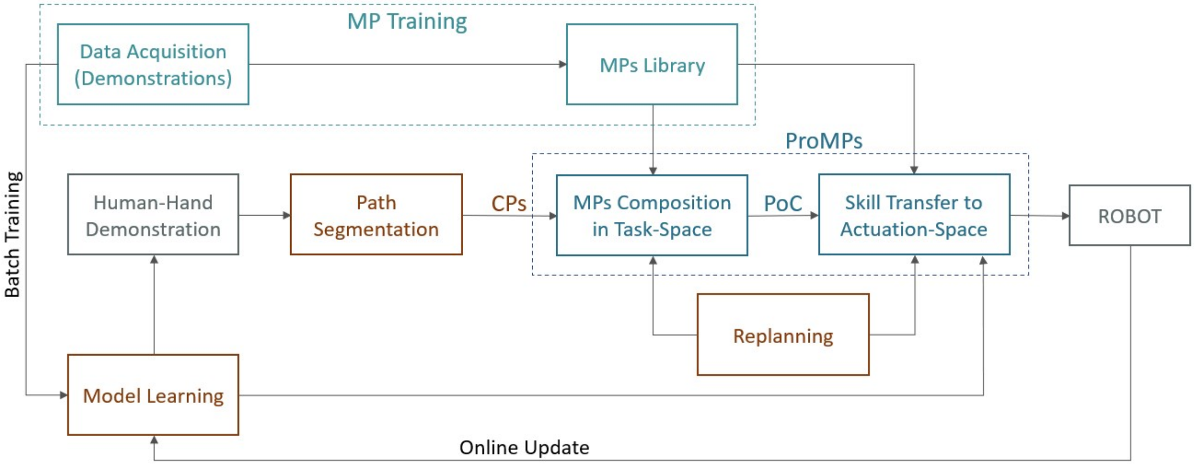

Fig. 2: The overall architecture of the proposed controller [CPs : Conditioning Points, PoC : Parameters of Composition]

are grouped into classes under a similarity criterion, before proceeding to MP training. Eventually, a grid of primitives is formed as shown in Figs. 1 and 3.

Even though each formed ProMP constitutes a trajectory distribution, which implies that it is defined by a mean vector and a covariance matrix, from now on it is supposed that the second is the identity matrix. Hence only the mean vector affects the resulting trajectory. In that sense, the stochasticity introduced by the demonstrations' variation is eliminated, and the resulting trajectory extracted by each primitive does not differ from one execution to the other.

B. Human-Hand Demonstration

During the human motion's recording, the hand is mainly moved on the xy-plane - defined by the human's back surface - while the movement on the z-axis is negligible. Before computing the controller's parameters, a set of pre-processing steps takes place in order to ensure that the trajectory's execution is feasible by the robot. The first one is to scale it on the xy-plane so that it lies within the limits of the subspace W_{Sub}. At the same time, the robot's motion must start and finish at points lying on the border of subspace W_{Sub} where the primitive blocks that will compose the requested trajectory have zero velocity. As a consequence, two additional points p_{CS} and p_{CE} should be determined on the limit of W_{Sub} where the robot's motion will start from and stop at, respectively, as shown in Fig. 3. Subsequently, for each (x,y) pair, the corresponding z coordinate that sets a feasible target in 3D-space for the robot should be estimated, using the method described in Section III-D.

C. Path Segmentation

In [23], the authors present a variety of algorithms for segmentation of paths lying on a plane with application in animal movement patterns' change detection, ranging from time to topology-based methods. Focusing on the second category, the Change Point Test (CPT) method [24] optimally divides the path into linear segments by detecting significant changes in the movement direction. As a result, a set S_{CP} of conditioning points is derived after neglecting any changes on z-axis. Subsequently, our implementation inserts intermediate points into S_{CP} where the distance h between two consecutive ones exceeds a predefined threshold h_{max}. Eventually, the two corner points p_{CS} and p_{CE} defined in Section III-B are also added to S_{CP}. It should be noted that each point in S_{CP} is accompanied by its velocity, as this is captured motion during the hand's motion. An illustrative example of path segmentation is depicted in Fig. 3.

D. Model learning for inverse kinematics

Focusing on the approximation of the robot's inverse kinematics, the implemented algorithm should be capable of exploiting the total available information. In [25] a novel data-driven method is presented, called Incremental Sparse Spectrum Gaussian Process Regression (I-SSGPR). The authors capitalised on the exhaustively studied Gaussian Process Regression aiming at designing a method that cope with unstructured and non-stationary environments where adaptability to changing conditions is required - as in our application. At the same time, low computational complexity is achieved, while automated hyperparameter optimization is provided. Another interesting feature is the capability to perform both offline training using an existing dataset, as well as online updates as soon as new data is available.

Apart from the I-SSGPR-based module that approximates the inverse kinematics, the operation of an additional one is required to provide the robot with a feasible target by computing the z coordinate when an (x,y) pair is received, as explained in Section III-B. Both modules are initially trained offline using the dataset derived during the demonstration generation described in Section III-A, while online updates are performed during trajectory execution by the robot.

E. ProMP composition in task space and skill transfer to actuation space

As already stated, the qualitative reproduction of a complex trajectory could be accomplished with the asynchronous activation and combination of MPs drawn from an existing library. Passing through the sparse way-points should be done with the appropriate velocity at a specific time-instance, as

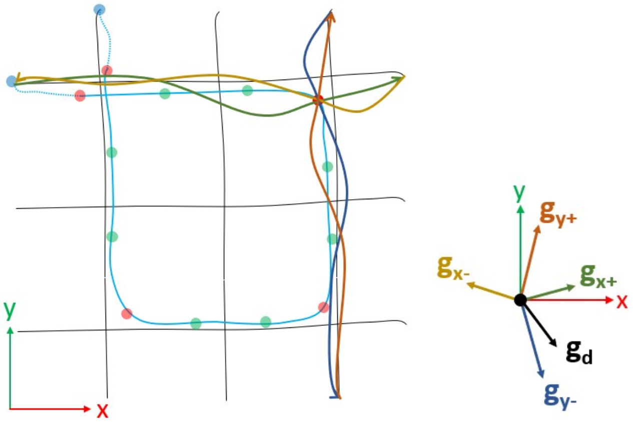

Fig. 3: **Left:** An illustrative example showing how some algorithmic components (ProMP training, Path Segmentation and ProMP composition in task-space) operate. The black curves forming the grid in xy-plane indicate the primitives of MP library obtained after training. The cyan solid curve is the human demonstration. The cyan dashed curves connect the human demonstration with the artificial conditioning points. The circles (o) denote the conditioning points [Red: CPT, Green: where distance between consecutive red exceeds a threshold, Blue: artificial conditioning points lying on the W_{Sub}]. The remaining colored curves are 4 primitives of different directions passing through the same conditioning point. **Right:** The desired velocity $\vec{g}_d$ is computed as the linear combination of the velocities $\vec{g}_{x+}$, $\vec{g}_{x-}$, $\vec{g}_{y+}$ and $\vec{g}_{y-}$ of the corresponding primitives at a specific conditioning point.

these are determined by the human hand's motion. To cope with such a challenge, each conditioning point should activate independently each selected primitive with the required features (conditioning and duration). The transition between consecutive points is realized with the replanning property, initially introduced in [16].

The following process applies to each conditioning point p_i in S_{CP}, except for the two corners p_{CS} and p_{CE}. At the first step, p_i is classified to one primitive for each direction $\{x+, x-, y+, y-\}$ - eventually classified to 4 primitives according to its distance from their closest point. As depicted in Fig. 3, all primitives, where p_i is classified to, are executed in the task space passing through p_i and the corresponding corner points of the primitive for which the conditioning property is also applied. After each selected primitive is (virtually) executed with a reference duration d_{ref}, a resulting velocity at p_i is derived.

The purpose here is to compute how slower/faster with respect to d_{ref} a primitive should be executed, so that the combination of all velocities at point p_i results in the desired one - noting that the duration is inversely proportional to the velocity. The velocities g_m with $m = \{x+, x-, y+, y-, d\}$ depicted in Fig. 3 could be written as follows:

$$\begin{aligned} \vec{g}_{x+} &= a_{x+}\hat{x} + b_{x+}\hat{y} \\ \vec{g}_{x-} &= a_{x-}\hat{x} + b_{x-}\hat{y} \\ \vec{g}_{y+} &= a_{y+}\hat{x} + b_{y+}\hat{y} \\ \vec{g}_{y-} &= a_{y-}\hat{x} + b_{y-}\hat{y} \\ \vec{g}_{d} &= a_{d}\hat{x} + b_{d}\hat{y} \end{aligned} \tag{1}$$

where a_m and b_m are the projections' coefficients of g_m onto axes x and y respectively. The desired velocity g_d is defined as the linear combination of velocities g_n with $n = \{x+, x-, y+, y-\}$ as follows:

$$\vec{g}_d = l_{x+}\vec{g}_{x+} + l_{x-}\vec{g}_{x-} + l_{y+}\vec{g}_{y+} + l_{y-}\vec{g}_{y-} \tag{2}$$

The goal then is to find the coefficients l_n for which Equation 2 holds. It should be noted that, since each velocity is derived from a primitive with a specific direction, negative coefficients l_n are not allowed. In this way, a system of linear equations is formulated that requires non-negative solutions, constituting a linear programming (LP) problem.

A common technique that treats such constrained systems is the simplex method described in [26]. Initially, two new artificial variables are introduced, as the number of equations derived by Equation 2. Proceeding to the solution, $L = [l_{x+}, l_{x-}, l_{y+}, l_{y-}, l_1, l_2]^T$ is requested that minimizes the linear objective function $c^T L$ with respect to L, where $c = [0, 0, 0, 0, 1, 1]^T$, subject to $A_{eq}L = b_{eq}$ and $L \geq 0$, where $b_{eq} = [a_d, b_d]^T$ and

$$A_{eq} = \begin{bmatrix} a_{x+} & a_{x-} & a_{y+} & a_{y-} & 1 & 0 \\ b_{x+} & b_{x-} & b_{y+} & b_{y-} & 0 & 1 \end{bmatrix} \tag{3}$$

As a result, coefficients l_{x+}, l_{x-}, l_{y+} and l_{y-} are derived, implying how slower/faster the corresponding primitive should be executed with respect to its reference velocity g_n at point p_i so that the desired velocity is accomplished as a linear combination of all primitives' velocities. Then each primitive's duration is computed by $d_n^{(i)} = d_{ref}/l_n$. The last parameter that is deduced is the starting time-instance $t_n^{(i)}$ of each primitive in the global time-frame.

Subsequently, the skill transfer process takes place where all derived parameters are transferred unchanged to the motor space, the MP library provides the corresponding primitives in the actuation-space (Section III-A), and the I-SSGPR module interprets all conditioning points into motor commands (Section III-D). Eventually, given that all primitives are formed, the execution takes place, where they are activated independently and asynchronously as determined by their starting time-instance $t_n^{(i)}$, as long as the replanning property handles the transition between primitives of consecutive conditioning points, as explained in the next section.

F. Replanning at the ProMP-level

In this work, the replanning at the primitive-level property (introduced in [16]) is assigned with the task of handling the transition between primitives of consecutive conditioning points by gradually decreasing the power of the last conditioning point's primitives, while increasing the power of the next one. Here, the replanning differs from blending in the sense that the primitives are not necessarily executed in parallel under synchronous activation, as it is assumed in [27]. Additionally, in this application the intuition behind replanning is that it ensures smooth transition between sequential primitives formed by consecutive transition points, rather than blending them.

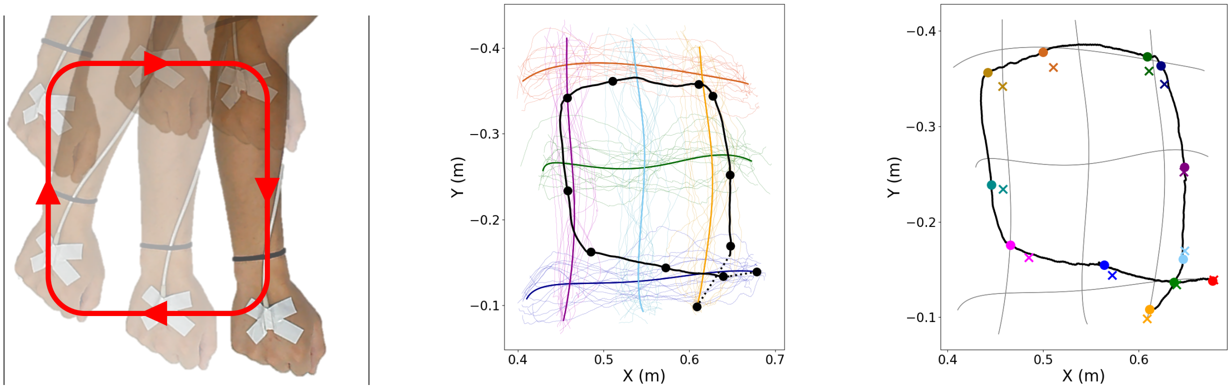

Fig. 4: Experimental results presenting the whole reproduction sequence of human hand motion demonstrations. **Left:** Human Hand Demonstration - A rectangular motion demonstration is captured using a 3D magnetic tracker attached to the user's hand. **Center:** Path Segmentation in Subspace - Adaptation of the demonstrated motion to the desired subspace of the robot's workspace visualized in 2D. The robot demonstrations along with the learned ProMP's mean trajectories are depicted with the respective colors. The result of the trajectory segmentation algorithm along with the extra conditioning points are shown with black dots. **Right:** Execution of trajectory #1 by the soft-robotic arm - The desired conditioning points are depicted with colored x symbol, while the executed points at the same timestep are depicted with the respective colored dots.

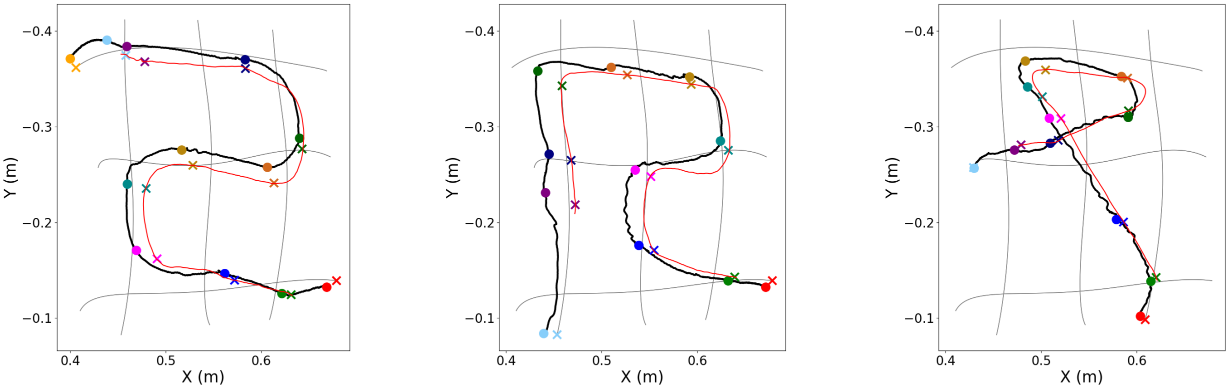

Fig. 5: Experimental results presenting the execution (solid black line) of the demonstrated trajectories (pale red line) by the soft-robotic arm. **Left:** Execution of trajectory #2 - The desired conditioning points are depicted with colored x symbols, while the executed points at the same timestep are depicted with the respective colored dots. The red symbols (x and dot) indicate the first conditioning point of the trajectory. **Center:** Execution of trajectory #3. **Right:** Execution of trajectory #4.

TABLE I: Mean error measurements for ten (10) consecutive executions of the human demonstrated trajectories by the soft-robotic arm, depicted in (Figs. 4-5). The position error e_p (cm), the velocity direction error $e_{\hat{v}_{xy}}$ (deg) and the ratio of the actual velocity to the desired one $e_{|\vec{v}|_{xy}}$ (dimensionless) are all measured in each conditioning point **CP##** at the corresponding time-step.

Traj. ID	**error type**	**CP01**	**CP02**	**CP03**	**CP04**	**CP05**	**CP06**	**CP07**	**CP08**	**CP09**	**CP10**	**CP11**	**CP12**	**average**
1	$e_p(cm)$	0.23	0.38	1.41	2.39	1.34	2.16	2.02	1.54	1.98	0.54	1.01	1.23	1.35
	$e_{\hat{v}_{xy}}(deg)$	-	7.66	4.91	2.67	0.63	23.34	18.11	0.47	2.25	0.32	2.55	-	6.29
	$e_{\lvert\vec{v}\rvert_{xy}}$	-	0.76	0.94	0.79	0.90	0.75	0.77	0.73	1.18	0.88	0.62	-	0.80
2	$e_p(cm)$	1.24	1.01	1.27	2.40	2.12	2.03	1.82	1.18	1.01	2.51	2.57	1.12	1.69
	$e_{\hat{v}_{xy}}(deg)$	-	5.44	13.14	2.21	14.87	5.73	5.32	0.13	15.04	17.25	17.18	-	9.63
	$e_{\lvert\vec{v}\rvert_{xy}}$	-	0.79	0.76	0.84	0.81	0.76	0.78	0.78	0.79	0.76	0.74	-	0.78
3	$e_p(cm)$	1.02	0.80	1.74	1.82	1.29	0.82	1.93	3.03	2.5	3.39	1.45	-	1.80
	$e_{\hat{v}_{xy}}(deg)$	-	8.22	22.70	1.43	3.67	0.48	18.88	17.68	2.41	12.98	-	-	9.83
	$e_{\lvert\vec{v}\rvert_{xy}}$	-	0.84	0.82	0.68	0.77	0.98	0.81	0.71	0.77	0.73	-	-	0.79
4	$e_p(cm)$	0.65	0.74	0.85	1.25	1.89	2.37	0.73	0.67	0.91	0.90	0.15	-	1.01
	$e_{\hat{v}_{xy}}(deg)$	-	3.05	26.95	10.76	12.27	13.37	0.47	16.97	23.77	9.73	-	-	13.04
	$e_{\lvert\vec{v}\rvert_{xy}}$	-	0.77	0.82	0.79	0.98	0.76	0.85	0.78	0.75	0.75	-	-	0.81

IV. Experimental Evaluation

In this work, the evaluation should focus on the various components' capability to perform as an entity in a collaborative way. Initially, directed demonstrations are generated by the robot, grouped into classes and trained to form the MP library (Section III-A). The resulting grid of primitives is illustrated in Fig. 4. Additionally, the collected data - the EE's positions along with the corresponding actuations - are used to offline train the two I-SSGPR modules (Section III-D). Note that, ten trigonometric basis functions are used to construct each I-SSGPR module. From now on, the learned models (ProMP library and I-SSGPR modules) are used as knowledge base by the planner in order to perform some desired tasks; the content of the ProMP library for both the actuation and the task-space is fixed, while both I-SSGPR modules are constantly updated as new data are obtained in the course of the robot's execution.

Four different captured hand's motions are chosen to be reproduced by the robot's EE, in such a way that each one introduces a variation in terms of complexity, while they are all indicative of the trajectories performed by humans during showering tasks. Concretely, trajectory #1 shown in Fig. 4 is the simplest one, consisting of linear segments whose corners are close to the corresponding primitives' edges. Hence they might be reproduced by the sequential activation of different primitives. On the other hand, trajectory #2 depicted in Fig. 5 uses just a small part of y-direction's primitives, while trajectory #3 constitutes a combination of the previous trajectories - it starts as the trajectory #2 and continues as the #1. Eventually, trajectory #4 is more complex, consisting of arbitrary movement blocks, including diagonal motion with respect to the primitives' direction forming the grid.

Figs. 4 and 5 depict the trajectories executed by the real robot projected onto the xy-plane, along with the corresponding desired and resulting conditioning points in the task space. The proximity level between the desired and the performed trajectories implies that the proposed methodology has the capability to successfully reproduce demonstrated hand's motions, since it manages to handle the movement not only close to the conditioning points but also in the intermediate space. Both the CPT algorithm (Section III-C) and the optimal composition of primitives for approximating the recorded velocity (Section III-E) contribute to the spatial similarity of the two motions.

On the other hand, it seems that the execution of trajectories #1, #2 and #3 is disturbed due to the following reason: from the moment the I-SSGPR modules are trained offline until the execution of each trajectory, many experiments took place resulting in loose cables (driven by the motors), and thus biased trajectory execution towards some directions. This phenomenon indicates the necessity for more active continuous update/tuning of the inverse kinematics model.

Despite the loose cables, from Table I it can be seen that the mean position error in each conditioning point after ten executions for all four trajectories is still relatively small, demonstrating the capability of the I-SSGPR model to approximate the complex kinematics of a soft-robotic arm, while ensuring qualitative reproduction of the hand's motion. At the same time, the measured velocity at each conditioning point is close to the targeted one - in terms of both magnitude and direction - even though it corresponds to different locations, indicating the robustness of our implementation.

The experimental evaluation shows that the proposed methodology is capable of reproducing complex movements out of simple demonstrations on a soft-robotic arm. A more illustrative presentation of the robot's performance is given in the video accompanying the manuscript (Suppl. Mat. 1).

V. Discussion and Conclusion

In the frames of this work, several methods operate in a collaborative way under the same control architecture whose performance is highly dependent on the efficiency of each component individually, as well as on their cooperation. The novelty of our work relies on exploiting the enhanced properties of ProMPs in order to control a soft-robotic arm, while avoiding the use of complex fixed models. The key principle here is to build a mapping at the primitive level between the task and the actuation space, enabling the capability of planning in the task and transferring the skill to the actuation space. At the same time, the auxiliary algorithms, namely CPT for path segmentation, I-SSGPR for model learning and the replanning at the ProMP-level, contribute towards this direction. The results show that the proposed architecture is able to qualitatively reproduce human demonstrations.

To the best of our knowledge, this is the first attempt that focuses on the composition of complex movements by asynchronously blending discrete building blocks such as the movement primitives in a parallel or a sequential way. The proposed architecture constitutes a one-shot approach since it manages to successfully execute the targeted trajectories after only one iteration without demanding big amount of data nor high computational effort.

In future work, we plan to implement a dynamic control scheme on top of the present methodology, assigned with the task to ensure active online correction to errors during execution. In addition, the capability to replan on-the-fly the trajectory with focus on coping with changes in the environment during execution (e.g. posture change of the human back) would provide added value. The action set could also be extended to include the pneumatic actuation, offering the ability to physically interact with the environment, handling external loads and applying forces. Eventually, we are planning to adapt this methodology in other soft robotic mechanisms, such as a soft robotic gripper developed in the frames of the EU-funded SoftGrip project.

Acknowledgments

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017054 (project: SoftGrip). This work has also been partially supported by the French Centre National de la Recherche Scientifique (CNRS), INS2I Appel Unique programme.

References

[1] W. H. Organization, *World report on ageing and health*. World Health Organization, 2015.

[2] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, 2016.

[3] F. Renda and C. Laschi, "A general mechanical model for tendon-driven continuum manipulators," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3813–3818.

[4] A. Grzesiak, R. Becker, and A. Verl, "The bionic handling assistant: a success story of additive manufacturing," *Assembly Automation*, 2011.

[5] Y. Ansari, M. Manti, E. Falotico, Y. Mollard, M. Cianchetti, and C. Laschi, "Towards the development of a soft manipulator as an assistive robot for personal care of elderly people," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416687132, 2017.

[6] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[7] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[8] A. C. Dometios, Y. Zhou, X. S. Papageorgiou, C. S. Tzafestas, and T. Asfour, "Vision-based online adaptation of motion primitives to dynamic surfaces: Application to an interactive robotic wiping task," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1410–1417, 2018.

[9] A. D. Kapadia, I. D. Walker, D. M. Dawson, and E. Tatlicioglu, "A model-based sliding mode controller for extensible continuum robots," in *Proceedings of the 9th WSEAS International Conference on Signal Processing, Robotics and Automation*, ser. ISPRA'10. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2010, p. 113–120.

[10] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic control of the bionic handling assistant," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 6–17, Feb 2017.

[11] T. George Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration and Biomimetics*, vol. 12, 08 2017.

[12] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamics for variable length multisection continuum arms," *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 695–722, 2016. [Online]. Available: https://doi.org/10.1177/0278364915596450

[13] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1270–1277, Dec 2007.

[14] P. Oikonomou, M. Khamassi, and C. S. Tzafestas, "Periodic movement learning in a soft-robotic arm," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4586–4592.

[15] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, Feb 2019.

[16] P. Oikonomou, A. Dometios, M. Khamassi, and C. S. Tzafestas, "Task driven skill learning in a soft-robotic arm," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1716–1723.

[17] Y. Ansari, M. Manti, E. Falotico, Y. Mollard, M. Cianchetti, and C. Laschi, "Towards the development of a soft manipulator as an assistive robot for personal care of elderly people," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416687132, 2017.

[18] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, June 2016, pp. 833–838.

[19] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, "Adaptation and robust learning of probabilistic movement primitives," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 366–379, 2020.

[20] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[21] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[22] O. Sigaud, C. Salaün, and V. Padois, "On-line regression algorithms for learning mechanical models of robots: a survey," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1115–1129, 2011.

[23] H. Edelhoff, J. Signer, and N. Balkenhol, "Path segmentation for beginners: an overview of current methods for detecting changes in animal movement patterns," *Movement ecology*, vol. 4, no. 1, pp. 1–21, 2016.

[24] R. Byrne, R. Noser, L. Bates, and P. Jupp, "How did they get here from there? detecting changes of direction in terrestrial ranging," *Animal Behaviour*, vol. 77, no. 3, pp. 619–631, 2009.

[25] A. Gijsberts and G. Metta, "Real-time model learning using incremental sparse spectrum gaussian process regression," *Neural networks*, vol. 41, pp. 59–69, 2013.

[26] G. Dantzig, *Linear programming and extensions*. Princeton university press, 2016.

[27] A. Paraschos, C. Daniel, J. Peters, G. Neumann *et al.*, "Probabilistic movement primitives," *Advances in neural information processing systems*, 2013.