Probabilistic learning and adaptation of shared control skills for assistive robots

Gabriel Quere[1,2], Freek Stulp[1], David Filliat[2], and João Silvério[1]

[1] RMC - German Aerospace Center (DLR)
[2] U2IS - ENSTA Paris, Institut Polytechnique de Paris

Keywords: Shared control, Learning from demonstrations, Probabilistic modeling

1 Introduction

Assistive robots promise to re-enable users with motor impairments to perform activities of daily living. Among different methods of assistance, with various degrees of autonomy, shared control empowers users with the ability to interact with their environment in a convenient manner. In previous work we introduced Shared Control Templates (SCT) [4] with the aim to reliably assist users in successful task execution, primarily demonstrated on the EDAN robot, see Fig. 1. One of the core SCT components are active constraints, restricting the robot task space to safely guide the user.

Currently, designing SCTs requires robotic expertise; to fully exploit their potential, SCTs should be easy to design and modify. Given their ability to approximate continuous functions, together with measures of uncertainty and strong adaptation properties, probabilistic learning from demonstration approaches are promising candidates to fulfill such requirements. In this work, we introduce an approach that combines Kernelized Movement Primitives (KMP) [2] and Generalized Cylinders [1] to derive active constraints from end-effector trajectories, for example demonstrated by kinesthetic teaching. Additionally, making use of KMP adaptation capabilities [5], we present a *correction mode* allowing users to modify SCTs at runtime, adapting the assistance provided by the framework to new environmental constraints and requirements, such as following a different path to complete a task.

2 Background

2.1 Shared Control Templates

An SCT consists of a set of states which, when executed sequentially, assist a user in the completion of manipulation tasks. Each state contains an input mapping – a mapping of user commands to manipulator end-effector velocities – as well as active constraints – restrictions of the available task-space, guiding the user to stay within regions that facilitate task completion and prevent undesired interactions with the environment. Within an active state, at every time step t an input mapping computes a displacement of the robot 6D end-effector pose from a 3D user command $\boldsymbol{u}(t) \in [-1,1]^3$. Standards mappings include 1-to-1 mappings from $\boldsymbol{u}$ to end-effector linear velocities or angular velocities. An active constraint then projects the desired end-effector pose to a subset of the task space, providing guidance. Previous approaches to learning active constraints have been proposed [3] albeit not easily adaptable and restricted to a pre-defined set of geometric constraints.

2.2 Kernelized Movement Primitives

KMP estimate a model that predicts the value of an output variable $\boldsymbol{\xi} \in \mathbb{R}^D$ given observations of an input $\boldsymbol{s} \in \mathbb{R}^I$ from a set of end-effector trajectories[3]. KMP assumes that a *reference trajectory distribution* $\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{n=1}^N$, encoding the means, variations and correlations of $\boldsymbol{\xi}$, is available to model $\mathcal{P}(\boldsymbol{\xi}|\boldsymbol{s}_n)$, where $\boldsymbol{s}_{n=1,\ldots,N}$ are N given inputs. The expectation of the output variable is computed, for a test input $\boldsymbol{s}^*$, using [2]:

$$\mathbb{E}[\boldsymbol{\xi}(\boldsymbol{s}^*)] = \boldsymbol{k}^* (\boldsymbol{K} + \lambda\boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}, \tag{1}$$

where $\boldsymbol{k}^*$, $\boldsymbol{K}$ are evaluations of $\boldsymbol{s}^*$ using the kernel function $k(\boldsymbol{s}_i, \boldsymbol{s}_j)$, $\lambda > 0$ is a normalization factor and $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_N^\top\right]^\top$, $\boldsymbol{\Sigma} = \mathrm{blockdiag}(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_N)$. The covariance of the output is also computed by KMP, see [2]. Querying a KMP model with equally spaced time inputs provides an expected trajectory of end-effector poses with associated covariance that can act as virtual guides, assisting users in shared control.

Model adaptation An interesting property of KMPs is their adaptation capabilities. For a certain $\boldsymbol{\mu}_n$, if the covariance $\boldsymbol{\Sigma}_n$ is small, the expectation at $\boldsymbol{s}_n$ will be close to $\boldsymbol{\mu}_n$. This provides a principled way for trajectory modulation: for a new input $\bar{\boldsymbol{s}}$, adding the pair $\{\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ to the reference distribution will ensure that the expected trajectory passes through a desired via-point $\bar{\boldsymbol{\mu}}$, provided that $\bar{\boldsymbol{\Sigma}}$ is small enough. Its downside is that it requires to invert the term $\boldsymbol{K} + \lambda\boldsymbol{\Sigma}$ every time a new point is added, which can be computationally costly ($O(n^3)$ complexity) depending on N. This can pose challenges if done during skill execution. In recent work [5], the original KMP formulation was extended with a term that locally modulates the trajectory distribution, enabling a more efficient adaptation with $O(n^2)$ complexity while preserving the covariance profile. In this work we introduce the online modulation of KMPs directly by user inputs leveraging the formulation in [5] (see Section 3.2).

2.3 Generalized Cylinder

Consider an ellipse ρ in a 2-dimensional plane, perpendicular to an arbitrary regular curve Γ, in $\mathbb{R}^3$. The 3D surface generated by translating the plane containing

[3] In this work we assume that $\boldsymbol{s}$ represents time and $\boldsymbol{\xi}$ the robot end-effector position.

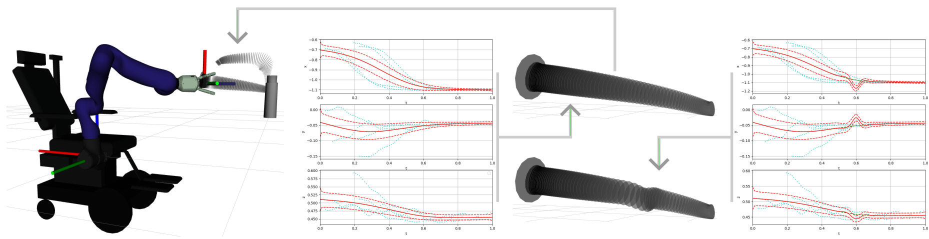

Fig. 1. Left: Simulation of the assistive robot EDAN starting a pick task, with an SCT skill composed of two states: *approach* and *lift*, with two Generalized Cylinders learned from demonstrations, one in each state, used as active constraints. **Center Left**: A KMP model fitted on trajectories for approaching a bottle. **Center right**: Top, a Generalized Cylinder derived from the fitted KMP; bottom, a Generalized Cylinder derived from a deformed KMP. **Right**: KMP model with deformation generated by the user with (2).

ρ along the arbitrary curve Γ (referred to as the directrix), while keeping the plane perpendicular to Γ, is a Generalized Cylinder [1]. A Generalized Cylinder can use any smooth simple closed-curve for ρ, but only ellipses are considered in this work. We here leverage the property that, while translating along the directrix, such ellipses vary smoothly to derive active constraints for SCTs.

3 Proposed approach

3.1 Deriving active constraints from a KMP

Let us assume a KMP with time as input and position as output, as described in Section 2.2. Provided with a dense linear sampling on the interval $[0, 1]$, this KMP outputs a trajectory distribution, containing a set of means and covariance matrices. We propose to use the reference trajectory obtained from the means as directrix Γ for a Generalized Cylinder. Additionally, for every point along Γ, the covariance matrix at an arbitrary variance threshold can be seen as an ellipsoid. The intersection of this ellipsoid with a plane slicing through its center, with the plane normal aligned with the trajectory, creates an ellipse ρ. The combination of those intersections for every point of Γ gives a set of ellipses representing the surface of a Generalized Cylinder.

A Generalized Cylinder can then be used as an active constraint acting in Cartesian space, by restricting the end-effector within its volume. A high variance at a specific section of the demonstrated trajectories provides motion freedom to the user, while a low variance constrains the user, e. g. at a specific grasp pose. Generalized Cylinders can also be directly derived from trajectory data, however adapting a Generalized Cylinder derived from data requires adaptation of individual trajectories, while when deriving from KMP, one can use the KMP adaptation capabilities, see below.

3.2 Skill adaptation

As described in 2.2, one can adapt a KMP with via-points, but the computational cost can be prohibitively high. Moreover, by altering the trajectory distribution $\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{n=1}^N$ through the inclusion of via-points $\{\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$, one modifies the covariance of the resulting KMP too, since $\bar{\boldsymbol{\Sigma}}$ needs to be small enough for the via-point to be fulfilled. This results in a very restrictive active constraint which may not always be desirable. Therefore we propose to act on the null space of the KMP (NS-KMP) [5] and adapt SCTs directly from user commands $\boldsymbol{u}$. In practice, the user could trigger at t_s a switch from the current skill execution to a correction mode, where applying commands would continually deform the expected KMP trajectory, visualized on a tablet. An new trigger would then allow the user to come back to the execution of the adapted skill. The deformation is computed as:

$$\mathbb{E}[\boldsymbol{\xi}(\boldsymbol{s}^*)] = \boldsymbol{k}^* \left(\boldsymbol{K} + \lambda\boldsymbol{\Sigma}\right)^{-1} \boldsymbol{\mu} + \left[\hat{\boldsymbol{k}}^* - \boldsymbol{k}^* \left(\boldsymbol{K} + \lambda\boldsymbol{\Sigma}\right)^{-1} \hat{\boldsymbol{K}}\right] \alpha \int_{t_s}^{t} \boldsymbol{u}(t)\, dt, \qquad (2)$$

where α is a scaling factor and $\hat{\boldsymbol{k}}^*$, $\hat{\boldsymbol{K}}$ are evaluations of the kernel function at the location where the null space action is applied. Note that the first term in (2) corresponds to (1), with the second term including a kernelized *soft* null space projector $\hat{\boldsymbol{k}}^* - \boldsymbol{k}^* \left(\boldsymbol{K} + \lambda\boldsymbol{\Sigma}\right)^{-1} \hat{\boldsymbol{K}}$ that deforms the original primitive only locally, see Fig. 1 for an example and [5] for details.

4 Conclusion

We propose an approach combining probabilistic skill representations and generalized cylinders to learn and adapt shared control skills. Preliminary results suggest that using NS-KMP as skill representation permits a computationally efficient modulation of SCTs without decreasing motion freedom for users.

References

1. Ahmadzadeh, S.R., Chernova, S.: Trajectory-based skill learning using generalized cylinders. Frontiers in Robotics and AI 5, 132 (2018)
2. Huang, Y., Rozo, L., Silvério, J., Caldwell, D.G.: Kernelized movement primitives. The International Journal of Robotics Research 38(7), 833–852 (2019)
3. Quere, G., Bustamante, S., Hagengruber, A., Vogel, J., Steinmetz, F., Stulp, F.: Learning and interactive design of shared control templates. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 1887–1894 (2021)
4. Quere, G., Hagengruber, A., Iskandar, M., Bustamante, S., Leidner, D., Stulp, F., Vogel, J.: Shared control templates for assistive robotics. 2020 IEEE International Conference on Robotics and Automation (ICRA) pp. 1956–1962 (2020)
5. Silvério, J., Huang, Y.: A non-parametric skill representation with soft null space projectors for fast generalization. arXiv preprint arXiv:2209.08522 (2022)