Online prediction of novel trajectories using a library of movement primitives

Paris Oikonomou, Athanasios Dometios, Mehdi Khamassi, Costas S Tzafestas

Online prediction of novel trajectories using a library of movement primitives

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

***Abstract*— Learning to anticipate other agents' future movements has gained increased interest in robotics, especially in situations requiring interaction. Such a prediction allows the robotic systems to plan their actions as a task evolves and before its completion. Specifically, in case of active robot collaboration, early decision making might ensure smoothness during motion and collision avoidance. While previous work has addressed prediction and recognition of primitive actions, to our knowledge little attention has been paid to the aspect of unseen ones. In this paper, we introduce a method for online continuous prediction of the evolution of previously unseen trajectories based on a recency weighting of past observations using a library of trained probabilistic movement primitives (ProMPs). The proposed method makes the assumption that parts of any novel trajectory could be considered as a combination of simpler ones. A probability distribution is derived across the evolution of the trajectory, and is updated in a recursive manner. We present a set of simulation experiments to showcase the new method, and compare it with a state-of-the-art method.**

I. Introduction

Equipping autonomous systems with the capability of predicting the future state of their surroundings based on current observations is a topic of high interest for the robotics community over the last years, due to the wide variety of applications (e.g., intention prediction in assistive robotics [1] and [2], and in industrial environments [3]). Such a knowledge is important for robots that operate in non-stationary environments whose continuous variability might be induced by other agents (e.g. humans, robots) that directly interact with and perform collaborative tasks, or operate in the same workspace. This dynamic nature constantly changes the perception of the robot which requires to replan its actions and demonstrate compliance. Predicting future states of the environment would allow robots to make fast decisions and adapt to rapidly evolving conditions.

A. Related Work

In the general context of multi-agent systems operating in a shared collaborative workspaces, a substantial amount of work have considered the prediction of intention topic. In particular, [4] develops a method for real-time intention inference using a model of the intention-driven dynamics. The method is evaluated in a table tennis task where an inference of the ball movement is made by analyzing the human player body movement before hitting it. The approach developed in [5] aims at online adapting the executing ProMP-based trajectory of the robot based on the prediction of the human motion, to ensure safety in tasks where the human and the robot share the same workspace. In this work, the model that provides the prediction consists of a goal tracker that learns the human motion goals, and a belief tracker that computes the transition probabilities between them based on past observations. Another work where a human and a robot operate in close proximity is presented in [6]. The developed method exploits patterns trained offline with Gaussian Mixture Models (GMMs) for predicting the human motion based on early observations, and perform a robotic action that minimizes the interference between them. Similarly, [7] attempts to predict the human motion using a cost function learned from a set of demonstrations representing the collaboration between two human agents, using inverse optimal control. In the context of shared workspaces, [8] manages to recognize a trajectory from early observations using a dynamic library of GMMs while predicting its remainder with Gaussian Mixture Regression. In addition, [9] proposes the prediction through a recurrent neural network after training a semantic graph modelling the interactions of all humans and objects in a scene. In [10], the authors introduce a multiple-predictor system that is a data-driven approach for predicting human motion by properly combining individual predictors drawn from a library.

Another class of related applications perform predictions based on the ProMP framework [11]. Such an approach is studied in [12] in the context of physical human-robot interaction. There, the authors attempt to predict the evolution of a movement whose early part is kinaesthetically demonstrated to the robot by a human, and complete the trajectory with the robot. The developed method initially classifies the movement to one of the learned primitives, and subsequently refines it based on the observed part. The outcomes of this method are exploited by [13] in an application which aims at eliminating the effects of communication delays in a master-slave teleoperation scheme where a human operator commands a humanoid robot in a remote environment, while receiving visual feedback. The properties of the ProMP framework are further enriched in [14] to allow online replanning and prediction of trajectory distribution.

The method introduced in [12] constitutes the state-of-the-art for prediction using ProMPs. Hence it is used as benchmark for evaluating the methodology proposed in this

[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.

‡Equally contributing senior authors

Email: oikonpar@mail.ntua.gr

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017054 (project: SoftGrip) and from CNRS INS2I IRP (project: APIER).

paper. The drawback of this approach is the limitation to only cope with trajectories that are close variations of the trained primitives. Therefore, the prediction of novel trajectories, i.e. those that largely differ from the ProMPs in the stationary library, is not recommended. Such a limitation is justified by the consideration that all observations contribute equally to the prediction, while normally the most recent ones provide a better notion of the future. Even trajectories that differ only by a scaling or a shifting factor from a primitive, are not taken into account. Moreover, the strict classification into a single primitive excludes the flexibility of exploiting more than one ProMPs in the prediction process.

B. Contribution

This paper proposes a method for predicting the evolution of novel trajectories using a library of trained ProMPs. Particularly, it attempts to compute a trajectory distribution implying the probability of the evolution of the observed trajectory, as the weighted combination of the known primitive distributions. To accomplish this, it is assumed that (a) the most recent observations play major role in the prediction compared to the older ones, and (b) parts of any trajectory could be approached as the linear sum of primitive ones. Another key factor is that the trained ProMPs are considered along with their affine transformations. The performance of the methodology is evaluated in simulation, where the prediction of various trajectories is requested. The results highlight its effectiveness and its robustness, especially when it is compared to a state-of-the-art method.

The key novelties of our work are summarized below:

- the prediction is a distribution derived as the proper combination of all primitive distributions;
- the library of primitives is enriched with their affine transformations, thus allowing scaling and shifting;
- weighted contribution of observations based on recentness; the most recent ones affect more the prediction;
- all updates are performed recursively as new observations are obtained, to reduce computational complexity.

II. Preliminaries

The methodology consists of some individual algorithmic components whose knowledge is considered preliminary. For the reader's convenience, their mathematical background and formulation are presented in this section for easy reference.

A. ProMPs: formulation, training and prediction

The ProMPs [11] provide a framework for learning and representing distributions over similar trajectories. Without loss of generality, the following presentation concerns the case of one-dimensional trajectories.

1) Mathematical formulation: Each ProMP is a Bayesian parametric model defined as $X_t = \mathbf{\Psi}_t \boldsymbol{\omega} + \epsilon_X$, where X_t is the random variable representing the trajectory distribution at time t, and ϵ_X denotes the trajectory noise, while $\mathbf{\Psi}_t = [\psi_1(t), \psi_2(t), \ldots, \psi_M(t)]$ is a vector of M radial basis functions (RBFs) computed at t. For uniformly distributed RBFs, $\psi_m(t)$ is Gaussian function with center $c_m = m/M$ and variance $h = 1/M^2$, normalized over all $\psi_j(t)$'s. Eventually, $\boldsymbol{\omega}$ is a parameter vector that weights the RBFs, and its probability distribution is normal $p(\boldsymbol{\omega}) \sim \mathcal{N}(\boldsymbol{\mu}_\omega, \mathbf{\Sigma}_\omega)$.

2) Training: During the ProMP training process, the weight vector $\boldsymbol{\omega}^{(j)}$ of the j-th primitive demonstration is computed through ridge regression:

$$\boldsymbol{\omega}^{(j)} = \left(\mathbf{\Psi}^T \mathbf{\Psi} + \gamma \mathbb{1}\right)^{-1} \mathbf{\Psi}^T \mathbf{x}^{(j)} \tag{1}$$

where $\mathbf{x}^{(j)} = \left[x_{t_1}^{(j)}, x_{t_2}^{(j)}, \ldots\right]^T$ represents the entire primitive demonstration, $\mathbf{\Psi} = \left[\mathbf{\Psi}_{t_1}^T, \mathbf{\Psi}_{t_2}^T, \ldots\right]^T$ is an array derived as the vertical concatenation of $\mathbf{\Psi}_t$ at all time-instances, and $\gamma \geq 0$ is the ridge parameter. From the set of weight vectors $\{\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)}, \ldots\}$ of all demonstrations, the mean vector $\boldsymbol{\mu}_\omega$ and the covariance matrix $\mathbf{\Sigma}_\omega$ are computed.

3) Prediction: The modulation of via-points of a trajectory is one of the most important properties provided by the ProMP framework. Such an operation is performed through Gaussian conditioning, resulting in the following update of the mean vector $\hat{\boldsymbol{\mu}}_\omega$ and the covariance matrix $\hat{\mathbf{\Sigma}}_\omega$, respectively:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_\omega &= \boldsymbol{\mu}_\omega + \mathbf{K}\left(y(t) - \mathbf{\Psi}_t \boldsymbol{\omega}_t\right) \\ \hat{\mathbf{\Sigma}}_\omega &= \mathbf{\Sigma}_\omega - \mathbf{K}\left(\mathbf{\Psi}_t \mathbf{\Sigma}_\omega\right) \\ \mathbf{K} &= \mathbf{\Sigma}_\omega \mathbf{\Psi}_t \left(\mathbf{\Sigma}_y + \mathbf{\Psi}_t \mathbf{\Sigma}_\omega \mathbf{\Psi}_t^T\right) \end{aligned} \tag{2}$$

where $\mathbf{K}$ is the Kalman gain matrix, and $\mathbf{\Sigma}_y$ is the measurement noise.

The operation of conditioning using Eqs. 2 is also exploited in [12] for predicting the evolution of a trajectory from early observations. Particularly, all observations are conditioned on the selected primitive, and the resulted trajectory distribution constitutes the probability of the prediction.

B. Recursive least squares formulation

In the task considered in this paper where the observations are obtained sequentially, it is preferable to perform computations recursively as new data become available in order to reduce computational effort and time. The recursive least squares (RLS) filter constitute one of the core mathematical tools whose properties are exploited in this work. According to [15] and [16], this adaptive filter provides estimations of slowly time-varying coefficients in an efficient recursive manner by minimizing the exponentially weighted least squares cost function shown below:

$$C(t) = \sum_{t_i=1}^{t} \lambda^{t-t_i} \left|\mathbf{e}_{t_i}\right|^2 = \sum_{t_i=1}^{t} \lambda^{t-t_i} \left|\mathbf{z}_{t_i} - \boldsymbol{\phi}_{t_i}^T \mathbf{w}_t\right|^2 \tag{3}$$

where $\mathbf{e}_{t_i}$ indicates the error at time t_i $\boldsymbol{\phi}_{t_i}$ is the regression matrix, $\mathbf{z}_{t_i}$ denotes the observation vector, $\mathbf{w}_t$ is the requested coefficients' vector at current time t, and $\lambda \in [0, 1]$ is the forgetting factor indicating the level that the new estimation is affected by the old one. The updates are derived

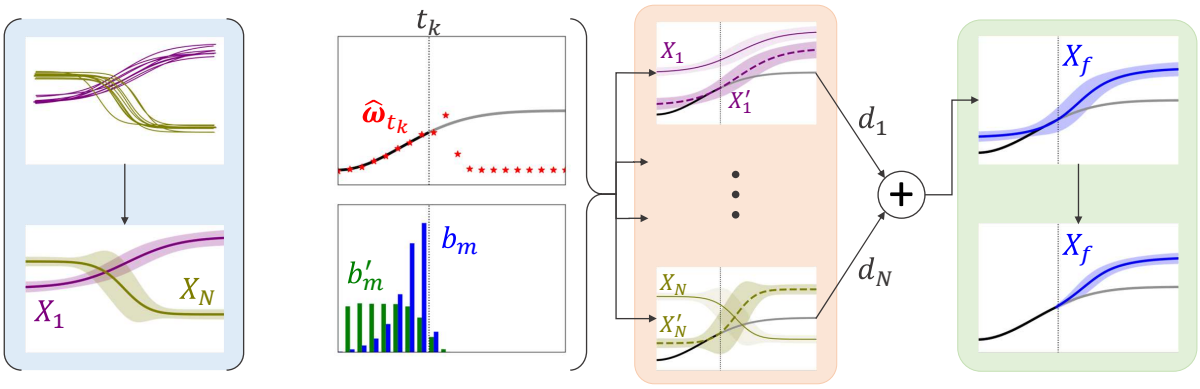

Fig. 1: Overview of the proposed pipeline using an illustrative example, depicting all processes taking place for the online prediction of a trajectory's (gray) evolution. The first column of figures (blue box) depicts the offline process of forming a library of ProMPs through demonstrations, which is executed only once. The remainder concerns the proposed methodology which runs after every new observation is obtained. In order of execution, the following processes are depicted: the online regression of target weight vector $\hat{\omega}_{t_k}$, the computation of confidence b'_m and approach b_m coefficients, the fitting of ProMPs (orange box), the computation of contribution coefficients d_n, and the formation of the predicted distribution before and after conditioning (green box) presented in Sections III-B-III-F, respectively.

by the following set of equations:

$$\begin{aligned}\mathbf{K}_t &= \mathbf{P}_{t-1}\boldsymbol{\phi}_t^T\left(\lambda\mathbb{1} + \boldsymbol{\phi}_t^T\mathbf{P}_{t-1}\boldsymbol{\phi}_t\right)^{-1}\\ \mathbf{P}_t &= \left(\mathbb{1} - \mathbf{K}_t\boldsymbol{\phi}_t^T\right)\mathbf{P}_{t-1}/\lambda\\ \mathbf{w}_t &= \mathbf{w}_{t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \boldsymbol{\phi}^T\mathbf{w}_{t-1}\right)\end{aligned} \tag{4}$$

where $\mathbf{w}_0 = \mathbb{0}_{L\times 1}$ for L coefficients to estimate, $\mathbf{P}_0 = \delta^{-1}\mathbb{1}_{L\times L}$, and δ is a small positive constant.

III. Prediction in the ProMP framework

The prediction of a trajectory's evolution under the ProMP framework is a task that has been considered in [12]. One question that arises concerns its capability to cope with novel trajectories, i.e. those not included in a library of learned primitives whose number is inevitably limited. Moreover, this method suggests the classification of the observed trajectory as one of the ProMPs, and the uniform consideration and exploitation of all observations for predicting the evolution. But what if the weighted contribution of observations or the weighted classification as primitives are considered?

A. Overview

To cope with the drawbacks of previous implementations, the proposed methodology has been built based on certain key factors. To start with, the main assumption behind this work is that any trajectory could be seen as the weighted combination and blending of simpler ones which are constantly being activated and de-activated. The simple trajectories are considered primitives and are stored in a (non-)stationary library along their features - mean and variance in case of trajectory distributions. In addition, it is expected that the most recent observations have greater effect on the evolution of a trajectory. Another key feature is that all computations are performed in a recursive or online manner, which principally results in low computational complexity. The recursive computation of the requested coefficients leads also to small updates based on previous estimations ensuring the smooth change of predictions.

Based on that, the goal of the proposed methodology is to compute a "living" - in the sense that it constantly changes as new data arrive - distribution as the weighted linear combination of more than one primitive trajectory distributions, based mainly on the most recent observations. To sum up, this strategy differs from the standard one [12] in three aspects: (a) the time-dependent influence of observations, (b) the enhancement of the ProMP library with the scaling and shifting transformations of the learned primitives and (c) the simultaneous activation of multiple ProMPs for the prediction. Fig. 1 provides an illustrative example of the step-by-step implementation and the information flow.

Before proceeding to the detailed presentation of the algorithmic procedures, keep note of the following: (a) The duration of the trajectory to predict is assumed to be a priori known which either might be the case in some applications, or it is computed with one of the methods described in [12] (e.g., maximum likelihood). (b) All primitives and trajectories are formed using the same kernels, in terms of type (RBFs) and distribution in time. (c) Primitive demonstrations have been generated, classified into groups, trained as ProMPs, and finally stored in a library - which here is considered to be stationary. In the remainder of this section, the algorithmic procedure taking place after every new observation is obtained, is analytically presented.

B. Online regression from observations

The data manipulation and the trajectory encoding are important factors in the prediction process. A common approach assumes that the observed trajectory might be represented as the weighted sum of RBFs uniformly distributed in time. Such a representation requires that $o_t = \boldsymbol{\Psi}_t\boldsymbol{\omega}_t$ holds, where o_t, $\boldsymbol{\Psi}_t$ and $\boldsymbol{\omega}_t$ denote the observation, the basis function, and the weight vector at time t, respectively.

As suggested in [17], the locally weighted regression with recursive least squares method allows the online learning of the weights of the forcing term in the discrete DMP framework [18]. A similar approach might also provide the

online estimation of $\hat{\boldsymbol{\omega}}_{t_k}$:

$$\begin{aligned}\mathbf{P}_{t_k} &= \frac{1}{\lambda}\left(\mathbf{P}_{t_{k-1}} - \frac{\mathbf{P}_{t_{k-1}}\boldsymbol{\Psi}_{t_k}^T\boldsymbol{\Psi}_{t_k}\mathbf{P}_{t_{k-1}}}{\lambda + \boldsymbol{\Psi}_{t_k}\mathbf{P}_{t_{k-1}}\boldsymbol{\Psi}_{t_k}^T}\right) \\ \hat{\boldsymbol{\omega}}_{t_k} &= \hat{\boldsymbol{\omega}}_{t_{k-1}} + \left(o_{t_k} - \boldsymbol{\Psi}_{t_k}\hat{\boldsymbol{\omega}}_{t_{k-1}}\right)\mathbf{P}_{t_k}\boldsymbol{\Psi}_{t_k}^T\end{aligned} \tag{5}$$

where λ serves like the forgetting factor in Eqs. 3 and 4, t_k denotes the current time-instance, and k is the k-th observation, while $\mathbf{P}_{t_0} = \mathbb{1}_{M\times M}$ and $\hat{\boldsymbol{\omega}}_{t_0} = \mathbb{0}_{M\times 1}$ for M RBFs. Section IV involves experiments that evaluate the performance of the online regression method in two indicative trajectories, with respect to the kernel estimation through ridge regression (Eq. 1). In the rest of the manuscript, the most recent estimation of the weight vector of the observed trajectory is denoted by $\hat{\boldsymbol{\omega}}$.

C. Confidence and approach coefficients

The weight vector $\hat{\boldsymbol{\omega}}$ of the observed trajectory computed in the previous step is used as target during the fitting process (Section III-D). Before that, it precedes the definition of a heuristic indicating the level of confidence that the estimation of the target weight $\hat{\boldsymbol{\omega}}_m$ is reliable. Intuitively, the more observations have been obtained around the m-th RBF, the higher its level of confidence should be. Considering that $\psi_m(t)$ implies the proximity of observation o_t to the m-th RBF, the corresponding confidence coefficient until current time t_k is computed as follows:

$$b'_m(t_k) = \frac{\sum_{t=t_1}^{t_k}\psi_m(t)}{\sum_{j=1}^{M} b'_j(t_k)} \tag{6}$$

where $t = \{t_1, \ldots, t_k\}$ denotes the total set of time-instances at which observations have been obtained. According to our strategy, the most recent observations - thus, the target weights whose RBFs are closer to t_k - should contribute more on the prediction. Hence, another heuristic is defined indicating the level that $\hat{\omega}_m$ affects the prediction based on its recentness. The approach coefficient of the m-th RBF whose center lie before the current time ($c_m \leq t_k$), is:

$$b_m(t_k) = b'_m(t_k)\cdot e^{-r(t_k - c_m)} \tag{7}$$

where r is a positive constant implying the decay rate of b_m towards past kernels. For future RBFs ($c_m > t_k$), Eq. 7 still holds for r being negative. Eventually, a vector of approach coefficients is derived: $\mathbf{b}(t_k) = [b_1(t_k), \ldots, b_M(t_k)]^T$. For simplicity, it is assumed that $\mathbf{b}$ always refers to the current time in the rest of the manuscript.

D. Recursive fitting of primitives to observations

Here, the first process that correlates the observed trajectory with the library of primitives takes place. Specifically, the goal is to fit each primitive distribution X_n to the observed trajectory using a scaling and a shifting factor. This approach provides the flexibility of predicting trajectories that constitute simple affine transformations of the primitive ones. The fitting process runs in a recursive manner (Eqs. 4) after every new observation o_{t_k} is obtained using the weight vector $\hat{\boldsymbol{\omega}}$ as target. The importance of each target weight is determined by the corresponding approach coefficient.

For each primitive X_n, the formulated problem attempts to: (a) reduce the error between the target weights and the primitive's, (b) ensure continuity between the observed and the predicted trajectory, and (c) restrict the scaling coefficient to be as low as possible. Therefore, the error $e_{t_k}^{(n)}$ for the n-th primitive is defined as the sum of three addends that cope with the aforementioned (a), (b) and (c) factors, respectively:

$$\begin{aligned}e_{t_k}^{(n)} = &\sum_{m=1}^{M} b_m \cdot \left(u_n\ \mu_{\omega_m}(n) + v_n - \hat{\omega}_m\right) \\ &+ h_1 \cdot \left(u_n\ E\left[X_n(t_k)\right] + v_n - o_{t_k}\right) + h_2 \cdot u_n\end{aligned} \tag{8}$$

where u_n and v_n are the scaling and shifting coefficients to estimate, $\mu_{\omega_m}(n)$ denotes the mean weight of the m-th RBF of X_n, $E\left[X_n(t_k)\right]$ indicates the expected value of X_n at current time t_k, and o_{t_k} is the last observation, while h_1 and h_2 are constants. Intuitively, b_m, h_1 and h_2 imply the priority to minimize the corresponding part of the cost function. In this application, h_1 should be considerably large to ensure continuity between the observed and the predicted part of the trajectory, while h_2 is preferred to be relatively small operating as a soft constraint to the scaling coefficient u_n.

Hence, N modified trajectory distributions are defined as:

$$X'_n = \sum_{n=1}^{N} u_n X_n + v_n \sim \mathcal{N}\left(u_n\boldsymbol{\mu}_{X_n} + v_n,\ u_n^2\boldsymbol{\Sigma}_{X_n}\right). \tag{9}$$

E. Contribution of primitives based on Bayes rule

The definition of N modified trajectory distributions X'_n as affine transformations of the primitive ones, is followed by the computation of the coefficients that properly combine them. A simple way to quantify the contribution of X'_n, is to compute the probability that the observed trajectory is derived by that. Aiming at reducing complexity, the target weight vector $\hat{\boldsymbol{\omega}}$ is used again for encoding the observed trajectory. The probability of X'_n given the target weight $\hat{\omega}_m$ is obtained using the Bayes rule as follows:

$$p\left(X'_n|\hat{\omega}_m\right) = \frac{p\left(\hat{\omega}_m|\theta^{(n)}\right)p(X'_n)}{\sum_{j=1}^{N} p\left(\hat{\omega}_m|\theta^{(j)}\right)p(X'_j)} \tag{10}$$

which could be simplified assuming uniform distribution for the initial probability of all primitives (i.e. $p(X'_j) = 1/N$). In Eq. 10, $\theta^{(j)}$ denotes the parameters of primitive X'_j, while $p(\hat{\omega}_m|\theta^{(n)})$ is the probability of $\hat{\omega}_m$ given $\theta^{(n)}$:

$$p(\hat{\omega}_m;\theta^{(n)}) = \mathcal{N}\left(\hat{\omega}_m|\boldsymbol{\Psi}_{c_m}\boldsymbol{\mu}'_\omega(n), \boldsymbol{\Psi}_{c_m}\boldsymbol{\Sigma}'_\omega(n)\boldsymbol{\Psi}_{c_m}^T + \boldsymbol{\Sigma}_{\hat{\omega}}\right) \tag{11}$$

where $\boldsymbol{\mu}'_\omega(n)$ and $\boldsymbol{\Sigma}'_\omega(n)$ are the mean vector and covariance matrix of X'_n respectively (Section III-D), $\boldsymbol{\Psi}_{c_m}$ is the radial basis vector computed at $t = c_m$, and $\boldsymbol{\Sigma}_{\hat{\omega}}$ is a diagonal matrix indicating the target's estimation noise.

The contribution of each transformed primitive is defined as the weighted combination of probabilities based on the approach factors. In particular, the contribution coefficient

that implies the proximity of X'_n to the observed trajectory is given by:

$$d_n = \frac{\sum_{m=1}^{M} b_m \cdot p\left(X'_n | \hat{\omega}_m\right)}{\sum_{j=1}^{N} d_j}. \tag{12}$$

F. Prediction as a new trajectory distribution

The previous sub-processes have provided all the necessary parameters to form a new trajectory distribution X_f that provides a prediction of the trajectory's evolution. Concretely, X_f is derived as the affine transformation of all ProMPs $X_n \sim \mathcal{N}(\boldsymbol{\mu}_{X_n}, \boldsymbol{\Sigma}_{X_n})$ as follows:

$$X_f = \sum_{n=1}^{N} d_n \cdot (u_n X_n + v_n) \sim \mathcal{N}\left(\boldsymbol{\mu}_{X_f}, \boldsymbol{\Sigma}_{X_f}\right) \tag{13}$$

where u_n and v_n are the scaling and shifting coefficients of the n-th primitive respectively, and d_n denotes the corresponding contribution coefficient, while $\boldsymbol{\mu}_{X_f}$ and $\boldsymbol{\Sigma}_{X_f}$ are the mean vector and covariance matrix of X_f, respectively:

$$\begin{aligned} \boldsymbol{\mu}_{X_f} &= \sum_{n=1}^{N} d_n \cdot \left(u_n \boldsymbol{\mu}_{X_n} + v_n\right) \\ \boldsymbol{\Sigma}_{X_f} &= \sum_{n=1}^{N} \left(d_n\, u_n\right)^2 \cdot \boldsymbol{\Sigma}_{X_n}. \end{aligned} \tag{14}$$

During the last step, the trajectory distribution X_f is properly modulated through conditioning so that it complies with the observed trajectory. In this case, the past target weights ($c_m \leq t_k$) are used as via-points using Eqs. 2 with time-dependent measurement noise $\boldsymbol{\Sigma}_y$. Preferably, the most recent target weights should affect the modulation more than the older ones. This outcome could be achieved through the proper definition of the measurement noise σ_m^2 ($= \boldsymbol{\Sigma}_y$) for weight $\hat{\omega}_m$ based on its recentness, i.e. low noise for recent ones, and high for older ones. Therefore, the noise σ_m^2 is mathematically expressed as follows:

$$\sigma_m^2 = r_1\ e^{r_2(t_k - c_m)} \tag{15}$$

where r_1 and r_2 are positive constants. The distribution X_f is also conditioned on the last observation o_{t_k} to ensure continuity between the observed and the predicted part of the trajectory. Eventually, X_f after conditioning implies the probability distribution of the trajectory's evolution, whose expected value indicates the most possible prediction.

IV. Experimental Evaluation

The advantage of the proposed methodology over a state-of-the-art method, to predict the evolution of novel trajectories (whose novelty metric is given in Table I), is showcased through a series of experiments.

A. Comparison with a state-of-the-art method

At first, the proposed method is evaluated based on its capability to predict the evolution of trajectories with varying deviation from the ProMPs constituting the library, in comparison with the method presented in [12]. The latter, as stated in Section I-A, initially classifies the observed trajectory as one of the trained primitives, and then modulates the selected ProMP through conditioning using the observations as via-points (Eqs. 2). The outcome is a trajectory distribution indicating the probability of the trajectory's evolution. For the comparison with our new method, the above process is repeated after every new observation is received.

Fig. 2 illustrates the results of the first set of experiments, where the two methods attempt to predict evolution of four different trajectories (one in each row), whose deviation from the trained ProMPs (green) varies. In the first case (Tr1), the deviation between the real trajectory to predict and the first ProMP is negligible. The second (Tr2) and third (Tr3) cases constitute shifting and scaling transformations of the first ProMP, respectively. In the last row (Tr4), there is an arbitrary trajectory formed as a Gaussian function with different left-hand and right-hand sides. The fifth column of the figure shows two different prediction error measurements: the mean distance between the prediction and the real trajectory (thick curve), and the same error computed using the Mahalanobis distance (thin curve) as following:

$$e_{Mah}(t_k) = \frac{1}{1 - t_k} \int_{t_k}^{1} \sqrt{\left(o_t - \mu_t\right)^T \left(\Sigma_t\right)^{-1} \left(o_t - \mu_t\right)}\, dt \tag{16}$$

where $\mu_t = \boldsymbol{\Psi}_t \boldsymbol{\mu}_{X_f}$ and $\Sigma_t = \boldsymbol{\Psi}_t \boldsymbol{\Sigma}_{X_f} \boldsymbol{\Psi}_t^T + \Sigma_o$.

In the first simple case (Tr1, first row), the state-of-the-art method (red) manages to converge faster than the proposed one (blue). This is expected since the trajectory to predict (gray) lies within the high-probability range of the first ProMP. However, in more difficult cases (second and third rows), our proposed method achieves a better performance due to its capability to implicitly enhance the library with the affine transformations of the ProMPs. In the last experiment (Tr4, fourth row), where the trajectory to predict is arbitrary compared to the trained primitives, the proposed method manages to adapt its estimation providing a more natural and reliable prediction while converging faster.

B. Evolution of internal coefficients over time and prediction

The second set of experiments focus on predicting the evolution of novel trajectories, while observing the evolution of the methodology's internal coefficients over time. The results are depicted in Fig. 3. The first three columns illustrate the prediction (blue) at three different time instances, while the next two (fourth and fifth columns) depict the evolution of the scaling and shifting coefficients, and the contribution of each ProMP with the corresponding colors, respectively. The last column shows the prediction errors. Here, four new arbitrary trajectories (one in each row) are presented. The first one (gray, Tr5) constitutes a shifted transformation of the last example (Tr4) of Fig. 2. It seems that our method handles the two trajectories similarly, computing a shifted prediction for the shifted trajectory (Tr5, first row in Fig. 3), which is also validated by their similar prediction error plots. In the next two cases (Tr6-Tr7), where the trajectories to predict represent a linear combination of the primitives

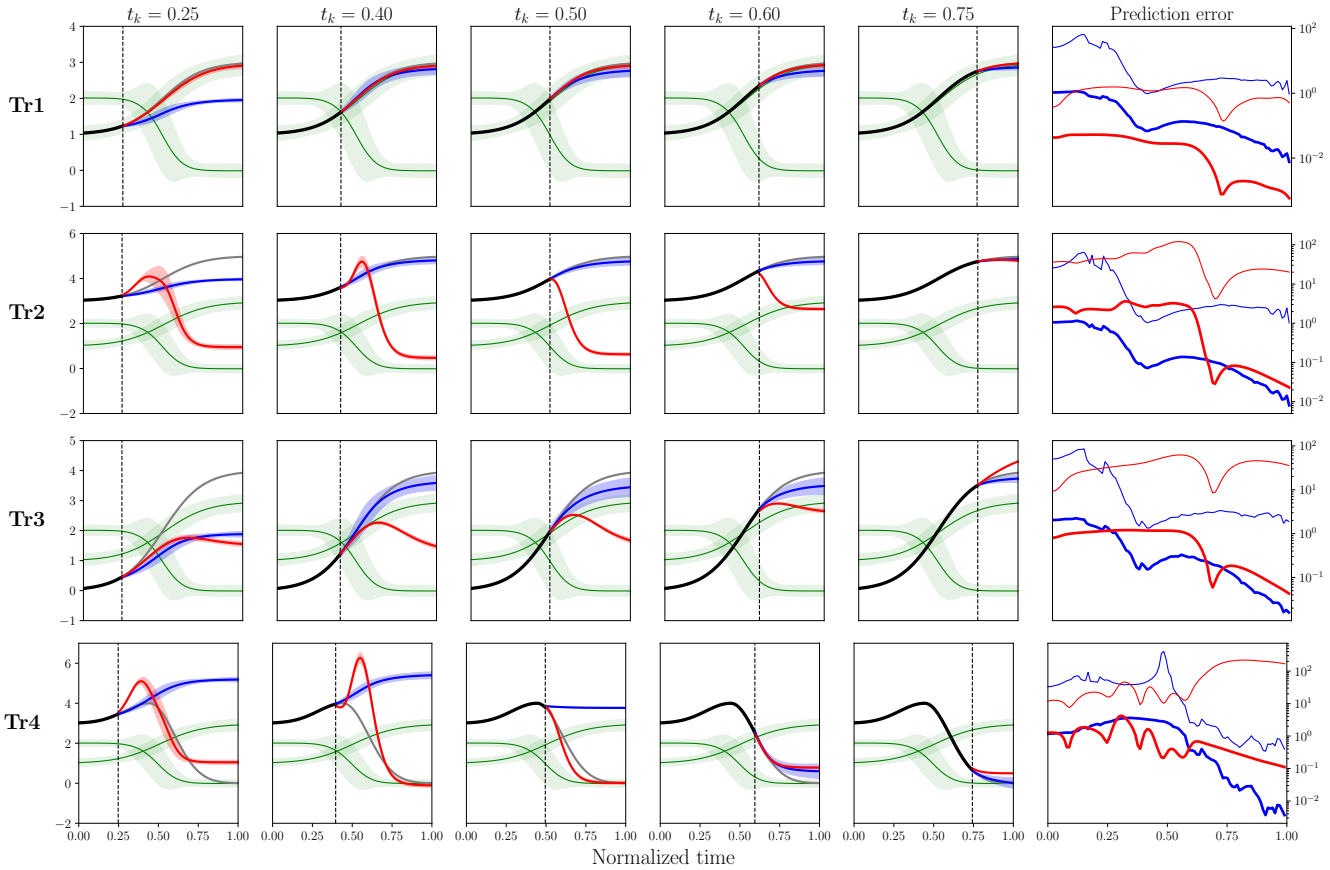

Fig. 2: Comparison of the proposed methodology (blue) with a state-of-the-art method using ProMPs [12] (red), for prediction of trajectories' evolution. Four examples are depicted (Tr1-Tr4), one in each row. The first five columns depict the outcome of prediction after 25, 40, 50, 60 and 75% of observed data points, where in blue and red are the predicted trajectory distributions. The green distributions indicate the two trained ProMPs forming the library. The gray curve is the trajectory to predict. The vertical black line indicates the current time t_k. The last column illustrates the prediction error; the thick curves are the mean distance between the prediction's expected value and the real trajectory to predict, while the thin curves denote the same term computed using Mahalanobis distance (Eq. 16).

TABLE I: Metric indicating the novelty of each trajectory to predict compared to the trained ProMPs. The first column implies the ID of the ProMP to compare with, the second one is the type of metric, and the rest columns contain the novelty metric - as a mean and a standard deviation values - for each trajectory examined in the experiments (Tr1-Tr8). Two metrics have considered: (a) the mean and standard deviation of the distance between the trajectory to predict and the expected value of the ProMP (n_d), and (b) the same terms using Mahalanobis distance (n_m) computed by Eq. 16 with $t_k = 0$. The higher the metrics, the more novel the trajectory to predict is.

ProMP	**dist.**	**Tr1**	**Tr2**	**Tr3**	**Tr4**	**Tr5**	**Tr6**	**Tr7**	**Tr8**
	n_d	0.04 ± 0.03	2.05 ± 0.03	0.7 ± 0.3	2.0 ± 0.67	2.3 ± 1.0	0.46 ± 0.14	1.67 ± 0.23	3.0 ± 1.6
1	n_m	0.28 ± 0.21	15.7 ± 3.92	5.73 ± 3.0	16.06 ± 6.7	18.74 ± 10.9	3.68 ± 1.8	13.1 ± 4.6	24.4 ± 16.9
	n_d	1.6 ± 1.0	3.02 ± 1.6	2.24 ± 1.2	1.33 ± 0.88	2.33 ± 0.88	1.62 ± 0.7	1.45 ± 0.9	3.9 ± 1.7
2	n_m	15.32 ± 12.33	26.1 ± 20.55	22.04 ± 15.4	8.55 ± 5.28	17.15 ± 7.5	15.8 ± 9.84	14.48 ± 12.24	36.1 ± 25.4

and an inverted right-hand side Gaussian, respectively, the prediction produced by our method converges fast to the real trajectory. In contrast, the methodology fails to anticipate the periodicity of the last trajectory (Tr8, fourth row), resulting in large prediction error after some initial observations and slow convergence. However, the error is still lower compared to the state-of-the-art (red curve).

As for the coefficients, the evolution of all scaling and shifting factors (fourth column) remains continuous, which is algorithmically ensured by the recursive least-square method that updates the estimations based on past ones. In the first three cases (Tr5-Tr7), one can notice that the contribution to the prediction is most of the time non-negligible for both primitives, while its evolution over time is continuous. The corresponding outcome differs in the last row (Tr8) where the contributions constantly and abruptly alternate over time due to the periodicity of the trajectory, resulting in negligible contribution by one of the two primitives most of the time. The outcomes strengthen the claim for robustness of the proposed methodology. It is also worth mentioning that all predictions are continuous with respect to the last observation, which is ensured by the second addend of Eq. 3.

C. *Evaluation of online regression*

In this section, we evaluate the online regression method (Section III-B) for computing of the target weight vector

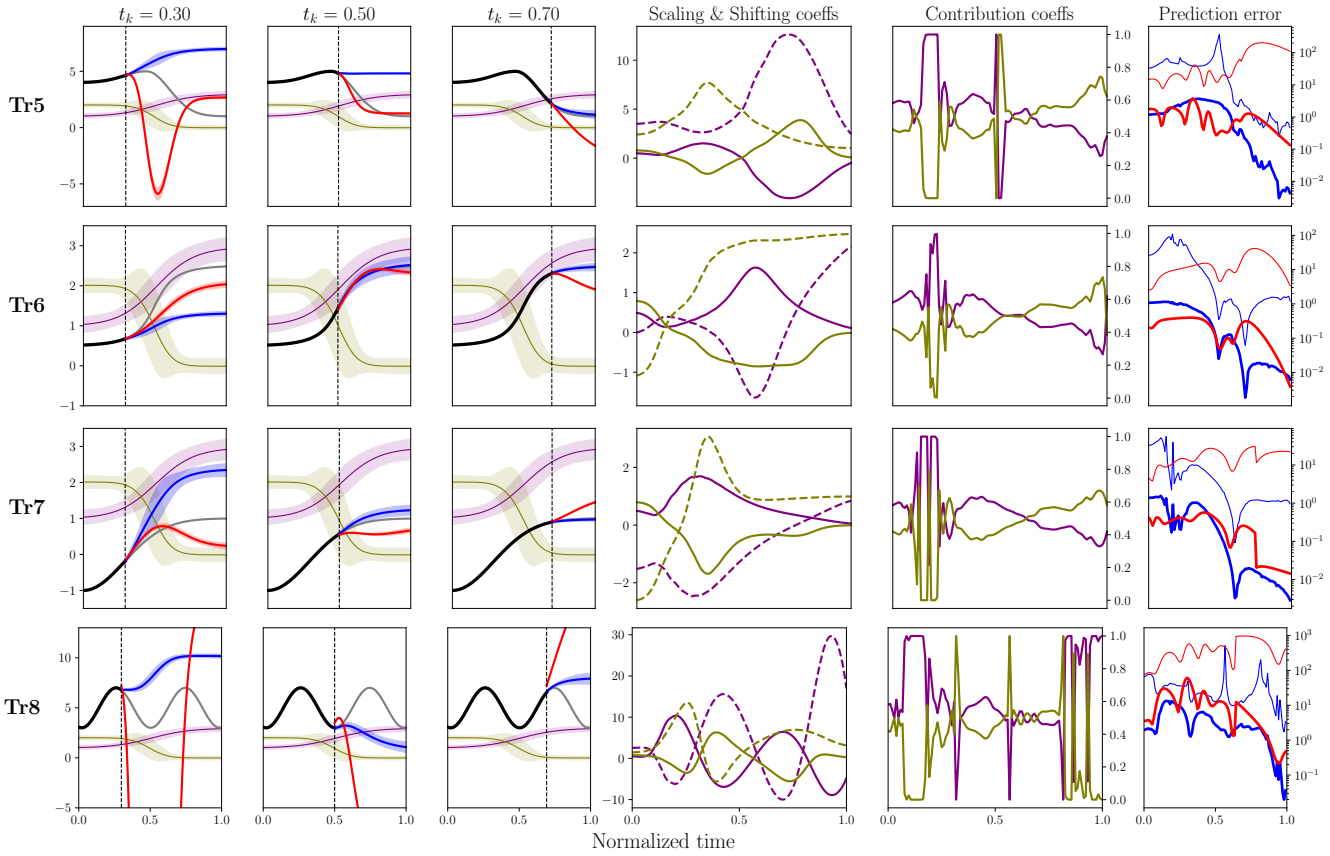

Fig. 3: Evaluation of the prediction provided by the proposed methodology in four new examples (Tr5-Tr8) and depiction of the evolution of the internal coefficients over time. For the whole figure, the elements colored in purple and yellow denote the two trained ProMPs and their corresponding coefficients, while the predicted trajectory distribution and its errors are illustrated in blue. Specifically, at each example (row), the first three columns depict the outcome of prediction in blue after 30, 50, and 70% of observed data points, the two trained ProMPs in purple and yellow, the trajectory to predict in gray, and the black vertical line indicating the current time t_k. The solid and dashed curves in the fourth column represent the scaling u_n and shifting v_n coefficients for each ProMP, respectively, while the fifth column illustrate the evolution of the contribution coefficients d_n over time. The last column depicts the prediction errors using the same format with Fig. 2. In all plots, the red elements correspond to the outcomes of the state-of-the-art method.

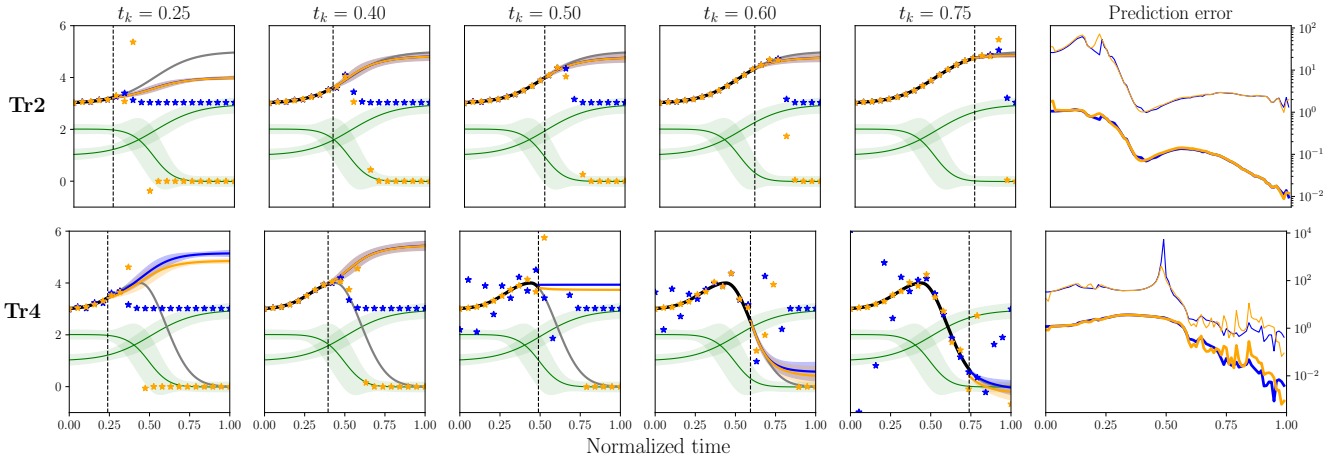

Fig. 4: Evaluation of the online regression method for estimation of the observed trajectory's target weights, in comparison with ridge regression. At each example (row) above, the proposed methodology was executed twice under the same parameters and initial conditions, with the only difference being the method that computes the target weights. In the figure, the blue and the orange asterisks (*), denote the target weights as computed with the online regression and the ridge regression method, respectively. Apart from the weights' illustration, the format and the coloring of the whole figure follows the ones of Fig. 2, however it is noted that the orange elements here denote the outcome of the execution of the proposed methodology using the ridge regression method.

$\hat{\boldsymbol{\omega}}_{t_k}$. The suggested approach, which performs small updates to previous estimations as new observations are received, is compared with a batch method that computes the target weights from scratch using all collected data so as to provide a new estimation. To ensure a fair comparison and evaluate the impact on the prediction, the same methodology was executed twice under the same parameters and initial conditions, with the only difference relying on the first step: the regression from observations. Particularly, during the first execution the computation of the target weight vector $\hat{\boldsymbol{\omega}}_{t_k}$ is performed online using Eqs. 5, while during the second execution the weights are derived through ridge regression using Eq. 1 after setting the vector $\boldsymbol{x}^{(j)}$ with the observations collected until the current time t_k.

The comparison is held for two different target trajectories to predict, and the results are depicted in Fig. 4. Focusing on the observed part of the trajectory, in the first example (Tr2) the weight estimation is reliable for both approaches since the weights coincide with the observations. However, in the second example (Tr4) there are instances (third column) where the estimation is clearly distorted, most likely due to the change of trajectory's slope. Despite this misrepresentation by the suggested online method, the prediction is quite similar in both approaches, which is also validated by the similar prediction error plot. It seems that the robustness of the complete methodology has considered the temporary inaccuracy as noise, and eliminated its contribution to the prediction. Moreover, the distorted estimation of the old weights at time $t_k = 0.75$ (fifth column) does not affect the outcome, since the prediction is mainly determined by the most recent ones. In conclusion, the online estimation of the target weight vector suggested in Section III-B is preferred over the batch method since it reduces significantly the computational complexity without affecting the prediction.

V. Discussion

The prediction method introduced in this paper addresses the main drawback of other approaches, i.e. the limitation to provide reliable predictions for trajectories that differ significantly from the trained primitives. This claim is validated through a series of experiments where it is showcased that the proposed strategy outperforms the state-of-the-art in such cases. The research findings constitute important progress in a field that recently gains increasing attention in robotic applications, e.g., the prediction of motion intentions.

In future work, we plan to further improve the developed methodology focusing mainly on two evident candidate upgrades. At first, here the estimation of the total duration is computed a priori based on the trained ProMPs, and remains fixed as the observed trajectory evolves. Therefore, in addition to the prediction considered in this paper, the estimation of the duration might also be updated as new observations are received. Moreover, the static nature of the library does not promote lifelong learning. Making the library dynamic by refining existing ProMPs and/or adding new ones when required would result in more reliable predictions from less observations. Eventually, the extension of the current version of the methodology to multi-dimensional ProMPs could be considered, while further evaluation of the prediction under noisy measurements is required.

References

[1] T. Carlson and Y. Demiris, "Human-wheelchair collaboration through prediction of intention and adaptive assistance," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 3926–3931.

[2] G. P. Moustris and C. S. Tzafestas, "Intention-based front-following control for an intelligent robotic rollator in indoor environments," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–7.

[3] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[4] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013. [Online]. Available: https://doi.org/10.1177/0278364913478447

[5] D. Koert, J. Pajarinen, A. Schotschneider, S. Trick, C. Rothkopf, and J. Peters, "Learning intention aware online adaptation of movement primitives," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3719–3726, 2019.

[6] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 299–306.

[7] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 885–892.

[8] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Autonomous Robots*, vol. 42, pp. 631–648, 2018.

[9] E. Corona, A. Pumarola, G. Alenya, and F. Moreno-Noguer, "Context-aware human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] P. A. Lasota and J. A. Shah, "A multiple-predictor approach to human motion prediction," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2300–2307.

[11] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf

[12] O. Dermy, A. Paraschos, M. Ewerton, J. Peters, F. Charpillet, and S. Ivaldi, "Prediction of intention during interaction with icub with probabilistic movement primitives," *Frontiers in Robotics and AI*, vol. 4, 2017. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2017.00045

[13] L. Penco, J.-B. Mouret, and S. Ivaldi, "Prescient teleoperation of humanoid robots," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–8.

[14] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[15] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2002.

[16] K. J. Åström and B. Wittenmark, *Adaptive control*. Courier Corporation, 2013.

[17] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649231201196, 0. [Online]. Available: https://doi.org/10.1177/02783649231201196

[18] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1398–1403.