Dimensionality Reduction of Movement Primitives in Parameter Space

Samuele Tosatto[1], Jonas Stadtmüller[1] and Jan Peters[1,2]

Abstract— Movement primitives are an important policy class for real-world robotics. However, the high dimensionality of their parametrization makes the policy optimization expensive both in terms of samples and computation. Enabling an efficient representation of movement primitives facilitates the application of machine learning techniques such as reinforcement on robotics. Motions, especially in highly redundant kinematic structures, exhibit high correlation in the configuration space. For these reasons, prior work has mainly focused on the application of dimensionality reduction techniques in the configuration space. In this paper, we investigate the application of dimensionality reduction in the parameter space, identifying *principal movements*. The resulting approach is enriched with a probabilistic treatment of the parameters, inheriting all the properties of the Probabilistic Movement Primitives. We test the proposed technique both on a real robotic task and on a database of complex human movements. The empirical analysis shows that the dimensionality reduction in parameter space is more effective than in configuration space, as it enables the representation of the movements with a significant reduction of parameters.

Fig. 1: Our binamual platform performing a pouring task using Pro-Primos (left). The generation of trajectories via PriMos (right): the blue line corresponds to the mean movement, while the dashed lines to the principal movements. By only linearly combining two principal movements, it is possible to generate a wide variety of movements (gray lines).

I. INTRODUCTION

Robot learning is a promising approach to enable more intelligent robotics, which can easily adapt to the user's desires. In recent years, the field of reinforcement learning (RL) has experienced an enormous advance in solving simulated tasks such as board- or video-games [1], [2], [3], in contrast to little improvements in robotics. The major challenges in the direct application of RL to real robotics, are mainly the limited availability of samples and the fragility of the system, which disallow the application of unsafe policies. These two disadvantages become even more evident when we consider that usual robotic tasks such as industrial manipulation, are defined in a high dimensional state and action space. A usual approach in the application of RL to robotics, is to initialize the policy via imitation learning [4], [5], [6], [7]. In the past, there has been some effort in providing a safe representation of the policy for robotics, mainly by the means of Movement Primitives (MPs) [8], [9], [10], [11]. The general framework of MPs has been extensvely studied and employed in a large variety of settings [12], [13], [14], [15], [16]. MPs have been shown to be effective when used for the direct application of RL to robotics [17], [18], [19]. The drawback of the general framework of MPs, is the usual high number of parameters. In fact, in the configuration-space, the number of parameters is equal to
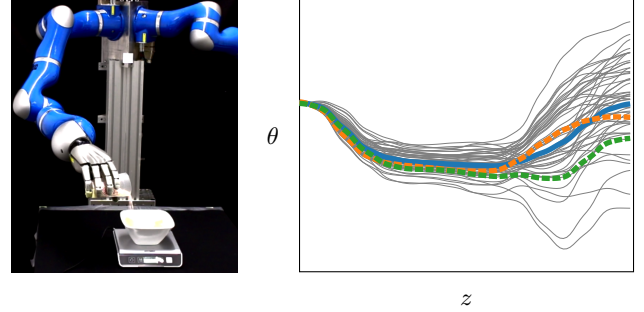
the number of basis functions (typically greater than 10), times the number of degree-of-freedom (DoF). The number of parameters grows even quadratically w.r.t. the number of basis function and DoF when we want to represent the full covariance matrix in the case of Probabilistic Movement Primitives (ProMPs) [11], [20]. In recent years, many authors have proposed techniques to solve the problem by using a latent representation of the robot's kinematic structure [21], [22], [23], [24]. Especially for complex systems, such as humanoids, many DoF are redundant, and a compressed representation of the configuration-space results in a lighter parametrization of the MP. However, this approach is less effective in the case of a robotic arm with fewer DoF Moreover, the parametrization of the MPs would be still linear w.r.t. the number of basis functions.

In this article, we build on the general framework of ProMPs and we propose to apply the dimensionality reduction in parameter space. In tasks where there is a high correlation between the movements, it makes more sense to seek a compressed representation of the movement instead of the configuration-space. To this end, we propose an approach where the movements can be seen as a linear combination of *principal movements* (Fig.1). We enrich our framework with a probabilistic treatment of the parameters, so that our approach inherits all the properties of ProMPs. We analyze the benefit of dimensionality reduction in the parameter space w.r.t. in configuration space, both on a challenging human motion reconstruction and in a robotic pouring task (Fig. 1). Our findings show that our proposed approach

[1]Department of Computer Science, Technische Universität Darmstadt, 7500 Darmstadt, Germany.

[2]Department for Empirical Inference and Machine Learning, Max Planck Institute for Intelligent Systems, 7500 Stuttgart, Germany. Correspondence to samuele.tosatto@tu-darmstadt.de.
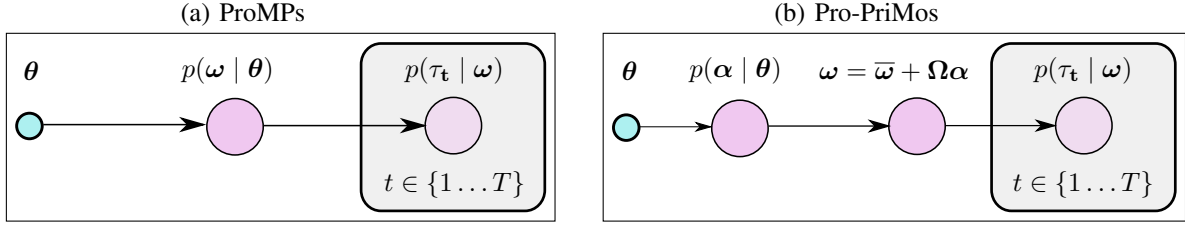
Fig. 2: Graphical model of the ProMPs framework (a) compared to Pro-PriMos (b). The main idea of Pro-PriMos is to generate a distribution of linear combinations of the principal-movements $\mathbf{\Omega}$.

achieve satisfying accuracy even with a significant reduction of parameters.

II. RELATED WORK

The issue of dimensionality reduction in the context of motor primitives has been extensively studied. To achieve dimensionality reduction for Dynamic Movement Primitives (DMPs) [9], the *autoencoded dynamic movement primitives* model proposed in [23], uses to find a representation of the movements in a latent feature space, while in [24] the DMPs are embedded into the latent space of a time-dependent variational autoencoder. In [22] a linear projection in the latent space of the configuration space is considered, as well as the adaption of the projection matrix in the RL context. Interestingly, they consider the possibility to address the dimensionality reduction in the parameter space, but they discard this option as the projection matrix is more difficult to adapt since it is of higher dimensions. We agree with this argument, however, when we assume that the projection matrix can be considered fixed this argument is not valid anymore. The dimensionality reduction of ProMPs has been addressed in [25], where the authors compare PCA versus an expectation-maximization approach in configuration space. Until this point, all the literature is focused in finding a mapping between configuration space and a latent space, and proposing the learning of the MPs in this lower representation. However, this approach results to be more efficient in complex kinematic structures, such as the human body, where the high number of joints facilitate the possibility of redundancies in the configuration space. Moreover, this reduction, does not affect the intrinsic high-dimensional nature of the MPs, which requires usually a high number of basis functions.

A way to overcome this problem is to focus in the parameter space of the movement primitives. The dimensionality reduction in this case exploits similarities between movements, and high correlations between parameters. To the best of our knowledge, the only work performing a reduction in parameter is [26]. The proposed setup is however complex as it considers a fully hierarchical Bayesian setting, where the movements are encoded by a mixture of Gaussian models. They learn those parameters using variational inference. Furthermore, their approach does not address the question of whether the parameter space reduction is more convenient or not.

In our paper, we want to focus on the comparison between the dimensionality reduction in configuration and parameter space, arguing that the latter is more convenient. To this end, we propose the Principal Movements (PriMos) framework, which enables the selection of *principal movements* and the subsequent representation of the movements in this convenient space. We extend PriMos to incorporate a probabilistic treatment of the parameters (Pro-PriMos), in a similar way to the ProMPs framework. Our approach only adds a linear transformation to the framework already developed by Parachos et al., as depicted in Fig. 2, therefore it maintains all the properties exposed by the ProMPs, such as time modulation, movements co-activation or movement conditioning. To maintain the method simple, we select PCA for the dimensionality reduction, even if more sophisticated techniques can be used.

III. THE PRINCIPAL MOVEMENT FRAMEWORK

Machine learning, should allow the robotic agent to interact with the real world in a non-predetermined way. However, the possible generated movement should be smooth, and possibly constrained to be safe (not colliding with other objects, without excessive speed or accelerations, etc). These issues require a whole field of research to be solved. MPs are useful in this context. The idea behind MPs, is to restrict the class of all possible robotic movements to a specific parameterized class usually by linearly combining a set of parameters with a set of features. For a particular class of features (e.g., radial basis function), the movements are guaranteed to be smooth. However, sometimes restricting the space of robotic movement to be smooth is not enough, and we want also the movements to be distributed similarly to some demonstration provided by a human expert. The ProMPs provides a probabilistic treatment of the movement's parameters [11], [20]. Both frameworks are useful and can be coupled with RL. In the following, we introduce the Principal Movements (PriMos) framework, as well as its probabilistic extension (Pro-PriMos). In Section III-A, we introduce the formal notation of the MPs framework. In Sections III-B and III-C, we introduce PriMos and Pro-PriMos relying on the assumption that a set of *principal movement* is given. In Section III-D we propose a technique to find the *principal movements*, offering a complete algorithm for dimensionality reduction for MPs.

A. Preliminaries

Let us concisely introduce the notation and formalize the classic MPs framework. For simplicity, we will first consider only a one-dimensional trajectory $\boldsymbol{\tau}^\mathsf{T} = [\tau_1, \ldots, \tau_T]^1$, where τ_i represent the joint's position at time t_i. In order to introduce the *time modulation*, we use a *phase*-vector $\mathbf{z}$ where $z_i = (t_i - t_1)/(t_T - t_1)$. Moreover, we consider n normalized radial-basis functions $\Phi_i : \mathbb{R} \to \mathbb{R}$, usually, ordered to evenly cover the phase-space, they are centered between $[-2h, 1 + 2h]$ where h is the bandwidth. At time t the features are described by a n-dimensional column vector $\boldsymbol{\Phi}_t$ where $\Phi_{t,i} = \Phi_i(z_t)$. Assuming that the observations are perturbed by a zero-mean Gaussian noise ϵ_t with variance σ_τ, we want to represent the movement $\tau_t = \boldsymbol{\Phi}_t^\mathsf{T}\boldsymbol{\omega} + \epsilon_t$ as a linear combination of parameters $\boldsymbol{\omega} \in \mathbb{R}^n$. , we obtain the following *maximum-likelihood estimation* (MLE) problem

$$\max_\omega \mathcal{L}(\boldsymbol{\omega}) = \max_\omega \prod_t \mathcal{N}(\tau_t \mid \boldsymbol{\Phi}_t^\mathsf{T}\omega, \sigma_t). \qquad (1)$$

Equation (1) can be solved with usual linear regression, or more commonly with Ridge-regression (which corresponds to have a prior Gaussian distribution on ω, $\mathcal{N}(\omega_i | 0; \sigma_\omega)$) that leads to better generalization and is more numerically stable,

$$\boldsymbol{\omega} = (\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi} + \lambda\mathbb{I})^{-1}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\tau} \qquad (2)$$

where λ is the ridge penalization term and $\boldsymbol{\Phi}^\mathsf{T} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \ldots, \boldsymbol{\Phi}_T]$. However, there is the possibility to extend this setting with a probabilistic treatment of the movement parameters. In this case, we want to find the best distribution over parameters which encode a distribution of trajectories, i.e.,

$$p(\boldsymbol{\tau} \mid \mu_\omega, \Sigma_\omega) = \int p(\boldsymbol{\tau} \mid \omega)\mathcal{N}(\boldsymbol{\omega} \mid \mu_\omega; \Sigma_\omega)\mathrm{d}\boldsymbol{\omega}. \qquad (3)$$

This is the setting described with ProMPs in [11]. For a finite set of stroke-movements $\{\boldsymbol{\tau}_i\}_{i=1}^m$, it is possible to use the parameters $\boldsymbol{\omega}_i$ estimated for each trajectory $\boldsymbol{\tau}_i$ in order to estimate $\boldsymbol{\mu}_\omega$ and $\boldsymbol{\Sigma}_\omega$, i.e.,

$$\boldsymbol{\mu}_\omega = \frac{1}{m}\sum_{i=1}^m \boldsymbol{\omega}_i, \quad \boldsymbol{\Sigma}_\omega = \frac{1}{m}\sum_{i=1}^m (\boldsymbol{\omega}_i - \boldsymbol{\mu}_\omega)(\boldsymbol{\omega}_i - \boldsymbol{\mu}_\omega)^\mathsf{T}. \qquad (4)$$

In a more realistic scenario, we need to represent multiple joints. Assuming a system with d joints, we can encode $\boldsymbol{\tau}^\mathsf{T} = [\tau_{1,1}, \ldots, \tau_{1,T}, \ldots, \tau_{d,T}]$ where $\tau_{i,j}$ is the position of the ith joint at time t_j, $\boldsymbol{\Psi} = \mathbb{I} \otimes \boldsymbol{\Phi}$ (where $\mathbb{I}$ is the $d \times d$ identity matrix) and $\boldsymbol{\omega}$ column-vector of length nd encoding the movement's parameters. Similarly to the notation already used, we denote with $\boldsymbol{\Psi}_t$ the feature matrix corresponding to the time step t.

B. Principal Movements

The MPs and ProMPs frameworks, usually require a large number of parameters for encoding the movements. In MPs, we need in fact $d \times n$ parameters (where d is

---

[1]Contrarily to [20], we do not consider the velocities, however they can be incorporated, without any loss of generality.

---

the dimension of the considered movement and n is the number of radial basis functions). rThe choice of the number of radial-basis functions usually depends by both the speed and the complexity of the movements, but most of the time it is greater than ten. For a 7 d.o.f. we, therefore, need at least 70 parameters to encode the MPs. This makes the application of RL challenging for a 7 d.o.f. robot arm. We will assume, from now on, that an oracle (i.e., a human expert or a dimensionality reduction method) will give us the parameters of a *mean-movement* $\overline{\boldsymbol{\omega}}$, and a matrix of *principal-movements* $\boldsymbol{\Omega}^\mathsf{T} = [\boldsymbol{\omega}_1^p, \ldots, \boldsymbol{\omega}_{n_c}^p]$. We therefore want to encode a given trajectory $\boldsymbol{\tau}$ as a linear combination of the principal movements, i.e.,

$$\boldsymbol{\tau}_t = \boldsymbol{\Psi}_t\overline{\boldsymbol{\omega}} + \boldsymbol{\Psi}_t\boldsymbol{\Omega}\boldsymbol{\alpha} + \epsilon_t. \qquad (5)$$

We consider $\boldsymbol{\alpha} \in \mathbb{R}^{n_c}$ the new parameter vector. Note that n_c is generally independent of the number of joints d and of the number of radial basis function n, but instead the choice of n_c is connected to the *complexity* of the movement-space that we aim to represent. The MLE problem

$$\max_{\boldsymbol{\alpha}} \prod_t \mathcal{N}(\boldsymbol{\tau}_t \mid \boldsymbol{\Psi}_t\overline{\boldsymbol{\omega}} + \boldsymbol{\Psi}_t\boldsymbol{\Omega}\boldsymbol{\alpha}, \sigma_t), \qquad (6)$$

induced by (5) has a Ridge regression solution

$$\boldsymbol{\alpha} = (\boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\Psi}^\mathsf{T}\boldsymbol{\Psi}\boldsymbol{\Omega} + \lambda\mathbb{I})^{-1}\boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\Psi}^\mathsf{T}(\boldsymbol{\tau} - \boldsymbol{\Psi}\overline{\boldsymbol{\omega}}). \qquad (7)$$

C. Probabilistic Principal Movements

A probabilistic treatments of the MPs, is often convenient, as it enables the application of statistical tools [11], [20]. To enrich our approach with a probabilistic treatment, we assume our parameter vector $\boldsymbol{\alpha}$ to be multivariate-Gaussian distributed, i.e., $\boldsymbol{\alpha} \sim \mathcal{N}(\cdot \mid \boldsymbol{\mu}_\alpha; \boldsymbol{\Sigma}_\alpha)$. Very similarly to the classic ProMPs approach, we have that

$$p(\boldsymbol{\tau}_t \mid \boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha)$$
$$= \int \mathcal{N}(\boldsymbol{\tau}_t \mid \boldsymbol{\Psi}_t(\boldsymbol{\Omega}\boldsymbol{\alpha} + \overline{\boldsymbol{\omega}}), \boldsymbol{\Sigma}_\tau)\mathcal{N}(\boldsymbol{\alpha} \mid \boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha)\mathrm{d}\boldsymbol{\alpha}$$
$$= \mathcal{N}(\boldsymbol{\tau}_t \mid \boldsymbol{\Psi}_t(\boldsymbol{\Omega}\boldsymbol{\mu}_\alpha + \overline{\boldsymbol{\omega}}), \boldsymbol{\Psi}_t\boldsymbol{\Omega}\boldsymbol{\Sigma}_\alpha\boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\Psi}_t^\mathsf{T} + \boldsymbol{\Sigma}_\tau). \qquad (8)$$

The mean $\boldsymbol{\mu}_\alpha$ and the covariance $\boldsymbol{\Sigma}_\alpha$ can be estimated as similarly done for ProMPs, (Eq. 4)

$$\boldsymbol{\mu}_\alpha = \frac{1}{m}\sum_{i=1}^m \boldsymbol{\alpha}_i, \quad \boldsymbol{\Sigma}_\alpha = \frac{1}{m}\sum_{i=1}^m (\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)(\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)^\mathsf{T}, \qquad (9)$$

where the parameters $\boldsymbol{\alpha}_i$ correspond to the trajectories $\boldsymbol{\tau}_i$.

We assume an affine transformation between the full parameters $\boldsymbol{\omega}$ and the reduced $\boldsymbol{\alpha}$, as shown in Fig. 2, and $\boldsymbol{\alpha}$ is assumed to be Gaussian distributed. Under these assuptions, $\boldsymbol{\omega}$ is also Gaussian distributed, allowing for a mapping from Pro-PriMos to ProMPs,

$$\hat{\boldsymbol{\mu}}_\omega = \boldsymbol{\Omega}\boldsymbol{\mu}_\alpha + \overline{\boldsymbol{\omega}}, \qquad \hat{\boldsymbol{\Sigma}}_\omega = \boldsymbol{\Omega}\boldsymbol{\Sigma}_\alpha\boldsymbol{\Omega}^\mathsf{T}.$$

Hence, the proposed framework enjoys all the properties of the ProMPs, such as movements co-activation or movement conditioning.
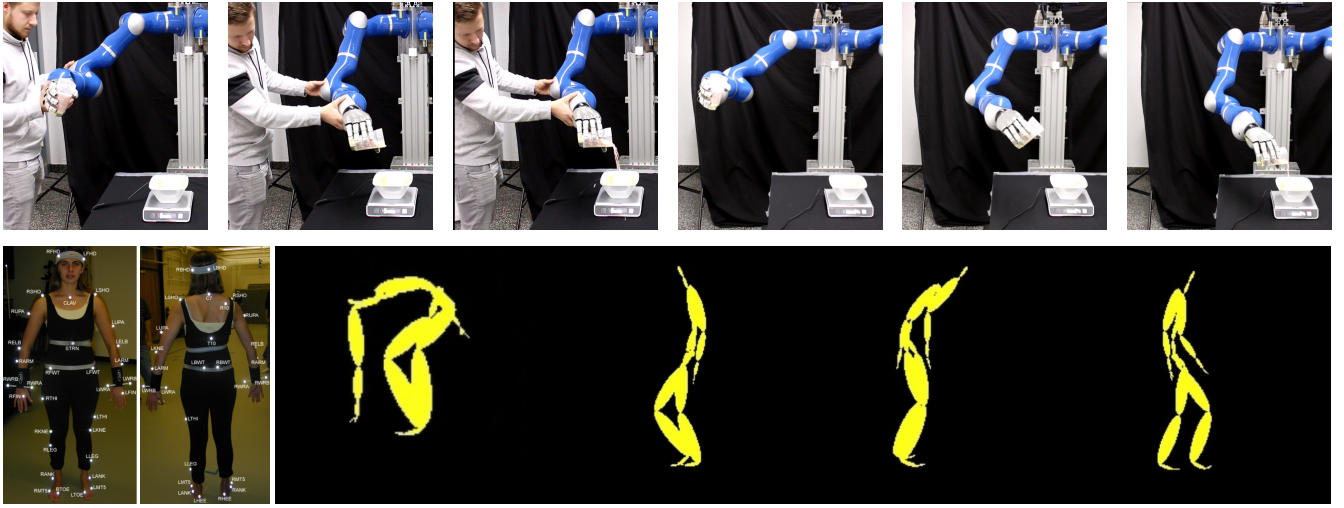
Fig. 3: Top: From left to right, the demonstration in kinestatic teaching mode of the pouring task, and the subsequent reconstruction of the movement with PriMos. Bottom: Markers placements on the human body, and illustrative example of the skeleton reconstruction for a "walk and pick-up" task of subject #143 of the `MoCap` database `http://mocap.cs.cmu.edu`.

D. Inference of the Principal Components

Until this point, we considered the parameters $\overline{\omega}$ and Ω given by an oracle. We mentioned that the underlying assumption of our work is that each movement can be seen as a linear combination of some movements which we call *principal movements*. To be precise, we assume that every trajectory τ can be seen as $\tau = \overline{\tau} + \sum_i \tau_i^p \alpha_i$. Since there exists a linear combination between the trajectory τ and its parameter ω, we can argue, that the same relation exists in the parameter space. Reasoning in the parameter space is more convenient since the trajectories can have different lengths, depending on the sampling frequency and duration. The parameter space is instead fixed. We can say that the parameter space is *frequency and duration agnostic.* There are many techniques that can be used to estimate $\overline{\omega}$ and Ω when they are unknown, but among these, given the Gaussian assumption very made and also thanks to its simplicity, the Principal Component Analysis (PCA) [27] seems to be the most suited. We can think of the movement being Gaussian distributed with a certain mean trajectory $\overline{\tau}$ (with parameters $\overline{\omega}$) and with some orientation. In its geometric interpretation, the PCA extracts the main axis of the covariance matrix (i.e. the Eigenvectors of the covariance), and we can express each point in the new space as a linear combination of the Eigenvectors. Even though the Singular Value Decomposition is a more efficient technique to perform the PCA [28], we continue with the Eigenvector decomposition to maintain this parallelism. Let $\{\mathbf{v}_i\}_{i=1}^{nd}$ be the Eigenvectors of Σ_ω and λ_i the corresponding Eigenvalues (with $|\alpha_1| \geq |\alpha_2| \geq \dots$). To compute the projection matrix Ω we select the first most informative n_c Eigenvectors and we multiply them by the square roots of their corresponding Eigenvalues

$$\Omega = \left[\boldsymbol{v}_1 \sqrt{|\lambda_1|}, \boldsymbol{v}_2 \sqrt{|\lambda_2|}, \dots, \boldsymbol{v}_{n_c} \sqrt{|\lambda_{n_c}|} \right]. \quad (10)$$

Algorithm 1 Pro-PriMos

Require: A dataset of trajectories $\{\mathbf{z}_i, \boldsymbol{\tau}_i\}_{i=1}^m$
  for each $\mathbf{z}_i, \boldsymbol{\tau}_i$ **do**
    Compute $\boldsymbol{\omega}_i$ with (2)
  end for
  Estimate $\boldsymbol{\mu}_\omega$ and $\boldsymbol{\Sigma}_\omega$ with (4)
  Compute $\boldsymbol{\Omega}$ with (10)
  Set $\overline{\boldsymbol{\omega}} := \boldsymbol{\mu}_\omega$
  for each $\mathbf{z}_i, \boldsymbol{\tau}_i$ **do**
    Compute $\boldsymbol{\alpha}_i$ with (7)
  end for
  return $\boldsymbol{\mu}_\alpha = \frac{\sum_i \boldsymbol{\alpha}_i}{n_c}$; $\boldsymbol{\Sigma}_\alpha = \frac{\sum_i (\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)(\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)^\intercal}{n_c}$

This re-scaling gives us the possibility to have the *principal movements* scaled according to the variance of the data, and results in standardized values of $\boldsymbol{\alpha}$ (i.e. $\boldsymbol{\alpha} \sim \mathcal{N}(\cdot \sim \mathbf{0}, \mathbb{I})$). Pro-PriMos is concisely summarized in Algorithm 1.

IV. EMPIRICAL ANALYSIS

We want to compare the dimensionality reduction in parameter space w.r.t. in configuration space. Therefore, we test the two approaches on two different scenario: the reconstruction of highly uncorrelated movements of a human subject, and on a dataset of pouring movements shown with a 7 DoF robotic arm. We also perform a qualitative analysis showing how our proposed method can achieve similar results to standard techniques, dramatically reducing the number of parameters.

a) The Human Motion Dataset: We want to apply PriMos to reconstruct some human motions contained in the `MoCap` dataset. The `MoCap` database contains a wide range of human motions (such as running, walking, picking up boxes, and even dancing) using different subjects (Fig. 3).
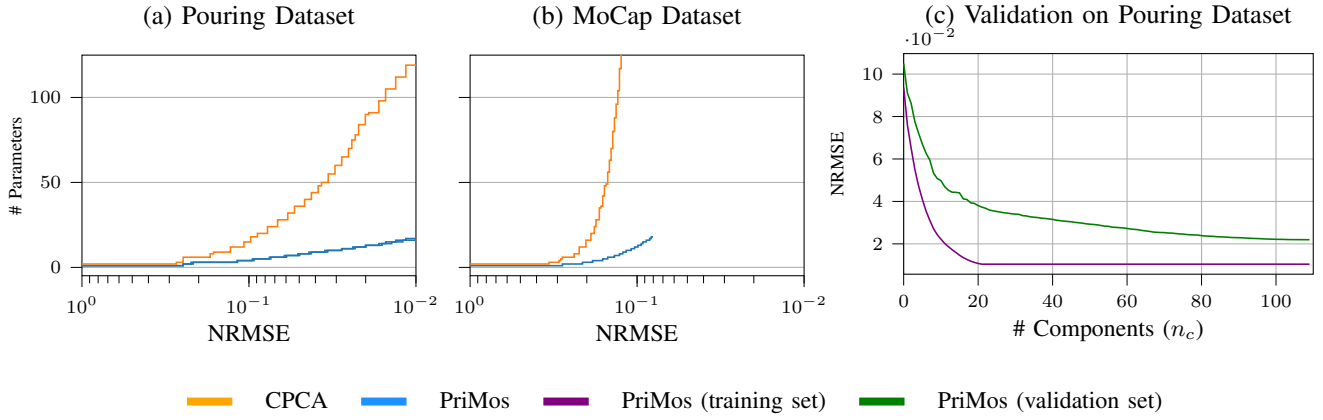
Fig. 4: The number of parameters (y-axis) needed to achieve a certain accuracy (x-axis) (a), (b). As can be seen PriMos achieve better results with less samples. The error has been normalized in order to highlight the difference in complexity between the two datasets. In plot (b), both PriMo and CPCA fail in reaching a NRMSE lower than 0.08. This is due to the complexity of the movement in the dataset, and by the maximum number of radial basis functions (20). In plots (c) we observe the comparison between train (blue) and validation error (green) achieved by PriMos in the pouring datasets.

Human motion is tracked using 41 markers with a Vicon optical tracking system. The data is preprocessed using Vicon Bodybuilder in order to reconstruct a schematic representation of the human body and the relative joint's angles including the system 3D system reference, for a total of 62 values. The human motion is known to be highly redundant (i.e. many configurations are highly correlated), and therefore this is in principle the best case for the dimensionality-reduction in the configuration-space. The presence of many different typologies of movements, makes the application of PriMos challenging since our algorithm relies on the correlation between movements. In our experiments, we use the 42 movements recorded for the subject #143.

b) The Pouring Task: In the pouring task, we use a KUKA light-weight robotic arm with 7 DoF accompanied with a DLR-hand as an end-effector to pour some "liquid" (which, for safety reasons, is replaced by granular sugar). We record some motions from a human demonstrator, pouring some sugar in a bowl. The motion is recorded setting the robotic arm in kinestatic teaching. The quantity of sugar contained in the bowl is recorded by a DYMO digital scale with a sensitivity of $2g$. We aim to reconstruct the movements and understand whether our method is able to pour a similar amount of sugar: this experiment gives us a qualitative understanding, beyond the numerical accuracy analysis, to investigate the effectiveness of our algorithm in a real robot-learning task.

A. Accuracy of Movement Reconstruction

We want to compare the quality of the reconstruction using PriMos, and the dimensionality reduction in the configuration space. In order to be fair, we use PCA as reduction technique for both methods. In the remainder of the paper, we refer to the dimensionality reduction in the configuration space with the acronym "CPCA".

a) Parameters vs Error Analysis: In this experiment, we want to assess the number of parameters needed to obtain a certain accuracy in the reconstruction. We, therefore, apply PriMos and CPCA to both the `MoCap` dataset and a dataset containing 15 different pouring movements. The CPCA admits several configurations for a fixed number of parameters (e.g., 100 parameters can be obtained using 2 components of the PCA and 50 features or 4 components and 25 features, and so on). Therefore we created a grid of configuration in order to obtain the minimal amount of parameters to assess a certain accuracy. In order to make the results comparable between the `MoCap` and the pouring datasets, we use the *normalized root mean square error* (NRMSE) where the normalization is achieved dividing the RMSE by the standard deviation of the dataset. Fig. 4 depicts the NRMSE of PriMos and CPCA for the `MoCap` and pouring datasets. In both cases, the reduction in parameter space performed by PriMos requires significantly fewer parameter than the one performed in joint space by CPCA. Notably, for the `MoCap` database, the maximum number of basis functions chosen is not sufficient to achieve less than the 0.08 reconstruction error.

b) Leave-One-Out Analysis: One might argue that our method "memorizes" the movements contained in the datasets. We want therefore to inspect the ability of PriMos to reconstruct movements not contained in the dataset (in other words, we want to understand if our method *generalizes*). We achieve this analysis using an increasing number of parameters and plotting the NRMSE using a leave-one-out strategy in the pouring dataset (we train on all the possible n subsets of $n-1$ movements, and we test on the movement left out). More precisely, we fit $\overline{\varpi}$ and $\mathbf{\Omega}$ using the training set and then we test the error in the validation set. Fig. 4 shows that the error in the validation sets is not significantly higher than the training-error. Interestingly, while the training error becomes almost constant after 20 components, the validation
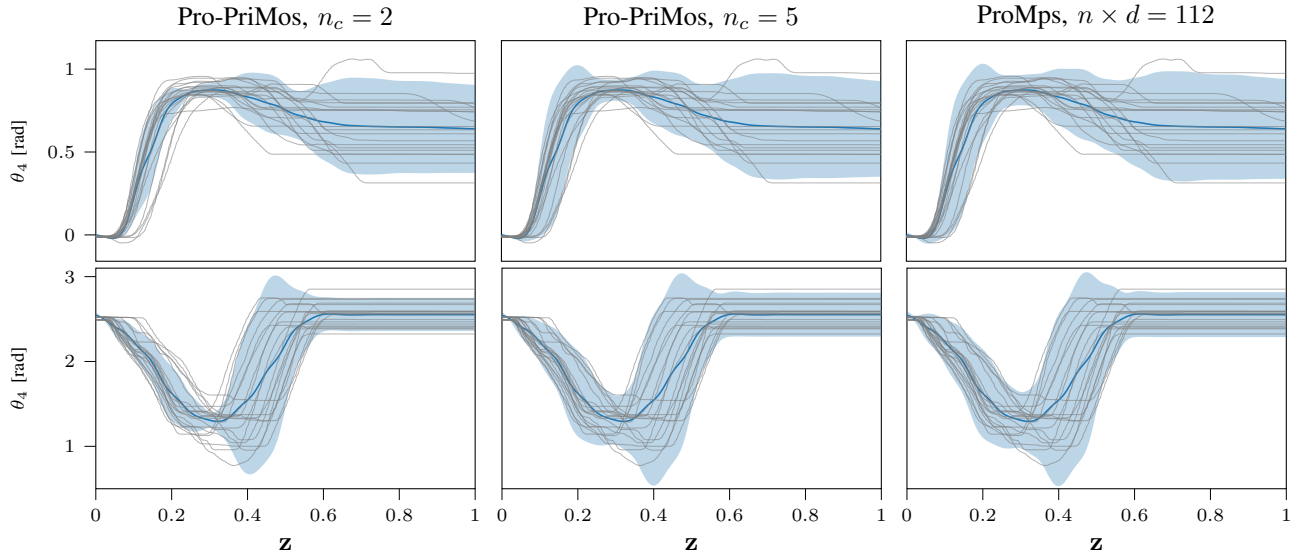
Fig. 5: Mean movement of the joints most involved in the movements (shoulder and elbow), accompanied by the standard deviation ($\pm 2\sigma$) for Pro-PriMos with 2 and 5 components, and with ProMPs with 112 parameters for the mean. Note that the covariance matrix is encoded with 4, 25 and 12544 values respectively.

error keeps getting lower. This behavior suggests that our approach is robust against overfitting.

B. A Qualitative Evaluation

We want to understand the applicability of PriMos to real robotics. The study of the accuracy conducted is important, but on its own, it does not give us a feeling about how the method works in practice. For this reason, we use the 15 trajectories contained in the pouring dataset, and we measured the quantity of sugar poured in the bowl.

a) Single Movement Reconstruction: We run all the 15 trajectories reconstructed with the classic MPs and with PriMos for three times in order to average the stochasticity inherent to the experiment (a slight perturbation of the glass position or of the sugar contained in the glass might perturb the resulting quantity of sugar poured). A video of the demonstration as well of the reconstruction is available as supplementary material. Fig. 6 represents a confusion scatter plot: on the x-axis we have the weight of the sugar observed during the demonstration, while on the y-axis we observe the reconstructed movement both with MPs (112 parameters) and with PriMos (5 parameters). The ideal situation is when the points lye down on the identity line shown in green (perfect reconstruction). We observe that PriMos, except few outliers, reaches a similar accuracy to the MPs, but with 4.5% of the parameters.

b) Probabilistic Representation: An important aspect of our framework is the possibility to represent the *distribution* of movements. For this reason, we show novel movements generated both with ProMPs and Pro-PriMos given 25 demonstrations. The approximation of the full covariance-matrix is usually demanding, as it scales quadratically w.r.t. the number of parameters encoding the mean movement.
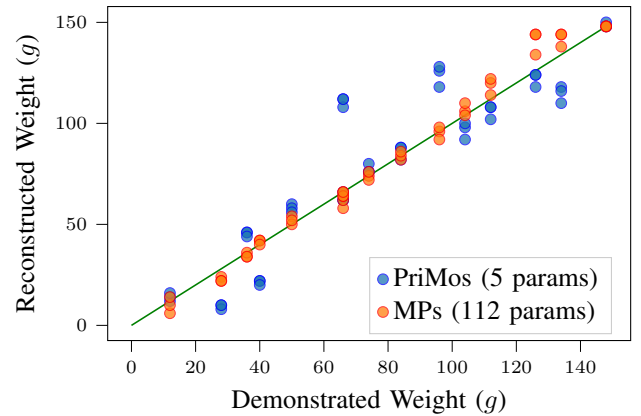

Fig. 6: Precision of the reconstructed weight. The diagonal line represents the best possible reconstruction. For each demonstration, we repeated three times the reconstructed movement in order.

Fig. 5 depicts the standard deviation of the shoulder of the robotic arm, within the demonstrated movement; Pro-PriMos seems to represent the stochasticity very similarly to the ProMPs, although the covariance matrix in the first case requires 5×5 values, in contrast to 112×112 required by the ProMPs. We also note that the variability of the Pro-PriMos seems to be lower than the ProMPs' one: this behavior is explained by the fact that for a fixed set of features, PriMos represents a subset of movements of ProMPs. We furthermore sample 40 movements both from ProMPs and Pro-PriMos with $n_c = 5$, and we measure the quantity of sugar poured. Table I shows that both the methods are able to pour all ranges of sugar (even if with different proportion from the demonstrated data). However, ProMPs occasionally

Method	0-40g	40-80g	80-120g	120-150g	Failure
Demonstrations	20%	33%	27%	20%	0%
ProMPs	43%	7%	20%	17%	13%
Pro-PriMos	53%	20%	3%	24%	0

TABLE I: Measurement of sugar poured from 40 movements sampled both from ProMPs and Pro-PriMos. The methods manage to roughly resemble the original distribution, however ProMPs exhibit unwanted behaviors, like pouring most of the sugar outside the bowl (classified as "failures").

fails to generate a good movement, and it pours the sugar outside the bowl. The video in the Supplementary Materials shows this situation.

V. CONCLUSION

The main contribution of this paper is the analysis of the dimensionality reduction in parameter space in the context of MPs. Our findings suggest that this reduction is more efficient than the one in the configuration space. The novel approach (PriMos), which operates dimensionality reduction in the parameter space using the principal component analysis, is enriched with a probabilistic treatment of the parameters in order to inherit all the convenient properties of the ProMPs. We tested our approach both on a robotic task as well as in a challenging dataset of human movements. Our method compares well against the dimensionality reduction in configuration space, and exhibits a significant reduction of parameters with a modest loss of accuracy, even in the probabilistic setting. We argue that these insights are helpful to develop more efficient robot learning techniques.

As future work, we will investigate the application of different dimensionality reduction techniques in parameter space, with a special focus in the RL context.

ACKNOWLEDGMENT

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: http://www.nature.com/articles/nature14236

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," in *International Conference on Learning Representations*, 2016, arXiv: 1509.02971. [Online]. Available: http://arxiv.org/abs/1509.02971

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[4] S. Schaal, "Is Imitation Learning the Route to Humanoid Robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[5] A. Billard and R. Siegwart, "Robot Learning from Demonstration," *Robotics and Autonomous Systems*, vol. 2, no. 47, pp. 65–67, 2004.

[6] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A Survey of Robot Learning from Demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[7] M. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, "Towards Robust Skill Generalization: Unifying Learning from Demonstration and Motion Planning," in *Intelligent robots and systems*, 2018.

[8] A. D'Avella, P. Saltiel, and E. Bizzi, "Combinations of Muscle Synergies in the Construction of a Natural Motor Behavior," *Nature Neuroscience*, vol. 6, no. 3, pp. 300–308, 2003.

[9] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning Movement Primitives," in *Robotics research. The eleventh international symposium*. Springer, 2005, pp. 561–572.

[10] S. M. Khansari-Zadeh and A. Billard, "Learning Stable Nonlinear Dynamical Systems with Gaussian Mixture Models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[11] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic Movement Primitives," in *Advances in Neural Information Processing Systems (NIPS)*. mit press, 2013.

[12] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction Primitives for Human-Robot Cooperation Tasks," in *IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[13] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, "Learning Interaction for Collaborative Tasks with Probabilistic Movement Primitives," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 527–534.

[14] D. Koert, G. Maeda, R. Lioutikov, G. Neumann, and J. Peters, "Demonstration Based Trajectory Optimization for Generalizable Robot Motions," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 515–522.

[15] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters, "Active Incremental Learning of Robot Movement Primitives," 2017.

[16] S. Stark, J. Peters, and E. Rueckert, "Experience Reuse with Probabilistic Movement Primitives," *arXiv preprint arXiv:1908.03936*, 2019.

[17] J. Peters and S. Schaal, "Reinforcement Learning of Motor Skills with Policy Gradients," *Neural networks*, vol. 21, no. 4, pp. 682–697, 2008.

[18] J. Kober and J. R. Peters, "Policy Search for Motor Primitives in Robotics," in *Advances in Neural Information Processing Systems*, 2009, pp. 849–856.

[19] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to Select and Generalize Striking Movements in Robot Table Tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[20] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using Probabilistic Movement Primitives in Robotics," *Autonomous Robots (AURO)*, no. 3, pp. 529–551, 2018.

[21] A. Colomé and C. Torras, "Dimensionality Reduction and Motion Coordination in Learning Trajectories with Dynamic Movement Primitives," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1414–1420.

[22] A. Colomé, G. Neumann, J. Peters, and C. Torras, "Dimensionality Reduction for Probabilistic Movement Primitives," in *International Conference on Humanoid Robots*. IEEE, 2014, pp. 794–800.

[23] N. Chen, J. Bayer, S. Urban, and P. Van Der Smagt, "Efficient Movement Representation by Embedding Dynamic Movement Primitives in Deep Autoencoders," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 434–440.

[24] N. Chen, M. Karl, and P. Van Der Smagt, "Dynamic Movement Primitives in Latent Space of Time-Dependent Variational Autoencoders," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 629–636.

[25] A. Colomé and C. Torras, "Dimensionality Reduction for Dynamic Movement Primitives and Application to Bimanual Manipulation of Clothes," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 602–615, 2018.

[26] E. Rueckert, J. Mundo, A. Paraschos, J. Peters, and G. Neumann, "Extracting Low-Dimensional Control Variables for Movement Primitives," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1511–1518.

[27] K. Pearson, "On Lines and Planes of Closest fit to Systems of Points in Space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[28] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. JHU press, 2012.