Hazara, Murtaza; Kyrki, Ville

Speeding Up Incremental Learning Using Data Efficient Guided Exploration

Published in:
Proceedings of the 2018 IEEE International Conference on Robotics and Automation, ICRA 2018

DOI:
10.1109/ICRA.2018.8461241

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Hazara, M., & Kyrki, V. (2018). Speeding Up Incremental Learning Using Data Efficient Guided Exploration. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation, ICRA 2018* (pp. 5082-5089). (IEEE International Conference on Robotics and Automation). IEEE.
https://doi.org/10.1109/ICRA.2018.8461241

Speeding Up Incremental Learning Using Data Efficient Guided Exploration

Murtaza Hazara and Ville Kyrki
Department of Electrical Engineering and Automation
Aalto University, Espoo, Finland
{murtaza.hazara, ville.kyrki}@aalto.fi

***Abstract*— To cope with varying conditions, motor primitives (MPs) must support generalization over task parameters to avoid learning separate primitives for each situation. In this regard, deterministic and probabilistic models have been proposed for generalizing MPs to new task parameters, thus providing limited generalization. Although generalization of MPs using probabilistic models has been studied, it is not clear how such generalizable models can be learned efficiently.**

Reinforcement learning can be more efficient when the exploration process is tuned with data uncertainty, thus reducing unnecessary exploration in a data-efficient way. We propose an empirical Bayes method to predict uncertainty and utilize it for guiding the exploration process of an incremental learning framework. The online incremental learning framework uses a single human demonstration for constructing a database of MPs. The main ingredients of the proposed framework are a global parametric model (GPDMP) for generalizing MPs for new situations, a model-free policy search agent for optimizing the failed predicted MPs, model selection for controlling the complexity of GPDMP, and empirical Bayes for extracting the uncertainty of MPs prediction.

Experiments with a ball-in-a-cup task demonstrate that the global GPDMP model generalizes significantly better than linear models and Locally Weighted Regression especially in terms of extrapolation capability. Furthermore, the model selection has successfully identified the required complexity of GPDMP even with few training samples while satisfying the Occam Razor's prinicple. Above all, the uncertainty predicted by the proposed empirical Bayes approach successfully guided the exploration process of the model-free policy search. The experiments indicated statistically significant improvement of learning speed over covariance matrix adaptation (CMA) with a significance of $p = 0.002$.

I. INTRODUCTION

Learning a skill in a perturbed environment often requires practising it under various conditions. For example, to learn to score in basketball, an individual needs to practice throwing from different locations. Subsequently, generalizing to a new situation (e.g. location) becomes easier as the individual learns incrementally the underlying regularities of the task. Incremental learning has been studied in the context of iterative learning control (ILC) where a desired trajectory is adapted to a known reference trajectory in an online incrementally manner [1]. However, in this paper, we propose incremental learning in the context of reinforcement learning (RL) where such a reference trajectory is unknown. In fact, finding the reference trajectory for the new perturbed environment is our objective.

*This work was supported by Academy of Finland, decisions #264239 and #268580

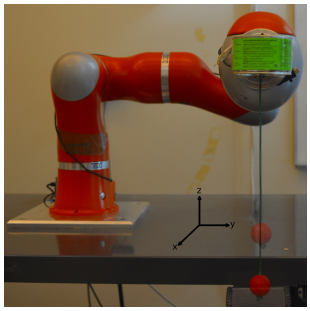


Fig. 1: Ball-in-a-cup game with two different string lengths.

Recently a global parametric model (GPDMP) [2] has been proposed for generalizing an imitated task to a new situation characterized by a measurable task parameter. GPDMP was found to perform better than local models in terms of extrapolation. As an integral part of GPDMP, its complexity has been controlled using a penalized log-likelihood method outperforming traditional model selection methods such as BIC, AIC, and cross validation. GPDMP has been combined with reinforcement learning (RL) in an incremental learning framework for constructing a database (DB) of motor primitives (MPs) [2]. GPDMP extracts the underlying regularities from the DB by mapping MPs to task parameters; GPDMP can then be utilized for predicting MPs for a new situation. Next, RL optimizes the predicted MPs which have led to unsuccessful reenactment of the task. In this case, the learning speed was increased since the predicted MPs have been a better guess than the original imitated MPs. However, RL can be more efficient when the exploration process is tuned with data uncertainty using a data-efficient way.

The main contribution of this paper is guiding the exploration process of a model-free RL using uncertainty predicted for a new task parameter. To our best knowledge, this is the first paper proposing to utilize uncertainty for enhancing incremental learning. First, the mapping of task parameters to MPs is formulated as a probabilistic multivariate regression problem. Next, the hyper-

parameters of the probability distribution provides us with the prediction uncertainty. We have derived an empirical Bayes (EB) method for extracting the hyper-parameter of the probabilistic GPDMP from the DB of MPs while considering their cross-correlation. To our best knowledge, both the mathematical derivation of EB for a multivariate regression problem and its application in guiding the exploration of a model-free RL method are novel.

We selected the ball-in-a-cup task to assess how effective our incremental learning framework is in boosting up the learning speed when generalizing MPs to a new situation using a model-free RL method. The kinematics of the task are encoded using Dynamic Movement Primitives (DMPs). Our experiments demonstrated that the proposed model selection has correctly identified the required complexity for GPDMP outperforming Locally Weighted Regression (LWR) significantly. Above all, the proposed empirical Bayes method has led to a statistically significant improvement of model-free RL speed.

II. Related Work

To adapt learning from demonstration (LfD) models to new environments, the model parameters need to be adjusted according to the task parameters characterizing the new environment. Existing generalizable LfD models can be categorized as (i) generalization by design where the parameters are explicit in the model structure such as the goal of a DMP; (ii) generalization based on local models which uses a weighted combination of trained models; and (iii) generalization by global models.

The task parameters can be mapped to model meta-parameters such as DMP initial position, goal, amplitude, and duration using regression [3]. The approach is suitable for learning tasks where the DMPs are adapted spatially and temporally without changing the overall shape of the motion. However, the skills considered in this paper require adjusting dynamics of motion which is not possibe using this approach.

Researchers have recently shown interest also in generalizing DMP shape parameters to new situations using local models such as support vector machines with local Gaussian kernels [4], Gaussian kernels [5], and a linear mixture of MPs [6]. Gaussian process regression (GPR) [7] and Locally Weighted Regression (LWR) [8], [9] have been the most popular regression models used for the generalization. LWR has the advantage of lower, linear computational complexity compared to the cubic complexity of GPR. On the other hand, GPR is able to provide estimates of prediction uncertainty.

Global parametric models have also been proposed. Both linear [10], [11] and non-linear [2] global models have been considered. The non-linear global models have been shown to outperform local models and the global linear models with respect to their extrapolation capability [2] and their computational complexity is linear. The choice of model order for non-linear parametric models is challenging, but a method for the model choice based on penalized log-likelihood was recently proposed [2]. However, the proposed global models do not provide estimates of prediction uncertainty.

The main contribution of this paper is guiding the exploration process of RL using the predicted uncertainty which is reflected in a covariance matrix extracted from the training samples. This covariance matrix is predicted using empirical Bayes approach which we have derived for multivariate regression case. Furthermore, unlike the basic version of GPR [12], the proposed empirical Bayes approach considers cross-correlation among multiple outputs. We have compared our approach with covariance matrix adaptation (CMA) [13] which updates the covariance matrix by taking the sample covariance matrix. We have observed that our approach outperforms CMA significantly.

III. Method

In this section, we review dynamic movement primitives (DMPs). After that, we clarify the global parametric dynamic movement primitives (GPDMPs) method which incorporates both linear and non-linear parametric models. Besides that, we review model selection approaches and explain our penalized log-likelihood based model selection method.

A. Incremental Learning

We propose an incremental learning framework for constructing a database (DB) of MPs automatically and generalizing an imitated skill to a new situation characterized by a measurable task parameter. The generalization is achieved using a probabilistic model mapping a new task parameter $\mathbf{l}_n$ to the MPs $\mathbf{w}_n$. This probabilistic function (33) provides us with not only a prediction of MPs for a new situation, but also the uncertainty associated with that prediction. The uncertainty is captured by a covariance matrix $\Sigma(\mathbf{l}_n)$ (34). We apply this information of uncertainty in guiding the exploration process of a policy search based RL. The main ingredients of our generalizable incremental learning framework as shown in Fig. 2 are GPDMP, model selection, empirical Bayes, model-free policy search, and a DB of MPs.

GPDMP extracts the underlying regularities in the DB of MPs. Such regularities are reflected in a global parametric function (8). When we encounter a new situation characterized by task parameter $\mathbf{l}_n$, new MPs $\mathbf{w}_n$ are predicted. If the prediction fails to reproduce the skill successfully, we apply a policy search method such as PoWER [14] for optimizing the predicted MPs $\mathbf{w}_n$.

The model-free RL approach updates the predicted MPs $\mathbf{w}_n$ iteratively. In each iteration, a noise vector is sampled from a multivariate Gaussian distribution with zero-mean and a covariance matrix. The structure of the covariance matrix is a key factor influencing the convergence rate of RL. We have already [15] proposed a pre-structured covariance matrix from which a correlated smooth noise vector is sampled providing safe exploration. In this paper, the covariance is modelled as a hyper-parameter of the

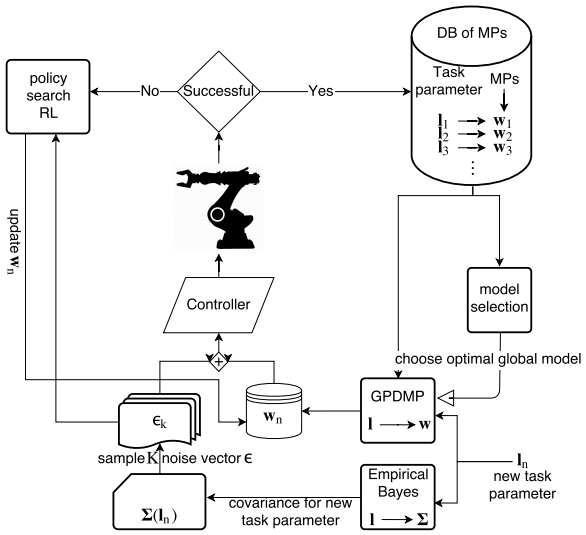


Fig. 2: Incremental learning framework.

GPDMP instead. We apply empirical Bayes for learning the covariance from the DB of MPs. This is achieved by maximizing an evidence function (39) which we have derived for multivariate normal distribution.

The optimized MPs $\mathbf{w}_n$ leading to a successful re-enactment of the skill will be added to the DB. As new MPs are added to the DB, the underlying parametric model of the GPDMP is updated using a novel penalized log-likelihood based model selection method (9). We have already shown [2] that this model selection method is suitable for online incremental learning because it works even with few training samples while traditional model selection methods such as AIC and BIC fail.

To put it in a nutshell, GPDMP and empirical Bayes are providing RL with a good starting point of both MPs and covariance for a new situation; RL, on the other hand, optimizes the predicted MPs in the new situation; thus, providing GPDMP and the empirical Bayes with more training samples enhancing their prediction accuracy. In this way, a DB of MPs is built incrementally and in an online manner.

B. Dynamic Movement Primitives

DMPs [16] encode a policy for a one-dimensional system using two differential equations. The first differential equation $\dot{z} = -\tau\alpha_z z$ formulates a canonical system where z denotes the phase of a movement; $\tau = \frac{1}{T}$ represents the time constant where T is the duration of a demonstrated motion, and α_z is a constant controlling the speed of the canonical system. This first order system resembles an adjustable clock driving the transform system $\frac{1}{\tau}\ddot{x} = \alpha_x(\beta_x(g-x) - \dot{x}) + f(z;\mathbf{w})$ consisting of a simple linear dynamical system acting like a spring damper perturbed by a non-linear component (forcing function) $f(z;\mathbf{w})$. x denotes the state of the system, and g represents the goal. The linear system is critically damped by setting the gains as $\alpha_x = \frac{1}{4}\beta_x$. The forcing function

$$f(z;\mathbf{w}) = \mathbf{w}^T\mathbf{g} \tag{1}$$

controls the trajectory of the system using a time-parameterized kernel vector $\mathbf{g}$ and a modifiable policy parameter vector (shape parameters) $\mathbf{w}$. Each element of the kernel vector

$$[\mathbf{g}]_n = \frac{\psi^n(z)z}{\sum_{i=1}^{N}\psi^i(z)}(g - x_0) \tag{2}$$

is determined by a normalized basis function $\psi^n(z)$ multiplied by the phase variable z and the scaling factor $(g - x_0)$ allowing for the spatial scaling of the resulting trajectory.

C. Global Parametric Dynamic Movement Primitives

Using DMPs, a task can be imitated from a human demonstration; however, the reproduced task cannot be adapted to different environment conditions. To overcome this limitation, we have integrated a parametric model to DMPs capturing the variability of a task from multiple demonstrations. We transform the basic forcing function (3) into a parametric forcing function [2]

$$f(z,\mathbf{l};\mathbf{w}) = \mathbf{w}(\mathbf{l})^T\mathbf{g} \tag{3}$$

where the kernel weight vector $\mathbf{w}$ is parametrized using a parameter vector $\mathbf{l}$ of measurable environment factors.

We model the dependency of the weights with respect to parameters as a linear combination of J basis vectors $\mathbf{v}_i$ with coefficients depending on parameters in a non-linear fashion,

$$\mathbf{w}(\mathbf{l}) = \sum_{i=0}^{J-1}\phi_i(\mathbf{l})\mathbf{v}_i = \mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}) \tag{4}$$

where $\mathbf{V}$ is a $J \times N$ matrix of parameters with N referring to the number of kernels $\mathbf{g}$. $\boldsymbol{\phi}(\mathbf{l})$ is a J dimensional column vector with elements $\phi_j(\mathbf{l})$. For example, the non-linear basis $\phi_j(\mathbf{l})$ for a polynomial model in one parameter is $\phi_j(l) = l^j$. The formulation captures linear models such as [11] as a special case. For a chosen non-linear basis (known functions ϕ_i), the basis vectors can be calculated by minimizing the difference between modelled and initial non-parametric DMP shape parameters,

$$\arg\min_{\mathbf{V}}\sum_{k=1}^{K}\|\mathbf{w}(\mathbf{l}_k) - \mathbf{w}_k\|_2 \tag{5}$$

where $\mathbf{w}_k$ denotes the initial weight vector of a non-parametric DMP optimized for parameter values $\mathbf{l}_k$. The initial weights can be merely imitated from a human demonstration using weighted linear regression [16] or improved using a policy search method [17]. In either case, reproducing an imitated task using $\mathbf{w}_k$ should lead to a successful performance in an environment parametrized by $\mathbf{l}_k$.

In order to solve (5), one needs to construct the design matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\mathbf{l}_1) & \phi_2(\mathbf{l}_1) & \dots & \phi_J(\mathbf{l}_1) \\ \phi_1(\mathbf{l}_2) & \phi_2(\mathbf{l}_2) & \dots & \phi_J(\mathbf{l}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{l}_K) & \phi_2(\mathbf{l}_K) & \dots & \phi_J(\mathbf{l}_K) \end{bmatrix} \tag{6}$$

where K denotes the number of initial DMPs weight vectors which must be at least equal to or greater than the order of the model J to avoid unconstrained optimization problems. Furthermore, the rows of the target matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_K^T \end{bmatrix} \tag{7}$$

represent initial DMP weight vectors. We can minimize (5) with respect to the matrix of basis vectors $\mathbf{V}$, giving

$$\hat{\mathbf{V}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{W}. \tag{8}$$

D. Model selection

The best generalization using a parametric regression model can be achieved by choosing an optimal order of complexity for the model, which is addressed in a model selection method such as cross validation, AIC, or BIC. However, we have shown [2] that these traditional model selection methods can fail when only few training sample are available. Hence, we have derived a novel penalized log-likelihood model selection method which chooses a model by minimizing

$$B_M = tr((\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})^T(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})\Sigma_M^{-1}) + J\log K \tag{9}$$

where Σ_M represents a constant covariance matrix which needs to be determined prior to the model selection process. In our experiments, we have selected a scaled identity matrix $s\mathbf{I}$ as the constant covariance matrix where s denotes the scale. The scale can be determined with respect to the magnitude of the error (difference between $\mathbf{w}_k$ and $\hat{\mathbf{V}}^T\phi(\mathbf{l}_k)$). A simple way to estimate the scale is to look at the largest eigenvalue of the MLE estimate of the covariance matrix with linear fitting. The first term in B_M favours better fitting higher order model while the second term discourages a very high order; hence, it guarantees best prediction while avoiding over-fitting.

E. Predictive distribution

Empirical Bayes can be utilized for estimating the hyper-parameters such as the covariance of the noise of the linear regression model. First, we derive the predictive distribution and later an evidence function for multivariate regression. We assume a linear regression model

$$\mathbf{W} = \Phi\mathbf{V} + \mathscr{E}. \tag{10}$$

where $\mathscr{E}$ is the error matrix

$$\mathscr{E} = \begin{bmatrix} \boldsymbol{\epsilon}_1^T \\ \vdots \\ \boldsymbol{\epsilon}_K^T \end{bmatrix} \tag{11}$$

with each row

$$\boldsymbol{\epsilon}_i = \mathbf{w}_i - \hat{\mathbf{V}}^T\boldsymbol{\phi}(\mathbf{l}_i) \tag{12}$$

representing the difference between the ith training sample $\mathbf{w}_i$ and its prediction $\hat{\mathbf{V}}^T\phi(\mathbf{l}_i)$. Furthermore, we assume that the targets (rows of (7)) are independent. Thus, the likelihood of data is

$$\begin{aligned} p(\mathbf{W}|\mathbf{V},\mathbf{l},\mathbf{\Sigma}) &= \prod_{i=1}^{K}\mathcal{N}(\mathbf{w}_i|\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i),\mathbf{\Sigma}) \\ &= \frac{1}{\sqrt{(2\pi)^{NK}|\mathbf{\Sigma}|^N}}\times \\ &\prod_{i=1}^{K}\exp\{-\frac{1}{2}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i))^T\mathbf{\Sigma}^{-1}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i))\} \\ &= (2\pi)^{-\frac{NK}{2}}|\mathbf{\Sigma}|^{-\frac{K}{2}}\times \\ &\exp\{-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i))^T\mathbf{\Sigma}^{-1}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i))\} \end{aligned} \tag{13}$$

where N is the number of DMPs kernels (size of $\mathbf{g}$ in (2)). After completing the square over $\mathbf{V}$ (see [1]), one can write

$$\begin{aligned} &\sum_{i=1}^{N}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i))^T\mathbf{\Sigma}^{-1}(\mathbf{w}_i-\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_i)) = \\ &\mathrm{tr}\{(\mathbf{W}-\mathbf{\Phi}\mathbf{V})^T(\mathbf{W}-\mathbf{\Phi}\mathbf{V})\mathbf{\Sigma}^{-1}\} = \mathrm{tr}\{(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})^T(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})\mathbf{\Sigma}^{-1}\} \\ &+ vec\{\mathbf{V}-\hat{\mathbf{V}}\}^T(\mathbf{\Sigma}^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi})\,vec\{\mathbf{V}-\hat{\mathbf{V}}\} \end{aligned} \tag{14}$$

where vec represents the vectorization operation; tr denotes the trace of a matrix; and $\otimes$ is the Kronecker product. Using (14), the likelihood of target (13) can be rewritten into

$$\begin{aligned} p(\mathbf{W}|\mathbf{V},\mathbf{l},\mathbf{\Sigma}) &= (2\pi)^{-\frac{NK}{2}}|\mathbf{\Sigma}|^{-\frac{K}{2}}\times \\ &\exp(-\frac{1}{2}\mathrm{tr}\{(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})^T(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})\mathbf{\Sigma}^{-1}\})\times \\ &\exp(-\frac{1}{2}vec\{\mathbf{V}-\hat{\mathbf{V}}\}^T(\mathbf{\Sigma}^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi})\,vec\{\mathbf{V}-\hat{\mathbf{V}}\}). \end{aligned} \tag{15}$$

The last term in (15) can be converted into a normal multivariate Gaussians PDFs over basis vectors ($\mathbf{V}$) multiplied by a constant. Therefore, (15) can be reduced to:

$$p(\mathbf{W}|\mathbf{V},\mathbf{l},\mathbf{\Sigma}) = c_l\mathcal{N}(vec(\mathbf{V})|vec(\hat{\mathbf{V}}),\boldsymbol{S}) \tag{16}$$

where

$$\begin{aligned} c_l &= (2\pi)^{-\frac{(J-K)N}{2}}|\mathbf{\Sigma}|^{-\frac{K}{2}}|\boldsymbol{S}| \\ &\times\exp(-\frac{1}{2}\mathrm{tr}\{(\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}})^T((\mathbf{W}-\mathbf{\Phi}\hat{\mathbf{V}}))\mathbf{\Sigma}^{-1}\}) \end{aligned} \tag{17}$$

denotes the constant coefficient of the likelihood and

$$\boldsymbol{S} = (\mathbf{\Sigma}^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi})^{-1} \tag{18}$$

represents the covariance of the estimated basis vectors $\hat{\mathbf{V}}$ (8). Furthermore, we consider a zero-mean isotropic Gaussian as the prior distribution over basis vectors (rows of $\mathbf{V}$)

$$p(\mathbf{V}|\alpha) = \mathcal{N}(vec(\mathbf{V})|0,\alpha^{-1}I) \tag{19}$$

[1] available at `http://irobotics.aalto.fi/pdfs/empirical_Bayes_for_multivariate_regression.pdf`

which is governed by a single precision parameter α. We can now derive the posterior probability over basis vectors $\mathbf{V}$ by applying (16) and (19) and using Bayes formula

$$p(\mathbf{V}|\mathbf{W},\mathbf{l},\mathbf{\Sigma},\alpha) = \frac{p(\mathbf{W}|\mathbf{V},\mathbf{l},\mathbf{\Sigma})p(\mathbf{V}|\alpha)}{\int p(\mathbf{W}|\mathbf{V},\mathbf{l},\mathbf{\Sigma})p(\mathbf{V}|\alpha)d\mathbf{V}} = \frac{c_l\mathcal{N}(vec(\mathbf{V})|vec(\hat{\mathbf{V}}),\boldsymbol{S})\mathcal{N}(vec(\mathbf{V})|0,\alpha^{-1}I)}{\int c_l\mathcal{N}(vec(\mathbf{V})|vec(\hat{\mathbf{V}}),\boldsymbol{S})\mathcal{N}(vec(\mathbf{V})|0,\alpha^{-1}I)d\mathbf{V}}. \quad (20)$$

Both the nominator and the denominator of (20) involve the product of two multivariate Gaussian PDFs leading to another normal PDF (see [1] for a proof)

$$\mathcal{N}(vec(\mathbf{V})|vec(\hat{\mathbf{V}}),\boldsymbol{S})\mathcal{N}(vec(\mathbf{V})|0,\alpha^{-1}I) = c_p\mathcal{N}(vec(\mathbf{V})|\mathbf{u}_N,\mathbf{M}_N) \quad (21)$$

where the covariance

$$\mathbf{M}_N = (\alpha I + \boldsymbol{S}^{-1})^{-1} \quad (22)$$

the mean

$$\mathbf{u}_N = \mathbf{M}_N(\boldsymbol{S}^{-1}vec(\hat{\mathbf{V}})) \quad (23)$$

and the constant coefficient of this product is

$$\begin{aligned} c_p = \exp(-\frac{1}{2}\{&NJlog(2\pi) - log(|\boldsymbol{S}^{-1}|) - NJlog(\alpha) \\ &+ vec(\hat{\mathbf{V}})^T\boldsymbol{S}^{-1}vec(\hat{\mathbf{V}}) + log(|\mathbf{M}_N^{-1}|) \\ &- vec(\hat{\mathbf{V}})^T\boldsymbol{S}^{-1}\mathbf{M}_N\boldsymbol{S}^{-1}vec(\hat{\mathbf{V}})\}). \end{aligned} \quad (24)$$

Next we use the result in (21) for rewriting the posterior distribution of $\mathbf{V}$ (20) into

$$p(\mathbf{V}|\mathbf{W},\mathbf{l},\mathbf{\Sigma},\alpha) = \frac{c_l c_p\mathcal{N}(vec(\mathbf{V})|\mathbf{u}_N,\mathbf{M}_N)}{c_l c_p\int\mathcal{N}(vec(\mathbf{V})|\mathbf{u}_N,\mathbf{M}_N)d\mathbf{V}} = \mathcal{N}(vec(\mathbf{V})|\mathbf{u}_N,\mathbf{M}_N). \quad (25)$$

Now that we have the posterior distribution over basis vectors $\mathbf{V}$, we can make prediction of DMP shape parameters $\mathbf{w}_n$ for new values of task parameter $\mathbf{l}_n$. This requires evaluating the predictive distribution

$$p(\mathbf{w}_n|\mathbf{W},\alpha,\mathbf{\Sigma},\mathbf{l}_n,\mathbf{l}) = \int p(\mathbf{w}_n|\mathbf{V},\mathbf{\Sigma},\mathbf{l}_n)p(\mathbf{V}|\mathbf{W},\mathbf{l},\mathbf{\Sigma},\alpha)d\mathbf{V} \quad (26)$$

where the posterior distribution of $\mathbf{V}$ is given by (25) and the conditional distribution of DMP shape parameters ($\mathbf{w}$) is given by

$$p(\mathbf{w}_n|\mathbf{V},\mathbf{\Sigma},\mathbf{l}_n) = \mathcal{N}(\mathbf{w}_n|\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_n),\mathbf{\Sigma}). \quad (27)$$

Since $\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_n)$ is a vector, we can write

$$\begin{aligned} \mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_n) &= vec(\mathbf{V}^T\boldsymbol{\phi}(\mathbf{l}_n)) \\ &= vec(\boldsymbol{\phi}(\mathbf{l}_n)^T\mathbf{V}) = vec(\boldsymbol{\phi}(\mathbf{l}_n)^T\mathbf{V}I) \\ &= (I\otimes\boldsymbol{\phi}(\mathbf{l}_n)^T)vec(\mathbf{V}) \end{aligned} \quad (28)$$

because $vec(\boldsymbol{ABC}) = (\boldsymbol{C}^T\otimes\boldsymbol{A})vec(\boldsymbol{B})$. Using the result in (28), we can rewrite the conditional distribution of DMP shape parameters (27) into

$$p(\mathbf{w}_n|\mathbf{V},\mathbf{\Sigma},\mathbf{l}_n) = \mathcal{N}(\mathbf{w}_n|\mathbf{A}_c\,vec(\mathbf{V}),\mathbf{\Sigma}) \quad (29)$$

where $\mathbf{A}_c = (I\otimes\boldsymbol{\phi}(\mathbf{l}_n)^T)$. One can see that (26) involves the convolution of two Gaussian distributions (29) and (25). We utilize the result (2.115) in [18] for evaluating the marginal distribution of new DMP shape parameters $\mathbf{w}_n$ (26). Given a marginal Gaussian distribution for $\mathbf{x}$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{\Lambda}^{-1}) \quad (30)$$

and a conditional Gaussian distribution for $\mathbf{y}$ given $\mathbf{x}$ in the form

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b},\mathbf{L}^{-1}) \quad (31)$$

one can show (see [18]) that the marginal distribution of $\mathbf{y}$ is given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\mathbf{\Lambda}\mathbf{A}^T). \quad (32)$$

We can now derive the predictive distribution (26) using the result in (32)

$$p(\mathbf{w}_n|\mathbf{W},\alpha,\mathbf{\Sigma},\mathbf{l}_n,\mathbf{l}) = \mathcal{N}(\mathbf{A}_c\mathbf{u}_N,\mathbf{\Sigma}_N(\mathbf{l}_n)) \quad (33)$$

where the covariance $\mathbf{\Sigma}_N(\mathbf{l}_n)$ of the predictive distribution is given by

$$\mathbf{\Sigma}_N(\mathbf{l}_n) = \mathbf{\Sigma} + \mathbf{A}_c\mathbf{M}_N\mathbf{A}_c^T. \quad (34)$$

The first term in (34) represents the noise on the DMP shape parameters, and the second term reflects the uncertainty associated with the basis vectors $\mathbf{V}$. Both α and $\mathbf{\Sigma}$ are referred to as the hyper-parameters of the predictive distribution and can be estimated by adopting the empirical Bayes framework.

F. Empirical Bayes

In the Empirical Bayes framework, the hyper-parameters of a predictive distribution are found by maximizing the marginal likelihood function achieved by integrating over basis vectors $\mathbf{V}$

$$p(\mathbf{W}|\alpha,\mathbf{\Sigma}) = \int p(\mathbf{W}|\mathbf{V},\mathbf{\Sigma},\mathbf{l})p(\mathbf{V}|0,\alpha)d\mathbf{V}. \quad (35)$$

Using (16), (19) and the result in (21), marginal likelihood function (35) can be rewritten to

$$p(\mathbf{W}|\alpha,\mathbf{\Sigma}) = c_l\times c_p\int\mathcal{N}(vec(\mathbf{V})|\mathbf{u}_N,\mathbf{M}_N)d\mathbf{V} = c_l\times c_p. \quad (36)$$

Furthermore, we set a gamma prior for α

$$p(\alpha|a,b) = ga(\alpha|a,b) = \frac{b^a\alpha^{a-1}e^{-b\alpha}}{\Gamma(a)} \quad (37)$$

and a Wishart prior for $\mathbf{\Sigma}$

$$p(\mathbf{\Sigma}|\mathbf{\Lambda},\nu) = Wi(\mathbf{\Sigma}|\mathbf{\Lambda},\nu) = \frac{|\mathbf{\Sigma}|^{\frac{\nu-N-1}{2}}e^{-\frac{1}{2}tr(\mathbf{\Lambda}^{-1}\mathbf{\Sigma})}}{2^{\frac{\nu N}{2}}|\mathbf{\Lambda}|^{\nu}\Gamma_p(\frac{\nu}{2})} \quad (38)$$

since they are the conjugate priors of the corresponding distributions. Next, we include the hyperpriors over α and Σ in the evidence function

$$\begin{aligned} Ev(\alpha,\mathbf{\Sigma}) &= logp(\mathbf{W}|\alpha,\mathbf{\Sigma}) + logp(\alpha|a,b) + logp(\mathbf{\Sigma}|\mathbf{\Lambda},\nu) \\ &= log(c_l) + log(c_p) + (a-1)log(\alpha) - b\alpha \\ &+ \frac{\nu-N-1}{2}log|\mathbf{\Sigma}| - \frac{1}{2}tr(\mathbf{\Sigma}\mathbf{\Lambda}^{-1}). \end{aligned} \quad (39)$$

b^a from gamma distribution (37) and all the terms in the denominators of (37) and (38) are behaving as constant when maximizing the evidence function (39) with respect to hyper-parameters α and $\mathbf{\Sigma}$; thus, they are not considered in the evidence function (39). The evidence function (39) is governed by four free parameters a, b, ν, $\mathbf{\Lambda}$. In our experiments, we have selected non-informative hyperprior for α by setting $a = 0.001$ and $b = 0.001$ to small numbers. Furthermore, the degree of freedom ν in the Wishart distribution should be bigger than the number of DMPs kernels ($\nu > N+1$); the higher the degree of freedom ν, the more we believe in the scale matrix $\mathbf{\Lambda}$; however, we set it to $\nu = N+2$ where N is the number of DMP kernels (size of $\mathbf{g}$ in (2)). In this case, the hyper-prior is as weak as possible. One can set the scale matrix $\mathbf{\Lambda}$ which is the initial guess of the covariance matrix to an identity matrix; however, we selected a structured covariance matrix $\mathbf{\Lambda} = s_p(\mathbf{H}_2^T\mathbf{H}_2)^{-1}$ where $\mathbf{H}_2$ is a second-order finite difference matrix (see [2]) and the scale s_p for the prior is set to 0.2 percent of the variance of imitated DMP shape parameters. We have already shown [15] that the correlated noise sampled from this structured covariance matrix $s_p(\mathbf{H}_2^T\mathbf{H}_2)^{-1}$ is smooth and leads to safe exploration with a faster convergence rate. Hence, it is a good initial guess for the scale matrix $\mathbf{\Lambda}$. Finally, we utilize numerical optimization for maximizing the evidence function (39) with respect to the hyper-parameters α and $\mathbf{\Sigma}$. Once, these hyper-parameters are found, we can predict the covariance for any new task parameter using (34). We expect that sampling from this predicted covariance matrix leads to a faster convergence rate in a policy search based reinforcement learning (RL).

G. Reinforcement Learning

Executing a DMP with predicted shape parameters might not lead to a successful reproduction of a task. One way to refine the shape parameters is to learn them using policy search reinforcement learning (RL). In this paper, PoWER [14] is utilized updating the DMP shape parameters $\mathbf{w}$ iteratively. In each iteration, stochastic roll-outs of the task are performed, each of which is achieved by adding Gaussian random noise to the DMP shape parameters. Each noisy vector is weighted by the returned accumulated reward. Hence, the higher the returned reward, the more the noisy vector contributes to the updated policy parameters. This exploration process continues until the algorithm converges to the optimal policy. Our complete setup of the PoWER is elaborated in [15].

IV. Experimental Evaluation

We studied experimentally the generalization performance of the proposed incremental learning framework using a ball-in-a-cup task taught to KUKA LBR 4+ initially using kinesthetic teaching. In this section, we explain the ball-in-a-cup task, incremental learning scenario, and the effect of empirical Bayes on speeding up the learning process.

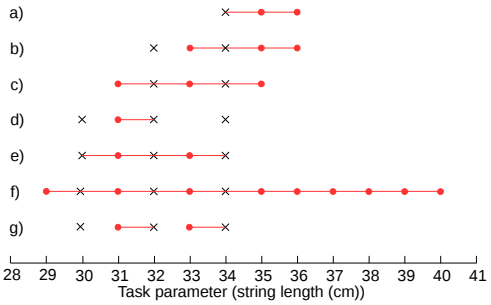


Fig. 3: Validity ranges (red lines) of models learned from database of different sizes. (a) zero order (model selection) global model trained only on one MP. (b) zero order model trained on 2 MPs. (c) linear model (indicated by model selection) trained on the same 2 MPs as in (b). (d), (e) and (f) represent zero, first, and second order models trained on the same DB of 3 MPs.The second order model (f) with the highest extrapolation capability was indicated by the proposed model selection method. (g) Locally weighted regression trained on the same DB of 3 MPs as in (d), (e) and (f).

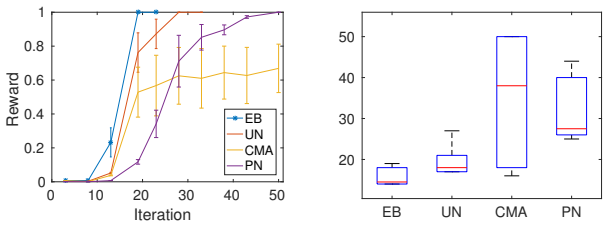
A. Ball-in-a-Cup Task

The ball-in-a-cup game consists of a cup, a string, and a ball; the ball is attached to the cup by the string (see Fig. 1). The objective of the game is to get the ball in the cup by moving the cup in a suitable fashion. We chose the ball-in-a-cup game because variation in the environment can be generated simply by changing the string length. The string length is observable and easy to evaluate, thus providing a suitable task parameter representing the environment variation. Nevertheless, changing the length requires a complex change in the motion to succeed in the game. Hence, the generalization capability of a parametric LfD model can be easily assessed using this game. Similar to our previous set-up in [2], the trajectories along y and z were encoded using separate DMPs. However, in this paper, we utilize 20 kernels. Thus, in total, $N = 40$ parameters need to determined when generalizing to a new task parameter.

B. Incremental Learning

We studied first the generalization performance of global models with different complexities and compared them with LWR. The results are depicted in Fig. 3 where X denotes training samples in the database, and red line the validity region of the model.

We started the incremental learning process in a ball-in-a-cup game with a string length of 34 cm. Using a zero-order global model (Fig. 3.a), the game could be re-enacted successfully for string lengths of 34 to 36 cm. Next, we used the model for predicting an initial policy for string length of 32 cm and optimized it using RL. This newly optimized MP was added to the database, and subsequently the global model and its complexity were updated. This time, the model selection indicated first order for the complexity of the global model. This linear



(a) (b)

Fig. 4: RL convergence rate of different exploration strategies (EB=Empirical Bayes, UN=unstructured noise with a scaled diagonal matrix sI, CMA=Covariance Matrix Adaptation, PN=Pre-structured Noise). (a) Distribution of indexes of first successful RL roll-out. (b) The average reward is displayed here, while the variance is depicted by a vertical bar.

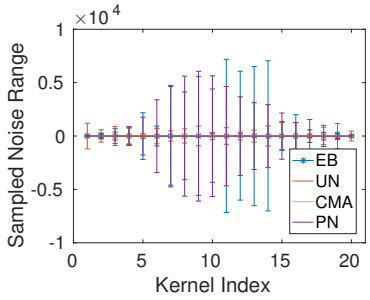


Fig. 5: Variance (shown for every kernel separately) of sampled noise for 10 RL roll-outs at iteration 14th of different exploration strategies (EB=Empirical Bayes, UN=unstructured noise with a scaled diagonal matrix sI, CMA=Covariance Matrix Adaptation, PN=Pre-structured Noise).

model (Fig. 3.c) works successfully for string lengths of 31 to 35 cm. We can see here that a zero-order model (Fig. 3.b) is incapable of interpolating to string length of 32 cm, indicating the necessity of model selection.

Next, an initial policy for string length of 30 cm was estimated using the model learned from the current database, optimized using PoWER, and then added to the database. With this database of three MPs, the model selection indicated a second order model. After that, the generalization capability of a constant, linear, second order and LWR model was tested while fitted to the DB of same three MPs. Both the constant (Fig. 3.d) and LWR (Fig. 3.g) models were not sufficient as they could not even interpolate in the whole range; although, the linear model (Fig. 3.e) could interpolate successfully within the range of training samples, it could not extrapolate at all. The second-order model (Fig. 3.f) could achieve the best extrapolation performance generalizing the task for string length of 29 cm up to 40 cm. This indicates superior generalization capabilities of global models in this task. Furthermore, it indicates that the proposed model selection is able to identify the required complexity.

C. Convergence rate

We next studied the effect of predicted uncertainty on the convergence rate of RL when optimizing the policy parameters for a new task parameter. As a starting point, we utilized the second order model (Fig. 3.f) for predicting MPs for the string length of 28 cm, which led to an unsuccessful re-enactment; thus, we utilized PoWER for optimizing the predicted MPs.

We applied four different approaches for generating the noise in order to provide a testbed for empirical Bayes by comparing their learning speed. In approach PN, we generated the noise samples from a pre-structured covariance matrix ($\mathbf{\Lambda} = s_p(\mathbf{H}_2^T\mathbf{H}_2)^{-1}$). In EB, the pre-structured covariance was exploited as a prior, and a covariance matrix ($\mathbf{\Sigma}(\mathbf{l}_n)$) was predicted by the proposed empirical Bayes approach (34). It took 20 seconds to estimate the covariance components on a Linux machine equipped with an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz.

In UN, an unstructured covariance matrix formed by a diagonal matrix sI was applied and the scale s was fine-tuned specifically for the new task parameter; in fact, the scale parameter s in the diagonal covariance matrix was set equal to 20 percent of the trace of the predicted covariance matrix $s = 0.2\,\mathrm{tr}\{\mathbf{\Sigma}(\mathbf{l}_n)\}$. Changing the scale parameter s to another value did not lead to a better performance. In CMA, covariance matrix adaptation [13] was exploited for updating the fine-tuned diagonal covariance matrix sI.

10 RL roll-outs were performed for each one of these four exploration strategies. The result is displayed in Fig. 4a where the lines show the average reward, while the vertical bars represent the variance of the achieved rewards. We stopped the RL after 50 iterations where some of the roll-outs (yellow line in Fig. 4a) generated by the CMA approach failed to optimize the predicted MPs. This is mainly because CMA is approximating the covariance of the noise using the sample covariance matrix. In this case, a matrix of size (20×20) is approximated by a few samples(7 noise vectors), which is an ill-posed problem. At least 20 samples would be required to get a full rank covariance matrix; this required updating the covariance matrix after every 20 iterations; however, CMA would be totally unnecessary in this case since most of the RL roll-outs performed by the fine-tuned diagonal matrix (orange line in Fig. 4a) converged after approximately 20 iterations.

The unstructured noise is a blind exploration strategy which explores evenly each policy parameter as can be seen from the orange vertical bars in Fig. 5; besides that, we have recently observed that a large scale parameter s was required for optimizing the initial imitated shape parameters, which has led to unsafe exploration with too high acceleration [15]. On the other hand, the pre-structured covariance matrix explore less in the beginning and end, but more in between (where active kernels reside) providing safe exploration trajectories. Nevertheless, neither the fine-tuned unstructured noise nor the pre-structured noise could outperform the empirical Bayes approach (blue line in Fig. 4a).

The proposed empirical Bayes approach has exploited the pre-structured covariance as a prior and by fitting to

the DB of MPs, it found that some kernels such as 11 to 15th (see blue vertical bar in Fig. 5) need to be explored more than the prior indicates (the purple vertical bar in Fig. 5), while some other kernels such as 6th to 10th need to be explored less. This indicates that the structure of the predicted covariance matrix has been effective in speeding up the learning process.

In order to study whether the empirical Bayes increases learning speed, we collected the first successful iteration of every approach in a separate vector which is displayed as a box plot in Fig. 4b. Next, we tested three hypotheses using the Mann-Whitney U test. Under the first null hypothesis, both empirical Bayes and unstructured noise have the same distribution of first successful iteration, which was rejected with a significance of $p = 0.0149$, indicating the superiority of empirical Bayes over unstructured noise. Under the second null hypothesis, empirical Bayes and CMA follow the same distribution, which was also rejected with a significance of $p = 0.002$. The third null hypothesis says that empirical Bayes and the pre-structured approach have the same distribution, which was also rejected with significance of $p = 0.00016$. Hence, the proposed empirical Bayes approach has led to statistically significant improvement in speeding up the convergence rate of RL for the ball-in-a-cup task.

V. Conclusion

In this paper, we proposed an incremental learning framework in the context of RL. The main ingredients of this framework are a global parametric model mapping a task parameter to policy parameters, model selection controlling the complexity of the global model, and empirical Bayes predicting the uncertainty for a new task parameter. The global parametric GPDMP model is simple and can be scaled to accommodate for non-linearities when mapping a task parameter to policy parameters, thus allowing for generalizing a policy to new situations and incremental construction of a database of MPs. We observed that the complexity of the global parametric model was needed to be updated online as new MPs were added to the database, indicating that model selection is integral for constructing online incremental DB of MPs. Experiments showed that the proposed penalized log-likelihood based model selection lead to a global model which is simple, overcomes over-fitting, and performs better than locally weighted regression both in terms of inter- and extrapolation. It also works even with few training samples, which indicates its suitability for online incremental learning. Most importantly, comparing the convergence rate of empirical Bayes with CMA, pre-structured and unstructured noise, we observed that the proposed empirical Bayes approach led to statistically significant improvement in learning speed when generalizing to a new task parameter.

All things considered, only a single human demonstration was needed for constructing the database of MPs. Our experiments demonstrated that the global model worked hand in hand with empirical Bayes, thus providing RL with a more accurate initial policy and a better structure of covariance matrix for generating noise in a new situation resulting in a faster convergence. In return, RL provided the global model and the empirical Bayes with additional training examples enhancing their predictive accuracy.

References

[1] A. Gams, M. Denisa, and A. Ude, "Learning of parametric coupling terms for robot-environment interaction," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 304–309, IEEE, 2015.

[2] M. Hazara and V. Kyrki, "Model selection for incremental learning of generalizable movement primitives," in *18th IEEE International Conference on Advanced Robotics (ICAR 2017), Hong Kong*, 2017.

[3] J. Kober, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 2650, 2011.

[4] B. Da Silva, G. Konidaris, and A. Barto, "Learning parameterized skills," *arXiv preprint arXiv:1206.6398*, 2012.

[5] F. Stulp, G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud, "Learning compact parameterized skills with a single regression," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 417–422, IEEE, 2013.

[6] 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.

[7] D. Forte, A. Gams, J. Morimoto, and A. Ude, "On-line motion synthesis and adaptation using a trajectory database," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1327–1339, 2012.

[8] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[9] B. Nemec, R. Vuga, and A. Ude, "Efficient sensorimotor learning from multiple demonstrations," *Advanced Robotics*, vol. 27, no. 13, pp. 1023–1031, 2013.

[10] A. Carrera, N. Palomeras, N. Hurtós, P. Kormushev, and M. Carreras, "Learning multiple strategies to perform a valve turning with underwater currents using an i-auv," in *OCEANS 2015-Genova*, pp. 1–8, IEEE, 2015.

[11] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Networks*, vol. 24, no. 5, pp. 493–500, 2011.

[12] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[13] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," *arXiv preprint arXiv:1206.4621*, 2012.

[14] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Advances in neural information processing systems*, pp. 849–856, 2009.

[15] J. Lundell, M. Hazara, and V. Kyrki, "Generalizing movement primitives to new situations," in *Conference Towards Autonomous Robotic Systems*, pp. 16–31, Springer, 2017.

[16] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1398–1403, IEEE, 2002.

[17] J. R. Peters, *Machine learning of motor skills for robotics*. PhD thesis, University of Southern California, 2007.

[18] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.