

On Improving the Extrapolation Capability of Task-Parameterized Movement Models

Sylvain Calinon, Tohid Alizadeh and Darwin G. Caldwell

Abstract—Gestures are characterized by intermediary or final landmarks (real or virtual) in task space or joint space that can change during the course of the motion, and that are described by varying accuracy and correlation constraints. Generalizing these trajectories in robot learning by imitation is challenging, because of the small number of demonstrations provided by the user. We present an approach to statistically encode movements in a task-parameterized mixture model, and derive an expectation-maximization (EM) algorithm to train it. The model automatically extracts the relevance of candidate coordinate systems during the task, and exploits this information during reproduction to adapt the movement in real-time to changing position and orientation of landmarks or objects. The approach is tested with a robotic arm learning to roll out a pizza dough. It is compared to three categories of task-parameterized models: 1) Gaussian process regression (GPR) with a trajectory models database; 2) Multi-streams approach with models trained in several frames of reference; and 3) Parametric Gaussian mixture model (PGMM) modulating the Gaussian centers with the task parameters. We show that the extrapolation capability of the proposed approach outperforms existing methods, by extracting the local structures of the task instead of relying on interpolation principles.

I. INTRODUCTION

One important challenge of imitation learning is to design models of multidimensional continuous movements that can fulfil various purposes such as storing, recognizing, predicting and reproducing movements. The models need to be compact to facilitate the joint use of self-refinement learning strategies. The main difficulty is that gestures depend on task parameters such as locations of intermediary via-points or targets (virtual or real) that can locally influence the shape, amplitude, direction and timing of movements. Several denominations have been used to describe these models such as *task-parameterized* [1], *parametric* [2]–[4] or *stylistic* [5].

Research in robot learning by imitation tackles skills of increasing complexity. Often, this complexity does not grow with the size of the training data, which need to be small to keep the skill transfer interaction user-friendly. This trend results in generalization problems requiring stronger extrapolation requirements, see Fig. 1.

Existing methods to tackle such challenge can be classified in three categories:

- 1) Approaches employing M models for the M demonstrations, performed in M different situations (e.g., Gaussian process with trajectory models database) [4], [6]–[11];

The authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. T. Alizadeh is also with the Università degli Studi di Genova. `name.surname@iit.it`.

This work was partially supported by the STIFF-FLOP European project (FP7-ICT-287728), and by the SAPHARI European project (FP7-ICT-287513).

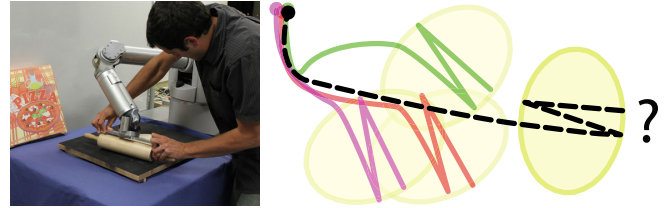


Fig. 1. Illustration of the challenge. The aim is to learn, from very few demonstrations, how to modulate a movement with respect to two different frames of reference. The observed movements (in red, green and violet) consist of smoothly switching from a motion in the robot frame to the motion of rolling a dough (shaped as an ellipse) back and forth, following a specific direction (smallest diameter). The robot is not provided with this information, and must extract the important characteristics of the task to extrapolate the skill to a new situation (in black), that cannot, in this case, be represented as a convex sum of the previously observed situations.

- 2) Approaches employing P models for the P frames of reference that are possibly relevant for the task (e.g., multi-streams methods encoding the same movement in several coordinate systems) [12]–[16];
- 3) Approaches employing a single model whose parameters are modulated by task parameters (e.g., parametric hidden Markov model) [2], [3], [17].

We recently showed in [1] that a promising trend in this last category is to exploit the functional nature of manipulation tasks to build models that can learn the local structures of the task from a low number of demonstrations. The model was employed in the context of early movement acquisition in a humanoid, with simple time-dependent and time-independent gestures (clapping hands and pointing at a toy object).

The purpose of this paper is to compare this model to the three categories of approaches described above. The paper is organized as follows. Examples of models falling in these three categories are first presented in Section II. The proposed approach is then described in Section III. Sections IV and V present the experimental setup and comparison results. Sections VI and VII are discussions and conclusion.

II. PREVIOUS WORK

Various task-parameterized movement models exist, differing both in the research questions being addressed and in the techniques being employed [2]–[11], [14]–[16], [18]. It would not be possible to review the specificities of each model in this paper. We instead give a global overview of existing strategies by classifying them into three main branches. We will focus on task parameters that can be described as coordinate systems, each represented by a position \mathbf{b} (origin

of the observer) and a set of basis vectors $\{e_1, e_2, \dots\}$ forming a transformation matrix $\mathbf{A} = [e_1, e_2, \dots]$. The coordinate systems can include time as coordinate, or any other variable relevant for the task. The frames need to have the same number of rows but can have various numbers of columns (e.g., to consider constraints in both configuration space and task space). We will consider here only squared frames of reference defined by orthogonal coordinate systems in task space.

We will assume that each demonstration $m \in \{1, \dots, M\}$ contains T_m datapoints forming a dataset of N datapoints $\{\xi_n\}_{n=1}^N$ with $N = \sum_m T_m$. Each datapoint $\xi_n = [t_n, \mathbf{x}_n]^T \in \mathbb{R}^{D+1}$ (e.g. $D = 3$ Cartesian datapoints augmented with timestamp t_n) is associated with the observed task parameters $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$ that represent respectively P candidate frames of reference, with offset position vectors $\mathbf{b}_{n,j}$ and linear transformation matrices $\mathbf{A}_{n,j}$.¹ Some task-parameterized approaches work only with task parameters constant during each demonstration. In this case, the task parameters will be described as $\{\mathbf{A}_{m,j}, \mathbf{b}_{m,j}\}_{j=1}^P$, with $m \in \{1, \dots, M\}$.

A. Gaussian process regression (GPR) with trajectory models database

1. Task demonstrations

- Set P (number of candidate frames)
- for** $m \leftarrow 1$ **to** M (for each demonstration)
 - Collect task parameters $\{\mathbf{A}_{m,j}, \mathbf{b}_{m,j}\}_{j=1}^P$
 - for** $n \leftarrow 1$ **to** T_m (for each time step)
 - Collect datapoint $\xi_{m,n}$

end

2. Model fitting

- Set K (number of components in each GMM)
- for** $m \leftarrow 1$ **to** M (for each demonstration)
 - Fit a GMM to ξ_m with EM
 - $\theta_m = \{\pi_{i,m}, \mu_{i,m}, \Sigma_{i,m}\}_{i=1}^K$
 - Associate query points $\mathbf{q}_m = \{\mathbf{A}_{m,j}, \mathbf{b}_{m,j}\}_{j=1}^P$

end

- Precompute $[\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma^2 \mathbf{I}]^{-1} \Theta$ in Eqs (1)-(2)

3. Reproduction

- Set the input \mathcal{I} and output \mathcal{O} elements
- Collect/select new query point $\mathbf{Q}^d = \{\mathbf{A}_j, \mathbf{b}_j\}_{j=1}^P$
- Estimate resulting GMM using Eq. (2)
- for** $n \leftarrow 1$ **to** T (for each reproduction step)
 - Collect/select $\xi_n^{\mathcal{I}}$
 - Use Eq. (3) to retrieve $\xi_n^{\mathcal{O}}$ through GMR as
 - $\xi_n^{\mathcal{O}} | \xi_n^{\mathcal{I}} \sim \mathcal{N}(\hat{\mu}_n^{\mathcal{O}}, \hat{\Sigma}_n^{\mathcal{O}})$

end

Algorithm 1: GPR with trajectory models database.

One possible encoding of task-parameterized movement is

¹It is possible to remove the effects of the task parameters on ξ_n by setting null values in the corresponding elements of vectors $\mathbf{b}_{n,j}$, and identity matrices in the corresponding blocks of $\mathbf{A}_{n,j}$ (e.g., if it is not desired to modulate the duration of the movement with the task parameters).

to fit a model to each demonstration and associate it with a task-specific feature, goal, style variable or perceptual feedback. The challenge of generalizing the observed movement to new situations requires here an effective combination of the existing models to generate new movements that can interpolate or extrapolate the task [4], [6]–[11].

In the special case of task parameters constant during each demonstration, one possible implementation is to build a trajectory models database, where each model and associated task parameters are represented in the form of output variables (model parameters) and query points (task parameters). This information is then used to retrieve new model parameters from the new task parameters. A popular non-linear interpolation approach is to use *Gaussian process* (GP), which has been employed in various Bayesian probability modeling and regression problems [19].

For a set of new query points, and after centering the training data, the joint distribution of the demonstrated and new outputs can be estimated as

$$\begin{bmatrix} \Theta \\ \Theta^d \end{bmatrix} = \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma^2 \mathbf{I} & \mathbf{K}(\mathbf{Q}, \mathbf{Q}^d) \\ \mathbf{K}(\mathbf{Q}^d, \mathbf{Q}) & \mathbf{K}(\mathbf{Q}^d, \mathbf{Q}^d) \end{bmatrix} \right), \quad (1)$$

where \mathbf{Q} is the concatenation of query points \mathbf{q}_m , and Θ the concatenation of outputs θ_m , with $m \in \{1, \dots, M\}$. Squared exponential covariance functions \mathbf{K} are considered, with hyperparameters optimized for the specific extrapolation requirements of the experiment.

The expected outputs $\hat{\Theta}$ associated with the new query points \mathbf{Q}^d is given by

$$\hat{\Theta} = \mathbf{K}(\mathbf{Q}^d, \mathbf{Q}) [\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma^2 \mathbf{I}]^{-1} \Theta, \quad (2)$$

with the covariance of the prediction given by

$$\hat{\Sigma}^{\Theta} = \mathbf{K}(\mathbf{Q}^d, \mathbf{Q}^d) - \mathbf{K}(\mathbf{Q}^d, \mathbf{Q}) [\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{K}(\mathbf{Q}, \mathbf{Q}^d).$$

The above formulation is independent of the trajectory model parameterization employed. We implemented it here as a *Gaussian mixture model* (GMM), and used *Gaussian mixture regression* (GMR) to regenerate new trajectories [12]. The parameters of a GMM with K components are defined by $\{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K$, with π_i the prior (mixing coefficient), μ_i the center, and Σ_i the covariance matrix of the i -th Gaussian component. By defining which variables span for input and output parts (noted respectively by \mathcal{I} and \mathcal{O} superscripts), a block decomposition of the datapoints ξ_n , vectors μ_i and matrices Σ_i can be written as

$$\xi_n = \begin{bmatrix} \xi_n^{\mathcal{I}} \\ \xi_n^{\mathcal{O}} \end{bmatrix}, \quad \mu_i = \begin{bmatrix} \mu_i^{\mathcal{I}} \\ \mu_i^{\mathcal{O}} \end{bmatrix}, \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{\mathcal{I}\mathcal{I}} & \Sigma_i^{\mathcal{I}\mathcal{O}} \\ \Sigma_i^{\mathcal{O}\mathcal{I}} & \Sigma_i^{\mathcal{O}\mathcal{O}} \end{bmatrix}.$$

GMR relies on the learned joint distribution $\mathcal{P}(\xi_n^{\mathcal{I}}, \xi_n^{\mathcal{O}})$ of the data ξ_n . At each reproduction step, the conditional probability $\mathcal{P}(\xi_n^{\mathcal{O}} | \xi_n^{\mathcal{I}})$ is estimated as an output distribution $\mathcal{N}(\hat{\mu}_n^{\mathcal{O}}, \hat{\Sigma}_n^{\mathcal{O}})$, that is also Gaussian, with

$$\hat{\mu}_n^{\mathcal{O}} = \sum_i h_i(\xi_n^{\mathcal{I}}) \left[\mu_i^{\mathcal{O}} + \Sigma_i^{\mathcal{O}\mathcal{I}} \Sigma_i^{\mathcal{I}\mathcal{I}}^{-1} (\xi_n^{\mathcal{I}} - \mu_i^{\mathcal{I}}) \right],$$

and

$$\hat{\Sigma}_n^{\mathcal{O}} = \sum_i h_i^2(\xi_n^{\mathcal{I}}) \left[\Sigma_i^{\mathcal{O}\mathcal{O}} - \Sigma_i^{\mathcal{O}\mathcal{I}} \Sigma_i^{\mathcal{I}\mathcal{I}}^{-1} \Sigma_i^{\mathcal{I}\mathcal{O}} \right]. \quad (3)$$

The activation functions h_i in the above equations are defined as

$$h_i(\xi_n^x) = \frac{\pi_i \mathcal{N}(\xi_n^x | \mu_i^x, \Sigma_i^x)}{\sum_k^K \pi_k \mathcal{N}(\xi_n^x | \mu_k^x, \Sigma_k^x)}.$$

Eq. (3) is computed in real-time from the model parameters. The retrieved trajectory presents interesting smoothness properties (infinitely differentiable) and encapsulates variation and correlation information in the form of a probabilistic flow tube [20]. GMR has mostly been used in three manners:

- 1) as an autonomous system with $\xi = [x, \dot{x}]^\top$, by learning $\mathcal{P}(x, \dot{x})$ with a GMM, with x and \dot{x} representing position and velocity (either in task space or joint space), and by retrieving iteratively velocity commands during reproduction by estimating $\mathcal{P}(\dot{x}|x)$ with GMR [13], [21];
- 2) as an approach to couple the dynamical systems of a *dynamic movement primitive* (DMP) [22], where GMR provides a probabilistic formulation of DMP [1];
- 3) as time-indexed trajectories with $\xi = [t, x]^\top$, by learning $\mathcal{P}(t, x)$ with a GMM, and retrieving $\mathcal{P}(x|t)$ with GMR for each time step to reproduce a trajectory [12].

This last form will be used throughout this paper, but the approach remains valid for the other encoding schemes.

Alg. 1 summarizes the GPR approach in the case of trajectory models encoded as GMMs. The loop step in red (model fitting) highlights the specificity of the approach to encode each demonstration in a separated model. Although GPR can handle nonlinear interpolation problems, its generalization capability can degrade if the query points are too far from the demonstrations (it collapses to an average of the models). This will be tested later in the experiment.

B. Multi-streams approach

Another category of approaches relies on the encoding of trajectories in multiple candidate frames of reference [12]–[16]. The core strategy consists of observing the same movement from different landmarks or coordinate systems, and training a separated model in each frame. The learned models are then merged during reproduction to reproduce a generalized version of the movement. We will use the term *multi-streams* to refer to this category of approaches.

Alg. 2 shows an implementation of such approach. The loop step in red (model fitting) highlights the specificity of the approach to encode data for each candidate frame in a separated model. The core idea is to train GMMs with the same dataset projected in the different frames, generate new trajectories (and associated variations) with GMR in the different frames, project back the resulting GMR representations in task space, and multiply the resulting Gaussians at each iteration to reconstruct the movement, see [13] for details. GMR serves here the purpose of realigning the different models (trained separately) with respect to a common input. While accurate, the approach has some computational disadvantages regarding the use of multiple GMMs trained separately. This will be tested and discussed later in the experiment.

1. Task demonstrations (M demonstrations)

- Set P (number of candidate frames)
- for $n \leftarrow 1$ to N , with $N = \sum_m^M T_m$ (for each step)
 - Collect datapoint ξ_n
 - Collect task parameters $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$

end

2. Model fitting

for $j \leftarrow 1$ to P (for each candidate frame)

- Set K_j (nb of components for frame j)
- Fit a GMM to $\{\mathbf{A}_{n,j}^{-1}[\xi_n - \mathbf{b}_{n,j}]\}_{n=1}^N$ with EM $\{\pi_{i,j}, \mu_{i,j}, \Sigma_{i,j}\}_{i=1}^{K_j}$

end

3. Reproduction

- Set the input \mathcal{I} and output \mathcal{O} elements

for $n \leftarrow 1$ to T (for each reproduction step)

- Collect/select ξ_n^x and $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$

for $j \leftarrow 1$ to P (for each candidate frame)

- Use GMR in Eq. (3) to estimate $\mathcal{N}(\hat{\mu}_{n,j}^\circ, \hat{\Sigma}_{n,j}^\circ)$
- Project the resulting Gaussians with $\mathbf{A}_{n,j}, \mathbf{b}_{n,j}$

end

- Multiply the projected Gaussians to retrieve ξ_n°

end

Algorithm 2: Multi-streams approach.

1. Task demonstrations (same as in Alg. 2)

2. Model fitting

- Set K (number of components in the GMM)
- Fit a PGMM to ξ with EM algorithm of [2] $\{\pi_i, \tilde{\mu}_i, \tilde{\Sigma}_i\}_{i=1}^K$

3. Reproduction

- Set the input \mathcal{I} and output \mathcal{O} elements

for $n \leftarrow 1$ to T (for each reproduction step)

- Collect/select ξ_n^x and $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$ (concatenated in a vector \mathbf{Q}^d)

- Use Eq. (4) to estimate temporary GMM parameters $\{\mu_{n,i}\}_{i=1}^K$ modeling ξ_n^x and ξ_n° as

$$\xi_n^x, \xi_n^\circ \sim \sum_{i=1}^K \pi_i \mathcal{N}(\mu_{n,i}, \Sigma_i)$$

- Use Eq. (3) to retrieve ξ_n° through GMR as

$$\xi_n^\circ | \xi_n^x \sim \mathcal{N}(\hat{\mu}_n^\circ, \hat{\Sigma}_n^\circ)$$

end

Algorithm 3: Standard PGMM.

C. Parametric Gaussian mixture model (PGMM)

The last category of approaches is to encode all demonstrations in a single model. The *parametric hidden Markov model* (PHMM) is a representative approach in this category. It was originally introduced for recognition and prediction of gestures [2], and extended in robotics to movement generation [3], [17]. We will refer to a *parametric Gaussian mixture model* (PGMM) when the transition and initial state probabilities are not taken into account in the likelihood

estimation.²

The original model modulates each center through a linear relationship with the task parameters, concatenated in a vector \mathbf{Q}^d . Namely,

$$\boldsymbol{\mu}_{n,i} = \tilde{\mathbf{Z}}_i [\mathbf{Q}^d, \mathbf{1}]^\top, \quad (4)$$

with model parameter $\tilde{\mathbf{Z}}_i$, see [2] for details.

Alg. 3 gives a summary of the approach. One drawback of standard PGMM/PHMM model is that only the centers of the Gaussians are parameterizable. The covariances are estimated as constant matrices $\boldsymbol{\Sigma}_i$, with the standard EM procedure for GMM. We will show in the experiment that this can be problematic when encoding continuous movements, making EM converge to local optima unable to extract the underlying structures of the task.

III. PROPOSED APPROACH

1. **Task demonstrations** (same as in Alg. 2)
 2. **Model fitting**
 - Set K (number of components in the model)
 - Fit a model to $\boldsymbol{\xi}$ with the EM algorithm in Eq. (6)
$$\{\pi_i, \{\mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}_{j=1}^P\}_{i=1}^K$$
 3. **Reproduction**
 - Set the input \mathcal{I} and output \mathcal{O} elements for $n \leftarrow 1$ to T (for each reproduction step)
 - Collect/select $\boldsymbol{\xi}_n^x$ and $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$
 - Use Eq. (5) to estimate temporary GMM parameters $\{\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}\}_{i=1}^K$ modeling $\boldsymbol{\xi}_n^x$ and $\boldsymbol{\xi}_n^o$ as
$$\boldsymbol{\xi}_n^x, \boldsymbol{\xi}_n^o \sim \sum_{i=1}^K \pi_i \mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})$$
 - Use Eq. (3) to retrieve $\boldsymbol{\xi}_n^o$ through GMR as
$$\boldsymbol{\xi}_n^o | \boldsymbol{\xi}_n^x \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_n^o, \hat{\boldsymbol{\Sigma}}_n^o)$$
- end

Algorithm 4: Proposed approach.

The proposed approach shares connections with PGMM models, but modulates both the centers and covariances of the Gaussians, see also [1]. The learning problem is set as maximizing the log-likelihood of the observations in different candidate frames, under the constraint that these observations are generated by the same source. Namely, each frame j observes the same training datapoint $\boldsymbol{\xi}_n$ from its own perspective through local projection. Similarly to the estimation of the parameters of a standard GMM, deriving this constrained optimization problem results in an *expectation-maximization* (EM) algorithm that guarantee to improve the likelihood of the model at each iteration.

The parameters of the proposed model are $\{\pi_i, \mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}$, representing respectively the mixing coefficients, centers and covariances matrices for each frame j and mixture component i . At iteration n , the resulting

²Note here that the term *parametric* in PGMM/PHMM (referring to task parameters) can be ambiguous because a standard GMM can also be described as a parametric model (i.e., with model parameters). We still keep this terminology here to match with the original model description.

center $\boldsymbol{\mu}_{n,i}$ and covariance matrix $\boldsymbol{\Sigma}_{n,i}$ of each component i correspond to products of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) = \prod_{j=1}^P \mathcal{N}(\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}, \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top),$$

computed as

$$\begin{aligned} \boldsymbol{\Sigma}_{n,i} &= \left(\sum_{j=1}^P (\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top)^{-1} \right)^{-1}, \\ \boldsymbol{\mu}_{n,i} &= \boldsymbol{\Sigma}_{n,i} \sum_{j=1}^P (\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top)^{-1} (\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}). \end{aligned} \quad (5)$$

The parameters of the model $\{\pi_i, \mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}$ are iteratively estimated with the following EM procedure.³

E-step:

$$\gamma_{n,i} = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})}{\sum_{k=1}^K \pi_k \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,k}, \boldsymbol{\Sigma}_{n,k})}.$$

M-step:

$$\begin{aligned} \pi_i &= \frac{\sum_{n=1}^N \gamma_{n,i}}{N}, \quad \mathbf{Z}_{i,j}^\mu = \frac{\sum_{n=1}^N \gamma_{n,i} \mathbf{A}_{n,j}^{-1} [\boldsymbol{\xi}_n - \mathbf{b}_{n,j}]}{\sum_{n=1}^N \gamma_{n,i}}, \\ \mathbf{Z}_{i,j}^\Sigma &= \frac{\sum_{n=1}^N \gamma_{n,i} \mathbf{A}_{n,j}^{-1} [\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}] [\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}]^\top \mathbf{A}_{n,j}^{-\top}}{\sum_{n=1}^N \gamma_{n,i}}, \\ &\text{with } \tilde{\boldsymbol{\mu}}_{n,i,j} = \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}. \end{aligned} \quad (6)$$

Note that, in contrast to the multi-streams approach, a single EM process is used to iteratively refine the model parameters, where the *E-step* considers the influence of the different frames for clustering the data in a common frame, see Eq. (5) for the computation of $\boldsymbol{\mu}_{n,i}$ and $\boldsymbol{\Sigma}_{n,i}$ in the *E-step*. The model parameters are initialized with a *k-means* procedure modified similarly as for the EM algorithm.

Alg. 4 describes the overall process.⁴ Model selection is compatible with the techniques employed in standard GMM (Bayesian information criterion, Dirichlet process, etc.). The *Matlab* and C++ source codes of the proposed model are available on <http://programming-by-demonstration.org>.

In the next section, the advantage of the proposed approach is demonstrated with an experiment of rolling out a pizza dough.

³The proof is omitted here due to space restriction but can be retrieved by differentiating (with respect to $\mathbf{Z}_{i,j}^\mu$ and $\mathbf{Z}_{i,j}^\Sigma$) the log-likelihood function from Eq. 5. The result shares similarities with the estimates for $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ in EM applied to GMM, with the difference that in the case of $\mathbf{Z}_{i,j}^\mu$ and $\mathbf{Z}_{i,j}^\Sigma$, the data are locally projected through the inverse of the local transformation defined by $\mathbf{A}_{n,j}$ and $\mathbf{b}_{n,j}$. Indeed, the role of EM in a standard GMM is to estimate constant model parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, while in the proposed task-parameterized model, EM estimates $\mathbf{Z}_{i,j}^\mu$ and $\mathbf{Z}_{i,j}^\Sigma$ by incrementally refining the local importance of the candidate frames for the overall task.

⁴In step 3), the temporary GMM parameters do not need to be re-estimated if $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j=1}^P$ do not change over time. Namely, if the candidate frames do not move during the movement, $\{\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}\}_{i=1}^K$ can be evaluated only once at the beginning of the movement to reduce computation.

IV. EXPERIMENTAL SETUP

We consider the movement of rolling out a pizza dough demonstrated by kinesthetic teaching, see Fig. 1. Such task is used as an example of movement modulated locally by the position and orientation of an object (the dough). The user simulates a movement that would increase the dough surface to make the pizza circular.⁵ The task is to move the rolling pin from an initial pose in the robot workspace to the center of the dough, then move back and forth the rolling pin in a direction following the minor axis of the dough shape, and finally lift the rolling pin to let the vision system track the dough again.

The experiment is implemented in a *Barrett WAM* torque-controlled 7 DOFs manipulator with a rolling pin mounted at the end-effector. The robot is gravity-compensated during demonstrations and reproductions, with a wrench command to keep the orientation of the end-effector perpendicular to the worktop.

The resulting trajectories are tracked by a virtual spring-damper system, whose attractor and stiffness matrix are respectively defined by the center of the retrieved Gaussians in Eq. (2) and associated precision matrices (inverse of covariances). Forces acting as gravity compensation are superposed to the tracking forces, resulting in a safe controller for the user, who can exploit the redundancy of the robot and the redundancy of the task during demonstration and reproduction, see [23] for details of the controller. For example, when sharing the workspace with the robot, the user can during reproduction change the position of the robot's elbow to have more space.

Two candidate frames are considered ($P = 2$): the fixed robot frame (useful for the first part of the motion) and a frame defined by the dough location and shape extracted by a camera (based on color information). By assuming that the duration of the movement is not modulated by the task parameters, we have $\mathbf{b}_{n,1} = \mathbf{0}$, $\mathbf{A}_{n,1} = \mathbf{I}$ and $\mathbf{b}_{n,2} = [0, \mathbf{p}_n]^\top$, $\mathbf{A}_{n,2} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_n \end{bmatrix}$, with \mathbf{p}_n and \mathbf{R}_n the position and orientation of the dough at time step n ($\mathbf{A}_{n,j} \in \mathbb{R}^{4 \times 4}$, \mathbf{I} is the identity matrix, $\mathbf{0}$ are zeros vectors of appropriate sizes and \mathbf{R}_n is a direction cosine matrix).

The motions are described by three variables ($D = 3$) representing the position of the rolling pin, with an orientation actively maintained to follow the movement direction. Four demonstrations with different positions and orientations of the dough are provided to the robot by kinesthetic teaching. Models with $K = 6$ components are considered in the experiment (selected empirically).

V. EXPERIMENTAL RESULTS

We compared the proposed approach (Alg. 4) with the three models described in Algs 1, 2 and 3.

⁵In a real pizza making process, other elements would come into play such as friction, softness of the dough, flour on the worktop, etc. In our experiment, the dough is pre-shaped in an elliptical form and no force is applied to it.

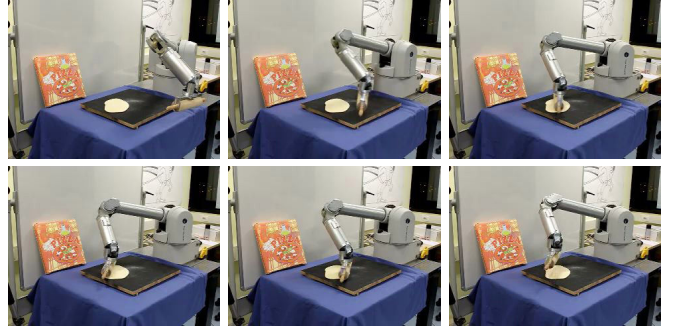


Fig. 2. Snapshots of a typical reproduction result performed in a new situation that has not been demonstrated to the robot.

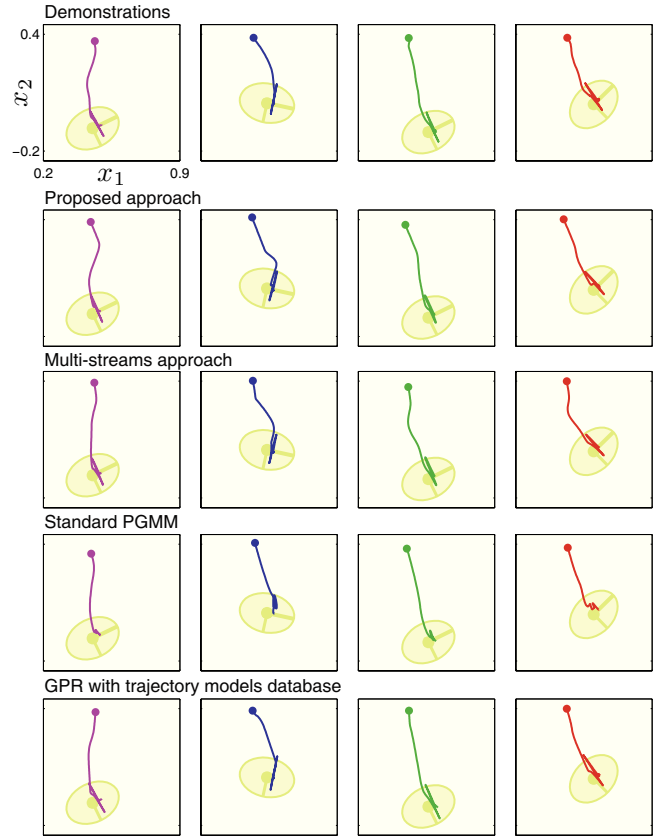


Fig. 3. Demonstrations and reproductions in the same situations (top-view of the worktop). Initial positions are plotted with points, and doughs are plotted with ellipses.

Fig. 2 presents snapshots of the reproduction results in new situations,⁶ which can also be seen in the video accompanying the paper. Fig. 3 presents the demonstrations and reproductions results. For the same situations, all models produce smooth trajectories passing over the dough. Namely, the movement starts with a path in the robot frame that progressively moves towards the dough, and when reaching it, moves the rolling pin back and forth in a direction parallel to the minor axis of the dough.

Fig. 4 presents interpolation and extrapolation results. All

⁶The most representative reproduction attempts are depicted here, since the results can randomly differ with the k-means initialization.

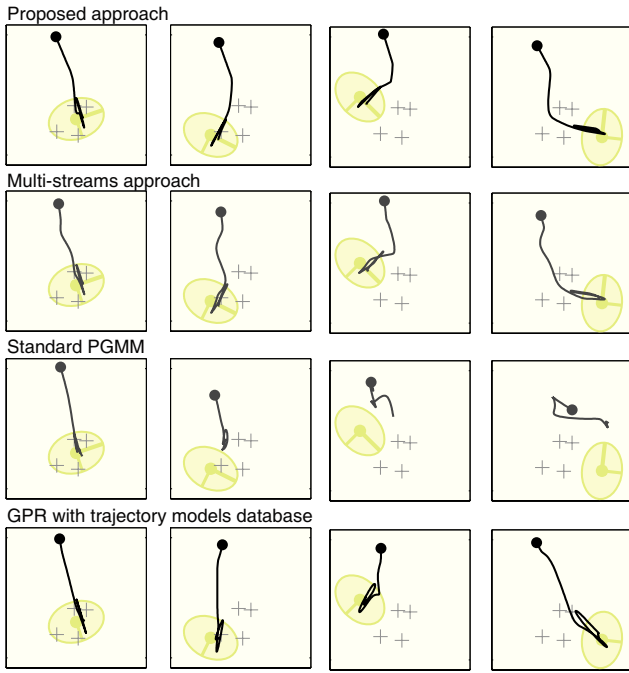


Fig. 4. Reproductions in new situations (new task parameters). The ‘+’ signs depict the dough positions used to train the model.

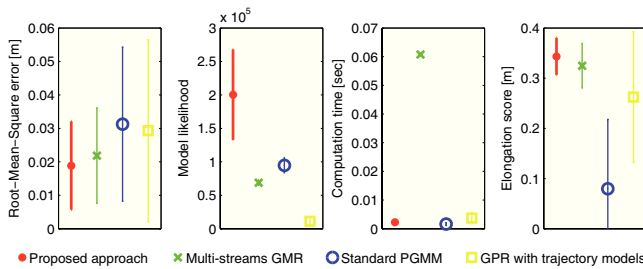


Fig. 5. Comparisons by considering the *root-mean-square* error between the demonstrated and reproduced trajectories (for the same situations as the demonstrations), the likelihood of the model retrieved after regression, the computation time for each reproduction step and a simulated surface elongation score for the reproductions with new task parameters. This score estimates the extrapolation capability by using the datapoints on the surface of the dough to measure the cumulated distance along the minor axis minus the cumulated distance along the major axis. This score reflects the requirement of spreading the dough in a desired direction. The error bars show standard deviations over the sets of datapoints and trials.

approaches show good interpolation capability (first graph). The proposed approach and the multi-streams approach stand out in terms of extrapolation (last three graphs), both being capable of adapting the movement outside the region covered by the demonstrations (depicted by ‘+’ signs).

Fig. 5 provides quantitative comparisons with four criteria, highlighting the overall advantages of the proposed approach. The first two graphs show that for the same situations, the proposed model can regenerate movements closely matching the demonstrations, with the best likelihood fit over the other approaches. The third graph shows the computation time improvement over multi-streams approaches that require more Gaussians products operations. The last graph shows that the proposed approach has on average the

best extrapolation capability, with more consistent results than the other approaches. These results concord with our expectation that task-parameterized models extracting the local structures of the task can improve generalization and speed up the retrieval process.

VI. DISCUSSION

The proposed model circumvents the shortcoming of standard PGMM/PHMM that only the centers are influenced by the task parameters. The results show that adapting covariance matrices to external task parameters is crucial when modeling continuous movements. The covariances represent the principal direction of the movement, the local synergies among the variables, and the variations between demonstrations (e.g., to determine which parts of the motion need to be reproduced precisely and which parts can be more loosely reproduced).

A common practice to prevent this problem in standard PGMM/PHMM is to increase the number of Gaussians, which has the effect of reducing the relevance of the covariance information in the modeling of the movement. It however degrades the parsimony of the model and comes at the expense of losing information about the local shape of the movement. By increasing the number of Gaussians, each covariance will provide only narrow and unusable information instead of the important local neighborhood characteristics, synergies and shapes of the movement.

Other approaches were previously proposed for adapting covariances to task parameters. Inspired by PHMM, Brand and Hertzmann proposed in [5] a parameterization modulating both centers and covariances, used to synthesize novel motion data in computer graphics applications. The perspective was however different, by identifying common choreographic elements (and the different styles in which each element is performed), with a small number of stylistic variables describing the variations in the dataset.

By using PHMM, Krueger *et al* [3] proposed to tackle the problem of parameterizing both the centers and covariances by encoding each demonstration in a separated model. After proper realignment, a resulting model is estimated as a weighted sum of the centers and covariances of the different models. For our application, the drawback of this approach is that it only considers scalar scaling transformations of the covariances instead of linear transformations. It does not allow the system, for example, to re-orient a normal distribution with respect to landmarks in the robot’s environment (e.g. to approach an object from a desired angle to facilitate its prehension).⁷

The approach that we propose is well suited for problems in which the task parameters can be represented in the form of coordinate systems. However, for task parameters that cannot be represented in such way, the standard PGMM/PHMM

⁷For example, if a Gaussian in a first demonstration has a main axis oriented at 0 degree, and if the corresponding Gaussian in a second demonstration has a main axis oriented at 90 degree, the weighted sum (with equal weights) will result in a circular covariance, instead of a Gaussian with main axis oriented at 45 degrees.

remains a better modeling choice because it does not require the separation of the task parameters into an offset value (position \mathbf{b} of the observer) and basis vectors forming a matrix \mathbf{A} . The standard PGMM/PHMM can also tackle the reverse problem of estimating the task parameters from observations in a more straightforward manner. Indeed, in the proposed approach, the parameters representing the coordinate systems are combined in such a way that the estimation of the task parameters from observed data becomes more complex. This issue will require further investigation (\mathbf{b} can easily be estimated in a closed form, but \mathbf{A} requires an iterative procedure).

The proposed model and the multi-streams approach share the common perspective of probabilistically representing the local importance of different coordinate systems. While this is done as separated batch learning processes in multi-streams GMR (projection, EM learning, back-projection and recombination with Gaussians products), the proposed approach permits us to formulate the different steps directly in the EM procedure, resulting in a *mixture of Gaussian products* organizing the different Gaussian components in a principled manner, and speeding up the retrieval process.

VII. CONCLUSION AND FUTURE WORK

By building up on previous work in PGMM/PHMM, multi-streams GMR and GPR with trajectory models, we presented a statistical model and an associated EM algorithm to encode and reproduce task-parameterized movements. We showed that the proposed approach has several advantages. It ameliorates PGMM/PHMM adaptability by parameterizing not only the centers of the Gaussians but also the covariances. It is faster and more consistent than multi-streams GMR by relying on a single EM process rather than learning separated models. It shows extrapolation capability that go beyond approaches relying on interpolation principles such as GPR. This generalization capability is crucial for scaling up imitation learning challenges to skills of increasing complexity, for which it can be difficult to guarantee that the demonstrations cover a sufficient range of situations.

Our future work include several perspectives. The frames of reference considered in the experiment were defined by Cartesian coordinate systems. The approach, however, does not require this assumption to be made, and the candidate frames do not need to have the same dimensions. Non-squared \mathbf{A} matrices could also be considered, which could for example be exploited to learn skills requiring joint space and task space coordinates. Also, the current paper focused on acquiring trajectories. We will study in future work if such models could similarly learn reflex behaviors that are shaped by task parameters (including force signals), which would be relevant for robots faced with continuous sources of perturbation.

REFERENCES

- [1] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Osaka, Japan, 2012, pp. 323–329.
- [2] A. D. Wilson and A. F. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.
- [3] V. Krueger, D. L. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning actions from observations: Primitive-based modeling and grammar," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 30–43, 2010.
- [4] 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, June 2011.
- [5] M. Brand and A. Hertzmann, "Style machines," in *Proc. ACM Intl Conf. on Computer graphics and Interactive Techniques (SIGGRAPH)*, New Orleans, Louisiana, USA, July 2000, pp. 183–192.
- [6] 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, pp. 1327–1339, 2012.
- [7] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, April 2012.
- [8] K. Kronander, M. S. Khansari-Zadeh, and A. Billard, "Learning to control planar hitting motions in a minigolf-like task," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 710–717.
- [9] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.
- [10] C. L. Campbell, R. Peters, R. E. II Bodenheimer, W. J. Bluethmann, E. Huber, and R. O. Ambrose, "Superpositioning of behaviors learned through teleoperation," *IEEE Trans. on Robotics*, vol. 22, no. 1, pp. 79–91, 2006.
- [11] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *Intl Journal of Robotic Research*, vol. 23, no. 4-5, pp. 363–377, 2004.
- [12] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.
- [13] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, June 2010.
- [14] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Action, state and effect metrics for robot imitation," in *IEEE Intl Symposium on Robot and Human Interactive Communication (Ro-Man)*, Hatfield, UK, September 2006, pp. 232–237.
- [15] M. Mühlig, M. Gienger, and J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, no. 2, pp. 97–114, 2012.
- [16] S. Dong and B. Williams, "Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes," *Intl Journal of Social Robotics*, vol. 4, no. 4, pp. 357–368, 2012.
- [17] T. Yamazaki, N. Niwase, J. Yamagishi, and T. Kobayashi, "Human walking motion synthesis based on multiple regression hidden semi-Markov model," in *Proc. Intl Conf. on Cyberworlds*, 2005, pp. 445–452.
- [18] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura, "Learning, generation, and recognition of motions by reference-point-dependent probabilistic models," *Advanced Robotics*, vol. 25, no. 5, 2011.
- [19] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.
- [20] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2, pp. 115–131, 2011.
- [21] S. M. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with Gaussian mixture models," *IEEE Trans. on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [22] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [23] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010, pp. 249–254.