# A Task-Parameterized Probabilistic Model
# with Minimal Intervention Control

Sylvain Calinon[1,2], Danilo Bruno[1] and Darwin G. Caldwell[1]

*Abstract*— We present a task-parameterized probabilistic model encoding movements in the form of virtual spring-damper systems acting in multiple frames of reference. Each candidate coordinate system observes a set of demonstrations from its own perspective, by extracting an attractor path whose variations depend on the relevance of the frame at each step of the task. This information is exploited to generate new attractor paths in new situations (new position and orientation of the frames), with the predicted covariances used to estimate the varying stiffness and damping of the spring-damper systems, resulting in a minimal intervention control strategy. The approach is tested with a 7-DOFs Barrett WAM manipulator whose movement and impedance behavior need to be modulated in regard to the position and orientation of two external objects varying during demonstration and reproduction.

## I. INTRODUCTION

Two important challenges in learning by imitation are to generalize an observed skill to new situations and to generate movements that are natural, efficient and safe for the surrounding users [1], [2]. We present an approach combining a statistical mixture model with a dynamical system to encode movements, exploiting the predicted task variations and couplings to regulate the impedance of virtual spring-damper systems acting in several frames of reference. The model shares links with optimal feedback control strategies in which deviations from an average trajectory are corrected only when they interfere with task performance, such as in the *minimal intervention principle* [3], [4].

A widespread approach for movement primitives learning in robotics is to combine dynamical systems in sequence and in parallel such as in the *dynamic movement primitives* (DMP) model [5]. In DMP, a forcing term for each dimension of the movement modulates a spring-damper system centered on a target, where the different forcing terms are synchronized by another dynamical system acting as a decay term. After converting an observed movement into forcing term trajectories, and after setting a set of basis functions sequentially activated through the decay term, the learning task consists of individually approximating the forcing term profiles.

Although the DMP formulation does not restrict the way in which forcing terms are learned [5], most work relied in practice on *locally weighted regression* to train the model parameters with predefined basis functions (e.g., equal bandwidth and equal interval spacing). The forcing terms in
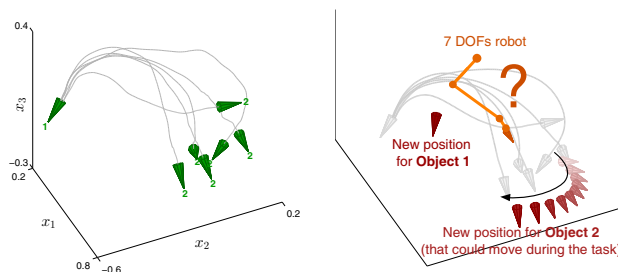


Fig. 1. Illustration of the challenges addressed in the paper. The first challenge consists of generalizing the movement to new situations (new positions and orientations of the cones, plausibly moving during the execution of the task). The second challenge consists of exploiting the redundancy of the task to regulate the stiffness and damping gains of a virtual spring-damper system actuating the robot. This is achieved by using the predicted variability along the movement with a minimal intervention control strategy based on linear quadratic regulators.

such dynamical systems can however be learned by other learning strategies. Several regression techniques have been developed for multidimensional inputs and unidimensional output problems [6]. Here, representing movements as a set of univariate outputs can be restrictive if we want to exploit different sources of local correlations among inputs, among outputs, and in-between input-output variables (e.g., to discover and re-use sensorimotor patterns or synergies in the output variables). This paper explores the use of multi-output regression in the context of proportional-derivative control systems, by exploiting the predicted task regularities modeled from consecutive demonstrations of a task.

We presented in [7] a probabilistic formulation of dynamic movement primitives, by encoding the joint evolution of the input (decay term) and the output (forcing terms) within a multivariate *Gaussian mixture model* (GMM). *Gaussian mixture regression* (GMR) [8] could then be used to retrieve at each iteration the forcing terms corresponding to the current input (either time-dependent or time-invariant). We showed in [9] that such mixture model formulation could be exploited to adapt the centers and covariances of a GMM to the location and orientation of multiple objects, virtual landmarks or coordinate systems. The model allows the automatic transitions between different coordinate systems that are potentially relevant for the task. This *task-parameterized GMM* probabilistically encodes the changing relevance of candidate frames throughout the task. The combination of the two approaches [7], [9] extends the generalization capability of dynamic movement primitives, offering the possibility to adapt movements with respect to multiple viapoints (which

can be in the middle of the movement), with local position, orientation and shape modulation.

The contributions of the current paper are threefold: 1) An improved formulation of the task-parameterized mixture model [9], making its computation more efficient; 2) A method to include end-effector orientation data in the model; and 3) A minimal intervention control strategy using the predicted covariances to reduce the control commands, by adapting the tracking gains in regard to the current relevance of the reference signal for the completion of the task.

Fig. 1 illustrates the tackled challenges. The proposed approach is detailed in Section II. The experiment is described in Section III. Conclusion and future work are presented in Section IV.

## II. PROPOSED APPROACH

The complete procedure consists of a *demonstration* phase, a *learning* phase and a *reproduction* phase. In the demonstration phase, a set of movements is recorded as position and orientation of the robot end-effector (output) with associated time stamp (input). The multiple demonstrations are aligned in time with *dynamic time warping*. The position and orientation of a set of candidate frames (related to objects in the robot workspace) is also collected. In our application, the orientation of the robot end-effector is represented with pan-tilt angles, while the orientations of objects are represented as rotation matrices whose columns form the orthogonal basis of a frame of reference. The recorded movements are projected in these frames (observation of the same movement from multiple viewpoints). The input and output variables are concatenated for each frame, forming a 3rd order tensor dataset. Here, time is used as input variable, but a decay term, the robot state or other external object position variables can similarly be employed [7].

In the learning phase, a task-parameterized mixture model is fit to the tensor training set by following an *expectation-maximization* (EM) procedure (subsection II-A). The training set can then be discarded. In the reproduction phase, for a situation involving new position and orientation of objects, the learned model is first used to estimate a temporary Gaussian mixture model (GMM), that is automatically updated if there is a change in position/orientation of the objects. Depending on the application, this temporary GMM either needs to be updated at each time step (e.g., adapting movements to moving targets), or for each new reproduction attempt (planning approach). *Gaussian mixture regression* (GMR) is then used to retrieve statistical information about the current reference to track, corresponding to the equilibrium point of a virtual spring-damper system (subsection II-B). This information is finally used by a linear quadratic regulator to form a minimal intervention controller (subsection II-C).

The source codes of the proposed approach are available at http://programming-by-demonstration.org/ICRA2014/.

### A. Task-parameterized model

The task parameters are represented as $P$ coordinate systems, defined at time step $n$ by $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^{P}$, representing respectively the origin of the observer and a set of basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, ...\}$ forming a transformation matrix $\boldsymbol{A} = [\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots]$.

A movement $\boldsymbol{\xi} \in \mathbb{R}^{D \times N}$ is observed from these different viewpoints, forming a third order tensor dataset $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D \times N \times P}$, composed of $P$ trajectory samples $\boldsymbol{X}^{(j)} \in \mathbb{R}^{D \times N}$ observed in $P$ candidate frames, corresponding to matrices composed of $D$-dimensional observations at $N$ time steps. In our application, $D = 6$, corresponding to the aggregation of time variable (1 dimension), Cartesian position attractors (3 dimensions), and pan-tilt orientation attractors (2 dimensions).

The parameters of a model with $K$ components are defined by $\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^{P}\}_{i=1}^{K}$, where $\pi_i$ are the mixing coefficients. $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are the mode-$j$ center and covariance matrix of the $i$-th Gaussian component.

Learning of the parameters is achieved with the constrained problem of maximizing the log-likelihood under the constraints that the data in the different frames are generated from the same source, resulting in an EM process to iteratively update the model parameters until convergence.

E-step:

$$\gamma_{n,i} = \frac{\pi_i \prod_{j=1}^{P} \mathcal{N}\big(\boldsymbol{X}_n^{(j)} | \, \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\big)}{\sum_{k=1}^{K} \pi_k \prod_{j=1}^{P} \mathcal{N}\big(\boldsymbol{X}_n^{(j)} | \, \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)}\big)}.$$

M-step:

$$\pi_i = \frac{\sum_{n=1}^{N} \gamma_{n,i}}{N} \, , \quad \boldsymbol{\mu}_i^{(j)} = \frac{\sum_{n=1}^{N} \gamma_{n,i} \, \boldsymbol{X}_n^{(j)}}{\sum_{n=1}^{N} \gamma_{n,i}},$$

$$\boldsymbol{\Sigma}_i^{(j)} = \frac{\sum_{n=1}^{N} \gamma_{n,i} \, (\boldsymbol{X}_n^{(j)} - \boldsymbol{\mu}_i^{(j)})(\boldsymbol{X}_n^{(j)} - \boldsymbol{\mu}_i^{(j)})^{\top}}{\sum_{n=1}^{N} \gamma_{n,i}}. \quad (1)$$

The model parameters are initialized with a *k-means* procedure. Model selection is compatible with the techniques employed in standard GMM (*Bayesian information criterion* [10], *Dirichlet process* [11], etc.). In a standard GMM, the role of EM is to estimate constant Gaussian parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. Here, EM is used to estimate task-parameterized model parameters $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ by incrementally modeling the local importance of the candidate frames. In the proposed experiment, the overall learning process typically takes 1 to 4 *sec*. The reproduction is much faster and can be computed online (below 1 *msec*).

The above model is equivalent to the model presented in [9], but is computationally more efficient. The E-step as formulated above involves a product of probabilities (multiplication of scalars), while the E-step in [9] first computes the intersection of Gaussians (products of Gaussians) before evaluating the likelihoods. With the above formulation, there is no need of explicitly providing the parameters $\boldsymbol{A}_{n,j}$ and $\boldsymbol{b}_{n,j}$ in the learning phase (this information is already contained in the third order tensor dataset $\boldsymbol{\mathcal{X}}$, with the demonstrations observed from different perspectives).

The learned model can then be used to reproduce movements in other situations (for new positions and orientations of candidate frames). The model first retrieves at each time step $n$ a GMM by computing a product of linearly transformed Gaussians

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

$$\Leftrightarrow \boldsymbol{\Sigma}_{n,i} = \Big(\sum_{j=1}^{P}(\boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^{\top})^{-1}\Big)^{-1}, \qquad (2)$$

$$\boldsymbol{\mu}_{n,i} = \boldsymbol{\Sigma}_{n,i}\sum_{j=1}^{P}(\boldsymbol{A}_{n,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{n,j}^{\top})^{-1}(\boldsymbol{A}_{n,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{n,j}).$$

*B. Gaussian mixture regression*

Gaussian mixture regression (GMR) is used to generate the movements [8], [12]. GMR can be viewed as a trade-off between a global and local approach in the sense that the placement and spread of the basis functions are learned, together with their response, as a soft partitioning problem through *expectation-maximization* (EM),[1] while the prediction is a weighted superposition of locally linear systems. The prediction provides information about the local variations allowed by the task and about the correlations among the different output terms, thus allowing the extraction of local coordination patterns. It allows the robot to generate natural movements with a co-variability following the essential characteristics of the task, which can be exploited for stochastic exploration [13] or for natural interaction in human-robot collaboration [14].

In GMR, the underlying representation as mixture of Gaussians is independent from the training algorithm used to estimate the model parameters. Various methods can be employed depending on the application requirements, such as *expectation-maximization* (EM) [15], *online EM* [16] or *spectral learning* [17]. If the application requires the encoding of high-dimension data from few observations, subspace learning techniques such as *mixtures of factor analyzers* (MFA) [18] can be used to locally reduce the dimensionality without modifying the representation (full covariances can be reconstructed from the MFA parameters). Common synergy information can be shared among the Gaussians with *parsimonious GMM* [19] (e.g., to re-use and adapt previously discovered coordination patterns).

The superscripts $\mathcal{I}$ and $\mathcal{O}$ will be further used to describe the dimensions that span for input and output variables (for vectors and matrices). For the movement data, at iteration $n$, $\boldsymbol{\xi}_n^{\mathcal{I}}$ and $\boldsymbol{\xi}_n^{\mathcal{O}}$ represent the input and output variables, while $\boldsymbol{\xi}_n$ represents the same datapoint in a concatenated form. For trajectory encoding in task space, $\mathcal{I}$ corresponds to the time input dimension, and $\mathcal{O}$ corresponds to the output dimensions describing a path in task space (position and orientation).

---

[1]Competition/collaboration arise due to the weighting term $\gamma_{n,i}$ in Eq. (1) summing over the influence of the other Gaussian components.

With this notation, a block decomposition of the datapoints $\boldsymbol{\xi}_n$, vectors $\boldsymbol{\mu}_{n,i}$ and matrices $\boldsymbol{\Sigma}_{n,i}$ can be written as

$$\boldsymbol{\xi}_n = \begin{bmatrix} \boldsymbol{\xi}_n^{\mathcal{I}} \\ \boldsymbol{\xi}_n^{\mathcal{O}} \end{bmatrix}, \quad \boldsymbol{\mu}_{n,i} = \begin{bmatrix} \boldsymbol{\mu}_{n,i}^{\mathcal{I}} \\ \boldsymbol{\mu}_{n,i}^{\mathcal{O}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_{n,i} = \begin{bmatrix} \boldsymbol{\Sigma}_{n,i}^{\mathcal{I}} & \boldsymbol{\Sigma}_{n,i}^{\mathcal{IO}} \\ \boldsymbol{\Sigma}_{n,i}^{\mathcal{OI}} & \boldsymbol{\Sigma}_{n,i}^{\mathcal{O}} \end{bmatrix}.$$

By using the temporary GMM parameters computed in Eq. (2), GMR relies on the joint distribution $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{I}}, \boldsymbol{\xi}_n^{\mathcal{O}})$ to estimate $\mathcal{P}(\boldsymbol{\xi}_n^{\mathcal{O}}|\boldsymbol{\xi}_n^{\mathcal{I}})$. At each reproduction step $n$, this conditional probability is estimated as an output distribution $\mathcal{N}(\hat{\boldsymbol{\xi}}_n^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_n^{\mathcal{O}})$, that is also Gaussian, with parameters

$$\hat{\boldsymbol{\xi}}_n^{\mathcal{O}} = \sum_i h_{n,i}(\boldsymbol{\xi}_n^{\mathcal{I}})\Big[\boldsymbol{\mu}_{n,i}^{\mathcal{O}} + \boldsymbol{\Sigma}_{n,i}^{\mathcal{OI}}\boldsymbol{\Sigma}_{n,i}^{\mathcal{I}\,-1}(\boldsymbol{\xi}_n^{\mathcal{I}} - \boldsymbol{\mu}_{n,i}^{\mathcal{I}})\Big],$$

$$\hat{\boldsymbol{\Sigma}}_n^{\mathcal{O}} = \sum_i h_{n,i}^2(\boldsymbol{\xi}_n^{\mathcal{I}})\Big[\boldsymbol{\Sigma}_{n,i}^{\mathcal{O}} - \boldsymbol{\Sigma}_{n,i}^{\mathcal{OI}}\boldsymbol{\Sigma}_{n,i}^{\mathcal{I}\,-1}\boldsymbol{\Sigma}_{n,i}^{\mathcal{IO}}\Big], \qquad (3)$$

and activation functions $h_{n,i}$ defined as

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

The estimated output in Eq. (3) encapsulates variation and correlation information in the form of a probabilistic flow tube [20], continuously differentiable in time.

*C. Minimal intervention controller*

With the above method, a reference trajectory is estimated as a full distribution $\mathcal{N}(\hat{\boldsymbol{\xi}}_n^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_n^{\mathcal{O}})$ varying at each time step $n$ given by Eq. (3). Similarly as the solution proposed by Medina *et al.* in the context of risk-sensitive control for haptic assistance [21], the predicted variability can be exploited to form a minimal intervention controller (in task space or in joint space). The procedure will first be described for a controller in task space, where an acceleration command

$$\boldsymbol{u}_n = \hat{\boldsymbol{K}}_n^{\mathcal{P}}(\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n) - \hat{\boldsymbol{K}}_n^{\mathcal{V}}\dot{\boldsymbol{x}}_n \qquad (4)$$

is used to control the robot, with $\hat{\boldsymbol{x}}_n$ estimated by GMR in Eq. (3).

$\hat{\boldsymbol{K}}_n^{\mathcal{P}}$ and $\hat{\boldsymbol{K}}_n^{\mathcal{V}}$ are full stiffness and damping matrices estimated by a *linear quadratic regulator* (LQR) with time-varying weights. For a finite horizon LQR, this is achieved by minimizing the cost function

$$c^{(1)} = \sum_{n=1}^{T}(\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n)^{\top}\boldsymbol{Q}_n(\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n) + \boldsymbol{u}_n^{\top}\boldsymbol{R}\,\boldsymbol{u}_n, \qquad (5)$$

subject to the constraints of a double integrator system. The solution can be computed by backward integration of a *Riccati ordinary differential equation* with varying full weighting matrix $\boldsymbol{Q}_n = \hat{\boldsymbol{\Sigma}}_n^{\boldsymbol{x}\,-1}$ estimated with Eq. (3). It provides a time-varying feedback control law in the form of Eq. (4) with full stiffness and damping matrices $\hat{\boldsymbol{K}}_n^{\mathcal{P}}$ and $\hat{\boldsymbol{K}}_n^{\mathcal{V}}$.

To solve the above minimization problem, a boundary condition needs to be set on the final feedback term, which is set to zero in our experiment. Namely, we assume that the task is fulfilled at the end of the movement and that the robot can become compliant again.

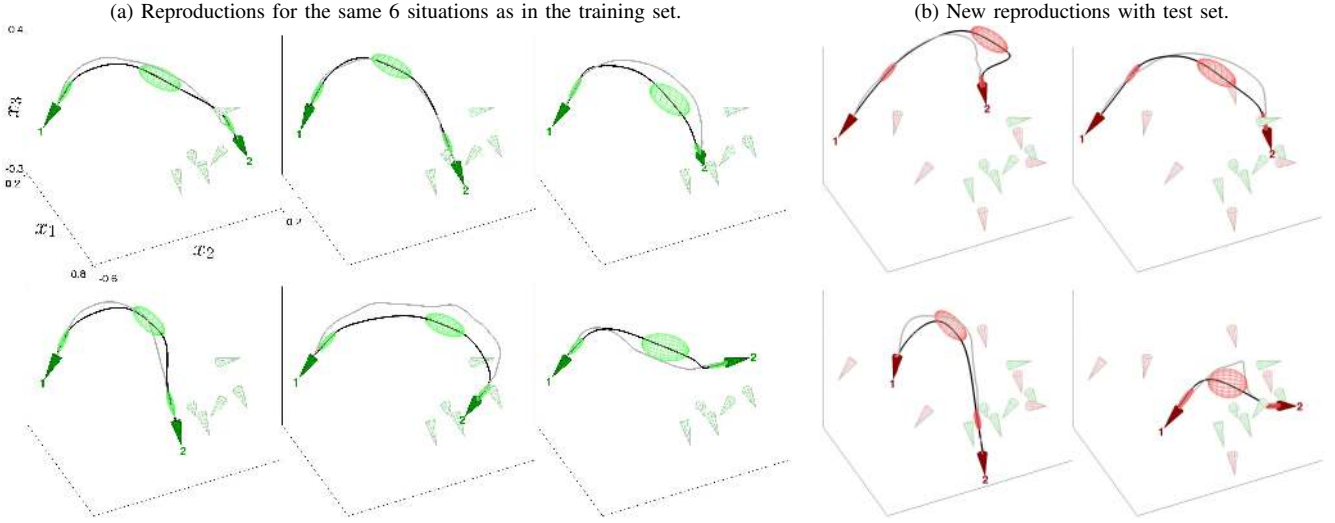(a) Reproductions for the same 6 situations as in the training set.    (b) New reproductions with test set.

Fig. 2.  (a) Reconstruction results from the model parameters. The dark cones are the candidate frames (the frames that were part of the training set are shown in lighter color). The ellipsoids represent the temporary GMM with Gaussians $\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})$ computed in Eq. (2). The light and dark lines represent respectively demonstrations and reproductions. (b) Reproductions of movements for the 4 new situations that were not part of the training set.

At iteration $n$, the backward recursion to minimize Eq. (5) requires the estimation of $\hat{\boldsymbol{\Sigma}}_t^{\boldsymbol{x}}$ for $t \in \{n, n+1, ..., T\}$. If the position and orientation of external objects are changing over time, the predicted trajectory and associated covariances need to be recomputed during the movement, providing a new recursion path for the Riccati equation.

In some situations, it might be computationally expensive to recompute at each iteration $n$ a prediction on the remaining movement. An approximation can in this case be locally computed by considering an infinite horizon LQR formulation to estimate a feedback term at iteration $n$ by considering only the current estimate $\hat{\boldsymbol{\Sigma}}_n^{\boldsymbol{x}}$. This corresponds to the estimation of a feedback controller that does not know in advance whether the precision at which it should track a target will vary. The corresponding cost function at iteration $n$ corresponds to

$$c_n^{(2)} = \sum_{t=n}^{\infty} (\hat{\boldsymbol{x}}_n - \boldsymbol{x}_t)^\top \boldsymbol{Q}_n (\hat{\boldsymbol{x}}_n - \boldsymbol{x}_t) + \boldsymbol{u}_t^\top \boldsymbol{R}\, \boldsymbol{u}_t, \quad \forall n \in \{1,...,T\} \tag{6}$$

which can be solved iteratively through the *algebraic Riccati equation*, providing a feedback controller in the form of Eq. (4) with full stiffness and damping matrices $\hat{\boldsymbol{K}}_n^{\mathcal{P}}$ and $\hat{\boldsymbol{K}}_n^{\mathcal{V}}$.

In [22], a similar feedback controller was heuristically estimated by computing a stiffness matrix at each iteration $n$ as proportional to the estimated precision matrix $\boldsymbol{Q}_n = \hat{\boldsymbol{\Sigma}}_n^{\boldsymbol{x}-1}$ of the current point to be tracked. The LQR approaches minimizing Eqs (5) and (6) result in a controller sharing similar characteristics, but it provides a formal way of adapting the impedance parameters.

The approach described above can similarly be applied in configuration space by computing a reference trajectory in joint space with inverse kinematics, and locally projecting the covariance information through the Jacobian at the current configuration. This variant will be used in the experiment to reduce the control commands at the joints level.

## III. EXPERIMENT

A torque-controlled Barrett WAM 7 DOFs manipulator is used in the experiment. The aim of the task is to move a conic peg from one place to another, by moving the peg from one extruded cone to an other extruded cone.[2] The task is recorded 10 times with different positions and orientations of the holes, by physically moving the robot through the task while actively compensating for the gravity (kinesthetic teaching process). While demonstrating the task, the robot records the position and orientation of the peg attached to its end-effector. The position and orientation of the extruded cones are pre-recorded by bringing the robot to these two candidate frames prior to the demonstration of the movement. For the first 8 recordings, only the second cone is moved from one demonstration to the other. For the remaining 2 recordings, both cones are moved to new poses.

The first 6 recordings are used as training set. The last 4 recordings are used as test set, in order to compare the generalized movements reproduced by the robot with the recordings of the user achieving the task in the same situation. Thus, for two reproduction attempts, the robot will not only need to generalize the movement to new ending frames (characterized by position and orientation), but it will also need to adapt the movement to new starting frames (even though only the same starting frame was observed in the training phase). A task-parameterized model with 3 components is used to learn the movement (selected by *Bayesian information criterion* [10]).

In this experiment, $\boldsymbol{\xi}_n$ and $\{\boldsymbol{b}_{n,j}, \boldsymbol{A}_{n,j}\}_{j=1}^{P}$ are defined as

$$\boldsymbol{\xi}_n = \begin{bmatrix} t_n \\ \boldsymbol{x}_n^p \\ \boldsymbol{x}_n^r \end{bmatrix} \begin{matrix} \} \boldsymbol{\xi}_n^{\mathcal{I}} \\ \} \boldsymbol{\xi}_n^{\mathcal{O}} \end{matrix}, \; \boldsymbol{b}_{n,j} = \begin{bmatrix} 0 \\ \boldsymbol{p}_{n,j} \\ \boldsymbol{r}_{n,j} \end{bmatrix}, \boldsymbol{A}_{n,j} = \begin{bmatrix} 1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{n,j} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{7}$$

[2]We focus here on the transportation aspect, i.e., the peg is smaller than the extruded cones and no insertion force is considered.

where $t_n$ is a time step, $\boldsymbol{x}_n^p$ is a 3-dimensional Cartesian position variable, and $\boldsymbol{x}_n^r$ is a 2-dimensional orientation variable (pan-tilt angles). $\boldsymbol{p}_{n,j}$ denotes the 3-dimensional Cartesian position of frame $j$. $\boldsymbol{r}_{n,j}$ and $\boldsymbol{R}_{n,j}$ both represent the orientations of frame $j$, expressed respectively as pan-tilt angles and rotation matrices. $\boldsymbol{0}$ and $\boldsymbol{I}$ are zeros matrices/vectors and identity matrices of appropriate size.

Note that this choice of coordinate system remains valid for a wide range of tasks. It corresponds to the situation in which time is not modulated by the frames, and in which pan-tilt data should be shifted with $\boldsymbol{r}_{n,j}$ to obtain a relative orientation (no additional rotation).

Currently, this parameterization of the candidate frames is left to the experimenter. A potential solution to omit this step it is to pre-select many candidate frames and let the system discover which are the most relevant (at the expense of requiring more demonstrations to obtain sufficient statistical information to learn the model).

The optimal control part is implemented in configuration space, with the aim of reducing the acceleration control commands at the joint angles level. for this control strategy, the only parameter left to the experimenter is the weight matrix $\boldsymbol{R}$ in Eqs (5) and (6). It is set in our experiment to $\boldsymbol{R} = \boldsymbol{I}$ (identity matrix). Alternatively, another strategy would be to set $\boldsymbol{R}_n$ in an online manner to apply the local minimal intervention selectively to the joints (e.g., to cope with temporary damaged, unpowered or weak motors, or to preserve some degrees of freedom for additional task constraints).

In order to analyze the effects of the LQR methods, the reproductions are also achieved with a proportional-derivative controller of constant diagonal stiffness and damping gains, as in Eq. (4), with a stiffness empirically tuned by the experimenter ($\hat{\boldsymbol{K}}_n^{\mathcal{P}} = \boldsymbol{I} \cdot 100$), and a damping adjusted to obtain a damping ratio of $1/\sqrt{2}$.

### A. Experimental results

Fig. 2 shows the generalization capability of the approach (with a controller of constant gains). We can see that smooth movements are reproduced and that the system can easily extrapolate the task to new situations that are far from the demonstrations.

Fig. 3 presents the results when combining the probabilistic estimation with an optimal control strategy. The results show only little difference in the movements retrieved by the two methods. We can see that the two LQR approaches with finite and infinite horizons, minimizing costs in Eqs (5) and (6), efficiently exploit the predicted task redundancy (red flow tube). This is achieved according to the predicted precision requirements varying along the movement, by privileging control commands oriented towards the most relevant directions of the task (corrections in the most invariant directions).

Fig. 4-(a-b) presents the averages and standard deviations of the cumulated accelerations and jerks over the four reproduction attempts in Fig. 3. Compared to a proportional-derivative controller with constant gains, the LQR methods
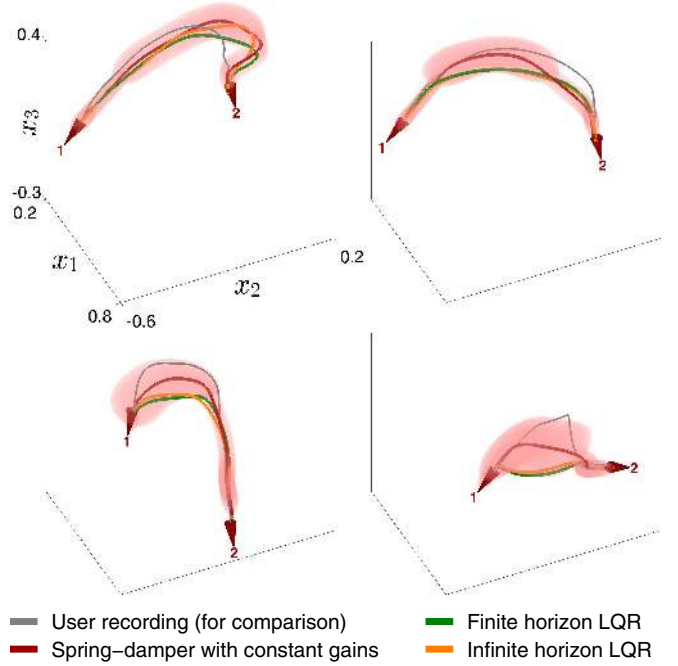


Fig. 3. Reproduction results for the 4 new situations with minimal intervention control (green and orange lines). For comparison, the user recordings (not provided in the training set) and the reproductions with constant gains are also depicted (gray and red lines). The transparent flow tube in red depicts the series of Gaussians $\mathcal{N}(\hat{\boldsymbol{\xi}}_n^{\boldsymbol{x}}, \hat{\boldsymbol{\Sigma}}_n^{\boldsymbol{x}})$ computed in Eq. (3), estimated at each time step $n$. We can see that the volume is larger in the middle of the motion than at the beginning and at the end.
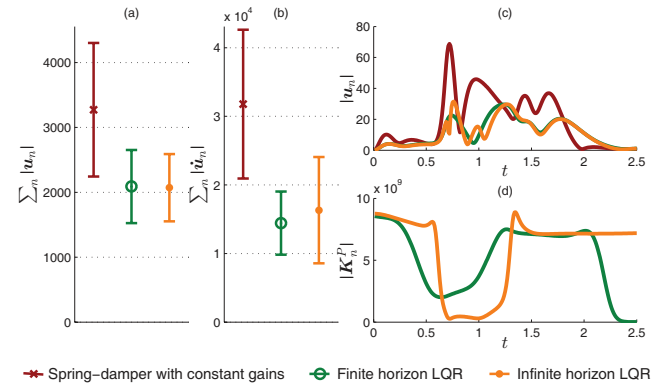


Fig. 4. Effects of the minimal intervention controller.

with finite and infinite horizons could, on average, reduce the cumulated accelerations in joint space by $36\%$ and $37\%$. The average jerks in joint space were respectively reduced by $55\%$ and $49\%$. Fig. 4-(c) shows the evolution of the joint accelerations for the first reproduction attempt.

Fig. 4-(d) depicts the evolution of the stiffness matrix determinant for the first reproduction attempt. The profile for the controller with constant gains is out of the range and not depicted (constant determinant of $10^{14}$). We can observe that both LQR controllers first require to guide the peg out of the first cone and can then reduce the gains in the phase of the transportation that does not require high precision. When reaching the second cone, the stiffness increases to

guide the robot toward the cone. When the peg reaches its final destination, the solution with finite horizon returns to a fully compliant mode (desired final feedback terms set by the experimenter), while the solution with infinite horizon regulates the movement based on the latest estimate of the required precision $Q_T$.

Qualitatively, the difference of behaviors between the two versions of LQR is in this experiment very small (see also accompanying video[3]). However, this difference could increase for other types of movements in which we can expect LQR with finite horizon to provide a better estimate, due to the consideration of the varying precision in the remaining part of the movement during the adjustment of the gains.

For tasks in which the candidate frames can change between two consecutive reproduction trials but will not change during the execution of the task, both LQR approaches can be computed with an iteration time slightly below 1 *msec* on a standard laptop. For coordinate systems moving during the execution of the task, the infinite horizon LQR conserves the same computation time, while the finite horizon LQR varies from 0.1 *sec* to 1 *msec* depending on the remaining part of the motion to be completed.

## IV. CONCLUSION AND FUTURE WORK

We presented an approach capable of adapting the centers and covariance matrices of a GMM to external task parameters represented as candidate frames of reference. This task-parameterized model is applied in the context of learning from demonstration to encode and generalize a demonstrated task to new situations. We showed that the approach could be combined with a virtual spring-damper system with variable impedance gains. For new position and orientation of candidate frames, the system generates a flow tube predicting the path of the virtual spring and its variations. The covariance information is exploited in an optimal control strategy to locally reduce the control commands according to the precision required at each step of the task. Two different minimization strategies were considered to estimate varying full stiffness and damping matrices for the regulation of the movement, depending on the (non-)availability of the predicted covariances over the whole movement.

Our future work aims at exploiting the current model within a context-based inverse optimal control (IOC) approach developed in [23]. After providing candidate rewards such as minimizing torques, torque-changes, target tracking errors, etc, IOC could first be used to determine the parts of the tasks in which these costs are prominent. The variability of the task could then be exploited in different manners depending on the context and on the most salient objective functions extracted during demonstrations.

## REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371–1394.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[3] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature Neuroscience*, vol. 5, pp. 1226–1235, 2002.

[4] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," *Nature Reviews*, vol. 12, pp. 739–751, 2011.

[5] 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.

[6] J. Ting, A. D'Souza, S. Vijayakumar, and S. Schaal, "Efficient learning and feature detection in high dimensional regression," *Neural Computation*, pp. 831–886, 2010.

[7] 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.

[8] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 1994, pp. 120–127.

[9] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013, pp. 610–616.

[10] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[11] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 554–560.

[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, P. Kormushev, and D. G. Caldwell, "Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 369–379, April 2013.

[14] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *Proc. AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013, pp. 1422–1428.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[16] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*. Cambridge, MA, USA: MIT Press, 1999, pp. 355–368.

[17] D. Hsu and S. M. Kakade, "Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions," in *Conf. on Innovations in Theoretical Computer Science*, 2013, pp. 11–20.

[18] G. J. McLachlan, D. Peel, and R. W. Bean, "Modelling high-dimensional data by mixtures of factor analyzers," *Computational Statistics and Data Analysis*, vol. 41, no. 3-4, pp. 379–388, 2003.

[19] C. Bouveyron and C. Brunet, "Model-based clustering of high-dimensional data: A review," *Computational Statistics and Data Analysis*, 2013, in press.

[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] J. R. Medina, D. Lee, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *IEEE Intl Conf. on Robotics and Automation (ICRA)*, May 2012, pp. 1025–1031.

[22] 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.

[23] M. S. Malekzadeh, D. Bruno, S. Calinon, T. Nanayakkara, and D. G. Caldwell, "Skills transfer across dissimilar robots by learning context-dependent rewards," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013, pp. 1746–1751.

[3] http://programming-by-demonstration.org/ICRA2014/