

# Learning from Demonstrations with Partially Observable Task Parameters

Tohid Alizadeh<sup>1</sup>, Sylvain Calinon<sup>1,2</sup> and Darwin G. Caldwell<sup>1</sup>

**Abstract**—Robot learning from demonstrations requires the robot to learn and adapt movements to new situations, often characterized by position and orientation of objects or landmarks in the robot’s environment. In the task-parameterized Gaussian mixture model framework, the movements are considered to be modulated with respect to a set of candidate frames of reference (coordinate systems) attached to a set of objects in the robot workspace. Following a similar approach, this paper addresses the problem of having missing candidate frames during the demonstrations and reproductions, which can happen in various situations such as visual occlusion, sensor unavailability, or tasks with a variable number of descriptive features. We study this problem with a dust sweeping task in which the robot requires to consider a variable amount of dust areas to clean for each reproduction trial.

## I. INTRODUCTION

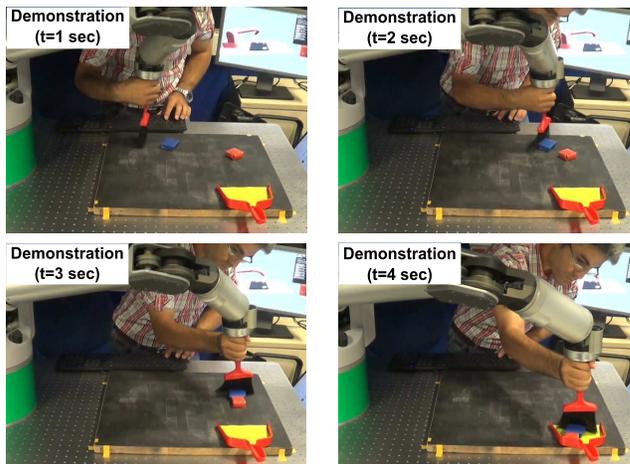
Robot programming by demonstration provides a way of teaching the robot to acquire a new skill and perform a specific task, without requiring the end user to be a robotic expert or to have professional programming skills [3].

In most tasks, the motion depends on a set of coordinate systems, or task parameters representing the locations of intermediary via-points or targets (virtual or real) that can locally influence the shape, amplitude, direction and timing of movements. Thus, approaches that take into account the dependence of the motion with respect to external task parameters are needed. In this way, after providing demonstrations with associated task parameters, the robot can generalize the task to a new set of task parameters. This process should rely on a small number of demonstrations from the user. Several approaches towards this aim have been proposed [2], [4]–[13]. These approaches use a variety of tools including Gaussian mixture regression, hidden Markov models, dynamic movement primitives, Gaussian process regression and reinforcement learning, see [1] for an overview of the available approaches.

In the above approaches, the task parameters (query points or style variables), which are used for the demonstrations and reproductions are considered to be always available. This can be a limiting factor in situations where some task parameters could be missing in the reproduction phase. This problem is studied here with an approach based on an extension of the task-parameterized Gaussian mixture model [1] able to reproduce trajectories when some task parameters are not available.

<sup>1</sup>Department of Advanced robotics (ADVR), 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

<sup>2</sup>Idiap Research Institute, CH-1920 Martigny, Switzerland.



**Fig. 1:** Snapshots of a demonstration of the dust sweeping task through kinesthetic teaching with a gravity-compensated robot.

A dust sweeping task using a broom is considered to illustrate the applicability of the proposed model, see Fig. 1. In this task, the position and orientation of a set of dust areas and the dust pan are used to construct the task parameters. Some of these task parameters could be available or missing during demonstrations and/or reproductions. For each set of provided task parameters, the task is performed by the demonstrator and the trajectory data are recorded. The learned model is used to re-generate new trajectories for new task parameters, by considering cases when some of them are unavailable.

The paper is organized as follows. Section II summarizes the proposed extension of the task-parameterized GMM to potentially missing task parameters. Section III presents the experimental setup and the achieved results. Section IV concludes the paper and presents future work.

## II. TASK-PARAMETERIZED GAUSSIAN MIXTURE MODEL WITH POTENTIALLY MISSING TASK PARAMETERS

Each task parameter is considered here as a candidate frame of reference, represented by a position  $\mathbf{b}$  and a transformation matrix  $\mathbf{A}$ , which is composed of a set of basis vectors  $\{\mathbf{e}_1, \mathbf{e}_2, \dots\}$ . 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  $\xi_n$  from its own perspective through local projection. Similarly to the estimation of the parameters of a standard GMM, deriving this constrained

TABLE I: Overall process.

<p><b>1. Task demonstration</b></p> <ul style="list-style-type: none"> <li>- Determine <math>P</math> (maximum number of frames)</li> <li>- Perform the following loop during demonstration <ul style="list-style-type: none"> <li>- Record data point <math>\{\xi_n\}_{n=1}^N</math></li> <li>- Determine the available task parameters and store them as indices sets <math>\mathbf{V}_n</math> and <math>\mathbf{W}_j</math></li> <li>- Record available task parameters <math>\{\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{n \in \mathbf{W}_j}\}_{j \in \mathbf{V}_n}</math></li> </ul> </li> </ul> <p><b>2. Model fitting</b></p> <ul style="list-style-type: none"> <li>- Determine <math>K</math> (number of components of GMM)</li> <li>- Initialize model parameters <math>\{\pi_i, \{\mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}_{j=1}^P\}_{i=1}^K</math></li> <li>- Estimate model parameters with EM (Eq. (3-4))</li> </ul> <p><b>3. Reproduction</b></p> <ul style="list-style-type: none"> <li>- Determine which are the input <math>\mathcal{I}</math> and output <math>\mathcal{O}</math> variables</li> <li>- Determine the available task parameters, <math>\mathbf{V}_n</math></li> <li>- Perform the following computation iteratively <ul style="list-style-type: none"> <li>- Collect/select <math>\xi_n^{\mathcal{I}}</math> and <math>\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j \in \mathbf{V}_n}</math></li> <li>- Estimate temporary GMM parameters <math>\{\pi_i, \mu_{n,i}, \Sigma_{n,i}\}_{i=1}^K</math> modeling the joint distribution of <math>\xi_n^{\mathcal{I}}</math> and <math>\xi_n^{\mathcal{O}}</math> as <math display="block">\xi_n^{\mathcal{I}}, \xi_n^{\mathcal{O}} \sim \sum_{i=1}^K \pi_i \mathcal{N}(\mu_{n,i}, \Sigma_{n,i})</math> (Eq. (4))</li> </ul> </li> <li>- Use GMR to retrieve <math>\xi_n^{\mathcal{O}}</math> as <math display="block">\xi_n^{\mathcal{O}}   \xi_n^{\mathcal{I}} \sim \mathcal{N}(\hat{\mu}_n, \hat{\Sigma}_n)</math> (Eq. (7))</li> </ul>
--

optimization problem results in an expectation-maximization (EM) algorithm that guarantees to improve the likelihood of the model at each iteration.

The original task-parameterized GMM is formulated by considering that all task parameters are available. Here, instead, we study the case where some frames are unobservable during demonstrations and reproductions.

When cleaning a table, the available visual information are used to form a path and move through a set of uncleaned area. If the same task is done in the dark while only some parts of the table are visible, this path will be modified according to the missing information. In complete darkness, an average path covering the table would be produced. We aim to achieve similar behaviours with our model. Therefore, when a task parameter is not accessible, it will be simply ignored and the model will be estimated or will retrieve movements based on the available task parameters.

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  $\xi_n = [t_n, \mathbf{x}_n] \in \mathbb{R}^{D+1}$  (e.g.  $D = 4$  for Cartesian position in 3D and orientation angle) is associated with the observed task parameters  $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j \in \mathbf{V}_n}$  that represent the available candidate frames of reference (at most  $P$ ). Here  $n$  and  $j$  refer to time step and frame of reference, respectively.  $\mathbf{V}_n$  is a set of indices corresponding to the observable frames of references at time step  $n$ . For example, in the case of a task with 4 frames of references in which the second frame is missing, we have  $\mathbf{V}_n = \{1, 3, 4\}$ .

The parameters of the task-parameterized GMM with  $K$  components are  $\{\pi_i, \mathbf{Z}_{i,j}^\mu, \mathbf{Z}_{i,j}^\Sigma\}$ , representing respectively the mixing coefficients, centres and covariance matrices for each frame  $j$  and mixture component  $i$ . At time step  $n$ , during both learning and reproduction phases, the resulting center  $\mu_{n,i}$  and covariance matrix  $\Sigma_{n,i}$  of each component

$i$  correspond to products of linearly transformed Gaussians

$$\mathcal{N}(\mu_{n,i}, \Sigma_{n,i}) \sim \prod_{j \in \mathbf{V}_n} \mathcal{N}(\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}, \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top),$$
 (1)

computed as

$$\Sigma_{n,i} = \left( \sum_{j \in \mathbf{V}_n} \left( \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top \right)^{-1} \right)^{-1},$$

$$\mu_{n,i} = \Sigma_{n,i} \sum_{j \in \mathbf{V}_n} \left( \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\Sigma \mathbf{A}_{n,j}^\top \right)^{-1} (\mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}).$$
 (2)

The difference between Eqs. (1-2) and the original one lies in the indices used for the summation and multiplication. In the original one, the indices includes all frames of references, while here it includes only the available ones. The parameters of the model are iteratively estimated with the following EM procedure:

E-step:

$$\gamma_{n,i} = \frac{\pi_i \mathcal{N}(\xi_n | \mu_{n,i}, \Sigma_{n,i})}{\sum_{k=1}^K \pi_k \mathcal{N}(\xi_n | \mu_{n,k}, \Sigma_{n,k})}.$$
 (3)

M-step:

$$\pi_i = \frac{\sum_{n=1}^N \gamma_{n,i}}{N}, \quad \mathbf{Z}_{i,j}^\mu = \frac{\sum_{n \in \mathbf{W}_j} \gamma_{n,i} \mathbf{A}_{n,j}^{-1} [\xi_n - \mathbf{b}_{n,j}]}{\sum_{n \in \mathbf{W}_j} \gamma_{n,i}},$$

$$\mathbf{Z}_{i,j}^\Sigma = \frac{\sum_{n \in \mathbf{W}_j} \gamma_{n,i} \mathbf{A}_{n,j}^{-1} [\xi_n - \tilde{\mu}_{n,i,j}] [\xi_n - \tilde{\mu}_{n,i,j}]^\top \mathbf{A}_{n,j}^{-\top}}{\sum_{n \in \mathbf{W}_j} \gamma_{n,i}},$$
 (4)

with  $\tilde{\mu}_{n,i,j} = \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^\mu + \mathbf{b}_{n,j}$ . The set  $\mathbf{W}_j$  contains the time steps indices in which the frame  $j$  is available,

$$\mathbf{W}_j = \{n = 1 \dots N | j \in \mathbf{V}_n\}.$$

Compared to the original EM formulation for task-parameterized GMM [1], the indices include only the the datapoints in which the frame of reference is available during the demonstration, instead of encompassing all the data points for all the frames of references.

The model parameters are initialized by clustering the data with equal time intervals. After the model is learnt for the provided demonstration data, it can be used to produce new trajectories, given a set of new task parameters. At first step, the input and output variables,  $\xi^{\mathcal{I}}$  and  $\xi^{\mathcal{O}}$ , should be specified (here, we consider the time as the input variable, but in general, any variable or set of variables could be considered as the input of the system). Accordingly,  $\mu_{n,i}$  and  $\Sigma_{n,i}$  will be partitioned into input and output parts as:

$$\mu_{n,i} = \begin{bmatrix} \mu_{n,i}^{\mathcal{I}} \\ \mu_{n,i}^{\mathcal{O}} \end{bmatrix}, \quad \Sigma_{n,i} = \begin{bmatrix} \Sigma_{n,i}^{\mathcal{I}} & \Sigma_{n,i}^{\mathcal{IO}} \\ \Sigma_{n,i}^{\mathcal{OI}} & \Sigma_{n,i}^{\mathcal{O}} \end{bmatrix}$$

Then, for each time step, the set of accessible task parameters is provided as  $\{\mathbf{A}_{n,j}, \mathbf{b}_{n,j}\}_{j \in \mathbf{V}_n}$ . A GMM is built for the

input and output variables at time step  $n$ , based on the learnt task-parameterized GMM parameters as:

$$\xi_n^I, \xi_n^O \sim \sum_{i=1}^K \pi_i \mathcal{N}(\xi_n^I, \xi_n^O | \mu_{n,i}, \Sigma_{n,i}), \quad (5)$$

in which  $\mu_{n,i}$  and  $\Sigma_{n,i}$  are computed using Eqs. (1-2). Note that the unobservable frames do not affect the centers and covariance matrices. Then Gaussian mixture regression (GMR) is applied to estimate the value of the output variable,  $\xi_n^O$  given the value of the input variable,  $\xi_n^I$ :

$$\xi_n^O | \xi_n^I \sim \mathcal{N}(\hat{\mu}_n, \hat{\Sigma}_n), \quad (6)$$

in which

$$\begin{aligned} \hat{\mu}_n &= \sum_{i=1}^K \phi_i(\xi_n^I) \left[ \mu_{n,i}^O + \Sigma_{n,i}^{OI} \left( \Sigma_{n,i}^{II} \right)^{-1} \left( \xi_n^I - \mu_{n,i}^I \right) \right], \\ \hat{\Sigma}_n &= \sum_{i=1}^K \phi_i^2(\xi_n^I) \left[ \Sigma_{n,i}^{OO} - \Sigma_{n,i}^{OI} \left( \Sigma_{n,i}^{II} \right)^{-1} \Sigma_{n,i}^{IO} \right], \\ \phi_i(\xi_n^I) &= \frac{\pi_i \mathcal{N}(\xi_n^I | \mu_{n,i}^I, \Sigma_{n,i}^{II})}{\sum_{k=1}^K \pi_k \mathcal{N}(\xi_n^I | \mu_{n,k}^I, \Sigma_{n,k}^{II})} \end{aligned} \quad (7)$$

The estimated output in Eq. (7) encapsulates variation and correlation information in the form of a probabilistic flow tube [15], continuously differentiable in time.

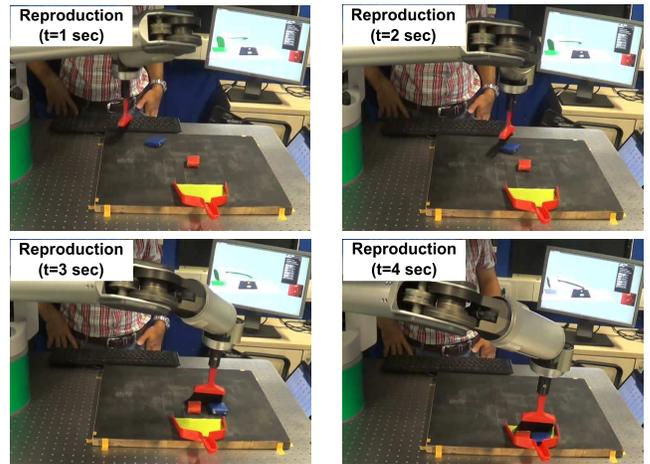
Afterwards, the estimated  $\hat{\mu}_n$  is used by the controller as the best estimate of  $\xi_n^O$  at this time step. We can see with (2) and (7) that, when a task parameter is missing, the corresponding  $Z_{i,j}^\mu$  and  $Z_{i,j}^\Sigma$  will still allow the estimation of  $\hat{\mu}_n$  and  $\hat{\Sigma}_n$ . Model selection is compatible with the techniques applied for the conventional GMM. The Matlab and C++ source codes of the proposed model are available at <http://programming-by-demonstration.org>.

A summary of the proposed approach is provided in Table I. In the next section, the proposed approach is demonstrated using a robotic dust sweeping experiment.

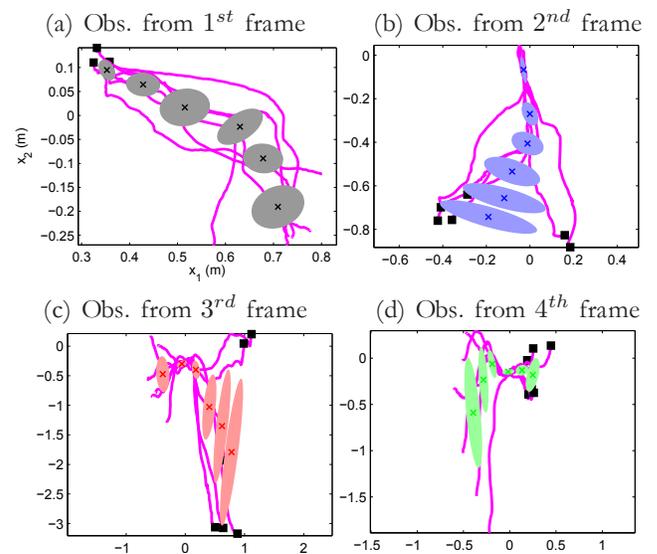
### III. ROBOTIC DUST SWEEPING EXPERIMENT

#### A. Experimental setup

In order to test the applicability of the proposed approach, the task of sweeping dust areas on a table is considered. The aim is to sweep a set of dust spots with a broom on a table and collect the dust inside a dust pan. For this task, the shape of the movement is modulated by the position and orientation of the dust spots and the dust pan. The task requires to move the broom attached to the end-effector of the robot from an initial pose in the robot workspace towards a first marker representing a dust spot, then move towards a second marker representing a second dust spot and finally move towards the dust pan, with suitable orientation (if the dust spots and the dust pan are approximated by an ellipse with two main directions in 2D, the broom will be oriented in the direction of the principal axis of the ellipse. In this way the dust pieces will be collected inside the dust pan (the broom is oriented according to the shape of the dust pan).



**Fig. 2:** Snapshots of a reproduction attempt of the dust sweeping task after model learning.

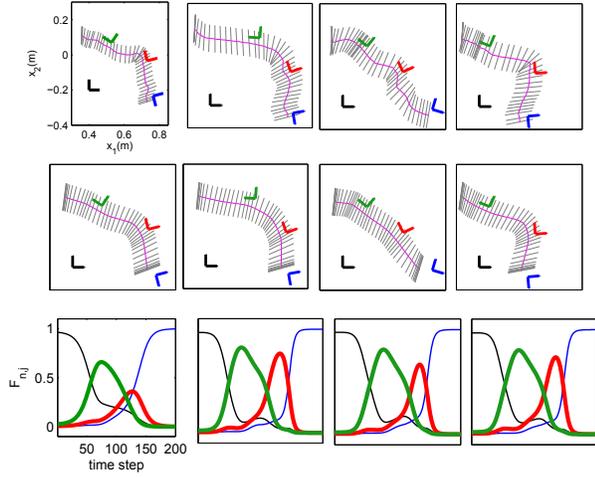


**Fig. 3:** Demonstration data observed from different frames of references and the obtained model after learning (the movements start from the black squares).

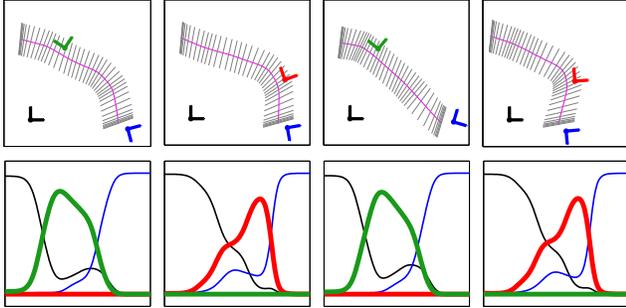
In this experiment two dust areas are considered and therefore there will be 4 frames of references as the task parameters ( $P = 4$ ):

1. The robot frame (which is fixed all the time),
2. The dust pan frame,
3. The dust piece #1 frame,
4. The dust piece #2 frame.

The task parameters are here constant during the movement for each demonstration. The experiment is implemented in a Barrett WAM torque-controlled 7 DOFs manipulator. The robot is gravity-compensated during demonstrations and reproductions. The orientation of the end-effector is kept perpendicular to the worktop, with the orientation along the vertical axis determined by the learning approach. The demonstrated trajectories are tracked by a virtual spring-damper system. Forces acting as gravity compensation are superposed to the tracking forces, resulting in a safe con-



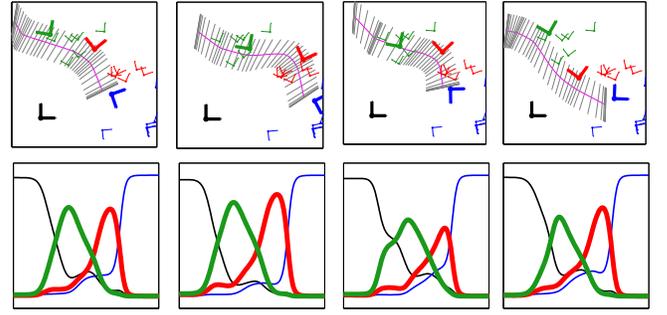
**Fig. 4:** Top: samples of the demonstrations, Middle: the corresponding reproductions, and Bottom: the relative importance of the frames during the reproduction, after model learning in the presence of all the frames (task parameters). The task parameters are shown as frames of references (Black, blue, red and green correspond to the first, second, third and fourth frame of reference, respectively).



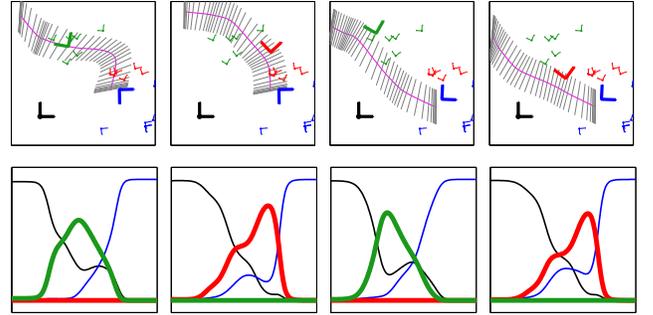
**Fig. 5:** Top: samples of the reproductions for the same set of parameters as the demonstration in the case of missing frames. Bottom: corresponding importance of the frames during the reproduction (See also Fig. 4 for the legend of the graphs).

troller for the user, who can exploit the redundancy of the robot and the redundancy of the task during demonstration and reproduction, see [14] 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.

The information regarding the position, size and orientation of the dust pieces and the dust pan are tracked by a camera using color-based image processing and homography transformation to unwrap the camera image. Task parameters for each frame are then built using this information. Fig. 1 shows the experimental setup. All demonstrations start from nearly the same position and orientation of the broom. The motions are described by 4 variables ( $D = 4$ ), representing the Cartesian position of the broom ( $x_1, x_2, x_3$ ) and its angle



**Fig. 6:** Top: reproductions for new set of task parameters. Bottom: the corresponding importance of the frames during the reproduction, in the presence of all the task frames (task parameters). The task parameters of the demonstrations are depicted using smaller and thinner frames of references (See also Fig. 4 for the legend of the graphs).



**Fig. 7:** Top: reproductions for the new set of parameters in the case of missing frames. Bottom: the corresponding importance of the frames during the reproduction. Only the available task parameters are shown in the first row. The task parameters used for the demonstrations are depicted using smaller and thinner frames of references (See also Fig. 4 for the legend of the graphs).

with respect to a vertical axis. Therefore, we have

$$\mathbf{b}_{n,1}=\mathbf{0}, \mathbf{b}_{n,j}=\begin{bmatrix} 0 \\ \mathbf{p}_{n,j} \\ \alpha_{n,j} \end{bmatrix}, \mathbf{A}_{n,1}=\mathbf{I}, \mathbf{A}_{n,j}=\begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{R}_{n,j} & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix}$$

in which  $\mathbf{p}_{n,j}$ ,  $\alpha_{n,j}$  and  $\mathbf{R}_{n,j}$  refer to the Cartesian position, the angle and the orientation (as full rotation matrix) of the dust pan and two dust pieces at time step  $n$ , respectively.  $\mathbf{0}$  is a vector or a matrix of 0s of appropriate size and  $\mathbf{I}$  is the identity matrix. In total, 6 demonstrations are provided with different locations of the dust pieces and dust pan. Models with  $K = 6$  components are considered in the experiment (selected empirically).

### B. Experimental results

Snapshots of the reproduction of the task by the robot is presented in Fig. 2. Please refer to the accompanying video of the experiment for a better insight. Fig. 3 presents the demonstration data observed from the point of view of frames 1-4 (in  $2D$ , on the horizontal plane). The obtained model after learning is also depicted with Gaussian components with centers  $\mathbf{Z}_{i,j}^\mu$  and covariance matrices  $\mathbf{Z}_{i,j}^\Sigma$  for

each frames of references. From the point of view of the 1<sup>st</sup> frame, the trajectories are close to each other at the beginning of the movements and therefore the corresponding Gaussian component has smaller covariance matrix than the other components. In the case of the 2<sup>nd</sup> frame, since all the movements end at this frame, the trajectories are close to each other at the end of the movements, and therefore, the corresponding Gaussian component has a small covariance matrix, while in the beginning of the movements, the trajectories are diverse and the corresponding Gaussian component has a larger covariance matrix. The 3<sup>rd</sup> and the 4<sup>th</sup> frames of references have small Gaussian components at different timing in the middle of the movement and bigger ones at the beginning and the end of the movement.

In Fig. 4, the demonstration samples are depicted in the first row (in  $2D$ , from the top view). In the second row, reproductions for the same set of task parameters as the demonstrations are presented, in which all the frames of references are available. The orientation of the dust broom is shown by segments along the trajectories. The reproduced data follows reasonably the trajectory and the orientation of the dust broom. In the third row of Fig. 4, the importance of the frames during the reproduction is shown. The importance of frame  $j$  at time step  $n$ ,  $F_{n,j}$  is computed as the ratio of the precision matrix determinant for a given frame with respect to the other frames at time step  $n$ , defined as:

$$F_{n,j} = \frac{|\Sigma_{n,j}^{-1}|}{\sum_{k=1}^K |\Sigma_{n,k}^{-1}|}, \quad (8)$$

where

$$\Sigma_{n,j} = \sum_{i=1}^K \phi_i(\xi_n^I) \left[ \Sigma_{n,i,j}^O - \Sigma_{n,i,j}^{OI} (\Sigma_{n,i,j}^I)^{-1} \Sigma_{n,i,j}^{IO} \right], \quad (9)$$

and

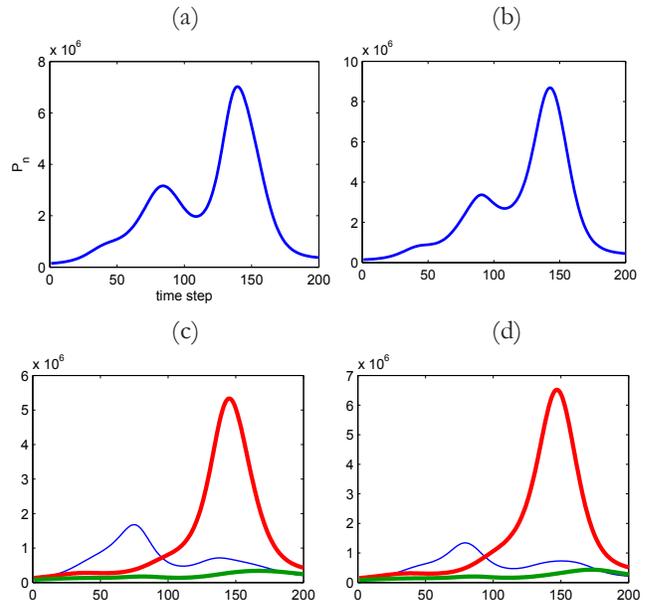
$$\Sigma_{n,i,j} = \mathbf{A}_{n,j} \mathbf{Z}_{i,j}^{\Sigma} \mathbf{A}_{n,j}^{\top}. \quad (10)$$

These results show that the first frame is the most important in the beginning of the movement, then gradually the fourth (1st dust area), third (2nd dust area) and the second (dust pan) frames become important.

In Fig. 5, reproductions in the case of missing frames are presented. In the first row, reproductions for the same set of demonstrated task parameters are presented but one of the frames is removed. Only the available task parameters are shown in the image and the importance of the frames are shown below the reproduction trajectories. Note that the importance of the missing frame is always 0, and the importance pattern for the available frames shares similarities with the full frame availability condition in case in Fig. 4. We can see that, the role of the missing frame is shared by the neighbouring frames. These results show that the proposed approach tackles the absence of task parameters in a natural way by reproducing the movement based on the available task parameters. The reproduction results show that the model can retrieve appropriate motions from, with

the dust piece collected in the dust pan at the end of the motion.

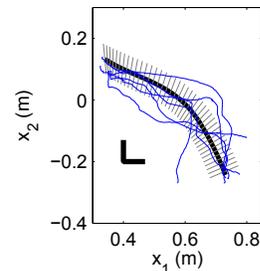
Similar reproduction results are illustrated in Figs 6 and 7 for new sets of task parameters, demonstrating the generalization capability of the approach. Reproductions in the case of new



**Fig. 8:** Average precision for reproductions in different situations. In (c) and (d), the blue, red and green graphs refer to the case in which 3<sup>rd</sup>, 4<sup>th</sup> and both 3<sup>rd</sup> and 4<sup>th</sup> frames is missing, respectively. See text for additional description.

task parameters in which one of the frames is missing are depicted in Fig. 7. We can see that, the approach is able to tackle the absence of the task parameter and at the same time generalize to new situations.

In Fig. 8 a measure of precision is shown for reproductions in different situations. This measure is calculated as the determinant of the precision matrix retrieved by GMR, at each time step ( $P_n = |\hat{\Sigma}_n^{-1}|$ ). Fig. 8 (a) presents the evolution of this variable for the reproductions in which all the frames of references are available and are the same as the demonstrations, averaged over 6 reproductions. Fig. 8 (b) shows the evolution for the new set of task parameters (all accessible), averaged over 4 reproductions. We can observe



**Fig. 9:** Reproduction (black, thick line) for the case in which only the robot frame is available. For comparison, the demonstrations are plotted using thinner (blue) lines.

similar patterns in (a) and (b). The two main peaks in both graphs correspond to time intervals in which the trajectory passes close to the 4<sup>th</sup> and the 3<sup>rd</sup> frames of references, respectively. Fig. 8 (c-d) correspond to the reproduction conditions respectively depicted in Figs. 5 and 7. It can be seen from these graphs that when a task parameter is not accessible, there is a natural smoothing effect on the trajectory around the corresponding time interval, where the movement is retrieved less precisely in that region. This behaviour is predictable: when the task parameter is missing, there is no need to be precise for the time intervals which correspond to the corresponding frame of reference.

In the extreme case where only the robot frame is available (namely, without vision tracking), the approach is still able to perform an appropriate average movement, as depicted in Fig. 9. The retrieved movement corresponds to an average behavior based on the previous experiences in the dataset.

#### IV. CONCLUSIONS

An extension of the task-parameterized GMM model was presented to handle partially observable task parameters, both in the demonstration and the reproduction phases. This is achieved by building the model and retrieving output variable based only on the available task parameters at each time step. The proposed extension is simple and effective, and its performance is demonstrated using a robotic dust sweeping task.

The approach that we propose is well suited for problems in which the task parameters can be represented in the form of coordinate systems, and where it is desirable to have a reasonable movement when a task parameter is missing. The approach retrieves a confidence measure on the estimated path, which could be exploited in future work to automatically determine if the movement can be reproduced when some frames are missing (e.g., partial occlusions), and wait for the occlusion to disappear if the measure of confidence on the retrieved movement is too low.

The proposed approach considers only the cases in which the number of task parameters in the reproduction is smaller or equal to the number of task parameters in the demonstration. In other words, each task parameter should be present at least once during the demonstrations. In order to deal with excessive task parameters, other approaches should be developed to automatically discard redundant and irrelevant frames from larger sets of candidate frames.

Another way of dealing with the missing task parameters, that could be investigated in future work, is to use a similar

strategy as in the problem of matrix and higher order tensor completion algorithms [16].

Investigating the similar case of missing task parameters using other learning approaches could be another line of the future works to consider.

#### REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [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, no. 10, pp. 1327–1339, 2012.
- [7] F. Stulp, G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud, "Learning compact parameterized skills with a single regression," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 2013.
- [8] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Real-time stylistic prediction for whole-body human motions," *Neural Networks*, vol. 25, pp. 191–199, 2012.
- [9] —, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Networks*, vol. 24, no. 5, pp. 493–500, 2011.
- [10] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, no. 4, pp. 361–379, 2012.
- [11] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800–815, 2010.
- [12] 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.
- [13] 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)*, San Francisco, California, USA, September 2011, pp. 710–717.
- [14] 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.
- [15] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2-3, pp. 115–131, 2011.
- [16] M. Signoretto, R. Van de Plas, B. De Moor, and J. A. Suykens, "Tensor versus matrix completion: a comparison with application to spectral data," *Signal Processing Letters, IEEE*, vol. 18, no. 7, pp. 403–406, 2011.