Statistical dynamical systems for skills acquisition in humanoids

Sylvain Calinon, Zhibin Li, Tohid Alizadeh, Nikos G. Tsagarakis and Darwin G. Caldwell

***Abstract*—Learning by imitation in humanoids is challenging due to the unpredictable environments these robots have to face during reproduction. Two sets of tools are relevant for this purpose: 1) probabilistic machine learning methods that can extract and exploit the regularities and important features of the task; and 2) dynamical systems that can cope with perturbation in real-time without having to replan the whole movement. We present a learning by imitation approach combining the two benefits. It is based on a superposition of virtual spring-damper systems to drive a humanoid robot's movement. The method relies on a statistical description of the springs attractor points acting in different candidate frames of reference. It extends dynamic movement primitives models by formulating the dynamical systems parameters estimation problem as a Gaussian mixture regression problem with projection in different coordinate systems. The robot exploits local variability information extracted from multiple demonstrations of movements to determine which frames are relevant for the task, and how the movement should be modulated with respect to these frames. The approach is tested on the new prototype of the COMAN compliant humanoid with time-based and time-invariant movements, including bimanual coordination skills.**

I. INTRODUCTION

In contrast to standard industrial robots designed to work in predefined factory settings, humanoids are designed to share our space without the need to modify the infrastructure of our home environment. These robots create new collaborative and social interaction opportunities, but these new environments also expose the robots to various sources of perturbation and unpredictable situations. It stresses the requirements of providing robots with the capability of generalizing movement behaviors and skills [1]–[7]. Since each home environment is different and the range of possible tasks that the humanoids can carry out is infinite, it is not possible to provide a predefined database of tasks that the robot can do. One of the key requirement is to provide user-friendly ways of programming these robots to transfer new skills and adapt existing skills to new situations.

The core idea of acquiring skills by imitation has been labeled over the years with various names such as *teaching by showing* [8], *robot coaching* [9], *robot programming by demonstration* [10], or *learning from demonstration* [11]. It offers a promising approach to transfer and refine tasks from observation of users who are not expert in robotics and computer programming. To achieve this, the representation of the robot's movement needs to be flexible and compact to let the system store, exchange, and re-use this knowledge in an efficient way, as well as to let the robot refine the skill on its own by exploration of new control policies.

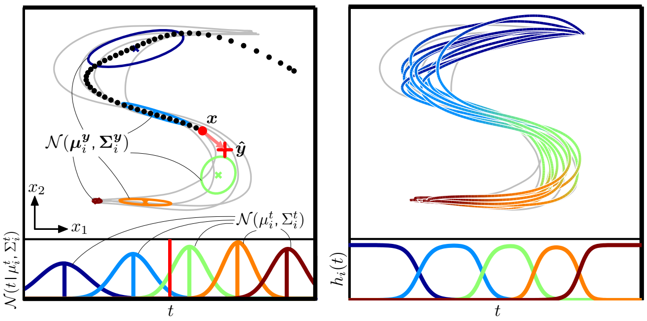


Fig. 1. Illustrative example of movement learning through the superposition of virtual springs. The left graph depicts an iteration step in the middle of the movement. The pink arrow represents the resulting spring force pointing to the current virtual attractor $\hat{\boldsymbol{y}}$. The right graph shows the stochastic movement generation process that preserves the smoothness and variability of the demonstrations.

Humanoids need to react in a smooth and fast manner to various sources of continuous perturbations (coming from users and changing environments), without necessarily having to replan the whole motion, but by modulating the movement with respect to the situation [2], [4], [5]. In other words, the frontier between planning and control algorithms should overlap in order to cope with unexpected situations in real-time.

Our work emphasizes that skills should not only be represented in terms of the shape of the movement, but also (and maybe more importantly) in terms of the degree of variations allowed by the task, and how the different variables are coordinated during the movement. Such local variation can change over time (different phases of the movement) and can depend on situation (adaptation to changing positions of objects). Thus, the important characteristics of a task do not only lie in the path that the robot should follow, but also in the variation and coordination patterns varying during the movement. This requirement is even stronger in the case of humanoids, where the robots require to exploit the characteristics of the task to respond to perturbations in real-time, with human-like movements that should be predictable to users sharing their space.

The kinematic redundancy of robots can be exploited to simultaneously solve multiple task constraints [12]. However, the exploitation of task redundancy has been largely overlooked so far. There are broadly two categories of machine learning tools that can provide humanoids with such flexibility. The first includes *statistical machine learning* approaches that provide a mechanism to encode in a compact and generic

The authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. name.surname@iit.it.

This work was partially supported by the AMARSi European project under contract FP7-ICT-248311, and by the STIFF-FLOP European project under contract FP7-ICT-287728.

manner the changing correlations across the movement variables and the variations observed among multiple demonstrations [2]–[4]. The second includes *dynamical systems* (DS) that can swiftly react to continuous perturbations [5]–[7]. DS also simplifies the coexistence with other robot's controller by sharing a similar state-space representation of the movement (e.g., with stiffness and damping parameters conventional to both fields). In particular, we are interested here in the subset of dynamical systems that can simulate virtual mass-spring-damper systems, which encapsulate physical notions of inertia, stiffness and damping that can be modified on-the-fly to modulate the characteristics of the movement in the vicinity of humans. Such systems may also facilitate the inclusion of active impedance control strategies into actuated systems with elements of passive compliance.

In previous work, we explored these two categories of approaches in parallel. On one side, we developed a representation of movements based on a sequential superposition of dynamical systems with varying full stiffness matrices [1]. On the other side, we developed a representation based on *Gaussian mixture regression* (GMR) to provide a statistical representation of the movement acting in different coordinate systems [2].

We present in this paper a model called DS-GMR that combines the advantages of these two approaches. The contributions of this paper are double. First, we bring a new probabilistic view to models such as [1] or *dynamic movement primitives* (DMP) [5], [6], by formulating the learning problem as GMR. In this way, DS systems can easily be extended to task-parameterized models such as [2] or *parametric hidden Markov models* (PHMM) [13], which allows the modulation of movements with respect to task parameters such as positions of objects. The second contribution concerns the development of a new task-parameterized model and associated *expectation-maximization* (EM) algorithm [14] to learn how different frames of reference can reshape the movement. The advantage over [13] is that both the centers and covariances of the Gaussians can be parameterized, while PHMM modulates only the centers. The advantage over [2] is that we do not need multiple models for the multiple frames. A single model is instead derived to efficiently encode the Gaussians with local projection in the different frames encapsulated directly at the level of the EM procedure.

The remainder of the paper is organized as follows. The formulation of DS in a GMR is described in Section II, and extended in Section II-A to projections in multiple coordinate systems. Section III presents experiments in a compliant humanoid with time-based and time-invariant movements. Section IV discusses the advantages and connections of the approach with other work.

II. PROPOSED APPROACH

The overall approach can be illustrated as follows. The motion of the robot is assumed to be driven by a set of virtual springs connected to a set of candidate objects or body parts of the robot (e.g., end-effectors). The learning problem consists of estimating when and where to activate these springs. This can be learned from demonstrations by exploiting the invariant characteristics of the task (the parts of the movement that are similar across the multiple demonstrations). Consistent demonstrations will result in stronger springs, while irrelevant connections will vanish.

A set of candidate frames of reference (represented as coordinate systems) is predefined by the experimenter. This set remains valid for a wide range of tasks (e.g., the hands of the robot are relevant for most manipulation skills). The role of the robot is to autonomously figure out which frames of reference matter along the task, and in which way the movement should be modulated with respect to these frames. The robot can also learn that a frame is not relevant for the task. However, predefining too many candidate frames may require the user to provide a large number of demonstrations to obtain statistically relevant information, which would conflict with the aspiration of the approach to transfer skills in a user-friendly manner.

In [15], the movement of a robot's end-effector is represented as a virtual mass driven by a weighted superposition of spring-damper systems. By setting $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ as positions, velocities and accelerations of the end-effector, the movement is described as $\ddot{\boldsymbol{x}} = \sum_{i=1}^{K} h_i \Big[\boldsymbol{K}_i^{\mathcal{P}}[\boldsymbol{\mu}_i^{\boldsymbol{x}} - \boldsymbol{x}] - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}\Big]$, where $\boldsymbol{K}_i^{\mathcal{P}}$, $\kappa^{\mathcal{V}}$ and $\boldsymbol{\mu}_i^{\boldsymbol{x}}$ are respectively the full stiffness matrix, damping term and attractor point of the i-th virtual spring. The connections of this model with *dynamic movement primitives* (DMP) is discussed in [1], [6], [15] (see also Section IV). A noticeable difference is that the non-linear force modulating the movement in the original DMP formulation is now expressed as additional sets of virtual springs, adding local corrective terms that can swiftly react to perturbations during reproduction.

More generically, the above system can be written as[1]

$$\ddot{\boldsymbol{x}} = \hat{\boldsymbol{K}}^{\mathcal{P}}[\hat{\boldsymbol{y}} - \boldsymbol{x}] - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}, \qquad (1)$$

where $\hat{\boldsymbol{y}}$ denotes the evolution/path of the virtual attractor. The learning problem is formulated in this paper as estimating the path of $\hat{\boldsymbol{y}}$ and changing stiffness $\hat{\boldsymbol{K}}^{\mathcal{P}}$ that will pull the end-effector to follow the behaviors demonstrated by the user. By first assuming that the movement is driven by (1) with a diagonal stiffness matrix with gain $\kappa^{\mathcal{P}}$, and after homogeneous rescaling of the human demonstrations to match the ratio robot/human's size, the collected datapoints $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are transformed into a variable $\boldsymbol{y} = \ddot{\boldsymbol{x}}\frac{1}{\kappa^{\mathcal{P}}} + \dot{\boldsymbol{x}}\frac{\kappa^{\mathcal{V}}}{\kappa^{\mathcal{P}}} + \boldsymbol{x}$, corresponding to the estimation of the position of the virtual attractor for each datapoint.[2]

In order to present the approach didactically, we will for now assume that time t is the driving mechanism of the

[1] Here, the equivalence is found by setting $\hat{\boldsymbol{y}} = \sum_i h_i \boldsymbol{\mu}_i^{\boldsymbol{x}}$ and $\hat{\boldsymbol{K}}^{\mathcal{P}} = \sum_i h_i \boldsymbol{K}_i^{\mathcal{P}}$.

[2] An illustrative analogy of the problem consists of estimating the trajectory of a boat pulling a water-skier such that the skier follows a desired path, with the rope acting as a spring of given stiffness and damping.

movement. We will later show that other mechanisms can be used here (e.g., time-independent autonomous system).

In this context, a *Gaussian mixture model* (GMM) is used to encode the joint distribution $\mathcal{P}(t,\boldsymbol{y})$. An *expectation-maximization* (EM) procedure [14] is used to estimate the priors (mixing coefficients) π_i, centers $\boldsymbol{\mu}_i = \begin{bmatrix}\mu_i^t \\ \boldsymbol{\mu}_i^{\boldsymbol{y}}\end{bmatrix}$ and covariances $\boldsymbol{\Sigma}_i = \begin{bmatrix}\Sigma_i^t & \boldsymbol{\Sigma}_i^{t\boldsymbol{y}} \\ \boldsymbol{\Sigma}_i^{\boldsymbol{y}t} & \boldsymbol{\Sigma}_i^{\boldsymbol{y}}\end{bmatrix}$ of the GMM. Several model selection approaches exist to estimate the number of Gaussians in a GMM (see e.g. [15]). Here, the number of Gaussians is predefined by the experimenter.

Gaussian mixture regression (GMR) [2] is then used to estimate, at each iteration, the conditional probability $\mathcal{P}(\boldsymbol{y}|t)$, estimated in the form of a new Gaussian distribution

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}^{\boldsymbol{y}} &= \sum_{i=1}^{K} h_i(t) \Big[\boldsymbol{\mu}_i^{\boldsymbol{y}} + \boldsymbol{\Sigma}_i^{\boldsymbol{y}t}(\Sigma_i^t)^{-1}[t-\mu_i^t]\Big], \\
\hat{\boldsymbol{\Sigma}}^{\boldsymbol{y}} &= \sum_{i=1}^{K} h_i^2(t) \Big[\boldsymbol{\Sigma}_i^{\boldsymbol{y}} - \boldsymbol{\Sigma}_i^{\boldsymbol{y}t}(\Sigma_i^t)^{-1}\boldsymbol{\Sigma}_i^{t\boldsymbol{y}}\Big], \qquad (2)
\end{aligned}
$$

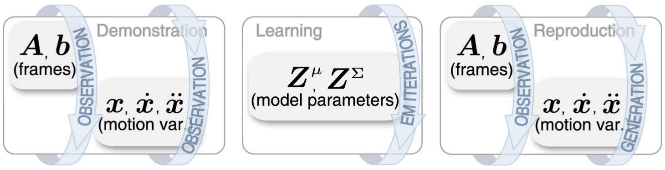
with activation weights $h_i(t)$ defined as

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

GMR can retrieve control commands in real-time, independently of the number of datapoints in the training set. Indeed, the data retrieval problem in GMR is considered as a joint probability estimation problem. The joint distribution of the data is first approximated by a mixture of Gaussians with EM. An estimate of the outputs can then be computed for each new inputs in the form of another mixture of Gaussians, by exploiting various properties of Gaussians (linear transformations, products and conditional probability). This output provides additional information about the variation and coordination of the movement variables (local shape of the movement).

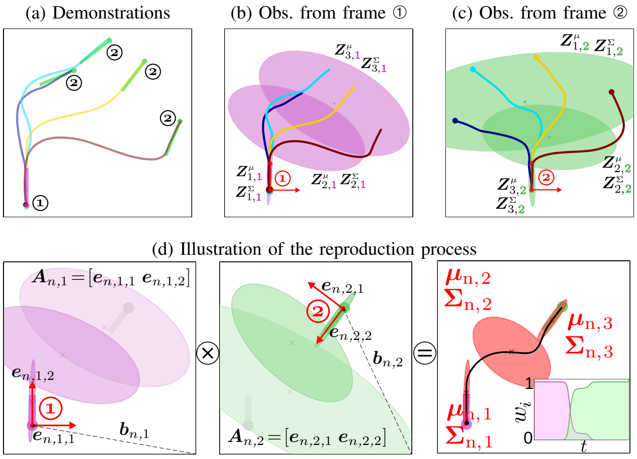
In GMR, there is no distinction between input and output components when learning the model. Namely, any subset of input-output dimensions can be selected, and expectations on the remaining dimensions can be computed in real-time. It corresponds to a convex sum of linear approximations (with weights varying non-linearly) [16]. In terms of computation, learning the model depends linearly on the number of datapoints, while prediction is independent on this number, which makes the approach an interesting alternative to kernel-based regression methods such as *Gaussian process regression* (GPR) whose processing grows with the size of the dataset.

Eq. (2) provides an estimate of the attractor point $\hat{\boldsymbol{\mu}}^{\boldsymbol{y}}$ at time step t, together with its variability in the form of a covariance matrix $\hat{\boldsymbol{\Sigma}}^{\boldsymbol{y}}$. The changing stiffness profile can be estimated as being inversely proportional to the variation in the movement. The details are presented in [1] and are omitted here due to space restriction. Fig. 1 illustrates the overall mechanism of the approach.



$\boldsymbol{A}, \boldsymbol{b}$
(frames)
Demonstration
OBSERVATION
$\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$
(motion var.)
OBSERVATION
Learning
$\boldsymbol{Z}^{\mu}, \boldsymbol{Z}^{\Sigma}$
(model parameters)
EM ITERATIONS
$\boldsymbol{A}, \boldsymbol{b}$
(frames)
Reproduction
OBSERVATION
$\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$
(motion var.)
GENERATION

Fig. 2. Information flow of the overall process. In the *demonstration phase*, movements are recorded as a set of datapoints $\{\boldsymbol{x}_n, \dot{\boldsymbol{x}}_n, \ddot{\boldsymbol{x}}_n\}$ together with the **task parameters** $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}$ representing coordinate systems. In the *learning phase*, the **model parameters** $\{\pi_i, \boldsymbol{Z}^{\mu}_{i,j}, \boldsymbol{Z}^{\Sigma}_{i,j}\}$ are estimated from the demonstrations using the EM algorithm in Eq. (4). In the *reproduction phase*, for new observations of task parameters $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}$, the system estimates on-the-fly movement commands $\ddot{\boldsymbol{x}}_n$ for each iteration n.



(a) Demonstrations
(b) Obs. from frame ①
(c) Obs. from frame ②
①
②
$\boldsymbol{Z}^{\mu}_{3,1}$ $\boldsymbol{Z}^{\Sigma}_{3,1}$
$\boldsymbol{Z}^{\mu}_{1,1}$ $\boldsymbol{Z}^{\Sigma}_{1,1}$
$\boldsymbol{Z}^{\mu}_{2,1}$ $\boldsymbol{Z}^{\Sigma}_{2,1}$
$\boldsymbol{Z}^{\mu}_{1,2}$ $\boldsymbol{Z}^{\Sigma}_{1,2}$
$\boldsymbol{Z}^{\mu}_{2,2}$ $\boldsymbol{Z}^{\Sigma}_{2,2}$
$\boldsymbol{Z}^{\mu}_{3,2}$ $\boldsymbol{Z}^{\Sigma}_{3,2}$
(d) Illustration of the reproduction process
$\boldsymbol{A}_{n,1}=[\boldsymbol{e}_{n,1,1}\ \boldsymbol{e}_{n,1,2}]$
$\boldsymbol{e}_{n,1,2}$
$\boldsymbol{e}_{n,1,1}$
$\boldsymbol{b}_{n,1}$
⊗
$\boldsymbol{e}_{n,2,1}$
$\boldsymbol{e}_{n,2,2}$
$\boldsymbol{b}_{n,2}$
$\boldsymbol{A}_{n,2}=[\boldsymbol{e}_{n,2,1}\ \boldsymbol{e}_{n,2,2}]$
⊜
$\boldsymbol{\mu}_{n,2}$ $\boldsymbol{\Sigma}_{n,2}$
$\boldsymbol{\mu}_{n,3}$ $\boldsymbol{\Sigma}_{n,3}$
$\boldsymbol{\mu}_{n,1}$ $\boldsymbol{\Sigma}_{n,1}$
1
0
w_i
t

Fig. 3. Illustration of task-parameterized movement learning based on DS-GMR. The task consists of reaching frame ② from frame ① with a desired approach angle, which is representative of skills such as reaching an object, inserting a peg in a hole, or moving a car from a parking spot to another. w_i shows the relative importance of the different frames (estimated as determinant ratio of precision matrices).

The representation of the dynamical systems parameters of Eq. (1) in the form of a standard GMM has several advantages. It is exploited here to encapsulate dynamical systems learning into a standard Gaussian mixture problem, which allows its extension to models encoding movements with respect to multiple landmarks in the robot's environment.

A. Extension to task-parameterized movements

We now consider the case where the virtual spring-damper systems can act in various candidate frames of reference. We define here a frame of reference as a coordinate system represented by a position $\boldsymbol{b}$ (origin of the observer) and a set of basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\}$ forming a transformation matrix $\boldsymbol{A} = [\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots]$. The coordinate systems can include time as coordinate, or any other variable relevant for the task. An observed movement (and the Gaussians associated to it) can be projected in different candidate frames.

As a follow-up of [2], we derive a task-parameterized model that can adapt motion and impedance behaviors in real-time with respect to the current position/orientation of frames. For example, in a bimanual task, the robot can cope with a perturbation on its left hand by moving its right hand.

The proposed extension is built upon the product properties of Gaussians.

Let us assume that each demonstration $m \in \{1,\ldots,M\}$ contains T_m datapoints forming a dataset $\{\boldsymbol{\xi}_n\}_{n=1}^{\sum_m^M T_m}$, where each datapoint $\boldsymbol{\xi}_n = [t_n, \boldsymbol{y}_n]^\top \in \mathbb{R}^{D+1}$ (e.g. $D=3$ in a 3D Cartesian space) is associated with **task parameters** $\{\boldsymbol{A}_{n,j}, \boldsymbol{b}_{n,j}\}_{j=1}^P$ representing respectively P candidate coordinate systems.

By denoting the **model parameters** $\{\pi_i, \boldsymbol{Z}_{i,j}^{\mu}$ and $\boldsymbol{Z}_{i,j}^{\Sigma}$, the center $\boldsymbol{\mu}_{n,i}$ and covariance matrix $\boldsymbol{\Sigma}_{n,i}$ of the i-th Gaussian in the GMM for the n-th datapoint are computed as products of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}) = \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{n,j}\boldsymbol{Z}_{i,j}^{\mu} + \boldsymbol{b}_{n,j},\ \boldsymbol{A}_{n,j}\boldsymbol{Z}_{i,j}^{\Sigma}\boldsymbol{A}_{n,j}^{\top}\Big), \tag{3}$$

which are used with Eq. (2) to estimate the trajectory $\hat{\boldsymbol{y}}$ of Eq. (1) through GMR.

Fig. 2 presents the overall process, and Fig. 3 a simple example. Fig. 3-*(a)* shows demonstrations with different position and orientation of frame ②. We can see in *(b-c)* these same four demonstrations observed from the point of view of frames ① and ②. *(d)* shows the reproduction process. The model parameters $\boldsymbol{Z}_{i,j}^{\mu}$ and $\boldsymbol{Z}_{i,j}^{\Sigma}$ are projected in the new positions and orientations of frames ① and ②, and each set of Gaussians in their candidate frames are multiplied to form a resulting Gaussian. These resulting Gaussians form a trajectory (or tube) of virtual attractors that are used to move the system from frame ① to frame ②. The covariance provides information on the possible ways of reproducing the movement (i.e., how the virtual attractors can be displaced while still reproducing the essential characteristics of the task).

The model parameters $\{\pi_i, \boldsymbol{Z}_{i,j}^{\mu}, \boldsymbol{Z}_{i,j}^{\Sigma}\}$ in Eq. (3) are iteratively estimated with the following EM procedure[3]

E-step:

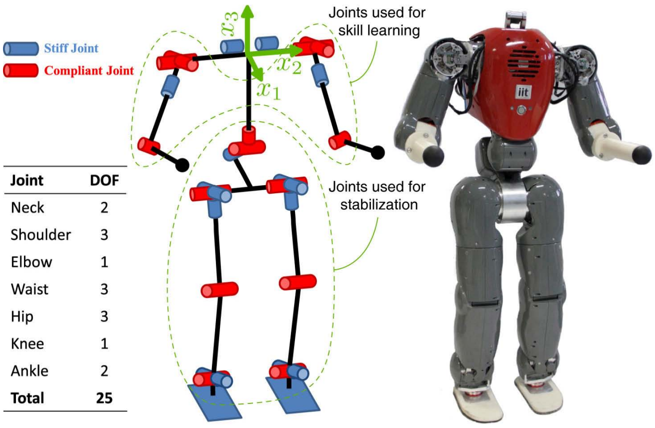
$$\gamma_{n,i} = \frac{\pi_i \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})}{\sum_k^K \pi_k \mathcal{N}(\boldsymbol{\xi}_n | \boldsymbol{\mu}_{n,k}, \boldsymbol{\Sigma}_{n,k})}.$$

M-step:

$$\boldsymbol{Z}_{i,j}^{\mu} = \frac{\sum_n \gamma_{n,i}\ \boldsymbol{A}_{n,j}^{-1}[\boldsymbol{\xi}_n - \boldsymbol{b}_{n,j}]}{\sum_n \gamma_{n,i}},$$

$$\boldsymbol{Z}_{i,j}^{\Sigma} = \frac{\sum_n \gamma_{n,i}\ \boldsymbol{A}_{n,j}^{-1}[\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}][\boldsymbol{\xi}_n - \tilde{\boldsymbol{\mu}}_{n,i,j}]^\top \boldsymbol{A}_{n,j}^{-\top}}{\sum_n \gamma_{n,i}}, \tag{4}$$

$$\text{with} \quad \tilde{\boldsymbol{\mu}}_{n,i,j} = \boldsymbol{A}_{n,j}\boldsymbol{Z}_{i,j}^{\mu} + \boldsymbol{b}_{n,j}.$$



Joint	DOF
Neck	2
Shoulder	3
Elbow	1
Waist	3
Hip	3
Knee	1
Ankle	2
Total	**25**

Fig. 4. Full-body compliant humanoid robot COMAN developed at IIT.

The priors π_i are estimated as in the EM procedure for standard GMM. The *Matlab* and C++ source codes of the proposed model are available on `http://programming-by-demonstration.org`.

III. SKILLS LEARNING EXPERIMENT

The COMAN humanoid robot is used in the experiment, which has been been designed to explore how compliance can be exploited for safer human-robot interaction, reduced energy consumption, and faster learning capabilities [17]. Fig. 4 shows the joints of the robots endowed with passive compliance (series elastic actuators).

For each arm of the robot, three candidate frames ($P=3$) are considered: a frame attached to a wooden box object ($p=1$), the robot's upper torso ($p=2$), and the robot's other hand ($p=3$). The position of the objects and the robot are tracked with a marker-based *NaturalPoint OptiTrack* motion capture system. It is composed of 12 cameras tracking position and orientation of predefined landmarks at a rate of 30 frames/second with a position tracking accuracy below 10mm. Two additional sets of markers are used in the demonstration phase, placed on the back of the demonstrator's hands.

During reproduction, the robot's upper-torso and the box are tracked by the vision system. The position of the hands are determined by proprioception (from the motors encoders and forward kinematics from the upper torso's tracked frame). The legs and torso are controlled to let the robot stand by continuously reacting to perturbations with a stabilization control scheme adapted from [18], exploiting the intrinsic and controlled compliance of the robot.

The arms are controlled with an admittance controller to let the user physically interact with the robot by grasping and moving its arms. In total, 8 *degrees of freedom* (DOFs) are used to reproduce the learned skills (namely, the two arms), while the rest of the body DOFs are used to react to perturbation and maintain balance.

Three examples of movement behaviors are presented, that will be labeled *clapping*, *tracking* and *sweeping the*

[3] The proof is omitted here due to space restriction but can be retrieved by differentiating (with respect to $\boldsymbol{Z}_{i,j}^{\mu}$ and $\boldsymbol{Z}_{i,j}^{\Sigma}$) the log-likelihood function from Eq. 3. The result shares similarities with the estimates for $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ in EM applied to GMM, with the sole difference that in the case of $\boldsymbol{Z}_{i,j}^{\mu}$ and $\boldsymbol{Z}_{i,j}^{\Sigma}$, the data are locally projected through the inverse of a local transformation defined by $\boldsymbol{A}_{n,j}$ and $\boldsymbol{b}_{n,j}$.

floor tasks. The first two are reproduced on the robot. The floor sweeping task is reproduced in simulation for practical reason, with the future goal of reproducing the task on the real platform when wrists and/or hands will be endowed in COMAN to hold a broom. The examples are chosen to be didactic and simple enough to be visualized and analyzed after learning. Each of these behaviors could have been preprogrammed separately with a few lines of code, but here, the aim is to provide a model that can be used to transfer various tasks without modification of the main program. The transfer of tasks is thus not limited to this specific set of examples. A video of the experiments accompanies the paper.

A. *Time-based and time-invariant movements*

The approach is tested with bimanual gestures, which is a challenging learning problem in humanoids [19], [20]. Let us denote the positions at iteration n of the left and right hands as $\boldsymbol{x}_n^{\mathcal{L}}$ and $\boldsymbol{x}_n^{\mathcal{R}}$. Let us represent their orientations with rotation matrices $\boldsymbol{R}_n^{\mathcal{L}}$ and $\boldsymbol{R}_n^{\mathcal{R}}$. For time-based movements, by using Eqs (1), (2) and (3), a bimanual motion is retrieved by estimating $\mathcal{P}(\boldsymbol{y}^{\mathcal{L}}|t)$ and $\mathcal{P}(\boldsymbol{y}^{\mathcal{R}}|t)$ from the Gaussians

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

$$\text{with}\quad \boldsymbol{b}_{n,P}^{*} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x}_n^{\times} \end{bmatrix}, \boldsymbol{A}_{n,P}^{*} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0}^{\top} \\ \boldsymbol{0} & \boldsymbol{R}_n^{\times} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{I} = 1$ for the special case of driving the motion through activation weights based on time (namely, scalar input for the regression process), and where $\boldsymbol{0}$ are vectors of zero elements with corresponding lengths. The exponents * and $^\times$ respectively denote $^{\mathcal{L}}$ and $^{\mathcal{R}}$ when controlling the left arm (and vice-versa for the right arm). Namely, bimanual coordination is achieved by considering the other hand as a candidate frame for the reproduction of the skill. The movement of the two hands will thus be coupled for parts of the movement in which regular patterns have been observed between the two hands. The strength of the coupling constraint will be automatically adapted with respect to the variations observed in the task.

Any other type of inputs can be used to drive the movement. We consider as an example activation weights changing with respect to the position $\boldsymbol{o}$ of the object in the 3D Cartesian space. Namely, the bimanual motion is retrieved by estimating in real-time $\mathcal{P}(\boldsymbol{y}^{\mathcal{L}}|\boldsymbol{o})$ and $\mathcal{P}(\boldsymbol{y}^{\mathcal{R}}|\boldsymbol{o})$.

B. *Experimental results*

$K = 4$, $K = 3$ and $K = 6$ are used for the clapping, tracking and sweeping tasks to initialize the models. For each task, the box is moved in the vicinity of the robot, from its right-hand side to its left-hand side by following an 'S'-shaped trajectory. The frame of reference used to present the results is shown in Fig. 4.

Fig. 5-*left* presents the results of the *hands clapping* task. The timeline graphs show that the important aspect of the task is to keep the motion of the hands coordinated (the respective hand frame is extracted as the most important). The robot does not react to the motion of the box (candidate frame irrelevant for the task). If the user grasps one hand of the robot and moves it to a new position, the robot reacts by adapting the movement of the other hand (the forces applied to the robot's right hand are represented with green arrows).

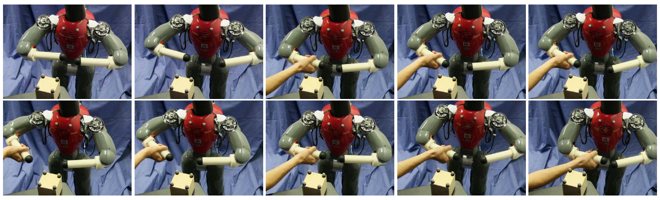


Fig. 6. Physical perturbation during reproduction of the learned hand clapping task. While grabbing and moving the right hand of the robot, the robot uses the motors of its legs and torso to keep balance, and the motors of its left hand to adapt the clapping motion to the perturbation.

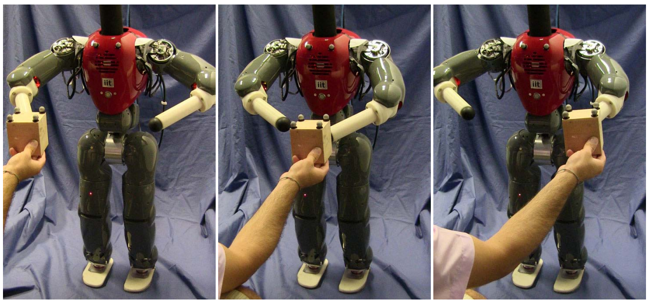


Fig. 7. Adaptive uni/bimanual tracking behaviors learned from a single demonstration. The robot autonomously acquired hands switching behaviors and changes of coordination patterns that are modulated with respect to the position of the box.

Fig. 5-*center* presents the results of the *tracking* task. The user demonstrated that the box should be reached with the right hand if it is on the right-hand side, with the left hand if it is on the left-hand side, and with the two hands if it is in the center (by pointing at the box with the forearms). After observing a single continuous demonstration showing these three behaviors in random orders, and training the model (1-10 seconds in our experiment), the robot is able to smoothly switch in real-time from one to two hands tracking, while progressively bringing back the unused hand to a natural pose (see also Fig. 7). The regions of activation of these behaviors are depicted by three ellipsoids representing the resulting Gaussians. We see in the timeline graphs that in the first part of the movement (when the object is on the right-hand side of the robot), the right hand points at it (object frame is relevant) while the left arm is in a comfortable pose (robot's frame is relevant). In the middle of the movement, when the box is in the center, the right hand continues tracking the box while the left hand moves from the comfortable pose to the left side of the box. In this case, the two relationships hand-hand and hand-box are detected to be important for the task (both object frame and hand frame equally matter). Finally, when the object moves to the left-hand side of the robot, the right hand progressively comes back to a comfortable pose,

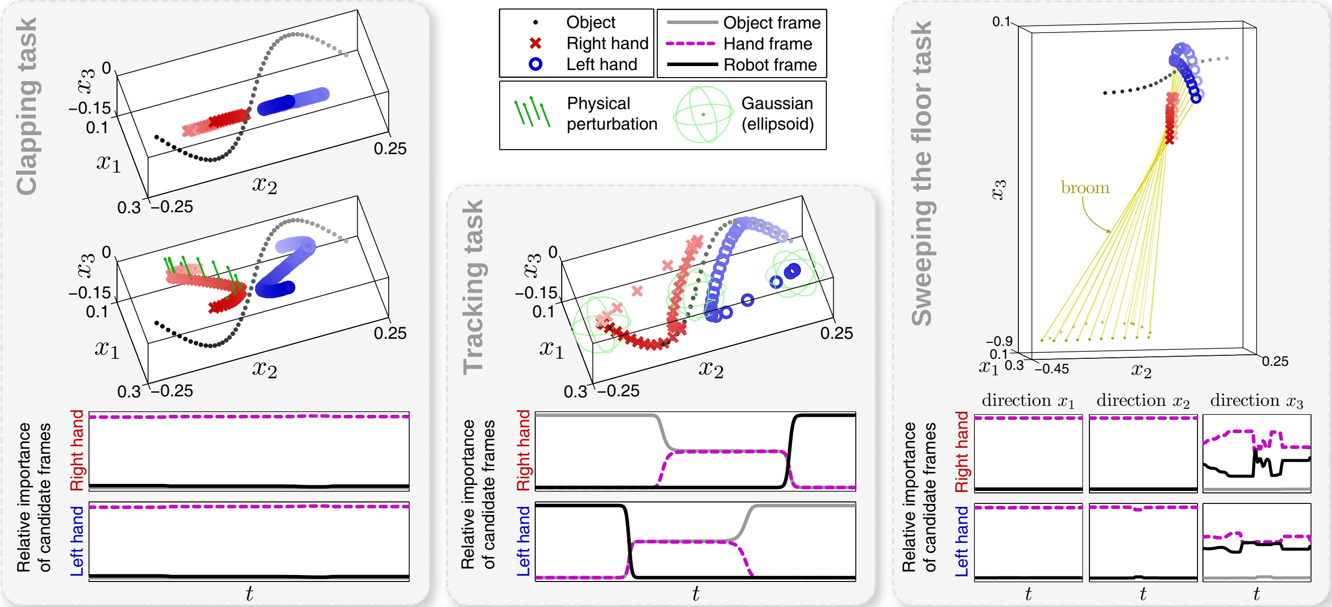


Fig. 5. Results for the *clapping*, *tracking* and *sweeping the floor* tasks. The movement of the object and the robot's two end-effectors are represented as a trail of changing colors (from dark to bright). The timeline graphs show the relative importance of the different candidate frames, estimated as a single variable to facilitate its visualization (but not used in the algorithm). It is computed as the ratio of the precision matrix determinant for a given frame with respect to the other frames ($\frac{h_m|\hat{\boldsymbol{\Sigma}}_m^{\boldsymbol{y}\,-1}|}{\sum_i h_i|\hat{\boldsymbol{\Sigma}}_i^{\boldsymbol{y}\,-1}|}$).

delegating the tracking task to the left hand.

Fig. 5-*right* presents the results of the *sweeping the floor* task. Here, the timeline graphs have been separated in different directions to show what the system learned. The movements in the demonstrations showed more variability in the horizontal plane than in the vertical direction. Indeed, the aim of the task is to sweep the floor, which can be done at several places on the floor, but the broom needs to touch the floor (consistent movement in the vertical direction). The system correctly extracts that the movement of the two hands requires bimanual coordination, and that the task can be generalized to different positions in the robot's frame, as long as vertical constraints are satisfied. Namely, the robot's frame is detected as being relevant only for the vertical direction.[4] Similarly to the clapping task, the position of the box is correctly detected as being irrelevant for the task.

Fig. 6 shows additional results for the clapping task where the user perturbs the robot by grasping one hand and moving it sideways. The robot reacts to the perturbation by reconfiguring its legs and torso to keep balance.

IV. DISCUSSION AND FUTURE WORK

The proposed DS-GMR model opens roads for new developments combining the versatility of dynamical systems and the robustness of statistical approaches. It can be applied to *dynamic movement primitives* (DMP) [5], either by considering the force components in DMP as virtual springs, similarly to [1], [6], [15], or by keeping the original formulation.

The activation weights mechanism in DMP is characterized by a second dynamical system $\tau\dot{s} = -\alpha s$ acting as a decay term, which could be used without further change in DS-GMR by learning the joint distribution $\mathcal{P}(s, \boldsymbol{y})$ from observation and estimating $\mathcal{P}(\boldsymbol{y}|s)$ through GMR during reproduction. A noticeable advantage of DS-GMR over the *weighted least-squares* (WLS) or *locally weighted regression* (LWR) [21] approaches standardly used to train DMP is that DS-GMR automatically adapts the span and position of the activation weights while learning the movement. Indeed, the conventional learning procedure in DMP is to predefine a set of activation functions h_i, usually in the form of Gaussian distributions equally spaced in time, and to use WLS/LWR to estimate the set of force components modulating a point-to-point movement. In DS-GMR, the system learns how to partition the activation weights h_i together with the search of force components. This could have potential advantages for automatically adapting the transition smoothness between the force components, and for encoding movements of varying complexity. An interesting point here is that the core DMP representation does not need to be modified: only the learning mechanism changes by reframing the problem into a mixture model.

It can be noted that the estimation of the virtual attractors in DS-GMR follows a model of degree 1 (non-linear combination of linear output-input components in GMR) instead of degree 0 (non-linear combination of constant

[4] Since the robot does not make walking steps in this experiment, the fixed world's frame and the robot's frame remain mostly aligned.

output components in WLS/LWR).[5]

One important characteristic is that the GMR process does not only provide a single estimate for each virtual attractor, but a Gaussian with full covariance. This can be exploited in future work: 1) to provide additional information when several demonstrations are available; 2) to encapsulate the local relationships between the variables of the task; 3) to regenerate movements with a natural variability that follows the essential characteristics of the task (also important for stochastic exploration issue); 4) to extend the approach to work in machine learning compatible with GMM representations. This last point opens up a host of new possibilities, with extensions towards the use of *hidden Markov model* (HMM) to encode transition probabilities [3], or the use of *Dirichlet processes* (infinite GMM) to automatically estimate the number of virtual springs required to imitate a movement [22]. A potential plan for future work is to study the extension of the proposed task-parameterized GMR approach to the *stable estimator of dynamical systems* (SEDS) proposed by Khansari-Zadeh and Billard in [7]. SEDS relies on the same GMR structure, but improves the EM learning strategy by incorporating stability constraints in the likelihood optimization.

V. CONCLUSION

We have shown that the representation of gestures in humanoids can benefit from the joint use of dynamical systems and statistics, by employing a *Gaussian mixture regression* approach encoding the position of virtual spring-damper systems in a set of candidate frames of reference. We discussed the connections and novelties with respect to existing models (including our previous work). The approach was presented with easily interpretable (but fairly simple) movements for analysis and illustrative purpose. Further work will concentrate on comparing the proposed model with state-of-the-art approaches in dynamical systems and probabilistic machine learning. In particular, we will implement comparisons with *parametric hidden Markov models* (PHMM) [13], in order to provide a more thorough analysis on the advantages/disadvantages of the two models for a wider range of tasks and problems. We also plan to extend the approach to tasks requiring more complex coordination patterns such as human-robot collaborative skills.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] S. Schaal, P. Mohajerian, and A. J. Ijspeert, "Dynamics systems vs. optimal control: a unifying view," *Progress in Brain Research*, vol. 165, pp. 425–445, 2007.

[6] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2009, pp. 2587–2592.

[7] 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.

[8] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Teaching by showing: Generating robot programs by visual observation of human performance," in *Proc. Intl Symposium of Industrial Robots*, Tokyo, Japan, October 1989, pp. 119–126.

[9] M. Riley, A. Ude, C. Atkeson, and G. Cheng, "Coaching: An approach to efficiently and intuitively create humanoid robot behaviors," in *Proc. IEEE-RAS Intl Conf. on Humanoid Robots (Humanoids)*, Genova, Italy, December 2006, pp. 567–574.

[10] 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.

[11] 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.

[12] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.

[13] 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.

[14] 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.

[15] S. Calinon, F. D'halluin, D. G. Caldwell, and A. G. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Proc. IEEE-RAS Intl Conf. on Humanoid Robots (Humanoids)*, Paris, France, 2009, pp. 582–588.

[16] 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. Morgan Kaufmann Publishers, Inc., 1994, pp. 120–127.

[17] N. G. Tsagarakis, Z. Li, J. Saglia, and D. G. Caldwell, "The design of the lower body of the compliant humanoid robot cCub," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2011, pp. 2035 –2040.

[18] Z. Li, B. Vanderborght, N. G. Tsagarakis, L. Colasanto, and D. G. Caldwell, "Stabilization for the compliant humanoid robot COMAN exploiting intrinsic and controlled compliance," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Minnesota, USA, 2012, pp. 2000–2006.

[19] E. Gribovskaya and A. Billard, "Combining dynamical systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot," in *Proc. ACM/IEEE Intl Conf. on Human-Robot Interaction (HRI)*, 2008.

[20] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 3955–3960.

[21] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 75–113, 1997.

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

[5] Namely, the non-linear force follows $\hat{\boldsymbol{f}} = \sum_i h_i(s)\ [\boldsymbol{f}_{1,i} s + \boldsymbol{f}_{0,i}]$ instead of $\hat{\boldsymbol{f}} = \sum_i h_i(s) \boldsymbol{f}_{0,i}$.