Bilateral Physical Interaction with a Robot Manipulator through a Weighted Combination of Flow Fields

Antonio Pistillo, Sylvain Calinon and Darwin G. Caldwell

***Abstract*— When collaboration between human users and robots involves physical interaction, the importance of the safety issue arises. We propose a method to transfer to robots several tasks demonstrated by the user through kinesthetic teaching and subsequently learned using a weighted combination of dynamical systems (DS). The approach used to encode the desired skills ensures a safe robot behavior during the task reproduction, allowing physical interaction with the user who can employ the manipulator as a tangible interface. By using a force sensor-less impedance controller with a back-drivable robot, this concept is exploited in two physical human-robot interaction (pHRI) scenarios. The first considers an emergency situation in which the user can stop or pause a task execution by grasping and moving the robot away from the region of space associated to the skill. The second studies the possibility to select one among several learned tasks and switch to its execution by physically guiding the robot towards the task region.**

I. Introduction

The consideration of robots as both manipulators and actuated interfaces offers new perspective in human-robot interaction, human-centered robotics and ubiquitous computing. Such actuated interfaces can have many roles and will require to merge expertise from various fields of research such as robot control, haptics and interaction design, whose respective research advances tend to follow separated tracks. Haptic interfaces are often considered as input devices and robots are traditionally viewed as actuators, but in terms of hardware capabilities, the frontier progressively disappears. On the one hand, haptic interfaces become stronger, their workspace get larger, and their passive degrees of freedom (DOFs) get progressively replaced by actuated DOFs, providing new movement/recording capabilities. On the other hand, the recent commercialization of back-drivable, actively compliant and gravity-compensated redundant manipulators provides new capabilities in terms of physical interaction.

This work goes towards exploiting these new hardware and software capabilities by stressing the issues arising when the robot has to operate in human environments and interact with non-professional users. The robot will be viewed as a tangible platform that provides both input and output capabilities. Instead of considering separated interfaces to start/stop a task, trigger an emergency signal, or select a task among a set of learned ones, we take the perspective that the most straightforward and intuitive communication medium for such human-robot collaboration is to transmit the information directly through contact with the robot. It is also the fastest way to convey information, similarly to interaction between human peers. For example, the best way to prevent a child from touching a hot plate is to grasp and move his/her arm away, which can be in some situations faster, safer and more intuitive than using spoken words. Human-robot collaborative tasks also limit the user to use or look at external interfaces such as teach pendants or screen interfaces.

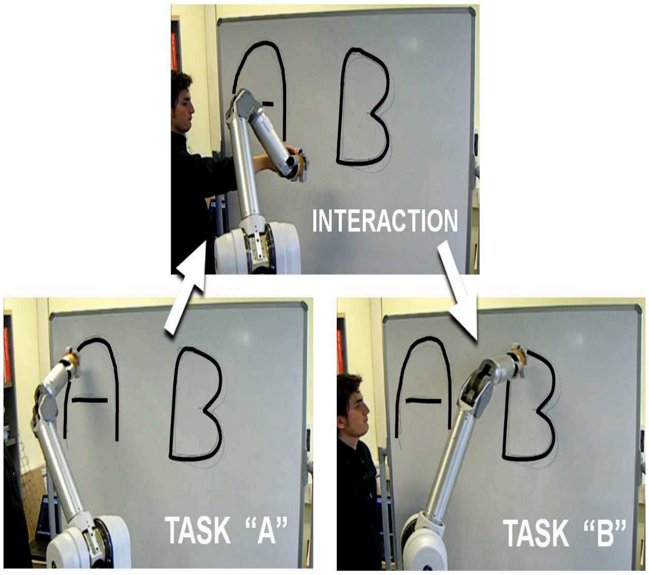


Fig. 1. The experimental set-up: a 7 revolute d.o.f. Barrett WAM arm employed to draw several alphabetical characters on a writing board. The user can stop the execution of a given task (letter ”A”) and start a different one (letter ”B”) by grasping the robot and physically guiding it towards the regions of interest.

To guarantee safety, vision techniques have been explored to detect and react to human behaviors, but this task becomes very difficult for current state-of-the-art real-time tracking systems when complex occlusions or atypical postures occur. This is the cases of collaborative working situations, where the user is close or in contact with the robot. A possible way to cope with the safety issue is to use external systems that limit the movements leading to reach the workspace, velocity or other limits of the robot, or to use attractive or repulsive fields [1]–[5]. Our approach allows to avoid the use of such additional systems.

We assume that the desired tasks to be learnt and accomplished by the robot can be represented as a set of movements of its tip in the Cartesian space. The illustrative task presented in this work consists in learning how to draw

The authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. {antonio.pistillo, sylvain.calinon, darwin.caldwell}@iit.it.

the alphabetic characters ”A” and ”B” on a board, see Fig. 1. By using back-drivable and gravity-compensated lightweight arms, such tasks can be easily transferred through kinesthetic teaching.

Then, during task execution, the user is given the possibility to decide when to start, stop, pause, resume and select the execution of a particular skill by physically guiding the robot towards or away from the region of the desired letter.

As the way skills and movements are represented has big relevance, research in robot learning by imitation and exploration has recently expressed a strong interest towards the use of compact and flexible models of continuous motion based on a superposition of elementary motion elements. The inspiration behind these works comes from various research areas such as machine learning, control theory and biology. In machine learning, the idea of approximating a non-linear function through a mixture of simpler linear elements has led to systems based on local Gaussian representation [1], [6] or based on several layers of dynamical systems [7].

In this work a task is described as a weighted superposition of basis motion fields or local flow fields, see e.g. [1], [2], [7]–[9]. The main contribution consists in a method used to modulate locally the intensity of each flow field by using a suitable weight rescaling mechanism. As result, motion commands will be present only in the area of the robot workspace covered by the demonstrations and smoothly fade outside these. Then the user can physically guide the robot inside or outside such regions as desired. We achieve this by modifying the standard Gaussian Mixture Model (GMM) weighting scheme [9], [10] depicted in Fig. 2, to make each weight independent from the others. Namely the activation function is defined by using the parameters of the k-th Gaussian component only. This helps the system to cope with the problems arising when task reproduction is required in undiscovered regions. Since the training data-set is small, the generalization capabilities of the motion primitives, in terms of position and speed, can ensure good performance only in the regions of the demonstrations.

Each local flow field is represented by an autonomous system (not indexed by time) based on a superposition of affine subsystems that creates a motion flow field guiding the robot movements. The k-th affine subsystem is defined as $\dot{\mathbf{x}} = \mathbf{A}_k\,\mathbf{x} + \mathbf{b}_k$, with $\mathbf{x}$ and $\dot{\mathbf{x}}$ indicating, respectively, the Cartesian position and velocity.[1]

II. Related Work and proposed approach

Early works in robot learning by imitation considered imitation as a two-steps process, where an expert user first provided demonstrations (sometimes in a different environment than the robot's actual location) by wearing visual markers, exoskeletons or data-gloves. The collected data were then processed off-line to build an appropriate mapping for the reproduction of the skill on the robot, see [12] for a review of several Programming by Demonstraion approaches. The work in [9] explores the possibility to encode tasks as globally asymptotically stable DS, with the approach called Stable Estimator of DS (SEDS). In [13] the method is adapted to optimize separately the direction and intensity of the learned flow field, in order to reach a target with a desired velocity.

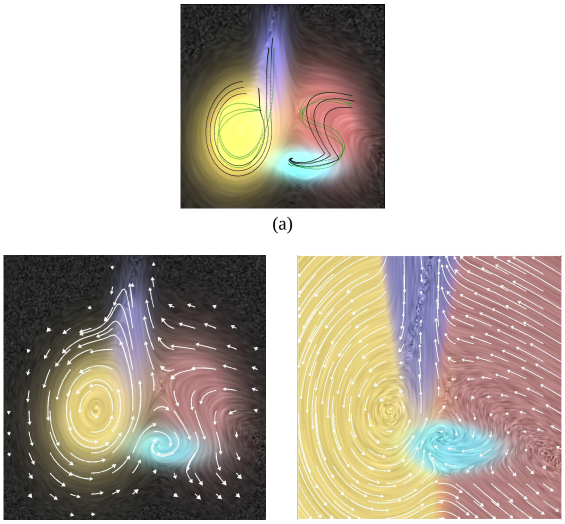


Fig. 2. Movements encoding with different weighting schemes to combine a set of basis motion fields. (a): demonstrations of the task of drawing the letters ”d” and ”s”; (b): reproduction of the tasks using weights computed with the proposed method; (c): reproduction of the same tasks with a standard GMM weighting mechanism. Color intensity represents the weight amplitude and arrows indicate direction and intensity of the velocity field.

Several rescaling techniques have been proposed to modulate the contribution of each subsystem to the overall robot motion. Usually the contribution of each motion primitive is suitably modulated in order to adapt the motion to goal variations and collision avoidance [2], along with suppression of unwanted movements in the null space [4] or avoidance of joint limits reaching [3].

These works consider scenarios in which goal modifications arise during the movements execution. Such variations have to be explicitly described either before each reproduction or should be directly measured. In both cases a suitable signal has to be specifically generated to detect and react to the perturbation. Another way to cope with safety in pHRI is to consider interaction control techniques, as in [14]–[16], along with pre-collision strategies [17].

In this paper we focus on active safety. As result to the perturbations introduced in the robotic system by the user, the robot should keep the execution of stable motions and show a compliant behavior, letting him/her move easily the tip.

In the following we will refer to task regions as particular portions of the task space covered by skill demonstrations and to subsystem as a particular movement primitive, describing the robot movement in the relative task region. Our method allows easy and rapid estimation of the model parameters needed to generalize the skill across the demonstrations

[1] In general, an augmented robot state can be employed, and other variables such as joint angles, or generalized forces can be considered, see e.g. [11].

and automatically extracts adequate stiffness and damping parameters from covariance information, as done in [6].

In this work we consider that goal modifications, intended as a desired start/stop of a movement or change in target motion, can occur during the execution of a task. Such modifications arise with two possible modalities. In the first case the user wishes to transit from the execution of a movement into a gravity compensation control mode (absence of motion). In the second case the execution of a different skill is desired and realized either after the transition to a state of pure gravity compensation or not.

The experiment shown in Fig. 1 presents the desired switching from the execution of task ”A” to ”B” through an intermediate pause with no movement. We propose a method that does not require the generation of a signal to detect such perturbations (user physically guiding the robot during task execution) since these, when present, automatically influence the weight of each motion primitive, as shown in Fig. 3-b. The user will then be allowed to execute this goal switching by grasping the robot and moving it towards the region of space of another desired task. In addition, any desired pause during a task execution can be realized by the user by just moving the robot to undiscovered regions of its workspace. In this case all the motion commands fade away smoothly and the robot remains still, using only gravity compensation, in the configuration where it was placed.

III. Description of the System

The proposed system iteratively estimates a desired Cartesian velocity command $\hat{\dot{\mathbf{x}}}$ with a mixture of K linear subsystems

$$\hat{\dot{\mathbf{x}}} = \sum_{k=1}^{K} h_k^*(\mathbf{x})\ (\mathbf{A}_k\,\mathbf{x} + \mathbf{b}_k), \tag{1}$$

where $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\mathbf{b}_k \in \mathbb{R}^3$; $\mathbf{A}_k \in \mathbb{R}^{3\times 3}$ and $h_k^*(\mathbf{x})$ is the scalar weight associated to the k-th motion primitive or local velocity field.

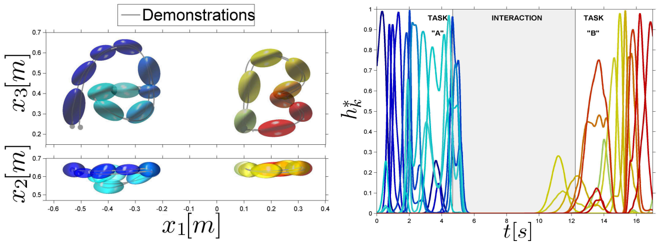
Several weighting schemes have been employed so far, including Dynamic Movement Primitives [7], [18], GMMs [9], [10] and Hidden Markov Models (HMMs) [1], [19], [20]. In the case of GMMs, the k-th mixing weight $h_k^{GMM}(\mathbf{x}) \in [0,1]$ is defined as

$$h_k^{GMM}(\mathbf{x}) = \frac{h_k(\mathbf{x})}{\sum\limits_{i=1}^{K} h_i(\mathbf{x})} = \frac{\mathcal{N}(\mathbf{x};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum\limits_{i=1}^{K} \mathcal{N}(\mathbf{x};\ \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}, \tag{2}$$

where the scalar term $\mathcal{N}(\mathbf{x};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ indicates the likelihood, at point $\mathbf{x}$, of the k-th d-dimensional multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$,

$$h_k(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}}\ \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\top}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right). \tag{3}$$

The use of the weights in (2) is not always advantageous in portions of space that have not been covered by the demonstrations. In fact, such a formulation forces at least one of the primitives to be active anywhere in the robot’s workspace. This may result in the presence of undesired movements outside the task regions, where, in addition, high velocity may arise.



(a) (b)

Fig. 3. (a): Demonstrations of the desired skills. The gray dots indicate the starting points of the trajectories (above: front view; bottom: top view). The parameters of a GMM with 18 components; their color can be compared with the color of the weights in the figure (b) on the right. (b): Evolution of the weights used to combine different motion primitives during the experiment using the proposed method. When the user moves the tip away from the task region (during the interaction phase, blue line) and releases it, all the weight have a null value causing the absence of motion commands to the robot end-effector.

The weight definition $h_k^*(\mathbf{x}) \in [0,1]$ that we suggest in this paper differs from the standard one used for GMMs:

$$h_k^*(\mathbf{x}) = \frac{h_k^{'}(\mathbf{x})}{h_k^{max}} = \frac{\mathcal{N}(\mathbf{x};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^*)}{\mathcal{N}(\boldsymbol{\mu}_k;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^*)}, \quad \boldsymbol{\Sigma}_k^* = c\,\boldsymbol{\Sigma}_k. \tag{4}$$

It makes each weight independent from the others, unlike the formulation in (2). A visual representation of the GMM learned from the collected N data points is depicted in Fig. 3-a. The constant scalar rescaling factor h_k^{max} corresponds to the maximum value of the distribution, computed at the mean $\boldsymbol{\mu}_k$. It is important to notice that, as defined in (4), the rescaling factors h_k^{max} depend only upon the parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, which have been learnt previously from the demonstrations. Therefore it is possible to obtain the rescaling parameters h_k^{max} without the need of additional parameters to be learnt.

The difference between the two approaches is depicted in Fig. 2 with an illustrative example of a movement to draw the alphabet letters ”d” and ”s”. The mechanism employed to switch across the different parts of the letters influences the capability of the robot to learn and reproduce the desired tasks while keeping a safe behavior during its employment. Computing the weights as proposed in (4) can ensure safety by keeping good generalization performances in the task regions and by fading the motion outside these.

The proposed technique forces all weights $h_k^*(\mathbf{x})$ to have a value that approaches to zero when the robot moves towards regions not belonging to the learned task, as shown in Fig. 3-b. Thus, there will be no movement if the reproduction starts in or reaches undiscovered areas.This happens when the numerator $h_k(\mathbf{x})$ in (4) approaches to a null value according to an exponential mapping of the distance between the vectors $\mathbf{x}$ and $\boldsymbol{\mu}_k$. The weights $h_k^*(\mathbf{x})$ approaches the value of one when the tip position is close to the centers $\boldsymbol{\mu}_k$. In such a case, by assuming that the different Gaussian components have a limited overlapping, only the k-th motion primitive will contribute to the motion command.[2]

The regularization process for the variance $\boldsymbol{\Sigma}_k^*$ in (4) achieves a suitable overlapping of the Gaussian components by using the factor c, which is set to minimize the quantity

$$L^* = \sum_{i=1}^{N}\sum_{k=1}^{K}\left[h_k^*(\mathbf{x}(i)) - 1\right]^2. \qquad (5)$$

This criterion tries to retrieve the original property of the GMM weighting scheme, $\sum_{k=1}^{K} h_k^*(\mathbf{x}) = 1$, by providing independent activation weights.

It is important to notice that the patterns of the weights $h_k^*(\mathbf{x})$ and $h_k^{GMM}(\mathbf{x})$, used respectively at run-time and during the GMM training phase, are different. In fact, $h_k^*(\mathbf{x})$ reaches its maximum value of one only in correspondence of the mean $\boldsymbol{\mu}_k$, while $h_k^{GMM}(\mathbf{x})$ keeps a unitary value for a given duration. This would result in an undesired temporary reduction of the motion command amplitude during the usage of the robot. In addition, during the switching phase between two components, both the associated weights $h_k^*(\mathbf{x}(t))$ and their sum $\sum_{k=1}^{K} h_k^*(\mathbf{x}(t))$ may have values being much lower than one. This would result in a unwanted temporary decrease of the motion command amplitude.

By using (1) and (4), an acceleration command $\hat{\ddot{\mathbf{x}}}$

$$\hat{\ddot{\mathbf{x}}} = \sum_{k=1}^{K} h_k^*(\mathbf{x})\Big(-(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}\big(\mathbf{A}_k\,\mathbf{x}+\mathbf{b}_k\big)\dot{\mathbf{x}}+\mathbf{A}_k\,\dot{\mathbf{x}}\Big) \qquad (6)$$

is used to drive the robot. The analytical derivation of the desired velocity and acceleration allows the definition of smooth trajectories to be used by the impedance controller described in the next section. The desired position trajectory is computed by numerical integration of the velocity and acceleration commands

$$\hat{\mathbf{x}}(t) \approx \mathbf{x}(t-1) + \Delta_t\,\hat{\dot{\mathbf{x}}}(t-1) + \frac{1}{2}\,\Delta_t{}^2\,\hat{\ddot{\mathbf{x}}}(t-1) \qquad (7)$$

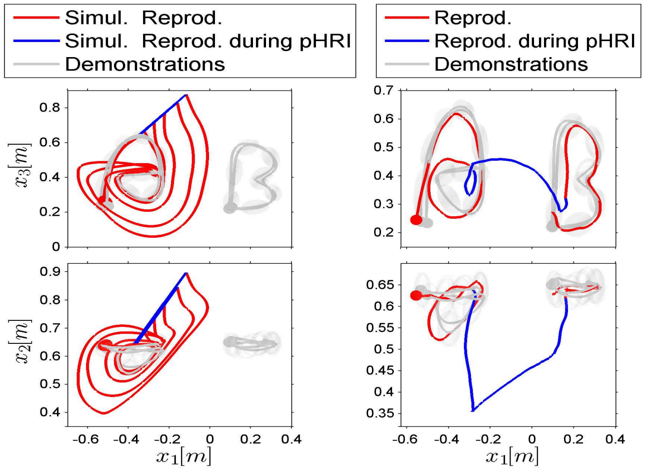
by using the constant control loop period Δ_t.

IV. EXPERIMENTS

A. Experimental set-up and control architecture

The experimental set-up used to validate the proposed approach uses the 7 revolute d.o.f. Barrett WAM arm shown in Fig. 1. The skills are represented in task space, considering only Cartesian position information. During the experiment, in both the phases of demonstration and reproduction, the control loop period is set to 10 [mS] and the orientation of the end-effector is actively kept constant. The actual displacement $\mathbf{x}$ of the tip is estimated by measuring the joint angles $\mathbf{q} \in \mathbb{R}^7$ and using forward kinematics; the actual tip velocity $\dot{\mathbf{x}}$ and acceleration $\ddot{\mathbf{x}}$ are computed by numerical derivation with suitable low-pass filtering.

Equations (7), (1) and (6) describe the computation of the desired Cartesian position, velocity and acceleration trajectories needed by a task space impedance controller. The joint-space reference model is described as



(a) (b)

Fig. 4. (a): robot tip position during a simulated experiment with standard GMM weights (above: front view; bottom: top view). (b): tip position during the experiment by using the proposed rescaling technique.

$$\mathbf{B}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\epsilon}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}, \qquad (8)$$

where $\mathbf{B},\mathbf{C} \in \mathbb{R}^{7\times 7}$ and $\mathbf{g},\boldsymbol{\epsilon},\boldsymbol{\tau},\boldsymbol{\tau}_{ext} \in \mathbb{R}^7$ account for the inertia matrix, Coriolis and centrifugal terms matrix, gravitational torques, unmodeled non linearities (static and viscous friction at joint level), motor torques and external torques. We consider the motor torque vector as the control input and the external torques generated from the physical interaction between the environment, including the user, and the robot. The required dynamic and kinematic parameters are provided with the robot.

We design an interaction controller to obtain an adjustable closed-loop dynamic behavior of the relation between the external Cartesian forces $\mathbf{F}_{ext}$ applied to the tip and its motion. The redundancy of the robot also allows interaction with its null space, which does not actually interfere with the task space tip dynamics if the proposed controller is employed.

We proceed with the derivation of the robot model described in task space coordinates, see [8] for details. Computing the analytical Jacobian $\mathbf{J}(\mathbf{q})$, with an Euler angles minimal representation of the tip orientation, and substituting $\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}}$ and $\boldsymbol{\tau}_{ext} = \mathbf{J}(\mathbf{q})^\top\,\mathbf{F}_{ext}$ in (8) leads to the model

$$\boldsymbol{\Lambda}(\mathbf{x})\,\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}})\,\dot{\mathbf{x}} + \mathbf{F}_{\mathbf{g}}(\mathbf{x}) = \mathbf{F}_{\boldsymbol{\tau}} + \mathbf{F}_{ext}. \qquad (9)$$

We use the control input

$$\mathbf{F}_{\boldsymbol{\tau}} = \mathbf{F}_{\mathbf{g}}(\mathbf{x}) + \boldsymbol{\Lambda}(\mathbf{x})\,\ddot{\mathbf{x}}_d + \boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}})\,\dot{\mathbf{x}}_d - \mathbf{K}_p\,\tilde{\mathbf{x}} - \mathbf{D}_d\,\dot{\tilde{\mathbf{x}}}, \qquad (10)$$

which is mapped in joint torques as

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{J}(\mathbf{q})^\top\,\mathbf{F}_{\boldsymbol{\tau}} = \mathbf{g}(\mathbf{q}) + \\ &+ \mathbf{J}(\mathbf{q})^\top\left(\boldsymbol{\Lambda}_d\,\ddot{\mathbf{x}}_d + \boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}})\,\dot{\mathbf{x}}_d - \mathbf{K}_p\,\tilde{\mathbf{x}} - \mathbf{D}_d\,\dot{\tilde{\mathbf{x}}}\right).\end{aligned} \qquad (11)$$

[2] Note that we can also regulate this overlapping by regularizing the full covariance matrices and/or the values in their diagonal by a multiplication factor .

The desired inertia matrix $\mathbf{\Lambda}_d$ is chosen to be equal to the robot inertia $\mathbf{\Lambda}(\mathbf{x})$, in order to avoid the use of external forces sensor, which leads to the desired closed loop behavior

$$\mathbf{\Lambda}(\mathbf{x})\,\ddot{\tilde{\mathbf{x}}} + (\boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{D}_d)\,\dot{\tilde{\mathbf{x}}} + \mathbf{K}_p\,\tilde{\mathbf{x}} = \mathbf{F}_{ext}. \tag{12}$$

$\mathbf{x}_d = \hat{\mathbf{x}}$, $\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x}$, $\dot{\tilde{\mathbf{x}}}$ and $\ddot{\tilde{\mathbf{x}}}$ represent, respectively, the desired position and the position, velocity and acceleration tracking errors. Note that, since the desired inertia matrix is configuration dependent, the Coriolis and centrifugal terms should also be considered in (12) to fulfill the skew symmetry property of the matrix $\dot{\mathbf{\Lambda}}(\mathbf{x}) - 2\boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}})$ and the passivity of the system in the regulation case.[3] $\mathbf{D}_d$ and $\mathbf{K}_p$ are symmetric and positive definite matrices representing the desired damping and stiffness. A constant and diagonal damping matrix is used. The stiffness matrix is chosen to be constant with values computed according to a trade-off between tracking performance and required compliance. An additional requirement is that the relation $\mathbf{K}_p\,\tilde{\mathbf{x}}$ should represent a passive mapping from the velocity error $\dot{\tilde{\mathbf{x}}}$ to the external forces $\mathbf{F}_{ext}$ in order to ensure the stability of the system in both the cases of interconnection with a passive environment and in free motion.

B. Experimental results

The experiments carried out consist of two phases. In the first phase shown in Fig. 3, several demonstrations of the two tasks to be learnt are supplied through kinesthetic teaching. These are represented by the drawing of the alphabetic characters ”A” and ”B” on a board, as shown in Fig. 1. During the demonstrations, the robot is controlled only with gravity compensation torques $\boldsymbol{\tau} = \mathbf{g}(\mathbf{q})$ and, given its high back-drivability, the user can easily move it with low effort.

At the end of this phase the GMM parameters $\{\boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{K}$ are estimated by Expectation-Maximization method [2], [5], see Fig. 3-a, showing the learnt 18 Gaussian components.[4] The computation of the motion primitives' parameters $\{\mathbf{A}_k, \mathbf{b}_k\}_{k=1}^{K}$ is carried out for each component using a weighted least-squares method. This considers the position and velocity data collected during the demonstrations and the weight computed off-line according to (4) and relative to the same positional data.

In the reproduction phase, the impedance controller is active to perform trajectory tracking with and without external forces applied to the tip.

Fig.4-a shows a simulated experiment with the standard GMM formulation described in (3) and (2). In this simulated scenario, the user grasps the tip, moves it far away from the task region (with a non-negligible effort, since a motion command is always present). The learned skill is reproduced in undiscovered regions, which may lead to undesired motion, possibly reaching the workspace and velocity limits. Because of this risk, we preferred not to carry out any experiment with the real robot for safety reasons by using this weighting scheme.

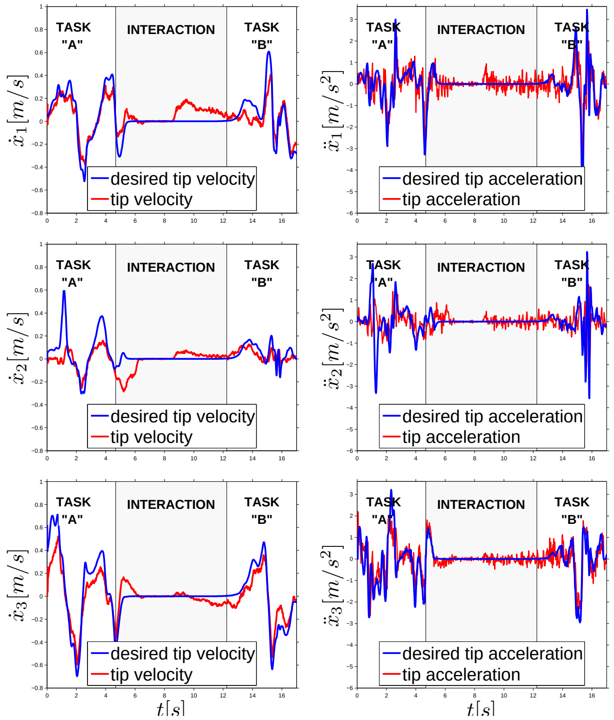


Fig. 5. Plot of the desired and actual trajectories of the end-effector velocity (left) and acceleration (right). During the interaction phase, the robot can either stand still or be moved by the user; in both the cases the motion commands tend to fade and all the control torques, except the ones needed for gravity compensation and orientation tracking, are null. The presence of motion outside the task regions is due uniquely to the forces applied by the user.

Then, we present experimental data relative to the use of the proposed method. The experiment consists in showing how moving the robot towards one of the multiple learned tasks can be used as simple mean to select which one of them to reproduce. As depicted in Fig. 4-b, the robot starts the reproduction of the skill by drawing the ”letter A” and the user interrupts the task by moving the robot away from the region of the task. The robot shows high compliance, thus allowing the user to employ low effort, while the velocity and acceleration commands are forced to smoothly fade before a complete stop occurs (when the tip is released). Fig. 3-b shows the value of the weights, that tends to zero during the interaction phase. Further data collected during the experiment is shown in Fig. 5, where the time plots of the desired and actual trajectories for position, velocity and acceleration of the robot are depicted. The difference between the actual and desired velocity during the interaction phase is caused by the user who moves the tip. When released outside the task region, the desired velocity and accelerations are null and the robot is controlled in pure gravity compensation mode.

[3] See [18] for further details and proof of closed loop stability.

[4] The number of components is set empirically, by visual inspection, in order to reach a good trade-off between the model complexity and fit of the data. A well-known drawback of Gaussian Mixture Models is that the optimal number of component may not be known beforehand. Alternative solutions exists, such as involving the use of information theory methods, e.g. the Bayesian Information Criterion (BIC) or information relative to the trajectory curvature that exploit the continuous properties of human body movements, as discussed in previous work of ours [12].

The user can then decide which particular task to perform by manually moving the robot towards the region of space relative to the other task, drawing the alphabetic character ”B”. The robot smoothly converges to the execution of such a task as shown in Fig. 4-b.

A good tracking capability, in terms of position, velocity and acceleration error, of our model-based control system is shown in the experiments in Fig. 5. This suggests that our knowledge of the dynamic model of the robot is consistent with the requirements of our application and there is no pressing need for a robust controller accounting for model uncertainty.

A video of the experiments accompanies this paper and is also available at `http://programming-by-demonstration.org`.

V. CONCLUSIONS

We proposed an approach to ensure safe pHRI during the reproduction of several learned tasks by using the robot as a bilateral tangible interface. Our perspective is to view the robot not only as an output device reproducing a learned skill, but also as an input device allowing interactions with the user during the execution of a task, with the aim to influence the robot's actions.

We showed through experiments that reformulating the weights in GMMs as being independent could be helpful in pHRI scenarios.

In the first scenario the user can stop or pause a task, by pushing or pulling the robot arm away from the relative region. In this case, the motion command applied to the tip fades as the robot is moved away, with a resulting compliant behavior. The second example shows that the user can select the task to be executed by physically guiding the towards the relative region, thus providing a simple and intuitive mean for the user to modify and control the robot behavior in a safe manner and without relying on any external interface.

In future works, the proposed method can be applied in more complex pHRI scenarios if force measures are available. This would allow encapsulating forces in the task description and could contribute to rescaling the weights for the motion/force primitives. We plan in future work to consider incremental kinesthetic teaching scenarios, such as in [21], to allow the user to gradually refine a learned skill without interrupting its execution. During the reproduction, the user would have the opportunity to grasp one part of the robot and locally modify the learned movement. After releasing the arm, the robot would then continue the execution of the skill by updating its model of the movement accordingly. Such interaction would allow the user to scaffold the robot's learning of the skill in an intuitive and efficient manner.

The operational space formulation proposed in [8] can be adopted to improve the desired tracking performances as well as to employ robot motion in its null space in order to avoid undesired collisions with the user or the environment.

In addition, strategies for the avoidance of any unwanted collision during the interaction phase can be considered, as done in [17], if tracking of the user position is available.

REFERENCES

[1] 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, 2010.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Kobe, Japan, May 2009.

[3] I. Iosiffidis and G. Schoner, "Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for a redundant robot arm," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[4] H. Reinmann, I. Iosiffidis, and G. Schoner, "Generating collision free reaching movements for redundant manipulators using dynamical systems," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[5] F. A. Mussa-Ivaldi, "From basis functions to basis fields: vector field approximation from sparse data," *Biological Cybernetics*, vol. 67, no. 6, pp. 479–489, 1992.

[6] 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.

[7] 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.

[8] O. Khatib, "A unified approach for motion and force control of robot manipulator: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[9] M. Khansari and A. G. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in *Proc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010, pp. 2676–2683.

[10] 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.

[11] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Proc. Intl Conf. on Advanced Robotics (ICAR)*, Munich, Germany, June 2009, pp. 1–6.

[12] 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.

[13] K. Kronander, S. M. Khansari-Zadeh, and A. Billard, "Learning to control planar hitting motions in a minigolf-like task," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[14] A. Bicchi and G. Tonietti, "Fast and soft arm tactics: Dealing with the safety- performance tradeoff in robots arm design and control," *IEEE Robotics and Automation Magazine*, vol. 11, no. 2, pp. 22–33, 2004.

[15] R. Schiavi and A. Bicchi, "Integration of active and passive compliance control for safe human-robot coexistence," *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pp. 1339–1345, 2009.

[16] A. De Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, September 2008.

[17] D. Kulić and E. Croft., "Pre-collision safety strategies for human-robot interaction," *Auton. Robots*, vol. 22, no. 2, pp. 149–164, 2007.

[18] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *Proc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS)*, 2001, pp. 752–757.

[19] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *Proc. IEEE-RAS Intl Conf. on Humanoid Robots (Humanoids)*, Genova, Italy, December 2006, pp. 40–47.

[20] S. Calinon, A. Pistillo, and D. G. Caldwell, "Encoding the time and space constraints of a task in explicit-duration hidden Markov model," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September 2011.

[21] D. Lee and C. Ott, "Incremental motion primitive learning by physical coaching using impedance control," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.