2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Simultaneous Kinesthetic Teaching of Positional and Force Requirements for Sequential In-Contact Tasks

Franz Steinmetz[1] and Alberto Montebelli[2] and Ville Kyrki[3]

***Abstract*— This paper demonstrates a method for simultaneous transfer of positional and force requirements for in-contact tasks from a human instructor to a robotic arm through kinesthetic teaching. This is achieved by a specific use of the sensory configuration, where a force/torque sensor is mounted between the tool and the flange of a robotic arm endowed with integrated torque sensors at each joint. The human demonstration is modeled using Dynamic Movement Primitives. Following human demonstration, the robot arm is provided with the capacity to perform sequential in-contact tasks, for example writing on a notepad a previously demonstrated sequence of characters. During the reenactment of the task, the system is not only able to imitate and generalize from demonstrated trajectories, but also from their associated force profiles. In fact, the implemented framework is extended to successfully recover from perturbations of the trajectory during reenactment and to cope with dynamic environments.**

I. INTRODUCTION

Over the last few years, several companies are releasing or undergoing intensive development of robotic arms endowed with integrated force/torque sensing at each joint. As well documented in robotics literature (see e. g. [1]), the availability of accurate joint torque measurements has huge implications, extending in significant ways the traditional scope of robotics. In particular, such an extension relates to the capacity to estimate magnitude and direction of external forces presented at the end effector of the robot and, conversely, to deliver desired forces. The specific level of accuracy in the measurement and production of external forces opens to the possibility to engage in tasks where the distribution of force requirements in space and time across a physical interface is the cause rather than the effect of positional specifications (*in-contact tasks*).

However, this novel capacity also offers new challenges, as the classical approach to robot programming falls short of the requirements when it comes to complex tasks whose declarative description can be hard to achieve. Significant time for software development, directly mapped into increased costs, is a metric that crucially highlights the issue at stake.

The *Programming by Demonstration* paradigm (hereafter PbD) offers a set of mathematical tools to directly remap a physical skill demonstrated by a human instructor into a robotic embodiment for automatic reenactment [2], [3]. A trajectory-level PbD approach is essentially constitutes of (i) a training modality that transfers the information about the required movement and (ii) a mathematical model that represents the demonstrated task. Traditionally, training modalities make often use of tele-operation, vision [4] and data gloves [5]. More recently, *kinesthetic teaching* is emerging as a particularly intuitive and natural method to the human user. In this case, the human expert demonstrates the task by directly grabbing the robot body and moving it to execute the task [6].

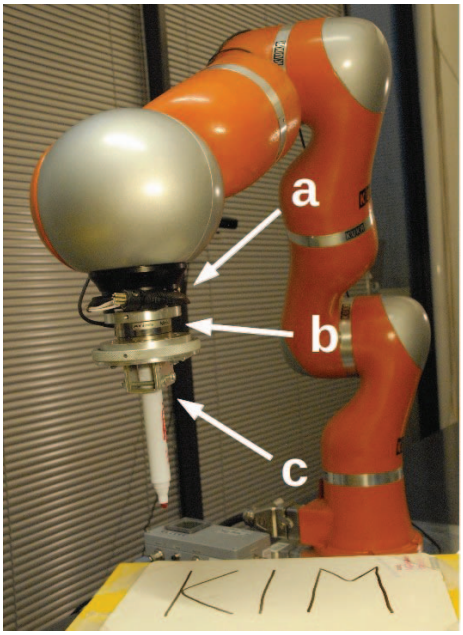

Fig. 1. Detail of the experimental setup: robot's flange (a), external force/torque sensor (b) and tool (c).

Several models capable of capturing the demonstrated task at the trajectory-level have been presented in the literature. State of the art PbD methods do not necessarily aim for high-fidelity replication of the demonstrated task. Rather, they can make use of a single or multiple demonstrations to capture basic features of the task. The automatic reenactment of the demonstration will reproduce such features, yet crucially including the capacity for generalization and the potential for coping with contingent perturbations during the execution. Schematically, we can separate state of the art trajectory-level models into two major families. On one hand, statistical models based on regression extract the main statistical

[1]Franz Steinmetz worked on the experiments described in this paper as part of his Master's Thesis at Aalto University. He is now with German Aerospace Center (DLR), Institute of Robotics and Mechatronics, 82234 Oberpfaffenhofen-Weßling, Germany. franz.steinmetz@dlr.de

[2]Alberto Montebelli (corresponding author) is with the School of Informatics, Univeristy of Skövde, SE-54128 Skövde, Sweden. alberto.montebelli@his.se

[3]Ville Kyrki is with the Department of Electrical Engineering and Automation, Aalto Univeristy, 00076 Aalto, Finland. ville.kyrki@aalto.fi

978-1-4799-6885-5/15/$31.00 ©2015 IEEE

features of the demonstration and use that information to reproduce the task [6], [7], [8]. On the other hand, dynamic models try to capture in their equations the salient dynamics of the task (e.g. see [9], [10], [11], [12], [13]).

Overall, the PbD paradigm can be extended to in-contact tasks. An example is given in [12], which uses a hybrid form of kinesthetic teaching, operatively split in two phases. First, the demonstrator teaches the robot the trajectory appropriate to the task. Second, the demonstrator creates and overlaps to this taught trajectory a force profile that is obtained using an external haptic device. Related to this is [14], where the human demonstrator directly handles a tool for plastiline sculpting.

However, in the experiments reported in the present paper, the demonstrator grabs the robot above the face of the F/T sensor on the flange side (see accompanying video) rather than on the tool side (as in [14]). This allows for a direct measurement of the contact forces exerted on the tool during demonstration, rather than requiring an external (haptic) device. This is similar to the work presented in [15], where wood planing is demonstrated and executed relying on a simplified feedforward version of the same robot controller described in this paper. This setup is a precondition for the implementation of a programming interface for the transfer of human skills to the robot that is intuitive and natural to the human expert (i. e. the human expert can execute the task in a familiar form). Positional and force requirements remain intrinsically correlated, as during the normal execution of the task by the human.

In [16], a similar setup is used for simultaneously learning trajectory and force profile for a human-robot collaborative transportation task. Instead of using DMP, Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) is utilized for modeling and reproduction, respectively. The robot is entirely controlled with a force controller and is constantly in contact with the environment. No explicit force feedback ensures proper force exertion.

Our method is here demonstrated on a robotics arm. However, it can be in principle extended to more complex robotic platforms (e. g. humanoid robots) as long as they are provided with the capacity to apply controlled forces onto the external environment.

Although much interesting work has been published for both statistic and dynamic design principles, our experimental setup focuses on the latter. More specifically, we make use of Dynamic Movement Primitives (DMP, [9], [10]) for its flexibility, intrinsic properties of generalization and capacity to manage perturbations during reenactment, property highlighted in Section II-B.

In the remainder of this paper we will first detail our robotic setup, software infrastructure, describe the DMP mathematical model and details about our proposed controller (Section II). The following Section III introduces our experiments. In Section IV, we will present and analyze our findings. Finally, we will discuss our results within state of the art PbD for in-contact tasks, highlighting directions for future investigations (Section V).

II. MODELING AND HYBRID CONTROL OF POSITIONAL AND FORCE REQUIREMENTS

A. System Description

The minimal setup required by the proposed method consists of a robotic arm with multiple degrees of freedom and integrated torque sensors at each joint. The latter property allows programmable active compliance, gravity compensation and Cartesian impedance control. The robot's overall control cycle must be executed in the range 1–5 ms, to allow effective force control.

A six-axis F/T sensor with appropriate sensing range and accuracy is rigidly mounted between the robot's flange and tool (see Fig. 1). The tool consists of a whiteboard marker, rigidly connected through a custom aluminum adapter to the F/T sensor.

B. Dynamic Movement Primitives

The DMP model has been successfully used in robotics within the PbD paradigm. The existing literature provides a detailed treatment of the model, of its general properties and examples of applications in robotics [9], [10]. The model consists of three main elements: a *transformation system*, a *function approximator* and a *canonical system*.

DMP models are compatible with a generic state space. For example, this work makes use of an extended eight dimensional Cartesian space $\mathbf{y} = (y_x, y_y, y_z, y_{q0}, y_{q1}, y_{q2}, y_{q3}, y_{fz})^T$. The Cartesian position is defined by $\mathbf{y_{tran}} = (y_x, y_y, y_z)^T$. The orientation is represented by the four components of the quaternion $\mathbf{y_{rot}} = (y_{q0}, y_{q1}, y_{q2}, y_{q3})^T$. The force in tool z direction is given by y_{fz}.

The DMP model isolates each dimension (generically indicated y_d in this paper) of the represented space. Each dimension is independently modeled as a spring-damper system (transformation system). Restricting our attention to the case of discrete motions, relevant to this paper, the transformation system would naturally relax from its initial position (i. e. the initial value for the specific dimension during the demonstration) to its final value (goal), the steady-state point attractor for the specific dimension. This is mathematically expressed as

$$\tau \ddot{y}_d = \alpha_z(\beta_z(g_d - y_d) - \dot{y}_d) + f_d(x), \tag{1}$$

where τ is a temporal scaling factor, g_d is the goal and y_d, $\dot{y}_d$, $\ddot{y}_d$ are current value, first and second derivative of the considered dimension (either positional or force information). The positive parameters α_z and β_z are related to stiffness and damping of the transformation system. The system is critically damped for $\beta_z = \alpha_z/4$.

The term $f_d(x)$ in Eq. (1) is the nonlinear function approximator. It acts on the transformation system, controlling its asymptotic convergence. It is implemented as a normalized linear combination of basis functions, modified by the scaling term $\xi(x)$:

$$f_d(x) = \frac{\sum_{i=1}^{N_w} \Psi_i(x) w_{i,d}}{\sum_{i=1}^{N_w} \Psi_i(x)} \xi(x) \tag{2}$$

with:

$$\Psi_i(x) = \exp\left(-\frac{1}{2\sigma^2}(x-c_i)^2\right). \quad (3)$$

Therefore, $f_d(x)$ is the sum of N_w external forcing terms $\Psi_i(x)$, centered in c_i and each having limited effect in time (see below for the relation between x and time) according to a Gaussian function parametrized by σ. Each forcing term $\Psi_i(x)$ in the sequence is scaled by a weighting factor $w_{i,d}$, identified for each dimension in order to appropriately stretch/compress the spring-damper system over time, thus allowing accurate reproduction of the demonstrated trajectory (details about the identification process are in [17]).

The canonical system synchronizes all the different components (i. e. the transformation systems and function approximator of the different dimensions), while abandoning an explicit use of a time variable [9]. The canonical system is implemented as a first-order ordinary differential equation:

$$\tau\dot{x} = -\alpha_x x + C_c \quad (4)$$

Accordingly, the canonical system generates an exponential relationship, modulated by α_x, between time and the phase variable x [9], [10]. The temporal coupling term C_c allows for online control of the evolution of the canonical system (see Section II-D).

Finally, the term $\xi(x)$ in Eq. (2) is defined according to [10]:

$$\xi(x) = A_{\max} x = (\max(y_d) - \min(y_d))x \quad (5)$$

Here, $\max(y_d)$ and $\min(y_d)$ are, respectively, the maximum and minimum value of the corresponding component of the demonstration. Multiplying the amplitude $A_{\max}$ in Eq. (2) introduces spatially invariant properties of the attractor landscape, thus allowing for spatial scaling of the reenactment [10]. Due to its exponential decay, the phase variable x fades in time the forcing term in Eq. (2), thus ensuring the asymptotic stability of the transformation system.

As a final note, the four components of the quaternion, representing the orientation of the tool, were normalized during reenactment of the demonstrated task.

C. Controller

During reenactment of the demonstration, the DMP algorithm outputs the state space vector $\mathbf{y}$. In our experiments, a *Cartesian impedance controller* operates the robot. It can be formally described as:

$$\tau_{\mathbf{cmd}} = \mathbf{J}^T(\mathbf{k_c}(\mathbf{y_{des}} - \mathbf{y_{msr}}) + \mathbf{f_{des}}) + \mathbf{D}(\mathbf{d_c}) + \tau_{\mathbf{dyn}}(q, \dot{q}, \ddot{q}), \quad (6)$$

where $\mathbf{J}$ is the Jacobian matrix of the robotic arm. The controller emulates a spring-damper-system driven by $\mathbf{y_{des}}$ and $\mathbf{y_{msr}}$, respectively the desired and measured Cartesian pose (position and rotation), through programmable stiffness vector $\mathbf{k_c}$. The robot can also exert an additional force/torque $\mathbf{f_{des}}$. $\mathbf{D}$ is a damping term, depending on the normalized damping parameters $\mathbf{d_c}$. The controller utilizes a dynamic model of the arm, represented by the term $\tau_{\mathbf{dyn}}(q, \dot{q}, \ddot{q})$, which allows for compensation of dynamic forces, such as gravity and Coriolis force.

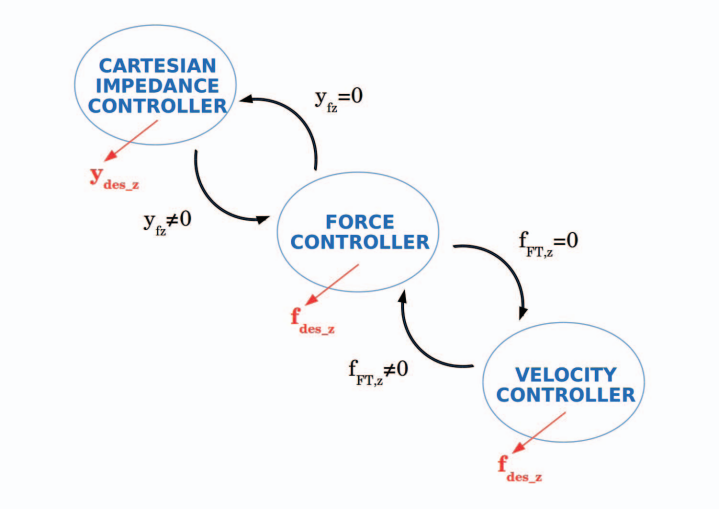

Fig. 2. State diagram of the switch robot control in the z direction. The Cartesian impedance controller commands the position in the y_z dimension, while force and velocity controllers command forces y_{fz} exerted in the same direction. The value of the force requested by the DMP model, y_{fz}, coordinates the transition between Cartesian impedance controller and force controller. The force measured by the F/T sensor, $f_{\text{FT,z}}$, determines the transition between force and velocity controllers (see text for details).

When the robot has to provide external forces, the calibration of the robot's internal dynamic model must be considered, for it can only be performed with finite precision before each execution. For example, when the robot used in our setup (KUKA LWR4+) is operated to apply controlled forces, the precision of the calibration process typically allows accuracy in the range 1–3 N. Therefore, two general situations need to be addressed: (i) When the task allows a coarse distribution of forces (for an example see [15]), such a deviation is negligible. In this case, the output of the transformation system component responsible for the force can be directly forwarded to the Cartesian impedance controller ($\mathbf{f_{des}}$ in Eq. 6). (ii) For tasks requiring fine control of forces, such as the ones described in the experimental section of this paper, the measurement error has to be compensated through a *force controller*.

The custom controller that we present in this section converts the DMP model's output into appropriate motor commands formally required in Eq. 6: $\mathbf{y_{des}} = (\mathbf{y_{tran,\,des}}, \mathbf{y_{rot,\,des}})^T$, i. e. $N_P = 7$ pose dimensions, and the desired force $\mathbf{f_{des}}$.

We propose the following *hybrid control* strategy for the robot, put on trial in the experiments described in this paper. All positional components of $\mathbf{y}$, with the exception of y_z, are permanently under position control by a Cartesian impedance controller. Since in our experiments contact forces are explicitly considered and controlled in the z direction only, the components in such a direction, y_z and y_{fz}, are subject to *switch control*, graphically represented in Fig. 2:

1) Cartesian Impedance Controller: When the application of a force is not required (i. e. the robot is in an aerial phase), the trajectory $\mathbf{y}$ generated by the transformation system is

directly forwarded to the Cartesian impedance controller ($\mathbf{y_{des}} = (\mathbf{y_{tran}}, \mathbf{y_{rot}})^T$ in Eq. 6). The desired force/torque is set to zero ($\mathbf{f_{des}} = 0$).

2) Force Controller: When, according to the demonstration, the DMP model requires a contact force y_{fz}, a PID-controller (hereafter force controller) receives as input the difference between the required force and the feedback of the external F/T sensor ($f_{\mathrm{FT,z}}$). Accordingly, the force controller generates the desired force $\mathbf{f_{des}}$:

$$e_{fz} = y_{fz} - f_{\mathrm{FT,z}} \tag{7}$$

$$g_{fz} = 1 + K_P \cdot e_{fz} + K_I \cdot \int e_{fz} dt + K_D \cdot \dot{e}_{fz} \tag{8}$$

$$\mathbf{f_{des}} = (0, 0, g_{fz} \cdot y_{fz})^T. \tag{9}$$

Operatively, the transition from Cartesian impedance to force controller can be forced by setting the z component of the stiffness $\mathbf{k_c}$ to zero whenever the DMP model is demanding a force to be applied in the same direction. This causes the z direction to be only controlled by the desired force $\mathbf{f_{des}}$. Conversely, when no force needs to be exerted, the stiffness in z direction is set back to its original value.

3) Velocity controller: In some situations, according to the demonstration the robot has to exert a force y_{fz}. However, in its current position, its tool has not yet established a contact with any physical surface ($f_{\mathrm{FT,z}} = 0$). For example, that can happen when an object has been displaced with respect to the demonstration (e. g. because the whiteboard lays on a lower surface than the one occupied during the demonstration, as in Section III-iii). Our solution was to design a further PID-controller (*velocity controller*) for approaching physical contact. The velocity controller adjusts the force in z direction to approach the object with a constant velocity v_{des}.

$$e_v = v_{des} - v_{\mathrm{msr}} \tag{10}$$

$$g_v = K_P \cdot e_v + K_I \cdot \int e_v dt + K_D \cdot \dot{e}_v \tag{11}$$

$$\mathbf{f_{des}} = (0, 0, g_v)^T \tag{12}$$

As in the previous case, when the velocity controller is active, the stiffness $\mathbf{k_c}$ in the z direction is set to zero. Therefore, the modeled positional information in the z direction plays no role in controlling the robot.

Finally, we addressed a further problem emerging when the robot is commanded to abandon the contact (i. e. force exertion is no longer required). As the z component of the stiffness $\mathbf{k_c}$ is reset, the Cartesian impedance controller might abruptly apply the elastic force due to the discrepancy between the current measured z position and the modeled z position. This discrepancy is acquired during the demonstration due to the finite stiffness that is driving the pose and stochastic internal and external friction. The robotic arm would then be commanded an undesirable swift movement between the two points. Therefore, when contact is abandoned, the offset between the current position and the desired position is calculated ($\Delta y_{\mathrm{tran}} = y_{\mathrm{tran,msr}} - y_{\mathrm{tran}}$) and added to all future pose commands: $y_{\mathrm{des}} = (y_{\mathrm{tran}} + \Delta y_{\mathrm{tran}}, y_{\mathrm{rot}})^T$.

D. Handling of perturbations

We propose a custom mechanism to manage external perturbations during reenactment, i. e. physical events causing the robot to deviate from its desired trajectory. The design of this mechanism aims for handling all N_P pose dimensions and for a simple parametrization making only use of the temporal coupling term C_c in Eq. (4).

In case of perturbations, the function of the coupling term C_c is to delay the execution, in order to allow time for the disturbance to fade and then restore the normal temporal and execution flow. For example, when $C_c = \alpha_x x$, the canonical system and thus the temporal evolution of the whole system is stopped. Smaller (positive) values reduce the execution speed.

We propose to set the value of C_c by comparing the absolute ratios of the current deviation between desired and measured pose with the maximum allowed deviation $\Delta e^j_{\max}$ for each dimension $j \in \{1, \ldots, N_P\}$:

$$\Delta e^j = y^j_{\mathrm{desired}} - y^j_{\mathrm{measured}} \tag{13}$$

$$e^j_{\mathrm{rel}} = \left| \frac{\Delta e^j}{\Delta e^j_{\max}} \right| \tag{14}$$

The highest ratio among all dimension is then limited to one:

$$e_{\mathrm{max,\,rel}} = \max(e^j_{\mathrm{rel}} \mid j \in \{1, \ldots, N_P\}) \tag{15}$$

$$e'_{\mathrm{max,\,rel}} = \min(1, e_{\mathrm{max,\,rel}}) \tag{16}$$

The perturbation handling mechanism described above is implemented by assigning the temporal coupling term as follows:

$$C_c = e'_{\mathrm{max,\,rel}} \alpha_x x, \tag{17}$$

The perturbation handling mechanism adds an additional control loop to the system. In the absence of such a mechanism, the arm would simply receive direct commands from the Cartesian impedance controller, thus disregarding the current state of the system. Conversely, the temporal evolution of the system can be modulated by comparing the current and desired pose.

III. Experiments

With the experiments presented in this section, we intend to test and analyze the feasibility of the proposed approach, its relationship to the integrated Cartesian impedance controller and its capacity for generalization.

Our setup develops around a commercial KUKA LWR 4+ robotic arm. The robot has 7 degrees of freedom, integrated torque sensors at each joint, a payload and reach of 7 kg and 800 mm, respectively. The robot is capable of programmable active compliance, torque control and gravity compensation, while its overall control cycle runs up to 1 kHz [18].

Through the KUKA *Fast Research Interface* (FRI) communication protocol, the KUKA LWR 4+ robot controller can be interfaced to a standard external computer to deliver data to and receive commands [19]. This interface has been adapted for real-time applications, allowing high frequency hard real-time control consistent with the robot's nominal

limit of 1 kHz. Our external computer runs a Xenomai 2.6.2.1 real-time patch for the 3.5.7 Linux kernel and RTnet 0.9.13 real-time networking. The development of components for *Open Robot Control Software* (Orocos), interfaced to and empowered by functionalities available under *Robot Operating System* (ROS 'Groovy'), results into a general, robust, modular and flexible robot control framework at a relatively high level of abstraction.

A six-axis ATI mini 45 F/T sensor (sensing range: ± 290 N F_x and F_y, ± 580 N F_z; ± 10 Nm for T_x, T_y and T_z; accuracy between: 0.75 % and 1.25 %, depending on the axis) has been rigidly mounted between the robot's flange and tool, providing it with thermal insulation from the robot.

During the demonstration, a human grabbed the robotic arm at its final section, above the flange side of the F/T sensor. That is a crucial detail, since this way of handling the robot allows a direct measurement of the contact forces exerted by the demonstrator. The demonstration consisted in moving the arm so that the marker traced a sequence of characters (e.g. 'KIM') on a whiteboard laying on a horizontal rigid surface. Thus, contact between the tip of the marker and the whiteboard was made and broken multiple times. During demonstration, gravity compensation mode was active, thus minimizing the demonstrator's effort in moving the system composed by the marker, the F/T sensor, adapters and the robotic arm. The robot can provide an estimate of the position and orientation of the tool during the movement. Pose data relative to a point located along the tool axis and laying on the pen-side of the F/T sensor were recorded. In parallel, data from the F/T sensor were received and stored for further analysis and identification of the model. Poses and force data were recorded in a single phase, with a frequency of 500 Hz. A single demonstration recorded as $M = 27854$ samples of state space vectors was performed. This single demonstration was used for model identification in all the experiments presented in this paper.

To allow a general evaluation of the reenactment of the demonstrated task, of the controller's robustness, of its capacity of generalization, and to physically validate the method for simultaneous modeling of the demonstrated positional and force features, four experiments are reported in this paper: (i) *basic reenactment*, the robot reenacted the demonstrated task in the absence of significant perturbations. (ii) *deactivated force controller*, the task was reenacted after isolating the custom force controller (described in Section II-C) from the control loop. Under this condition, the system relies on the commercial Cartesian impedance controller alone, comparatively emphasizing the role of the custom force controller for in-contact tasks. (iii) *lowered whiteboard*, the robot reenacted the task after the elevation of the whiteboard had been lowered with respect to the demonstration. In this case, the role of the velocity controller in approaching the whiteboard with constant velocity of 2 cm/s is emphasized together with its interaction with the force controller once the contact is established. (iv) *handling of perturbation*, the movement of the robotic arm during the aerial phase of the pen (i.e. when the pen was not in contact with the whiteboard) was suddenly stopped by human action. This final experiment highlights the performance of the perturbation handling, gracefully 'hibernating' the system during reenactment and then restoring the sequence after the interference is removed.

The following parameters were used: $N_w = 250$ for all dimensions; $\alpha_z = 2000$ and $\beta_z = \alpha_z/4$; $k_c = 2000$ N/m and 200 Nm/rad for linear and angular movements, respectively; $d_c = 0.7$ Ns/m for linear and 0.7 Nms/rad for the angular movements. The exponential decay of the canonical system started at $x_0 = 1$ with a decay rate of $\alpha_x = 1.1$. The robot was configured to exert a maximum force of 200 N and a maximum torque of 15 Nm in all directions. The hard real-time control cycle was running at 500 Hz.

IV. RESULTS

A. Basic reenactment

The first experiment displays the successful reproduction of the sequence of characters. Fig. 3 comparatively presents positional data extracted from the demonstration ('Demonstrated trajectory'), generated from the DMP model after training ('Modeled trajectory')) and during basic reenactment ('Measured trajectory'). In Fig. 4–top panels, the same data are presented over time for the three dimensions. The label 'Calculated trajectory' represents the output of the DMP model keeping into account the contingent temporal coupling term C_c (Section II-D). All the trajectories show significant qualitative similarities. The root mean square (RMS) error is 0.032 m when directly comparing the trajectories. This value reduces to 0.0085 m when temporally aligning the different trajectories by dynamic time warping [20]. Nevertheless, all spatial components show an increasing deviation from the desired trajectory, while the total duration of basic reenactment is about 3 s longer than the demonstration (see V). The average jerk, calculated in 343 m/s^3 over the demonstration, was 371 m/s^3 during reenactment.

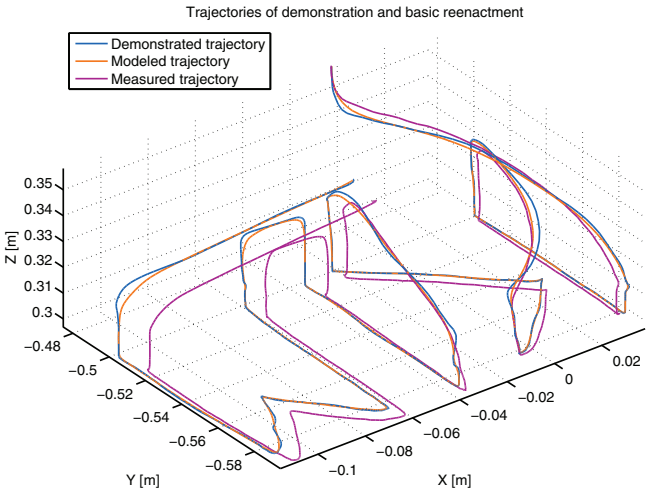

Fig. 3. Spatial comparison of the demonstrated trajectory, the trajectory generated by the DMP model and measured trajectory during basic reenactment.

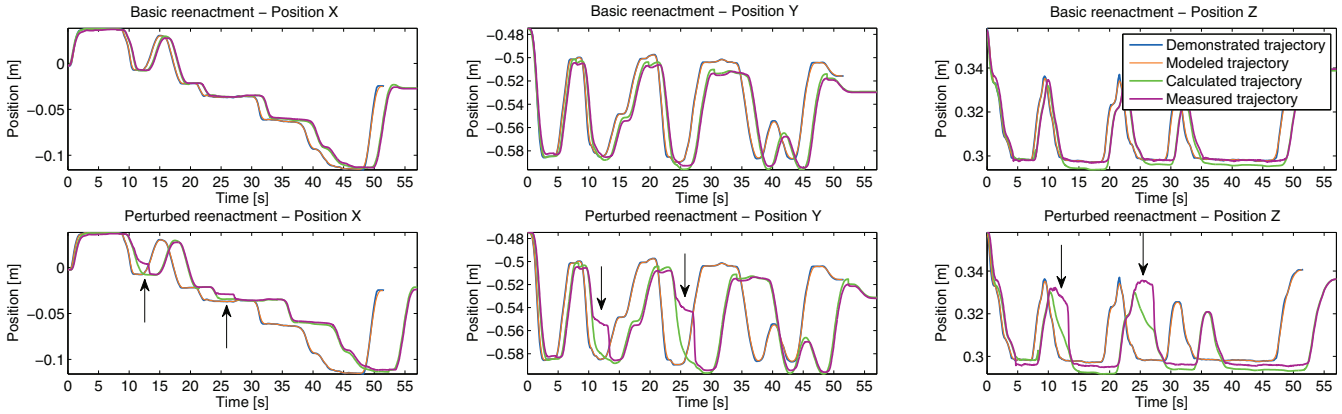

Fig. 4. Temporal evolution of the demonstrated trajectory, of the trajectory generated by the DMP model and measured trajectory during basic reenactment (top panels) and during exogenous perturbation (bottom panels – arrows mark the instants at which the robot is immobilized by a human).

The force profiles for all the experiments are shown in Fig. 5. The top panel displays force data from the demonstration, where the label 'Modeled forces' refers to the forces calculated by the learned DMP model. In the three lower panels, the first of which represents basic reenactment, the label 'Calculated forces' characterizes the output of the DMP model considering the temporal coupling term C_c. Finally, the label 'Commanded forces' refers to the output of the force controller, that is forwarded to the Cartesian impedance controller (Section II-C).

The DMP model of the force profile filters out high-frequency oscillations present in the demonstration. However, high-frequency components reappear in the forces measured during all the experiments, due to the properties of the mechanical interactions between the whiteboard and the physical system connected to the pen and, for even higher frequent components, due to the noise produced by the F/T sensor. During basic reenactment, the measured forces closely follow the calculated forces, with maximum deviation of about 1 N, proving the effectiveness of the force controller in correcting the error generated by the Cartesian impedance controller. Larger errors closely follow in time the conclusion of the aerial phase of the pen, when contact between the pen tip and the whiteboard is reestablished. However, the general RMS error is 1.3 N and 0.39 N for the raw data and the temporally aligned case, respectively.

In conclusion, trajectory and force profile are reenacted with satisfactory accuracy during basic reenactment. These observations reinforce our confidence in the method for simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks, presented in this paper.

B. Deactivated force controller

Once the force controller is isolated from the control loop, a deviation between measured and calculated forces becomes apparent (Fig. 5, third panel). The force error often reaches a magnitude larger than 3 N. Even using temporal alignment, the RMS error is 0.82 N, more than twice the result achieved with the force controller. Imperfect calibration of the dynamic model of the robotic arm, automatically executed at the beginning of each experiment, limits accurate force exertion.

Therefore, tasks calling for a fine grained distribution of forces require external force feedback measurements and a cascaded force controller.

C. Lowered whiteboard

Under this condition, the robotic arm moves downward, outreaching for the contact with the whiteboard. However, once the elevation occupied by the whiteboard during the demonstration is gained, no contact can be established. In fact, the whiteboard has been displaced to a lower elevation, thus generalizing over the demonstrated task. The velocity controller (Section II-C) is activated until contact is made.

The bottom panel in Fig. 5 demonstrates this condition in the time interval marked by the two arrows (i. e. between 4.6 s and 10.8 s). Differently from the previous experiments, no force is measured during that period, for the whiteboard has been displaced and no contact is yet made. However, the force commanded by the velocity controller causes the arm to move downwards along the current tool direction at constant speed, until contact is established (and the offset between the current and expected position can be calculated).

D. Handling of perturbation

Because the arm is controlled with an Cartesian impedance controller, a human can effortlessly interfere with the movement of the robotic arm and bring it to a stop. As a second example of generalization, this is shown in Fig. 4–bottom panels, where the temporal evolution of the position is perturbed in two occasions by a human operator during the aerial phase of the pen (marked with arrows around 13 s and 26 s). The robot's movement is transiently (and arbitrarily) stopped and also the position is slightly altered. Nevertheless, after releasing the robot, the movement is gracefully restored in all its features, including velocity. Therefore, the duration of the reenactment is consistently longer than under basic conditions.

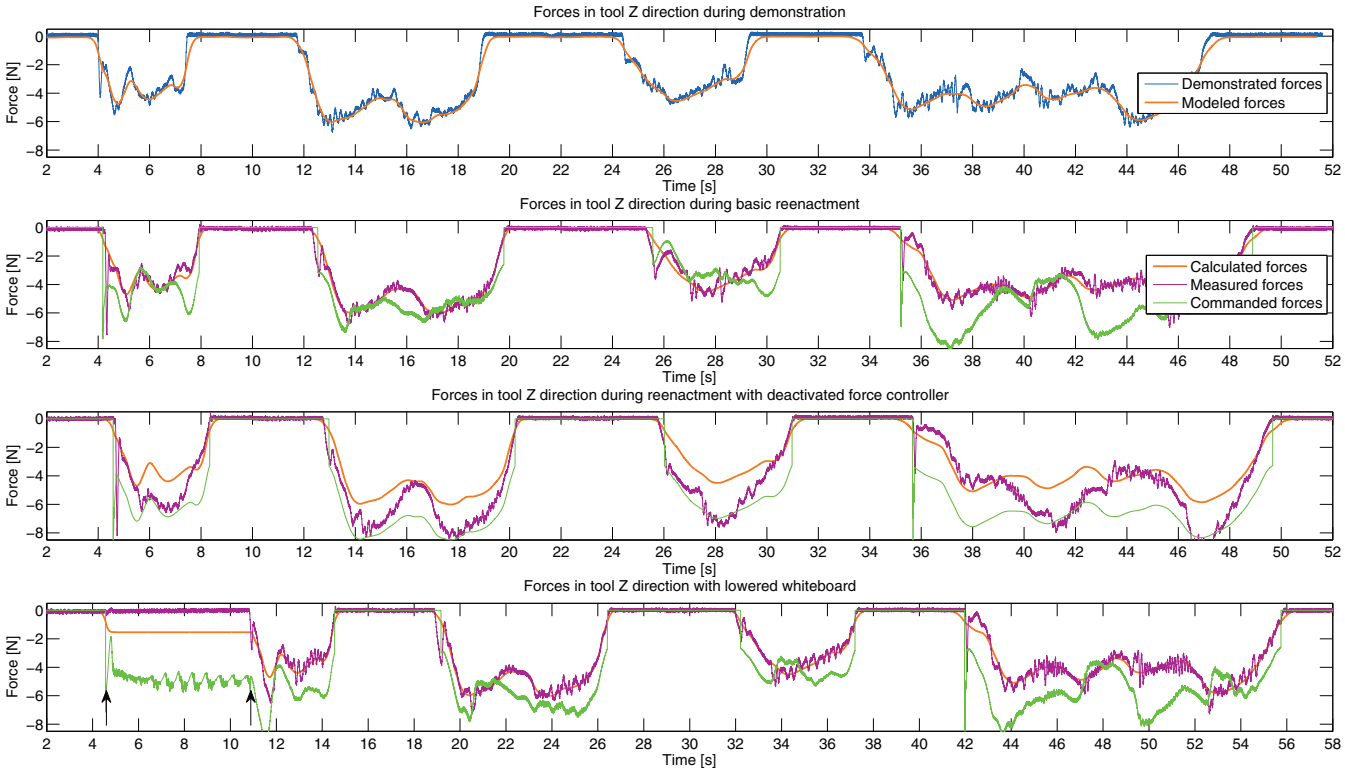

Fig. 5. Temporal evolution force profiles during demonstration (top panel), and in the different experimental conditions presented in the text. Two arrows in the bottom panel mark the time interval during which the robot, according to the demonstration, is expecting to make contact with a displaced whiteboard.

Further experiments, whose description goes beyond the scope of this paper, include reenactment over a whiteboard with fixed inclination and with dynamically changing inclination, randomly operated by a human and therefore offering poor reproducibility. The control system can cope with the two situations, under condition that the rate of change of the inclination does not exceed a certain threshold. Excessive rate of change currently leads the system to unstable behavior. [1]

V. DISCUSSION AND CONCLUSIONS

The paper presents a novel use of a common robotic configuration. An external F/T sensor is positioned between the robot's flange and its tool. By grabbing the robotic arm above the flange side of the F/T sensor, the human demonstrator allows a direct measurement of the contact force between marker and whiteboard during the demonstration. Therefore, simultaneous kinesthetic teaching of both positional (pose) and force requirements for sequential in-contact tasks becomes possible. This result extends the methods for programming by demonstration for in-contact tasks currently available in the literature. With respect to [12], the recording of pose and force data is simultaneous, rather than divided in two separate phases. Furthermore, the use of an (arbitrarily chosen) external device for the superposition of forces over the previously taught trajectory is avoided. Our implementation, developed around a state of the art robotic arm, can be considered as a first step towards the generalization of the method to more complex robotic embodiments, e. g. to humanoid robots.

Several experiments support our confidence in the proposed approach. The system is able to imitate with satisfactory accuracy the writing of a sequence of characters after human demonstration. The proposed controller proves also capable of generalization, coping with different types of perturbations of its behavior (e.g. transient immobilization of robotic arm) and environment (e.g. altered elevation, and static and dynamic changes in the inclination of the whiteboard with respect to the demonstration). We identify the condition under which the use of an external F/T sensor and the integration of a force controller in the control loop is desirable not only during the demonstration session, but in order to allow the exertion of fine grained distributions of forces. In contrast, on a related paper [15], we demonstrate the case of an in-contact task characterized by a more articulated dynamic of forces, although with no complex sequences (hand-plane on wood). In that case, the application of coarse forces suits well the task, at least as a first approximation.

However, the reenactment is not perfect and this suggests future updates in the control algorithms. The longer duration of the nominal reenactment compared to the demonstration (Fig. 4) can be explained by considering the handling of perturbations. As an impedance controller with finite stiffness

[1]The demonstration of this experiment, and of all the other experiments mentioned in this paper, can be found in the accompanying video.

is used, there is always a small offset between the set point and the current position. Such a discrepancy causes the perturbation handler to slightly delay the execution. Similarly, the use of finite stiffness together with the cumulative effect of stored positional offsets when contact is abandoned (Δy_{pos}, see Section II-C) give a reason for the increasing offset of the trajectory during reenactment (Fig. 3). Due to the geometric characterization of the presented task, only the offset in the tool direction should be considered.

The current implicit "strategy" of the perturbation handler (i. e. passively slacking and waiting for a perturbation to fade) may not be the ideal approach in every situation. For example, whenever higher friction between tool and surface cause a positional offset, this effect might be compensated by exerting higher forces on the tool, thus preventing the accumulation of temporal delay. However, as a first approximation, stalling is a very safe general strategy, especially when the robot is working in physically close collaboration with humans. For example, during our experiments (section IV-D), a person intentionally obstacles the robot's motion directly with his hand. The perturbation handler gracefully stops the execution of the ongoing movement until the disturbance is removed. Conversely, an increment of the exerted force in order to overcome the disturbance could potentially create risks for the robot's environment.

As mentioned above, all the experiments in this paper take advantage of the capacity of the system to learn from a single demonstration. Obviously, extracting relevant features by considering a collection of demonstrations is an equally important feature of the model. Such a feature has been effortlessly integrated in our model by using locally weighted regression [10]. For the sake of brevity, further evaluation has been left to future publications.

As a final note, a current limitation of the control system is that only force exertion in the Z direction is considered. However, the generalization to all directions and the exertion of torques is possible and its implementation is rather trivial with the presented approach. Also, the stiffness parameter could be dynamically adapted based on the variations across multiple demonstrations, thus emulating [12]. These and further improvements are left for future work.

ACKNOWLEDGMENT

The authors thank Jeannette Bohg for her support with real-time FRI development. Franz Steinmetz' work at Aalto University has been funded by ESA Directorate of Human Spaceflight and Operation. This work has been funded by the Academy of Finland (grant No. 264239).

REFERENCES

[1] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2004.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 59, pp. 1371–1394.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[4] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799 – 822, 1994.

[5] C. Tung and A. Kak, "Automatic learning of assembly tasks using a dataglove system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 1995, pp. 1 – 8.

[6] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286 – 298, 2007.

[7] S. Calinon, F. D'Halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[8] S. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[9] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control - a unifying view," *Progress in brain research*, vol. 165, no. 1, pp. 425–445, 2007.

[10] A. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[11] P. Pastor, M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou, and S. Schaal, "From dynamic movement primitives to associative skill memories," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 351–361, 2013.

[12] P. Kormushev, S. Calinon, and D.G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, pp. 581–603, 2011.

[13] M. Ito, N. Kuniaki, H. Yukiko, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[14] V. Koropouli, S. Hirche, and D. Lee, "Learning and generalizing force control policies for sculpting," *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pp. 1493–1498, 2012.

[15] A. Montebelli, F. Steinmetz, and V. Kyrki, "On handing down our tools to robots: Single-phase kinesthetic teaching for dynamic in-contact tasks," Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA 2015), 2015.

[16] L. Rozo, S. Calinon, and D.G. Caldwell, "Learning Force and Position Constraints in Human-robot Cooperative Transportation," Proceedings of the 2015 IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man 2015), 2015.

[17] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 10, no. 8, pp. 2047–2084, 1998.

[18] A. Albu-Schaffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot*, vol. 34, no. 5, pp. 376–385, 2007.

[19] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," 2010, pp. 15–21.

[20] M. Muhlig, M. Gienger, S. Hellbach, J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 1177 – 1184.