Motion Learning and Adaptive Impedance for Robot Control during Physical Interaction with Humans

E. Gribovskaya, A. Kheddar, and A. Billard

***Abstract*— One of the hallmarks of physical interaction between humans is haptic communication, i.e. an information exchange through force signals. Humans excel in tasks that require such interaction by adapting impedance and anticipating the partner's intentions. It is highly desirable to endow robots with similar capabilities. Recently, the robotics community renewed its interest in variable impedance control. A special emphasis is put on the development of controllers that incorporate learning as an essential element.**

This article combines programming by demonstration and adaptive control for teaching a robot to physically interact with a human in a collaborative task requiring sharing of a load by the two partners. Learning a task model allows the robot to anticipate the partner's intentions and adapt its motion according to perceived forces. As the human represents a highly complex contact environment, direct reproduction of the learned model may lead to sub-optimal results. To compensate for unmodelled uncertainties, in addition to learning we propose an adaptive control algorithm which tunes the impedance parameters, so as to ensure accurate reproduction.

To simplify the illustration of the concepts introduced in this paper and provide a systematic evaluation, we present experimental results obtained in physically-realistic simulation of a dyad of two planar 2-DOF robots.

I. INTRODUCTION

Biological organisms show a unique competence for *fast* adaptation to novel and unpredictable changes. Such ”on-line” adaptation may require the modification of the control law itself. This is different from typical error correction performed by conventional high-gain, trajectory-tracking robotic controllers [1]. Once a robot is required not only to accurately reproduce a pre-defined motion plan, but also to operate in physical contact with other objects or humans, the choice of control algorithm should be reassessed in favor of adaptable low-gain approaches.

We consider the problem of physical interaction between a robot and a human in situations where they jointly perform manipulation tasks, e.g. the collaborative carrying and positioning of a load.

In [2], [3], we explored learning a control law for driving physical interaction between the HRP-2 robot and a human in a collaborative task of lifting up a beam; see Figure 1-(a). Early analysis of the conducted experiments revealed the importance of the prediction of motion velocity and the anticipation of future perceived forces. Furthermore, additional mechanisms are required to mitigate unmodelled effects during task execution, i.e. sudden, unobserved deviations from a motion plan or varying human impedance. Finally, it appeared necessary to systematically investigate the prediction of the controller in controlled situations. To provide such validation, we present studies modelling the interaction between a pair of two planar robots in physically-realistic simulation[1]; see Figure 1-(b)-(d). We will further denote as *robot-leader* the robot that substitutes the human in the real-world experiments, while the other robot will be denoted as *robot-follower*. The *robot-follower* proactively changes the reference trajectory to synchronize with the *robot-leader*, which keeps its reference trajectory unchanged.

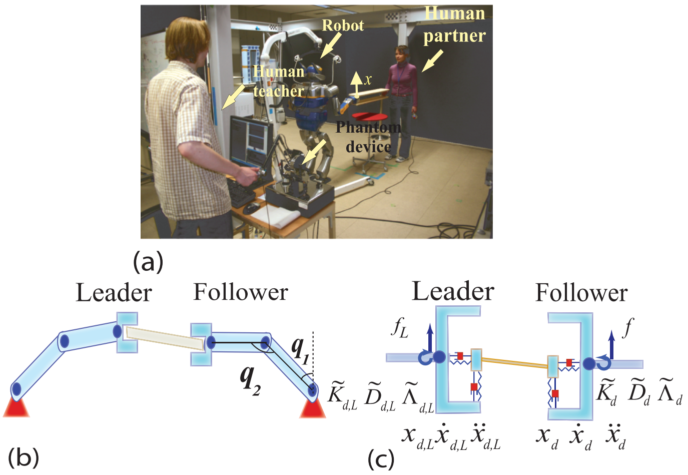
Robot
Human partner
Human teacher
x
Phantom device
(a)
Leader
Follower
q_1
q_2
(b)
f_L
f
$\tilde{K}_{d,L}$ $\tilde{D}_{d,L}$ $\tilde{\Lambda}_{d,L}$
$x_{d,L}$ $\dot{x}_{d,L}$ $\ddot{x}_{d,L}$
$\tilde{K}_d$ $\tilde{D}_d$ $\tilde{\Lambda}_d$
x_d $\dot{x}_d$ $\ddot{x}_d$
(c)

Fig. 1. (a) A human and a robot are collaboratively lifting a beam [3]. (b) Two planar robots (2 DOFs q_1 and q_2) are engaged in a collaborative task requiring lifting a beam up and down. The robot-leader substitutes the human. The robot-follower anticipates the motion intentions of the robot-leader and adapts accordingly. Task completion requires the satisfaction of ”soft constraints”: the two robots must coordinate and adapt their motions so as to avoid tilting the beam, which would let the box fall down. (c) The desired kinematic plan $\boldsymbol{x}_{d,L}, \dot{\boldsymbol{x}}_{d,L}, \ddot{\boldsymbol{x}}_{d,L}$ of the robot-leader is given. During demonstration, the robot-follower learns to generate a desired kinematic command $\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d$ in response to the perceived force $\boldsymbol{f}$. The dynamical behavior of both robot end-effectors is modeled as mechanical impedance, characterized by desired stiffness, damping, and inertia. During the reproduction of the task, the robot-follower adapts its desired stiffness $\tilde{\boldsymbol{K}}_d$ and inertia $\tilde{\boldsymbol{\Lambda}}_d$, so as to ensure accurate reproduction of a learned task model.

A generic approach to modeling robot behavior in contact tasks is *impedance control* [4]. Essentially, impedance control generates a control law consisting of two components: *feedforward control*, which produces the necessary forces to accomplish a task (i.e. to accurately follow the reference signals) in the perfect conditions and a *feedback* signal for correcting errors of the feedforward controller. One distinguishes between *positional feedback* for correcting tracking errors and *force feedback*, which is responsible for adaptation to perceived external forces. The relative importance of these components is controlled by the impedance parameters: stiffness and damping affect positional feedback and inertia influences force feedback.

Traditional position-tracking controllers represent a particular implementation of impedance control, where precedence is given to minimizing position errors, forcing the

E. Gribovskaya and A. Billard are with Learning Algorithms and Systems Laboratory (LASA), EPFL, Switzerland; {elena.gribovskaya,aude.billard}@epfl.ch. A.Kheddar is with CNRS-UM2 LIRMM, Montpellier, France, and the CNRS-AIST JRL, UMI3218/CRT, Japan; kheddar@ieee.org.

[1] During real-world human-robot interaction, one does not observe the motion that the human intends to execute. However, having this information would be desirable for comparing with that predicted by the robot. Simulation allows us access to the data at the ”human” side and to accordingly conduct an accurate comparison.

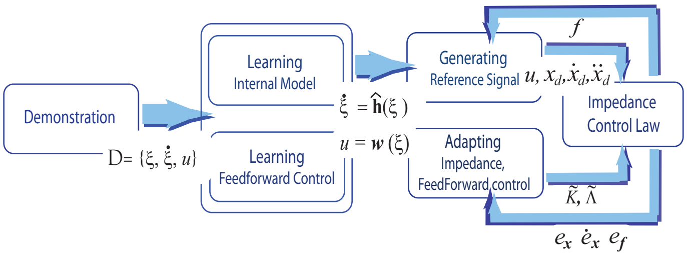

Fig. 2. After acquiring a *set of demonstrations* $\mathcal{D}$, the robot *learns* the task model $\dot{\boldsymbol{\xi}} = \hat{\mathbf{h}}(\boldsymbol{\xi})$ and a forward control signal $\boldsymbol{u} = \boldsymbol{w}(\boldsymbol{\xi})$ which maps the desired state of the internal model to actual motor commands. Representation of the internal task model as a dynamical system allows the robot to generate *reference signals* online adapting to the force applied by a human. The robot is controlled through an *impedance control law* so as to compensate for unmodeled aspects of the external dynamics. The *desired stiffness* $\tilde{\mathbf{K}}$ and *inertia* $\tilde{\mathbf{\Lambda}}$ are *continuously adapted* during task execution.

robot to stiffly reject all external dynamics. This behavior is strikingly different from that observed in humans, who manage to achieve both compliancy and accuracy. Human motion studies suggest that human motor control combines the feedforward and feedback signal, and that the dominance of one component over another gradually changes from feedback control during the early stages of skill acquisition, to feedforward control in highly trained individuals [5]. Recently, several bio-inspired algorithms have been proposed to control robots in contact tasks. Specifically, works [1], [6] advocate that skillful behavior can be accomplished by continuous adaptation of the feedforward control and stiffness. These works, however, consider interaction with either a static environment or random disturbances.

The problem that we address in the current paper goes beyond that of a robot reacting to random dynamical disturbances. Instead, it focuses on continuous prediction and adaptation to the unknown, but systematic, dynamics of a human. We demonstrate that to resolve this problem, the robot should be endowed, not only with variable stiffness [1], [6], but also with variable inertia and an adaptive algorithm to generate different reference kinematic profiles depending on the perceived force.

We present a *novel algorithm* that first infers a *task model*: once such a model is learned, the robot is able to generate reference kinematic signals that are adapted to the perceived force. To compensate for the unmodelled dynamics and natural variability of the human partner, the learned model is further integrated into an *adaptive impedance controller*. The proposed adaptive impedance controller includes an automatic gain-scheduling procedure inspired by the one described in [6]. Specifically, we extend the feedback function so as to consider both positional and force feedback, which leads to a new adaptation law for stiffness, inertia, and feedforward control.

To systematically validate the performance of our approach, we conducted physical simulations of two dynamically-coupled multi-joint robotic arm systems, whose behaviors incorporate *delays* and *signal-dependent* noise. We demonstrate that the proposed controller is able to compensate for varying environmental dynamics (i.e. variability in the human motion) and unknown impedances.

II. PROBLEM STATEMENT

A training set $\mathcal{D}$ consists of M demonstrated trajectories of length N^k, for $k = 1..M$, each trajectory is a sequence of states, $\boldsymbol{\xi}_t^k$, state derivatives $\dot{\boldsymbol{\xi}}_t^k$, and control inputs $\boldsymbol{u}_t^k$, $t = 1..N^k$.

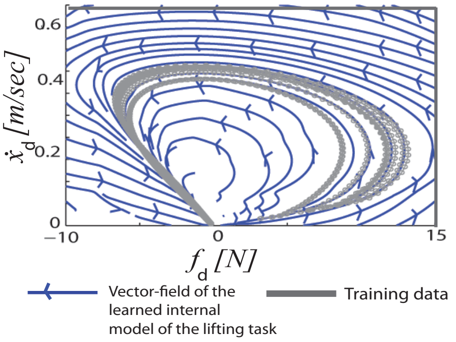

Fig. 3. The task model is represented by an autonomous dynamical system $\dot{\boldsymbol{\xi}} = \hat{\mathbf{h}}(\boldsymbol{\xi})$, $\boldsymbol{\xi} = [\dot{x}_d; f_d]$ and estimated from the training data. At each time step, the velocity $\dot{\boldsymbol{x}}_d$ and force $\boldsymbol{f}_d$ are inferred from these observed at the previous step. Their dynamical relationships follow vector fields displayed in blue in the figure. Dark grey lines show the demonstrations. One can observe an accurate fit between the inferred and demonstrated dynamics. Statistical inference extends prediction of the force-velocity pattern to ranges of these variables not observed during training. This offers a greater robustness during adaptation to a new human partner.

The state vector represents the kinematic and haptic parts of the task $\boldsymbol{\xi}_t^k = [\dot{\boldsymbol{x}}_t^k, \boldsymbol{f}_t^k]^T$. The kinematic part $\dot{\boldsymbol{x}}_t^k$ is the velocity of the robot's hand. The haptic part $\boldsymbol{f}_t^k$ is the external force perceived by a force sensor mounted at the robot's wrist. The control input $\boldsymbol{u}_t^k$ is the vector of joint torques[2]. The details of data acquisition in the real world task with the HRP-2 can be found in [2], [3]; here we develop an alternative procedure for acquiring data in simulation; see Section IV-A.

We assume that the dependency observed between the perceived force and robot's kinematics contains a pattern of adaptation, which should be reproduced by the robot during autonomous interaction. Therefore, we aim to estimate the task model from the observed data. To provide the robot-follower with the means to follow the task model, we learn an approximation of the feedforward control.

The demonstrations are implicitly influenced by some unobservable parameters, dependent on bio-mechanical properties of the arm (e.g. stiffness, damping, etc.). To keep our framework generic, we deliberately do not make any assumptions on the type of the parameters. Instead, we directly encode the internal task model, i.e. the evolution of the task dynamics and the associated feedforward control. To compensate for unmodelled effects and the varying impedance of the human collaborator, we further suggest an outer control loop which implements adaptive robotic impedance.

III. APPROACH

In the next sections we outline our approach to control a robot during physical interaction with a human; see also the schema in Figure 2 for a summary. We first outline our learning algorithm that estimates the task model from demonstrations. Next, we present adaptive impedance control, that allows for the compensation of effects not captured by the learned model. Such a compensation is achieved through the tuning of the impedance parameters and of the feedforward control signal.

A. LEARNING A TASK MODEL

We assume that the internal model of a task-related motion is governed by an autonomous (no explicit time-dependency)

[2] In the following, we consider robots with torque controlled joints.

dynamical system[3]:

$$\dot{\boldsymbol{\xi}} = \mathbf{h}(\boldsymbol{\xi}) + \boldsymbol{\eta}(\boldsymbol{\xi}), \tag{1}$$

where $\boldsymbol{\xi}$ is the state of the system, $\mathbf{h}(\boldsymbol{\xi})$ is the dynamical function governing the temporal evolution of the motion and $\boldsymbol{\eta}(\boldsymbol{\xi}) \sim N(0, \boldsymbol{\Sigma}_\eta(\boldsymbol{\xi}))$ is the signal-dependent noise.

In our previous work [7], we developed a statistical framework which allowed one to learn a kinodynamical model of the task from a series of demonstrations. Essentially, the method consists of representing a demonstrated dataset $\mathcal{D}$ of the state vectors and its derivatives as the joint probability distribution $\mathcal{P}(\dot{\boldsymbol{\xi}}, \boldsymbol{\xi})$ and then extracting the dynamics function $\hat{\mathbf{h}}(\boldsymbol{\xi})$ as the posterior mean estimate $\hat{\mathbf{h}}(\boldsymbol{\xi}) = \mathbb{E}[\mathcal{P}(\dot{\boldsymbol{\xi}}|\boldsymbol{\xi})]$.

Due to the particular choice of the parametrization through Gaussian Mixture Models (GMM), the learned non-linear estimate $\hat{\mathbf{h}}(\boldsymbol{\xi})$ is expressed as a sum of linear dynamical systems weighted with nonlinear coefficients. The parameters of the GMM, namely the means and covariance matrices, are learned through an incremental EM procedure that ensures that the final estimate is asymptotically stable at an attractor (see [7]–[9] for details on the learning algorithm).

Here the attractor is predetermined and corresponds to the origin of the system, i.e. force and velocity are zero. If no force is perceived, the robot maintains an initial posture without moving. When it perceives the force, the task model starts to generate the reference kinematic profile according to the non-linear force-velocity dependency; see Figure 3.

In our previous works, we considered that the state of the system in Eq.1 represents the end-effector's configuration in the task space. Here we define an *augmented state* of the system, that consists of the two parts: the kinematic part, that represents the reference velocity $\dot{\boldsymbol{x}}_d$, and the force feedback part, that represents the expected external force $\boldsymbol{f}_d$:

$$\boldsymbol{\xi} = [\dot{\boldsymbol{x}}_d; \boldsymbol{f}_d]^T. \tag{2}$$

The internal model in Eq.1 with the augmented state given by Eq.2 iteratively generates a reference acceleration for the robot's end-effector $\ddot{\boldsymbol{x}}_d$ together with the prediction of the force derivative $\dot{\boldsymbol{f}}_d$. The velocity and position signals $\dot{\boldsymbol{x}}_d$, $\boldsymbol{x}_d$, as well as force prediction $\boldsymbol{f}_d$ are then obtained by integration.

The important novelty here consists of representing the state of the task model through what we refer to as an *augmented state*. The augmented state encapsulates both the kinematic command, $\dot{\boldsymbol{x}}$, and the haptic input, $\boldsymbol{f}$, and allows one to learn a temporal evolution of the reference velocity correlated with the perceived external force. As such, this allows the robot to smoothly switch across different reference velocity profiles in response to a change in the human's intentions, as perceived through the force input. A further advantage of this encoding is that the internal model can be used to mitigate sensory delays and noise by predicting the perceived force. Specifically, to generate a reference kinematics, the robot does not need to get the actual value of the perceived force at each time step, it can predict the perceived force to a velocity. Later, once it gets actual value of the force, it may offset its prediction so as to switch to a different velocity profile if necessary.

B. LEARNING FEEDFORWARD CONTROL

Generation of the feedforward control $\boldsymbol{u}$ directly falls within the category of problems tackled in operational space control [10], [11]. It can be formalized as $\boldsymbol{u} = \boldsymbol{w}(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d)$: the signal $\boldsymbol{u}$ controls the robot such that the end-effector follows a desired Cartesian path $\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d$.

The learned task model $\dot{\boldsymbol{\xi}} = \hat{\mathbf{h}}(\boldsymbol{\xi})$, with state $\boldsymbol{\xi} = [\dot{\boldsymbol{x}}_d; \boldsymbol{f}_d]$, defines a manifold in the space $\{\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d\}$. Along this manifold, both the reference acceleration $\ddot{\boldsymbol{x}}_d$ and position $\boldsymbol{x}_d$ are functions of the state vector $\boldsymbol{\xi}_d$ ($\ddot{\boldsymbol{x}}_d$ is generated by $\hat{\mathbf{h}}(\boldsymbol{\xi})$ and $\boldsymbol{x}_d$ is computed by integrating forward the reference velocity $\dot{\boldsymbol{x}}_d$ starting from an initial condition $\boldsymbol{x}_{d,0}$.) Therefore, in this case, the feedforward control $\boldsymbol{u}$ can be rewritten in a more compact form as follows: $\boldsymbol{u} = \boldsymbol{w}(\boldsymbol{\xi}, \boldsymbol{x}_{d,0})$.

The adaptive impedance control that we develop in the next sections requires a linear parametrization of the control signal $\boldsymbol{u}$. Following [6], we write an approximation of the control law as:

$$\boldsymbol{u} = [\boldsymbol{\Phi}(\boldsymbol{\xi})^T \boldsymbol{\Theta}]^T, \tag{3}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{K(N_x+N_f)}$ is a vector of G basis functions and $\boldsymbol{\Theta} \in \mathbb{R}^{K(N_x+N_f)\times N_q}$ is a matrix of the tunable parameters (each column $\boldsymbol{\theta}^i$, $i = 1..N_q$, of the matrix $\boldsymbol{\Theta}$ corresponds to one degree of freedom in the joint space). N_x, N_f, N_q refer to the dimensionality of the Cartesian space of the robot's end-effector, perceived force, and joint space, respectively. This type of linear control parametrization is commonly used in the adaptive control literature [12], [13]. The basis $\boldsymbol{\Phi}$ consists of G Gaussian functions:

$$\boldsymbol{\Phi}_j = \boldsymbol{\Phi}(\boldsymbol{\xi})_j = \frac{\pi_j(\boldsymbol{\xi})\ \boldsymbol{\xi}}{\sum_{g=1}^{G} \pi_g(\boldsymbol{\xi})},\ j = 1..G \tag{4}$$
$$\text{where } \pi_j(\boldsymbol{\xi}) = \exp^{-0.5(\boldsymbol{\xi}-\boldsymbol{\mu}_{\boldsymbol{\xi},j})^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{\xi}-\boldsymbol{\mu}_{\boldsymbol{\xi},j})}$$

where $\boldsymbol{\mu}_{\boldsymbol{\xi},j}$, $\boldsymbol{\Sigma}_j$ are the mean and the diagonal covariance of a jth Gaussian kernel. We learn the parametrization in Eq.3 through Linear Weighted Regression [14], by fixing the means and covariances of Gaussian kernels.

C. IMPEDANCE CONTROL

The goal of impedance control is to implement a desired dynamical relationship between the robot motion and the external forces/torques [4]. Our particular interest lies with the control of impedance at contact points defined in Cartesian space, specifically, at the robot's end-effector.

We write the rigid body dynamics in the task coordinates [15] [4] as:

$$\boldsymbol{\Lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{J}^{-T}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}^{-T}\boldsymbol{\tau} + \boldsymbol{f}. \tag{5}$$

The matrices $\boldsymbol{\Lambda}(\boldsymbol{x})$ and $\boldsymbol{\mu}(\boldsymbol{x})$ are given by:

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1} \quad \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{J}^{-1}(\mathbf{C} - \boldsymbol{M}\boldsymbol{J}^{-1}\dot{\boldsymbol{J}})\boldsymbol{J}^{-1}, \tag{6}$$

where $\boldsymbol{q} \in \mathbb{R}^{N_q}$ is the vector of joint angles, $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Coriolis/centrifugal matrix, $\boldsymbol{g}(\boldsymbol{q})$ is the vector of gravity torques, $\boldsymbol{\tau}_{ext}$ is the vector of external torques, and $\boldsymbol{\tau}$ are the applied joint torques. The relation between the Cartesian coordinates $\boldsymbol{x}$ and the joint coordinates $\boldsymbol{q}$ is given by the kinematic function with the Jacobian $\boldsymbol{J}(\boldsymbol{q})$.

Impedance control defined in Cartesian space consists of the following control objective [4]:

$$\boldsymbol{\Lambda}_d\ddot{\boldsymbol{e}}_x + \boldsymbol{D}_d\dot{\boldsymbol{e}}_x + \boldsymbol{K}_d\boldsymbol{e}_x = \boldsymbol{e}_f \tag{7}$$
$$\text{with } \boldsymbol{e}_x = \boldsymbol{x} - \boldsymbol{x}_d \text{ and } \boldsymbol{e}_f = \boldsymbol{f} - \boldsymbol{f}_d.$$

[3] Encoding a task model as an autonomous dynamical system allows to represent behaviors that cannot be unambiguously expressed by a single non-autoregressive function. Specifically, the dependency between force and velocity in our case is not functional: the same value of force corresponds to different velocities; see Figure 3.

[4] The notation $^{-T}$ refers to pseudo-inverse of a transposed matrix

where $\boldsymbol{e}_x$ is the position error between the actual position $\boldsymbol{x}$ and the reference position $\boldsymbol{x}_d$; $\boldsymbol{e}_f$ measures how much the actual perceived force $\boldsymbol{f}$ deviates from the predicted one $\boldsymbol{f}_d$. $\boldsymbol{\Lambda}_d$, $\boldsymbol{D}_d$, and $\boldsymbol{K}_d$ are the symmetric and positive definite matrices of desired inertia, damping, and stiffness, respectively. The external forces defined in the Cartesian space are projected onto the joint torques according to: $\boldsymbol{\tau}_{ext} = \boldsymbol{J}^T\boldsymbol{f}$.

Substituting $\ddot{\boldsymbol{x}}$ from Eq.7 into Eq. 5, the Cartesian impedance controller can be implemented via the joint torques $\boldsymbol{\tau}$ as follows:

$$\boldsymbol{\tau} = \boldsymbol{u} + \boldsymbol{J}^T\tilde{\boldsymbol{K}}\boldsymbol{e}_x + \boldsymbol{J}^T\tilde{\boldsymbol{D}}\dot{\boldsymbol{e}}_x + \boldsymbol{J}^T\tilde{\boldsymbol{\Lambda}}_d\boldsymbol{e}_f. \tag{8}$$

where

$$\begin{aligned}&\boldsymbol{u} = \boldsymbol{g} + \boldsymbol{J}^T(\boldsymbol{\Lambda}\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}\dot{\boldsymbol{x}}_d) \qquad (9)\\ &\tilde{\boldsymbol{K}}_d = \boldsymbol{\Lambda}\boldsymbol{\Lambda}_d^{-1}\boldsymbol{K}_d,\ \tilde{\boldsymbol{D}}_d = \boldsymbol{\Lambda}\boldsymbol{\Lambda}_d^{-1}\boldsymbol{D}_d + \boldsymbol{\mu},\ \tilde{\boldsymbol{\Lambda}}_d = \boldsymbol{\Lambda}\boldsymbol{\Lambda}_d^{-1} - \boldsymbol{I}.\end{aligned}$$

Eq.8 can be rewritten as a sum of the feedforward and feedback components. The feedback signal can be decomposed into the kinematic feedback $\boldsymbol{v}_x$ and the force feedback $\boldsymbol{v}_f$.

$$\begin{aligned}&\boldsymbol{\tau} = \boldsymbol{u} + \boldsymbol{v},\ \boldsymbol{v} = \boldsymbol{v}_x + \boldsymbol{v}_f. \qquad (10)\\ &\boldsymbol{v}_x = \boldsymbol{J}^T(\tilde{\boldsymbol{K}}_d\boldsymbol{e}_x + \tilde{\boldsymbol{D}}_d\dot{\boldsymbol{e}}_x),\ \boldsymbol{v}_f = \boldsymbol{J}^T\tilde{\boldsymbol{\Lambda}}_d\boldsymbol{e}_f\end{aligned}$$

Recently different approaches for shaping the desired stiffness and damping have been suggested (see [6], [16], [17]). In these works, the shaping of the desired inertia $\boldsymbol{\Lambda}_d$ has been avoided by assigning it to be equal to the inertia of the robot: $\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}$. This simplifies control law computations and eliminates the need to obtain an accurate dynamical model of the robot. However, such an assumption also leads to an important compromise: any system, interacting with a robot, will sense its actual dynamical properties, which are different from that of a human. A user will be exposed to the unnatural robot's dynamics. In tasks where the robot physically interacts with humans, adaptation through positional feedback is not sufficient, and shaping of the desired inertia becomes important [18], [19].

In the remainder of this paper, we assume the desired impedance matrices to be diagonal: $\tilde{\boldsymbol{K}}_d = \text{diag}\{\tilde{K}_d^j\}_{j=1}^{\mathbb{R}^{N_x}}$, $\tilde{\boldsymbol{D}}_d = \text{diag}\{\tilde{D}_d^j\}_{j=1}^{\mathbb{R}^{N_x}}$, and $\tilde{\boldsymbol{\Lambda}}_d = \text{diag}\{\tilde{\Lambda}_d^j\}_{j=1}^{\mathbb{R}^{N_x}}$. Common practice assigns the damping factor $\tilde{\boldsymbol{D}}$ to be a function of the stiffness $\tilde{\boldsymbol{K}}$. Here we set $\tilde{D}^j = \lambda\sqrt{\tilde{K}^j}$, where λ is an empirical parameter ($\lambda = 2$ in our experiments).

D. ADAPTIVE IMPEDANCE

In this section we sketch an adaptive control algorithm for learning the impedance parameters. The algorithm derived in this section is an extension of a bio-mimetic approach for tuning the stiffness of the robot end-effector presented in [6]. We provide a brief outline of the original approach in Appendix I; for more details, please refer to the original paper.

Consider an adaptation law for feedforward control, defined as follows:

$$\Delta\boldsymbol{\theta}^i = \frac{\beta}{2}(1-\chi)\boldsymbol{\Phi}\epsilon^i + \frac{\beta}{2}(1+\chi)\boldsymbol{\Phi}|\epsilon^i| - \gamma\boldsymbol{1},\ i = 1..N_q \tag{11}$$

where ϵ is the error function, which will be defined later and β, χ, and γ are empirical constants.

During free motion, deviations of the robot from a reference trajectory are solely caused by inaccuracies in the dynamical models of either the robot's body or the object being manipulated. In contrast, during physical interaction, such a deviation can also be due to variations in the partner's intentions. The view on processing kinematics errors, which is taken in the original paper [6], suggests counteracting them by applying force in the opposite direction. In our case the robot should only counteract kinematics errors which are due to model inaccuracies. Therefore, we introduce a new feedback function which includes the force feedback and the object-related kinematic error:

$$\begin{aligned}&\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{x,m} + \boldsymbol{\epsilon}_f = J^T\boldsymbol{e}_{x,m} + \rho_3 J^T\boldsymbol{e}_f. \qquad (12)\\ &\text{where } \boldsymbol{e}_{x,m} = \ell\min(\boldsymbol{f}\cdot\dot{\boldsymbol{e}}_x, 0)(\rho_1\boldsymbol{e}_x + \rho_2\dot{\boldsymbol{e}}_x)\end{aligned}$$

where $\cdot$ refers to the inner vector product and $\ell = \|\boldsymbol{f}\cdot\dot{\boldsymbol{e}}_x\|^{-1}$ is the normalization factor. The term $\boldsymbol{e}_{x,m}$ defines the kinematic error which is due to the varying mass of the object. We can now rewrite Eq.11 as:

$$\begin{aligned}\Delta\boldsymbol{\theta}^i = &\frac{\beta}{2}(1-\chi_1)\boldsymbol{\Phi}\epsilon_x^i + \frac{\beta}{2}(1+\chi_1)\boldsymbol{\Phi}|\epsilon_x^i| + \qquad (13)\\ &\frac{\beta}{2}(1-\chi_2)\boldsymbol{\Phi}\epsilon_f^i + \frac{\beta}{2}(1+\chi_2)\boldsymbol{\Phi}|\epsilon_f^i| - \gamma\boldsymbol{1}\end{aligned}$$

In Eq.13, the terms $\frac{\beta}{2}(1-\chi_1)\boldsymbol{\Phi}\epsilon_x^i$ and $\frac{\beta}{2}(1-\chi_2)\boldsymbol{\Phi}\epsilon_f^i$ correspond to the conventional adaptation law from the control literature. $\frac{\beta}{2}(1-\chi_1)\boldsymbol{\Phi}\epsilon_x^i$ generates a force in the opposite direction of the kinematic error $\boldsymbol{e}_{x,m}$, and updates the feedforward signal $\boldsymbol{u}$. $\frac{\beta}{2}(1-\chi_2)\boldsymbol{\Phi}\epsilon_f^i$ compensates for the deviation of the actual external force from the reference external force and contributes to the adaptation of the desired inertia.

The terms dependent on the absolute values of the errors aim to tune stiffness. $\frac{\beta}{2}(1+\chi_1)\boldsymbol{\Phi}|\epsilon_x^i|$ increases stability in response to kinematic perturbation, while $\frac{\beta}{2}(1+\chi_2)\boldsymbol{\Phi}|\epsilon_f^i|$ decreases the stiffness if the deviation of the external force is increasing. Indeed, a sudden increase in the force error $\boldsymbol{e}_f$ signifies that the human is attempting to impose a different motion plan, and hence the robot should decrease the stiffness so as to maintain stable interaction.

Analogous to [6], the update mechanism emulates automatic relaxation through the term $\gamma\boldsymbol{1}$. This is similar to a behavior observed in humans who, in the absence of motion errors, tend to relax muscles so as to minimize energy consumption. Regrouping the terms in Eq.20 according to the analysis above, the updating procedure for the forward signal and the impedance parameters can be written as follows:

$$\Delta\boldsymbol{\theta}^i = \kappa_x\boldsymbol{\Phi}\boldsymbol{\epsilon}_{x,m} - \boldsymbol{\gamma}_\theta,\quad \kappa_x > 0 \tag{14}$$

$$\Delta\tilde{K}_d^j = \beta_x|e_{x,m}^j| - \beta_f|e_f^j| - \gamma_{\tilde{K}},\quad \beta_x, \beta_y, \gamma > 0. \tag{15}$$

$$\Delta\tilde{\Lambda}_d^j = \kappa_f e_f^j - \gamma_{\tilde{\Lambda}},\ \ j = 1..N_x\ \ \kappa_f > 0 \tag{16}$$

Eq.8 together with Eq.3,14-16 represent the control algorithm which allows for the simultaneous on-line adaptation of the feedforward signal, desired stiffness and inertia.

IV. RESULTS

We validate our method in simulations that control the robot-follower interacting with another planar robot; see Figure 1-(b). To highlight different types of adaptation handled by our algorithm, we simulate different conditions that may arise during execution of the collaborative tasks, and that would require on-the-fly adaptation of the robot's control law.

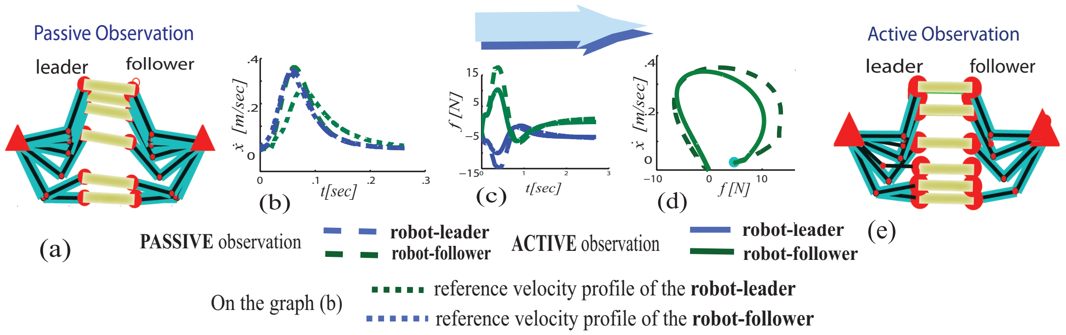

Fig. 4. TWO-STAGE TRAINING PROCEDURE. To simulate real-world training, where the robot is teleoperated by a human, we adopt a two stage training procedure. Figures (a), (e) present the robots' configurations during training. Desyncronization between the partners is greatly reduced during active observation, as expressed by the reduced tilting of the beam. *PASSIVE OBSERVATION*: The stiffness of the robot-follower is set to be low ($5N/m$) and the stiffness of the robot-leader is high ($50N/m$). This allows the robot-leader to impose its kinematic plan; see Figure (b). The actual velocity of the robot-follower is higher than its reference signal and coincides with the actual and reference velocities of the robot-leader. Such a forced adaptation is achieved at the cost of considerable energy injection; see Figure (c)-(d). The robot-follower perceives high positive external forces, which are due to the effort of the robot-leader. After observing the task "passively", the robot-follower stores the kinematic information and discards the force signals. *ACTIVE OBSERVATION*: The stiffness of both partners is medium ($15N/m$). The robot-leader repeats the same reference kinematical profile as at the previous stage, while the robot-follower utilizes the adapted kinematic profile acquired during passive observation. Reproducing the adapted motion, the robot-follower is more coordinated with its partner. Improvements in coordination lead to a decrease in the magnitude of the forces perceived by both partners; see Figure (c)-(d), solid line. The final training set is composed of the velocity signal recorded during passive observation, and the external forces/applied torques recorded during active observation.

A. TWO-STAGE TRAINING PROCEDURE

Demonstration data, which we will use next for validating our algorithm, are acquired through a two-stage training procedure: in each demonstration, the robots alternate between "*passive*" and "*active*" stages. In total, 15 demonstrations at different speeds are provided. This procedure is inspired from the way humans incrementally learn to synchronize with each other. Here, the leader keeps the same pace during passive and active stages. The follower adapts its desired kinematics to synchronize with the leader.

We simulate the two types of sensorimotor limitations of the human motor system: a signal-dependent noise and sensory delay. We assign an *initial reaction* delay (a time span between the moment when the robot-follower starts to perceive the changing force and the moment when it actually starts moving) to be $150ms$; the *constant perception delay* along the motion (delay between perception of a force and generation of a reaction) is $3ms$.

The robots are controlled with the impedance control law according to Eq.8 with the pre-defined impedance parameters and the zero reference force $\boldsymbol{f}_d$. The reference kinematics profiles share the same goal (i.e. bring the beam in a specified location), but have dissimilar timing (due to different reference velocity profiles and sensory delays). For both the leader and the follower, the reference kinematics are generated with a dynamical system parameterized with a multiplicative parameter, so as to produce the same task space trajectories but with different velocity profiles. Specifically, we use the VITE dynamical model of human reaching motions [20], [21]: $\ddot{\boldsymbol{x}}_d = \alpha(-\dot{\boldsymbol{x}}_d + 4(\boldsymbol{x}_{\text{tar}} - \boldsymbol{x}_d))$, where α is the multiplicative parameter, $\boldsymbol{x}_{\text{tar}}$ is the given target location[5].

To provide examples of adaptation to different velocities, the parameter α of the robot-leader is varied from one demonstration to another (but not between the two stages of one demonstration), so as to generate motions with a maximum desired velocity of $0.2m/s$-$0.5m/s$ and duration of 0.4-$0.8\ s$. The parameter α of the robot-follower is fixed; the VITE model generates the reference motion of $0.8s$ with the maximum velocity $0.2m/s$.

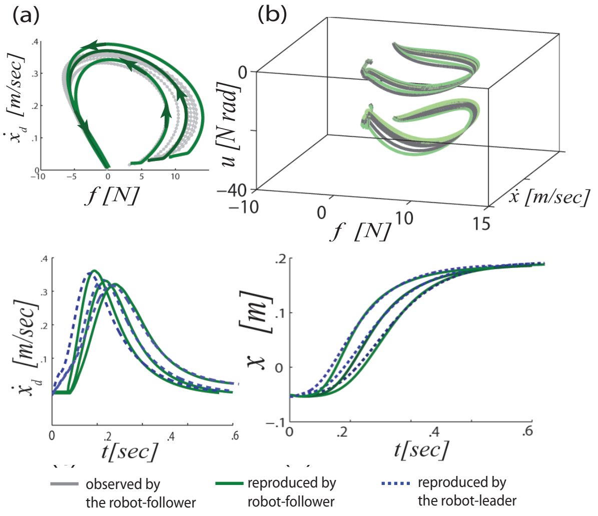

Fig. 5. TASK LEARNING AND REPRODUCTION. In this experiment, the robot-follower learns the lifting task by observing several demonstrations performed with different velocity profiles imposed by the robot-leader. Next, during reproduction, the robot-leader varies its kinematics plan from one attempt to another; it does so by changing motion duration and maximum velocity. The robot-follower, governed by the learned task and control model, adapts to the motion of the robot-leader and successfully accomplishes the task. (a) The state-space view of the data used for training a task model (grey line) and the reproduction attempts (green line). Note the non-linear correlation pattern between the perceived force and generated velocity. (b) The forward control signal generated by the robot-follower during demonstration (grey line) and reproduction (green line). The two datasets correspond to the two joints of the planar robot-follower. (c)-(d) The time-series of the Cartesian velocity and trajectory of the robot-follower during reproduction. After an initial delay in synchronization, the robot-follower (green line) adjusts its reference kinematics and successfully synchronizes with the robot-leader (dashed blue line).

1) Passive Observation: The two robots are controlled to follow their reference kinematical profiles generated with the VITE system as discussed above. The stiffness of the robot-follower is set to be low ($5N/m$) and the stiffness of the robot-leader is high ($50N/m$). The robot-leader imposes its motion plan to the partner; see Figure 4-(b). This requires the robot-leader to inject a considerable amount of energy; see Figure 4-(c),(d), dashed line. Therefore, even though the robot-follower follows the motion of the robot-leader (the actual velocity of the robot-follower is close to that of the

[5] A model parameterized with $\alpha = 10$ will produce the same trajectories in the state-space $\{x_1; x_2\}$ as a model parameterized with $\alpha = 20$, however the latter one converges to the target twice as fast.

leader), the perceived forces are different from those that would be observed if it did so intentionally. To observe these forces, the robot-follower reproduces the actual kinematics during the next training step.

2) Active Observation: The robot-leader tracks the same reference trajectory $\boldsymbol{x}_{d,L}, \dot{\boldsymbol{x}}_{d,L}, \ddot{\boldsymbol{x}}_{d,L}$ as during the passive stage. The robot-follower utilizes as the reference signal the actual kinematics $\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$ recorded during passive observation (as it more accurately matches the reference kinematics of the leader; see Figure 4-(b)). The stiffness of both partners is medium ($15N/m$).

By deliberately reproducing this imposed kinematic profile, the robot-follower generates forces that are better aligned with those of the robot-leader. At this stage the robot-leader injects less energy and, therefore, the forces perceived by the robot-follower are smaller than those at the first stage; see Figure 4-(c),(d), solid line.

Figure 4-(a),(e) highlights improvements in synchronization between the robots across the two training stages.

The collected data (the actual velocity, the perceived forces, and the feedforward commands of the robot-follower) are further used to learn models of the task $\hat{\boldsymbol{h}}$ and of the feedforward control $\boldsymbol{w}$.

B. LEARNING A TASK MODEL

The acquired training data are depicted in Figure 5 - (a),(b); note that the data exhibit a non-linear force-velocity correlation, and that the pattern of this correlation is similar to the one observed in real-world data acquired with HRP-2 robot (see our previous work [3]).

After acquiring the training data, the robot learns the internal model of the task $\dot{\boldsymbol{\xi}} = \hat{\mathbf{h}}(\boldsymbol{\xi})$ and the forward control model $\boldsymbol{u} = \boldsymbol{w}(\boldsymbol{\xi})$. During reproduction, the learned models are fed into the control law in Eq. 8. The results of the task execution are depicted in green in Figure 5. Note that the robot-follower manages to successfully choose different velocity profiles, which are synchronized with those applied by the robot-leader.

C. ADAPTATION TO PERTURBATIONS

We tested the ability of the trained model to adapt to changing intentions of the robot-leader during motion execution. We simulated uncertainties about the motion objectives by varying the target position $\boldsymbol{x}_{\text{tar}}$ in the motion plan of the robot-leader. Three different cases are considered where the robot-leader changes the motion plan in real-time, during task execution. It decides to move the beam (1) higher than it has initially planned, (2) lower, but still higher than the actual position of the robots at the moment of taking the decision, and (3) lower than both the original target position and the actual position.

Figure 6 shows that the robot-follower succeeds in following the robot-leader, by bringing the beam to the new desired location. This exploits the fact that the learned internal model generalizes the force and velocity patterns to values not observed during training, as discussed in Figure 3. Note that the robot-follower, governed by the learned model, succeeds in following the robot-leader according to the demonstrated motion pattern. In all trials the robot-follower brings the beam to the new target position imposed by the robot-leader.

In the first case, after a short period during which the velocity decreases (due to inertia and perception delay), the robot-follower catches up with the intentions of the robot-leader keep higher; and in turn starts to reaccelerate to also move higher. In the second case (placing the beam into the lower position before this position has been passed), the robot-follower decelerates faster than during the normal conditions; see Figure 6-(c). In the third case, when the robot-follower manages to rapidly decelerate and smoothly switch to the negative velocity. Note, that in the second and the third cases, the robot-follower slightly overshoots after the perturbation, as it requires some time to readjust its reference kinematics.

D. ADAPTATION OF UNKNOWN IMPEDANCE

In the previous experiments we reused the impedance parameters which the two robots had during training. However, in general the impedance parameters of the robot-follower are unknown, e.g. if the demonstrations are provided through teleoperation. We now assume that the robot-follower has no information about the impedance it should apply at the end-effector. Therefore, the robot-follower should adapt the parameters on-line, during task reproduction; see Figure 7.

The experiment starts with the robot-follower stiffness and inertia set to mid-range values of $35N/m$ and 0 respectively (in contrast, during demonstration the two parameters were $10N/m$ and 0.7). This is an important difference in the parameter values. One can see in Figure 7-(a), dashed line, that reproduction without adaptation of the impedance parameters leads to an overestimated reference velocity and instabilities at the target. Adaptation allows the tuning of the parameters so as to facilitate stable interactions and convergence to the target; see Figure 7-(a), green line. After slightly growing in the beginning of the motion, to enable for smooth acceleration, the stiffness gradually decreases due to the relaxation term and errors in tracking the desired force profile; see Figure 7-(d). The inertia, in turn, decreases in the beginning, to ensure the stable onset of the motion. It further increases to endow the robot-follower with greater reactivity to the partner's intentions; see Figure 7-(c).

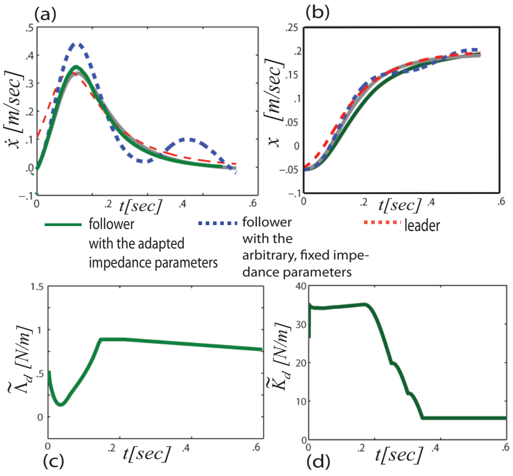

Fig. 7. ADAPTATION OF UNKNOWN IMPEDANCE. In general, the impedance parameters that would be optimal for the task are unknown. Therefore, the robot-follower should tune its stiffness and inertia during task execution. Indeed, the arbitrary impedance parameters may cause undesirable effects, e.g. overestimated reference kinematics and contact instabilities (blue dashed line). Our algorithm provides an efficient adaptation law, which allows to tune the parameters, so as to ensure accurate reproduction of the observed behavior (green lines in (a)-(b) show the results with impedance adaptation).

E. COMPARISON WITH A DAMPING CONTROLLER

To highlight the advantages of the proposed approach for controlling a robot during collaborative manipulation tasks, we implemented a damping controller [22]. This easy to

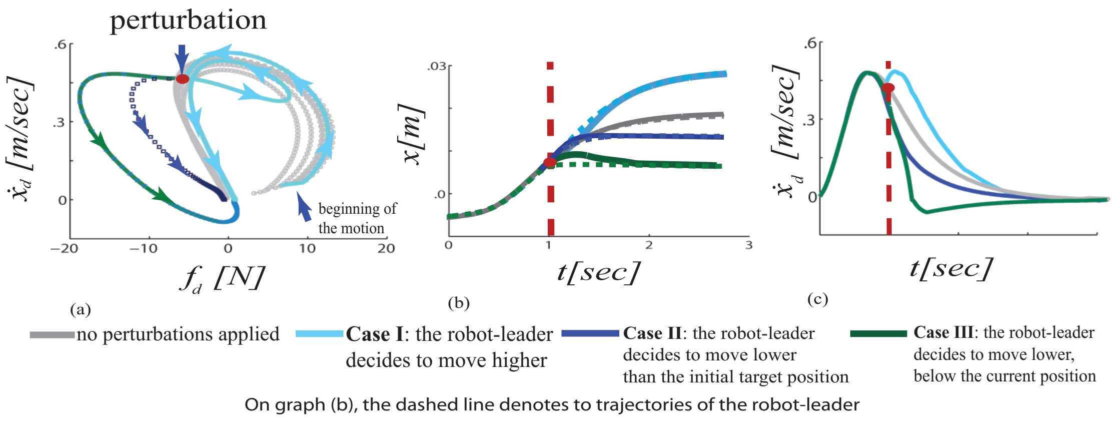

Fig. 6. ADAPTATION TO PERTURBATIONS. We analyze three different cases where the robot-leader decides to change the motion plan online and move the beam (1) higher than it has initially planned, (2) lower, but higher than the actual position of the beam at the moment of taking the decision about changing, and (3) lower than both the original target position and the actual position at the decision-taking moment. In the first case, the robot-follower manages to reaccelerate (see the two peaks in the velocity profile). While in the two other cases the robot-follower proactively decelerates. In the third case, the robot also manages to smoothly drop velocity below zero and lower the beam.

implement and computationally cheap method is often used to control robots during physical collaboration with people. The only free parameter in this controller is the damping coefficient; the reference kinematics and all other impedance parameters are set to zero. The resultant robot behavior is purely reactive, i.e. the only source of mechanical energy is a human partner, who should constantly inject energy to keep the robot moving. The damping control has been proven to be useful and efficient in many applications; however, it puts additional workload on the human and is not adequate to control for fast motions.

We compare the performance of our system versus that of the damping controller in Figure 8. The forces perceived by the robot are much higher than those observed with our learned system (Figure 8-(d)). This is due to the higher forces that the robot-leader has to apply to control the interaction. Additionally, the beam undergoes stronger rotation (Figure 8-(b)) due to the imbalanced forces on its two sides. This will be highly undesirable if the beam is loaded with unfixed objects, which may fall down if the rotation is too strong.

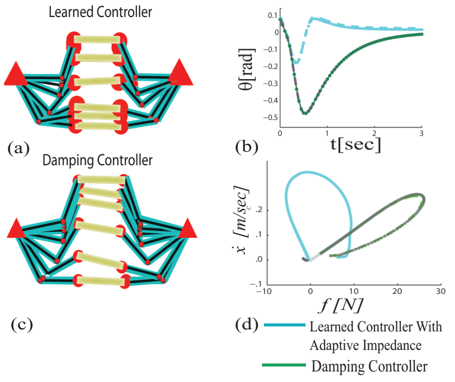

Fig. 8. COMPARISON WITH A DAMPING CONTROLLER. We compare the performance of our system versus that of the damping controller. (a), (c) Snapshots of the planar robots performing lifting; the robot follower is controlled by (a) the proposed learned controller, and (c) the damping controller. One may visually notice the improvements in coordination between the partners when the robot-follower adapts its kinematic profile (a): the beam is mostly kept horizontal all along the motion. It is persistently tilted when the robot-follower is controlled with the damping (c). (d) θ characterizes deviation of the beam from the horizontal orientation. The damping controller produces force imbalance on the two sides of the beam which leads to greater variation in θ along the motion. (b) The robot-leader has to apply considerably higher forces to make the system move.

V. RELATED WORK

While interaction control has been investigated for several decades, the available tools remain substantially less efficient than those developed for free-space motion. Later attempts to implement physical human-robot interaction introduced the notions of variable impedance and active following. For instance in [23] a robot adjusts its damping depending on the perceived force; in [22] it tries to predict the target of a motion to proactively generate a reference trajectory. Although their validity is confirmed with successful experimental results, they are still ad-hoc solutions: no generic framework for tackling both the learning of task models and the variable impedance is provided.

With recently renewed interest in impedance control, several methods have been suggested to address the problem of the automatic scheduling of the impedance parameters. The existing algorithms implement learning the variable stiffness through reinforcement learning [16] or adaptive control [6]. The major assumption shared by these approaches is the repeatability of both a task and the external dynamics, which allows a robot to adapt its behavior between multiple trials.

In robot learning, the few existing works consider a learned trajectory as a reference signal for a hard-coded impedance controller. In [24], a robot is taught to clap hands with a human. The robot utilizes Hidden Markov Models (HMM) to recognize the human behavior from motion data and generate a reference trajectory. This trajectory is further incorporated into a hard-coded impedance controller to compensate for a potential physical impact. The considered scenario does not require continuous physical interaction and haptic signals do not effect the reference signal.

A hand-shaking robot is presented in [25]. In this method, the authors encode motion trajectories with an HMM, where the hidden variables represent the human impedance. Such encoding requires the robot to measure human impedance and further recognize which motion model to choose. The motion model is chosen at the onset of the task and governs the robot through the rest of the task without adaptation. This is different from our approach that allows continuous adaptation of the motion throughout the task.

VI. CONCLUSIONS AND FUTURE WORKS

We present an approach to learning robot control during physical interaction with humans. The method addresses the problem of controlling a robot so that it can coordinate its motions with that of a human in collaborative tasks, and this while relying solely on haptic and proprioceptive feedback (no vision or verbal commands are involved).

In contrast to other works on variable impedance, our method allows for adaptation *within* an execution trial, and not only from trial to trial. Since there is a human in the loop, this characteristic is essential. We cannot ensure that at the next trial the person will identically repeat the task and provide the robot with time to tune its controller. However, our method still benefits from a model-free approach which eliminates the necessity to derive a complete dynamical model of the robot.

In the current contribution, we address the case of torque-controlled robots and derive an impedance-based approach. Currently, we are working on an extension of our algorithm that encapsulates an admittance-based controller for position-controlled robots. We will implement the suggested algorithm on an actual HRP-2 robot. We believe that our method contributes importantly to research on physical human-robot interaction. The proposed system endows the robot with two fundamental features of human motor control that emerge during physical interaction: learning haptic communication in a natural manner, and continuous adaptation to incoming forces during task execution. Additionally, the simulator developed in this work provides an efficient means to study physical interactions between two agents for which we have yet very few models. It offers a framework for systematically assessing and comparing performance of different algorithms to control human-robot interaction, a very intricate problem.

VII. APPENDIX

We further briefly summarize the major concepts of the biomimetic adaptive algorithm presented in [6], for details, please, refer to the original publication.

We use the following cost function [6] which penalizes the feedback cost and activation of the feedforward command:

$$\min_{\boldsymbol{\theta}^i} \ R^i(\boldsymbol{\theta}^i) = 0.5\beta\ (v^i)^2 + \gamma \sum_{k=1}^{K} \theta_k^i, \text{ for all } i = 1..N_q. \quad (17)$$

where $\beta > 0, \gamma > 0$ are empirical constants controlling the influence of the two components. In [6], it has been suggested to use a special form of the feedback signal v^i for derivation of the adaptation policy:

$$\begin{aligned} v^i &= 0.5[(1-\chi)\epsilon^i + (1+\chi)|\epsilon^i|], \quad (18)\\ \epsilon^i &= \rho_1 e^i + \rho_2 \dot{e}^i,\ \ \rho_1, \rho_2 > 0 \end{aligned}$$

e^i is deviation of the controlled signal from its desired value, $\chi, \rho_1, \rho_2 > 0$ are empirical constants.

To optimize the trained parameters $\boldsymbol{\Theta}$ in Eq.3, the cost function R_i is minimized by gradient descent:

$$\Delta\boldsymbol{\theta}_t^i = -\frac{dR^i}{d\boldsymbol{\theta}^i} = -\beta(\frac{\partial v^i}{\partial \boldsymbol{\theta}_k^i})^T v^i - \gamma\boldsymbol{1}, \quad (19)$$

The control $\boldsymbol{\tau}$, feedforward $\boldsymbol{u}$, and feedback signal $\boldsymbol{v}$ are linked: $\boldsymbol{\tau} = \boldsymbol{u} + \boldsymbol{v}$; see Eq.10. $\boldsymbol{\tau}$ represents the environment being learned and is assumed to be independent of $\boldsymbol{\Theta}$, therefore the adaptation law in Eq.19 can be rewritten as:

$$\Delta\boldsymbol{\theta}^i = \beta(\frac{\partial u^i}{\partial \boldsymbol{\theta}_k^i})^T v^i - \gamma\boldsymbol{1} = \beta\boldsymbol{\Phi} v^i - \gamma\boldsymbol{1}. \quad (20)$$

VIII. ACKNOWLEDGMENTS

This work was supported in part by the European Commission under contract numbers FP7-248258 (First-MM).

References

[1] D. Braun, A. Aertsen, D. Wolpert, and C. Mehring, “Learning optimal adaptation strategies in unpredictable motor tasks,” *The Journal of Neuroscience*, vol. 29, pp. 6472–6478, 2009.

[2] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, “Learning collaborative manipulation tasks by demonstration using a haptic interface,” in *Proceedings of the International Conference on Advanced Robotics*, 2009.

[3] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, “Teaching Physical Collaborative Tasks: Object-Lifting Case Study with a Humanoid,” in *Proceedings of IEEE International Conference on Humanoid Robots*, 2009.

[4] N. Hogan, “Impedance control: An approach to manipulation, part i-iii,” *ASME Journal of Dynamic Systems, Measurements, and Control*, vol. 107, pp. 1–24, 1985.

[5] D. Franklin, E. Burdet, K. Tee, R. Osu, C.-M. Chew, T. Milner, and M. Kawato, “CNS learns stable, accurate, and efficient movements using a simple algorithm,” *The Journal of Neuroscience*, vol. 28(44), p. 1116511173, 2008.

[6] G. Ganesh, A. Albu-Schaffer, M. Haruno, M. Kawato, and E. Burdet, “Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 2010.

[7] E. Gribovskaya, K. Zadeh, S. Mohammad, and A. Billard, “Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators,” *International Journal of Robotics Research*, 2010.

[8] S. M. Khansari-zadeh and A. Billard, “Imitation learning of Globally Stable Non-Linear Point-to-Point Robot Motions using Nonlinear Programming,” in *Proceeding of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[9] M. Khansari-Zadeh and A. Billard, “BM: An iterative method to learn stable non-linear dynamical systems with gaussian mixture models,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 2010.

[10] O. Khatib, “A unified approach for motion and force control of robot manipulators: the operational space formulation,” *IEEE Journal of Robotics and Automation*, vol. 3(1), pp. 43–53, 1987.

[11] J. Peters and S. Schaal, “Learning to control in operational space,” *The International Journal of Robotics Research*, pp. 197–212, 2008.

[12] E. Burdet, A. Codourey, and L. Rey, “Experimental evaluation of nonlinear adaptive controllers,” *IEEE Control Systems Magazine*, vol. 18, pp. 39–47, 1998.

[13] K. Astrom and B. Wittenmark, *Adaptive Control*. Addison-Wesley, 1989.

[14] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally weighted learning for control,” *Arificial Intelligence Review*, vol. 11, pp. 75–113, 1997.

[15] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation.” *IEEE Journal of Robotics and Automation*, vol. 3(1), pp. 1114–1120, 1987.

[16] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, “Variable impedance control a reinforcement learning approach,” in *Proceedings of International Conference on Robotics Science and Systems*, 2010.

[17] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, B. Siciliano, O. Khatib, and F. Groen, Eds. Spinger, 2008.

[18] S. Buerger and N. Hogan, “Complementary stability and loop shaping for improved humanrobot interaction,” *IEEE Transaction on Robotics*, vol. Vol.23, pp. 232–244, 2007.

[19] T. Worsnopp, M. Peshkin, M. Lynch, and J. Colgate, “Controlling the apparent inertia of passive human-interactive robots,” *Journal of Dynamics Systems, Measurement and Control*, vol. 128(1), pp. 44–52, 2006.

[20] D. Bullock, R. Bongers, and M. Lankhorst, “A vector-integration-to-endpoint model for performance of viapoint movements,” *Neural Networks*, vol. 12, pp. 1–29, 1999.

[21] E. Gribovskaya and A. Billard, “Combining dynamical systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot,” *Proceeding of IEEE/ACM International Conference on Human-Robot Interaction*, 2008.

[22] Y. Maeda, T. Hara, and T. Arai, “Human-robot cooperative manipulation with motion estimation,” in *Proceedings on International Conference on Intelligent Robots and Systems*, 2001.

[23] V. Duchaine and C. Gosselin, “General model of human-robot cooperation using a novel velocity based variable impedance control,” in *Proceedings of the Joint EuroHaptics Conference and Symposium*, 2007.

[24] D. Lee, C. Ott, and Y. Nakamura, “Mimetic communication model with compliant physical contact in Human–Humanoid interaction,” *International Journal of Robotics Research*, 2010.

[25] Z. Wang, A. Peer, and M. Buss, “An HMM approach to realistic haptic human-robot interaction,” in *Proceedings of the Joint EuroHaprics Conference and Symposium*, 2009.