

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

Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard

Abstract—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 that tunes the impedance parameters, so as to ensure accurate reproduction. To facilitate the illustration of the concepts introduced in this paper and provide a systematic evaluation, we present experimental results obtained with simulation of a dyad of two planar 2-DOF robots.

I. INTRODUCTION

We consider the problem of adapting on-the-fly the control law of a *robot-follower* when engaged in a collaborative task with another robot, the *robot-leader*. The task consists of lifting a rigid beam, Fig. 1. This work builds upon our

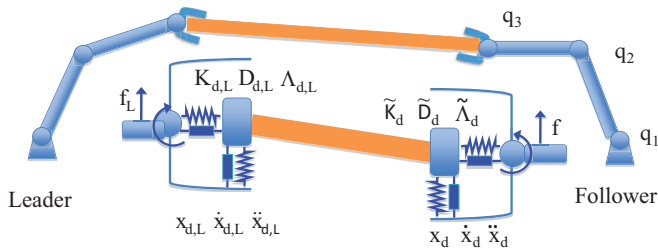


Fig. 1. Two planar robots are engaged in a collaborative task of lifting a beam. The robot-leader substitutes the human. The robot-follower anticipates the motion intentions of the robot-leader and adapts accordingly. Task completion requires the satisfaction of “soft constraints”: the two robots must coordinate and adapt their motions so as to avoid tilting the beam. The desired kinematic plan $x_{d,L}, \dot{x}_{d,L}, \ddot{x}_{d,L}$ of the robot-leader is given. During demonstration, the robot-follower learns to generate a desired kinematic command $x_d, \dot{x}_d, \ddot{x}_d$ in response to the perceived force f . The dynamical behavior of robots’ end-effectors is modeled as mechanical impedance characterized by desired stiffness, damping, and inertia. During task execution, the robot-follower adapts its desired stiffness \tilde{K}_d and inertia $\tilde{\Lambda}_d$, so as to ensure accurate reproduction of a learned task model.

previous research where we proposed a learning algorithm to control the motion of the HRP-2 robot engaged in a similar task with a human [1], [2]. These experiments have revealed how important it is to be able to anticipate the forces (applied at the robot’s end-effector), so as to generate an appropriate

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.

response. Furthermore, as prediction may not account for all perturbations, e.g. those induced by changes in behavior of the leading partner (sudden deviations from the motion plan or varying arm impedance), additional mechanisms are required to mitigate non-modeled effects during task execution.

The problem addressed in the current paper goes beyond that of a robot reacting to random disturbances and focuses on continuous prediction and adaptation to the dynamics of the partner. We decompose the problem in two parts. First, we learn a *task model* from demonstrations; the task model is then used to generate an appropriate reference kinematic and feedforward control signal in response to the perceived force. Importantly, the task model also allows to predict the force that will be perceived; the force prediction enables on-line adaptation of the robot’s motion. Second, we propose an *adaptive impedance controller* that compensates for non-modeled effects. Here, we extend the method proposed by [3] to encapsulate the force feedback error into the adaptation laws. We evaluate the system’s performance in controlled situations using a physical simulation of a pair of planar robots, where the *robot-leader* mimics the role played by the human in the real-world experiments. To simulate changes in the control law of the human (here the robot-leader), we introduce *delays*, *signal-dependent* noise, as well as change in the impedance of the controller of the follower.

II. APPROACH

The Fig. 2 shows the control flow of our model. We discuss how a task model is learned from a set of demonstrations, and present the adaptive impedance control law that allows for compensation of effects not captured by the learned model.

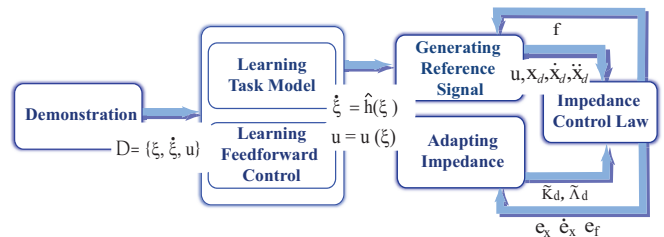


Fig. 2. After acquiring a set of demonstrations \mathcal{D} , the robot learns the task model $\hat{\xi} = \mathbf{h}(\xi)$ and a forward control signal $\mathbf{u} = \mathbf{u}(\xi)$ that maps the desired state ξ of the task model to actual motor commands. The robot is controlled through an *impedance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

A. Learning a Task Model

A training set \mathcal{D} consists of M demonstrated trajectories of length N^k , for $k = 1 \dots M$. Each trajectory is a sequence of states, ξ_t^k , state derivatives $\dot{\xi}_t^k$, and control inputs \mathbf{u}_t^k , $t = 1 \dots N^k$. The control input \mathbf{u}_t^k is the vector of joint torques. The task state is described by an *augmented state* $\xi = [\dot{x}_d, f_d]^T$ and is composed of the reference velocity \dot{x}_d , and the force feedback part that represents the expected force f_d , perceived by a force sensor mounted at the robot's wrist. Following our previous work [4], we assume that the task

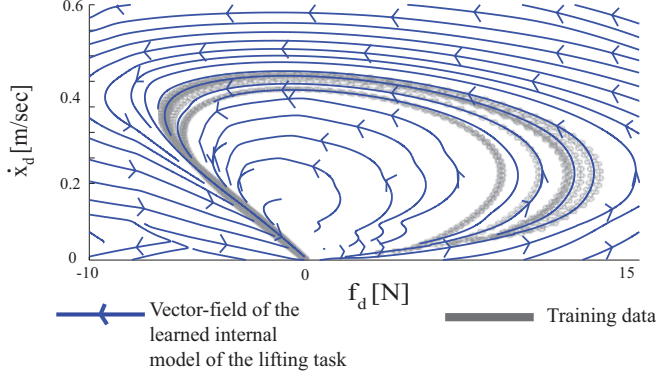


Fig. 3. The task model is represented by a dynamical system $\dot{\xi} = \mathbf{h}(\xi)$, $\xi = [\dot{x}_d; f_d]$ and estimated from the training data. At each time step, the velocity \dot{x}_d and force f_d are inferred from these observed at the previous step. Their dynamical relationships follow vector fields displayed in blue. Dark gray 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.

model is governed by an autonomous (time-independent) dynamical system, where the acceleration of the robot's end-effector is a function of its velocity and the perceived force¹:

$$\dot{\xi} = \mathbf{h}(\xi) + \eta(\xi) \sim \hat{\mathbf{h}}(\xi) = \mathbb{E}[\mathcal{P}(\dot{\xi}|\xi)] \quad (1)$$

where $\mathbf{h}(\xi)$ is the dynamic function governing the temporal evolution of the motion, and $\eta(\xi) \sim N(0, \Sigma_\eta(\xi))$ is the signal-dependent noise.

Given a training set \mathcal{D} (see Section III-A for a description of the data acquisition procedure), we learn the task model $\hat{\mathbf{h}}(\xi)$ by estimating the joint density $\mathcal{P}(\xi, \dot{\xi})$ through Gaussian Mixture Models; given the joint density, we compute the conditional expectation from Eq. 1. We apply the incremental EM procedure that we developed in [4] to ensure that the estimate of the velocity is asymptotically stable at the origin of the system. In this context, the stability ensures that when the force perceived at the end-effector is null, the robot stops moving, as shown in Fig. 3.

The important novelty here consists in introducing the augmented state ξ . The augmented state encapsulates both

¹Encoding 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: a single value of force corresponds to different velocities; see Fig. 3.

the kinematic command \dot{x} and the haptic input f and allows one to learn a temporal evolution of the reference velocity correlated with the perceived external force. An advantage of this formulation is that the robot can switch across different reference velocity profiles in response to a change in the partner's intentions.

Another advantage of this encoding is that the task model can be used to mitigate sensory delays by predicting the perceived force. Specifically, to generate a reference kinematics, the robot does not need to get actual value of the force at each time step, it can predict the perceived force from the task model. Later, once it gets actual value of the force, the robot may offset its prediction so as to switch to a different velocity profile if necessary.

B. Impedance Control

The rigid body dynamics in task coordinates [5]² is:

$$\begin{aligned} \Lambda(\mathbf{x}, \dot{\mathbf{x}})\ddot{\mathbf{x}} + \mu(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{J}^{-T}\mathbf{g}(\mathbf{x}) &= \mathbf{J}^{-T}\boldsymbol{\tau} + \mathbf{f}, \\ \Lambda(\mathbf{x}, \dot{\mathbf{x}}) &= \mathbf{J}^{-T}\mathbf{M}\mathbf{J}^{-1} \quad \mu(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{J}^{-1}(\mathbf{C} - \mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}})\mathbf{J}^{-1}, \end{aligned} \quad (2)$$

with $\mathbf{q} \in \mathbb{R}^{N_q}$ the vector of joint angles, \mathbf{M} the inertia matrix, \mathbf{C} the Coriolis/centrifugal matrix, $\mathbf{g}(\mathbf{x})$ the vector of gravity torques, \mathbf{f} is the vector of external forces and $\boldsymbol{\tau}$ are the applied joint torques, \mathbf{J} is the Jacobian.

Impedance control in the task space consists of the following control objective [6]:

$$\Lambda_d \ddot{\mathbf{e}}_x + \mathbf{D}_d \dot{\mathbf{e}}_x + \mathbf{K}_d \mathbf{e}_x = \mathbf{e}_f \quad (3)$$

where $\mathbf{e}_x = \mathbf{x} - \mathbf{x}_d$ is the position error between the actual position \mathbf{x} and the reference position \mathbf{x}_d ; $\mathbf{e}_f = \mathbf{f} - \mathbf{f}_d$ measures how much the actual perceived force \mathbf{f} deviates from the predicted one \mathbf{f}_d . Λ_d , \mathbf{D}_d , and \mathbf{K}_d are the symmetric and positive definite matrices of desired inertia, damping, and stiffness, respectively.

Substituting $\ddot{\mathbf{x}}$ from Eq. 3 into Eq. 2, the Cartesian impedance controller can be implemented via the joint torques $\boldsymbol{\tau}$ as follows:

$$\boldsymbol{\tau} = \mathbf{u} + \mathbf{J}^T \tilde{\mathbf{K}}_d \mathbf{e}_x + \mathbf{J}^T \tilde{\mathbf{D}}_d \dot{\mathbf{e}}_x + \mathbf{J}^T \tilde{\Lambda}_d \mathbf{e}_f, \quad (4)$$

$$\begin{aligned} \text{where } \mathbf{u} &= \mathbf{g} + \mathbf{J}^T (\Lambda \ddot{\mathbf{x}}_d + \mu \dot{\mathbf{x}}_d) \\ \tilde{\mathbf{K}}_d &= \Lambda \Lambda_d^{-1} \mathbf{K}_d, \quad \tilde{\mathbf{D}}_d = \Lambda \Lambda_d^{-1} \mathbf{D}_d + \mu, \quad \tilde{\Lambda}_d = \Lambda \Lambda_d^{-1} - \mathbf{I}. \end{aligned} \quad (5)$$

Eq. 4 can be rewritten as a sum of the feedforward and feedback components. The feedback signal can be decomposed into the kinematic feedback \mathbf{v}_x and the force feedback \mathbf{v}_f .

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{u} + \mathbf{v}, \quad \mathbf{v} = \mathbf{v}_x + \mathbf{v}_f \\ \mathbf{v}_x &= \mathbf{J}^T (\tilde{\mathbf{K}}_d \mathbf{e}_x + \tilde{\mathbf{D}}_d \dot{\mathbf{e}}_x), \quad \mathbf{v}_f = \mathbf{J}^T \tilde{\Lambda}_d \mathbf{e}_f. \end{aligned} \quad (6)$$

Note that the feedforward control \mathbf{u} in Eq. 4-5 corresponds to the learned feedforward control in Eq. 7. Analytical computation of \mathbf{u} through Eq. 5 requires an accurate dynamic model of the robot that is rarely available. Many authors suggest different algorithms for approximating the feedforward

²The notation $^{-T}$ refers to pseudo-inverse of a transposed matrix. The notation $^{-1}$ refers to pseudoinverse if a matrix is noninvertible

control, we learn \mathbf{u} from demonstrations. Therefore, if we further refer to feedforward control \mathbf{u} , we mean the learned control \mathbf{u} from Eq. 7.

During motion, most of variation occurs along the vertical axis, therefore only the impedance parameters along the vertical axis are estimated. Following common practice, we further set $\tilde{D}^j = \lambda\sqrt{\tilde{K}^j}$; λ is a scalar, and $\lambda = 2$ in our experiments.

C. Learning Feedforward Control

The feedforward control input \mathbf{u} is generally a function of the robot's desired and actual states. Given a task model $\hat{\mathbf{h}}(\boldsymbol{\xi})$, which defines the nonlinear dependency between these parameters, \mathbf{u} becomes a function of the task state $\boldsymbol{\xi}$. Following [3], we learn an estimate of the feedforward command \mathbf{u} in the linear form:

$$\mathbf{u} = [\Phi(\boldsymbol{\xi})^T \Theta]^T, \quad (7)$$

where $\Phi \in \mathbb{R}^{K(N_x+N_f)}$ is a vector of G basis functions and $\Theta \in \mathbb{R}^{K(N_x+N_f) \times N_q}$ is a matrix of the tunable parameters (each column θ^i , $i = 1 \dots N_q$, of the matrix Θ 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 the joint space, respectively. This type of linear control parameterization is commonly used in the adaptive control literature [7], [8]. The basis $\Phi(\boldsymbol{\xi})$ consists of G Gaussian functions:

$$\Phi_j = \Phi(\boldsymbol{\xi})_j = \frac{\pi_j(\boldsymbol{\xi}) \boldsymbol{\xi}}{\sum_{g=1}^G \pi_g(\boldsymbol{\xi})}, \quad j = 1 \dots G \quad (8)$$

$$\text{where } \pi_j(\boldsymbol{\xi}) = \exp(-0.5(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi},j})^T \Sigma_j^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi},j}))$$

with $\boldsymbol{\mu}_{\boldsymbol{\xi},j}$, Σ_j are the mean and the diagonal covariance of a j th Gaussian kernel. Given \mathcal{D} , we learn the parameterization in Eq.7-8 through Linear Weighted Regression [9].

D. Adaptive Impedance

We use an iterative algorithm to update the impedance parameters and the parameters of the feedforward controller. We follow [3] and write an adaptation law for the feedforward parameters θ^i ($\forall i = 1 \dots N_q$) from Eq. 7 as follows:

$$\Delta \theta^i = \frac{\beta}{2}((1 - \chi)\Phi \epsilon^i + (1 + \chi)\Phi |\epsilon^i|) - \gamma \mathbf{I} \quad (9)$$

where ϵ is the error function, that will be defined later and β , χ , and γ are empirical constants. During physical interaction, a robot may deviate from the learned task model because of varying human intentions. Hence, we introduce two supplementary feedback terms, \mathbf{e}_f and \mathbf{e}_m to the original formulation in [3]. These terms denote respectively an error between the predicted and the actual perceived force and a kinematic error, e.g. that caused by changes in the mass of a manipulated object. That is:

$$\begin{aligned} \boldsymbol{\epsilon} &= J^T(\mathbf{e}_m + \rho_3 \mathbf{e}_f) \\ \mathbf{e}_m &= \ell \min(\mathbf{f} \cdot \dot{\mathbf{e}}_x, 0)(\rho_1 \mathbf{e}_x + \rho_2 \dot{\mathbf{e}}_x) \end{aligned} \quad (10)$$

where \cdot refers to the inner vector product and $\ell = \|\mathbf{f} \cdot \dot{\mathbf{e}}_x\|^{-1}$ is the normalization factor. The parameters $\rho_1, \rho_2, \rho_3 \in \mathbb{R}$

weight the importance given to errors in the trajectory, the velocity, and the force. We can now rewrite Eq. 9 as:

$$\begin{aligned} \Delta \theta^i &= \frac{\beta}{2}(1 - \chi_1)\Phi J^T \mathbf{e}_m + \frac{\beta}{2}(1 + \chi_1)\Phi |J^T \mathbf{e}_m| + \\ &\frac{\beta}{2}(1 - \chi_2)\Phi J^T \mathbf{e}_m + \frac{\beta}{2}(1 + \chi_2)\Phi |J^T \mathbf{e}_m| - \gamma \mathbf{I}. \end{aligned} \quad (11)$$

The bio-mimetic approach [3] suggests to transform Eq. 11 into adaptation laws for both feedforward and feedback components. Next, we provide analytical discussion of components in Eq. 11 and propose the algorithm for adaptation of the impedance parameters.

In Eq. 11, the terms $\frac{\beta}{2}(1 - \chi_1)\Phi \epsilon_m^i$ and $\frac{\beta}{2}(1 - \chi_2)\Phi \epsilon_f^i$ correspond to the conventional adaptation law from the control literature. $\frac{\beta}{2}(1 - \chi_1)\Phi \epsilon_m^i$ generates a force in the direction opposite to the kinematic error \mathbf{e}_m , and updates the feedforward signal \mathbf{u} . $\frac{\beta}{2}(1 - \chi_2)\Phi \epsilon_f^i$ compensates for the deviation of the external force from its reference value and contributes to the adaptation of the desired inertia.

The terms dependent on the absolute values of the errors aim at tuning the stiffness. $\frac{\beta}{2}(1 + \chi_1)\Phi |\epsilon_m^i|$ increases stability in response to kinematic perturbation, while $\frac{\beta}{2}(1 + \chi_2)\Phi |\epsilon_f^i|$ decreases the stiffness if the deviation of the external force is increasing. Indeed, a sudden increase in the force error \mathbf{e}_f means 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 [3], the update mechanism emulates automatic relaxation through the term $\gamma \mathbf{I}$. 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 [10]. The authors in [3] advocate that the components involving absolute values of the errors are responsible for muscle "co-activation" and should affect adaptation of impedance in robots. Following this reasoning, we rearrange the terms in Eq. 11; the update procedure for the forward signal and the impedance parameters then can be written as follows:

$$\Delta \theta^i = \kappa_x \Phi (J^T \mathbf{e}_m)^i - \gamma_{\theta}, \quad \kappa_x > 0, i = 1 \dots N_q \quad (12)$$

$$\Delta \tilde{K}_d^j = \beta_x |e_m^j| - \beta_f |e_f^j| - \gamma_{\tilde{K}}, \quad \beta_x, \beta_f, \gamma > 0. \quad (13)$$

$$\Delta \tilde{\Lambda}_d^j = \kappa_f e_f^j - \gamma_{\tilde{\Lambda}}, \quad j = 1 \dots N_x \quad \kappa_f > 0 \quad (14)$$

Eq. 4 together with Eq. 7,12-14 represent the control algorithm that enables the simultaneous on-line adaptation of the feedforward signal, the desired stiffness, and the inertia. Note, if we update the desired stiffness and inertia according to Eq. 13-14, their values may go below zero; to avoid this, we consider adaptation only within given boundaries, e.g., $2 < K_d^j < 100$.

III. RESULTS

We assess our method in simulations where the robot-follower interacts with another planar robot, Fig. 1. 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.

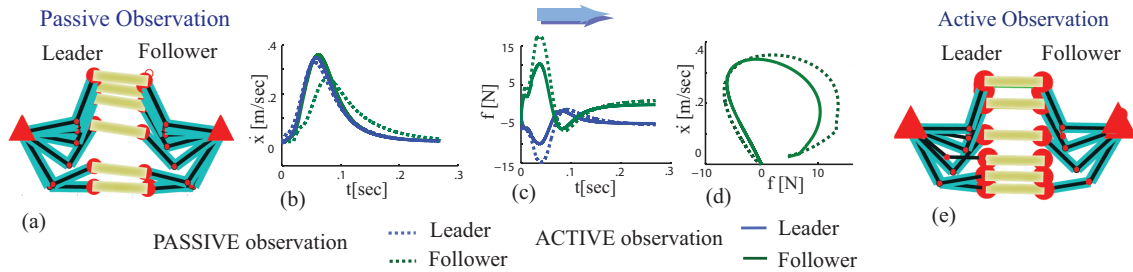


Fig. 4. TWO-STAGE TRAINING PROCEDURE. To simulate real-world training, where the robot is teleoperated by a human, we adopt a two stage training procedure. Figs. (a), (e) present the robots' configurations during training. Desynchronization between the partners is greatly reduced during active observation, 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 Fig. (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 Figs. (c)-(d). The robot-follower perceives high positive external forces that 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 kinematic profile acquired during passive observation. Improved synchronization decreases the magnitude of the forces perceived by both partners; see Figs. (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 used for validation of 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 have been acquired. This procedure is inspired from the way humans incrementally learn to synchronize with each other. The leader keeps the same pace in passive and active stages. The follower adapts its desired kinematics to synchronize with the leader.

We simulate two types of sensorimotor limitations of the human motor system: a signal-dependent noise and sensory delay. An *reaction* delay (a time span between the moments when the follower starts perceiving the changing force and when it actually starts moving) is assigned to be 150ms; the *perception* delay along the motion (delay between perception of a force and reaction) is 3ms.

During training the robots are controlled with the impedance control law according to Eq. 4 with the pre-defined impedance parameters and the zero reference force 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 robots, the reference kinematics are generated by a dynamical system parameterized with a multiplicative parameter. Specifically, we use the VITE dynamical model of human reaching motions [11]: $\ddot{x}_d = \alpha(-\dot{x}_d + 4(x_{tar} - x_d))$, where α is the multiplicative parameter, x_{tar} is the given target location. The dynamical system produces the same task space trajectories but with different velocity profiles³.

To provide examples of adaptation to different velocities, the parameter α of the 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 0.2-0.5m/s and duration of 0.4-0.8s. The α of the follower is fixed so to generate the reference motion of 0.8s with the maximum velocity 0.2m/s.

1) *Passive Observation*: The two robots are controlled to track their reference kinematic 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 Fig. 4-(b). This requires the leader to inject a considerable amount of energy; see Fig. 4-(c),(d), dashed line. Therefore, even though the follower tracks the motion of the leader (the actual velocity of the follower is close to that of the leader), the perceived forces are different from those that would be observed if it did so intentionally. To observe these forces, the follower reproduces the actual kinematics during the next training step.

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

By deliberately reproducing this imposed kinematic profile, the follower generates forces that are better aligned with those of the leader. The leader injects less energy and, therefore, the forces perceived by the follower are smaller than those at the first stage; see Fig. 4-(c),(d), solid line. Fig. 4-(a),(e) highlights improvements in synchronization across the two stages⁴. The collected data (actual velocity, forces, and the feedforward commands of the follower) are further used to learn the task model \hat{h} and the feedforward control u .

⁴During active observation the decrease in the perceived forces is mainly caused by an improved synchronization. But the lowered stiffness of the robot-leader also contributes to this decrease. Indeed, the robot-leader does not forcefully guide the robot-follower.

³A model parameterized with $\alpha = 10$ produces the same trajectories in the state-space $\{x_1; x_2\}$ as a model parameterized with $\alpha = 20$, however the latter converges to the target twice as fast.

B. Learning a Task Model

The acquired training data are depicted in Fig. 5-(a),(b); note that the data exhibit a force-velocity correlation, this correlation is similar to the one observed in real-world data acquired with the HRP-2 robot (see [2]). Importantly, the

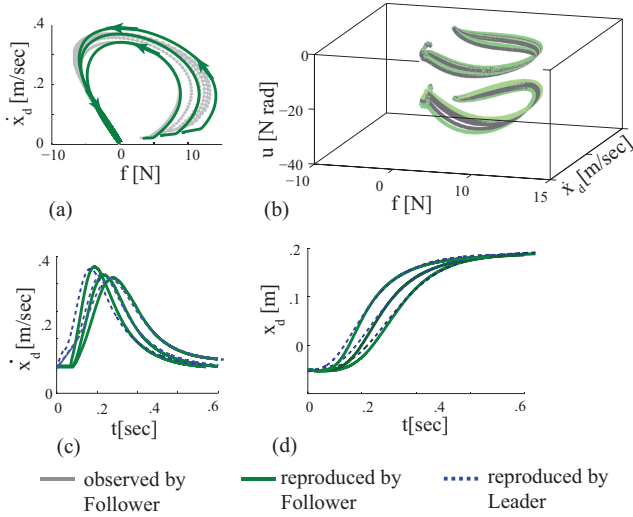


Fig. 5. TASK LEARNING AND REPRODUCTION. In this experiment, the robot-follower learns the lifting task by observing demonstrations performed with different velocity profiles imposed by the leader. During reproduction, the robot-leader varies its kinematics plan from one attempt to another; it does so by changing motion duration and maximum velocity. The robot-follower, governed by the learned task and control model, adapts its motion and successfully accomplishes the task. (a) The state-space view of the data used for training a task model (gray line) and the reproduction attempts (green line). (b) The forward control signal generated by the follower during demonstration (gray 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. The follower (green line) adjusts its kinematics and synchronizes with the leader (dashed blue line).

force-velocity dependency defines a task model: how to generate a motion that is consistent with perceived force; i.e. for a given value of the perceived force, the robot estimates the relevant velocity. After data acquisition, the robot learns the task model $\dot{\xi} = \mathbf{h}(\xi)$ and the forward control model $\mathbf{u} = \mathbf{u}(\xi)$. During reproduction, the learned models are fed into the control law in Eq. 4. The results of the task execution are depicted in green in Fig. 5. Note that the robot-follower successfully adapts its velocity and synchronizes with the robot-leader.

C. Adaptation of Unknown Impedance

In the previous experiments we reused the impedance parameters that 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, i.e. during task reproduction, Fig. 6.

The experiment starts with the robot-follower's stiffness and inertia set to mid-range values of 35N/m and 0 respec-

tively (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 Fig. 6-(a), dashed line, that reproduction without adaptation of the impedance parameters leads to an overestimated reference velocity and difficulties to stop at the target (oscillations in the velocity profile). Adaptation allows for tuning of the parameters so as to ensure stable interaction; see Fig. 6-(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 Fig. 6-(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 Fig. 6-(c).

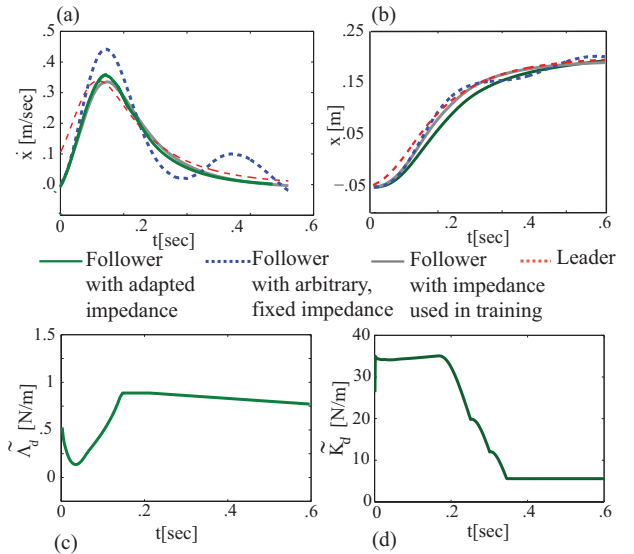


Fig. 6. ADAPTATION OF UNKNOWN IMPEDANCE. In general, the impedance parameters that would be optimal for the task are unknown. Therefore, the follower should tune its stiffness and inertia during task execution. The arbitrary impedance parameters may cause undesirable effects, e.g. overestimated reference kinematics and contact instabilities (blue dashed line). Our algorithm provides the adaptation law, to tune the parameters and to ensure accurate reproduction (green lines in (a)-(b) show the results with impedance adaptation).

D. Adaptation to Perturbations

We tested the ability of the learned model to adapt to changing intentions of the leader during task execution. We simulated uncertainties about the motion objectives by varying the target position \mathbf{x}_{tar} in the motion plan of the leader. Three cases are considered where the leader changes the motion plan during task execution. It decides to move the beam (1) higher than it has planned initially, (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. Fig. 7 shows that the follower succeeds in bringing the beam to the new desired location. Such an adaptation is possible because the learned model generalizes the force and velocity patterns to values not observed during training (as discussed in Fig. 3).

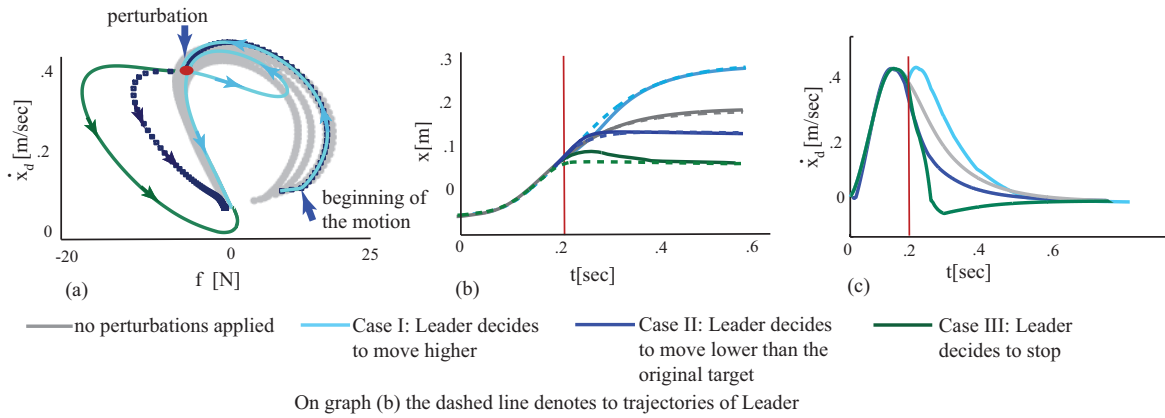


Fig. 7. ADAPTATION TO PERTURBATIONS. Three cases are analyzed: the robot-leader changes the motion plan on-line and move the beam (1) higher than initially planned, (2) lower, but higher than the actual position of the beam at the moment the change decision is taken, and (3) stops at a position lower than the actual position at the decision-taking moment. In case (1), the robot-follower manages to re-accelerate (see the two peaks in the velocity profile). In cases (2) the robot-follower pro-actively decelerates. In case (3), the robot also manages to smoothly drop velocity below zero and lower the beam.

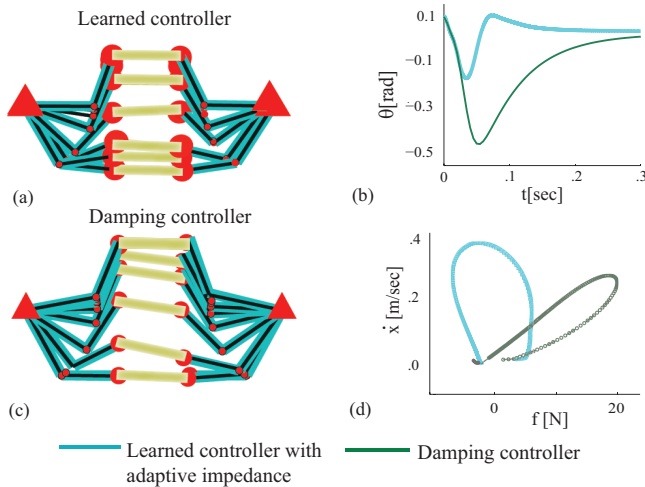


Fig. 8. COMPARISON WITH A DAMPING CONTROLLER. We compare our system versus the damping controller. (a), (c) The planar robots performing lifting; the follower is controlled by our controller (a), and the damping controller (c). One may notice the improvements in coordination between the partners when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

E. Comparison with a Damping Controller

To highlight the advantages of the proposed approach for controlling a robot during collaborative manipulation tasks, we also implemented a damping controller [12]. This easy to 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 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 our system versus the damping controller in Fig. 8. The forces perceived by the robot are much higher than those observed with our learned system (Fig. 8-(d)). This is due to the higher forces that the leader has to apply to maintain the interaction. The beam undergoes stronger rotation (Fig. 8-(b)) due to the imbalanced forces on its two sides. This will be highly undesirable if the beam is loaded with unfixed objects that may slip down. However, we should emphasize, that our controller possesses local knowledge and far from the demonstrated data, its prediction is irrelevant. In this case, we suggest switching to a damping controller to ensure safety.

IV. DISCUSSION AND RELATED WORK

While interaction control has been investigated since long, the available tools remain 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 [13] a robot adjusts its damping depending on the perceived force; in [12] it tries to predict the target of a motion so as to generate a reference trajectory proactively. Although their validity is confirmed with successful experimental results, they are still ad-hoc solutions: no generic framework for tackling both task learning and the variable impedance during physical interaction exists. In robot learning, the few existing works consider a single learned trajectory as a reference signal for a hard-coded impedance controller. In [14], 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. The considered scenario does not require continuous interaction and haptic signals do not effect the reference kinematics. A hand-shaking robot is presented in [15]. 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 to recognize which motion model to choose. Recognition happens at the onset of the motion and governs the robot through the rest of the task without adaptation. This is different from our approach that advocates continuous adaptation of the motion.

V. 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. The presence of a human in the loop makes it complicated to conduct a formal stability analysis of the adaptive impedance. Indeed, the prove of stability should demonstrate that the errors e_x , \dot{e}_x , and e_f converge to zero with time. Human brings uncertainty into errors and, therefore, the stability of the system cannot be proved unless hypotheses are made on human behavior. To ensure safety during interaction, we incorporate several security mechanisms that bound the force produced by the robot.

Here, we consider a non-redundant case; the method is also applicable to redundant set-ups; one plausible way is to follow a projection based approach [16]. Currently, we are working towards implementing the algorithm on the physical WAM robot and consider higher dimensional tasks.

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. The simulator offers a framework for systematical assessment and performance comparison of different algorithms for control of human-robot interaction.

APPENDIX

We briefly summarize the major concepts of the bio-mimetic adaptive algorithm presented in [3], for details, please, refer to the original publication. The following cost function that penalizes the feedback cost and activation of the feedforward command is:

$$\min_{\theta^i} R^i(\theta^i) = 0.5\beta (v^i)^2 + \gamma \sum_{k=1}^K \theta_k^i, \quad \forall i = 1 \cdots N_q. \quad (15)$$

where $\beta > 0, \gamma > 0$ are empirical constants controlling the influence of the two components. In [3], it is suggested to use a special form of the feedback signal v^i for derivation of the adaptation policy:

$$v^i = 0.5[(1 - \chi)\epsilon^i + (1 + \chi)|\epsilon^i|], \quad \epsilon^i = \rho_1 e^i + \rho_2 \dot{e}^i, \quad (16)$$

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 Θ in Eq.7, the cost function R_i is minimized by gradient descent:

$$\Delta \theta_t^i = -\frac{dR^i}{d\theta^i} = -\beta \left(\frac{\partial v^i}{\partial \theta_k^i} \right)^T v^i - \gamma \mathbf{I}, \quad (17)$$

The control τ , feedforward u , and feedback signal v are linked: $\tau = u + v$; see Eq.6. τ represents the environment being learned and is assumed to be independent of Θ , therefore the adaptation law in Eq.17 can be rewritten as:

$$\Delta \theta^i = \beta \left(\frac{\partial u^i}{\partial \theta_k^i} \right)^T v^i - \gamma \mathbf{I} = \beta \Phi v^i - \gamma \mathbf{I}. \quad (18)$$

VI. ACKNOWLEDGMENTS

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

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] E. Gribovskaya, K. Zadeh, S. Mohammad, and A. Billard, "Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators," *International Journal of Robotics Research*, vol. 1, 2011.
- [5] 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.
- [6] 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.
- [7] E. Burdet, A. Codourey, and L. Rey, "Experimental evaluation of nonlinear adaptive controllers," *IEEE Control Systems Magazine*, vol. 18, pp. 39–47, 1998.
- [8] K. Astrom and B. Wittenmark, *Adaptive Control*. Addison-Wesley, 1989.
- [9] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, pp. 75–113, 1997.
- [10] 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), pp. 11165–11173, 2008.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in Human-Humanoid interaction," *International Journal of Robotics Research*, 2010.
- [15] Z. Wang, A. Peer, and M. Buss, "An HMM approach to realistic haptic human-robot interaction," in *Proceedings of the Joint EuroHaptics Conference and Symposium*, 2009.
- [16] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, B. Siciliano, O. Khatib, and F. Groen, Eds. Springer, 2008.