

Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction

Klas Kronander and Aude Billard

Abstract—Robot Learning from Demonstration (RLfD) has been identified as a key element for making robots useful in daily lives. A wide range of techniques has been proposed for deriving a task model from a set of demonstrations of the task. Most previous works use learning to model the kinematics of the task, and for autonomous execution the robot then relies on a stiff position controller. While many tasks can and have been learned this way, there are tasks in which controlling the position alone is insufficient to achieve the goals of the task. These are typically tasks that involve contact or require a specific response to physical perturbations. The question of how to adjust the compliance to suit the need of the task has not yet been fully treated in Robot Learning from Demonstration. In this paper, we address this issue and present interfaces that allow a human teacher to indicate compliance variations by physically interacting with the robot during task execution. We validate our approach in two different experiments on the 7 DoF Barrett WAM and KUKA LWR robot manipulators. Furthermore, we conduct a user study to evaluate the usability of our approach from a non-roboticists perspective.

Index Terms—Robot Learning from Demonstration, Compliant Control, Physical Human-Robot Interaction, Haptic feedback, Tactile interfaces

1 INTRODUCTION

HUMANS demonstrate an impressive capability to manipulate fragile objects without damaging them, graciously controlling the force and position of hands or tools. Traditionally, robotics has favored position control over force control to produce fast, accurate and repeatable motion. In well organized work-cells, position controlled robots excel in repetitive tasks, e.g. in the auto manufacturing industry. However, for extending the applicability of robotic manipulators outside the strictly controlled environments of industrial work cells, position control is inadequate. In tasks which involve contact with objects whose positions are not known with perfect certainty, the use of pure position control can cause unbounded rise of the contact force, ultimately leading to dangerous behaviors such as breakage or unstable control. For such tasks, smooth and compliant manipulation requires a controller that regulates the relationship between positional deviations and forces on the robot rather than stiffly following a position or force trajectory.

There are different ways to achieve compliant manipulation, depending on the task representation and

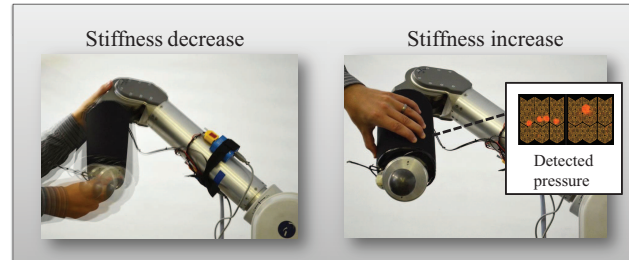


Fig. 1: The figure shows the principles behind the proposed interface. The teacher interacts with the robot as it is moving to alter its stiffness. To decrease the stiffness the teacher wiggles the robot around its current position (left figure). To increase the stiffness, the teacher increases the grip force with which he holds the robot (right figure). The robot responds online to these stimuli, so that the teacher gets immediate haptic feedback on the new stiffness.

control system of the robot. For torque controlled robots, compliance can be described and controlled either in joint coordinates or Cartesian coordinates [1]. Recently, there has been a strong interest in developing robots with inherently compliant joints [2]. These robots have the advantage of being safer than robots for which compliance is achieved actively, since the robot remains compliant even in the case of sensor failures etc. Furthermore, these mechanisms allow energy to be stored which has been demonstrated to be very useful for explosive tasks such as hammering [3]. Regardless of how the physical compliance is achieved (joint space/task space and active/passive), the compliance should be chosen to suit the needs of the task at hand. This is often a hard problem, and previous approaches typically need accurate models of the interaction environment [4], [5], [6]. Furthermore, the task may require that these parameters change during the task execution. In this work, we treat this problem in a Learning from Demonstration framework. To this end, we propose teaching interfaces that allow to change the stiffness of the robot by physically interacting with it. Using these interfaces, the teacher can teach the robot how it should vary its stiffness during the task. These interfaces work both in task space and joint space.

In our previous work [7], we presented a system for teaching Cartesian space stiffness variations. With that system, the user could modify the compliance of the robot online. To reduce the stiffness of the robot

in a particular direction, the teacher would wiggle the end-effector along that direction. The robot would then respond immediately by reducing the stiffness proportionally to the amplitude of the wiggling. In this work, we improve that interface by introducing frequency domain separation of the measured interaction, where only interaction in a selected band have an effect on the stiffness. This has the benefits of reducing the effect of interaction components which arise from contact with the environment and that it allows the teacher to use slow interaction to feel the stiffness of the robot without actively changing it. In this work, we also generalize the interface from [7] so that it can be used for stiffness modulation both in Cartesian and joint space. We further extend our previous work by introducing a new mode of interaction for *increasing* the stiffness. To this end, the robot monitors the pressure with which the teacher grasps the robot arm, and uses high grasping force as an indication to increase the stiffness, see Fig. 1. In [7], the position command came from prerecorded reference trajectories. Both the desired trajectory and stiffness variations were time-dependent. In this work, we show that our techniques for teaching stiffness variations can be used together with autonomous dynamical systems for modeling the positional aspects of the task, while encoding the stiffness variations as a function of the position of the robot rather than time. Which of the time-dependent or time-independent encodings to use depends on the task. Specifically, for tasks that require stiffness variations to occur in certain physical configurations, a time-independent encoding is advantageous since it is robust to perturbations delaying the execution of the task. Summarizing, this paper extends [7] in the following ways:

- 1) Frequency domain separation of the measured interaction.
- 2) Modified version of the interface for teaching joint space stiffness variations.
- 3) The introduction of a new mode of interaction for *increasing* the stiffness.
- 4) Learning stiffness variations as a function of the robot position in place of time.

We demonstrate the proposed approach for learning compliant manipulation in two tasks. The first experiment consists in teaching a robot to vary its compliance when pouring a drink, so that moderate perturbations can be absorbed without spilling. Here, the proposed system for decreasing and increasing stiffness in Cartesian space is demonstrated. The second experiment demonstrates the system for teaching joint space stiffness variations in a match-lighting task. The results show that by teaching a varying stiffness profile, the success-rate when lighting a match is greatly increased compared to both fixed low and fixed high stiffness. Furthermore, the efficiency of the proposed teaching interfaces is evaluated in a user study, where subjects were asked to teach the robot a stiffness for improving the performance

of a drawing task.

This paper is structured as follows: Section 2 provides an overview of related research, Section 3 introduces the controllers that are used in this work. This is followed by a detailed presentation of the teaching interfaces in Section 4. In Section 5, we present two examples of task learning that use the proposed interfaces together with different learning algorithms. Section 6 presents the user study and the results obtained from it. The paper is concluded with a discussion and an outlook into future directions of research in Section 7.

2 RELATED WORK

Robot Learning from Demonstration has been identified as a key element for making robots useful in everyday situations [8][9]. During the last decade, extensive efforts have been made for developing systems for learning various aspects of manipulation tasks. With a few exceptions, previous works in RLfD for manipulation ultimately rely on a stiff position controller for executing the task that has been learned. For tasks involving free motion and simpler tasks involving contact, this may be sufficient. However, for tasks requiring specific forces to be exerted on the environment, or tasks involving contact as important elements, the use of stiff position control is insufficient [4].

Calinon et al. [10] proposed to derive stiffness variations for a compliant controller from the kinematic demonstrations of the position trajectories. A probabilistic model is fit to the demonstrated trajectories, and the stiffness profile is shaped so that the robot adopts a high stiffness in directions of low variance. This approach is attractive in that the stiffness does not have to be demonstrated separately. The stiffness determines how the robot responds to unexpected perturbations, so the assumption that variability of position and stiffness are directly related in the positional demonstrations may not always be reasonable. In our approach, we take inspiration from this work by mapping variability in the interaction to reduced stiffness in our teaching interface. The stiffness profile is however learned independently and does not depend on the positional demonstrations.

In [11], the same approach for setting the stiffness as in [10] was used. Furthermore, the robot learned a model of the contact forces that should arise in correct manipulation. First, the positional profile of the task was taught using kinesthetic teaching. When the positional profile for the task was learned, the teacher demonstrated the interaction forces using a haptic device while the robot was executing the learned motion. A model describing the desired contact forces was then built, and used by the robot to determine the desired force during task execution. The robot was then controlled by adding the desired contact force to the output of the compliant position controller. Another approach for learning interaction forces from demonstration is presented in [12]. In that work, the force profile for the

task was described by goal-directed dynamical system and the task was executed using the parallel approach to force/position control [13]. In [12], there is no learning of stiffness, and in [11] and [10] the stiffness is learned from kinematic data, with no feedback provided to the teacher during training. Our work is thus complementary to these as we propose to learn stiffness variations from demonstrations, with haptic feedback given to the teacher during the demonstration process. It should be emphasized that for demonstrating force or impedance trajectories, it is crucial to provide haptic feedback to the teacher, since it is not generally possible to demonstrate meaningful force and stiffness variations without feeling the contact and/or resistance. In both [11] and [12] the demonstrations of the interaction forces are given using external haptic devices. In our work, the teacher indicates stiffness variations by physically interacting with the robot. The robot updates its stiffness online, and thus acts as a haptic display for the teacher who can immediately feel the effect of the interaction. The use of an external haptic device is thus not necessary.

In our previous work [14] we took a different approach at incorporating force information in RLfD. That work developed a controller learned from demonstration that allowed human robot collaboration for lifting a beam. A teacher provided demonstrations of desired velocity response to force stimuli while a human was performing the collaborative lifting with the robot. The perceived force/torque at the end-effector was then encoded together with the demonstrated velocity in a Gaussian Mixture Model (GMM), and after learning the robot could perform the collaborative lifting by conditioning the GMM on the perceived force/torque and generate the velocity response using Gaussian Mixture Regression (GMR). This work was extended in [15], where the apparent inertia of the robot was varied using adaptive control to increase the performance of the collaborative task. The concept of using force as stimuli was also used in [16] for teaching motions based on haptic input. The perceived force/torque was displayed to the teacher using a haptic device, and the teacher demonstrated motion response. These works use force as stimuli for generating velocity, and are thus different from [12] and [11] which include desired interaction forces as outputs from the task model. In contrast to all of these works, we take the perspective that stiffness variations can be crucial to successful task execution in many situations. The possibility to teach stiffness variations is not only relevant from a task performance perspective, but also for safety in situations where the robot works closely to humans.

In [17], an incremental approach for adjusting the position profile encoded in a Hidden Markov Model is presented. A specialized impedance controller with a refinement tube is presented, which allows for control with partial compliance so that the teacher can refine the motion by physical coaching. Note that while [17] uses impedance control in their system, no learning of

task-appropriate impedance is involved.

Learning variable impedance control policies has been formulated as an optimal control problem [18],[19]. These works specify the task constraints as a cost function and optimize the control actions subject to the dynamics of the robot. This has the advantage that the impedance profile is tailored to each robotic platform. The cost functions used in the optimization typically include a task performance term and an energy term. [20] further uses inverse optimal control to infer a task-based cost-function in order to transfer variable impedance policies between different systems. Solutions based on optimal control theory for varying the stiffness to maximize link velocity of VSA systems has been reported by [3] and [21]. Closely related to optimal control is reinforcement learning (RL), which has been used for learning variable impedance policies in [22]. In [23], an EM-based reinforcement learning algorithm initialized by human demonstrations is presented. With task-specific cost-functions, these algorithms are demonstrated to tune the stiffness for high task performance in various tasks, e.g. via-point trajectory following and flipping a light switch [22] and flipping pancakes [23]. These works move RL in the same direction as we wish to move RLfD in that they seek to improve task performance by controlling the impedance of the robot in addition to its position/velocity. A full discussion of the progress in RL for variable impedance control is outside the scope of this article.

Humans control the impedance of their limbs through co-contraction of agonist-antagonist muscle pairs. An apparatus for measuring human limb stiffness was presented in [24] and used for examining hypothesis regarding human arm movement control. Similar setups have since then been used in a number of experiments, including [25] which reports that humans modulate stiffness to deal with instabilities due to interactions. A controller that adapts position trajectory, feed-forward force and impedance according to principles derived from human studies was presented in [26]. In contrast to transferring the principles of human impedance adaptation is transferring impedance strategies for specific tasks. The strategies used by humans to modulate impedance are interesting and may provide valuable insight for selecting impedance for robots. In [20], human impedance estimation via EMG is used to derive optimization criteria that can explain the stiffness modulation strategy chosen by the human. This strategy is then transferred to the robot by performing optimization on the resulting cost-function subject to the dynamics of the robot. In contrast to that work, we do not use the human as a model for the controller, but rather as a teacher which is assumed to be capable of selecting an impedance which is appropriate *for the robot and the task at hand*.

3 CONTROLLERS

Compliant control can be achieved in different ways, depending on the platform and task space representa-

tion. In this paper we work with torque-controlled manipulators. We thus achieve compliance actively. Below, we describe the Cartesian controller and the joint space controller that are used in this work, highlighting the parameters which are learned through interaction with the teacher.

3.1 Cartesian Impedance Control

For robot position or velocity control, depending on the task it may be easier to describe the motion in Cartesian space (CS) rather than the robot joint space (JS). Similarly, describing the desired compliance of the robot is most often (but not always) easier in CS. The desired compliance of the end-effector in CS can be described in terms of a mechanical impedance, relating the desired wrench response $\mathbf{F} \in \mathbb{R}^6$ of the robot end-effector to a deviation $\Delta \mathbf{x} \in \mathbb{R}^6$ from the desired pose:

$$\mathbf{F} = \mathbf{M}\Delta\ddot{\mathbf{x}} + \mathbf{D}\Delta\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} \quad (1)$$

Here, $\mathbf{M}, \mathbf{D}, \mathbf{K} \in \mathbb{R}^{6 \times 6}$ respectively denote the desired inertia, damping and stiffness of the robot end-effector. By choosing the desired inertia to the intrinsic inertia of the robot in its current state, the control objective (1) can be achieved without the need of force-feedback in the control loop [1] with the following controls:

$$\mathbf{F} = \mathbf{K}\Delta\mathbf{x} + \mathbf{D}\Delta\dot{\mathbf{x}} \quad (2a)$$

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} + \boldsymbol{\tau}_F \quad (2b)$$

where $\boldsymbol{\tau}$ represents the vector of commanded torques, \mathbf{J} represents the manipulator Jacobian and $\boldsymbol{\tau}_F$ represents a feedforward command containing e.g. gravity and friction compensation torques¹. The stiffness matrix \mathbf{K} consists of four sub-matrices:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_P & \mathbf{K}_{PR} \\ \mathbf{K}_{RP} & \mathbf{K}_R \end{bmatrix} \quad (3)$$

with $\mathbf{K}_P \in \mathbb{R}^{3 \times 3}$ relating forces to positional errors, $\mathbf{K}_{PR} \in \mathbb{R}^{3 \times 3}$ relating forces to rotational errors, $\mathbf{K}_{RP} \in \mathbb{R}^{3 \times 3}$ relating torques to positional errors and $\mathbf{K}_R \in \mathbb{R}^{3 \times 3}$ relating torques to rotational errors. In this work we impose a certain structure on the stiffness matrix by decoupling it so that positional errors only result in force corrections, and rotational errors only result in torque corrections. This is achieved by choosing $\mathbf{K}_{PR} = \mathbf{K}_{RP} = \mathbf{0}$. Furthermore, we assign a diagonal structure to the rotational stiffness $\mathbf{K}_R = \text{diag}([k_R^x, k_R^y, k_R^z]^T)$, where the elements on the diagonal represent the rotational stiffness around the three axes of the end-effector frame of reference. The behavior of the robot under control of (2) is determined by the choice of stiffness \mathbf{K} and damping \mathbf{D} . In our case, the translational stiffness \mathbf{K}_P is allowed to vary, while \mathbf{K}_R is constant. The damping

\mathbf{D} is designed to have the same structure as the stiffness matrix:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_R \end{bmatrix} \quad (4a)$$

Consider the eigendecomposition of the symmetric and positive definite translational stiffness² $\mathbf{K}_P = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. The translational damping is constructed around the same eigenvectors, and with eigenvalues that are proportional with constant $\nu \in \mathbb{R}^+$ to the square-root of the corresponding eigenvalues of \mathbf{K}_P :

$$\mathbf{D}_P = \mathbf{Q}(\nu\mathbf{\Lambda}^{\frac{1}{2}})\mathbf{Q}^T \quad (4b)$$

The rotational damping is designed similarly as a function of the rotational damping, resulting in a diagonal rotational damping:

$$\mathbf{D}_R = \nu\mathbf{K}_R^{\frac{1}{2}} \quad (4c)$$

The constant ν can be tuned to adjust the damping of the system. In all our experiments, we used $\nu = 2$. This controller is often referred to as ‘simple impedance control’, has been used in numerous applications [27]. In our previous work [7], the robot learned to vary the translational stiffness \mathbf{K}_P as a function of time, $\mathbf{K}_P = \mathbf{K}_P(t)$. In Section 5.1 in this paper, the translational stiffness is instead learned as a function of the robot position, $\mathbf{K}_P = \mathbf{K}_P(\mathbf{x})$.

3.2 Joint Proportional-Derivative Control with Varying Gains

To control compliance at joint level, we use a proportional derivative controller with gravity compensation [28]. The command in this case is:

$$\boldsymbol{\tau} = \mathbf{K}\Delta\mathbf{q} + \mathbf{D}\Delta\dot{\mathbf{q}} + \boldsymbol{\tau}_F \quad (5)$$

where $\boldsymbol{\tau} \in \mathbb{R}^M$ represents the torques commanded to the M robot joints, $\boldsymbol{\tau}_F \in \mathbb{R}^M$ a feedforward command. The feedback error is denoted $\Delta\mathbf{q} \in \mathbb{R}^M$. The physical compliance of the robot can be varied by adjusting the feedback gain matrices³ $\mathbf{K} \in \mathbb{R}^{M \times M}$ and $\mathbf{D} \in \mathbb{R}^{M \times M}$. We constrain the gains to be diagonal matrices, which means that a positional deviation on a joint only result in corrective torques on that same joint. The stiffness k_i of each joint $i = 1 \dots M$ are learnable parameters while the damping of the corresponding joint is adjusted automatically similarly to the Cartesian controller:

$$\mathbf{D} = \nu\mathbf{K}^{\frac{1}{2}} \quad (6)$$

where ν is again a constant determining the damping, which in this work was chosen to $\nu = 2$. Varying stiffness is achieved by letting the joint stiffness \mathbf{K} vary as a

2. $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ contains the orthonormal eigenvectors of the stiffness matrix in its columns, and $\mathbf{\Lambda} \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix that contains the eigenvalues on its diagonal.

3. Note that the difference in dimensionality between CS and JS stiffness and damping matrices. In the Cartesian case, the dimensionality is always 6×6 whereas in JS the dimensionality is $M \times M$, where M is the number of joints.

1. The more of the inherent position and velocity dependent dynamics are included in this term, the closer the robot behavior will be to the control objective (1).

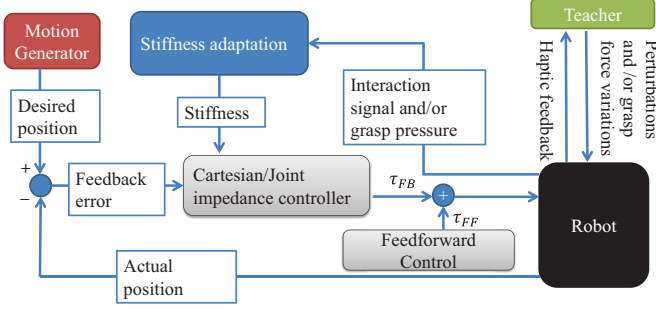


Fig. 2: Flowchart of the proposed control scheme. The robot is controlled with either Cartesian or joint impedance control. The stiffness used by the controller is adapted online according to the interaction provided by the teacher. Wiggling motions are used to decrease the stiffness and increased grasp pressure is used to increase the stiffness.

function of e.g. time, $\mathbf{K} = \mathbf{K}(t)$, or position, $\mathbf{K} = \mathbf{K}(q)$ of the robot. In our match-lighting experiment in Section 5.2, the joint stiffness is a function of time.

4 INTERFACES FOR TEACHING VARYING STIFFNESS

In Section 3, we presented two types of controllers that can be used for achieving motion with varying physical compliance. To execute a task using one of the controllers, the robot needs the desired pose along with desired stiffness parameters at each time step. This work is focused on how the robot can learn the variations of stiffness parameters from demonstration, and we will thus assume in this section that the desired position profile is already known.

In order to demonstrate the desired stiffness to the robot, the teacher must have a way of communicating to the robot what stiffness it should use. We propose two different modes of interaction that the teacher can use to alter the stiffness of the robot. The first one is based on perturbations imposed by shaking or tapping on the robot. If the teacher wiggles or shakes the robot around its current reference point, the robot will interpret this as a signal to reduce the stiffness. The second mode of interaction measures the force that the teacher grasps the robot with, and interprets high measured force as a signal to increase the stiffness. Both of these modes of interaction are inspired by how we believe that humans indicate decrease and increase of stiffness between one another. For example, in dance and other sports, when the teacher wishes to convey to the student that she should relax, she may say ‘relax’ and at the same time wiggle the limb that is too stiff. Similarly, to insist that a certain body posture is important, a teacher may grasp the students arm with a firm grip while guiding it to the correct posture. Fig. 2 shows the flowchart of the robot control while the teacher is providing input through these interfaces.

4.1 Stiffness Decrease Based on Imposed Perturbations

As outlined above, this interface is designed such that the robot will decrease its stiffness based on the perturbations imposed by the teacher. To this end, it is necessary to measure the perturbations, which can be done in many different ways depending on the available sensors. The most straightforward way to measure the interaction is to look at the position feedback error. If the teacher is wiggling the robot around its desired reference point, this will result in errors which are observable in the feedback signal. Alternatively, for robots equipped with joint torque sensors, e.g. the KUKA LWR, the sensed interaction torques can be used as interaction signal. This is preferable to using the positional feedback signal as it allows interaction to be detected even if it does not result in movement of the robot away from its equilibrium position. This is important for applications in which the maximum stiffness is so high that humans will find it difficult to move the robot off its path.

In our previous work [7], we proposed to use the perturbations in a time-window⁴ $[t-S, t]$ for determining the stiffness at time t . In this paper, we incorporate a preparatory band-pass filtering step before proceeding to computing the stiffness using the filtered signal. The role of this filter is two-fold. First, it serves to remove high-frequency content from the interaction. This is desirable, as the raw signal is typically corrupted by high frequency sensory noise, and because interaction other than teaching (e.g. contact with environment) may contribute to the interaction signal. While separating the interaction signal in the frequency domain does not guarantee that such effects are avoided, it does make them less probable. Secondly, the lower frequency bound on the filter gives the teacher the possibility to use slow (low frequency) perturbations to feel how stiff the robot is without actively changing the stiffness. Choosing the lower cutoff frequency to $\underline{f} \approx 0.5$ Hz allows the teacher to comfortably perform slow perturbations to feel the stiffness of the robot. The upper cutoff frequency \bar{f} should be set to a value slightly above the frequency that we can expect that a human teacher can be responsible for. There have been physiological studies on this topic, with reports of peak frequency for voluntary manipulation involving hand and arm movements in the range 4-8 Hz [29] and 2-5 Hz [30] [31]. To have some margin, the upper cut-off frequency was set to $\bar{f} = 10$ Hz in this work.

Let $\xi_t \in \mathbb{R}^N$ denote the filtered interaction signal at time t . The dimensionality N of the ξ_t is 3 in the Cartesian case and equal to the number of joints in the joint space case. To determine the stiffness based on the teacher’s interaction, we take into account the interaction signal received in the time frame $[t-S, t]$. Below, we assume that there are M perturbation data points in this

⁴ S is a parameter which acts as a trade-off between smoothness of the stiffness profile and delay in the adaptation.

time frame, and we index these points as $\{\xi_i\}_{i=1}^M$. The first step is computing the empirical covariance matrix of the perturbations:

$$\Sigma_t = \frac{1}{M} \sum_{t-S}^t (\xi_t - \mu_t)(\xi_t - \mu_t)^T \quad (7a)$$

with

$$\mu_t = \frac{1}{M} \sum_{t-S}^t \xi_t \quad (7b)$$

Based on this covariance matrix, we wish to determine the direction(s) of interactions and a measure of the intensity of the interaction in these direction. The way this is done depends on whether CS or JS stiffness is to be learned.

In the JS case with the controller from Section 3.2, the stiffness is determined independently at joint level, c.f. Eq. (5). The parameters to be determined are diagonal entries of the joint stiffness matrix. To measure the intensity of the perturbations on the joints, we consider the diagonal of the interaction covariance matrix Σ_t . Let σ_t^i denote the i^{th} diagonal element of Σ_t . We map this value to the corresponding joint stiffness k_t^i as:

$$k_t^i = \begin{cases} \bar{k} & \lambda < \lambda \\ \bar{k} - \frac{\bar{k}-k}{\lambda-\lambda}(\sigma_t^i - \lambda) & \lambda < \sigma_t^i < \bar{\lambda} \\ k & \lambda > \bar{\lambda} \end{cases} \quad (8)$$

The parameters \bar{k} and k are the maximum and minimum allowed stiffness values. The maximum stiffness can be determined e.g. to prohibit too high stiffness values in order to ensure safety of interaction, or it can be set to the maximum stiffness allowed by the hardware. The minimum stiffness can be set to some low value that ensures that the robot is still capable of unconstrained motion if the stiffness is reduced maximally. The sensitivity of the stiffness as a function of the perturbations is controlled by the parameters λ and $\bar{\lambda}$, which determine the amplitudes required to start reducing the stiffness and to achieve minimum stiffness respectively.

In the Cartesian case, we wish to infer not only the magnitude of stiffness but also its principal directions, since the translational stiffness matrix is allowed to have off-diagonal elements, refer to Section 3.1. The teacher should be able to make the robot compliant in a particular direction by perturbing the robot selectively in that direction. To achieve this, we consider the eigendecomposition of the covariance matrix Σ_t :

$$\Sigma_t = Q_t \Lambda_t Q_t^T \quad (9)$$

where $\Lambda_t \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix containing the eigenvalues of Σ_t , and $Q_t \in \mathbb{R}^{3 \times 3}$ contains the corresponding orthonormal eigenvectors in its columns. The eigenvectors represent the three principal directions of the spatial perturbations imposed by the teacher. The stiffness matrix is constructed around these same eigenvectors, but with eigenvalues that are negatively

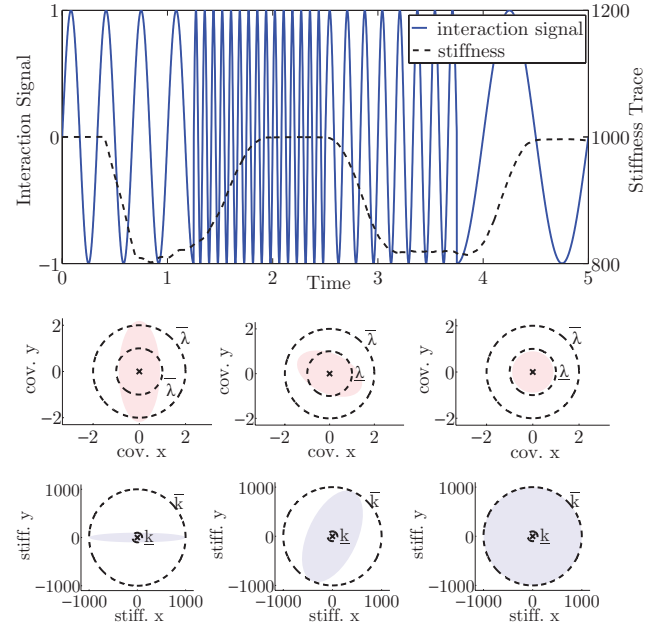


Fig. 3: Top: An example of interaction signal and resulting stiffness trajectory. This is a sinusoidal interaction with a frequency that starts at 3 Hz, climbs to 12 Hz, drops to 6Hz and finally drops to 1Hz. Note how the interaction signal does not affect the stiffness for the frequencies outside the band-pass, chosen to [2Hz-10Hz] in this example. Middle: 2d examples of covariance of the interaction, Bottom: Stiffness ellipsoids resulting from the interaction examples.

proportional to the eigenvalues of the interaction signal covariance matrix:

$$K_t = Q_t \Gamma_t Q_t^T \quad (10)$$

with Γ is a diagonal with entries $k_t^i, i = 1 \dots 3$ determined by a truncated linear map as in Eq. (8), replacing the diagonal values of Σ_t with the eigenvalues $\lambda_i, i = 1, 2, 3$ of Σ_t . Fig. 3 provides a set of examples of 2d interaction covariance matrices and resulting stiffness.

4.2 Stiffness Increase Based on Grasp Pressure

The perturbation-based interface described in Section 4.1 can be used to *decrease* the stiffness of the robot. When using that interface, the default⁵ stiffness of the robot is set to the maximum allowed stiffness, and decreased locally based on the perturbations, refer to Section 4.1. In this section, we present an extension that allows increase of stiffness. The teacher indicates increase in stiffness by increasing the grasp pressure with which he holds the robot. This way, the default stiffness can be chosen to a moderately compliant level, and increased or decreased locally as required.

Depending on the sensing available for detecting the grasp pressure, one can envision different ways to map the pressure to a selective increase in stiffness. For example, on a manipulator where several links are covered with artificial skin, the detected pressure on a link could

5. By default stiffness, we mean the stiffness that the robot would use in absence of any indication from the teacher.

be mapped to a stiffness increase selectively to its parent joint. Below, we describe the grasp pressure used in our CS implementation. For CS, we propose to map the perceived pressure from any part of the body to a uniform increase of all the eigenvalues of the stiffness matrix. This allows to use the two interactive modes in conjunction to vary the stiffness at the end-effector, using perturbations to shape and decrease the stiffness, while using pressure to uniformly increase stiffness. To avoid interpreting skin pressure signals arising from collisions or other disturbances, approximately equal pressure on antagonistic parts of the arm is required for the stiffness to be changed. Let Ψ_t^a and Ψ_t^b denote the vectors of pressure detected on the agonist and antagonist parts of one part of the arm. Furthermore, let ψ_t denote a scalar which represents the grasp pressure of the arm at time t , defined by:

$$\psi_t = h(\Psi_t^a, \Psi_t^b) \frac{\max\{\Psi_t^a\} + \max\{\Psi_t^b\}}{2} \quad (11a)$$

$$h(\Psi_t^a, \Psi_t^b) = \begin{cases} 1 & \text{if } |\max\{\Psi_t^a\} - \max\{\Psi_t^b\}| < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (11b)$$

The stiffness matrix is then shaped according to Eq. (10), with eigenvalues:

$$k_t^i = k_d + \frac{\bar{k} - k_d}{\bar{\psi} - \underline{\psi}} \max(\psi_t - \underline{\psi}, 0) - \frac{k_d - \underline{k}}{\bar{\lambda} - \underline{\lambda}} \max(\lambda_t^i - \underline{\lambda}, 0) \quad (12)$$

The first term is the default stiffness, which can be set to some value in the lower allowed stiffness range. The second term is the linear increase of stiffness from the default value k_d to the maximum allowed value \bar{k} when the pressure changes from its lower threshold $\underline{\psi}$ to its upper threshold $\bar{\psi}$. The lower threshold is important since without it, a normal grasp force used e.g. for wiggling the robot could unintentionally make the robot stiffen up. The upper threshold should be a value which is close to the maximum pressure a human teacher can be expected to exert on the robot. Both these thresholds are easily tunable parameters. The third term is the decrease from k_d to \underline{k} as eigenvalue λ^i goes from its minimum threshold $\underline{\lambda}$ to its upper threshold $\bar{\lambda}$.

5 TASK LEARNING

The interfaces presented in Section 4 allow the teacher to modify the stiffness of the robot online. To learn from this interaction for subsequent autonomous task executions, a learning algorithm must be used. As for learning positional profiles, there are a wide range of algorithms in machine learning that can be used to learn a model based on demonstrations. In our previous work [7], we presented the Online Learning of Varying Stiffness (OLVS) algorithm for incrementally learning time-dependent stiffness variations using the perturbation-based interface for decreasing the stiffness. That approach did not learn the stiffness variations

directly, but instead saved all the perturbations imposed by the teacher during the demonstration phase. These perturbations were then replayed during subsequent task executions, to achieve incremental refinement of the stiffness profile. In this work, we revisit the task of pouring a drink (initially used as an example task in [7]), but instead learn the stiffness variations as a function of the position of the robot.

5.1 Task: Pouring a drink

This learning scenario illustrates how the interfaces described in Section 4 can be used in real world task. The task consists in first transporting a bottle full of soda toward a glass, and pour the soda into the glass when it has been reached. In this task, it is preferable to be very compliant in the reaching phase, where position is not important and stiff behavior may cause aggressive response to perturbations which result in spilling. However, when pouring, it is preferable to be stiff, since the robot should attempt to reject any external perturbations which may move the bottle from the glass. The robot used for this task is the 7 DoF Barrett WAM. The forearm of the robot is covered with artificial skin of the type presented in [32], see Figures 1 and 6.

Our previous treatment of this task in [7] used fixed reference trajectory for the position and orientation, and OLVS for learning the stiffness variations. The reference trajectories as well as the learned stiffness profile were thus time dependent, which meant that perturbations that significantly delayed the task would result in failure such as the drink being poured before the glass was reached. Below, we revisit the pouring task and address these shortcomings by using autonomous dynamical systems for modeling the kinematics of the task. Furthermore, the stiffness is learned as a function of the robot position, which ensures that the stiffness variations happen when the task requires it rather than at a specific time.

5.1.1 Position profile

To learn a generalized model of the kinematics of the task, we model the position profile as an autonomous Dynamical System (DS). The DS is modeled by a Gaussian Mixture Model, trained with the Stable Estimator of Dynamical Systems (SEDS) [33] algorithm to guarantee that the motion converges to the glass regardless of starting position. This approach uses a set of demonstrations to optimize the parameters of a GMM encoding an estimate of the joint probability of the position and the velocity of the robot. The velocity then becomes a function of position through regression on the GMM, see [33] for details. The representation of the desired position profile as a DS is illustrated in Fig. 4.

At the onset of motion, the position of the end-effector, $\mathbf{x}_0^r \in \mathbb{R}^3$ with respect to the target position $\mathbf{x}_0^* \in \mathbb{R}^3$ is computed:

$$\mathbf{x}_0 = \mathbf{x}_0^r - \mathbf{x}_0^* \quad (13a)$$

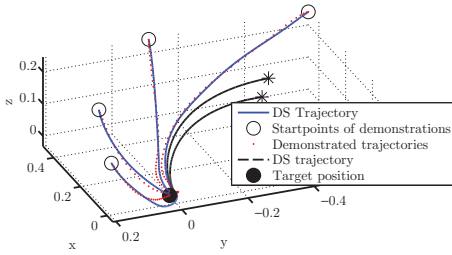


Fig. 4: The figure shows the demonstrations provided by the teacher (red dotted line) and the DS trajectories generated from the same starting positions (blue). Two trajectories from other starting positions (black) illustrate that similar motion is reproduced, with convergence to the target.

The desired velocity w.r.t the target is then computed using the DS, which maps the current relative position to the desired relative velocity through the non-linear function $f: \mathbb{R}^3 \mapsto \mathbb{R}^3$:

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t) \quad (13b)$$

To find a set point \mathbf{x}_{t+1}^d for the controller, we integrate the previous relative position and change coordinates to the robot reference frame via addition of the current target position:

$$\mathbf{x}_{t+1}^d = \mathbf{x}_t + \Delta t \dot{\mathbf{x}}_t + \mathbf{x}_t^* \quad (13c)$$

Above, Δt denotes the length of one iteration of the control loop ($\delta t = 0.002$ in our implementation). The target position is updated from its actual position⁶, while the robot's position is integrated open loop:

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^d - \mathbf{x}_{t+1}^* \quad (13d)$$

5.1.2 Orientation Profile

When pouring a drink, the bottle should ideally remain vertically oriented everywhere except when it is correctly positioned above the glass. Having a time dependent orientation profile will cause the robot to pour the drink outside the glass if the robot is unable to reach the glass in the planned time. The control of the orientation should therefore depend on the state of the reaching motion, and exhibit coupling so that if the end-effector is moved away from the glass, the orientation of the bottle immediately gets corrected. In our previous work [34], such coupling between reach- and grasp motions was observed in human subjects. In that paper, we also developed the Coupled Dynamical Systems (CDS) approach to mimic this capacity in robots. In this work, we use a simplified version of CDS, which couples the orientation of the end-effector to the state of the reaching motion by using distance between the end-effector and the glass as coupling variable.

We adopt the unit quaternion representation for the orientation of the end-effector. Let \mathbf{r}^a and \mathbf{r}^b denote the quaternions corresponding to the default and pouring

orientation respectively. These orientations were captured by moving the robot to the desired orientations and recording their quaternions. To compute the desired orientation at time t , \mathbf{r}_t^d , we interpolate between \mathbf{r}^a and \mathbf{r}^b using spherical linear interpolation:

$$\mathbf{r}_t^d = \frac{\sin(\Omega(1-w))\mathbf{r}^a + \sin(\Omega w)\mathbf{r}^b}{\sin(\Omega)} \quad (14a)$$

where Ω satisfies $\cos(\Omega) = \mathbf{r}^a \cdot \mathbf{r}^b$, and $w \in [0, 1]$ is the interpolation parameter, which we define as function of the difference of the current position of the end-effector \mathbf{x}_t^r and the position of the glass \mathbf{x}_t^* :

$$w(\mathbf{x}_t^r, \mathbf{x}_t^*) = \exp(-\kappa(\mathbf{x}_t^r - \mathbf{x}_t^*)^T(\mathbf{x}_t^r - \mathbf{x}_t^*)) \quad (14b)$$

Here, κ is a parameter that controls how quickly the rotation is changed as a function of movements to and from the glass. This parameter was set experimentally to $\kappa = 0.06$.

5.1.3 Stiffness Profile

The desired stiffness profile in the pouring task is to be compliant during transport and stiff when pouring. As in the choice of end-effector orientation, the stiffness should thus depend on the state of the reaching motion rather than time. This is a common situation for manipulation tasks, i.e. that there is some kind of free motion where it is generally preferable to use a low stiffness, and a manipulation step that may involve contact that requires a specific stiffness to be used locally. Therefore, it is natural to learn the stiffness variations as a function of the state of the robot. Furthermore, the teaching process can be simplified by first demonstrating the general stiffness that the robot should use as its default, and in a secondary step demonstrating the local changes from this default stiffness as the robot is performing the task.

We used the combined perturbation and pressure based interface described by Eq. (12). First, the default stiffness k_d was set to the value in the middle of the allowed stiffness range $[\underline{k}, \bar{k}]$. In our implementation, this corresponds to $\bar{k} = 1000$ N/m, $\underline{k} = 70$ N/m and $k_d = 535$ N/m. For demonstrating the default stiffness, the robot was set to hold a constant position while the teacher was interacting with it to find a satisfactory default behavior. Since a uniformly low default stiffness was desirable, the teacher provided spherical perturbations with increasing amplitude until a sufficiently low stiffness was achieved. Once the interaction stopped, the resulting stiffness matrix was saved as default stiffness. The trace of the end-effector and the final stiffness at the end of the first teaching step are shown in Fig. 5.

The robot then executed the task with the combined perturbation and pressure-based interface (Section 4.2), but now with $k_{def}^i, i = 1, 2, 3$ according to the default stiffness learned in the preceding step. The orientation of the default stiffness matrix is described by its eigenvectors \mathbf{Q}_{def} . To preserve this orientation, the computation of the perturbation covariance matrix, Eq. (7a),

6. The position of the glass is tracked at 200 Hz by a marker-based vision system.

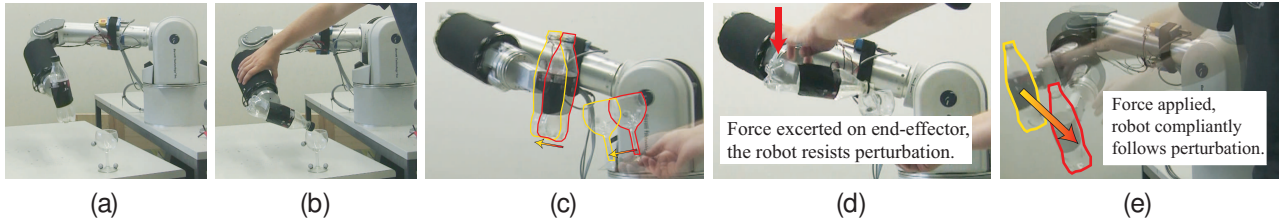


Fig. 6: **a,b)** Teaching the stiffness variations for the pouring task. Here, the robot uses the default, low stiffness learned in the preceding step, and stiffens up locally when the teacher applies pressure on the skin. **c)** Illustration of online motion adaptation as the glass is moved. **d)** The robot resists the perturbation when pouring and stiffly remains in position. **e)** The robot is compliant when it is far from the glass.

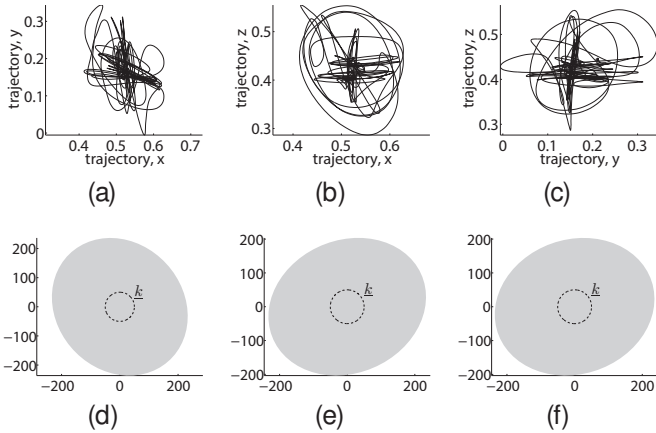


Fig. 5: Top: the end-effector trajectory (left: XY, middle: XZ, right YZ) during interaction for determining the default stiffness. Bottom: resulting stiffness ellipses (left: XY, middle: XZ, right YZ)

is preceded by rotation of the perturbation data points onto the basis given by the columns of Q_{def} . During task execution, the teacher grasped the forearm of the robot and applied pressure as the robot approached the glass, see Fig. 6. Three such demonstrations were provided, and the stiffness trace was recorded when interaction was detected together with the relative position between the end-effector and the glass $x = x^r - x^*$.

Based on the recorded data set, we seek a functional relationship that allows the robot to choose its stiffness based on the current relative position to the glass. To minimize the amount of parameters used for the modeling of this relationship while ensuring that the resulting stiffness is always symmetric and positive definite, we represent the stiffness matrix by its Cholesky vector $l = [L_1, \dots, L_6]^T$, defined by⁷:

$$K = L^T L, \quad L = \begin{bmatrix} L_1 & L_2 & L_3 \\ 0 & L_4 & L_5 \\ 0 & 0 & L_6 \end{bmatrix} \quad (15)$$

To represent the demonstrated data in a compact model, we chose to use a GMM. Since the complexity of the data is expected to be low, we set the number of Gaussians to two only. The GMM was then trained using the standard Expectation Maximization (EM) algorithm [35]. The resulting GMM is a joint density estimate of

the vector $\xi \in \mathbb{R}^9$ concatenating the relative position between the end-effector and the glass and the Cholesky vector:

$$p(\xi) = \sum_{k=1}^K \pi^k \mathcal{N}(\xi; m^k, C^k) \quad (16)$$

During task execution the robot should use this model to choose its stiffness Cholesky vector as a function of the relative position to the glass. This can be done by GMR, which computes the conditional expectation $E\{l|x\}$ based on the GMM. One of the properties of standard GMR is generalization, which means that relationships found locally in the training data can be used when inferring distant points in the input space. For many applications, this is a desirable effect since it can reduce the quantity of training data. However, in some situations it is inappropriate to generalize in this manner. The case of inferring stiffness is an example of such a situation. Generalizing a locally linear relationship to regions outside the training range would mean that the inferred stiffness could grow unbounded. To prevent this, we propose a novel regression strategy in which the generalization can be controlled with a single parameter. This regression method is described in detail in the appendix. In addition to the GMM, this method makes use of a basic conditional distribution $p_b(l|x)$ that describes the desired relationship between inputs and outputs in regions poorly covered by the training data. As explained in the appendix, for regression it is sufficient to define the mean of this basic distribution. In this particular application, the mean of the basic distribution is the Cholesky vector of the default stiffness from the first demonstration phase.

As the robot performs the task autonomously, the Cholesky vector $l = l(x)$ is found with the regression method described in the appendix. The output from the regression is then used to reconstruct the Cholesky matrix which is finally used to compute the stiffness according to Eq. (15). Fig. 7 shows how the maximum eigenvalue of the stiffness matrix varies as a function of the distance to the glass in the xy-plane. Clearly, the desired behavior of the stiffness increasing only locally when the robot is close to the glass is fulfilled. We plotted only one of the eigenvalues, since the pattern is similar

7. Note that K here represents the translational part of the stiffness matrix.

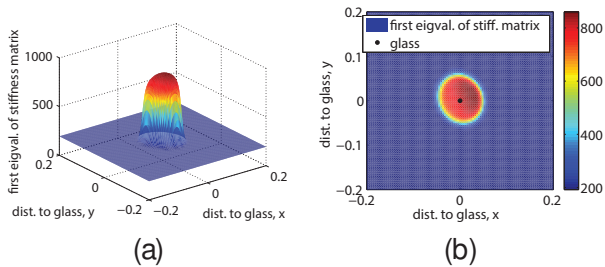


Fig. 7: Left: The plot shows a 3d plot of the maximum eigenvalue of the stiffness matrix as a function of the distance to the glass in the xy-plane. Right: Same data as in the left plot, with a color-map for visualizing variations of the maximum eigenvalue of the stiffness matrix.

for all three⁸.

5.2 Task: Lighting a match

Lighting matches is a typical example of a task that humans perform with ease, whereas a position controlled robot will either break the match or apply too little force if the matchbox is not exactly in the programmed position. One solution would be to control the contact force between the match and the matchbox directly. This would, however, require very accurate force sensing at the wrist and a more complicated control system, with more parameters to be tuned. We hypothesized that decreasing the stiffness as the robot strikes the match is sufficient to limit the contact force in this task.

A JS control architecture was chosen for this task. The primary reason for this choice was to exemplify how our system can be used in JS⁹. One demonstration of a joint space trajectory was given using kinesthetic teaching. To get a smoothed version of the demonstrated trajectory that is able to reproduce the trajectory with high precision, we employed a Gaussian Process (GP) to encode the demonstrated motion. For reproducing the motion, the reference position \hat{q}_t^i of each joint i at time t is retrieved through Gaussian Process Regression (GPR), [36]:

$$\hat{q}_t^i = \mathbf{c}_t^T (\mathbf{C} + \sigma_q \mathbf{I})^{-1} \mathbf{q}^i \quad (17a)$$

where $\mathbf{c}_t^T = [c(t, t_1), \dots, c(t, t_N)]^T$ is a vector containing pairwise evaluation of the covariance between t and every training point, \mathbf{C} is a matrix containing the pairwise covariance between all the points in the training set, $[\mathbf{C}]_{ij} = c(t_i, t_j)$. We define the covariance function as the standard squared exponential:

$$c(t, t') = \sigma_n \exp\left(-\frac{(t - t')^2}{l^2}\right) \quad (17b)$$

In order to showcase the need for varying stiffness, we first carried out a series of trials with fixed stiffness values. In the first series of trials, the joint stiffness was set to a high, constant value of 1000 Nm/rad for

8. In this task, a uniform increase was desired, and the pressure-based interface increases the stiffness uniformly.

9. This task could also have been encoded in CS.

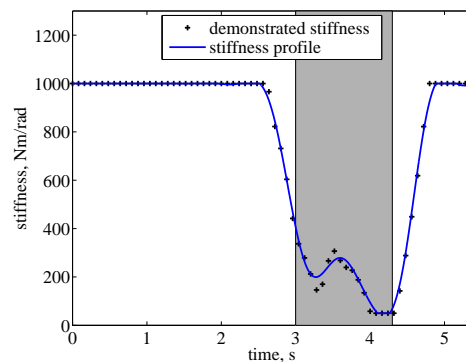


Fig. 8: The stiffness profile for the elbow joint in the match-lighting task. The shaded area indicates the striking phase. The remaining 6 joints of the robot had a constant stiffness of 1000 Nm/rad.

all joints. This stiffness is similar to position control, as the robot is very stiff and effectively rejects most external forces. 20 trials were carried out, and most common outcome of these attempts was that contact was established but the match broke. In a second set of trials, the stiffness of the elbow joint was set to a constant value of 50 Nm/rad. Motion around the elbow corresponds in this case to approximately perpendicular motion between the match and the matchbox, hence reducing this stiffness reduces the contact force between the match and the matchbox¹⁰. The lower the stiffness in this joint is set, the lower the contact force, and 50 Nm/rad is the minimum value that was considered acceptable from the perspective of maintaining the possibility to track the position trajectory. Again, 20 trials were carried out, with the most common outcome being that the match was struck through thin air, due to inadequate precision in positioning the match before the striking motion.

To achieve high precision in positioning while ensuring compliant motion when striking, the stiffness should be decreased only locally in the striking phase. To achieve this, the perturbation based interface for JS (Section 4.1) was used. As the robot executed the motion, the teacher intervened in the striking phase by wiggling the elbow joint around its reference trajectory. The resulting stiffness was recorded with corresponding time stamps. For smooth reproduction of the stiffness profile, we used a GP similarly to the joint position trajectory. The resulting stiffness profile can be seen in Fig. 8.

The result was accurate positioning before establishing contact with the matchbox, and a compliant motion through the striking phase. While this controller still occasionally breaks the matches, the success rate is significantly higher (refer to Table 1) than for the two settings with fixed stiffness values. Large variability in the quality of the matches as well as small positioning errors of the matchbox are believed to be the most

10. The contact force arises from the controller trying to reject the perturbation of the planned trajectory of the match as it comes in contact with the matchbox. The lower the stiffness, the lower the controller effort and hence the contact force.

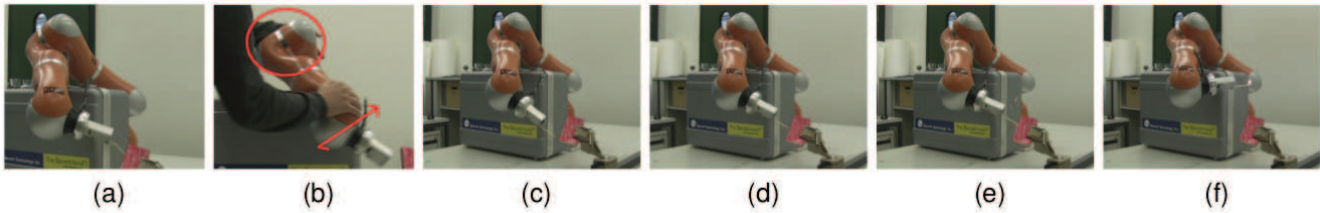


Fig. 9: a): An attempt at trying to light the match using the position controller, resulting in the match breaking. b): The demonstration of stiffness variations. The teacher indicates decrease of stiffness of the elbow joint (encircled) during the striking motion. The red arrow indicates the direction of wiggling. Note that the demonstration was carried out without a match mounted on the robot. c,d,e,f): Snapshots from task execution where the robot successfully lights the match using the learned stiffness profile.

TABLE 1: Rates of success the match lighting task with constant high stiffness, constant low stiffness in the elbow joint and learned varying stiffness in the elbow joint. A total of 20 trials was carried out for each case.

	Broke	Broke and was lit	Not lit	Lit	Rate of success
const. high stiff.	4	11	2	3	15%
const. low stiff.	1	3	14	3	15%
learned var. stiff.	0	2	1	17	85%

important reasons of failure.

6 USER STUDY

In the previous Section, we described and exemplified how our interfaces can be used for learning the stiffness for different tasks. A crucial aspect of any LfD method is how convenient it is for non-roboticists to use it. Indeed, the whole motivation of LfD is to allow people without technical knowledge in robotics to transfer skills to the robot. In this section, we address this by evaluating our approach through a user study. To provide a simple alternative to our system, we developed a Graphical User Interface (GUI) with which the user can change the stiffness along the three Cartesian axes using sliders.

6.1 Subjects

Two groups of 14 participants took part in the study. The participants in one group were asked to select the stiffness using the GUI. The participants in the second group used our proposed approach, combining wiggling motions for selectively decreasing the stiffness with grasp force for uniformly increasing it, refer to Section 4. In the following, the two groups will be referred to as the GUI group and the PHRI (Physical Human-Robot Interface) group. The subjects were mainly recruited on campus at EPFL. They were naive to robotics with an age-range of 19-35. 10 out of the 28 participants were female, with equal gender distribution in each group.

6.2 Task

The task considered was chosen specifically such that there are different stiffness requirements along different

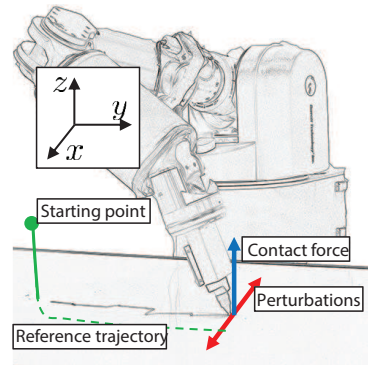


Fig. 10: The figure illustrates the drawing task. Force impulses along the x-axis cause the robot to depart from the desired, straight trajectory. The dashed line represents the part of the reference trajectory that lies below the contact surface, giving rise to a contact force along the z-axis. The teachers role is to change the stiffness of the robot, such that the perturbations are rejected while the contact force remains small.

directions in task space. The task consisted in drawing a straight line on a white-board. While the robot was drawing, a series of impulses were added to the control output, acting equivalently to external perturbations. When such an impulse was applied, the robot departed from its straight line trajectory. Furthermore, the reference trajectory was deliberately placed behind the contact surface, such that a contact force was established between the pen and the board, see Fig. 10. For safety reasons, the robot was programmed to abort the task whenever the contact force exceeded 15N. For maximum performance in this task, the stiffness along the x-axis (see Fig. 10) should be high, so that the robot can counteract the force impulses. To ensure a low contact force, the stiffness along the z-axis should be low. The stiffness along the y-axis has no influence on the task performance and could hence take arbitrary values. The stiffness was set to be constant throughout the task.

6.3 Experiment Protocol

Each subject was given a description of the task, and the instruction to choose the stiffness of the robot so as to minimize the contact force and the departure from the straight line. The instructions did not reveal the correct task stiffness (described in Section 6.2). Subjects were also given a description of the interface (GUI or PHRI

depending on the group). After reading the description, the subjects were allowed to familiarize themselves with the interface during a practice period of two minutes. During this time, the robot maintained a static reference position while the participants were allowed to use the interface and interact with the robot to understand how their input affected the stiffness of the robot. The subjects were also allowed to ask the experiment supervisor interface-related questions during this time.

After the two minutes of practice, the subjects were asked to improve the performance of the drawing task by changing the stiffness of the robot. During the teaching process, they could at any time ask the experiment supervisor to start a new trial of the task. Subjects were allowed to change the stiffness while the robot was stationary between trials, or during the trials. They were also allowed to temporarily remove the pen if they wished to do so. Questions relating to the task (e.g. what effect could be expected by increasing or decreasing the stiffness) were not answered, while questions regarding safety (e.g. where to hold the robot) were.

A successful trajectory was defined as follows:

- 1) The contact force never exceeded 15N, i.e. the robot did not abort the drawing before reaching the end of the line.
- 2) The standard deviation of the trajectory along the x-axis was below a predefined threshold value.

The subjects were informed about their progress only once they had managed to tune the stiffness such that a successful trajectory had been completed.

After completing the teaching, the subjects were asked to fill out a digital questionnaire containing the NASA Task Load Index (TLX) [37] and the System Usability Scale (SUS) [38] forms. The NASA Task Load Index is a commonly used evaluation method for the workload of a task. The evaluation consists in a series of workloads (e.g. physical, mental and temporal demand) which are rated using a 20-point Likert-scale. The System Usability Index is based on 5-point Likert-scale agreement response to statements about the user-friendliness of a system, e.g. 'I thought that the system was easy to use'.

6.4 Results

Our main hypothesis was that teaching stiffness variations using the PHRI would be more efficient than using the GUI, which does not provide the immediate haptic feedback that the PHRI does. We could observe that subjects in the GUI group needed haptic feedback to understand the effect of a change in stiffness value. Indeed, the vast majority of the subjects in the GUI group conducted their teaching by changing a value on the GUI and subsequently pushing or pulling the end-effector in different directions to get an idea of the effect that the change in stiffness brought.

As quantitative measures of teaching efficiency, we used the total amount of trials and the total teaching

TABLE 2: Objective and Subjective results from the user study. The table entries are formatted as mean (standard deviation) for each of the performance measures.

	Teaching time	Number of trials	SUS	TLX
PHRI	239.8 (100.1)	5.4 (2.2)	68.2 (16.7)	34.4 (16.9)
GUI	467.9 (260.7)	11.0 (6.9)	69.4 (15.6)	43.92 (17.3)

time¹¹ before the subjects had completed a successful trajectory. These measures are presented in Table 2 along with the subjective results from the TLX and SUS questionnaires. The mean and standard deviation for each measure is also plotted in Fig. 11. The PHRI is clearly advantageous in terms of the time it took the participants to teach the task (Oneway ANOVA: $p=0.006$; Welsh: $p=0.004$; Mann-Whitney: $p=0.02$). It also required fewer trials (Oneway ANOVA: $p=0.01$; Welsh: $p=0.007$; Mann-Whitney: $p=0.03$). Regarding usability, the GUI and the PHRI are on par with no statistically significant difference between the SUS-scores of the two groups (Oneway ANOVA: $p=0.84$, Welsh: $p=0.42$, Mann-Whitney: 0.47). While the PHRI achieved a lower value of TLX than the GUI (the lower the TLX, the better; range 0-100), this difference is not enough to give a statistically significant advantage to the PHRI (Oneway ANOVA: $p=0.16$; Welsh: $p=0.08$; Mann-Whitney: $p=0.08$). A pairwise comparison for each of the factors underlying the SUS and TLX scores was also carried out, but no statistically significant difference was found.

From the results of the user study, we draw three conclusions:

- 1) All subjects, while being naive to robotics, were able to very rapidly figure out how to use each interface and to determine the appropriate stiffness for the task.
- 2) Both interfaces were perceived as user-friendly (An interface with a SUS-score above 65 is generally considered user-friendly).
- 3) The PHRI was the most efficient of the two interfaces, as it allowed the users to teach the task faster and with fewer trials.

7 DISCUSSION

We have addressed the problem of teaching tasks that require or can benefit from task-based varying stiffness. We proposed two modes of interaction and presented interfaces that adapt the robot stiffness based on these interactions. To make the robot more compliant, the teacher wiggles the robot around its planned trajectory as it is performing the task. To make the robot more stiff, the teacher increases the pressure with which he holds the robot. The robot responds online by adapting the stiffness based on this interaction, providing the teacher with instant haptic feedback on the effect of the teaching. We have exemplified and demonstrated the usefulness

¹¹ The total teaching time included the trials and the time for adjusting the stiffness while stationary in between trials.

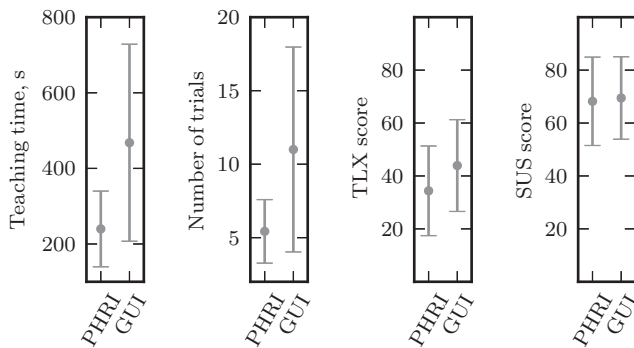


Fig. 11: The figures show the mean and standard deviation of the performance measures for the two groups.

of our approach in both CS and JS, and validated the efficiency of the proposed interface through a user study.

In the second task (Section 5.2), we exemplified the system for teaching joint-space stiffness variations in the challenging task of lighting a match with the KUKA LWR. We showed that the rate of success for lighting matches was significantly increased by varying the stiffness compared to both constant high and constant low stiffness. None of the task coordinates CS and JS are universally better than the other. Which one to use depends on the task and the robot at hand.

As opposed to other approaches to tuning the compliance parameters in manipulation, our approach does not use a model of the environment, nor does it require to optimize over a cost-function describing the task. Instead, it assumes that the teacher is capable of demonstrating appropriate stiffness variations to the robot. For slow manipulation tasks, it is easy for a human to determine when the robot is too stiff or too compliant. This was shown by the results of our user study, in which all subjects were able to figure out how the stiffness should be adapted to increase the task performance, regardless of the interface (GUI or PHRI) that was used for teaching. For tasks that involve high speed motion, it can be difficult for the teacher to have the time to achieve the desired stiffness variations along the trajectory. A straightforward remedy to this is to slow down or pause the motion during the teaching process. The use of incremental learning can also make it easier for the user to gradually transfer the appropriate stiffness profile. However, our interface (or any other instruction-based interface) is likely not applicable to highly dynamic tasks where varying stiffness is required to achieve explosive motions¹², such as kicking a ball. For such tasks, the optimal stiffness control involves fast switching between saturation levels [39], which is problematic to demonstrate with the proposed interfaces as the possible rate of change is limited. Furthermore, it is unlikely that even an expert would be able to determine near-optimal stiffness variations for such tasks.

12. This type of motions can be achieved with passively compliant joints, which have the possibility to store energy.

Varying stiffness is often important in tasks involving tools in physical contact. For such tasks, the wiggling motions to decrease the stiffness may not be possible. One solution to this difficulty is to let the robot execute the task without contact during the teaching phase. This can be achieved in several ways, e.g. by adding an offset to the reference trajectory or simply by removing the tool during teaching (as we did in the match-lighting task in Section 5.2).

An alternative approach to learn and control contact tasks such as lighting matches is to close the loop on the contact force, using e.g. hybrid [40] or parallel force control [13]. This requires a force sensor, and the control system becomes more complex with additional parameters which directly influence the task performance and hence should be chosen carefully. Therefore, it is in our view generally preferable to avoid feedback force control when not absolutely necessary. For tasks in which it is important to track a reference contact force, force feedback has to be used for good performance. In future work, we plan on extending the proposed framework by including position, stiffness and desired contact forces as learnable and controllable parameters. Future work will also extend the Cartesian teaching interface to include rotational degrees of freedom.

All the experiments in this paper were carried out on torque-controlled robots. Our approach can also be implemented on a position-controlled robot, provided that it is equipped with a force sensor, allowing compliant motion by generating the position reference online based on the sensed force [4].

ACKNOWLEDGMENT

This research was supported by the Swiss National Science Foundation through the National Center of Competence in Research Robotics.

REFERENCES

- [1] N. Hogan, "Impedance control: An approach to manipulation," *Journal of Dynamic Systems Measurement and Control*, vol. 107/17, no. March, pp. 304–313, 1985.
- [2] R. Ham, T. Sugar, B. Vanderborght, K. Hollander, and D. Lefeber, "Compliant actuator designs," *IEEE Robotics & Automation Magazine*, vol. 16(3), no. September, pp. 81–94, 2009.
- [3] M. Garabini, A. Passaglia, F. Belo, P. Salaris, and A. Bicchi, "Optimality Principles in Variable Stiffness Control : The VSA Hammer," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [4] L. Villani and J. De Schutter, *Force Control, Chapter in handbook of robotics*, B. Siciliano and O. Khatib, Eds. Springer, Jul. 2008.
- [5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. Springer, 2011.
- [6] S. Chan and H. Liaw, "Generalized impedance control of robot for assembly tasks requiring compliant manipulation," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 4, pp. 453–461, 1996.
- [7] K. Kronander and A. Billard, "Online Learning of Varying Stiffness Through Physical Human-Robot Interaction," in *IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2012.
- [8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A Survey of Robot Learning from Demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009.

- [9] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Handbook of Robotics Chapter 59: Robot Programming by Demonstration," in *Handbook of Robotics*. Springer, 2008.
- [10] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 249–254.
- [11] P. Kormushev, S. Calinon, and D. Caldwell, "Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input," *Advanced Robotics*, vol. 25, pp. 581–603, 2011.
- [12] V. Koropouli and S. Hirche, "Learning interaction control policies by demonstration," in *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 344–349.
- [13] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.
- [14] P. Evrard, E. Gribovskaia, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009, pp. 399–404.
- [15] E. Gribovskaia, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *IEEE Intl. Conf. on Robotics and Automation*, 2011.
- [16] L. Rozo, P. Jimenez, and C. Torras, "Robot Learning from Demonstration of Force-based Tasks with Multiple Solution Trajectories," in *International Conference on Advanced Robotics (ICAR)*, 2011, pp. 124–129.
- [17] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, pp. 115–131, 2011.
- [18] D. Mitrovic, S. Klanke, and S. Vijayakumar, "Learning Impedance Control of Antagonistic Systems Based on Stochastic Optimization Principles," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 556–573, 2011.
- [19] D. Braun, M. Howard, and S. Vijayakumar, "Optimal variable stiffness control: formulation and application to explosive movement tasks," *Autonomous Robots*, vol. 1, no. 17, pp. 237–253, 2012.
- [20] M. Howard, D. Mitrovic, and S. Vijayakumar, "Transferring impedance control strategies between heterogeneous systems via apprenticeship learning," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 98–105.
- [21] S. Haddadin, M. Weis, S. Wolf, and A. Albu-Schäffer, "Optimal control for maximizing link velocity of robotic variable stiffness joints," *IFAC World Congress*, pp. 6863–6871, 2011.
- [22] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.
- [23] P. Kormushev, S. Calinon, and D. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," in *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 3232–3237.
- [24] H. Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement." *Science (New York, N.Y.)*, vol. 272, no. 5258, pp. 117–120, Apr. 1996.
- [25] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance." *Nature*, vol. 414, no. 6862, pp. 446–9, Nov. 2001.
- [26] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-Like Adaptation of Force and Impedance in Stable and Unstable Interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.
- [27] N. Hogan, "Impedance and Interaction Control 19.1," *Robotics and automation handbook*, pp. 19–1, 2005.
- [28] B. Paden and R. Panja, "Globally asymptotically stable 'PD+' controller for robot manipulators," *International Journal of Control*, 1988.
- [29] E. Kunesch, F. Binkofski, and H. Freund, "Invariant temporal characteristics of manipulative hand movements," *Experimental Brain Research*, vol. 78, no. 3, pp. 539–546, 1989.
- [30] I. Cathers, N. O'Dwyer, and P. Neilson, "Tracking performance with sinusoidal and irregular targets under different conditions of peripheral feedback," *Experimental brain research*, vol. 111, no. 3, pp. 437–46, Oct. 1996.
- [31] L. Jones, "Kinesthetic sensing," in *Human and Machine Haptics*, 2000.
- [32] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 434–438, 2008.
- [33] S. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with Gaussian Mixture Models," *IEEE Transactions on Robotics*, vol. 27, pp. 1–15, 2011.
- [34] A. Shukla and A. Billard, "Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 424–440, 2011.
- [35] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [36] C. Williams and C. Rasmussen, "Gaussian processes for regression," *Advances in Neural Information Processing Systems*, vol. 8, pp. 514–520, 1996.
- [37] S. Hart and L. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," *Human mental workload*, pp. 139–183, 1988.
- [38] J. Brooke, "SUS - A quick and dirty usability scale," *Usability Evaluation in Industry*, pp. 189–194, 1996.
- [39] M. Garabini, A. Passaglia, F. Belo, P. Salaris, and A. Bicchi, "Optimality principles in stiffness control: The VSA kick," in *IEEE Intl. Conf. on Robotics and Automation*, no. 1, 2012, pp. 3341–3346.
- [40] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems Measurement and Control*, vol. 102, no. 2, pp. 126–133, 1982.



Klas Kronander is a Ph.D. student in the Learning Algorithms and Systems Laboratory (LASA) at the Swiss Federal Institute of Technology in Lausanne (EPFL). He received his M.S. degree in Applied Physics and Electrical Engineering from Linköping Institute of Technology, Sweden, in 2010. His research interests focus on robot control, manipulation and human-robot interaction.



Aude Billard is Associate Professor and head of the LASA Laboratory at the School of Engineering at the Swiss Federal Institute of Technology in Lausanne (EPFL). Prior to this, she was Research Assistant Professor at the Department of Computer Sciences at the University of Southern California, where she retains an adjunct faculty position to this day. Aude Billard received a B.Sc. (1994) and M.Sc. (1995) in Physics from EPFL, with specialization in Particle Physics at the European Center for Nuclear

Research (CERN), an M.Sc. in Knowledge based Systems (1996) and a Ph.D. in Artificial Intelligence (1998) from the Department of Artificial Intelligence at the University of Edinburgh. Her research interests focus on machine learning tools to support robot learning through human guidance. This extends also to research on complementary topics, including machine vision and its use in human-robot interaction and computational neuroscience to develop models of learning in humans.

APPENDIX GAUSSIAN MIXTURE REGRESSION WITH PARAMETRIZED GENERALISATION

This appendix describes a slightly modified version of Gaussian Mixture Regression, which can limit the generalisation in regions poorly covered by the training data and instead fall back on an a priori defined basic distribution in these regions. Consider a GMM with K components that models the joint probability distribution of the variable $\xi \in \mathbb{R}^M$:

$$p(\xi) = \sum_{k=1}^K \pi^k \mathcal{N}(\xi; \mathbf{m}^k, \mathbf{C}^k) \quad (18)$$

where $\pi^k > 0$, $\mathbf{m}^k \in \mathbb{R}^M$, $\mathbf{C}^k \in \mathbb{R}^{M \times M}$ for $k = 1 \dots K$ are the priors, means and covariances of the Gaussians in the GMM. Assume that the modeled variable has an input part and an output part $\xi = [\xi_I, \xi_O]^T$. The distribution of the output conditioned on the input can then be written:

$$p(\xi_O | \xi_I) = \sum_{k=1}^K p(k | \xi_I) \mathcal{N}(\xi_O | \xi_I; \mathbf{m}_{O|I}^k, \mathbf{C}_{O|I}^k) \quad (19)$$

where

$$\mathbf{m}_{O|I}^k = \mathbf{m}_O^k + \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} (\xi_I - \mathbf{m}_I^k) \quad (20a)$$

$$\mathbf{C}_{O|I}^k = \mathbf{C}_O^k - \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} \mathbf{C}_{IO}^k \quad (20b)$$

with

$$\mathbf{m}^k = \begin{bmatrix} \mathbf{m}_I^k \\ \mathbf{m}_O^k \end{bmatrix}, \quad \mathbf{C}^k = \begin{bmatrix} \mathbf{C}_I^k & \mathbf{C}_{IO}^k \\ \mathbf{C}_{OI}^k & \mathbf{C}_O^k \end{bmatrix} \quad (20c)$$

The weighting function $p(k | \xi_I)$ represents the probability of ξ being generated by component k . In standard GMR, this weighting function is defined as:

$$p(k | \xi_I) = \frac{\pi^k \mathcal{N}(\xi_I; \mathbf{m}_I^k, \mathbf{C}_I^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\xi_I; \mathbf{m}_I^i, \mathbf{C}_I^i)} \quad (21)$$

GMR is a functional relationship between the input ξ_I and the output ξ_O , which is achieved by taking the mean of the conditional distribution in Eq. (19):

$$E\{\xi_O | \xi_I\} = \sum_{k=1}^K p(k | \xi_I) \mathbf{m}_{O|I}^k \quad (22)$$

GMR is a convex sum of linear models, weighted non-linearly across the input space. The normalization of the mixing weights in Eq. (21) means that at least one of the linear models will always have a significant contribution to the regression signal. The practical effect of this normalization is generalization, i.e. that the relationships found locally in training data are used in inferring outputs for inputs which lie far from the training data in input space. This is often a desirable effect, since it can reduce the amount of training data needed for good inference. However, sometimes it is inappropriate to generalize in this manner. We propose a novel regression strategy for GMM, which preserves the advantages of the standard GMM/GMR formulation

while it allows to control the degree of generalization. We introduce a basic distribution, which describes the default relationship between the input and the output, $p_b(\xi_O | \xi_I)$. We define the global conditional distribution as:

$$p'(\xi_O | \xi_I) = \sum_{k=1}^K p'(k | \xi_I) \mathcal{N}(\xi_O | \xi_I; \mathbf{m}_{O|I}^k, \mathbf{C}_{O|I}^k) \cdots + \left(1 - \sum_{k=1}^K p'(k | \xi_I)\right) p_b(\xi_O | \xi_I) \quad (23)$$

We have put prime on the quantities that are different compared to the standard GMR formulation. In contrast to Eq. (19), the mixing weights for the components of the GMM does not sum to 1. Instead, $\sum_{k=1}^K p'(k | \xi_I)$ can take any value in the interval $]0, 1[$. The second term describes the contribution from the basic distribution $p_b(\xi_O | \xi_I)$. The key to controlling the generalization are the mixing weights of the components in the GMM, $p'(k | \xi_I)$, which we define as:

$$p'(k | \xi_I) = \frac{\pi^k \mathcal{N}(\xi_I; \mathbf{m}_I^k, \mathbf{C}_I^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\xi_I; \mathbf{m}_I^i, \mathbf{C}_I^i) + (1 - g(\xi_I))} \quad (24a)$$

where

$$g(\xi_I) = \max_k \exp\left(-\frac{1}{2}((\xi_I - \mathbf{m}_I^k)^T (c \mathbf{C}_I^k)^{-1} (\xi_I - \mathbf{m}_I^k))\right) \quad (24b)$$

where $c > 0$ is a constant. Compared to the mixing weights in standard GMR in Eq. (21), our formulation adds a second term in the denominator. This term describes how distant ξ_I is to the closest Gaussian in the GMM. Note that the distance to each Gaussian is computed with the inverse covariance of the Gaussian as metric. For this computation, the covariance is scaled by $c > 0$. The added term in the denominator will cause the mixing weights for the Gaussians in the GMM to go to zero as the input moves far away from the Gaussians, while the weight for the default distribution tends to one. The amount of generalisation is controlled with the parameter c , and as can be seen from Eq. (24), $c \rightarrow \infty$ leads to full generalization (as in the standard formulation of GMR) while $c \rightarrow 0$ gives weights that drop quickly as the query input moves away from the Gaussians. To move between these two extremes, one can for example control c via a map $\phi : [0, 1[\mapsto [0, \infty[$:

$$c = c(z) = \exp\left(\frac{z}{1-z}\right) - 1 \quad (25)$$

where moving z in $[0, 1[$ corresponds to going from minimal to full generalization. To perform regression using this model, we proceed as in standard GMR and take the mean of the distribution in Eq. (23), which yields:

$$E\{\xi_O | \xi_I\} = \sum_{k=1}^K p'(k | \xi_I) \mathbf{m}_{O|I}^k + \left(1 - \sum_{k=1}^K p'(k | \xi_I)\right) \mathbf{m}_b \quad (26)$$

where m_b is the mean of the basic distribution $p_b(\xi_O|\xi_I)$. The basic distribution p_b is a key element of this regression technique. It should represent the basic relationship between the input and the output variable. The conceptual difference to standard GMR is that in regions poorly covered in the demonstrations, and hence in the GMM, we infer from the basic distribution instead of insisting that the GMM should generalize globally. Note that for the regression it is sufficient to define the mean of the basic distribution.