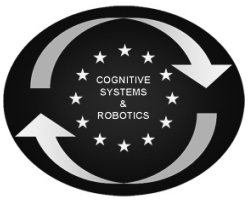
SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION



Project funded by the European Community's 7th Framework Programme (FP7-ICT-2011-7)
Grant Agreement ICT–287513

Deliverable D5.3.1

Learning of force patterns and impedance behaviors

Deliverable due date: 30 October 2014	**Actual submission date:** 30 November 2014
Start date of project: 1 November 2011	**Duration:** 48 months
Lead beneficiary: TUM	**Revision:** DRAFT

Nature: R	**Dissemination level:** CO
R = Report P = Prototype D = Demonstrator O = Other	PU = Public PP = Restricted to other programme participants (including the Commission Services) RE = Restricted to a group specified by the consortium (including the Commission Services) CO = Confidential, only for members of the consortium (including the Commission Services)

Executive Summary

This deliverable of WP5 deals with the problem of learning force and impedance behaviors. This is a key aspect in human-robot interaction, since an always stiff (compliant) robot is not able to co-operate with humans in a safe and profitable manner.

IIT extended the task-parametrized Gaussian mixture model (TP-GMM) proposed in the first part of the project in order to consider human-robot collaborative tasks. For these tasks, in fact, it is of importance to adapt the robot impedance in order to effectively co-operate with the human. The probabilistic encoding provided by TP-GMM is used to determine an optimal feedback control law that exploits the variability in position and force spaces observed during the demonstrations. The whole framework allows the robot to modify its movements as a function of the parameters of the task, while showing different impedance levels. Tests were successfully carried out in a scenario where a 7 DOFs backdrivable manipulator learns to cooperate with a human to transport an object, by generalizing the skill to new initial and target positions.

TUM investigated the possibility of learning position-dependent impedance from human demonstrations. State of the art approaches for learning variable impedance usually generate a time-dependent impedance, and this is not desirable when external perturbation can delay the execution of the task. In the developed approach consists in learning kinematic aspects of a task using Gaussian mixture models and stable dynamical systems. The variability in the demonstrations, retrieved by Gaussian mixture regression, is then used to learn variable impedance behaviors, following the idea that the robot has to be stiff (accurate tracking) when the demonstrations are similar, compliant otherwise.

In redundant manipulators, a compliant (safe) interaction with the environment can be achieved in the robot's null-space without affecting the main task execution. In WP3 UNINA developed a specialized controller, namely the null-space impedance controller, capable to separate end-effector and null-space dynamics to obtain different impedance behaviors. Starting from the null-space impedance controller, TUM developed a Reinforcement Learning system to learn variable null-space impedance behaviors, with a particular focus on generating safe movements in case of unexpected collisions. In this system, the robot tries to avoid possible collision within its null-space. At the same time, the null-space stiffness is decreased while approaching the obstacle to guarantee a safe interaction in case of collisions.

Table of contents

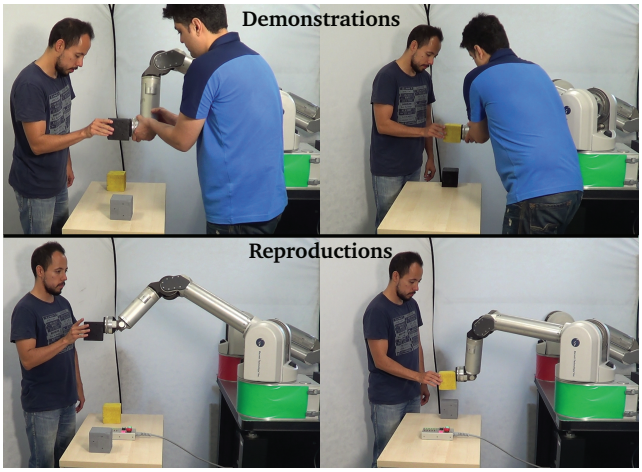


Figure 1: Experimental setting of the human-robot transportation task: (*top*) kinesthetic demonstrations, and (*bottom*) reproduction phase.

1 Collaborative transportation tasks involving force and impedance behaviors

IIT developed a human-robot collaborative transportation experiment in which a robot manipulator learned the cooperative behavior from a set of demonstrations, which are probabilistically encoded by a task-parametri zed Gaussian mixture model (TP-GMM) [2]. Preliminary results were reported in [3].

1.1 Task-parameterized model

Task-parameterized models of movements refer to representations that can automatically adapt to a set of external task parameters. The *task parameters* refer here to the variables that can be collected by the system and that describe a situation, such as positions of objects in the environment or landmark points. The *task parameters* can in some cases be fixed during an execution trial, or they can vary while the motion is executed. The *model parameters* refer to the variables learned by the system, namely, that are stored in memory (the internal representation of the movement). During reproduction, the new set of *task parameters* (description of the present situation) is combined with the *model parameters* (information about the skill) to produce a movement that is not necessary the same as the demonstrations (e.g., adaptation to new positions of objects after having observed the skill in a different situation).

The retrieval of movements from the model parameters and the task parameters is most often viewed as a regression problem. This generality might look appealing at first sight, but it also strongly limits and bounds the generalization scope of these models (mostly interpolation). We showed in [1] that a promising trend in this last category is to exploit the functional nature of the task parameters to build models that can learn the local structures of the task from a low number of demonstrations.

This new approach to task-parametrized models comes from the observation that most of the task parameters can be related to some form of frames of reference, coordinate systems or basis functions, whose structure can be exploited to speed up learning and provide the system with extrapolation capability.

In order to be exploited by a wide range of learning approaches, the proposed model relies on mixtures of Gaussians as core representation. It provides a compact encoding scheme that is also beneficial for storage and stochastic optimization purposes. The task parameters are represented in the general form of coordinate systems, with an origin $\boldsymbol{b}$ (offset vector) and axes concatenated in a matrix $\boldsymbol{A}$. Note that there is no constraint

on the length and orthogonality of the axes, and that $\{\boldsymbol{A}, \boldsymbol{b}\}$ can thus represent any linear transformation. A simple example of task parameterization is to set $\boldsymbol{b}$ as the Cartesian position of an object, and $\boldsymbol{A}$ as the orientation of the object as a direction cosine matrix (rotation matrix).

The demonstrations of a movement/force are simultaneously collected in different coordinate systems. In other words, the same movement is monitored from the perspective of several observers. With multiple demonstrations of the same task or the same category of movements, the variability and correlation information can differ depending on the coordinate system being considered. Typically, the invariant patterns will change during the course of the movement, with transitions between objects, coordinate systems and/or hierarchy constraints.

After training the model, the model parameters for a mixture of K components and for P reference frames are described by $\{\pi_i, \{\boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)}\}_{j=1}^{P}\}_{i=1}^{K}$ (π_i are the mixing coefficients, $\boldsymbol{\mu}_i^{(j)}$ and $\boldsymbol{\Sigma}_i^{(j)}$ are the center and covariance matrix of the i-th Gaussian component in frame j). During reproduction, at each time step t, the present situation (e.g. location of objects) is characterized by the set of reference frames (coordinate systems) $\{\boldsymbol{b}_{t,j}, \boldsymbol{A}_{t,j}\}_{j=1}^{P}$, which is used to evaluate the product of linearly transformed Gaussians

$$\mathcal{N}(\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}) \propto \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{A}_{t,j}\boldsymbol{\mu}_i^{(j)} + \boldsymbol{b}_{t,j},\ \boldsymbol{A}_{t,j}\boldsymbol{\Sigma}_i^{(j)}\boldsymbol{A}_{t,j}^{\top}\Big). \tag{1}$$

The above equation results in a temporary GMM with parameters $\{\pi_i, \boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}\}_{i=1}^{K}$ that is adapted to the current situation, and that can be used to synthesize a new movement. This can for example be done by combining the TP-GMM approach with a regression approach based on *Gaussian mixture regression* (GMR), see [2] for details. Such approach does not only retrieve a trajectory: it also retrieves an estimate of the variations and correlations of the movement/force in the form of a full covariance modeling the output variables, re-estimated at each time step.

1.2 Minimal intervention controller

Similarly as the solution proposed by Medina *et al.* in the context of risk-sensitive control for haptic assistance [4], the predicted variability can be exploited to form a minimal intervention controller [5].

We define the state of the robot as $\boldsymbol{\zeta} = [\,\boldsymbol{x}^{\top}\ \dot{\boldsymbol{x}}^{\top}\ \boldsymbol{f}^{\top}\,]^{\top}$, with $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\boldsymbol{f}$ are the position, velocity and sensed force of the robot. We define the inputs of the system as the vector $\boldsymbol{\nu} = [\,\boldsymbol{u}^{\top}\ \boldsymbol{v}^{\top}\,]^{\top}$, where $\boldsymbol{v}$ represents an external input related to the interaction of the human with the robot during the cooperative task, and $\boldsymbol{u}$ is the control input expressed as

$$\boldsymbol{u} = -\begin{bmatrix} \boldsymbol{K}^{\mathcal{P}} & \boldsymbol{K}^{\mathcal{V}} & \boldsymbol{K}^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \tilde{\dot{\boldsymbol{x}}} \\ \tilde{\boldsymbol{f}} \end{bmatrix}, \tag{2}$$

where $\tilde{\boldsymbol{x}} = (\boldsymbol{x} - \bar{\boldsymbol{x}})$, $\tilde{\dot{\boldsymbol{x}}} = (\dot{\boldsymbol{x}} - \bar{\dot{\boldsymbol{x}}})$ and $\tilde{\boldsymbol{f}} = \big(\boldsymbol{f} - \bar{\boldsymbol{f}}\big)$, with $\bar{\boldsymbol{\zeta}} = [\bar{\boldsymbol{x}}^{\top}, \bar{\dot{\boldsymbol{x}}}^{\top}, \bar{\boldsymbol{f}}^{\top}]^{\top}$ estimated by GMR. $\boldsymbol{K}^{\mathcal{P}}$, $\boldsymbol{K}^{\mathcal{V}}$ and $\boldsymbol{K}^{\mathcal{F}}$ are full stiffness, damping and force gain matrices, respectively.

The state space representation of the robot in task space can be written as[1]

$$\dot{\boldsymbol{\zeta}} = \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}^{\boldsymbol{A}} \boldsymbol{\zeta} + \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}^{\boldsymbol{B}} \boldsymbol{\nu}, \tag{3}$$

[1] $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices defining the dynamical system, not to be confounded with the $\boldsymbol{A}_{t,j}$ and $\boldsymbol{b}_{t,j}$ defining the coordinate systems in the TP-GMM.

namely $\frac{d}{dt}\boldsymbol{x} = \dot{\boldsymbol{x}}$, $\frac{d}{dt}\dot{\boldsymbol{x}} = \boldsymbol{u} + \boldsymbol{f}$, and $\frac{d}{dt}\boldsymbol{f} = \boldsymbol{v}$. Note that the latter equation indicates that the variation of the sensed forces depends on the external input $\boldsymbol{v}$, in other words, the physical interaction between the human and the robot directly influences the variation of the robot's force perception. Lastly, we denote the column space of the input matrix being $\boldsymbol{B} = [\boldsymbol{B}_1\ \boldsymbol{B}_2]$.

Once the reference position, velocity and force profiles are retrieved by the TP-GMM at a given time step, the controller gains can be estimated by following an optimal control strategy. Optimal feedback controllers allow the robot to plan a feedback control law tracking the desired state. The problem is stated as finding the optimal input $\boldsymbol{\nu}$ that minimizes the cost

$$J_t = \sum_{n=t}^{\infty} (\boldsymbol{\zeta}_n - \bar{\boldsymbol{\zeta}}_t)^{\top} \boldsymbol{Q}_t (\boldsymbol{\zeta}_n - \bar{\boldsymbol{\zeta}}_t) \ + \ \boldsymbol{\nu}_n^{\top} \boldsymbol{R}_t \, \boldsymbol{\nu}_n, \tag{4}$$

where $\bar{\boldsymbol{\zeta}}_t$ represents the reference or desired state obtained by GMR, while the matrices $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ are weighting variables that determine the proportion in which the tracking errors and control inputs affect the minimization problem. The aforementioned problem is known as an infinite horizon linear quadratic regulator. The novel use of the above cost is that we exploit the variability observed during the demonstrations to adapt on-the-fly the error costs in (4). Specifically, we define

$$\boldsymbol{Q}_t = \hat{\boldsymbol{\Sigma}}_t^{-1}, \quad \boldsymbol{R}_t = \begin{bmatrix} \boldsymbol{R}_t^u & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_t^v \end{bmatrix}, \tag{5}$$

using the covariances $\hat{\boldsymbol{\Sigma}}_t$ retrieved by GMR. In our experiment, $\boldsymbol{R}_t$ is defined as a diagonal matrix.

This cost is updated at each time step t, and is then used for computing the next control command. This formulation is better suited for interactions in weakly structured environments, where the robot actions might depend on the state and/or actions of its human counterpart, and the state of the environment. Technically, finite-horizon requires the recursive computation of an ordinary differential equation, which is better suited for planning situations in which the candidate frames are not expected to move. In constrast, our minimization problem can be solved through the algebraic Riccati equation, providing an optimal feedback controller in the form of (2) with full stiffness, damping and force gain matrices.

Specifically, the LQR solution for our problem is represented by

$$\boldsymbol{\nu}_t = \boldsymbol{R}_t^{-1} \boldsymbol{B}^{\top} \left[-\boldsymbol{S}_t (\boldsymbol{\zeta}_t - \bar{\boldsymbol{\zeta}}_t) + \boldsymbol{d}_t \right], \tag{6}$$

where the robot controller is obtained as

$$\boldsymbol{u}_t \ = \ \boldsymbol{R}_t^{u^{-1}} \boldsymbol{B}_1^{\top} \left[-\boldsymbol{S}_t \left(\boldsymbol{\zeta}_t - \bar{\boldsymbol{\zeta}}_t \right) + \boldsymbol{d}_t \right], \tag{7}$$

with $\boldsymbol{S}_t$ and $\boldsymbol{d}_t$ as solutions of the equations

$$\boldsymbol{A}^{\top} \boldsymbol{S}_t + \boldsymbol{Q}_t + \boldsymbol{S}_t \boldsymbol{A} - \boldsymbol{S}_t \boldsymbol{B} \boldsymbol{R}_t^{-1} \boldsymbol{B}^{\top} \boldsymbol{S}_t \ = \ \boldsymbol{0}, \tag{8}$$

$$-\boldsymbol{A}^{\top} \boldsymbol{d}_t + \boldsymbol{S}_t \boldsymbol{A} \boldsymbol{\zeta}_t + \boldsymbol{S}_t \boldsymbol{B} \boldsymbol{R}_t^{-1} \boldsymbol{B}^{\top} \boldsymbol{d}_t - \boldsymbol{S}_t \dot{\boldsymbol{\zeta}}_t \ = \ \boldsymbol{0}, \tag{9}$$

and $\boldsymbol{B}_1$ belonging to the column space of $\boldsymbol{B}$, as specified previously. In the above, $\boldsymbol{d}_t$ is the feedforward term (which can optionally be neglected for low dynamic movements). The solution for $\boldsymbol{u}_t$ provides optimal feedback gains $\boldsymbol{K}^{\mathcal{P}}$, $\boldsymbol{K}^{\mathcal{V}}$ and $\boldsymbol{K}^{\mathcal{F}}$, which allow the robot to optimally track its desired state during the cooperative task, shaping the robot's impedance level according to the invariant characteristics of the demonstrations.

1.3 Experimental results

The experiment consists of teaching a robot to simultaneously handle position and force constraints arising when a human and a robot cooperatively manipulate/transport an object (see Fig. 1). At the beginning of the

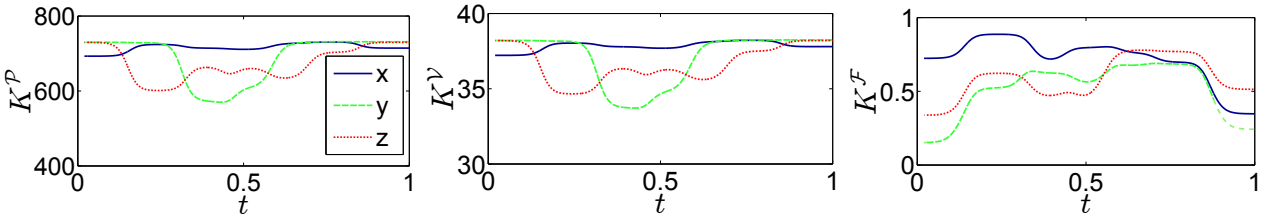


Figure 2: Profiles of the estimated stiffness, damping and force gain matrices along a reproduction of the cooperative transportation. Only their diagonal values are plotted.

transportation task, two participants simultaneously reach for the object. Once they make contact with the load, they start jointly transporting the object along a bell-shaped path to reach the target location. When the object gets to the final position, the two persons release it and move away from it. Note that both the initial and goal object position/orientation may vary across repetitions. The aim is to introduce a robot into such a task by replacing one of the human participants by the robot.

We used a torque-controlled 7 DOFs WAM robot endowed with a 6-axis force/torque sensor. The robot's controller is defined by (2). In the demonstration phase, the gravity-compensated robot is kinesthetically guided by the teacher while cooperatively achieving the task with the other human partner, as shown in Figure 1. The teacher shows the robot both the path to be followed and the force pattern it should use while transporting the load.

Two candidate coordinate systems ($P = 2$) are considered, namely, the frames representing the initial and target locations of the object. During reproduction, the initial and target frames are given to the model. At each time step t, the robot obtains an updated reference state $\bar{\boldsymbol{\zeta}}$ along with optimal stiffness, damping and force gain matrices, that generate a new desired acceleration in the operational space of the robot.

Figure 2 shows how $\boldsymbol{K}^{\mathcal{P}}$, $\boldsymbol{K}^{\mathcal{V}}$ and $\boldsymbol{K}^{\mathcal{F}}$ vary over time along one of the reproductions, with $\boldsymbol{R}_t = r\boldsymbol{I}_{6\times 6}$ and $r = 0.01$. Notice that at the beginning and at the end of the reproduction, the robot behaves less stiffly along the x axis, while being stiffer along the axes y and z. The robot does not allow high variations on the plane yz, guaranteeing that the object is picked up and released by passing through paths that are consistent with the demonstrations. In contrast, as expected, when the human-robot dyad is cooperatively transporting the load with a bell-shaped path, the robot behaves stiffly along x, while allowing some deviations on the plane yz.

The proposed approach brings together the advantages of probabilistic encoding and robustness of optimal control. In the proposed experiment, this allowed the robot to *(i)* automatically extract the constraints of the task from demonstrations (in both position and force spaces), and *(ii)* exploit the observed variability to obtain an optimal feedback law that accordingly shapes the robot impedance along the reproduction of the task.

Here, the proposed model was used to learn a time-driven robot motion. We plan in future work to avoid this explicit time dependency by taking advantage of methods that also encapsulate the sequential information of the task, such as in hidden Markov models. Moreover, we will explore the inclusion of the state of the human into the loop, so that the robot could cope with a wider range of perturbations based on the user's actions.

2 Learning Motion and Impedance Behaviors from Human Demonstrations

TUM developed an approach to learn state-dependent stiffness form human demonstrations [10]. For many tasks, infact, a state-varying, time independent impedance behavior is desirable, being more robust to delays in the execution of the task. Indeed, a time-varying stiffness can fail to provide adequate impedance behaviors

at the right time and in the right place when the execution time changes. Our approach consists in learning a state-dependent stiffness exploring the variability of human demonstrations.

In this section we firstly underline some important concepts concerning the SEDS algorithm [6], used to train a motion primitive in the form of a globally asymptotically stable (GAS) DS. Secondly, we explain how the stiffness is estimated from Gaussian regression. Simulation results are provided to show the effectiveness of the proposed approach.

2.1 Learning stable motion primitives

We assume that the set of N demonstrations $\{\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n}\}_{t=0,n=1}^{T,N}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the position and $\dot{\boldsymbol{x}} \in \mathbb{R}^d$ the velocity, are instances of a first order, nonlinear DS in the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta} \,, \tag{10}$$

where $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a nonlinear continuous function with a unique equilibrium point in $\dot{\boldsymbol{x}}^* = \boldsymbol{f}(\boldsymbol{x}^*) = \mathbf{0}$, and $\boldsymbol{\eta} \in \mathbb{R}^d$ is a zero mean Gaussian noise. Having the noise distribution a zero mean it is possible to use regression to estimate the noise-free model $\dot{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\boldsymbol{x})$.

To estimate the noise-free DS, a probabilistic framework is used that models $\hat{\boldsymbol{f}}$ as a finite mixture of Gaussian functions. Therefore, the nonlinear function $\hat{\boldsymbol{f}}$ is parametrized by the priors $\mathcal{P}(k) = \pi^k$, the means $\boldsymbol{\mu}^k$ and the covariance matrices $\boldsymbol{\Sigma}^k$ of the $k = 1, \ldots, K$ Gaussian functions. The means and covariance matrices are defined by:

$$\boldsymbol{\mu}^k = \begin{bmatrix} \boldsymbol{\mu}_x^k \\ \boldsymbol{\mu}_{\dot{x}}^k \end{bmatrix}, \quad \boldsymbol{\Sigma}^k = \begin{bmatrix} \boldsymbol{\Sigma}_x^k & \boldsymbol{\Sigma}_{x\dot{x}}^k \\ \boldsymbol{\Sigma}_{\dot{x}x}^k & \boldsymbol{\Sigma}_{\dot{x}}^k \end{bmatrix}. \tag{11}$$

A probability density function $\mathcal{P}(\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n}; \boldsymbol{\Theta})$, where $\boldsymbol{\Theta} = [\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^K]$ and $\boldsymbol{\theta}^k = [\pi^k, \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k]$, is associated to each point in the demonstrated trajectories:

$$\mathcal{P}(\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n} | \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi^k \mathcal{N}(\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n} | \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k) \,. \tag{12}$$

Taking the posterior mean probability $\mathcal{P}(\dot{\boldsymbol{x}}|\boldsymbol{x})$ as an estimation of $\hat{\boldsymbol{f}}$ yields [7]:

$$\dot{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{k=1}^{K} h^k(\boldsymbol{x})(\boldsymbol{A}^k \boldsymbol{x} + \boldsymbol{b}^k) \,, \tag{13}$$

where:

$$\begin{aligned} \boldsymbol{A}^k &= \boldsymbol{\Sigma}_{\dot{x}x}^k (\boldsymbol{\Sigma}_x^k)^{-1} \\ \boldsymbol{b}^k &= \boldsymbol{\mu}_{\dot{x}}^k - \boldsymbol{A}^k \boldsymbol{\mu}_x^k \\ h^k(\boldsymbol{x}) &= \frac{\pi^k \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_x^k, \boldsymbol{\Sigma}_x^k)}{\sum_{i=1}^{K} \pi^i \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_x^i, \boldsymbol{\Sigma}_x^i)} \,. \end{aligned} \tag{14}$$

The nonlinear function $\hat{\boldsymbol{f}}$ is then expressed as a nonlinear sum of linear dynamical systems. To guarantee that the DS in Eq. (13) has a GAS equilibrium in $\boldsymbol{x}^*$, the parameters $\boldsymbol{\Theta}$ can be estimated solving the following optimization problem [6]:

$$\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) = -\sum_{n=1}^{N} \sum_{t=0}^{T} \log \mathcal{P}(\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n} | \boldsymbol{\Theta})$$

subject to

$$\left.\begin{array}{l} \boldsymbol{b}^k = -\boldsymbol{A}^k \boldsymbol{x}^* \\ \boldsymbol{A}^k \text{ negative definite} \\ \boldsymbol{\Sigma}^k \text{ positive definite} \\ 0 \leq \pi^k \leq 1 \\ \sum_{k=1}^{K} \pi^k = 1 \end{array}\right\} \quad \forall k \in 1, \ldots, K \tag{15}$$

where $\mathcal{P}(\boldsymbol{x}^{t,n}, \dot{\boldsymbol{x}}^{t,n} | \boldsymbol{\Theta})$ is defined in Eq. (12).

2.2 Learning variable stiffness

The probabilistic framework described in Sec. 2.1 can be also used to retrieve an estimation of the stiffness for each position, following the principle that the robot must be stiff where demonstrations are similar and compliant otherwise. The variability of the demonstrations, at the position level[2], is captured by the mixture regression in the covariance matrices $\boldsymbol{\Sigma}_x^k$ (Eq. (11)). Given the current robot position $\boldsymbol{x}$ we calculate the covariance matrix:

$$\hat{\boldsymbol{\Sigma}}_x = \sum_{k=1}^{K} (h^k(\boldsymbol{x}))^2 \boldsymbol{\Sigma}_x^k \,, \tag{16}$$

where $h^k(\boldsymbol{x})$ is defined in Eq. (14). In practise, we weight the contribution of each matrix using the responsibility $h^k(\boldsymbol{x})$ that each Gaussian has in $\boldsymbol{x}$. The computed covariance matrix is symmetric and positive definite. Hence, we are allowed to calculate its eigenvalues decomposition:

$$\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{E} \boldsymbol{\Lambda} \boldsymbol{E}^{-1} \,, \tag{17}$$

where $\boldsymbol{E}$ is the matrix of the eigenvectors (principal directions) and $\boldsymbol{\Lambda} = \text{diag}(\lambda^1, \ldots, \lambda^d)$ is the diagonal matrix of the eigenvalues.

The root square of the eigenvalues of $\hat{\boldsymbol{\Sigma}}_x$, $\sigma^i = \sqrt{\lambda^i}$, represents the variability (standard deviation) of the data along each direction. We propose to construct the stiffness saving the principal directions of $\hat{\boldsymbol{\Sigma}}_x$, choosing the eigenvalues inversely proportional to σ^i. The stiffness matrix can be written as:

$$\boldsymbol{K} = \boldsymbol{E} \boldsymbol{S} \boldsymbol{E}^{-1} \,, \tag{18}$$

where $\boldsymbol{S} = \text{diag}(s^1, \ldots, s^d)$. Between the eigenvalues s^i of the stiffness matrix and σ^i the following nonlinear inverse relationship holds:

$$s^i(\sigma^i) = \begin{cases} s_{min} & \sigma^i > \sigma_{max} \\ p_1 \left(1 - \tanh\left(p_2\right)\right) + s_{min} & \sigma_{min} \leq \sigma^i \leq \sigma_{max} \\ s_{max} & \sigma^i < \sigma_{min} \end{cases} \tag{19}$$

where

$$p_1 = \frac{s_{max} - s_{min}}{2}, \;\; p_2 = \frac{2k_0}{s_{max}} \left(\sigma^i - \frac{\sigma_{max} - \sigma_{min}}{2} \right) . \tag{20}$$

The stiffness values in each direction are bounded by the tunable parameters s_{min} and s_{max}. The parameters σ_{min} and σ_{min} are also tunable parameters for the learning system. The nonlinear relationship in Eq.

[2]The variability at the velocity level, or between position and velocity is not taken into account. We claim, in fact, that the user can hardly take into account these kinds of variability while he performs the demonstrations.

(19), (20) is shown in Fig. 3. There are two saturation areas corresponding to $\sigma^i > \sigma_{max}$ and $\sigma^i < \sigma_{min}$ where the eigenvalues assume the values s_{min} and s_{max} respectively. Among them there is an almost linear area in which s^i is inversely proportional to σ^i. The size of the saturation areas and, consequently, the slope of the linear part can be modulated by varying k_0. To avoid rapid changes in the stiffness values, the adopted function guarantees a smooth transition between the linear area and the saturation ones.

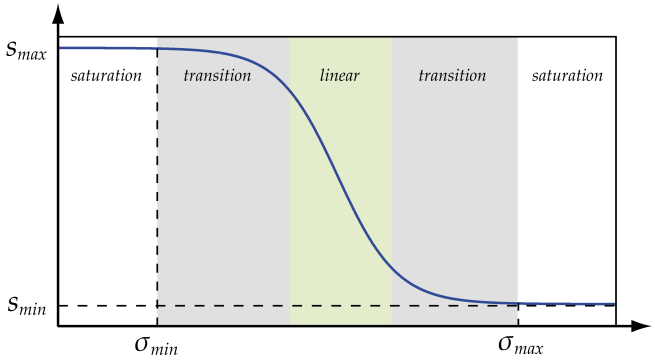


Figure 3: Nonlinear relationship between covariance and stiffness matrices eigenvalues.

2.3 Control law

We assume that our robot can be controlled by an impedance control law (torque feedback) [8]. Given the desired velocity, position and stiffness, the following control law realizes the desired motion-impedance behavior:

$$\boldsymbol{\tau} = \boldsymbol{J}^{\mathsf{T}}\boldsymbol{F} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \,, \tag{21}$$

where $\boldsymbol{\tau}$ is the input torque, $\boldsymbol{J}$ is the Jacobian of the manipulator and $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ compensates the nonlinearities in the dynamical model of the robot.

The force term $\boldsymbol{F}$ is chosen as:

$$\boldsymbol{F} = \boldsymbol{K}\boldsymbol{x} + \boldsymbol{D}\dot{\boldsymbol{x}} \,, \tag{22}$$

where $\boldsymbol{K}$ is the state-dependent stiffness matrix in Eq. (18), $\dot{\boldsymbol{x}}$ is the velocity computed in Eq. (13) and $\boldsymbol{x}$ is obtained integrating $\dot{\boldsymbol{x}}$. The damping matrix $\boldsymbol{D}$ is chosen to have the same eigenvectors (principal directions) of the stiffness matrix (Eq. (18)) with eigenvalues $d^i = 2\sqrt{s^i},\ i = 1, \ldots, d$. Being the DS in Eq. (13) globally asymptotically stable and $\boldsymbol{K}$, $\boldsymbol{D}$ positive definite, the force term drives the robot towards the desired position imposing a state dependent impedance behavior.

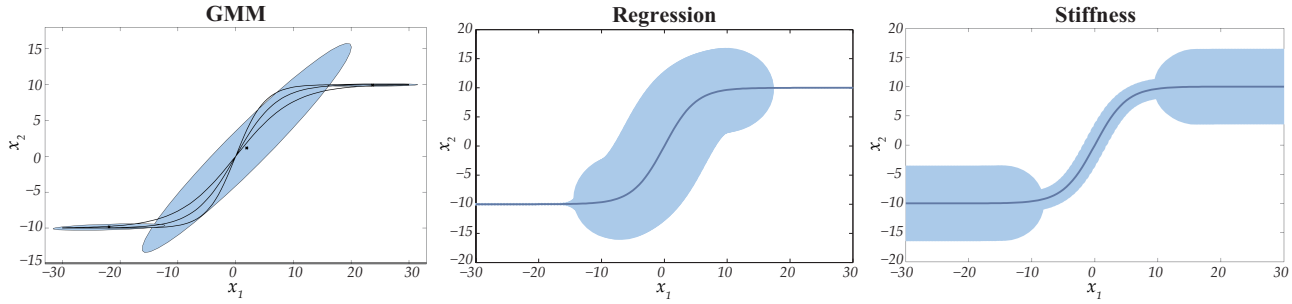
2.4 Simulation Results

In this section we validate the effectiveness of our approach in two cases. The first simulation is used to show how the task constraints are learned from demonstrations. Synthetic 2-dimensional data are used. In the second experiment, a point-to-point task is learned from demonstrations.

Learning task constraints

In this simulation we generate three 2-dimensional position trajectories[3] that are constrained at the beginning and at the end of the motion. The trajectories start almost identical, exhibit variations and end again identical.

[3] The velocity is computed by numerical differentiation.



(a) (b) (c)

Figure 4: Results of the learning algorithm on the synthetic dataset. (a) Demontrations (black lines) and learned model. (b) Smooth motion retrieved using GMR and related covariance matrix $\hat{\mathbf{\Sigma}}_x$. (c) Learned stiffness.

The results obtained learning the DS in Eq. (13) with three Gaussian components are shown in Fig. 4. As expected, the algorithm is able to detect constraints in the demonstrations and to return coherent values in terms of position and stiffness. The covariance matrices $\mathbf{\Sigma}_x^k,\ k = 1, \dots, 3$ are represented as ellipses in Fig. 4(a), where the dimension of each axis of the ellipse is proportional to the standard deviation in that direction. The two covariance matrices close to the constrained areas have a small variance, while the other has a big variance. As a results, the covariance matrix $\hat{\mathbf{\Sigma}}_x$, computed by the regression technique in Sec. 2.2 , has a small standard deviation for points close to the constrained areas, big otherwise (Fig. 4(b)). Conversely, the learned stiffness is big where the motion is constrained and small otherwise (Fig. 4(c)).

Point-to-point motion

In this simulation we learn a point-to-point motion task from human demonstrations. The data, collected by kinesthetic teaching, are shown in Fig. 5(a). The demonstrations have high variability at the beginning of the motion, while they converge to the goal position at the end.

Again, the learning algorithm is able to capture this variability. The two learned covariance matrices $\mathbf{\Sigma}_x^k,\ k = 1, 2$ are represented as ellipsoids in Fig. 5(a), where the dimension of each axis of the ellipsoid is proportional to the standard deviation in that direction. The covariance matrix closer to the initial points in the trajectories has bigger covariance than the other, being the variability at the beginning of the motion considerably bigger. The generated motion, obtained integrating the learned DS velocity with a sample time $\delta t = 0.01s$, is shown in Fig. 5(b). As expected, the trajectory converges to the goal position, being the learned DS globally asymptotically stable. Figure 5(c) shows the eigenvalues of the covariance matrix $\hat{\mathbf{\Sigma}}_x$. The eigenvalues have big values at the beginning of the motion (high variability) and decrease while the trajectory converges to the goal position (low variability). Conversely, the learned stiffness matrix eigenvalues in Fig. 5(d) are small at the beginning of the motion (high variability) and increase while the trajectory converges to the goal position (low variability). The eigenvalues in Fig. 5(d) are obtained firstly scaling the standard deviation along each direction, i.e. the square root σ^i of the eigenvalues of $\hat{\mathbf{\Sigma}}_x$, in the interval $[\sigma_{min} = 1,\ \sigma_{max} = 4]$ and then applying the inverse relationship in Eq. (19) with $[s_{min} = 0,\ s_{max} = 1]$.

The learned DS and stiffness are then used to generate the impedance behavior in Eq. (21)-(22). To this end, we used a dynamic simulator of a KUKA lightweight 7 degree-of-freedom robot [9]. The manipulator end-effector is driven by the learned DS to reach the target position $\boldsymbol{g} = [-0.6\ 0.18\ 0.25]\ m$ (the orientation is kept constant), starting from $\boldsymbol{x}(0) = [-0.56\ \ -0.42\ 0.22]\ m$. The stiffness eigenvalues range is chosen as $[s_{min} = 50,\ s_{max} = 300]$, while the sample time is chosen as $\delta t = 1ms$. For comparison, the same DS is used to drive the robot with a constant stiffness $K = \text{diag}(300, 300, 300)$.

Firstly, we compare the end-effector position error between the executed trajectory and the generated

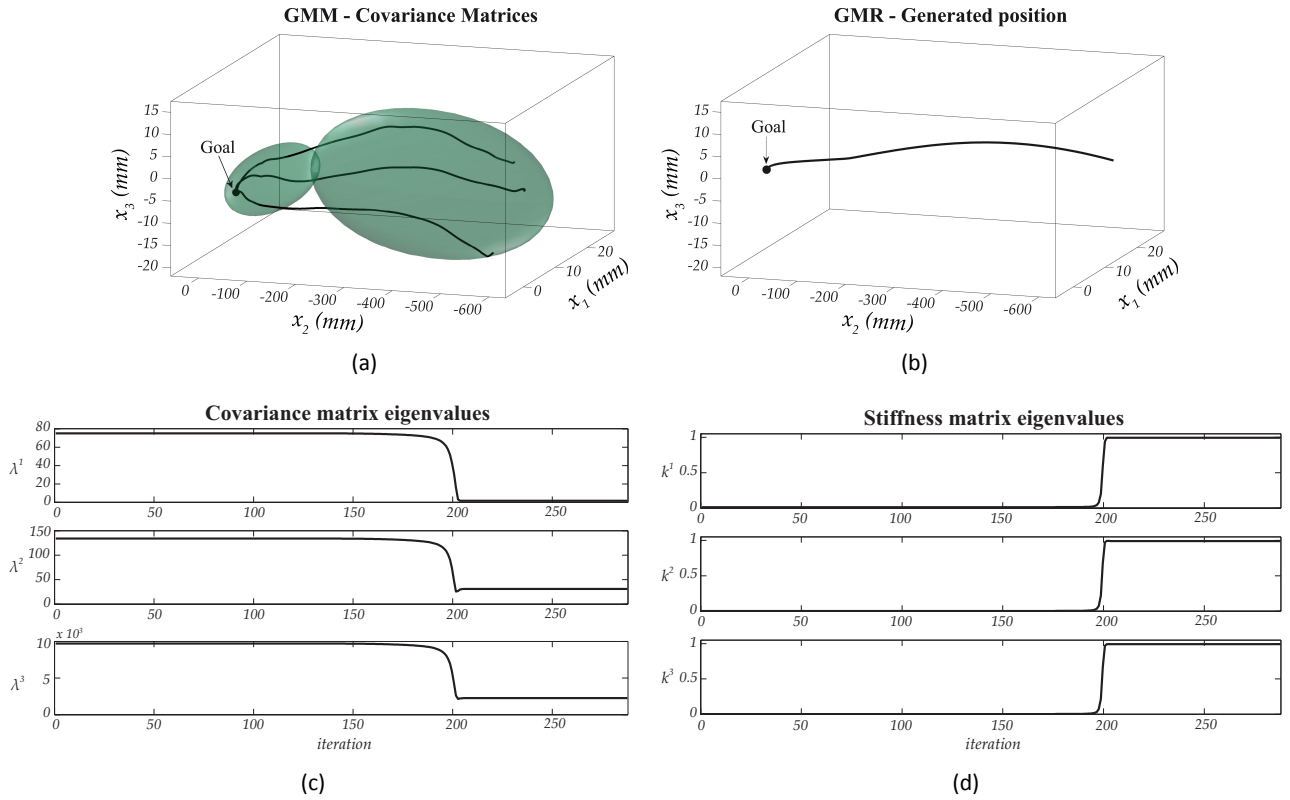


Figure 5: Results of the learning algorithm on the point-to-point motion dataset. (a) Demontrations (black lines) and learned model. (b) Smooth motion retrieved using GMR. (c) Eigenvalues of the covariance matrix $\hat{\boldsymbol{\Sigma}}_x$ for the generated motion. (d) Eigenvalues of the stiffness matrix, normalized to 1.

(integrating the DS) one. As expected, the robot is able to reach the target both with constant and variable stiffness (see Fig. 6). With constant high stiffness, the robot stays significantly closer to the reference trajectory. Hence, as already mentioned, if the goal is an accurate tracking an high stiffness is required.

Secondly, we test the proposed approach when a collision occurs. To simulate a collision, an impulsive external force $f = [20\,0\,0]\,N$ is applied for $5ms$ starting at $t = 0.5s$. As shown in Fig. 7, when the robot has a small stiffness, the external force generates a big deviation from the reference trajectory. In this case, in fact, the robot accomplish the applied force. Instead, with high stiffness, the robot generates higher accelerations at the end-effector to suddenly react to the external disturbance. This results in a considerably smaller deviation, but into a possibly dangerous behavior. Hence, the learned behavior guarantees a compliant and safe interaction with the environment in a certain area of the state space (until a certain distance from the target). The high stiffness close to the target point guarantees instead to reach and keep the desired final position.

3 Learning null-space impedance behaviors

For redundant manipulators, it possible to obtain different impedance behaviors at the end-effector and on the robot body. TUM investigated the possibility to learn safe null-space impedance behaviors [11], i.e. motions that can guarantee a compliant interaction with the environment without affecting the end-effector task.

In this section, we firstly describe the main aspects of the null-space impedance control developed by UNINA in WP3 and refer to [12] for further details. Secondly, we describe the proposed Reinforcement Learning

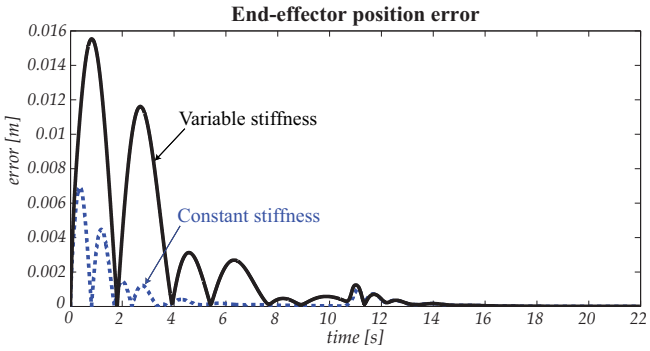


Figure 6: Norm of the end-effector position error with constant (blue dashed line) and variable (black solid line) stiffness.

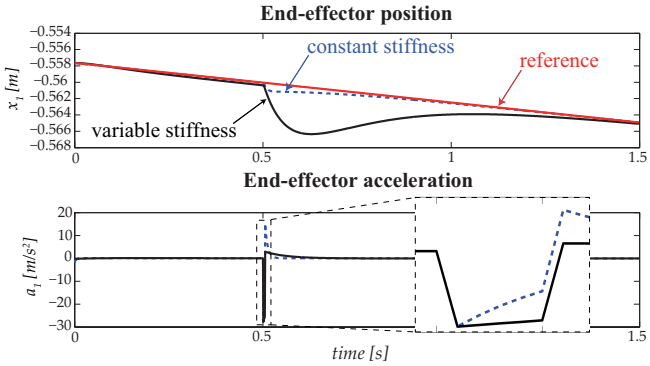


Figure 7: End-effector position and acceleration when an impulsive force of $20N$ is applied along the x_1 direction. To clearly show the effects of the applied force, only the x_1 axis (the most affected) and the first $1.5s$ of the trajectory are considered.

(RL) based stiffness to learn a variable null-space stiffness and present experimental results.

3.1 Null-space impedance control

Robot's redundancy can be solved at the acceleration level using the well-known equation:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\ddot{\boldsymbol{x}}_c - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + \boldsymbol{N}\ddot{\boldsymbol{q}}_{ns} \,, \tag{23}$$

where $\boldsymbol{q}$ is the measured joint position, $\ddot{\boldsymbol{x}}_c$ is the commanded end-effector acceleration, $\ddot{\boldsymbol{q}}_{ns}$ is the desired null-space acceleration, $\boldsymbol{J}$ is the Jacobian matrix, $\boldsymbol{J}^{\dagger}$ is the Jacobian pseudo-inverse and N project $\ddot{\boldsymbol{q}}_d$ in the null-space of $\boldsymbol{J}$. The commanded end-effector acceleration can be chosen according to the impedance law:

$$\ddot{\boldsymbol{x}}_c = \ddot{\boldsymbol{x}}_d + \boldsymbol{D}_{ee}(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) + \boldsymbol{K}_{ee}(\boldsymbol{x}_d - \boldsymbol{x}) \,, \tag{24}$$

where x is the measured position, x_d the desired position, $\boldsymbol{D}_{ee}$ and $\boldsymbol{K}_{ee}$ are the damping and stiffness matrices respectively. The desired null-space acceleration can be chosen as:

$$\ddot{\boldsymbol{q}}_{ns} = \ddot{\boldsymbol{q}}_d + \boldsymbol{M}^{-1}(\boldsymbol{D}_{ns}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_{ns}(\boldsymbol{q}_d - \boldsymbol{q}) - \boldsymbol{\tau}_{ext}) \,, \tag{25}$$

where $\boldsymbol{M}$ is the inertia matrix of the manipulator and $\boldsymbol{\tau}_{ext}$ the applied external torque.

With this choices, the task-space dynamics are decoupled from the null-space dynamics at acceleration level [12]. Then, by properly choosing $\boldsymbol{K}_{ee}, \boldsymbol{K}_{ns}, \boldsymbol{D}_{ee}$ and $\boldsymbol{D}_{ns}$, we can have a compliant behavior of the robot body while ensuring a precise end-effector task execution.

3.2 Reinforcement learning of null-space stiffness

Reinforcement Learning (RL) is one of the most used approach to increase robot's abilities by self-practice. RL is a trial and error process in which the robot explores the environment and its own body. The goal of RL is specified by the reward function, which acts as positive reinforcement or negative punishment depending on the performance of the robot with respect to the desired goal. The reward function is defined by the user according to the particular task. We developed a RL based system capable to modify the learned motion primitives in order to avoid possible collisions. At the same time, null-space stiffness is decreased in the neighborhood of the obstacle to guarantee a safe (compliant) interaction when it is not possible to avoid collisions. In this work, we used the *policy learning by weighting exploration with the return (PoWER)* RL algorithm proposed in [14].

Motion primitives are encoded using a second order dynamical system, namely the Dynamic Movement Primitives (DMP) [13]:

$$\begin{cases} \tau \dot{y} = z \\ \tau \dot{z} = \alpha(\beta(g - y) - z) + f(s) \\ \dot{s} = -\gamma s \end{cases} \tag{26}$$

where y is the position, g is the target position, τ is a scaling factor, the non-linear forcing term $f(s)$ is usually a weighted summation of Gaussian basis function. The additional state s converges to zero ($\gamma > 0$) and guarantees the converges of y to g ($f(s) \to 0$ if $s \to 0$). The forcing term $f(s)$ can be represented as:

$$f(s) = \boldsymbol{g}_{tra}(s)^T(\boldsymbol{\theta}_{tra} + \boldsymbol{\epsilon}_{tra}) \,, \tag{27}$$

where $\boldsymbol{g}(s)$ are the Gaussian basis functions, $\boldsymbol{\theta}_{tra}$ are the *policy parameters* (weights) updated by RL, and $\boldsymbol{\epsilon}_{tra}$ is the exploration noise. To update the stiffness, we assume that the stiffness dynamics is regulated by the first order DS:

$$\dot{\boldsymbol{K}}_{ns} = \gamma_{sti}(\boldsymbol{g}_{sti}(s)^T(\boldsymbol{\theta}_{sti} + \boldsymbol{\epsilon}_{sti}) - \boldsymbol{K}_{ns}) \,, \tag{28}$$

where $\boldsymbol{K}_{ns} = diag(k_x, k_y, k_z)$ is the null-space stiffness matrix (diagonal) and γ_{sti} is a tunable gain. The other quantities have the same definition of those in Eq. 27.

Being our goal to avoid collisions in the null-space while reducing the robot's body stiffness close to the obstacle, we propose the following reward function:

$$\begin{cases} r = w_1 \exp(-\|p - p_d\|) \quad d \geq d_s \,, \\ r = -w_1 \exp(-d) - w_2(k_x + k_y + k_z) \quad d < d_s \end{cases} \tag{29}$$

where d is the robot-obstacle distance, d_s is a safety distance, p is the points of the robot closest to the obstacle, p_d is the desired position of p, k_i are the entries of the stiffness matrix, and w_1 and w_2 are tunable weights.

3.3 Results

Experiments are conducted on a KUKA LWR4+ 7 DoF manipulator [9]. An obstacle is putted along the robot's elbow trajectory in such a way that the collision cannot be avoided in the null-sapce (see Figure 8). By using a constant stiffness (red solid line in Figure) the robot hardly collide with the environment affecting also the end-effector task execution (see Figure). Instead, with the learned variable stiffness (red solid line in Figure), a smooth interaction with the environment is achieved with a resulting better execution of the end-effector task (see Figure).

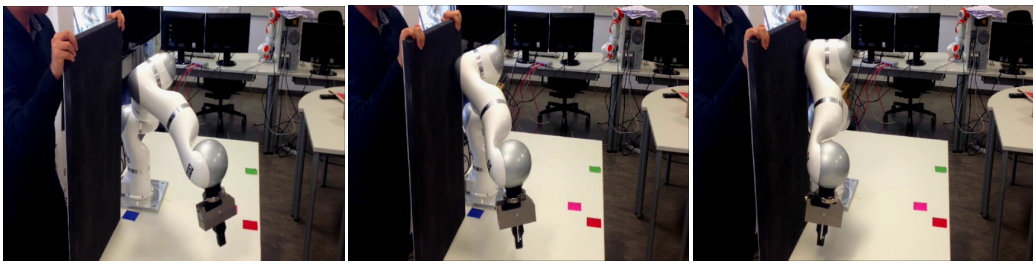


Figure 8: Robot's elbow collide with an obstacle: (left) end-effector task starts, (middle) collision, (right) a compliant behaviour is generated in the null-space to guarantee smooth interaction and end-effector task execution.

References

[1] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," IEEE Intl Conf. on Humanoid Robots, pp. 323–329, 2012.

[2] S. Calinon, D. Bruno and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," IEEE Intl Conf. on Robotics and Automation, pp. 3339–3344, 2014.

[3] L. Rozo, S. Calinon and D. G. Caldwell, "Learning Force and Position Constraints in Human-robot Cooperative Transportation," IEEE Intl Symposium on Robot and Human Interactive Communication, pp. 619–624, 2014.

[4] J. R. Medina, D. Lee and S. Hirche, "Risk-Sensitive Optimal Feedback Control for Haptic Assistance," IEEE Intl Conf. on Robotics and Automation, pp. 1025–1031, 2012.

[5] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," Nature Neuroscience, vol. 5, pp. 1226–1235, 2002.

[6] S. M. Khansari-Zadeh and A. Billard, "Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models," Transaction on Robotics, vol. 27, no. 5, pp. 943–957, 2011.

[7] D. A. Cohn, Z. Ghahramani and M. I. Jordan, "Active Learning with Statistical Models," Journal of Artificial Intelligence Research, vol. 4, no. 1, pp. 129-145, 1996.

[8] N. Hogan, "Impedance control: An approach to manipulation: Part i, ii, iii," Journal of Dynamic Systems, Measurement, and Control, vol. 107, no. 1, pp. 1-24, 1985.

[9] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald and G. and Hirzinger, "The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing," International Symposium on Robotics and German Conference on Robotics, pp. 1–8, 2010.

[10] M. Saveriano and D. Lee, "Learning Motion and Impedance Behaviors from Human Demonstrations," Intl Conf. on Ubiquitous Robots and Ambient Intelligence, 2014.

[11] J. Liu, M. Saveriano and D. Lee, "Robot Compliant Behavior through Reinforcement Learning," 7th International Workshop on Human-Friendly Robotics, 2014.

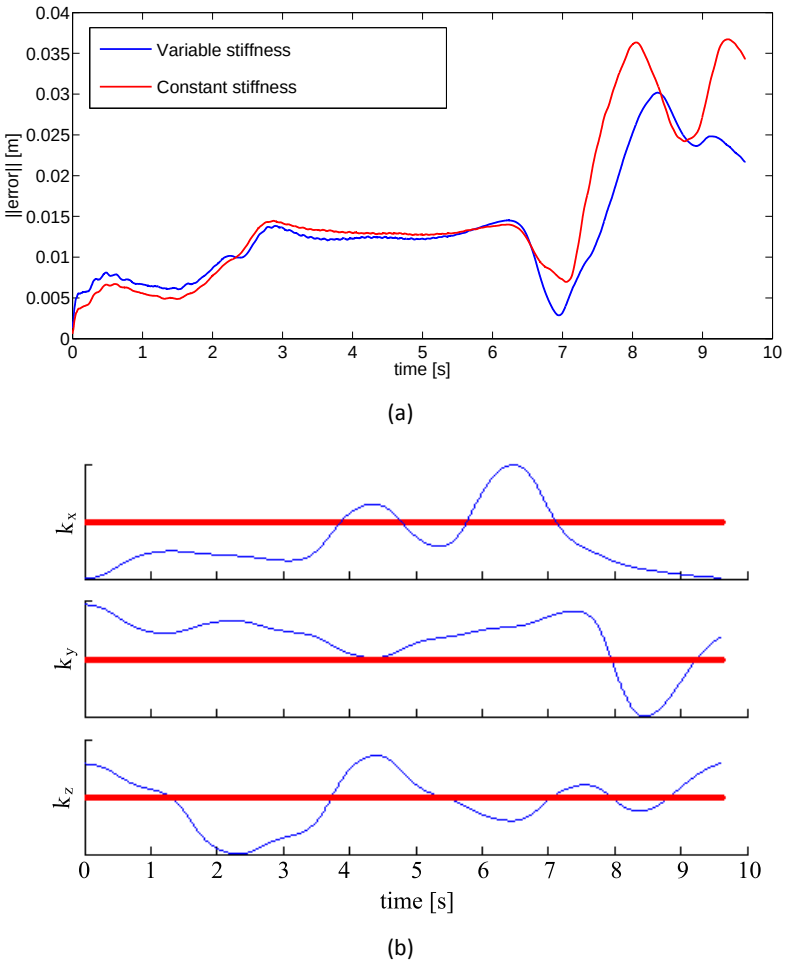


Figure 9: (a) Norm error in task-space. (b) Constant null-space stiffness (red line) and learned variable null-space stiffness (blue line).

[12] H. Sadeghian, L. Villani, M. Keshmiri and B. Siciliano, “Task-Space Control of Robot Manipulators With Null-Space Compliance,” Transactions on Robotics, vol. 30, no. 2, pp. 493–506, 2014.

[13] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann and S. Schaal, “Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors,” Neural Computation, vol. 25, pp.328-373, 2013.

[14] J. Kober and J. Peters, “Learning Motor Primitives for Robotics,” IEEE International Conference on Robotics and Automation, pp. 2112–2118, 2009.