Variable Impedance Control in Cartesian Latent Space while Avoiding Obstacles in Null Space

David Parent, Adrià Colomé and Carme Torras

Abstract—Human-robot interaction is one of the keys of assistive robots. Robots are expected to be compliant with people but at the same time correctly perform the tasks. In such applications, Cartesian impedance control is preferred over joint control, as the desired interaction and environmental feedback can be described more naturally, and the force to be exerted by the robot can be readily adjusted.

This paper addresses the problem of controlling a robot arm in the operational space with variable stiffness so as to continuously adapt the force exerted in each phase of motion according to the precision requirements. Moreover, performing dimensionality reduction we can separate the degrees of freedom (DoF) relevant for the task from the redundant ones. The stiffness of the former can be adjusted constantly to achieve the required accuracy, while task-redundant DoF can be used to achieve other goals such as avoiding obstacles by moving in the directions where accuracy is not critical. The designed method is tested teaching the robot to give water to drink to a model of human head. Our empirical results demonstrate that the robot can learn precision requirements from demonstration. Furthermore, dimensionality reduction is proved to be useful to avoid obstacles.

I. INTRODUCTION

Population ageing poses one of the main social and economic challenges. We live in long-living societies that require a greater effort in the dependency area. Within the European Union, the old-age dependency ratio is expected to rise from 28.1% in 2014 to 49.4% by 2050 [1]. Assistive robots can provide a solution to this problem.

Industrial robots are nowadays already able to cooperate with humans while carrying out complex tasks [2]. However, these tasks are not focused on direct contact between the robot and the human as in assistive robotics. In this context, for instance, a robot has to be able to dress [3] or to feed people [4]. Hence, it is clear that robots need to be compliant and safe. Safety can be eased by using low-inertia robots [5]. These robots have decoupled some motors' inertia from the links' inertia, by placing the heavy motors at the base. This improves safety at high speed because the stored kinetic energy is small. Besides, lightweight materials are used to further enhance safety properties. Even with these characteristics, we need to implement the so-called compliant controllers [6]-[7], i.e., controllers that can accommodate to human contact without



Fig. 1. Teaching the WAM robot to give water to drink.

exerting too much force. This behaviour can be obtained with impedance control [8], where the force exerted by the robot is related to the error in tracking a desired trajectory.

However, in human-robot interaction, we often have different scenarios where the relative importance between the precision of the robot and its compliance might vary. As it is very difficult to perform highly in both respects, a tradeoff between precision and compliance needs to be attained. The precision requirements are usually provided by a motion characterization that encodes a time-varying precision at every part of the trajectory. In the case of Probabilistic Movement Primitives (ProMPs) [9], such precision can be extracted from the time variance through several motion demonstrations to the robot. Nevertheless, ProMPs suffer from the highdimensionality of their parametrical representation. For this reason, we use dimensionality reduction techniques as in [10]. Moreover, [9] presents a stochastic controller that tracks the desired precision at every timestep. However, such controller suffers from a high computational cost if the data used for generating the ProMP is not rich enough to fully characterize its covariance. On the other hand, some authors [11] use characteristics of the motion to regulate the stiffness of the robot, but they do not take into account the requirements introduced by the user.

Using a latent space as defined in [10], we obtain a projection matrix that specifies the relevant variables to do the task, while its null space contains the redundant variables for the task. This distinction is useful to apply separate control laws to the different DoF depending on their importance for the task.

Avoiding obstacles is a problem extensively dealt with in literature. Assistive robots have to be able to detect and adapt to unexpected situations like finding obstacles in their path. In [12], the ProMP definition is adapted to find a new path while avoiding an obstacle, but it presents real-time implementation difficulties. Therefore, if the object moves, the robot cannot easily recalculate a new path. For this reason, other authors [13] use potential fields that are fast to calculate and also

This work was partially developed in the context of the project CLOTHILDE ("CLOTH manIpulation Learning from DEmonstrations"), which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Advanced Grant agreement No 741930). This work is also supported by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI MdM-IP-2018-05.

The authors are with the Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC, Barcelona, Spain. [dparent, acolome, torras]@iri.upc.edu.

offer a solution.

In this work, we propose to reduce the problem to a latent space by selecting the DoF according to their importance in the task. While the compliance of the relevant DoF is adapted in each phase of motion depending on its precision requirements, the task-redundant DoF are forced to remain always compliant, so as to avoid obstacles in the space, as we show in the experimental section.

In Section II we introduce the operational space control problem as well as the definition of the trajectory with ProMPs and its reduced latent space. In Section III we first present our solution to the wrist singularity to work robustly in the Cartesian space. Next we explain how to adapt the compliance in real time of the relevant DoF and how to avoid obstacles using the redundant ones. Finally, in Section IV we present the obtained results with their discussion, followed by some conclusions.

II. PRELIMINARIES

Throughout this work we will study the control problem in the operational space, using ProMPs to encode the trajectories learned from demonstration and an Expectation-Maximization (EM) algorithm [14] to reduce the dimension of the problem. In this section, we introduce these concepts which are going to be used along the paper.

A. Jacobian transpose control

Jacobian transpose control [15] is a well-known method to calculate the torques required to apply at joints to follow an end-effector pose (position and orientation) trajectory. By using the principle of virtual work on the manipulator system, it can be easily seen that the relation between the 6×1 Cartesian force-moment vector acting at the end-effector f(wrench) and the $d \times 1$ vector of torques at joints is the geometric Jacobian of the manipulator:

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{f} + \boldsymbol{g}(\boldsymbol{q}), \tag{1}$$

where the gravity compensation term g(q) depending on the joint configuration q is included.

Note that this equation allows us to work in the operational space without using any inverse kinematics. In this paper, we calculate the forces in Eq. (1) as a proportional-derivative (PD) law (see Figure 2)

$$\boldsymbol{f} = \boldsymbol{K}_{P}\boldsymbol{e} + \boldsymbol{K}_{D}\dot{\boldsymbol{e}}, \qquad (2)$$

where K_P and K_D are the proportional and derivative gains matrices an e and \dot{e} are the position and velocity errors.



Fig. 2. Jacobian transpose control scheme with gravity compensation.

B. Kinematic redundancy control

A manipulator is kinematically redundant when it has more DoF (d) than the dimension of the operational space [16] - 6 in case of Cartesian control -. These extra DoF can be exploited to achieve a secondary goal while performing a task as, for instance, keeping the redundant joints as far as possible from people. To this end, it can be constructed a second controller projecting the secondary goal τ_{null} into the null-space orthogonal projection $(I - J^T J^T^{\dagger})$ [17], obtaining:

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{f} + \boldsymbol{g}(\boldsymbol{q}) + (\boldsymbol{I} - \boldsymbol{J}^T \boldsymbol{J}^{T^{\dagger}}) \boldsymbol{K}_{null} \boldsymbol{\tau}_{null}, \quad (3)$$

where K_{null} is a diagonal matrix used to select the relevance of the secondary goal.

C. Probabilistic Movement Primitives

ProMPs are a stochastic approach to learn and encode a set of similar motion trajectories that present time-dependent variances over time [9]. Given a number of basis functions per DoF, N_f , ProMPs use time-dependent Gaussian kernels Φ_t to encode the state of a trajectory, Φ_t being the vector of normalized kernel basis functions (e.g., uniformly distributed Gaussian basis function over time). Thus, the position and/or velocity state vector \mathbf{y}_t can be represented as

$$\mathbf{y}_t = \boldsymbol{\Psi}_t^T \boldsymbol{\omega} + \boldsymbol{\epsilon}_y, \tag{4}$$

where $\Psi_t^T = I_d \otimes \Phi_t^T$, I_d being the *d*-dimensional identity matrix and Φ_t an N_f -dimensional column vector with the Gaussian kernels associated to one DoF at time *t*. Moreover, $\epsilon_y \sim \mathcal{N}(0, \Sigma_y)$ is a zero-mean Gaussian noise and the weights $\boldsymbol{\omega}$ are also treated as random variables with a distribution $p(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega} | \boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})$.

This distribution can be fitted, given a set of demonstration trajectories, by obtaining the weights ω_j of each demonstration through least squares. Subsequently, the parameters of the distribution $\theta = \{\mu_{\omega}, \Sigma_{\omega}, \Sigma_{y}\}, \Sigma_{y}$ being the state covariance, are fitted by means of a maximum likelihood estimate, i.e., computing the sample mean and the sample covariance of ω . We know that the distribution for our current state \mathbf{y}_t is Gaussian with mean $\boldsymbol{\mu} = \boldsymbol{\Psi}_t^T \boldsymbol{\mu}_{\omega}$ and covariance

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_\omega \boldsymbol{\Psi}_t. \tag{5}$$

ProMPs also provides a model-based stochastic controller that reproduces the encoded trajectory distribution [9].

D. Dimensionality reduction for ProMP

Given a robot working in the 6 DoF Cartesian space, we can reduce the dimensionality of its motion representation to a latent space [10] of dimension r, which is manually given. We can express the robot's state vector \mathbf{y}_t with latent space variables \mathbf{x}_t as

$$\mathbf{y}_t \simeq \mathbf{\Omega} \mathbf{x}_t,$$
 (6)

where Ω (6 × r), will be used throughout this paper as the coordination matrix, a linear mapping from an r-dimensional Cartesian latent virtual space into the 6-dimensional Cartesian space. We select r as the minimum number of variables so the information lost doesn't reach a 5 %.

In order to represent the trajectory as a linear combination of some parameters ω as in ProMP, Eq. (6) is written as

$$\mathbf{y}_t = \mathbf{\Omega}\mathbf{x}_t + \boldsymbol{\epsilon}_{fit} = \mathbf{\Omega}\left(\mathbf{\Phi}_t^T \boldsymbol{\omega} + \boldsymbol{\epsilon}_x\right) + \boldsymbol{\epsilon}_{fit}, \qquad (7)$$

with Φ_t being the $N_f \times 1$ matrix with the kernels used for the trajectory, and ϵ_{fit} , ϵ_x the DR fitting error and the Gaussian noise for **x**, respectively. Thus, the probability of being in the latent state \mathbf{x}_t given the weights $\boldsymbol{\omega} = [\boldsymbol{\omega}_1^T, ..., \boldsymbol{\omega}_r^T]^T$, $(rN_f \times 1)$ is

$$p(\mathbf{x}_t|\boldsymbol{\omega}) = \mathcal{N}(\mathbf{x}_t|\boldsymbol{\Psi}_t^T\boldsymbol{\omega}, \boldsymbol{\Sigma}_x), \qquad (8)$$

with $\Psi_t^T = I_r \otimes \Phi_t^T$ $(r \times rN_f)$. In this case the set of parameters of the DR-ProMP representation are $\theta = \{\mu_{\omega}, \Sigma_{\omega}, \Omega, \Sigma_y\}$, with $\Sigma_y = \Sigma_{fit} + \Omega \Sigma_x \Omega^T$ the total system noise, being Σ_x the latent space covariance matrix, computed as

$$\boldsymbol{\Sigma}_x = \boldsymbol{\Psi}_t^T \boldsymbol{\Sigma}_w \boldsymbol{\Psi}_t. \tag{9}$$

III. PROPOSED METHOD

Anthropomorphic robotic arms are widely used for cooperative tasks because of their motion and appearance similarities to humans arms. These resemblance makes social connections with humans easier. One example of it is the WAM arm. It has 7 revolute joints implemented in three groups similarly to a human arm, as shown in Figure 3.



Fig. 3. Joints in the WAM arm.

A. Wrist singularity control

Working in the operational space with non-redundant robots implies that the Jacobian matrix can lose rank when two or more axis are aligned, known as kinematic singularity. In the WAM robot, since it is redundant, the Jacobian always has full rank inside its workspace¹. Even though, a configuration where two axis are aligned is not desirable.

In the wrist singular configuration (joint 5 and 7 aligned), the elbow redundancy is used to completely orientate the end-effector. As a result, the null space changes: instead of including the elbow redundancy directions, it includes the directions of the axis aligned (joint 5 and 7).

In this situation wrist vibrations are observed because joint 5 and 7 swap easily, as they are expressing the same rotation

and the wrist has low inertia. To avoid these vibrations we propose to control the kernel of the Jacobian matrix. A torque vector τ_{null} proportional to the gradient of a function H_w is defined. This function expresses how fast joint 5 and 7 are moving, therefore the goal is to minimize this function so the wrist joints moves as minimum as possible and, hence, avoid its swap.

$$\boldsymbol{\tau}_{null} = -\nabla(H_w)\mu H_w = (\dot{\theta}_5^2 + \dot{\theta}_7^2); \quad \mu = e^{-4\theta_6^2}$$
 (10)

Note that a regulating factor μ is defined. This variable is 1 when the robot is in the singular configuration ($\theta_6 = 0$) and is 0 when is far from it.

B. Compliance adaption using ProMPs

Some authors compute covariance matrices from the residual errors of the least-squares estimation for each state of the different demonstrations [18]. They use this covariance to calculate the stiffness matrix using eigendecomposition.

We have used a similar idea but with the covariance of the Cartesian variables obtained from ProMPs in Eq. (5). This matrix is an indicator of the precision requirements in each phase of motion: if a variable has a low variability the robot should be precise to repeat the same path taught by demonstration. Therefore, the variability introduced by the user can be used to calculate the gains of Eq. (2).

For this purpose, the orientation part of Σ_t has to be reconstructed (see Appendix). Then, we propose to extract from Σ_t the covariance of the position and orientation independently, not taking into account the coupling terms. By doing this, we can separate the position and orientation gains to give them different magnitudes. Then, computing their inverses we can link the precision needed (Σ_t) to the compliance of the robot (K_P), obtaining a stiffness matrix with a structure

$$\boldsymbol{K}_{P} = \begin{bmatrix} \boldsymbol{K}_{P}^{pos} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{K}_{P}^{ori} \end{bmatrix}$$
(11)

where each sub-matrix is a 3x3 matrix.

Linking these two matrices is not straightforward. Since the scale of Σ_t^{-1} and K_P is not the same, [18] proposes to use the eigendecomposition of Σ_t^{-1} to take its eigenvalues and rescale them to obtain the stiffness in the desired range $[K_{min}, K_{max}]$ based on the range of the eigenvalues of the trajectory $[\lambda_{min}, \lambda_{max}]$ as

$$D_i = K_{min} + (K_{max} - K_{min}) \frac{\ln(\lambda_i) - \ln(\lambda_{min})}{\ln(\lambda_{max}) - \ln(\lambda_{min})}.$$
 (12)

Note that we use a logarithmic scale to reduce the effect of the extremely low variability at the initial point compared to the variability along the trajectory. This is due to the fact that in the learning process the robot moves itself to the initial point and the precision is always higher than that introduced by demonstration.

With this result, the position and orientation gain matrix can be calculated using the eigendecomposition as

$$\boldsymbol{K}_{P}^{pos/ori} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1}.$$
(13)

¹Not considering the workspace limits where the robot is outstretched and in practice will never pass through this point.

Up to this point, if there is a change in the environment, since the impedance is time indexed, the robot would not be able to properly react to it. This becomes critical for interaction tasks. We propose to compute the trajectory time as

$$t_{k+1} = t_k + \frac{\Delta t}{1+e^2} pauseflag, \tag{14}$$

where the elapsed time between two trajectory points depends on the tracking error e and a *pauseflag* that can be set to zero when the error is high. By doing this time modification we make the controller indirectly dependent on the state of the robot obtaining a safest movement.

C. Latent space control

Dimensionality reduction techniques allows us to extract the relevant DoF of the task, while the task-redundant DoF are contained in the null Latent space. Using this division, we have implemented two independent control laws. The relevant DoF are controlled continuously to adapt its stiffness according to the requirements whereas the task-redundant DoF are kept compliant all the time.

To this end, the actual and desired trajectory are mapped into the latent space obtaining x(q) and x_d . Using a PD law it is defined the control law in the latent space as

$$\boldsymbol{\nu} = \tilde{\boldsymbol{K}}_P(\boldsymbol{x}_d - \boldsymbol{x}(\boldsymbol{q})) + \tilde{\boldsymbol{K}}_D(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}(\boldsymbol{q})), \quad (15)$$

where \tilde{K}_P , $\tilde{K}_D \in \mathbb{R}^{r \times r}$. These matrices are obtained from the compliance adaption technique (sec III-B) but reduced into the latent space using

$$\tilde{\boldsymbol{K}}_{P/D} = \boldsymbol{\Omega}^{\dagger} \boldsymbol{K}_{P/D} \boldsymbol{\Omega}.$$
(16)

Finally, the reduced control law is decoded to the Cartesian space and a gravity compensation term g(q) is added:

$$\boldsymbol{\tau}_{lat} = \boldsymbol{J}^T \boldsymbol{\Omega} \, \boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{q}). \tag{17}$$

With Eq. (17) the relevant DoF are controlled, leaving the redundants free. These can be regulated using the orthogonal projection into the kernel of the latent space as

$$\boldsymbol{\tau}_{null} = \boldsymbol{J}^T (\boldsymbol{I} - \boldsymbol{\Omega} \ \boldsymbol{\Omega}^{\dagger}) \boldsymbol{f}_{null}. \tag{18}$$

Taking equations (17) and (18) we obtain the control law

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{lat} + \boldsymbol{\tau}_{null} = \boldsymbol{J}^T \boldsymbol{\Omega} \, \boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T (\boldsymbol{I} - \boldsymbol{\Omega} \, \boldsymbol{\Omega}^{\dagger}) \boldsymbol{f}_{null}.$$
(19)

It is important to note that the force f_{null} will only have effect on the task-redundant DoF. As said before, the main idea is to maintain as compliant as possible the redundant DoF but the force vector can be constructed to achieve other goals, as avoiding obstacles in the Cartesian space.

1) Compliance of the task-redundant DoF

To regulate the stiffness of the task-redundant DoF, we propose to use a proportional law to the position error and impose low eigenvalues to the stifness matrix ($K_{P_{null}}$), so a soft behaviour is achieved.

$$\boldsymbol{f}_{null} = \boldsymbol{f}_{compliant} = \boldsymbol{K}_{P_{null}} \boldsymbol{e}.$$
 (20)

2) Avoiding obstacles

As the redundant DoF of the task are not relevant to perform the trajectory, they can be used to avoid obstacles. Obstacles in the Cartesian space can be defined as a point and a safety distance, considering that a perfect sphere covers the object. This simple description allows us to create a force defined as

$$\boldsymbol{f}_{null} = \boldsymbol{f}_{avoid} = K_{avoid} \frac{TCP - xyz}{\|TCP - xyz\|}, \qquad (21)$$

where the force is pointing from the center of the obstacle (xyz) to the end-effector (TCP) with constant magnitude (K_{avoid}) , manually defined by the user.

This repulsive force has to be regulated according to the distance between the end-effector and the object, increasing the force when the end-effector gets closer to the obstacle. To do so, a parameter λ is defined with a Bézier basis function, accomplishing a C^2 spline as represented in Figure 4



With Eq. (22) we have that when the distance is very small the force is the maximum (K_{avoid}) and when distance increases, the force smoothly decreases.

Even so, when the robot is far from the obstacle the taskredundant DoF have to also be regulated with Eq. (20). To combine both goals, we propose to use the λ parameter defined before to impose the predominance of $f_{compliant}$ or f_{avoid} according to the distance to the object as

$$\begin{aligned} \boldsymbol{f}_{nulli} &= \boldsymbol{f}_{compliant_i} \left(1 - \lambda \right) + \boldsymbol{f}_{avoidi} \lambda \, \alpha \qquad i = 1, 2, 3. \\ \boldsymbol{f}_{nulli} &= \boldsymbol{f}_{compliant_i} \qquad \qquad i = 4, 5, 6. \end{aligned}$$

$$(23)$$

Note that the regulating factor λ is only applied to the position components because the orientation of the end-effector is not used to avoid the obstacle. Moreover, a parameter α restricts the force: f_{avoid} only has effect when the end-effector is getting closer to the object, but the force disappears when the robot moves away.

We propose to calculate the deactivation parameter α as

$$\alpha = \frac{1}{2} (1 - sign(\vec{\boldsymbol{r}}_t \cdot \vec{\boldsymbol{v}}_a)), \qquad (24)$$

where \vec{r}_t is the normalized radius vector pointing from the center of the obstacle to the end-effector and \vec{v}_a is the projection of the end-effector's velocity (\vec{v}) onto the direction of the radius vector, calculated as

$$\vec{\boldsymbol{v}}_a = \|\boldsymbol{v}_a\|\hat{\boldsymbol{r}}_t = \frac{\vec{\boldsymbol{v}}\cdot\vec{\boldsymbol{r}}_t}{\|\vec{\boldsymbol{r}}_t\|^2}\vec{\boldsymbol{r}}_t$$
(25)



Fig. 5. Euclidean norm of the columns of the stiffness matrix. On the left, position stiffness; on the right, orientation stiffness.

As \vec{v}_a and \vec{r}_t are co-linear, the scalar product in Eq. (24) results in $\alpha = 1$ when \vec{r}_t and \vec{v}_a have opposite sign (endeffector getting close to the obstacle) and $\alpha = 0$ if the have the same sign. Therefore f_{avoid} will be zero when the distance between the robot and the obstacle is increasing.

IV. EXPERIMENTATION

To evaluate the proposed method, we decided to teach the robot to give water to a model of a human head (Fig. 1). In this task the position coordinates can be compliant when the robot approaches the person, but must be rigid when the person is drinking to prevent the movement of the glass. On the contrary, the orientation coordinates have to be stiff when the robot approaches to avoid pouring the water into the person, but should be compliant to accommodate when the person drinks.

A. Parameters pre-calculation

To achieve the desired compliant behaviour we have used the technique explained in Section III-B. Nevertheless, the computational cost of the covariance matrix and the stiffness matrix is very high (see Appendix). Once we have it, position and orientation sub-matrices are extracted and, for each one, we invert them and calculate their eigendecomposition. Finally, the stiffness matrix can be calculated with Eq. (13). Performing all these operations in real-time is computationally expensive and can threaten to break the real-time loop.

We propose to pre-calculate the eigendecomposition of the position and orientation covariance and store them with a sampling time of T = 0.1s. With this, we are able to adapt the impedance of the pose of the end-effector in real time.

B. Results

1) Compliance adaptation

As a first test, we executed the ProMP trajectory with the Cartesian controller adapting the stiffness matrix, without reducing the dimensionality. We have set a range for the position stiffness (see Eq. (12) between $[K_{min}, K_{max}] = [150, 1000]$ and for the orientation $[K_{min}, K_{max}] = [0.5, 6]$ (all values decided according to previous experiences).

The Euclidean norms of each column of the stiffness matrix are shown in Fig. 5. We observe that when the robot moves, the position stiffness decreases and the orientation remains more or less constant. On the other hand, when the person is drinking, the position stiffness remains high while one of the orientation rigidities (Kp_{ϵ_y}) reduces. This one that decreases is the rotation associated with the gesture of giving water. The results are also shown in the video included as supplemental material.

2) Avoiding obstacles

Even if the results of the previous section are good, we are controlling the six variables of the Cartesian space. In this task there are some DoF that do not require variable impedance and can be regulated independently. For this reason, we have reduced the dimension into a latent space. Performing a singular value decomposition of the whole trajectory we have seen that with only 3 DoF we can already have the 95% of the information, calculated as the ratio between the 3 first singulars values and the sum of all of them.

Moreover, we use the null space of the projection to avoid obstacles as explained in Section III-C. We put an obstacle of 5 cm of radius in the middle of the trajectory and we have obtained the results presented in Figure 6. We can see that the end-effector avoids the obstacle in the vertical direction (contained in null space of the projection) and then it returns properly to the desired trajectory.



Fig. 6. Avoiding one obstacle while giving water. In black, reference trajectory; in blue, performed trajectory.

To extend the method to multiple obstacles, we put another object in the trajectory. Adding other objects to Eq. (23), we obtain the equation

$$\boldsymbol{f}_{null} = \boldsymbol{f}_{compliant} \prod_{p=1}^{n_p} (1 - \lambda_p) + \sum_{p=1}^{n_p} \boldsymbol{f}_{avoidp} \, \lambda_p \, \alpha_p,$$
(26)

where n_p are the number of obstacles and each one has its regulation parameter (λ_p) , deactivation parameter (α_p) and force to avoid it (f_{avoid_p}) . Implementing this last equation, we obtain the results of Figure 7. We can see that the endeffector avoids both obstacles and when there is not any one near, it returns to the reference position.



Fig. 7. Avoiding two obstacles while giving water. In black, reference trajectory; in blue, performed trajectory.

Other experiments testing the Cartesian control, variable impedance technique and latent space control are shown in: http://bit.ly/2lLMl4n. We can see how accurate is trajectory tracking with the designed controller, how impedance is regulated and also the effects of separating the DoF in the latent space.

C. Discussion

The obstacle avoidance proposed is done on the kernel of the dimensionality reduction matrix. As such kernel projection results in a lower-priority, one can wonder whether the obstacle avoidance will always be feasible or not. But in fact, looking at the xyz components, in order not to be able to avoid an object - assuming we are avoiding a bounding sphere of a given radius -, it should be in the situation where the normal plane to the radius vector \vec{r} defined in Sec. III fully belongs to the projection matrix Ω image space. However, the direction of motion is also most likely to belong to the image of Ω , resulting in a three-dimensional manifold for the cartesian position, thus yielding no dimensionality reduction on the trajectory position - which makes the proposed method of smaller applicability - . Therefore, it is a very unlikely situation that, even if it would happen, the compliant control would prevent the robot from being too aggressive.

V. CONCLUSIONS

In this work we proposed a compliant Cartesian control strategy. It uses the variability introduced by the user when teaching a task to the robot (giving water to a head model) to adapt the stiffness according to the precision required in each phase of motion. Furthermore, we have used Dimensionality Reduction (DR) to avoid obstacles using the redundant DoF of the task while constantly adapting the stiffness of the relevant DoF. The obtained results show a suitable tradeoff between precision and compliance. We also provided visual demonstration of the proposed method tested on a WAM robot. Future work will include exploiting the redundant DoF to achieve other goals as improving manipulability or adaptation to external disturbances.

APPENDIX: ORIENTATION

Quaternion reduction

We use unit quaternions to orientate the end-effector. This orientation representation has four components ($Q = [\eta, \epsilon_x, \epsilon_y, \epsilon_z]$) although orientation has only 3 DoF. This difference involves a problem when working with reinforcement learning (RL). In RL we should use the minimum number of variables because more parameters imply exploring a bigger space. Furthermore, if we use the complete quaternion then the sampled model in RL can give as a result a quaternion not expressing a truly orientation. For these reasons, we have used only the vector part of the quaternion in RL and afterwards reconstructing the scalar component. Even though, when we remove the scalar part we lose the sign². To ensure not changing from the positive to the negative covering, we express the orientation in the end-effector frame using the initial rotation matrix R0. In this frame, the changes in the angles are smaller than in the base-frame so we can prevent from moving from the positive to the negative covering.

To do this process we use that a quaternion can also be expressed [20] as

$$\mathcal{Q} = e^{\frac{\theta}{2}(u_x \boldsymbol{i} + u_y \boldsymbol{j} + u_z \boldsymbol{k})} = \cos \frac{\theta}{2} + (u_x \boldsymbol{i} + u_y \boldsymbol{j} + u_z \boldsymbol{k}) \sin \frac{\theta}{2},$$
(27)

where we can clearly see that the scalar and the vector parts are not independent. Therefore, we can reconsciruct the scalar part with

$$\eta = \cos(\arcsin(\|\epsilon_x \epsilon_y \epsilon_z\|)). \tag{28}$$

Covariance matrix conversion

As said before, ProMP calculates the covariance between the vectorial part of the quaternion in the end-effector frame. To calculate the stiffness matrix as in Section III-B, we need to have the covariance of the geometric orientation in the base frame.

In first place, we have estimated the covariance between the scalar part of the quaternion and the other parameters. To do so, we have created 10 samples with multivariate normal distribution $\tilde{y}_t \sim \mathcal{N}(y_t, \Sigma_t)$. In each sample, the scalar part of the quaternion is reconstructed using Eq. (28). This process allows us to compute the covariance between all the parameters with the scalar part and construct the 7th row and 7th column of Σ_t .

Having the covariance of the complete quaternion, it can be transformed into the geometric orientation covariance. Using that the quaternion rate matrix connects the angular velocities and the temporal derivatives of the quaternion as $\omega = 2H\dot{Q}$ [21], we can also calculate the geometric orientation covariance as

$$\boldsymbol{\Sigma}_{t}^{geo} = \begin{bmatrix} \boldsymbol{I}_{3} \\ & 2\boldsymbol{H} \end{bmatrix} \boldsymbol{\Sigma}_{t}^{quat} \begin{bmatrix} \boldsymbol{I}_{3} \\ & 2\boldsymbol{H} \end{bmatrix}^{T}$$
(29)

where the quaternion rate matrix is

$$\boldsymbol{H} = \begin{bmatrix} -\epsilon_x & \eta & -\epsilon_z & \epsilon_y \\ -\epsilon_y & \epsilon_z & \eta & -\epsilon_x \\ -\epsilon_z & -\epsilon_y & \epsilon_x & \eta \end{bmatrix}.$$
 (30)

Finally, this covariance matrix is rotated from the endeffector frame to the base frame using

$$\boldsymbol{\Sigma}_{t}^{Cart} = \begin{bmatrix} \boldsymbol{I}_{3} & \\ & \boldsymbol{R}\boldsymbol{0} \end{bmatrix} \boldsymbol{\Sigma}_{t}^{geo} \begin{bmatrix} \boldsymbol{I}_{3} & \\ & \boldsymbol{R}\boldsymbol{0} \end{bmatrix}^{T}$$
(31)

²Quaternions represent a double covering of the group SO(3). As a consequence, Q = -Q. Without the scalar part we cannot know in wich covering is located the quaternion.

REFERENCES

- Eurostat (European Comission). "People in the EU: who are we and how do we live?" *EU publications*, 2015.
- [2] A. Tellaeche, I. Maurtua and A. Ibarguren, "Human robot interaction in industrial robotics. Examples from research centers to industry", *IEEE 20th Conference on Emerging Technologies & Factory Automation* (*ETFA*), Luxembourg, pp. 1-6, 2015.
- [3] G. Canal, G. Alenyà and C.Torras. "Adapting robot task planning to user preferences: an assistive shoe dressing example." *Autonomous Robots*, pp. 1343-1356, 2018.
- [4] M.Vila, G. Canal and G. Alenyà. "Towards safety in physically assistive robots: eating assistance". *Proceedings of the 2018 IROS Workshop on Robots for Assisted Living.*, p. 1-4, 2018.
- [5] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero and J. Pérez-Oria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," *IEEE Access*, vol. 5, pp. 26754-26773, 2017.
- [6] A. Colomé, A. Planells and C. Torras, "A friction-model-based framework for Reinforcement Learning of robotic tasks in non-rigid environments," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, pp. 5649-5654, 2015.
- [7] A. Albu-Schäeffer, O. Eiberger, M. Grebenstein, S. Haddadini, C. Ott, T. Wimböck, S. Wolf and G. Hirzinger. "Soft robotics: From Torque Feedback Controlled Lightweight Robots to Intrinsically Compliant Systems". *IEEE Robotics and Automation Magazine*. vol. 15, pp. 20-30, 2008.
- [8] N. Hogan. "Impedance Control: An Approach to Manipulation: Part I—Theory." ASME J. Dyn. Sys., Meas., Control. 107(1): 1–7, March 1985.
- [9] A. Paraschos, G. Neumann, C. Daniel and J. Peters, "Probabilistic movement primitives". *In Advances in NIPS*, pp. 2616-2624, 2013.
- [10] A. Colomé, G. Neumann, J. Peters and C. Torras. "Dimensionality reduction for probabilistic movement primitives", *IEEE-RAS Humanoid Robots*, pp. 794-800, 2014.
- [11] F. Ficuciello, L. Villani and B. Siciliano, "Variable Impedance Control of Redundant Manipulators for Intuitive Human–Robot Physical Interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850-863, Aug. 2015.
- [12] A. Colomé and C. Torras, "Demonstration-free contextualized probabilistic movement primitives, further enhanced with obstacle avoidance," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, pp. 3190-3195, 2017.
- [13] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning." *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615-620, Oct. 2000.
- [14] Christopher M. Bishop. "Pattern Recognition and Machine Learning (Information Science and Statistics)." *Springer-Verlag*, Berlin, Heidelberg, 2006.
- [15] John J. Craig. "Introduction to Robotics: Mechanics and Control." Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989.
- [16] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo. "Robotics: Modelling, Planning and Control." *Springer Publishing Company*, Incorporated, 1st edition, 2008.
- [17] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43-53, February 1987.
- [18] S. Calinon, I. Sardellitti and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 249-254, Taipei, 2010.
- [19] P. Arena, L. Fortuna, G. Muscato, M.G. Xibilia. "Applications of quaternions in robotics," *Neural Networks in Multidimensional Domains*. *Lecture Notes in Control and Information Sciences*, vol 234. Springer, London, 1998.
- [20] S. Sarabandi and F. Thomas. "Accurate computation of quaternions from rotation matrices." *Advances in Robot Kinematics. Springer International Publishing.*, pp. 39-46, 2019.
- [21] J. Diebel. "Representing attitude: Euler angles, unit quaternions, and rotation vectors". *Matrix* 58, 2006.