Human posture prediction during physical human-robot interaction

Lorenzo Vianello, Jean-Baptiste Mouret, Eloïse Dalin, Alexis Aubry, Serena Ivaldi

HAL Id: hal-03115242

https://hal.science/hal-03115242v2

Submitted on 21 Jun 2021

Human Posture Prediction during Physical Human-Robot Interaction

Lorenzo Vianello[1,2], Jean-Baptiste Mouret[1], Eloise Dalin[1], Alexis Aubry[2], Serena Ivaldi[1]

Abstract— When a human is interacting physically with a robot to accomplish a task, his/her posture is inevitably influenced by the robot movement. Since the human is not controllable, an active robot imposing a collaborative trajectory should predict the most likely human posture. This prediction should consider individual differences and preferences of movement execution, and it is necessary to evaluate the impact of the robot's action from an ergonomics standpoint. Here, we propose a method to predict, in probabilistic terms, the human postures of an individual for a given robot trajectory executed in a collaborative scenario. We formalize the problem as the prediction of the human joints velocity given the current posture and robot end-effector velocity. The key idea of our approach is to learn the distribution of the null space of the Jacobian and the weights of the weighted pseudo-inverse from demonstrated human movements: both carry information about human postural preferences, to leverage redundancy and ensure that the predicted posture will be coherent with the end-effector position. We validate our approach in a simulated toy problem and on two real human-robot interaction experiments where a human is physically interacting with a Franka robot.

pHRI; Posture Prediction; Ergonomics.

I. INTRODUCTION

Cobots (i.e., industrial manipulators for collaboration) and exoskeletons are designed to physically interact with humans and to assist their movement in accomplishing one or more tasks [1]. The general objective is to reduce the human physical effort and improve his/her ergonomics, which requires the evaluation of several ergonomics criteria, most often determined by the human posture [2]. The way this assistance is provided depends on the platform and on the type of the collaboration, which often translates to defining contact points, collaboration control laws with structured roles (e.g., leader-follower) and the amount of provided assistance [3]. Recent works in collaborative robotics suggest that the robot's knowledge of the task to be executed could be used to plan movements that are less physical demanding [4] and more ergonomic [5]. This is particularly relevant in industrial scenarios because it has the potential to reduce work-related musculoskeletal disorders, which are currently the second largest cause of disabilities worldwide [6].
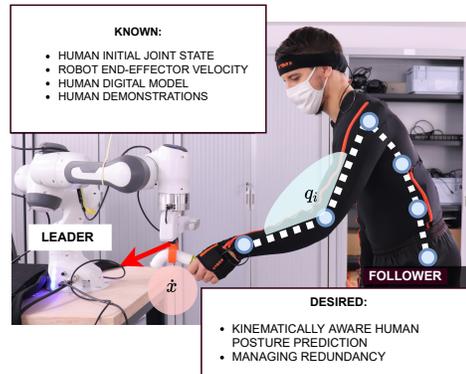
Fig. 1. The human posture is influenced by the robot's trajectory during physical interaction, but the human may adopt different postures during each task execution. In this paper we want the robot to predict the human posture given a known Cartesian trajectory of its end-effector and prior observations of the task executed by the human. The human posture is measured online by a wearable Xsens MVN suit.

An open problem, when a robot wants to assist the human, is that humans are not entirely "controllable": humans are highly redundant systems that are over-actuated for many manipulation tasks. For instance, lifting a box from the floor might be performed by bending the back, but also by bending the knees. Individual preferences of movement and musculo-skeletal problems might add to the intrinsic variability of the human movement, thus increasing the variance of all possible postures in response to a robot action. For these reasons, when the human is physically coupled with the robot to accomplish a task, it is not possible to know with certainty how a human will move when the robot imposes a trajectory, which makes it challenging to select the best trajectories for the robot in collaborative tasks. In this context, data-driven probabilistic models of human movements, learned from demonstrations, can provide interesting insights into human preferences while capturing the variance of demonstrated movements. A limit of this kind of solutions is that a small error in the joint estimation can cause a large error in the estimation of the end-effector position (i.e., the human hand), which makes the prediction kinematically inconsistent. This error poses a nontrivial problem, especially when the human is physically coupled to the robot because it can compromise the quality of the collaboration.

In this paper, we consider a leader robot that is physically coupled with the human follower at the level of the end-effector / hand (see Fig. 1). This asymmetric role assignment represents the case where a collaborative robots leads the human towards more ergonomic trajectories. It applies to both cobots (i.e., manipulators physically coupled to the human at their hands) and upper-body exoskeletons (physically

attached to the human body at the level of the hands and/or arms, depending on the application – maintenance, rehabilitation, etc.). For a known end-effector trajectory and an initial human posture, we want to determine the probability distribution of the human postures along the trajectory of the end-effector[1]. We record several demonstrations of the human interacting with the robot. We start by modeling the human with a Digital Human Model (DHM) which is a rigid body model, similar to a humanoid robot. We constrain the DHM and the robot's end-effectors to be physically linked.

Our main idea is to learn, from the human demonstrations, a model in the null space of the DHM Jacobian, which describes a set of human configurations that lead to the same end-effector position. For each point of the robot trajectory, our method consists of first projecting the postures in the null space defined by the end-effector position and the kinematic model of the human, then learning Gaussian processes (GP) that predict projected configurations, and finally projecting back to the original joint space. We use GP [7] for predictors because they make accurate and smooth predictions with little data compared to alternatives like neural networks. In addition, they associate each prediction with an estimate of its uncertainty.

We demonstrate the method on a toy problem first (a 5-revolute (5R) joints planar robot) and then on a human interacting with a robot manipulator (Franka) in a cooperative pick & place task. We show in the experiments that our method can predict the future movement "planned" by the human with high accuracy and kinematic consistency.

II. RELATED WORK

A. Human Posture Measurement and Inverse Kinematics

Collaborative robots need to have an estimation of the current human posture and its future intended evolution to plan appropriate collaborative actions. The human posture can be retrieved in real-time essentially using cameras or wearable motion tracking sensors [8]. However, robots often do not have access to the human posture measurement in real-time, and the only information they have is the fact that human is physically attached to their end-effector in some tasks. In such cases, Inverse Kinematics (IK) is used to predict the human pose starting from the end-effector position using simplified human models as in [5].
The problem is that the human posture is not uniquely defined by its end-effector position, because of the intrinsic human body redundancy but also task preferences and other individual factors; for a robot, it is hard to predict the human posture given only the task description. Indeed, IK alone cannot solve this problem given the overactuated nature of the human.

[1]There are many situations, even in manufacturing, where the initial posture of the operator can be known. It can be estimated by cameras or wearable sensors. In the case of ergonomic optimization (i.e., the offline procedure for optimizing the workstation and the robot's movements to maximize the ergonomics comfort of a group of operators), workers are often allowed to use wearable sensors to measure their movement. If the human posture cannot be measured, we can assume it known and fixed by the procedures for the ergonomic use of a robotic workstation.

$q \in \mathbb{R}^n$	vector of joints values
$x \in \mathbb{R}^m$	hand Cartesian pose
$k = n - m > 0$	degrees of redundancy
$f(.) : \mathbb{R}^n \to \mathbb{R}^m$	forward kinematics function
$\dot{q}$ and $\dot{x}$	joint and Cartesian velocities
$J(q) \in \mathbb{R}^{m \times n}$	Jacobian Matrix
$y \in \mathbb{R}^k$	null space of the Jacobian $J(q)$
$\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=1}^{N_D}$	Dataset of N_D demonstrations

TABLE I
NOTATION

To address this kind of problem, a common approach is to sample in the space of the possible solutions and to evaluate them accordingly to the kinematic properties [9] and to task-specific loss functions [10]. We refer to these kinds of methods as sampling based approaches. These kinds of methods are computationally expensive and highly dependent on the choice of the parameters. Moreover, they are not well designed to integrate human demonstrations that capture human preferences of movement.

B. Human Posture Prediction

Predicting the human intention, i.e., the future intended movement [11] is an active field of research, where traditionally movements are represented by trajectories or movement primitives issued with a probabilistic description. The prediction with motion primitives is most often done in the task space, e.g., the Cartesian space, and IK is used to find the most appropriate corresponding joint trajectories to fulfill the robot task. Motion primitives can also be learned in the joint space. However, each joint primitive cannot be learned independently as all the primitives must be kinematically consistent, and conditioning may not be sufficient to properly ensure this property [12]. In [13], this problem was solved using Probabilistic Motion Primitives (ProMPs), by conditioning the motion primitive in the joint-space with the one in the operational space. A limit of this approach is that it requires the knowledge of the ProMP for each possible movement, which only makes sense for specific applications and gestures.

Recurrent neural networks have also been proposed for predicting future human postures [14], [15]. One of the main challenges of these methods is to encode the multi-value behaviour of the human, coming by its redundant structure, and to evaluate the different solutions [16]. Data-driven methods are, in general, time efficient and they do not require hard coded evaluation functions because they learn directly from demonstrations. The main limit of these algorithms is the loss of the kinematic consistency in the prediction: it was demonstrated that applying regression for mapping from task space to joint space using standard regression can lead to inconsistent predictions [17]. In [18] the human pose is predicted using learned models (e.g. nearest neighbour); but then an IK correction is required to fix the hand positions to match the kinematic constraints imposed by the collaborative robot. Also, authors reported it was too computationally expensive to be used for online planning.

C. Digital Human Models for Ergonomics

A collaborative robot can be used to assist the human worker and improve ergonomics at work [19]. Ergonomics scores typically rely on kinematics and dynamics information about the human's movement, which are often extracted from simulations of Digital Human Models (DHMs). There are two main types of DHMs: the first are musculo-skeletal models, which are rather complex, have many degrees of freedom, and allow the analysis of the human movement by simulating the muscular efforts [20]; the second are rigid body models, which are simplified models with less degrees of freedom, where the human is basically represented as a humanoid robot made of rigid body links [19]. Such a DHM can be used to reproduce a variety of motions demonstrated by humans operators and captured via motion tracking devices [21]. While the first ones are rather complex and expensive in terms of computational resources (it can take several minutes to simulate a small movement), the second ones are simpler but faster to simulate. As such, they are better suited for real-time applications such as model-based prediction, control and ergonomics assessment [19]. Several ergonomics scores exist (e.g., RULA, REBA), and they are primarily based on postural information [22].

III. PROBLEM FORMULATION

Notation: In our study, the human is represented by a DHM, a rigid body model with n degrees of freedom. The notation presented in Table I is used for the DHM. The robot state is determined by $x_R, \dot{x}_R$, i.e., the Cartesian position and velocity of its end-effector (EE).

We consider a cooperating human-robot interaction scenario, where human and a robot manipulator interact to perform a joint task. The robot's task trajectory at the EE is known at each time step[2]: $\{\dot{x}_R(t)\}_{t=0}^{T-1}$. The two agents are physically coupled at their EEs; the robot is leading (leader role), while the human (follower role) is guided by the robot; hence, we assume $\dot{x} = \dot{x}_R$. We assume a rigid, constant roto-translation between the two frames. Given the current human joint configuration q (known, we suppose its measure is accessible to the robot) and the robot EE velocity $\dot{x}_R$, we want to predict the human joint velocity $\dot{q}$. Since the human is over-actuated, we want to predict a distribution of solutions that capture the "preference" of human movement (i.e., analogously to the concept of most likely solutions [10]); such solutions must be kinematically feasible, i.e., they must verify that $\dot{x} = J(q)\dot{q}$. The problem can be formalized as computing the conditional probability:

$$p(\dot{q}|q, \dot{x}) \quad \text{s.t.} \quad \dot{x} = J(q)\dot{q}, \tag{1}$$

where the second term is the kinematic constraint which determines the set of possible solutions.

IV. BACKGROUND

A. Kinematics for redundant DHMs

A redundant[3] DHM is a DHM that has more degrees of freedom than the number nominally required to perform a given set of tasks ($n > m$). Redundancy yields increased dexterity and versatility for performing a task due to the infinite number (∞^{n-m}) of joint motions which produce the same EE motion. Given an EE pose $x \in \mathbb{R}^m$, the space which contains all the solutions of the IK equation $\{q : x = f(q)\}$ is defined as the inverse kinematics's manifold $\mathcal{M}_x$. It is considered a union of more simple and continuous manifolds, called "self-motion manifold" ($\mathcal{M}_s$) [23]. Any change of joint configuration ($\dot{q}_s$) along a self-motion manifold is called "self-motions" and it does not change the position of the EE ($J(q)\dot{q}_s = 0$). The space containing these joint velocities is the null-space of the Jacobian matrix evaluated in q, which is the set of vectors $\dot{q}_s$ which satisfy $J(q)\dot{q}_s = 0$ and with $\dot{q}_s \neq 0$. A basis for the null space of $J(q)$ is composed of the columns of the matrix $V_N = [\mu_1, ..., \mu_{n-m}]$; this matrix can be obtained by singular value decomposition:

$$J = USV^\top = U(S_R \quad 0)\begin{pmatrix} V_R^\top \\ V_N^\top \end{pmatrix} \tag{2}$$

where V_R and V_N are the range and the null-space components, respectively [17]. Thus, each self-motion velocity could be represented by a linear combination of the columns of V_N: $\dot{q}_s = V_N(q)y$, where $y \in \mathbb{R}^k$ is the vector of the coefficients of the linear combination. This consideration is particularly useful for interpreting local redundancy resolution technique: each movement in the joint state can be seen as the sum of the minimal velocity needed to match $\dot{x}$ plus a movement in the joint space which has no effect in the workspace. In the literature, this approach is usually referred to as the dual projection method: the joint velocity is computed as $\dot{q} = J_W^\dagger(q)\dot{x} + [I - J_W^\dagger(q)J(q)]z(q)$, with the weighted pseudo-inverse $J_W^\dagger(q) = WJ^\top(q)[J(q)WJ^\top(q)]^{-1}$ that instantaneously minimizes the symmetric weighted quadratic form $\dot{q}^\top W^{-1}\dot{q}$, and $z(q) \in \mathbb{R}^n$ is a joint velocity projected onto the null space of the manipulator Jacobian and thus on the tangent space of the self-motion manifold. Typically, $z(q)$ is designed as a potential function that minimizes a desired cost function $\mathcal{C}(.)$ [24][25].

B. Gaussian Processes

A Gaussian Process (GP) [7] is a collection of random variables such that any finite collection has a joint Gaussian distribution. In regression the random variables represent the value of the function $f(x) \in \mathcal{Y}$ for the given input $x \in \mathcal{X}$.

A GP, denoted by $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$, is entirely characterized by the mean $m(x) = E[f(x)]$ and covariance $k(x, x') = E[(f(x) - m(x))(f(x) - m(x))]^\top$, which is symmetric and positive semi-definite.

Let $\mathcal{D} = \{(x_i, y_i)|x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be a training set and (x_*, y_*) a point we did not observe in $\mathcal{D}$. The GP predictive

---

[2]In the following, we drop the time dependence t in the equations, unless necessary, to improve the readability of the equations.

[3]The reader may notice that "redundant DHM" is equivalent to "redundant robot", since the DHM is modeled essentially as a robot with rigid bodies.
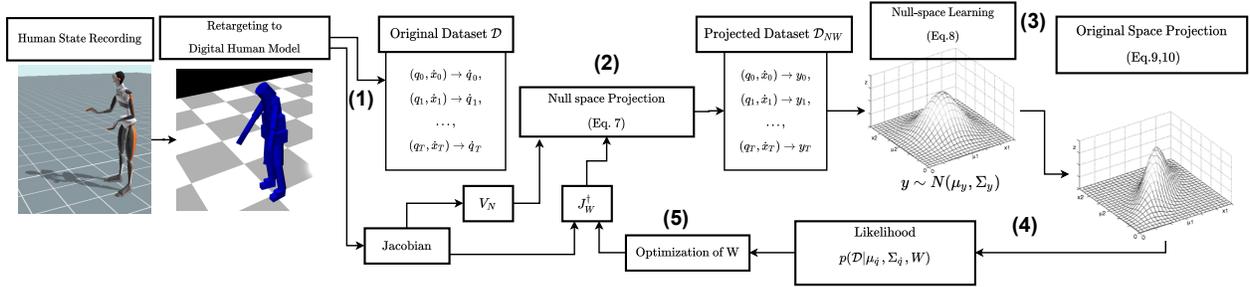
Fig. 2. Flowchart of the offline training: (1) We collect human movements using a motion capture suit. The joint states are passed to a digital human model and they are used to calculate the Jacobian at each joint configuration. From the digital human model we record also the dataset $\mathcal{D}$. (2) We project the joint velocities $\dot{q}$ on the null space of the Jacobian; at the first iteration of the algorithm the matrix W used for the pseudo-inverse is an identity matrix. (3) The projected dataset is used to train k independent GP. (4) We invert the projection to obtain a distribution over $\dot{q}$ and we calculate the likelihood; (5) We optimize the W matrix accordingly to the likelihood using a gradient free optimizer and we repeat from point (2).
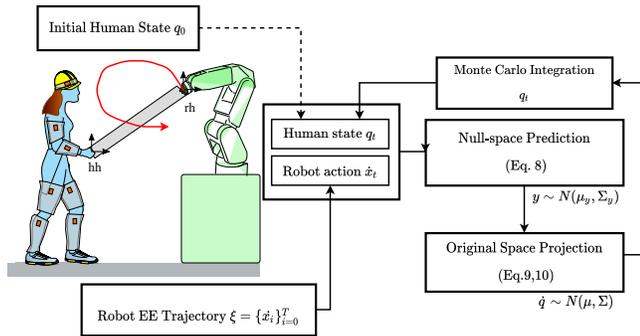

Fig. 3. Flowchart of the online prediction: Given an EE trajectory imposed by the robot and the knowledge of the initial human state we can predict a distribution over the future human states. To do that we sample on the human joint velocity $\dot{q}$ calculated with our method (MI-NsGP) then we integrate the current human state, in this way we could propagate the uncertainty to the next human state. We repeat this procedure throughout the EE trajectory, in this way we create a probabilistic estimation of the human joint trajectory (Monte Carlo rollout).

distribution for the output y_* at the test input x_*, given in vector form, is

$$p(y_*|\mathcal{D}, x_*) = \mathcal{N}(\mu_*, \Sigma_*),$$
$$\mu_* = k_*^\top (k + K_{err})^{-1} y,$$
$$\Sigma_* = k_{**} - k_*^\top (k + K_{err})^{-1} k_*$$

where, given a kernel function $k(.,.) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ we use the notation $k = k(x,x), k_* = k(x,x_*), k_{**} = k(x_*,x_*)$ and K_{err} is the measurement error variance. In this work, we use the radial basis function (RBF) kernel: $k_{\sigma^2,\lambda}(x,y) = \sigma^2 exp(-\frac{||x-y||^2}{2\lambda}), \lambda > 0$, where, the parameters (σ^2, λ) are chosen by maximizing the marginal likelihood $P(y|(\sigma^2, \lambda))$.

V. METHOD

We consider a DHM with n degrees of freedom. We assume that the human/DHM follows this classic control law from robotics (section IV-A, [24], [25]):

$$\dot{q} = J_W^\dagger(q)\dot{x} + (I - J_W^\dagger(q)J(q))z(q) \qquad (3)$$

where $z(q)$ is an unknown vector of null-space velocities, often it is calculated as the gradient of a cost function $\mathcal{C}(.)$.

The weights W of the weighted pseudo-inverse $J_W^\dagger$ are also unknown. The EE velocity $\dot{x}$ is known. Our objective is to learn $z(q)$ and W from data. In this way, the solutions we find must always satisfy the kinematic constraint: $\dot{x} = J(q)\dot{q}$.

A. Learning the null-space velocity with Gaussian processes

Let us consider a dataset $\mathcal{D}$ of DHM motions, composed of N_D pairs: a tuple with the current joint state $q \in \mathbb{R}^n$ and the EE velocity $\dot{x} \in \mathbb{R}^m$, and the joint velocity $\dot{q} \in \mathbb{R}^n$: $\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=1}^{N_D}$. Such a dataset can be generally acquired via human motion tracking (see (1) in Fig. 2).

At this stage, we consider the values of the weight matrix W of Eq. 3 to be known, for example $W = I$, where I is the identity matrix (see section V-B for learning W).

Instead of learning directly the value of $z(q)$, we notice that we can write [26]:

$$V_N(q)y = (I - J_W^\dagger(q)J(q))z(q) \qquad (4)$$

where $V_N(q)$ is the basis of the null space of the Jacobian $J(q)$, which is computed using Singular Value Decomposition (SVD), and $y \in \mathbb{R}^k$ represents the coordinates of the self-motion joint velocity in the null space.

We learn y instead of $z(q)$ because it is the minimal size representation for a self-motion joint velocity (see (2) in Fig. 2). We therefore project every joint velocity $\dot{q}$ of our dataset to the null space of the Jacobian $J(q)$ evaluated in the current joint configuration q by applying:

$$y(q, \dot{x}) = V_N^\dagger(q)\Big(\dot{q} - J_W^\dagger(q)\dot{x}\Big) \qquad (5)$$

Thus, given the dataset $\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=1}^{N_D}$, we apply Eq. 5 to obtain $\mathcal{D}_{NW} = \{(q_i, \dot{x}_i), y_i\}_{i=1}^{N_D}$.

We learn $y(q, \dot{x})$ using GPs (sec. IV-B) that map the current joint state and the EE velocity to the joint velocity:

$$y|(q, \dot{x}) \sim \mathcal{GP}\Big(m(q, \dot{x}), k((q, \dot{x})_i, (q, \dot{x})_j)\Big) \qquad (6)$$

After experimental testing and, following [27], we train k independent GP, one for each dimension of y. We choose this approach to keep the method simple, but we expect that the performance of our method could be improved with the

right choice of multidimensional GP and we will investigate this improvement in future work. Since Eq. 5 is linear, given the Gaussian distribution $p(y) \sim \mathcal{N}(\mu_y, \Sigma_y)$, we can get the Gaussian distribution of $p(\dot{q}) \sim \mathcal{N}(\mu_{\dot{q}}, \Sigma_{\dot{q}})$ by inverting it:

$$\mu_{\dot{q}}(q, \dot{x}) = J_W^\dagger(q)\dot{x} + V_N(q)\mu_y \tag{7}$$

$$\Sigma_{\dot{q}}(q, \dot{x}) = J_W^\dagger \Sigma_{\dot{x}}(J_W^\dagger)^\top + V_N(q)\Sigma_y V_N(q)^\top \tag{8}$$

where $\Sigma_{\dot{x}}$ is the covariance matrix of the noise of $\dot{x}$ learned-from the data (see (4) in Fig. 2).

B. Learning the parameters W

We want to find the values of W that maximizes the likelihood of the $\dot{q}$ of the training set (see (5) in Fig. 2). To do so, we introduce a score function $S(W)$ that is maximized with a non-linear optimizer: $S(W) = \frac{1}{N_D} \sum_{i=1}^{N_D} \mathcal{L}(\dot{q}_i | W)$, where $\mathcal{L}(\dot{q}_i | W)$ is the likelihood of $\dot{q}_i$ given a particular value of W and N_D is the size of the training set. For a given W and $\dot{q}$, $\mathcal{L}(\dot{q} | W)$ can be computed using $\mu_{\dot{q}}$ and $\Sigma_{\dot{q}}$ from Eq. 8 (since $\mu_{\dot{q}}$ and $\Sigma_{\dot{q}}$ define a multivariate Gaussian distribution and we know $\dot{x}$ and q from the training set):

$$\mathcal{L}(W|\dot{q}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_{\dot{q}}|}} \exp\left(-\frac{1}{2}(\dot{q} - \mu_{\dot{q}})^\top \Sigma_{\dot{q}}^{-1}(\dot{q} - \mu_{\dot{q}})\right)$$

where $|\Sigma_{\dot{q}}|$ denotes the determinant of $\Sigma_{\dot{q}}$, $\mu_{\dot{q}} = \mu_{\dot{q}}(q, \dot{x})$, and $\Sigma_{\dot{q}} = \Sigma_{\dot{q}}(q, \dot{x})$.

Any non-linear optimizer can be used to maximize $S(W)$. For simplicity and robustness, we used BIPOP-CMA-ES [28], which is a gradient-free stochastic optimizer available in the "pycma" Python library[4]. In each iteration of the algorithm new candidate solutions of W are generated by variation. Then, some solutions are selected to become the parents in the next generation based on the score function $S(W)$ evaluated after learning $y(q, \dot{x})$ using GPs as explained in the previous section.

C. Prediction Phase

Once the model is trained, it can be used to predict the human joints' trajectories given the current configuration q_t and the expected EE trajectory executed by the robot $\{x_1^d, \ldots, x_T^d\}$. At each time step we can sample the EE velocity as: $\dot{x}_t \sim \frac{1}{\Delta_t}\left(x_{t+1}^d - f(q_t) + \mathcal{N}(0, \Sigma_{\dot{x}})\right)$, where Δ_t is the distance between two time-steps, $\Sigma_{\dot{x}}$ is the robot repeatability when executing a trajectory (which we estimated empirically by executing a desired trajectory 10 times).

At each time-step, given the current configuration q_t, we can get $\mu_{\dot{q}}(q_t, \dot{x}_t)$ and $\Sigma_{\dot{q}}(q_t, \dot{x}_t)$ by querying the model (Eq. 8). From this multivariate Gaussian distribution, we can sample $\dot{q}_t$, which allows us to compute the value of $q_{t+1} \sim q_t + \Delta_t \, \mathcal{N}(\mu_{\dot{q}}(q_t, \dot{x}_t), \sigma_{\dot{q}}(q_t, \dot{x}_t))$. To sample a whole trajectory, we repeat this procedure by propagating the sampling over time from $t = 0$ to $T - 1$. If we repeat this sampling procedure many times for a given trajectory, we get a Monte-Carlo estimation of the distribution over the human joint trajectories according to our model [29]. A schema of the prediction phase is depicted in Fig. 3.

VI. EXPERIMENTS

To evaluate our method (denoted as MI-NsGP), we compare it experimentally to alternative approaches that use only a subsets of our elements (i.e., we make several ablation experiments):

1) **MI-NsGP**: Null-Space Gaussian Process with weight identification: our method, which learns both W and $y(q, \dot{x})$ (Sec. V): $\dot{q}|(q, \dot{x}) \sim J_W^\dagger(q)\dot{x} + V_N(q)\mathcal{GP}(q, \dot{x})$
2) **GP**: learning directly from data: $\dot{q}|(q, \dot{x}) \sim \mathcal{GP}(q, \dot{x})$.
3) **W-IK**: learning W but not $y(q, \dot{x})$ (i.e., $y(q, \dot{x}) = 0$) $\dot{q} = J_W^\dagger(q)\dot{x}$
4) **NsGP**: learning $y(q, \dot{x})$ but not W (i.e., $W = I$) $\dot{q}|(q, \dot{x}) \sim J_I^\dagger(q)\dot{x} + V_N(q)\mathcal{GP}(q, \dot{x})$
5) **Sb-M**: fitting a normal distribution $\mathcal{N}(\mu_y, \Sigma_y)$ on the training set for $y(q, \dot{x})$ and not learning W: $\dot{q}|(q, \dot{x}) \sim J_I^\dagger(q)\dot{x} + V_N(q)\mathcal{N}(\mu_y, \Sigma_y)$

where $\mathcal{GP}(q, \dot{x})$ denotes the distribution that corresponds to the GP model learned from data. The same training set and test set was used for all the methods.

Moreover we compared our method with a state-of-the-art method for predicting joint trajectory while satisfying a task space motion primitive:

6) **ProMP** [13]: We fit both the joint space and operational space using probabilistic movement primitive. And then we use Bayesian task prioritization to condition the joint space using the operational space.

All methods were evaluated on three experiments. The first (**5R**) consists of predicting the joint state of a simulated 5R planar robot controlled by a biased IK function. The second (**EXP1**) and third (**EXP2**) consist in predicting the human posture (i.e., joints) during a co-manipulation trajectory, where a human is physically attached to the Franka robot to do a task.

A. Toy problem: 5R Manipulator

We simulate an overactuated planar robot with 5 degrees of freedom. Like a human, this 5R planar robot is overactuated for the two-dimensional position of its EE.

The robot controller is conceptually similar to the (unknown) human controller (Eq. 3), except that the ground truth is known $(W, z(q))$. The 5R robot is controlled using the control law from Eq. 3, with:

$$z(q) = \frac{\nabla \mathcal{C}(q)}{\nabla q} + \mathcal{N}(0, \sigma_z) \tag{9}$$

To define a $z(q)$ similar to the human model, we hypothesized, as in [10], that the joint velocity minimizes an ergonomic cost function $\mathcal{C}(q)$ that depends on the joint configuration. We designed a cost function similar to the RULA continuous ergonomic score [22]. To define W, we assumed that some joints have more contribution than others (for example, in humans, the shoulders and elbows are typically more involved than lumbar's joints, but any musculoskeletal disorder can change this distribution drastically). To model these situations, we choose a weight matrix W that has non-uniform values (e.g., a low value for the first joint
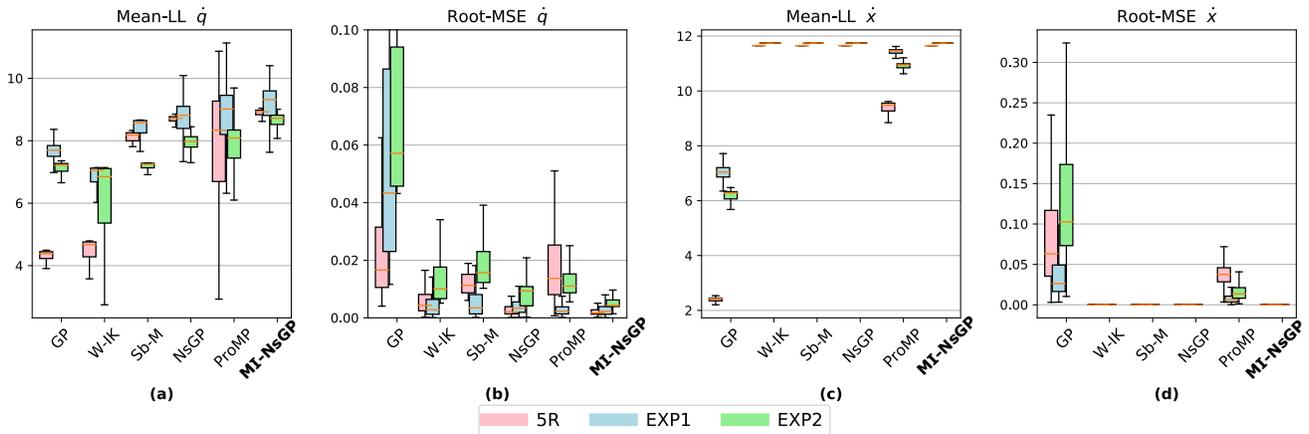
Fig. 4. Comparison of methods for joint velocity prediction: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the EE velocity (d) R-MSE over the EE velocity. The methods were evaluated on three experiments: (red) simulated 5R planar robot controlled by a biased IK function; (blue) human posture prediction during a human-robot collaboration task; (green) human posture prediction during a human-robot collaboration task using different tasks in the training-set and in the test-set.

means that it is not used much). Specifically, we selected a diagonal and positive definite matrix with values bounded in $[0 + \epsilon, 2 - \epsilon]$. We choose to bound the values otherwise we risk falling into a singular configuration in which a joint never moves or always moves, which appears far from a human-like behavior. Starting from a configuration q_0, we apply N_D times the control law specified in Eq. 3 and 9, with a random EE velocity $\dot{x}_i \in [-u_{max}, u_{max}]$. The successive joint state is then updated as $q_t = q_{t-1} + \dot{q}_{t-1}\Delta_t + \omega_a \Delta_t^2$, where $\omega_a \sim \mathcal{N}(0, \Sigma_a)$ is a Gaussian noise. If the robot falls in a singular configuration, the data collection stops and restarts from the q_0 configuration. At each time step t, we collected $\{(q, \dot{x}), \dot{q}\}_t$ to create the training set $\mathcal{D}$. The dataset, composed of $N_D = 10^3$ points, was normalized and divided into a training and a validation set following the proportion 70/30. We trained the models using the training set. Each GP was implemented in Python using *gpytorch* library, with a constant mean and the RBF kernel. The optimization was done on a Intel Core^{TM} i7-8850H with 6 cores at 2.6GHz, requiring about 10 hours. We repeated the experiment 10 times varying the starting point and the parameters of the control model (W).

Results: We first analyze the quality of the predicted distribution by computing the mean log-likelihood over the test set (red box-plots in Fig. 4a). Overall, our method (MI-NsGP) leads to significantly better likelihood values than all the other control approaches. The worst likelihoods are obtained by the methods that do not use the null space. Among the methods that use the null space, learning W makes a significant difference. The low likelihood for ProMP is due to the lack of a primitive for the movements, which results in a large variance for the solution obtained using ProMP. We then focused on the mean prediction by computing the root mean square error on $\dot{q}$ (red box-plots in Fig. 4b) (we ignored the variance). As before, the best results are obtained with our method, and using the null space makes a significant difference. However, learning a simple Gaussian model instead of a GP leads to very bad mean square errors

whereas it corresponds to high likelihood values (red box-plots in Fig. 4a). This means that this method has a very large variance, which makes the test set likely (high likelihood score) but the predictions very inaccurate. Also in this case, the solutions found using the ProMP are inaccurate.

Last, we computed the mean log-likelihood and the root-MSE for the EE position (red boxplots in Fig. 4c and Fig. 4d). As expected, perfect scores are obtained with the methods that exploit the null space (W-IK, Sb-M, NsGP, MI-NsGP), but learning directly a GP that predicts $\dot{q}$ directly leads to significant errors in the EE position. The solutions obtained using ProMPs have lower score since the task space has large variance trying to fit a ProMP to trajectories that are not related to a movement primitive. These results suggest that if the human's IK model is similar to the one we used for the 5R robot, our method is likely to improve the quality of posture prediction while satisfying the kinematic constraint.

B. Human IK prediction

We then evaluated our method in two experiments (EXP1 and EXP2) where a human is interacting with the Franka Emika Panda robot. The human is facing the robot, his right hand is in physical contact with the robot's EE (see Fig. 1 and video attachment). The Xsens MVN suit is used to capture the human posture (and to have the ground truth of the posture prediction). The human poses are fitted (retargeted) to a DHM of 66 segments (Fig. 5a), based on the Xsens MVN model. The segments are scaled with the human height, while the dynamic properties (e.g., mass) are computed from anthropometric data available in literature [19]. We modeled the human spherical joints collected by the motion capture suit as a series of 3 one-dimensional revolute joints, where each DoF is controlled by a single actuator. The resulting DHM posture is represented by the 66 joints. A URDF (Universal Robot Description Format) model is then created to represent the kinematics and dynamics of the DHM, and used by the Pinocchio library [30] to calculate the Jacobian going from the human pelvis to the right hand

for a given human joint configuration.

In this paper, we predicted only the joints that belong to the active kinematic chain, i.e., joints that connect segments from the pelvis to the right hand. The reason is twofold: first, a simpler model speeds up the computation; second, it is the set of joints that are used to compute ergonomics score (e.g., RULA, back angle), which is our final objective. The human posture is thus characterized by 24 revolute joints and the resulting dataset $\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=0}^{N_D}$ contains $q \in \mathbb{R}^{24}$, i.e., the joints which link the human pelvis to the right hand, and $x \in \mathbb{R}^6$, i.e., the EE position and orientation.

We evaluated our method on two experiments: In EXP1 the training set used to train the algorithm consists of postures recorded during the repeated execution of similar co-manipulations. The robot executes four "pick and place" trajectories spanning 50cm, and its orientation is maintained constant. Each trajectory is repeated 10 times: during the first experiment (EXP1) the first five trajectories comprise the training set, and the five remaining ones, the test set. Fig. 6a shows the intrinsic variability of the human repetitions (for the same EE movement, the joint trajectories change). In EXP2 the training set consists of pseudo-random trajectories that do not necessarily refer to motion primitives. With this second experiment we tested the ability of our method to generalize to new movement primitives. To do that we collected a not pre-defined training set of "pick and place" movements controlling the robot using a Joystick. We trained our algorithm on this dataset and we tested on the same test-set as EXP1. In the prediction phase we sampled 10 trajectories using the Monte-Carlo approach and for each of them we calculated four different ergonomics scores from the state of the art in human ergonomics [19]: RULA, REBA, RULA continuous and cumulative back angle (Fig. 5b). The purpose is to show that the probabilistic IK also impacts the prediction of ergonomics scores, which is critical information for a collaborative robot.

Results: We evaluate the performance of our method on predicting the human posture when both in the training set and the test set are considered the same movement primitive (EXP1). We analyze the quality of the predicted distribution by computing the mean log-likelihood over the test set (blue in Fig. 4a). Overall, our method (MI-NsGP) leads to significantly better likelihood values than all the control approaches. Moreover its performance is comparable with the state-of-the-art method for human posture prediction (ProMP). In fact MI-NsGP performs better in the median and max value of the 95th percentile (MI-NsGP: $9.32[8.81, 9.59]$, ProMP: $9.02[8.20, 9.46]$). Regarding the the root-mean-square error on $\dot{q}$ (Fig. 4b), our method is comparable with the SoA method (MI-NsGP: $(2.19[1.24, 3.38]) \times 10^{-3}$, ProMP: $(2.30[1.37, 3.83]) \times 10^{-3}$) and presents better results with respect to the other model-based methods (W-IK, Sb-M, NsGP) and with respect to using the GP in the original space (Fig. 4b). Regarding the ability to satisfy the kinematic constraint, we observed a behavior similar to the toy-problem. In fact, model based methods (W-IK, Sb-M, NsGP, MI-NsGP) always have bigger
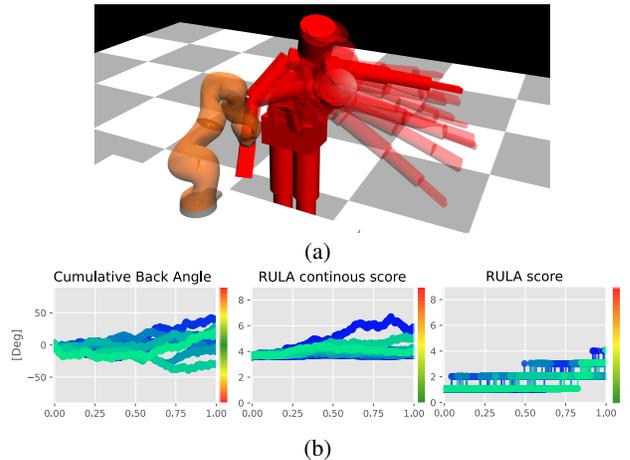

Fig. 5. (a) The DHM in Simulation, showing the variance of the solutions calculated via Monte-Carlo integration. (b) Ergonomic scores computed on different sampled trajectories: RULA, REBA, RULA continuous, cumulative back angle.

likelihood and smaller root-mean-square error with respect to GP regression. ProMP results are also accurate in this case because the training set and the test set belong to the same movement primitive. The superiority of model based methods is even more evident at trajectory level: if we use the GP alone to predict the DHM postures while the prediction horizon is growing, the R-MSE between the EE of the DHM and the robot's EE (the red progression in Fig. 6b) grows too fast to be used in a safe human-robot collaboration scenario while if we use MI-NsGP (the green progression in Fig. 6b) the error is acceptable. In the case of the human, W is unknown; thus, it is not straightforward to evaluate the resulting values from model identification. Nonetheless some considerations are possible: even considering different training-sets, the optimization converges to the same values of W; these values agree with our expectations regarding the distribution of the joint velocity. In fact, the joints which move less (e.g. lumbar joints) have a smaller value with respect to those which are more involved in the execution of the movement (e.g. shoulder and elbow). In EXP2 we evaluate the ability of our method to generalize the information learned for one trajectory to another. The results show that our method outperforms the others both the likelihood (MI-NsGP: $8.72[8.52, 8.81]$, ProMP: $8.0[7.44, 8.34]$) and the root-mean-square error (MI-NsGP: $(4.34[3.93, 6.22]) \times 10^{-3}$, ProMP: $(11.04[8.70, 15.25]) \times 10^{-3}$) while continuing to satisfy the kinematic constraint (green in Fig. 4). The results suggest that our method could be used to have a probabilistic estimation of the human posture also for trajectories which do not share the same movement primitive.

VII. CONCLUSIONS

We presented a method for predicting human posture in a Human-Robot Collaboration scenario where the human hand motion is constrained by the robot's end-effector. We propose a two-phase method: in the first phase, we leverage a dataset of human demonstrations to learn a distribution over the null-space of the human Jacobian using a Gaussian Process; in the second phase we optimize the weights of
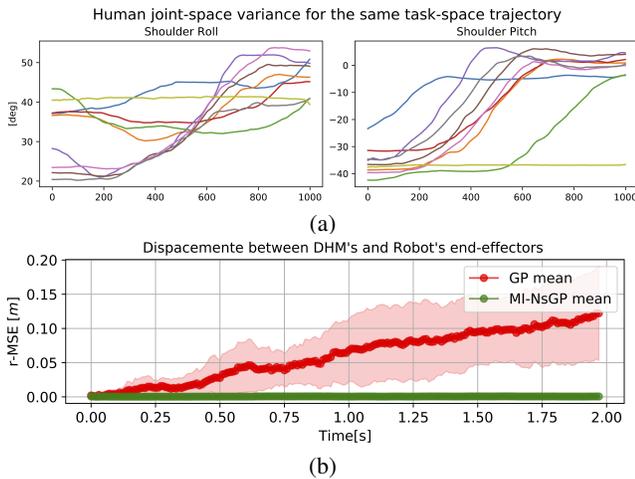
Fig. 6. (a) Human joint trajectories (shoulder roll and pitch) in response to the same EE movements. (b) MSE in offline prediction with GP and MI-NsGP.

the weighted pseudo-inverse of the Jacobian. Our method computes a probabilistic estimation of the future postures that satisfy the kinematic constraints imposed by the physical link between the human and the robot, and at the same time is coherent with the human preferences of movement.

In the future, we want to consider the full human model in the posture prediction and integrate the algorithm into our framework for ergonomics control, which aims to optimize a collaborative robot's motions to maximize the comfort and the ergonomics of the human collaborator. A byproduct of our method is the probabilistic computation of ergonomics scores for a given robot's EE trajectory, which is a critical element for planning the robot's trajectories. Further, we want to remove the leader/follower hypothesis, and address the case where the leadership role may vary over time.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[2] A. Malaisé, P. Maurice, F. Colas, and S. Ivaldi, "Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1132–1139, 2019.

[3] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1656–1674, 2012.

[4] L. Peternel, C. Fang, N. Tsagarakis, and A. Ajoudani, "A selective muscle fatigue management approach to ergonomic human-robot co-manipulation," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 69–79, 2019.

[5] A. Shafti, A. Ataka, B. U. Lazpita, A. Shiva, H. A. Wurdemann, and K. Althoefer, "Real-time robot-assisted ergonomics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[6] T. Vos, R. M. Barber, B. Bell *et al.*, "Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–analysis for the global burden of disease study 2013," *The Lancet*, vol. 386, no. 9995, pp. 743–800, 2015.

[7] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[8] P. Maurice, A. Malaise, C. Amiot, N. Paris, G.-J. Richard, O. Rochel, and S. Ivaldi, "Human movement and ergonomics: an industry-oriented dataset for collaborative robotics," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1529–1537, 2019.

[9] A. Yazdani, R. S. Novin, A. Merryweather, and T. Hermans, "Estimating human teleoperator posture using only a haptic-input device," *arXiv preprint arXiv:2002.10586*, 2020.

[10] R. Rahal, G. Matarese, M. Gabiccini, A. Artoni, D. Prattichizzo, P. R. Giordano, and C. Pacchierotti, "Caring about the human operator: haptic shared control for enhanced user comfort in robotic telemanipulation," *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 197–203, 2020.

[11] O. Dermy, A. Paraschos, M. Ewerton, F. Charpillet, J. Peters, and S. Ivaldi, "Prediction of intention during interaction with iCub with probabilistic movement primitives," *Frontiers in Robotics and AI*, vol. 4, p. 45, 2017.

[12] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[13] A. Paraschos, R. Lioutikov, J. Peters, and G. Neumann, "Probabilistic prioritization of movement primitives," *IEEE Robotics and Automation Letters*, 2017.

[14] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao, "Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly," *CIRP Annals*, 2020.

[15] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of human full-body movements with motion optimization and recurrent neural networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1792–1798.

[16] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, "Human motion prediction using semi-adaptable neural networks," in *American Control Conference (ACC)*, 2019, pp. 4884–4890.

[17] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *International Conference on Robotics and Automation (ICRA)*, 2000.

[18] L. F. van der Spaa, T. Bates, M. Gienger, and J. Kober, "Predicting and optimizing ergonomics in physical human-robot cooperation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[19] P. Maurice, V. Padois, Y. Measson, and P. Bidaud, "Digital human modeling for collaborative robotics," in *DHM and Posturography*. Elsevier, 2019, pp. 771–779.

[20] T. Bassani, E. Stucovitz, Z. Qian, M. Briguglio, and F. Galbusera, "Validation of the anybody full body musculoskeletal model in computing lumbar spine loads at l4l5 level," *Journal of Biomechanics*, vol. 58, pp. 89 – 96, 2017.

[21] P. Maurice, "Virtual ergonomics for the design of collaborative robots," Ph.D. dissertation, 2015.

[22] B. Busch, G. Maeda, Y. Mollard, M. Demangeat, and M. Lopes, "Postural optimization for an ergonomic human-robot interaction," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2778–2785.

[23] J. W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds," in *Advanced Robotics*. Springer, 1989, pp. 25–34.

[24] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, 2015.

[25] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of intelligent and robotic systems*, vol. 3, no. 3, pp. 201–212, 1990.

[26] M. Z. Huang and H. Varma, "Optimal rate allocation in kinematically-redundant manipulators-the dual projection method," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

[27] K. Chatzilygeroudis and J.-B. Mouret, "Using parameterized black-box priors to scale up model-based policy search for robotics," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[28] N. Hansen, "Benchmarking a bi-population CMA-ES on the BBOB-2009 function testbed," in *11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (GECCO): Late Breaking Papers*, 2009, pp. 2389–2396.

[29] K. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades, and J.-B. Mouret, "Black-box data-efficient policy search for robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 51–58.

[30] J. Carpentier, F. Valenza, N. Mansard *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2019.

Human Posture Prediction during Physical Human-Robot Interaction
Supplementary Material

Lorenzo Vianello[1,2], Jean-Baptiste Mouret[1], Eloise Dalin[1], Alexis Aubry[2], Serena Ivaldi[1(*)] [‡]

1 Toy problem: 5R Manipulator

To define $z(q)$ similar to the human model, we hypothesized, as in [1], that the joint velocity minimizes an ergonomic cost function $\mathcal{C}(q)$ that depends on the joint configuration. We designed a cost function similar to the RULA continuous ergonomic score [2]. It is composed as the sum of a second order polynomial: $\mathcal{C}(q) = \sum_{j=1}^{n}(p_{2,j}q_j^2 + p_{1,j}q_j + p_{3,j})$ where (p_1, p_2, p_3) have been calculated by fitting a second degree polynomial within the RULA score.

$(p_1, p_2, p_3)_{j=0}$	$(1.1 \times 10^{-03}, 0.0, 0.0)$
$(p_1, p_2, p_3)_{j=1}$	$(9.8 \times 10^{-04}, 0.0, 1.0)$
$(p_1, p_2, p_3)_{j=2}$	$(1.6 \times 10^{-04}, -2.5 \times 10^{-02}, 2.0)$
$(p_1, p_2, p_3)_{j=3}$	$(1.2 \times 10^{-04}, 0.0, 0.0)$
$(p_1, p_2, p_3)_{j=4}$	$(2.1 \times 10^{-03}, 0.0, 1.0)$

Table 1: Values of the parameters of the cost function $\mathcal{C}(q)$ similar to the RULA continuous ergonomic score.
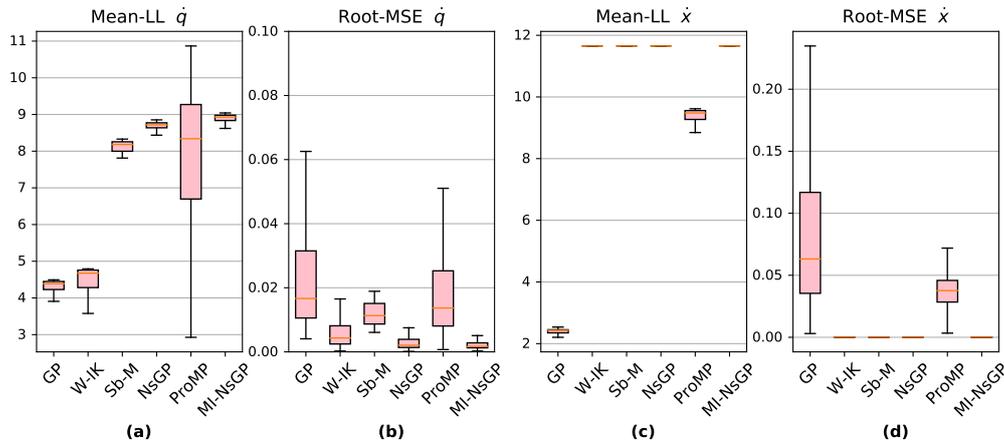

Figure 1: Comparison of methods for joint velocity prediction for 5R manipulator: our method (MI-NsGP), learning directly from data using Gaussian Process (GP), learning W and apply pseudo-inverse (W-IK), learning null space (NsGP), motion primitives based method (ProMP), sampling in the null space (Sb-M). The criteria: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the end-effector's velocity obtained applying the methods (d) R-MSE of the end-effector's velocity.

	M-LL $\dot{q}$	R-MSE $\dot{q}$	M-LL $\dot{x}$	R-MSE $\dot{x}$
GP	4.39 [4.23, 4.45]	$(16.98\ [\ 10.34,\ 30.90])\times 10^{-3}$	2.43 [2.35, 2.45]	$(6.31\ [\ 3.54,\ 11.68])\times 10^{-2}$
W-IK	4.67[4.28, 4.76]	$(4.33\ [\ 2.46,\ 8.13])\times 10^{-3}$	**11.65 [11.65, 11.65]**	$(\mathbf{3.57[\ 1.47,\ 4.25]})\times 10^{-5}$
Sb-M	8.18[8.00, 8.26]	$(11.30\ [\ 8.69,\ 15.10])\times 10^{-3}$	**11.65 [11.65, 11.65]**	$(\mathbf{2.96\ [\ 1.73,\ 3.22]})\times 10^{-5}$
NsGP	8.72[8.64, 8.77]	$(2.06\ [\ 1.36,\ 3.91])\times 10^{-3}$	**11.65 [11.65, 11.65]**	$(\mathbf{1.27\ [\ 0.63,\ 2.35]})\times 10^{-5}$
ProMP	8.34[6.70, 9.27]	$(13.69\ [\ 8.07,\ 25.29])\times 10^{-3}$	9.47[9.27, 9.55]	$(3.76\ [\ 2.84,\ 4.58])\times 10^{-2}$
MI-NsGP	**8.93[8.83,8.97]**	$(\mathbf{1.69\ [\ 1.26,\ 2.80]})\times 10^{-3}$	**11.65 [11.65, 11.65]**	$(\mathbf{2.41\ [\ 1.05,\ 3.81]})\times 10^{-5}$

[†1] Inria, University of Lorraine, CNRS, LORIA, F-54000, France.
[‡2] Université de Lorraine, CNRS, CRAN, F-54000, France. `lorenzo.vianello@univ-lorraine.fr`

2 Human posture prediction

2.1 Experiment 1

In this first experiment we record ten trajectories during a human-robot collaboration scenario (Fig. 2): in particular, in this experiment, the robot is repeating the same task-space trajectory while the human follows. The robot executes four pick and place movements for each trajectory.

Figure 2: Set-up for the Experiment 1: Human and robot are coupled, robot executes the same trajectory ten times while the human follows its movements.
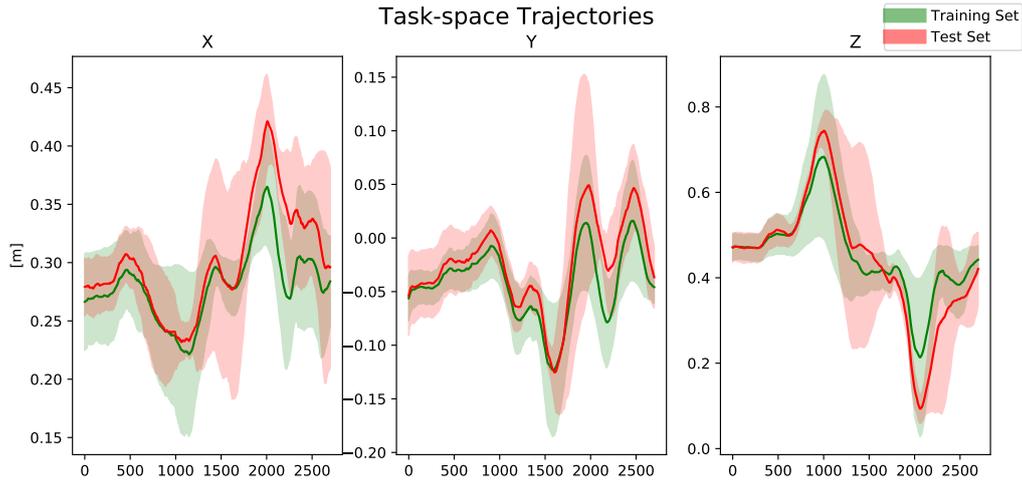

Figure 3: End-effector's trajectories used in the first human-robot interaction scenario: in this case the training set and the test set are belong to the same movement primitive.

	M-LL $\dot{q}$	R-MSE $\dot{q}$	M-LL $\dot{x}$	R-MSE $\dot{x}$
GP	7.69 [7.50, 7.84]	$(43.33\ [\ 23.05,\ 86.41])\times 10^{-3}$	7.04 [6.86, 7.20]	$(2.63\ [\ 1.64,\ 4.91])\times 10^{-2}$
W-IK	7.05 [6.68, 7.12]	$(2.82\ [\ 1.24,\ 6.40])\times 10^{-3}$	**11.74 [11.74, 11.74]**	$(\mathbf{1.66\ [\ 0.78,\ 2.85]})\times 10^{-4}$
Sb-M	8.57 [8.25, 8.65]	$(3.48\ [\ 1.37,\ 8.09])\times 10^{-3}$	**11.74[11.74, 11.74]**	$(\mathbf{1.66\ [\ 0.78,\ 2.85]})\times 10^{-4}$
NsGP	8.82 [8.39, 9.10]	$(2.9\ [\ 1.31,\ 5.53])\times 10^{-3}$	**11.74 [11.74, 11.74]**	$(\mathbf{1.66\ [\ 0.76,\ 2.83]})\times 10^{-4}$
ProMP	9.02 [8.20, 9.46]	$(2.30\ [\ 1.37,\ 3.83])\times 10^{-3}$	11.45 [11.37, 11.50]	$(6.22\ [3.37,\ 10.69])\times 10^{-3}$
MI-NsGP	**9.32 [8.81, 9.59]**	$(\mathbf{2.19\ [\ 1.24,\ 3.38]})\times 10^{-3}$	**11.74[11.74, 11.74]**	$(\mathbf{1.66\ [\ 0.78,\ 2.84]})\times 10^{-4}$

2.2 Experiment 2

In this experiment we record ten trajectories during a human-robot collaboration scenario (Fig. 5) while the robot end-effector is controlled using a joystick. We record ten trajectories while the human is collaborating with the robot. In each trajectory the operator which uses the joystick choose some random movements and the human subject which collaborates with the robot must follow these movements.
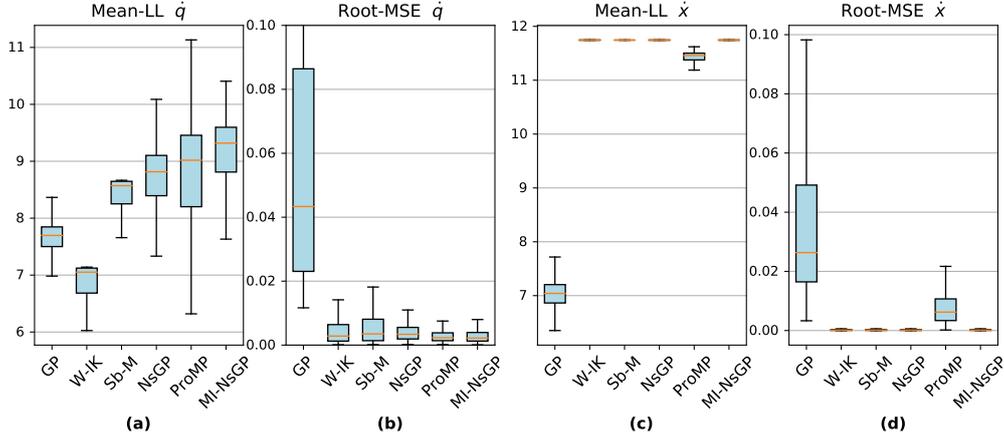
Figure 4: Comparison of methods for joint velocity prediction in Human Joint Velocity Prediction: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the EE velocity (d) R-MSE on the EE velocity.
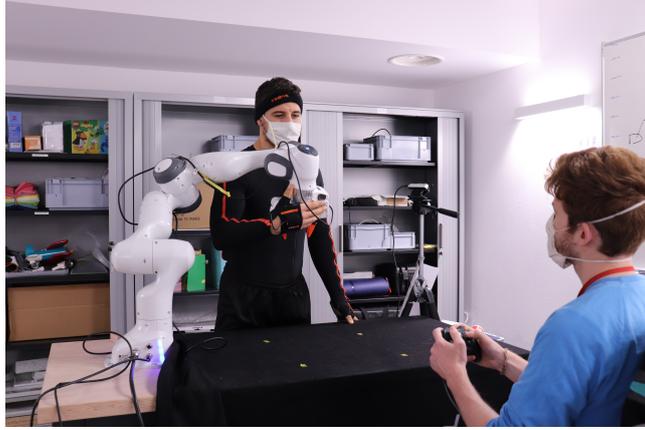

Figure 5: Set-up for the Experiment 2: Robot executes pseudo-random movements while it is controlled by a second operator using a joystick.

	M-LL $\dot{q}$	R-MSE $\dot{q}$	M-LL $\dot{x}$	R-MSE $\dot{x}$
GP	7.22 [7.02, 7.27]	$(57.13\ [\ 45.74,\ 94.05])\times10^{-3}$	6.27 [6.06, 6.32]	$1.02\ [\ 0.73,\ 1.73])\times10^{-1}$
W-IK	6.85 [5.36, 7.10]	$(10.08\ [\ 6.67,\ 17.64])\times10^{-3}$	**11.74[11.74, 11.74]**	**1.66 [0.77, 2.83])**$\times10^{-4}$
Sb-M	7.24 [7.13, 7.27]	$(15.73\ [12.27,\ 23.03])\times10^{-3}$	**11.74[11.74, 11.74]**	**1.66 [0.76, 2.83])**$\times10^{-4}$
NsGP	7.97 [7.79, 8.13]	$(9.35\ [4.18,\ 10.85])\times10^{-3}$	**11.74[11.74, 11.74]**	**1.66 [0.76, 2.83])**$\times10^{-4}$
ProMP	8.09 [7.44, 8.34]	$(11.04\ [\ 8.70,\ 15.25])\times10^{-3}$	10.94 [10.84, 10.99]	$1.35\ [\ 0.81,\ 2.11])\times10^{-2}$
MI-NsGP	**8.72 [8.52, 8.81]**	**(4.34 [3.93, 6.22])**$\times10^{-3}$	**11.74[11.74, 11.74]**	**1.66 [0.76, 2.83])**$\times10^{-4}$

2.3 Experiment 3

In this third experiment we record ten trajectories during a human-robot collaboration scenario (Fig. 8): we chose another human operator to test if the algorithm could be used for a different person with different body shape (in the first experiment the operator is a male subject of of height $1.84m$ and weight $90kg$, in the second experiment the operator is a female subject of height $1.68m$ and weight $64kg$). We record ten trajectories while the human is collaborating with the robot in a seated position; each trajectory consists of a combination of three pick and place gestures over a table.

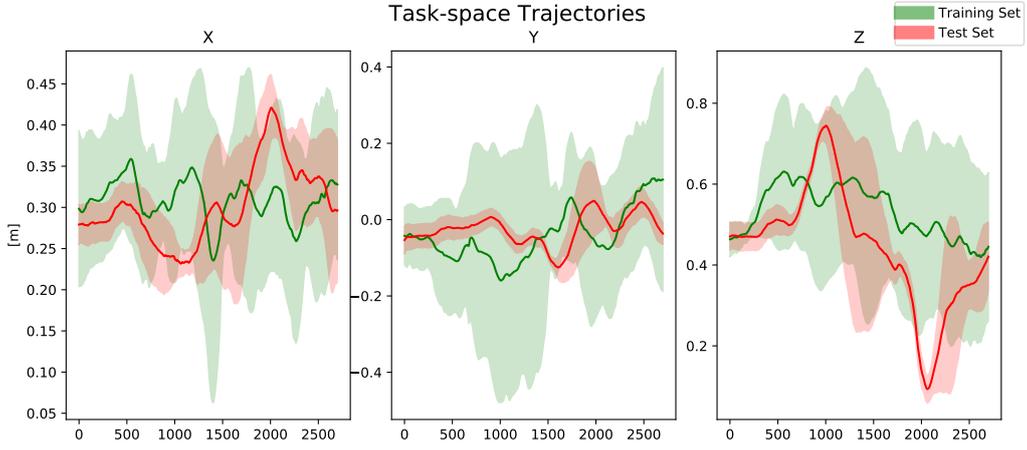|	M-LL $\dot{q}$	R-MSE $\dot{q}$	M-LL $\dot{x}$	R-MSE $\dot{x}$
GP	8.50 [8.17, 8.81]	$(28.93\ [\ 25.94,\ 44.06])\times10^{-3}$	8.78 [8.51, 9.07]	$9.10\ [\ 4.28,\ 20.98])\times10^{-3}$
W-IK	7.12 [7.00, 7.14]	$(3.47[1.0,7.92])\times10^{-3}$	**12.91 [12.91, 12.91]**	**2.43 [0.25, 8.40])**$\times10^{-6}$
Sb-M	8.92 [8.55, 8.97]	$(4.20\ [\ 1.63,\ 10.37])\times10^{-3}$	**12.91 [12.91, 12.91]**	**2.43 [0.25, 8.40])**$\times10^{-6}$
NsGP	9.28 [9.10, 9.47]	$(2.01[\ 1.15,\ 4.38])\times10^{-3}$	**12.91 [12.91, 12.91]**	**2.43 [0.25, 8.40])**$\times10^{-6}$
ProMP	9.22 [8.39, 9.61]	$(3.67\ [\ 2.43,\ 5.35])\times10^{-3}$	12.45 [12.16, 12.70]	$6.76\ [\ 4.75,\ 10.02])\times10^{-6}$
MI-NsGP	**9.47 [9.28, 9.67]**	**(1.93 [1.05, 4.13])**$\times10^{-3}$	**12.91 [12.91, 12.91]**	**2.43 [0.25, 8.40])**$\times10^{-6}$

Figure 6: End-effector's trajectories used in the first human-robot interaction scenario: in this case the training set and the test set are belong to different movement primitives.
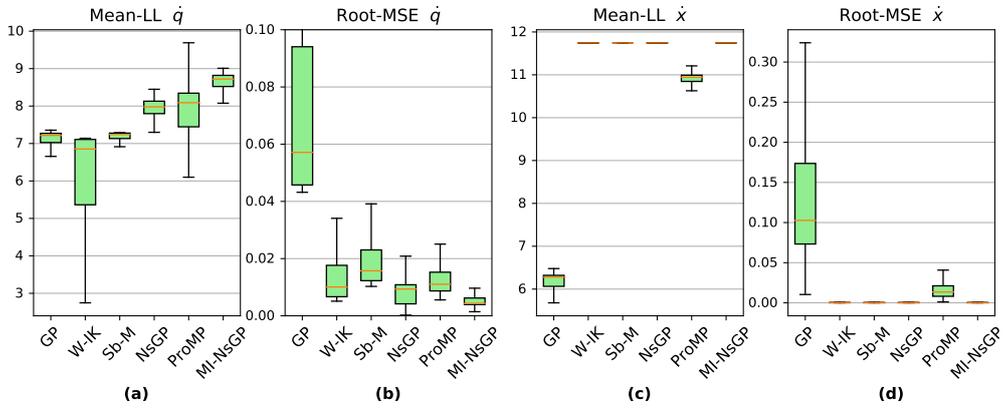

Figure 7: Comparison of methods for joint velocity prediction in Human Joint Velocity Prediction: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the EE velocity (d) R-MSE on the EE velocity.

References

[1] R. Rahal, G. Matarese, M. Gabiccini, A. Artoni, D. Prattichizzo, P. R. Giordano, and C. Pacchierotti, "Caring about the human operator: haptic shared control for enhanced user comfort in robotic telemanipulation," *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 197–203, 2020.

[2] B. Busch, G. Maeda, Y. Mollard, M. Demangeat, and M. Lopes, "Postural optimization for an ergonomic human-robot interaction," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2778–2785.

Figure 8: Set-up for the Experiment 3: Human and robot are coupled, robot executes the same trajectory ten times while the human follows its movements.
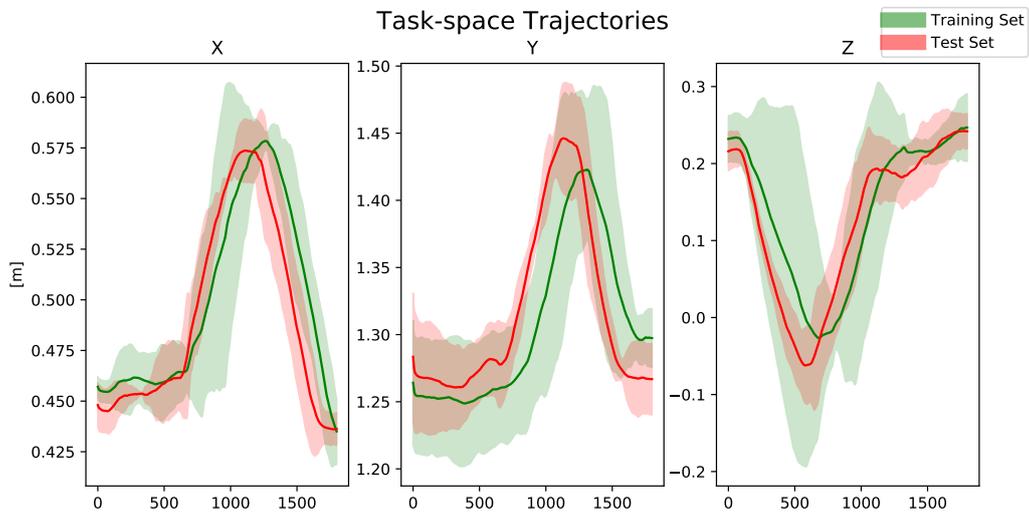

Figure 9: End-effector's trajectories in the third human-robot interaction scenario: in this case the training set and the test set are belong to the same movement primitive, a different human operator executes the movements with respect to the other scenarios.
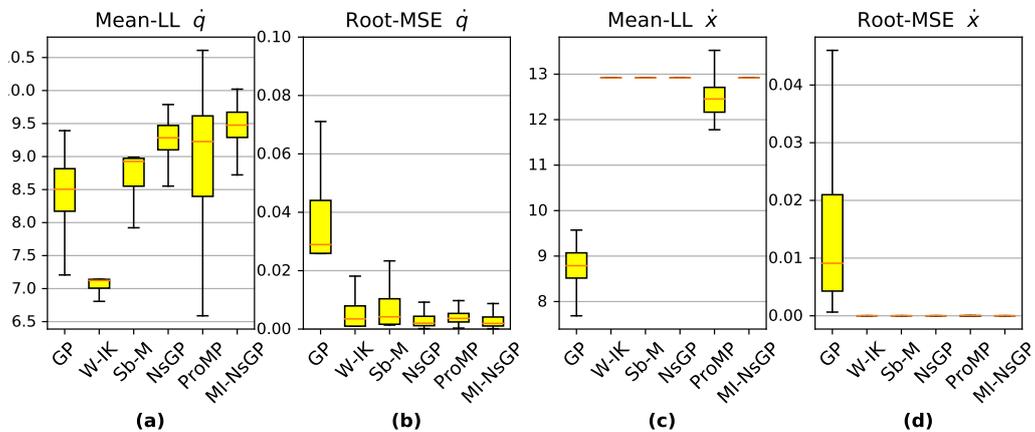

Figure 10: Comparison of methods for joint velocity prediction in Human Joint Velocity Prediction: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the EE velocity (d) R-MSE on the EE velocity.