Probabilistic Inverse Kinematics for Human Posture Prediction during Physical Human-Robot Interaction

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

Probabilistic Inverse Kinematics for 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 the point of view of ergonomics. 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. Previous approaches to solve this problem relied on the inverse kinematics, but did not consider the human body redundancy nor the kinematic constraints imposed by the physical collaboration, nor any prior observations of the human movement execution. We propose a data-driven approach that addresses these limits. The key idea of our algorithm 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 show in a simulated toy problem and on real human-robot interaction data that our method outperforms model-based inverse kinematics prediction, sample-based prediction and regression methods that do not consider geometric constraints. Our method is validated on a a collaboration scenario with a human interacting physically with the Franka robot.**

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 needs 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 in defining contact points, collaboration control laws with structured roles (e.g., leader-follower) and the amount of provided assistance [3].

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. In other words, humans can execute the same task in many different ways. 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.

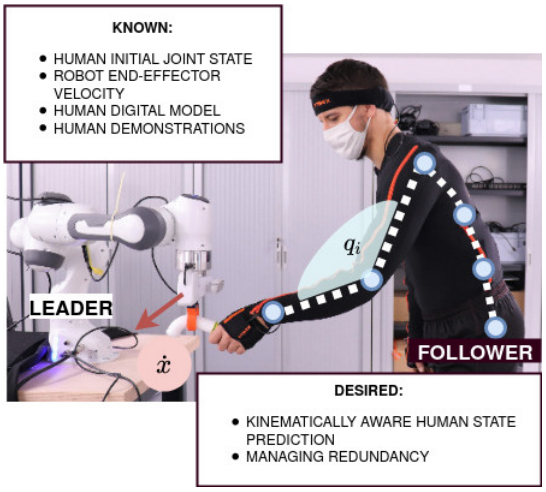


Fig. 1: The human posture is influenced by the robot's trajectory during physical interaction, but the human may adopt different postures during each task execution. In this paper we want the robot to predict the human posture given a known Cartesian trajectory of its end-effector and prior observations of the task executed by the human. The human posture is measured online by a wearable Xsens MVN suit.

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. Wearable motion tracking systems can be easily used to acquire postural information of humans interacting with robots [4]. The most common strategy to predict human movements is to use trajectory-centric models such as movement primitives, which are probabilistic models capturing the variability of human demonstrations for solving a task. However, these models usually represent task-level trajectories (e.g., Cartesian trajectories of the human hand and the robot end-

*This work was partially supported by the European Union Horizon 2020 Research and Innovation Program under Grant Agreement No. 731540 (project AnDy),the CHIST-ERA project HEAP, the European FEDER in the context of the CPER Sciarat and the Lorraine University of Excellence (LUE) project C-Shift.

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

effector) [5], [6], which means that they have to compute the inverse kinematics to find joint trajectories or learn directly task-specific primitives in the joint space. In the latter case, 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 is tricky to deal with, especially when the human is physically coupled to the robot because it could 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). 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. We call this problem "probabilistic inverse kinematics of the human body". 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 the one a humanoid robot, with similar anthropometrics data of the human (height, weight). 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 the human configurations that all lead to the same end-effector position. By doing so, we can query the learned model and combine it with the null space of the end-effector to be guaranteed that the posture leads to the end-effector position. For each point of the robot trajectory, our method consist in first projecting the postures in the null space defined by the end-effector position and the kinematic model of the human, then learn Gaussian processes that predicts projected configurations, and project back to the original joint space. We use Gaussian processes [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 the 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 a good accuracy and kinematics consistency, and this knowledge enables the robot to anticipate the human movement and discard trajectories that could make the human execute ergonomically unsafe movements.

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) has been used to predict the human pose starting from the end-effector position using simplified human models as in [9]. 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, and therefore we are forced to formulate the problem as a Probabilistic Inverse Kinematics problem. 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 [10] and to task specific loss functions [11]. We refer to this kind of methods as sampling based approaches. These kind of methods are computationally (and time) 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 [6] 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 Inverse Kinematics is used to find the most appropriate corresponding joint trajectories to fulfil 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]. Recurrent neural networks have also been proposed for predicting future human posture [13], [14]. One of the main challenge of these methods is to encode the multi-value behaviour of the human, coming by its redundant structure, and to evaluate the different solutions [15]. Data-driven of 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 [16]. In [17] a correction phase has been added to match the kinematic constraints imposed by the collaborative robot.

C. Digital Human Models for Ergonomics

A collaborative robot can be used to assist the human worker and improve ergonomics at work [18]. 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 (e.g., as in AnyBody or OpenSim), which are rather complex, have many degrees of freedom, and enable to

analyse the human movement by simulating the muscular efforts [19]; 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 [18]. 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 [18]. Several ergonomics scores exist (e.g., RULA, REBA), and they are mostly based on postural information [20].

III. Notation

In our study, the human is represented by a DHM, a rigid body model with n degrees of freedom. The following notation is used for the DHM:

- $q \in \mathbb{R}^n$ is the vector of joints values;
- $x \in \mathbb{R}^m$ hand Cartesian pose (position and orientation);
- $n > m$ overactuated condition, $k = n - m$ 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, such that $\dot{x} = J(q)\dot{q}$.
- We define as $y \in \mathbb{R}^k$ a vector in the null space of the Jacobian $J(q)$.

The robot state is determined by $x_R, \dot{x}_R$, i.e., the Cartesian position and velocity of its end-effector (EE). The robot joints q_R are not used in this work.

IV. Problem formulation

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 end-effector is known at each time step: $\dot{x}_R(t), t = 0, \ldots, T-1$.[1] The two agents are physically coupled at their end-effectors; 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 ignore for the moment any linear 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 end-effector 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 [11]); 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.

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

V. Background

A. Kinematics for redundant DHMs

A redundant DHM[2] 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 end-effector motion. Given an EE pose $x \in \mathbb{R}^m$, the space which contains all the solutions of the inverse kinematics equation $\{q : x = f(q)\}$ is defined as the inverse kinematics's manifold $\mathcal{M}_x$. It is considered as an union of more simple and continuous manifolds, called "self-motion manifold" ($\mathcal{M}_s$) [21]. Any change of joint configuration along a self-motion manifold does not change the position of the end-effector. Instantaneous joint velocity vectors that are tangential to the manifold generate so-called "self-motions". These motions $\dot{q}_s$ do not change the end-effector position: $J(q)\dot{q}_s = 0$. 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 by the columns of the matrix $V_N = [\mu_1, ..., \mu_{n-m}]$; this matrix could 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 [16]. 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 could 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 dual projection method:

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

where $J_W^\dagger(q)$ is the weighted pseudo-inverse:

$$J_W^\dagger(q) = WJ^\top(q)\Big(J(q)WJ^\top(q)\Big)^{-1} \tag{4}$$

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 $\mathcal{C}(.)$ [22][23].

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 $\mathbf{f}(\mathbf{x}) \in \mathcal{Y}$ for the given input $\mathbf{x} \in \mathcal{X}$.

[2] The reader may notice that "redundant DHM" is equivalent to "redundant robot", since the DHM is modeled essentially as a robot with rigid bodies.

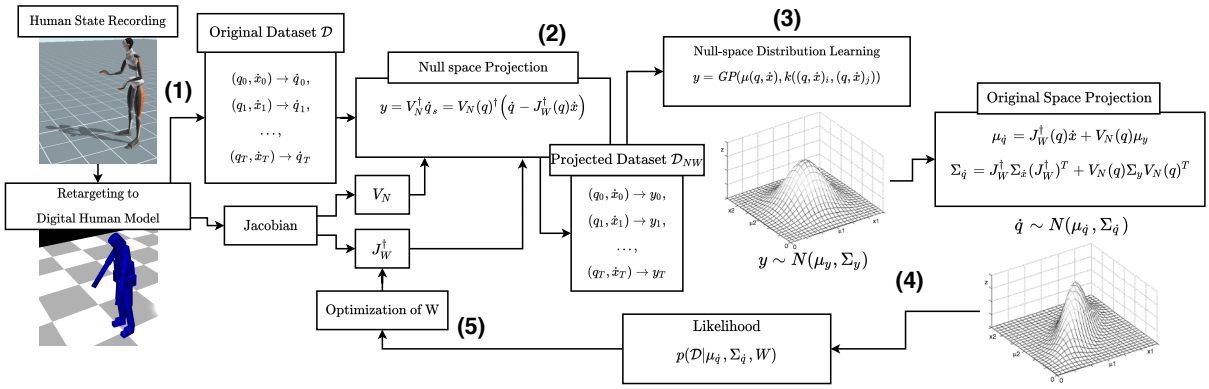


Fig. 2: Flowchart of the offline training: (1) We collect human movements using a motion capture suit. The joint states are passed to a digital human model and they are used to calculate the Jacobian at each joint configuration. From the digital human model we record also the dataset $\mathcal{D}$. (2) 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).

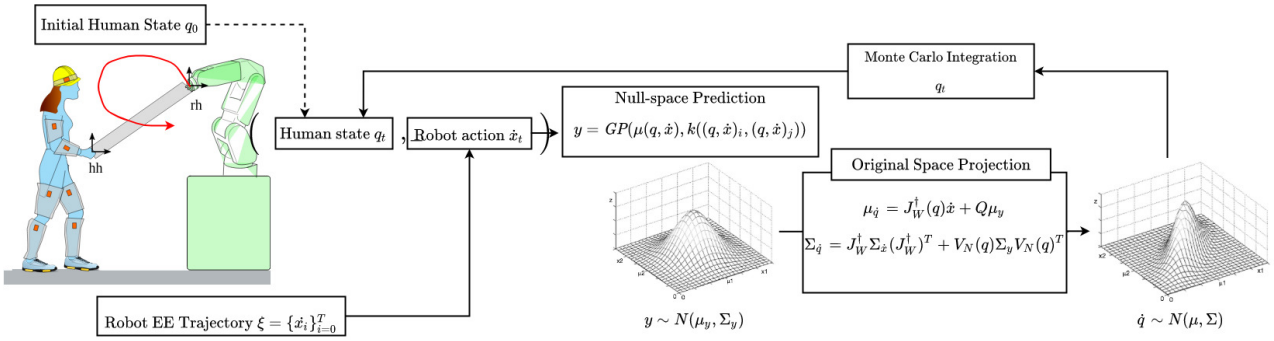


Fig. 3: Flowchart of the online prediction: Given an EE trajectory imposed by the robot and the knowledge of the initial human state we can predict a distribution over the future human states. To do that we sample on the human joint velocity $\dot{q}$ calculated with our method(MI-NsGP) then we integrate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

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

Let $\mathcal{D} = \{(\mathbf{x_i}, \mathbf{y_i}) | \mathbf{x_i} \in \mathcal{X}, \mathbf{y_i} \in \mathcal{Y}\}$ be a training set and $(\mathbf{x}_*, \mathbf{y}_*)$ a point we did not observe in $\mathcal{D}$. The GP predictive distribution for the output $\mathbf{y}_*$ at the test input $\mathbf{x}_*$, given in vector form, is

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

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

VI. Method

We consider a DHM with n degrees of freedom. We assume that the human/DHM follows this classic control law from robotics (section V-A, [22], [23]):

$$\dot{q} = J_W^\dagger(q)\dot{x} + (I - J_W^\dagger(q)J(q))z(q) \tag{5}$$

where $z(q)$ is an unknown cost function and the weight 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 cost function $z(q)$ with Gaussian processes

Let us consider a dataset related to the motions of the DHM, $\mathcal{D}$, 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 $\dot{q} \in \mathbb{R}^n$: $\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=1}^{N_D}$. This dataset

can be generally acquired via human motion tracking (see Fig.2-(1)).

At this stage, we consider the values of the weight matrix W of Eq. 5 to be known, for example $W = I$, where I is the identity matrix (see section VI-B for learning W).

Instead of learning directly the value of $z(q)$, we notice that we can write [24]:

$$V_N(q)y = (I - J_W^\dagger(q)J(q))z(q) \tag{6}$$

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 (Fig.2-(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:

$$y(q, \dot{x}) = V_N^\dagger(q)\Big(\dot{q} - J_W^\dagger(q)\dot{x}\Big) \tag{7}$$

Thus, given the dataset $\mathcal{D} = \{(q_i, \dot{x}_i), \dot{q}_i\}_{i=1}^{N_D}$, we apply Eq. 7 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. V-B) that map input the current joint state and the EE velocity to the joint velocity:

$$y|(q, \dot{x}) \sim \mathcal{GP}\Big(m(q, \dot{x}), k((q, \dot{x})_i, (q, \dot{x})_j)\Big) \tag{8}$$

Like it is often done [25], we train k independent GP, one for each dimension of y. Since Eq. 7 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:

$$\mu_{\dot{q}}(q, \dot{x}) = J_W^\dagger(q)\dot{x} + V_N(q)\mu_y \tag{9}$$

$$\Sigma_{\dot{q}}(q, \dot{x}) = J_W^\dagger \Sigma_{\dot{x}} (J_W^\dagger)^\top + V_N(q)\Sigma_y V_N(q)^\top \tag{10}$$

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

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 (Fig.2-(5)). 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 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. 10 (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 [26], which is a gradient-free stochastic optimizer available in the "pycma[3]" Python library.

[3] https://github.com/CMA-ES/pycma

C. Prediction Phase

Once the model has been trained, it can been used to predict trajectories of the human's joints given the current joint configuration q_t and 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). The use of $f(q_t)$ instead of x will be clear when we will compare our approach to methods that do not guarantee that $f(q_t) = x$ (Sec. VII).

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. 10). From this multivariate Gaussian distribution, we can sample $\dot{q}_t$, which allows us to compute the q_{t+1}:

$$q_{t+1} \sim q_t + \Delta_t\, \mathcal{N}\left(\mu_{\dot{q}}(q_t, \dot{x}_t), \sigma_{\dot{q}}(q_t, \dot{x}_t)\right) \tag{11}$$

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 estimate of the distribution over the human joint trajectories according to our model [27]. A schema of the prediction phase is depicted in Fig. 3.

VII. Experiments

To evaluate our method, 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. VI)
   $\dot{q}|(q, \dot{x}) \sim J_W^\dagger(q)\dot{x} + V_N(q)\mathcal{GP}(q, \dot{x})$
2) **GP**: learning directly from data with a GP
   $\dot{q}|(q, \dot{x}) \sim \mathcal{GP}(q, \dot{x})$.
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 has been used for all the methods.

The methods were evaluated on two experiments: A) predicting the joint state of a 5R planar robot controlled by a biased IK function (in simulation); B) predicting the human posture (joint configuration) during a human-robot collaboration task (from real data, a human interacting with the Franka robot).

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, thus a planar task can be executed with several joints velocities.

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

$$z(q) = \frac{\nabla \mathcal{C}(q)}{\nabla q} + \mathcal{N}(0, \sigma_z) \tag{12}$$

To define $z(q)$ similar to the human model, we hypothesized, as in [11], 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 [20], the sum of a second order polynomial for each joint:

$$\mathcal{C}(q) = \sum_{j=1}^{n} (p_{2,j} q_j^2 + p_{1,j} q_j + p_{3,j}) \tag{13}$$

Where $\{(1.1 \times 10^{-03}, 0.0, 0.0), (9.8 \times 10^{-04}, 0.0, 1.0), (1.6 \times 10^{-04}, -2.5 \times 10^{-02}, 2.0), (1.2 \times 10^{-04}, 0.0, 0.0), (2.1 \times 10^{-03}, 0.0, 1.0)\}$ are the (p_1, p_2, p_3) of the joint $j \in [0, ..., n]$; they have been calculated by fitting a second degree polynomial within the RULA score.

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 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 chose to bound the values because otherwise we could fall in a singular configuration in which a joint never moves or always moves, which appeared far from a human-like behavior.

Starting from a configuration q_0, we applied N_D times the control law specified in the Equations 5 and 12, 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 by $N_D = 10^3$ points, has been normalized and divided into a training and a validation set. We trained the models using the training set. Each GP has been implemented in Python using *gpytorch* library with constant mean and the RBF kernel. For finding the values of the parameters W, the BIPOP-CMA-ES optimizer searches in $[0 + \epsilon, 2 - \epsilon]$. 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 (Fig.4-a). Overall, our method (MI-Ns-GP) leads to significantly better likelihood values than all the control approaches. The worst likelihoods are obtained by the methods that do not use the null space and we observe a large gap between using and not using the null space projection.

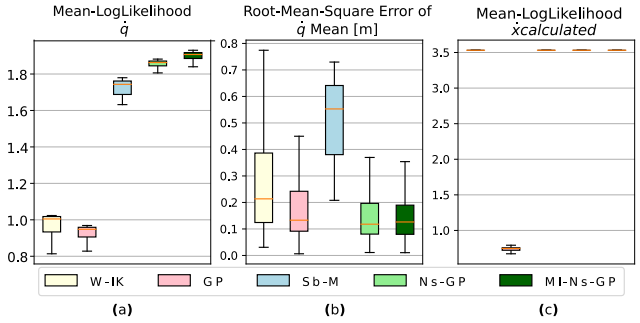


Fig. 4: Comparison of methods for joint velocity prediction for 5R manipulator: our method (MI-NsGP), learning directly from data using Gaussian Process (GP), learning W and apply pseudo-inverse (W-IK), learning null space (NsGP), sampling in the null space (Sb-M). The criteria: (a) Mean-Log-Likelihood 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.

Among the methods that use the null space, learning W makes a significant difference.

We then focused on the mean prediction by computing the root mean square error on $\dot{q}$ (Fig.4-b) (we ignored the variance). As before, the best results are obtained with our method, and using the 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 (Fig.4-a). This means that this method has a very large variance, which makes the test set likely (high likelihood score) but the predictions very inaccurate.

Last, we computed the mean log-likelihood of the end-effector position ($\dot{x}$, Fig.4-c). As expected, perfect scores are obtained with the methods that exploit the null space (W-IK, Sb-M, NS-GP, MI-NS-GP), but learning directly a GP that predicts $\dot{q}$ directly leads to significant errors in the end-effector position. 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 prediction of the joint velocity while it returns only solutions which satisfy the kinematic constraint.

B. Human IK prediction

We then evaluated our method with a real dataset of a human interacting with the Franka Emika Panda robot. We used a motion capture suit (Xsens MVN) to capture the human posture. The human and the robot are facing each other like in Fig. 1, and the right hand of the human is always in contact with the robot's EE (see video attachment). The robot executes four "pick and place" trajectories spanning 50cm, its orientation is maintained constant. Each trajectory is repeated 10 times: the first five trajectories make the training set, and the five remaining ones the test set. Fig.7a shows the intrinsic variability of the human repetitions (for

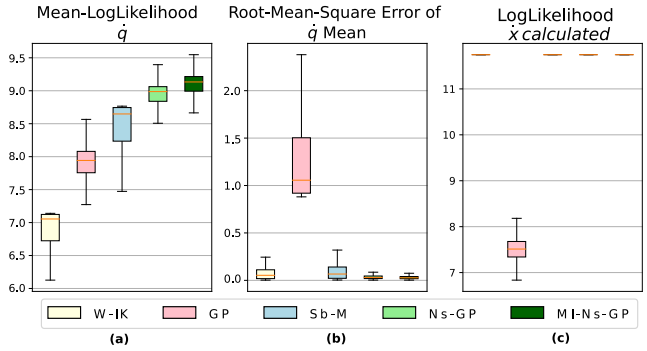
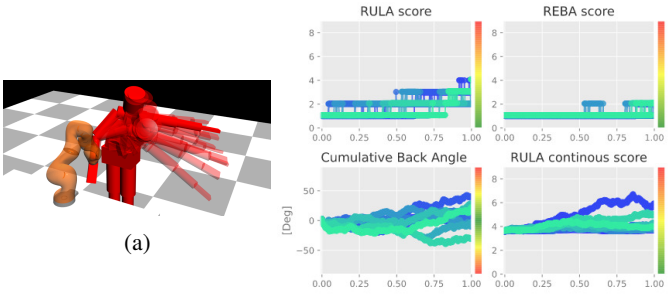


Fig. 5: Comparison of methods for joint velocity prediction in Human Joint Velocity Prediction: (a) Mean-Log-Likelihood of the predicted joint velocity (b) R-MSE between the mean of the predicted joint velocity and real value (c) Mean-LogLikelihood of the EE velocity.



(a) (b)

Fig. 6: (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.

the same EE movement, the joint trajectories change). The 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.

The human poses are fitted (retargeted) to a DHM of 66 segments (Fig. 6a), 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 [18]. 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 [28] to calculate the Jacobian going from the human pelvis to the right hand for a given human joint configuration. Regarding the parameters optimization, we applied BIPOP-CMA-ES and the search is in $[0+\epsilon, 2-\epsilon]$. In the prediction phase we also sampled 10 trajectories using Monte-Carlo approach and for each of them we calculated four different ergonomics scores from the state of the art in human ergonomics [18]: RULA, REBA, RULA continuous, cumulative back angle (Fig.6b). The purpose is to show that the probabilistic IK also impacts the prediction of ergonomics scores, which is a critical information for a collaborative robot.

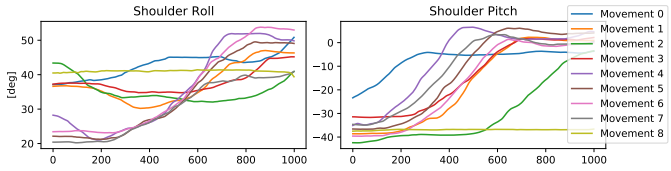
Results We observed that Sb-M improves the Log-Likelihood of the prediction with respect of using GP; there is a further improvement using Ns-GP to predict the self-motion velocities, in the end by identifying the weights W using MI-Ns-GP we can outperform the previous methods[4] (Fig.5.a, GP: (median: $7.94[7.75, 8.08]$), W-IK: $(7.05[6.73, 7.12])$, Sb-M: $(8.64[[8.23, 8.74])$, NsGP: $(9.13[[8.99, 9.21])$; MI-NsGP: $(8.95[8.87, 9.03])$). In the toy problem we observed that Sb-M performs worse than the other methods in simple regression. Instead in this situation the worst performance has been reached using GP; moreover, in this case W-IK has lower minimum value but NsGP and MI-NsGP perform better in the median and max value of the 95^{th} percentile (Fig.5.b, GP: (median: $1.055([0.92, 1.50])$, W-IK: $(0.053([0.014, 0.111])$, Sb-M: $(0.067([0.021, 0.14])$, NsGP: $(0.030([0.018, 0.045])$, MI-NsGP: $(0.027([0.017, 0.040]))$. Regarding the ability of satisfying the kinematic constraint, we observed a behavior similar to the toy-problem. In fact model based methods (Sb-M, Ns-GP, MI-Ns-GP) have always bigger likelihood (Fig.5.c, Log-Likelihood, GP: $7.51, [7.33, 7.67])$, Others: $(11.74, [11.74, 11.74]))$; R-MSE: GP:$(1.367[1.13, 1.68]) \times 10^{-04}$, Others: $(1.234[0.598, 2.52]) \times 10^{-09}$). This improvement is even more evident at trajectory level: if we use the GP alone to predict the DHM joint trajectories while the prediction horizon is growing, the R-MSE between the EE of the DHM (in Fig.7b the red progression) and the robot's EE grows too fast to be used in a safe human-robot collaboration scenario (after 0.5s of trajectory execution the median error is already 5.0cm, growing at 8.8cm after 1.0s) while if we use Ns-GP (in Fig.7b the green progression) the error is acceptable (after 0.5s of trajectory execution the median error is 0.03cm, 0.04cm after 1.0s). In searching the best W using MI, we applied the same considerations gained in the toy problem and looked for the solution with maximum module inside the bounds. In the case of the human, W is unknown; thus, it is not straightforward to evaluate the resulting values from model identification. Anyway some considerations are possible: even considering different training-sets, the optimization converges to the same values of W; this values agree with our expectations regarding the distribution of the joint velocity, in fact the joints which move less (like for example the lumbar joints) have a smaller value with respect to those which are more involved in the execution of the movement (like shoulder and elbow).

[inline]No Type 3 fonts

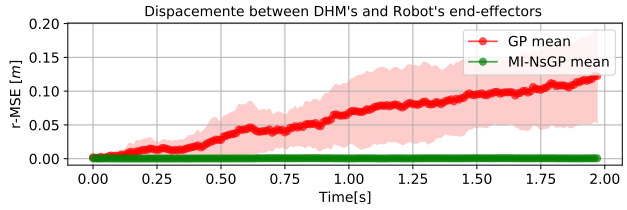
VIII. CONCLUSIONS

In this paper we presented a method for learning a probabilistic inverse kinematic model of the human in a Human-Robot Collaboration scenario where the human hand motion is constrained by the robot's end-effector. We propose a two phases 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

[4]We report the results with the notation *median* [*IQR* 25%, 75%].



(a)



(b)

Fig. 7: (a) Human joint trajectories (shoulder's roll and pitch) in response to the same EE movements. (b) Mean square Error in offline prediction with GP and MI-NsGP.

the second phase we optimize the weights of 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 integrate this probabilistic prediction into our framework for ergonomics control, which aims at optimizing 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 oh ergonomics scores for a given robot's EE trajectory, which is a critical element for planning and optimizing 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] F. Romano *et al.*, "The CoDyCo project achievements and beyond: Toward human aware whole-body controllers for physical human robot interaction," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 516–523, 2018.

[5] R. Lioutikov, G. Maeda, F. Veiga, K. Kersting, and J. Peters, "Learning attribute grammars for movement primitive sequencing," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 21–38, 2020.

[6] 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.

[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. 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.

[10] 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.

[11] 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.

[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] 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.

[14] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of human full-body movements with motion optimization and recurrent neural networks," *arXiv preprint arXiv:1910.01843*, 2019.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.