Learning torque control in presence of contacts using tactile sensing from robot skin

Roberto Calandra, Serena Ivaldi, Marc Peter Deisenroth, Jan Peters

Learning Torque Control in Presence of Contacts using Tactile Sensing from Robot Skin

Roberto Calandra[1] and Serena Ivaldi[2,1] and Marc Peter Deisenroth[3] and Jan Peters[1,4]

Abstract— **Whole-body control in unknown environments is challenging: Unforeseen contacts with obstacles can lead to poor tracking performance and potential physical damages of the robot. Hence, a whole-body control approach for future humanoid robots in (partially) unknown environments needs to take contact sensing into account, e.g., by means of artificial skin. However, translating contacts from skin measurements into physically well-understood quantities can be problematic as the exact position and strength of the contact needs to be converted into torques. In this paper, we suggest an alternative approach that directly learns the mapping from both skin and the joint state to torques. We propose to learn such an inverse dynamics models with contacts using a mixture-of-contacts approach that exploits the linear superimposition of contact forces. The learned model can, making use of uncalibrated tactile sensors, accurately predict the torques needed to compensate for the contact. As a result, tracking of trajectories with obstacles and tactile contact can be executed more accurately. We demonstrate on the humanoid robot *iCub* that our approach improve the tracking error in presence of dynamic contacts.**

I. Introduction

A key challenge for torque-controlled humanoid robots is to accurately control their dynamics and the contacts with the environment, for example during manipulation in clutter [1], locomotion [2] and whole-body movements [3]. In such complex situations, the dynamics of the system can be exploited to compute the required joint torques necessary to realize the desired trajectories of the joints. This is the idea at the base of computed torque control [4], also used in inverse dynamics control [5] and model predictive control applied to robot reaching and locomotion [6], [7].

Inaccurate joint torques estimation can deteriorate the performances of computed torque control and tracking of desired trajectories, which can ultimately cause severe problems when controlling the robot motion in presence of uncertainties, disturbances and unexpected contacts with the environment. However, accurate estimation of joint torques and external forces in presence of contacts is difficult without dedicated sensors. This is also true for robots equipped with whole-body distributed tactile sensing, as small errors in the estimation of the contact location can substantially deteriorate the quality of the inverse dynamics estimation [8].

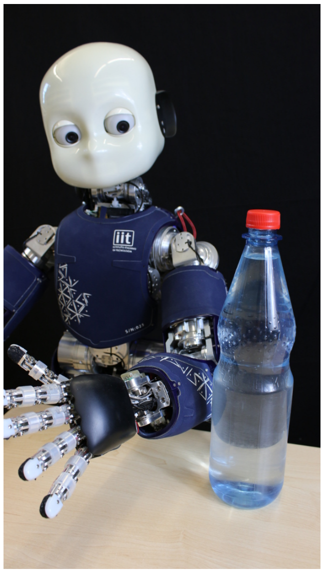

Fig. 1: The humanoid robot *iCub*. Its movement is obstructed by an unexpected obstacle, which decreases the tracking performance of a desired trajectory. We reduce the tracking error by learning the torque control to compensate the contact.

Two main lines of research are currently addressing this issue. On one side model-based approaches, which rely on identification techniques to improve the dynamics parameters of the analytic models [9], [10]. On the other side, we have machine learning approaches, where non-parametric models of the robot dynamics can be learned from data collected on the robot [11], [12], [13]. The main advantages of the latter approaches are that they do not usually require a prior calibration (kinematics/dynamics calibration [14] or spatial calibration of the tactile sensors [15]), are less prone to model errors, and can easily deal with noise and varying contexts [11].

In previous work [12], we proposed a data-driven mixture-of-experts learning approach based on Gaussian Processes, which predicts joint torques from joint encoders, tactile and force/torque sensors data. We applied such a model to learn the joint torques of the arm of *iCub* in presence of several varying contacts, showing that the learned model was more accurate than analytic dynamics models using the same sensors data [16]. However, the proposed learning approach required the availability of joint torque sensors to supervise the learning process and provide the ground truth; Therefore limiting the application of this approach in many robots, in particular to almost all *iCubs*[1].

In this paper, we generalize the previous approach, re-

The research leading to these results has received funding from the European Council under grant agreement #600716 (CoDyCo - FP7/2007–2013). MPD was supported by an Imperial College Junior Research Fellowship.

[1] Intelligent Autonomous Systems, TU Darmstadt, Germany calandra@ias.tu-darmstadt.de

[2] Inria, Villers-lès-Nancy, F-54600, France; CNRS, Loria, UMR n.7503 and Université de Lorraine, Vandoeuvre-lès-Nancy, F-54500, France serena.ivaldi@inria.fr

[3] Department of Computing, Imperial College London, United Kingdom m.deisenroth@imperial.ac.uk

[4] Max Planck Institute for Intelligent Systems, Tübingen, Germany mail@jan-peters.net

[1] Only one *iCub*, *iCubParis02*, is currently equipped with partial joint torque sensing.

laxing the need for joint torque sensors on the robot. Most importantly, we demonstrate in an active control task with the humanoid robot *iCub* (see Figure 1) the effectiveness of the proposed model, by showing that it can be efficiently used for computed torque control in presence of contacts with the environment. With the learned model we can decrease the tracking error in presence of known or novel contacts. To the best of our knowledge, this is the first demonstration of controlling a humanoid robot with a learned torque model using tactile and force/torque sensing in presence of dynamic contacts.

II. Inverse Dynamics

Without contacts with the environment, the inverse dynamics (ID) of a robot with m degrees of freedom can be generally described by

$$\boldsymbol{\tau} = \underbrace{\boldsymbol{M}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}} + \boldsymbol{h}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)}_{\boldsymbol{\tau}_{RBD}} + \epsilon\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}\right), \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^m$, $\dot{\boldsymbol{q}} \in \mathbb{R}^m$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^m$ are the joint positions, velocities and accelerations, respectively, $\boldsymbol{M}\left(\boldsymbol{q}\right) \in \mathbb{R}^{m \times m}$ is the inertia matrix. $\boldsymbol{h}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) \in \mathbb{R}^{m \times m}$ is the matrix combining the contributions from Coriolis and centripetal, friction (viscous and static) and gravity forces:

$$\boldsymbol{h}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + F_v\dot{\boldsymbol{q}} + F_s\,\mathrm{sgn}(\dot{\boldsymbol{q}})\,. \tag{2}$$

The term $\epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ captures the errors of the model, such as unmodeled dynamics (e.g., elasticities and Stribeck friction), inaccuracies in the dynamic parameters (e.g., masses, inertia), vibrations, couplings and sensor noise.

In presence of a set of contacts $\mathcal{C} = \{c_1 \ldots c_n\}$ between the robot and the environment, Equation (1) becomes

$$\boldsymbol{\tau} = \underbrace{\boldsymbol{M}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}} + \boldsymbol{h}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) + \epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})}_{\boldsymbol{\tau}_{ID}} + \underbrace{\sum_{c_i \in \mathcal{C}} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\,\boldsymbol{\gamma}_i}_{\boldsymbol{\tau}_{CM}}\,, \tag{3}$$

by including $\boldsymbol{\tau}_{CM}$ to accounts for the effect of the external wrenches (forces and moments) $\boldsymbol{\gamma}_i$ applied at the contact location c_i, and $\boldsymbol{J}_{c_i}(\boldsymbol{q})$ is the contact Jacobian[2].

The joint torques $\boldsymbol{\tau}_{\mathrm{RBD}}$ can be computed analytically through the rigid body dynamics model of the robot, a standard parametric description of the robot [17], with known or identified kinematics and dynamics parameters [9], [10]. A method to estimate joint torques and external forces integrating force/torque and tactile measurements was proposed in [16]. The term $\epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is often neglected, or modeled and estimated, or implicitly taken into account by considering a perturbation in the dynamics parameters of $\boldsymbol{\tau}_{\mathrm{RBD}}$.

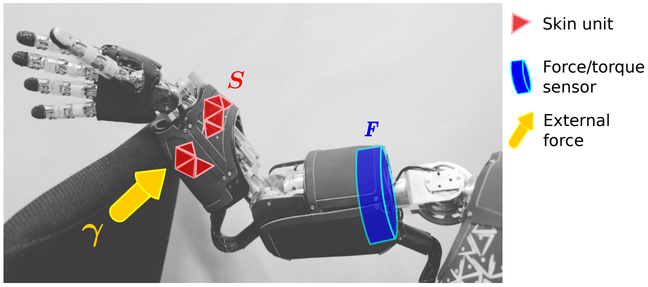

Fig. 2: Illustration of the force/torque and tactile sensors involved during a contact of the robot arm with the environment.

As discussed in [12], model-based approaches for computing the inverse dynamics have some limitations: for example, the required prior calibration of the robot, the difficulty in modeling accurately complex nonlinear dynamics and in generating suitable exciting trajectories for the parameters identification. By contrast, machine learning methods offer an alternative, data-driven approach for learning a robot dynamics model, without the need to describe it with complex parametric models [18], [19], [20]. Relying on collected robot data, learning the robot dynamics can be performed offline [12] or incrementally and online [11]. Without the need for compensating inaccurate dynamics parameters and accumulated errors, a learned dynamics model can improve the tracking and control performance of a robot, as shown in [18] for an industrial manipulator. In this paper, we make a step forward and show that learning the robot dynamics can also improve tracking and control performances of a humanoid arm in presence of contacts.

III. Control with Tactile Sensing

In this section, we describe our approach for learning ID models with contacts. We first formalize the problem as learning a mixture-of-experts model. Then we detail how to implement Gaussian processes as the corresponding experts.

A. Learning a Mixture-of-Contacts

When learning the inverse dynamics with contacts (Equation (3)), we assume that the (contact-free) inverse dynamics $\boldsymbol{\tau}_{ID}$ can be computed precisely, either from an analytic model or from a learned model [18]. As a result, only the model of the additional term of the contact forces $\boldsymbol{\tau}_{CM}$ has to be learned. In this paper, we consider a robot that is provided with skin measurements $\boldsymbol{s}$ from the tactile sensors, force measurements $\boldsymbol{F}$ from the force torque sensors (FTS). A visual representation of these relevant components is shown in Figure 2. Predicting the contact forces $\boldsymbol{\tau}_{CM}$ can be formalized as the regression task

$$\boldsymbol{y} = f(\boldsymbol{x}) + \epsilon\,, \tag{4}$$

where $\boldsymbol{y} = \boldsymbol{\tau}_{CM} = \sum_{i \in \mathcal{C}} \boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\gamma}_i$ and $\boldsymbol{x} = [\boldsymbol{q}, \boldsymbol{s}, \boldsymbol{F}]$ are the inputs. Additionally, ϵ is an i.i.d. Gaussian measurement

[2] The contact location c_i is not necessarily fixed, as the contacts may occur on the whole robotic structure and not exclusively at the end-effectors. If the contact location is not known a priori, it must be estimated, typically, through distributed tactile sensors. To compute the contact Jacobian $\boldsymbol{J}_{c_i}(\boldsymbol{q})$, we need the position of the contact point with respect to the reference frame of the link, which requires a kinematic/spatial calibration of the skin [15].

noise with mean 0 and variance σ_n^2. Therefore, our regression problem becomes

$$\boldsymbol{\tau}_{CM} = \sum_{i \in \mathcal{C}} \boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\gamma}_i = f([\boldsymbol{q}, \boldsymbol{s}, \boldsymbol{F}]) + \epsilon\,. \tag{5}$$

It is necessary to consider the skin $\boldsymbol{s}$ as an input since contacts with different parts of the body lead to different effects in the dynamics. Intuitively, $\boldsymbol{s}$ is required to identify the position of the contact. The force/torque measurements $\boldsymbol{F}$ could be avoided if we were interested in learning contacts that do not change between training and test time, which would restrict us to static objects, such as a rigid floor, walls or stationary obstacles. However, as this assumption is limiting in our context, we include the force/torque measurements $\boldsymbol{F}$ in our model.

The resulting regression of Equation (5) is challenging due to the high dimensionality of the input $\boldsymbol{x} \in \mathcal{X}$ (the skin measurements $\boldsymbol{s}$ alone account for hundreds of dimensions) and nonlinearity. Observing that the contact forces term $\boldsymbol{\tau}_{CM}$ is a sum over multiple contacts, we can tackle this challenging regression by decomposing the $\boldsymbol{\tau}_{CM}$ into a sum of multiple regression problems, where each regression problem is a single contact. Since each contact is uniquely defined by its position, we can make contact torque contribution of each single contact model independent of the skin, and use the skin only to decide which contact model is currently active. As a result, we can rephrase the regression of Equation (5) as the learning of the mixture-of-experts model ("mixture of contacts" in our case)

$$\boldsymbol{\tau}_{CM} = \sum_{i \in \mathcal{C}} \boldsymbol{J}_i(\boldsymbol{q})^T \boldsymbol{\gamma}_i = \sum_{j \in \mathcal{J}} f_j([\boldsymbol{q}, \boldsymbol{F}]) + \epsilon\,, \tag{6}$$

where $\mathcal{J}$ is the set of active experts f_j identified by the gating network. The purpose of the gating network is to select the experts that are currently active, based on the skin input $\boldsymbol{s}$, and to add their contributions. An illustration of this approach is shown in Figure 3. Note that in Equation 6 the skin input $\boldsymbol{s}$ is no longer explicitly part of the inputs of the experts, since it is now only used in the gating network. Hence, each single expert f_j is now sufficiently low-dimensional to be modeled independently, but at the same time the possibility of summing the contribution of each contact allows to account for complex behaviors. For expert f_j we use Gaussian processes (GP) mapping $[\boldsymbol{q}, \boldsymbol{F}] \mapsto \boldsymbol{J}_j(\boldsymbol{q})^T \boldsymbol{\gamma}_j$. Detailed information regarding GP models and their training are given in the next subsection. The gating network can be considered as a classifier $\mathcal{J} = g(\boldsymbol{q}, \boldsymbol{s}, \boldsymbol{F})$ that selects which contact is currently active. For simple tasks, this gating network can be designed using heuristics (e.g., using thresholds on the activation of the tactile sensors). Alternatively, for more complex systems an approach based on machine learning is more suitable (e.g., by learning a classifier using support vector machine).

Another important issue is chosing the number of contact models to use. Generally, the more contact models the better, as each one would specialize in its own particular contact class (i.e., contacts that results is similar effects on the dynamics) allowing for better generalization capabilities. However, it is important to notice that even within a single contact model, there are limited generalization capabilities as demonstrated in Section IV-D.

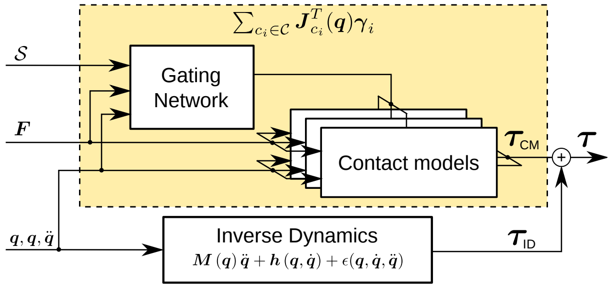
$\sum_{c_i \in \mathcal{C}} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{\gamma}_i$
$\mathcal{S}$
Gating Network
$\boldsymbol{F}$
Contact models
$\boldsymbol{\tau}_{\text{CM}}$
$\boldsymbol{\tau}$
+
$\boldsymbol{q}, \boldsymbol{q}, \ddot{\boldsymbol{q}}$
Inverse Dynamics
$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \epsilon(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$
$\boldsymbol{\tau}_{\text{ID}}$

Fig. 3: Our approach extends existing inverse dynamics without contacts by learning many contact models which serve as correction terms under different contacts type. The decision of which contact model to activate is taken by a gating network based on the skin measurements $\boldsymbol{s}$, the force torque sensors $\boldsymbol{F}$ and the current state $\boldsymbol{q}, \dot{\boldsymbol{q}}$.

B. Gaussian Processes as Expert Models

Gaussian Processes [21] are a state-of-the-art regression method. They have been used in robotics to learn dynamics models [20] and for control [22]. In the context of this paper, a GP is a distribution over inverse dynamics models

$$f \sim \mathcal{GP}\left(m_f, k_f\right)\,, \tag{7}$$

fully defined by a prior mean m_f and a covariance function k_f. When learn the contact models $\boldsymbol{\tau}_{CM}$, we choose as prior mean $m_f \equiv \boldsymbol{\tau}_{ID}$ and as covariance function k_f the squared exponential with automatic relevance determination and Gaussian noise:

$$k(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_f^2 \exp\left(-\tfrac{1}{2}(\boldsymbol{x}_p - \boldsymbol{x}_q)^T \boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_p - \boldsymbol{x}_q)\right) + \sigma_w^2 \delta_{pq} \tag{8}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}([l_1^2, ..., l_D^2])$ and δ_{pq} is the Kronecker delta (which is one if $p = q$ and zero otherwise). Here, l_i are the characteristic length-scales, σ_f^2 is the variance of the latent function $f(\cdot)$ and σ_w^2 the noise variance. In our experiments, when learning contact models, the input is defined as $\boldsymbol{X} = [\boldsymbol{q}, \boldsymbol{F}]$ and the output (observations) is $\boldsymbol{y} = \boldsymbol{\tau}$ are the torques. Hence, given n training inputs $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$ and corresponding training targets $\boldsymbol{y} = [y_1, ..., y_n]$, we define the training data set $\mathbb{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$. Training the GP corresponds to finding good hyperparameters $\boldsymbol{\theta} = [l_i, \sigma_f, \sigma_w]$, which can be done by the standard procedure of maximizing the marginal likelihood [21].

The GP yields the predictive distribution over torques for a new input $\boldsymbol{x}_* = [\boldsymbol{q}_*, \boldsymbol{F}_*]$

$$p(\boldsymbol{y}|\mathbb{D}, \boldsymbol{x}_*) = \mathcal{N}\left(\mu(\boldsymbol{x}_*), \sigma^2(\boldsymbol{x}_*)\right)\,, \tag{9}$$

where the mean $\mu(\boldsymbol{x}_*)$ and the variance $\sigma^2(\boldsymbol{x}_*)$ are

$$\mu(\boldsymbol{x}_*) = \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{y}\,, \quad \sigma^2(\boldsymbol{x}_*) = k_{**} - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_*\,. \tag{10}$$

The entries of the matrix $\boldsymbol{K}$ are $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and we define $k_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ and $\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{x}_*)$.

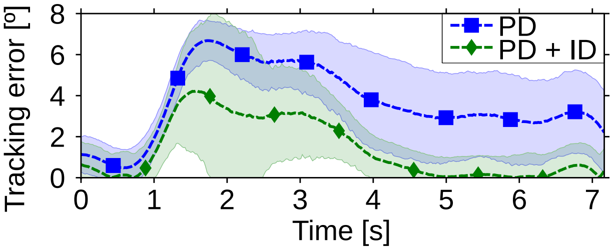

Fig. 4: Average tracking error in the absence of obstacles. Using the learned ID plus PD (green curve) reduce the error compared to using the PD controller only (blue curve). We conclude that the learned ID model is meaningful.

Controller	Shoulder Pitch	Shoulder Roll	Shoulder Jaw	Elbow
PD	0.72	4.64	3.68	2.37
PD + ID	**0.46**	**2.74**	**1.38**	**0.59**

TABLE I: Average tracking error (in degrees) in absence of obstacle. The use of the learned ID proved to be beneficial in reducing the error.

C. Control in presence of Contacts

In absence of contacts $\mathcal{C} = \{0\}$, and with a perfect model, we can define the task-space nonlinear feedforward control:

$$\boldsymbol{u} = \boldsymbol{\tau}_{ID}\,, \tag{11}$$

where the $\boldsymbol{u}$ are the torques given as motor command and $\boldsymbol{\tau}_{ID}$ are the torques computed from the inverse dynamics (or a learned model of it). To compensate for any noise or inaccuracies in the feed-forward dynamics, an additional feedback controller (impedance controller or PD controller) is added, such that

$$\boldsymbol{u} = \boldsymbol{\tau}_{ID} + \underbrace{K_P\left(\boldsymbol{q}^{\text{des}} - \boldsymbol{q}\right) + K_D\left(\dot{\boldsymbol{q}}^{\text{des}} - \dot{\boldsymbol{q}}\right)}_{\boldsymbol{\tau}_{PD}}\,. \tag{12}$$

Intuitively, the magnitude of the control signals contributed from the PD controller $\boldsymbol{\tau}_{PD}$ can be used to measure the goodness of the inverse dynamics model. Accurate inverse dynamics models will only need small corrections by the feedback controller during tracking of a desired trajectory, while inaccurate models will rely more heavily on it. In case of inaccurate models increasing the PD gains can still lead to acceptable tracking performance, but often at the expense of safety and compliance.

In case of contacts, we can further extend the controller of Equation (12) to become

$$\boldsymbol{u} = \boldsymbol{\tau}_{ID} + \boldsymbol{\tau}_{PD} + \boldsymbol{\tau}_{CM}\,, \tag{13}$$

where $\boldsymbol{\tau}_{CM}$ is the contribution from the contacts (see Equation (3)), which is estimated using the learned model from Equation (6). In the experimental evaluations presented in the next section we will compare the performance of all these three control schema.

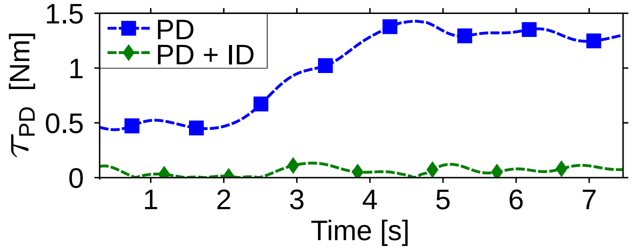

Fig. 5: Average torque contributed by the PD term $\boldsymbol{\tau}_{PD}$ in absence of obstacle (filtered for visualization purposes). $\boldsymbol{\tau}_{PD}$ drastically decrease when using the learned ID, therefore suggesting that the learned ID model is accurate.

IV. Experimental Results

In this section we present the experimental results obtained from a tracking task in the presence of contacts. First we demonstrate that an accurate (contact-free) ID model can be learned. This ID model will be used as base for comparing the tracking performance in presence of obstacles. Second we demonstrate that our proposed model can be used to compensate for an unknown obstacle and achieve better tracking performance. To conclude, we demonstrate the generalization capabilities of our learned model by testing it on a novel obstacle.

A. Experimental Setting

The experimental evaluations are performed on the *iCub-Darmstadt01*, the most recent version (v. 2.5) of *iCub* [23]. The *iCub* has 53 degrees of freedom and is 104 cm tall for 24 kg of weight. The robot is not equipped with joint torque sensors, but it has four 6-axis force/torque sensors placed proximally in the middle of legs and arms, and an artificial skin consisting of more than 3000 tactile sensors, mounted on the robot covers [24]. Each of these sensors provide a signal in the interval $\in [0, 255]$ correlated to the force applied orthogonally to the sensor. In our experiments, we control 4 DoF of the *iCub* arm: shoulder pitch, roll and jaw, and elbow. The skin input $\boldsymbol{s}$ from the forearm consists of 384 sensors. In the *iCub* the joint torques are not estimated by the motors currents, but via an analytic model with identified dynamics parameters [9], which updates the estimation thanks to the inertial, tactile and force/torque measurements [16]. We compute, on a PC outside the robot, the desired joint torques at the frequency of 50 Hz and then send the commands to the robot interface *Yarp*. A low-level controller implemented on the motor boards takes care of tracking the desired joint torques, sent through the robot interface. Both the estimation of the joint torques and the low-lever controller are transparent to our learned model as we use the desired torques (instead of the measured torques) as targets $\boldsymbol{y}$ during the training of the models.

B. Validation of the (Contact-free) Inverse Dynamics

Before learning the contact models, a pre-requisite of our approach is the existence of a (contact-free) rigid-body inverse dynamics model. In our case, we decided to learn such

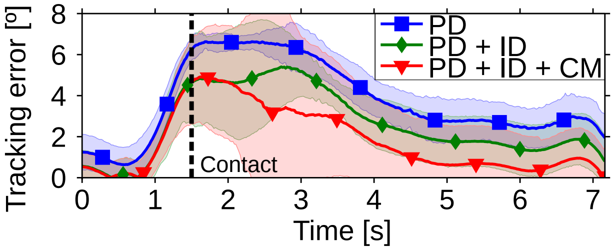

Fig. 6: Average tracking error in presence of obstacle. Our model based on learned inverse dynamics and contact models (red curve) outperformed both simple PD controller (blue curve) and PD + ID model (green curve).

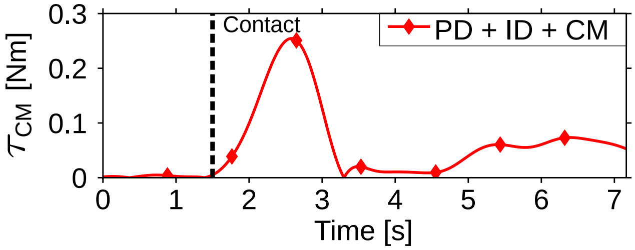

Fig. 7: Average torque contributed by the contact model term $\boldsymbol{\tau}_{CM}$ in presence of obstacle (filtered for visualization purposes).

Controller	Shoulder Pitch	Shoulder Roll	Shoulder Jaw	Elbow
PD	0.50	4.67	3.78	2.04
PD + ID	0.49	3.96	2.63	**0.25**
PD + ID + CM	**0.46**	**3.34**	**1.81**	0.46

TABLE II: Average tracking error (in degrees) in presence of obstacle.

ID model from scratch using Gaussian Processes. The reason for this choice are the unmodeled dynamics $\epsilon\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}\right)$, which introduce substantial errors even in absence of contacts. In this subsection we present an experimental validation of the learned (contact-free) inverse dynamics. To validate the learned ID model we firstly compared the tracking error, in absence of obstacles. The task used for the experimental validation of the proposed approach consists of a tracking task with the left arm of the *iCub* where each of the four joints of the arm should follow a pre-designed trajectory. In Table I we collected the tracking errors obtained averaging over 20 repetitions. The results shows that the learned ID model reduces the tracking error, compared to the simple PD controller. The visualization of Figure 4 shows the average error tracking for the shoulder jaw. As second comparison we analyzed the contribution to the desired torque from the PD controller $\boldsymbol{\tau}_{PD}$. In Figure 5 it is visible that when using the learned ID model, $\boldsymbol{\tau}_{PD}$ is drastically reduced. Both experimental results suggests that the learned ID model is accurate and useful for the desired tracking task.

C. Trajectory Tracking in Presence of Obstacles

For the validation of the inverse dynamics with contacts model we consider the same trajectory tracking task used in the previous evaluation. However, in this experiments we include the presence of a dynamic obstacle, specifically a filled 1-liter bottle of water. For classical controllers, when an obstruction occur, the rigid body inverse dynamics does not account for this variation. As a result, the tracking error increases and with it also the contribution of the PD feedback controller increases to compensate for this tracking error. In this scenario, we demonstrate that using a learned contact model improves the tracking accuracy when unforeseen and dynamic obstacles are encountered along the path.

We first performed the tracking experiment (with the obstacle) once using as controller the PD + ID, in order to collect training data. Using these data collected, we trained the contact model using as target the difference between the desired torque and the inverse dynamics model contribution $\boldsymbol{\tau}_{ID}$ (we here assume a single contact model). Following, we repeated the experiment comparing the full controller PD + ID + CM, the simple PD, and PD + ID. In Table II are collected the tracking errors for the three control schema and for all the four joint, averaged over 20 repetitions. The presence of the contact model proved beneficial in terms of tracking error for three out of four joints. For the elbow, the PD +ID performed slightly better. However, it is important to notice that the joints that are most involved in the trajectory are the shoulder pitch and shoulder jaw. In Figure 6 is shows the average tracking error (for the shoulder jaw) for the considered control modes. It is clearly visible that after about 1 second from the contact with the obstacle, the contact model reduce the tracking error. From these results we can conclude that using the additional learned contact model is beneficial. The profile of the average torque contributed by $\boldsymbol{\tau}_{CM}$ is shown in Figure 7.

One limitation of the current results is that the gating network does not always recognize the presence of the contact. This is due to hardware limitations of the *iCub* skin. In fact, the current skin presents two main issues: 1) high levels of noise. Even without contacts, some sensors will occasionally fire as if in presence of strong forces applied. 2) Mechanical compliance of the skin. When applying a force, The sum of these two issues resulted in the introduction of delays for the identification of contacts and therefore the use of the additional contact model. This delay, which we estimate around 500ms is visible in both Figure 6 and Figure 7. Often even for a human expert it is hard to distinguish between measurement noise and a real contact when observing short (i.e., less than 100 ms) windows of data. A second issue we encountered during the training of both inverse dynamics model and contact models, is that the torques applied from the basic PD controller implemented on the *iCub* suffer from a high noise. As a result of this high degree of noise, we had to filter the desired torques. However, filtering the desired torques inevitably introduce small errors in the models learned. We expect that a smoother desired torques would result in a more accurate learned model and even lower tracking error.

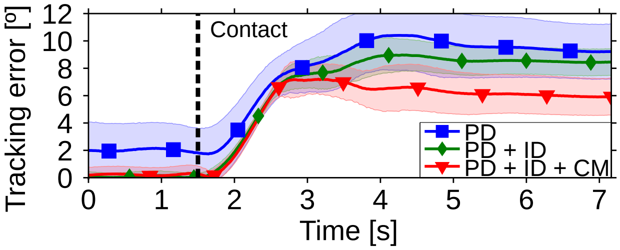

Fig. 8: Average tracking error in presence of obstacle. Our model based on learned inverse dynamics and contact models (red curve) outperformed both simple PD controller (blue curve) and PD + ID model (green curve).

Controller	Shoulder Pitch	Shoulder Roll	Shoulder Jaw	Elbow
PD	**0.60**	6.95	4.97	1.74
PD + ID	0.72	5.69	3.90	**0.37**
PD + ID + CM	0.67	**4.46**	**3.13**	0.59

TABLE III: Average tracking error (in degrees) in presence of an obstacle never observed before.

D. Generalization of the Contact Model

To analyze the generalization capabilities of a single contact model we experimented the use of obstacles not included in the training data. In a similar manner to the previous experiment we collected data using the PD + ID on two different obstacles: the 1-liter bottle of water and a bin filled with 2 liters of water. Using the data collected from both these obstacles we trained the contact model. We tested the resulting learned model on a new obstacle: a bin filled with 1.3 liters of water. Table III report the average tracking error over 25 experiments. It can be noticed that for the two joints with the highest usage during the task (i.e., shoulder roll and jaw), the contact model considerably improve the tracking accuracy (respectively of 2.5° and 1.8°). A visualization of the tracking error for the shoulder roll is shown in Figure 8. However, for the other two joints we observed a small ($< 0.25°$) decrease of the tracking performance. Overall, these results suggest that the contact model is capable of some degree of generalization to novel obstacles with different properties (e.g., weight).

V. Discussion & Conclusions

In this paper, we experimentally evaluated on the humanoid robot *iCub* a data-driven approach for learning inverse dynamics models in presence of contacts. Our proposed solution enables a fast and accurate prediction of the joint torques in situations when the robot is in contact with obstacles, trough the use of the tactile skin.

We demonstrate on a pushing task that our learned model substantially improve the tracking performance in presence of unknown obstacles. The estimation from the learned model does not rely on dynamic parameters, but it is completely data-driven: uncalibrated tactile sensors provide raw information about the contact, while force/torque sensors provide information about the wrenches perceived by the robotic structure. As a result, our approach does not require a spatially calibrated model of the skin. We believe that this feature is particularly relevant with the increasing availability of miniaturized skin sensors. For future robot skin, composed of thousands or millions of sensors, it would be extremely challenging and time consuming to perform a full spatial calibration. Our approach is a first step toward the use of learning methods to alleviate this problem.

References

[1] A. Jain, M. Killpack, A. Edsinger, and C. Kemp, “Reaching in clutter with whole-arm tactile sensing,” *IJRR*, vol. 32, no. 4, pp. 458–482, 2013.

[2] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth, “Bayesian gait optimization for bipedal locomotion,” in *LION8*, 2014.

[3] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, “iCub whole-body control through force regulation on rigid noncoplanar contacts,” *Frontiers in Robotics and AI*, vol. 2, no. 6, 2015.

[4] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[5] T. Erez and E. Todorov, “Trajectory optimization for domains with contacts using inverse dynamics,” in *IROS*, 2012.

[6] S. Ivaldi, M. Fumagalli, F. Nori, M. Baglietto, G. Metta, and G. Sandini, “Approximate optimal control for reaching and trajectory planning in a humanoid robot,” in *IROS*, 2010, pp. 1290–1296.

[7] A. Ibanez, P. Bidaud, and V. Padois, “Emergence of humanoid walking behaviors from mixed-integer model predictive control,” in *IROS*, 2014, pp. 4014–4021.

[8] A. Del Prete, F. Nori, G. Metta, and L. Natale, “Control of contact forces: The role of tactile feedback for contact localization,” in *IROS*, 2012.

[9] S. Traversaro, A. Del Prete, S. Ivaldi, and F. Nori, “Inertial parameters identification and joint torques estimation with proximal force/torque sensing,” in *ICRA*, 2015.

[10] Y. Ogawa, G. Venture, and C. Ott, “Dynamic parameters identification of a humanoid robot using joint torque sensors and/or contact forces,” in *HUMANOIDS*, 2014.

[11] L. Jamone, B. Damas, and J. Santos-Victor, “Incremental learning of context-dependent dynamic internal models for robot control,” in *ISIC*. IEEE, 2014, pp. 1336–1341.

[12] R. Calandra, S. Ivaldi, M. Deisenroth, E. Rueckert, and J. Peters, “Learning inverse dynamics models with contacts,” in *ICRA*, 2015.

[13] D. Nguyen-Tuong, J. Peters, and M. Seeger, “Local Gaussian process regression for real time online model learning,” in *NIPS*, 2008.

[14] K. Yamane, “Practical kinematic and dynamic calibration methods for force-controlled humanoid robots,” in *HUMANOIDS*, 2011.

[15] A. Del Prete, S. Denei, L. Natale, F. Mastrogiovanni, F. Nori, G. Cannata, and G. Metta, “Skin spatial calibration using force/torque measurements,” in *IROS*, 2011.

[16] S. Ivaldi, M. Fumagalli, M. Randazzo, F. Nori, G. Metta, and G. Sandini, “Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors: theory and implementation on the iCub,” in *HUMANOIDS*, 2011.

[17] R. Featherstone and D. Orin, “Dynamics,” in *Springer Handbook of Robotics*, 2008, pp. 35–65.

[18] D. Nguyen-Tuong and J. Peters, “Model learning for robot control: a survey,” *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[19] S. Vijayakumar and S. Schaal, “Locally weighted projection regression: Incremental real time learning in high dimensional space,” in *ICML*, 2000.

[20] M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters, “Toward fast policy search for learning legged locomotion,” in *IROS*, 2012.

[21] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[22] M. Deisenroth, D. Fox, and C. Rasmussen, “Gaussian Processes for Data-Efficient Learning in Robotics and Control,” *IEEE TPAMI*, vol. 37, no. 2, pp. 408–423, 2015.

[23] L. Natale, F. Nori, G. Metta, M. Fumagalli, S. Ivaldi, U. Pattacini, M. Randazzo, A. Schmitz, and G. G. Sandini, “The iCub platform: a tool for studying intrinsically motivated learning,” in *Intrinsically motivated learning in natural and artificial systems*, 2013.

[24] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, “An embedded artificial skin for humanoid robots,” in *MFI*, 2008.