

# Learning Inverse Dynamics Models with Contacts

Roberto Calandra<sup>1</sup>, Serena Ivaldi<sup>1,2</sup>, Marc Peter Deisenroth<sup>3</sup>, Elmar Rueckert<sup>1</sup>, Jan Peters<sup>1,4</sup>

**Abstract**—In whole-body control, joint torques and external forces need to be estimated accurately. In principle, this can be done through pervasive joint-torque sensing and accurate system identification. However, these sensors are expensive and may not be integrated in all links. Moreover, the exact position of the contact must be known for a precise estimation. If contacts occur on the whole body, tactile sensors can estimate the contact location, but this requires a kinematic spatial calibration, which is prone to errors. Accumulating errors may have dramatic effects on the system identification. As an alternative to classical model-based approaches we propose a data-driven mixture-of-experts learning approach using Gaussian processes. This model predicts joint torques directly from raw data of tactile and force/torque sensors. We compare our approach to an analytical model-based approach on real world data recorded from the humanoid *iCub*. We show that the learned model accurately predicts the joint torques resulting from contact forces, is robust to changes in the environment and outperforms existing dynamic models that use of force/torque sensor data.

## I. INTRODUCTION

A key challenge for torque-controlled humanoid robots is to accurately estimate their dynamics in presence of contacts, e.g., during manipulation in clutter [13], whole-body movements [14] or ground contacts in locomotion [1]. Analytic dynamics models suffer from inaccurate parameter estimation, unmodeled dynamics (e.g., friction, couplings, elasticities) and noisy sensor measurements. With contacts the problem is even more challenging due to discontinuities and additional non-linearities, which are difficult to model or estimate. Moreover, if contact locations are not fixed a priori or known with sufficient precision, small errors in the localization of the external force can substantially deteriorate the inverse dynamics computation [7].

Nevertheless, many modern control strategies like inverse dynamics control [8], computed torque control [22] or model predictive control [16] rely on accurate dynamic models. With inaccurate dynamics models they can produce suboptimal policies by not taking external forces into account, which are caused by contacts.

As a first step toward a more informed controller that explicitly considers the effect of contacts, we propose

to learn the inverse dynamics model from tactile sensor readings and force-torque sensors. In contrast to classical techniques based on the identification of dynamics parameters [26], [19], [24], we propose a fully data-driven machine learning approach based on non-parametric models, where both the rigid body dynamics as well as the effect of external forces on the robot structure are learned directly from data collected on the real robot. The proposed model makes use of the raw sensor data and does not require a kinematic/dynamics calibration [26], [19], [24]. In particular, it does not need a spatially calibrated model of the skin [6]. We propose to use a mixtures-of-experts based on Gaussian Processes (GP) to learn the non-linear system dynamics. Each of these GP experts models a single contact “type” and can be learned straightforwardly. By using a gating network that activates and deactivates the individual GP experts we can switch between contact models and generalize to more complex environments. We evaluate our model learning approach on the arm of the *iCub* humanoid robot [15] (see Fig. 1) and compare to a state-of-the-art model-based approach. The learned inverse dynamics model outperforms the analytic approach and we demonstrate that the learned model can generalize to changing contact locations. To the best of our knowledge this is the first demonstration of how joint torques can be learned on a humanoid robot equipped with tactile and force/torque sensors in presence of contacts.

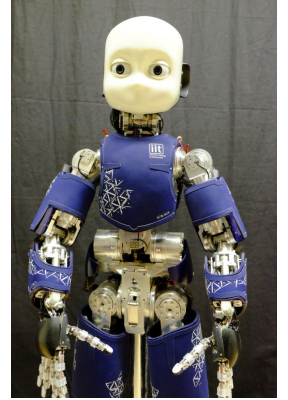


Fig. 1: The humanoid robot *iCub* used in the experiments.

## II. PROBLEM FORMULATION

The inverse dynamics of a robot with  $m$  degrees of freedom can be generally described by

$$\tau = \underbrace{M(q)\ddot{q} + h(q, \dot{q})}_{\tau_{\text{RBD}}} + \epsilon(q, \dot{q}, \ddot{q}), \quad (1)$$

where  $q$ ,  $\dot{q}$  and  $\ddot{q}$  are the joint positions, velocities and accelerations, respectively,  $M(q)$  is the inertia matrix and

$$h(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) + F_v\dot{q} + F_s \text{sgn}(\dot{q})$$

is the matrix combining the contributions from Coriolis and centripetal, friction (viscous and static) and gravity forces. The term  $\epsilon(q, \dot{q}, \ddot{q})$  in (1) captures the errors of the model, such as unmodeled dynamics (e.g., elasticities

\*The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007–2013) under grant agreements #270327 (CompLACS) and #600716 (CoDyCo). MPD has been supported by an Imperial College Junior Research Fellowship.

We thank V. Padois and A. Droniou for their help with *iCubParis02*.

<sup>1</sup>Intelligent Autonomous Systems, TU Darmstadt, Darmstadt, Germany  
<sup>2</sup>Inria, Villers-lès-Nancy, F-54600, France; CNRS, Loria, UMR n.7503 and Université de Lorraine, Vandoeuvre-lès-Nancy, F-54500, France

<sup>3</sup>Department of Computing, Imperial College London, London, UK

<sup>4</sup>Max Planck Institute for Intelligent Systems, Tübingen, Germany

and Stribeck friction), inaccuracies in the dynamic parameters (e.g., masses, inertia), vibrations, couplings, and sensor noise. With a set  $\mathcal{C} = \{c_1 \dots c_n\}$  of contacts  $c_i$  between the robot and the environment, (1) becomes

$$\tau = \underbrace{M(q)\ddot{q} + h(q, \dot{q})}_{\tau_{\text{RBD}}} + \epsilon(q, \dot{q}, \ddot{q}) + \sum_{c_i \in \mathcal{C}} J_{c_i}^\top(q) \gamma_i, \quad (2)$$

where the last term accounts for the additive effect of the external wrenches (forces and moments)  $\gamma_i$  applied at contact location  $c_i$ , and  $J_{c_i}(q)$  is the contact Jacobian<sup>1</sup>.

#### A. Classical model-based approaches for computing the robot dynamics

Classical approaches for computing  $\tau$  or  $\tau_{\text{RBD}}$  rely on the dynamics model with known or identified kinematics and dynamics parameters [12]. The torques  $\tau_{\text{RBD}} = M(q)\ddot{q} + h(q, \dot{q})$  can be computed analytically through the rigid body dynamics model of the robot, a standard parametric description of the robot [9]. The term  $\epsilon(q, \dot{q}, \ddot{q})$  is often neglected, or implicitly taken into account by considering a perturbation in the dynamics parameters of  $\tau_{\text{RBD}}$ , which need to be identified accurately.

Although parameter identification for industrial robots is relatively easy with exciting trajectories [20], the procedure for floating-base robots, such as humanoids, is not straightforward because of two main issues: 1) The generation of sufficiently large accelerations for the identification while maintaining the robot balance and the control of contacts. This issue was well explained by Yamane [26], who proposed a technique to identify the mass and the local COM of the links in a humanoid robot with fixed feet at the ground and slow joint trajectories. 2) The measurement of the external forces  $\gamma_i$  exerted on the robot. Note that it may not be straightforward to measure the external forces  $\gamma_i$  as it is not possible to cover the robot body with 6-axis force/torque sensors to measure the force exerted on every possible contact location  $c_i$ . Usually, such sensors are big, heavy and expensive. Thus, they are carefully placed where the external forces are critical for the main tasks, for example at the end-effectors for manipulation and at the feet for balancing. In such a case, it is possible to identify the dynamics parameters while balancing and walking without additional contacts [19]. When force/torque sensors are placed proximally, such as in the *iCub* arms [10], some of the dynamics parameters can be identified, but in absence of contacts [24].

When multiple contacts are exerted on the robot structure at locations other than the classical end-effectors, it is still possible to compute a precise inverse dynamics model, but this requires both pervasive joint torque sensing, such as in *Toro* [19], and additional force/torque and tactile sensing,

<sup>1</sup>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. In such a case, the contact location, if not known a priori, must be estimated, typically through distributed tactile sensors. To compute the contact Jacobian, we need the position of the contact point with respect to the reference frame of the link [10]. Such a knowledge requires a kinematic calibration of the skin as explained in [6].

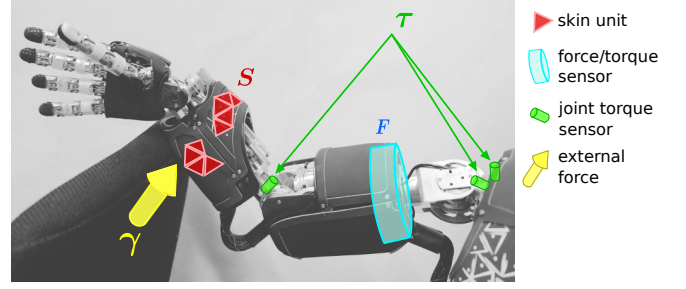


Fig. 2: Illustration of the force/torque and tactile sensors during a contact of the robot arm with the environment.

such as in *iCub* [11]. Moreover, it requires the precise knowledge of the contact locations detected by the tactile sensors, which necessitates a spatial calibration of the skin [6]. This procedure is prone to errors, and it has been shown that small errors in the kinematics calibration of the taxels (i.e., the tactile units) can induce non-negligible errors in the estimation of the contact forces [7].

Generally, these model-based approaches have three main limitations: 1) It is hard to add details about couplings, elasticity, friction and other nonlinear dynamics, which are required for high accuracy; 2) The performance of the data-driven identification strongly depends on the experimental setting (with/without contacts) and the exciting trajectories [20]; 3) They make strong assumptions to handle contacts.

#### B. Learning the inverse dynamics

An alternative and appealing approach to analytic dynamics computation is to use machine learning methods to learn the dynamics model of a robot [18], [25], [5]. Without the need for compensating for inaccurate dynamics parameters and accumulated errors, a learned dynamics model can improve the tracking and control performances of a robot, as shown in [17] for an industrial manipulator. The clear advantage of learning the inverse dynamics is that we can overcome the limitations of the aforementioned approaches: difficulty in modeling complex nonlinear dynamics, impossibility to generate suitable exciting trajectories, restrictive assumptions regarding contacts and sensors, prior accurate kinematics calibration of the tactile sensors.

**Tricky statement. It depends on your model.**

Despite the success of learned dynamics models in robotics, to the best of our knowledge there are no examples in the literature where dynamic contacts are also learned. The inclusion of dynamic contact models in the dynamics highlights two main problems: First, switching from a no-contact model to a contact-model requires to observe the system state and to model a discontinuous function [23]. Second, switching between different contacts  $c_i \in \mathcal{C}$  must be properly handled.

Here, we provide a first formulation to this problem, and we show that it is possible to learn the inverse dynamics model of the arm of the *iCub* robot by means of proximal force/torque measurements  $F$  and distributed tactile

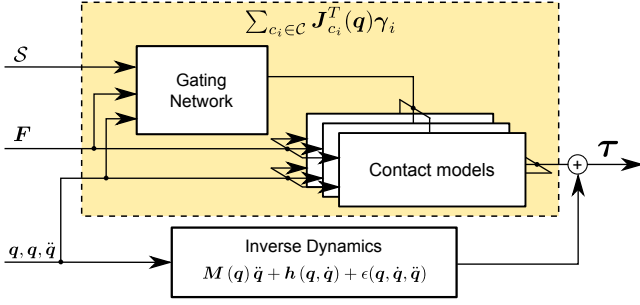


Fig. 3: Our approach extends existing inverse dynamics without contacts by learning many contact models, which serve as correction terms under different contact types. The decision of which contact model to activate is made by a gating network, which uses skin measurements  $S$ , the force torque sensors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

sensors  $S$  (without requiring a spatially calibrated model of the skin [6]).

### III. LEARNING INVERSE DYNAMICS WITH CONTACTS

In this section, we present our proposed approach to learning inverse dynamics with contacts. We first formalize the problem as learning a mixture-of-experts model. Then we detail how we implement Gaussian processes as the corresponding experts.

#### A. Learning contacts as a mixture-of-experts

When learning inverse dynamics with contacts (2), we assume that the (contact-free) inverse dynamics from (1) can be computed precisely, either from an analytical model or from a learned model [17]. In our experiments, we employ a learned GP model as contact-free inverse dynamics. The reason for this choice are the unmodeled dynamics  $\epsilon(q, \dot{q}, \ddot{q})$ , which introduce substantial errors even without contacts. As a result of the pre-existing contact-free inverse dynamics, only the model of the residual term of the external forces  $\sum_{c_i \in \mathcal{C}} J_{c_i}^\top(q) \gamma_i$  has to be separately learned. In this paper, we consider a robot that is provided with skin measurements  $S$  from the tactile sensors, force measurements  $F$  from the force torque sensors (FTS) and the ground truth of the torques  $\tau$  from the joint torque sensors (JTS). An illustration of these relevant components is shown in Fig. 2. Modeling the external forces  $\sum_{c_i \in \mathcal{C}} J_{c_i}^\top(q) \gamma_i$  can be formalized as the regression task

$$y = f([q, S]) + w, \quad (3)$$

where  $y = \sum_{c_i \in \mathcal{C}} J_{c_i}^\top(q) \gamma_i$  and  $w$  is an i.i.d. Gaussian measurement noise with mean 0 and variance  $\sigma_w^2$ . Contacts with different parts of the body lead to different effects in the dynamics. Intuitively, it is necessary to consider the skin input  $S$  to identify the position of the contact. Additionally, measurements of the force applied by the contacts are necessary to deal with a non-static environment.

Rephrase sentence above. It's unclear.

Theoretically, these measurements can be provided by the skin. However, the artificial skin used in our experiments

does not provide a precise six-dimensional measure of the contact force. Therefore, in the implementation of our model we substitute the force measurement from the skin with the force/torque measurements  $F$ . The corresponding regression problem (3) is complicated due to the high-dimensional space of the input  $x \in \mathcal{X}$  (the skin measurements  $S$  alone account for hundreds of dimensions). Therefore, we rephrase this regression task as a problem of learning a mixture-of-experts model. With this model, we decompose (3) as

$$\sum_{c_i \in \mathcal{C}} J_{c_i}^\top(q) \gamma_i = \sum_{j \in \mathcal{J}} f_j([q, F]) + w, \quad (4)$$

where  $\mathcal{J}$  is the set of active experts  $f_j$ .

What is an active expert? Unclear.

Note that the skin input  $S$  is no longer explicitly part of the inputs of the experts. Therefore, each single expert  $f_j$  is now sufficiently low-dimensional to be modeled independently. At the same time the possibility of summing the contribution of each contact allows to account for complex behaviors. As single expert  $f_j$  we use Gaussian processes for the mapping  $[q, F] \mapsto J_j^\top(q) \gamma_j$ . A gating network is used to select the experts that are currently active and to add their contributions. An illustration of our approach is shown in Fig. 3. In this paper, we implement this gating network as a multi-class classifier  $\mathcal{J} = g(q, S, F)$  that selects which contact is currently ongoing.

Why?

I don't like this word

In a general setting, the gating network does not make hard decisions. Please make this clear. This would probably also change eq. (4) since you would need weights.

For simple tasks, this gating network can be designed using heuristics (e.g., using thresholds on the activation of the tactile sensors). However, for more complex systems an adaptive, data-driven approach may be more suitable. In the experimental section we evaluate the learning of such gating network.

Strange end of this section. Shall we delete it? It only adds confusion.

#### B. Gaussian processes as expert models

Gaussian Processes (GPs) [21] are a state-of-the-art regression method. They have been used in robotics to learn dynamics models [5] and for control [4]. In this paper, a GP is a distribution over inverse dynamics models  $f \sim \mathcal{GP}(m_f, k_f)$ , fully defined by a prior mean  $m_f$  and a covariance function  $k_f$ . We choose as prior mean  $m_f \equiv \tau_{\text{RBD}}$  and as covariance function  $k_f$  the squared exponential with automatic relevance determination and Gaussian noise

$k(x_p, x_q) = \sigma_f^2 \exp(-\frac{1}{2}(x_p - x_q)^T \Lambda^{-1}(x_p - x_q)) + \sigma_w^2 \delta_{pq}$  where  $\Lambda = \text{diag}([l_1^2, \dots, l_D^2])$  and  $\delta_{pq}$  is the Kronecker delta (which is one if  $p = q$  and zero otherwise). Here,  $l_i$  are the length-scales,  $\sigma_f^2$  is the variance of the latent function  $f(\cdot)$  and  $\sigma_w^2$  the noise variance.

In our experiments, when learning contact models, the input is defined as  $x = [q, F]$ , while the output (observations)  $y = \tau$  are the torques. Hence, given  $n$  training



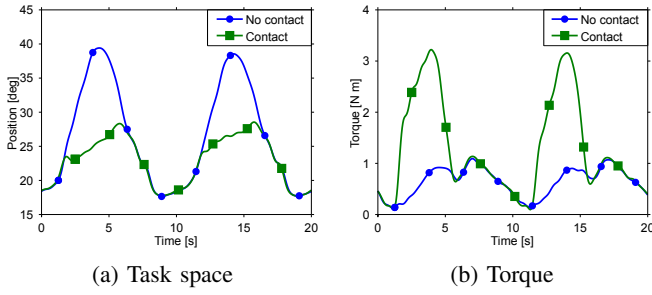


Fig. 4: **Learning a single contact:** Effects of a contact (green curve) compared to the free movement without obstacle (blue curve). These effects are visible in the task space position (a) and in the torque measured by the joint torque sensor (b).

inputs  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  and corresponding training targets  $\mathbf{y} = [y_1, \dots, y_n]$ , we define the training data set  $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$ . Training the GP corresponds to finding good hyperparameters  $\theta = [l_i, \sigma_f, \sigma_w]$ , which is done by the standard procedure of maximizing the marginal likelihood [21].

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

$$p(\mathbf{y}|\mathbb{D}, \mathbf{x}_*, \theta) = \mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*)) , \quad (5)$$

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

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}, \quad \sigma^2(\mathbf{x}_*) = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*, \quad (6)$$

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

It remains unclear how you use the GPs as experts in the MoE model. How do you determine the corresponding training data sets? What is the number of experts? And you also need to explain the gating network much better.

#### IV. EXPERIMENTAL SET-UP AND EVALUATION

In this section, we describe the experimental setting and the humanoid robot *iCub* used in the experiments. We present four different experiments where we demonstrate that 1) Our approach can learn single contact models; 2) A single learned model (i.e., an expert) is robust to small changes in the position of the contact; 3) Our approach extends to multiple contacts by combining models of single contacts; 4) The gating network activating the experts can be learned to reduce the complexity of manually design it.

##### A. Experimental set-up

The experiments were conducted on the *iCub* [15], a humanoid robot with 53 degrees of freedom, sized as a child (104 cm tall, 24 kg of weight). This robot is equipped with several sensors: an inertial sensor in the head, four 6-axis force/torque sensors placed proximally in the middle of legs and arms, and an artificial skin consisting of many distributed tactile elements (taxels) mounted on the robot covers [2]. The information from these three types of sensors is used to estimate the joint torques and the external contact forces by the *iDyn* library [11]. In the following,  $\tau_{\text{IDYN}}$  denotes the joint torques estimated by the *iDyn* library, which we use

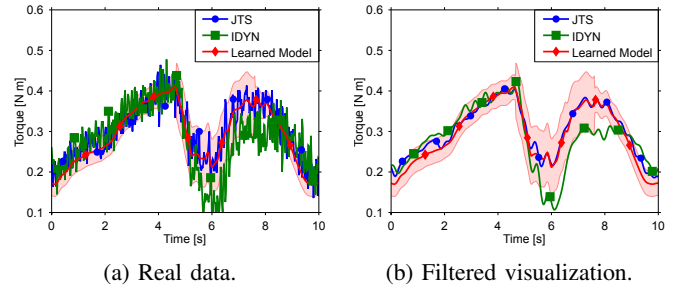


Fig. 5: **Learning a single contact:** Comparison of the torque measured at the elbow (with contact) by the JTS, estimated by *iDyn* and our learned model (shown as mean  $\pm 2$  std). Our learned model better predict the torque measured by JTS (a). Additionally, due to the identification of the noise in the model, its prediction is smoother compared to both the noisy JTS measurements and the prediction from *iDyn*. For visualization purposes we also show the predictions when filtering JTS and *iDyn* (b).

	Method	Shoulder 1 [Nm]	Shoulder 2 [Nm]	Elbow [Nm]
Full trajectory	<i>iDyn</i>	$0.09 \pm 1.1 \times 10^{-3}$	$0.16 \pm 1.8 \times 10^{-3}$	$0.05 \pm 7.4 \times 10^{-4}$
	Our model	$0.04 \pm 5.6 \times 10^{-4}$	$0.07 \pm 9.8 \times 10^{-4}$	$0.02 \pm 3.1 \times 10^{-4}$
Contact only	<i>iDyn</i>	$0.07 \pm 3.1 \times 10^{-3}$	$0.13 \pm 5.7 \times 10^{-3}$	$0.08 \pm 3.0 \times 10^{-3}$
	Our model	$0.03 \pm 1.5 \times 10^{-3}$	$0.12 \pm 5.9 \times 10^{-3}$	$0.03 \pm 1.3 \times 10^{-3}$

TABLE I: **Learning a single contact:** Mean and standard deviation of the mean for the RMSE of the test set for ground truth, predictions with the *iDyn* and our learned model. The learned model predicts the torque more accurately than *iDyn* for both the full trajectory and during the only contact phase.

as analytical model for comparison. For more detail on its contact detection and taxels calibration we refer to [6], [7].

The *iCub* used in the experiments is equipped with three additional Joint Torque Sensors (JTSs), two in the shoulder and one in the elbow. The JTSs are calibrated by computing the offset and gain through least-square regression with respect to the output of *iDyn*. We consider these calibrated JTSs as ground truth measurements of the joint torques  $\tau$ . In our experiments, we used the *iCub* torso and arms (3 and 7 degrees of freedom, respectively) and the skin input  $\mathbf{S}$  from the forearm, which consists of 270 sensor measurements.

##### B. Learning a single contact

In this experiment, we consider the *iCub* making contact with a single obstacle. The evaluation is performed on a simple tracking task with the *iCub*'s end-effector moving along a circular trajectory. We repeat the task twice: first without any contact and then with a contact at a fixed position. Fig. 4 shows the effects of the contact in terms of position and torque during the tracking task. When the contact occurs the position error increases considerably. As a result, the torque is increased to compensate for the obstacle. We collected 10 repetitions of the trajectory with the contact and used 8 of them to train the model. The remaining trajectories are used as test set to evaluate the predictive performances of our learned model. For this experiment we consider a single expert (the gating network still decides whether to activate the expert).

We compare the baseline joint torque (measured by the

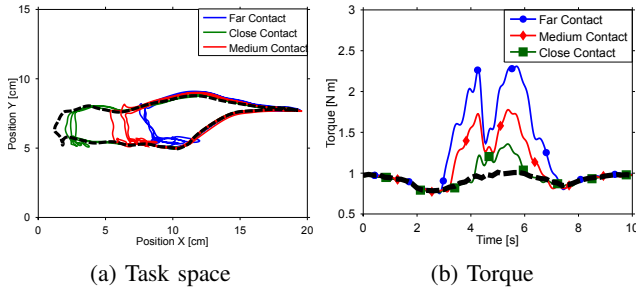


Fig. 6: **Robustness of the single contact model:** Effects of the contact on the task space and the torque for the three different contact types: **contact 1 (far)**, **contact 2 (medium)** and **contact 3 (close)**. The task in absence of contact is displayed as reference (**black dashed curve**).

JTS) to the joint torque estimated by the analytic model  $iDyn$  and the joint torque  $\tau_{IDM}$  predicted by our learned model. In Table I, we report the root mean square error (RMSE) and the standard deviation of the mean of  $iDyn$  and our learned model for all the three joints. Additionally, we report both the error of the learned models (learned RBD plus learned contact model) during the full trajectory and exclusively *during* the contact. In five out of six cases, the learned model performs better than the analytic model. In the sixth case (contact only, shoulder 2), the performance of the learned model is similar to the analytic model. However, increasing the amount of data used for training may further increase the performance of the learned model. A visual representation of the predictions of the test set for the elbow joints is shown in Fig. 5.

This experiment provides evidence that the classical rigid-body dynamics model  $\tau_{RBD}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  and the  $iDyn$  estimation (that also exploits proximal force/torque sensing) fail to accurately estimate the joint torques when the robot is in contact with the environment. Moreover, we show that the learned contact model, when combined with the RBD model, provides a better approximation of the joint torque.

### C. Robustness of the single contact model

In the following, we show that the prediction performance of each GP expert is robust to small variations in the position of the contact. This is important since the exact position of the obstacle does not need to be known in advance (within a single expert  $f_j$ ). As in the previous experiment we consider a tracking task along a circular trajectory. However, this time the obstacle is placed at one of three different positions along the trajectory: close, medium and far. Each of these obstacles is shifted 2 cm along the horizontal axis. Obstacles at different positions along the trajectory lead to different effects in terms of both joint position and torque signal, as clearly visible in Fig. 6. Note that the skin input  $\mathbf{S}$  will also be affected, as shown in Fig. 7. Hence, we could potentially learn a separate expert for each contact. However, we only consider a single expert as we want to demonstrate the robustness of a single expert, not of the gating network.

The contact model is learned using the data collected from contact 1 and contact 3 (far and close contacts) and as

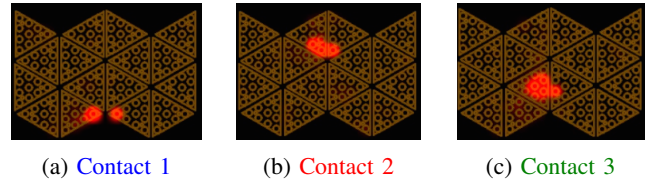


Fig. 7: **Robustness of the single contact model.** The different contact locations detected by the forearm skin respectively for the three contacts: **contact 1 (far)**, **contact 2 (medium)** and **contact 3 (close)**.

	Method	Shoulder 1 [Nm]	Shoulder 2 [Nm]	Elbow [Nm]
Far contact	$iDyn$	$0.13 \pm 3.9 \times 10^{-3}$	$0.40 \pm 9.7 \times 10^{-3}$	$0.06 \pm 1.9 \times 10^{-3}$
	Our model	<b><math>0.06 \pm 1.9 \times 10^{-3}</math></b>	<b><math>0.08 \pm 2.9 \times 10^{-3}</math></b>	<b><math>0.03 \pm 8.0 \times 10^{-4}</math></b>
Close contact	$iDyn$	$0.09 \pm 2.2 \times 10^{-3}$	$0.22 \pm 4.5 \times 10^{-3}$	$0.04 \pm 0.9 \times 10^{-3}$
	Our model	<b><math>0.06 \pm 1.4 \times 10^{-3}</math></b>	<b><math>0.06 \pm 1.4 \times 10^{-3}</math></b>	<b><math>0.02 \pm 6.3 \times 10^{-4}</math></b>
Medium contact	$iDyn$	$0.10 \pm 2.8 \times 10^{-3}$	$0.32 \pm 6.7 \times 10^{-3}$	$0.05 \pm 1.3 \times 10^{-3}$
	Our model	<b><math>0.06 \pm 1.7 \times 10^{-3}</math></b>	<b><math>0.12 \pm 4.7 \times 10^{-3}</math></b>	<b><math>0.05 \pm 1.7 \times 10^{-3}</math></b>

TABLE II: **Robustness of the single contact model:** Errors between the ground truth (JTS) and the predictions with either the  $iDyn$  and our learned model on the test set. A single expert is robust to small variations of the contact.

validation the data set generated from the *unseen* contact 2 (medium) is used. In Table II, the RMSE for all three contacts are reported for  $iDyn$  and our learned model, respectively. The results show that the learned model is robust to unseen contacts and performs equally well or better than the analytic model  $iDyn$ .

### D. Learning multiple contacts

After learning single contacts, we now show how to combine the learned models to adapt to unseen and more complex environments with multiple contacts. We consider a scenario having the *iCub* performing a circular motion with its left arm. We initially performed two experiments with an obstacle either on the left and on the right of the reference trajectory (see Fig. 8). With the data collected in these two contact cases, we trained two independent expert models  $f_1$ ,  $f_2$ , one for each contact. We repeated the experiment, but this time with both left and right contacts and used this last unseen case to validate our models. Fig. 9 shows an example of the prediction and the corresponding activation of the two contact models. During both the right and the left contact, the corresponding experts are activated by the gating network. Therefore, we can successfully combine the contributions of the single contact models learned to generalize to unseen cases with multiple contacts. Table III reports the RMSE for the predictions. We notice that even in this experiment the experts accurately learn the effects of single contacts. Moreover, the gating network allows us to combine the experts to generalize to unseen environments, such as in the case of both contacts.

### E. Learning the gating network

So far, we assumed a heuristic gating network to select the active experts. In this experiment we show that a learned gating network achieves a comparable accuracy as a manually devised heuristic. As ground truth to evaluate the performances, as well as for training the classifier, we labeled the data with one of the following labels: no contact, left

	Method	Shoulder 1 [Nm]	Shoulder 2 [Nm]	Elbow [Nm]
Right contact	<i>iDyn</i>	$0.10 \pm 1.3 \times 10^{-3}$	$0.13 \pm 1.6 \times 10^{-3}$	$0.06 \pm 8.1 \times 10^{-4}$
	Our model	$0.04 \pm 6.3 \times 10^{-4}$	$0.07 \pm 1.2 \times 10^{-3}$	$0.02 \pm 2.7 \times 10^{-4}$
Left contact	<i>iDyn</i>	$0.08 \pm 1.2 \times 10^{-3}$	$0.16 \pm 2.0 \times 10^{-3}$	$0.05 \pm 8.2 \times 10^{-4}$
	Our model	$0.03 \pm 5.7 \times 10^{-4}$	$0.07 \pm 9.6 \times 10^{-4}$	$0.02 \pm 2.8 \times 10^{-4}$
Both contacts	<i>iDyn</i>	$0.10 \pm 1.3 \times 10^{-3}$	$0.11 \pm 1.4 \times 10^{-3}$	$0.07 \pm 8.4 \times 10^{-4}$
	Our model	$0.05 \pm 8.3 \times 10^{-4}$	$0.10 \pm 1.6 \times 10^{-3}$	$0.03 \pm 4.0 \times 10^{-4}$

TABLE III: **Learning multiple contacts:** Root mean square error between the ground truth (JTS) and the predictions with the *iDyn* and our learned model on the test set. Our learned model predicts the torque more accurately than *iDyn*.

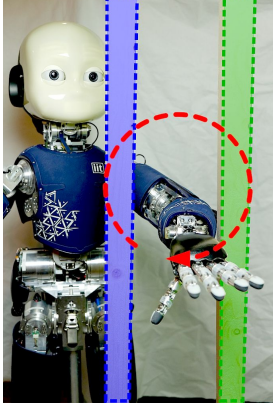


Fig. 8: **Learning multiple contacts:** The robot performs a circle with its left arm. The forearm collides alternatively with the left, the right or both contacts.

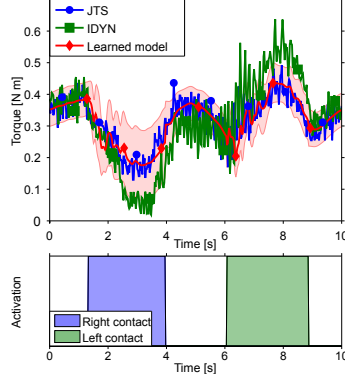


Fig. 9: **Learning multiple contacts:** Prediction of torques with multiple contacts and the corresponding activation of the gating network. Our mixture-of-experts model combines single-contact models to a multiple-contact model.

contact, right contact. The heuristic is based on thresholds of the activation of the skin input  $S$  and the force torque sensors  $F$ . We train a Support Vector Machine (SVM) classifier (using the library LIBSVM [3]) having as input  $q, S, F$  and as output the contact labels (none, left, right).

We evaluated the performance of the trained classifier on an unseen test set. Fig. 10 shows that the learned SVM achieved a classification accuracy that is similar to the heuristic gating network. Equivalent results are obtained in terms of RMSE of the inverse dynamics when comparing the experts models learned by the gating networks. However, training the gating network (i.e., training the SVM classifier) requires considerably less expert knowledge compared to designing a heuristic. As there is no visible performance difference, we conclude that training the gating network is generally preferable. Increasing the number of training data may further increase the accuracy of the gating network.

## V. CONCLUSIONS

Whole-body control strategies that exploit contacts need accurate models of the system dynamics. This is crucial for balance and stabilization, and to increase the number of potential actions that the robot is able to execute, e.g., creating a contact to reach for distant objects. We introduced a data-driven mixture-of-experts approach based on Gaussian processes for learning inverse dynamics models with contacts. We evaluated our model on the *iCub* humanoid

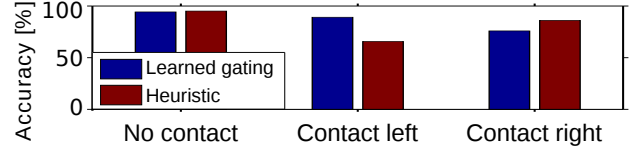


Fig. 10: **Learning the gating network:** Classification accuracy for the heuristic and learned SVM gating networks.

robot using tactile sensors and force/torque sensors as model inputs. We showed that the model accurately predicts contact forces and outperforms a state-of-the-art analytical approach used to estimate the joint torques in *iCub*. The estimation from the learned model does not rely on dynamic parameters, but it is completely data-driven and based on tactile sensors and force/torque sensors. As a result, our approach does not require a spatially calibrated model of the skin [6], [7]. This is a promising feature for robust control strategies that explicitly takes contacts into account.

## REFERENCES

- [1] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth. Bayesian gait optimization for bipedal locomotion. In *LION8*, 2014.
- [2] G. Cannata, M. Maggiali, G. Metta, and G. Sandini. An embedded artificial skin for humanoid robots. In *MFI*, 2008.
- [3] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27, 2011.
- [4] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian Processes for Data-Efficient Learning in Robotics and Control. *IEEE TPAMI*, 37(2):408–423, 2015.
- [5] M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters. Toward fast policy search for learning legged locomotion. In *IROS*, 2012.
- [6] A. Del Prete, S. Denei, L. Natale, F. Mastrogiorganni, F. Nori, G. Cannata, and G. Metta. Skin spatial calibration using force/torque measurements. In *IROS*, 2011.
- [7] 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.
- [8] T. Erez and E. Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *IROS*, 2012.
- [9] R. Featherstone and D. Orin. Dynamics. In *Springer Handbook of Robotics*, pages 35–65. 2008.
- [10] M. Fumagalli, S. Ivaldi, M. Randazzo, L. Natale, G. Metta, G. Sandini, and F. Nori. Force feedback exploiting tactile and proximal force/torque sensing - Theory and implementation on the humanoid robot *iCub*. *Autonomous Robots*, 33(4):381–398, 2012.
- [11] 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.
- [12] S. Ivaldi, J. Peters, V. Padois, and F. Nori. Tools for simulating humanoid robot dynamics: a survey based on user feedback. In *HUMANOIDS*, 2014.
- [13] A. Jain, M. Killpack, A. Edsinger, and C. Kemp. Reaching in clutter with whole-arm tactile sensing. *IJRR*, 32(4):458–482, 2013.
- [14] S. Kajita and B. Espiau. Legged robots. In *Handbook of Robotics*, pages 361–389. Springer, 2008.
- [15] 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.
- [16] M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse, and P. Soueres. METAPOD - Template META-PrOgramming applied to Dynamics: CoP-CoM trajectories filtering. In *HUMANOIDS*, 2014.
- [17] D. Nguyen-Tuong and J. Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.
- [18] D. Nguyen-Tuong, J. Peters, and M. Seeger. Local Gaussian process regression for real time online model learning. In *NIPS*, 2008.

- [19] 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.
- [20] N. Pedrocchi, E. Villagrossi, F. Vicentini, and L. Tosatti. Robot-dynamic calibration improvement by local identification. In *ICRA*, 2014.
- [21] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [22] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009.
- [23] M. Toussaint and S. Vijayakumar. Learning discontinuities with products-of-sigmoids for switching between local models. In *ICML*, 2005.
- [24] S. Traversaro, A. Del Prete, R. Muradore, L. Natale, and F. Nori. Inertial parameter identification including friction and motor dynamics. *HUMANOIDS*, 2013.
- [25] S. Vijayakumar and S. Schaal. Locally weighted projection regression: Incremental real time learning in high dimensional space. In *ICML*, 2000.
- [26] K. Yamane. Practical kinematic and dynamic calibration methods for force-controlled humanoid robots. In *HUMANOIDS*, 2011.