2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

# FOP Networks for Learning Humanoid Body Schema and Dynamics

Fernando Díaz Ledezma and Sami Haddadin[1]

*Abstract*— **Robot inverse dynamics modeling is performed mainly via standard system identification and/or machine learning techniques. In this paper we part from the theoretical framework of First-Order Principles Networks (FOPnet), combining data-aided learning with basic knowledge to learn the model of a targeted robot. The framework, previously used for learning the dynamics of a fixed-base serial manipulator, is now extended to the learning of the kinematics and dynamics of tree-structured robots with floating base. Our approach leverages the principle of compositionality to separate the main problem into two partially independent modules. The first defines the robot's body schema by characterizing its morphology and topology. The second is dependent upon the latter and defines the inertial properties of the multi-body system. To demonstrate the capabilities of the approach, a simulated humanoid robot with 30 degrees of freedom is used. We discuss the implementation of our method and evaluate its estimation and generalization capabilities in comparison with other common machine learning approaches. Finally, we present experimental results on a 7-DoF manipulator.**

## I. Introduction

### A. Robot model learning: state of the art

Control of robotic systems requires models that closely describe their inherent dynamics. To find such models two approaches are typically used. The first is system identification, which estimates the robot's inertial parameters using a previously calibrated kinematic model as well as joint torque and position measurements [17]. Similarly, online estimation methods have been devised to estimate the inertial parameters; with adaptive control being a well known example [16]. In contrast, the second approach attempts to find models of robots solely from data. To date, neural Networks (NN) are the quintessential representative in this area, where especially the feedforward [3] and recurrent [13] variants have provided good local approximations in the field of robotics. Other approaches include, e.g., locally weighted projection regression, support vector regression and Gaussian processes regression, as well as learning by demonstration [1], [10], [19], [20]. They all require rather large amounts of data and learn so far only manifolds of the input-output space.

In contrast to the identification of fixed-base robots, the process for humanoid robots is not as standardized yet. For instance, in [8] the static inertial parameters are found without force/torque measurements based on the free-floating base link, an inertial measurement unit and joint encoders measurements. In [21], experimental results on the identification of physically consistent kinematic and dynamic parameters of force-controlled biped humanoid robots with insufficient dynamics excitation are discussed. Authors in [18] introduce a theoretical framework for the inertial parameter identification from force/torque measurements. The work in [4] discusses an incremental semi-parametric inverse dynamics learning for robots based on a mixture of parametric rigid body dynamics models and incremental kernel methods. Likewise, in [11] classical joint-torque-based identification and the base-link approach are discussed for the identification of a humanoid robot. Furthermore, [2] presents a method to estimate the inertial parameters of free-floating systems in the absence of contact forces. Finally, [9] describes a framework based on the nullspace of the unmeasured degrees of freedom for whole-body inertial parameter estimation using only joint torque. Unlike the previous examples, the method we introduce here is suited for kinematic trees with and without floating bases.

A common denominator in robot identification is the assumption that the kinematic structure and geometry are known. Lately, researchers have looked at the kinematic model of a robot as a form of *body schema* and have attempted to vest robots with the capability of learning it; see for instance [6] for a comprehensive review of the concept of body schema and its application in robotics. In our work, learning the body schema will be a key element in the overall definition of the robot model.

### B. Contributions

In this work we introduce a method that aims to answer the following research questions:

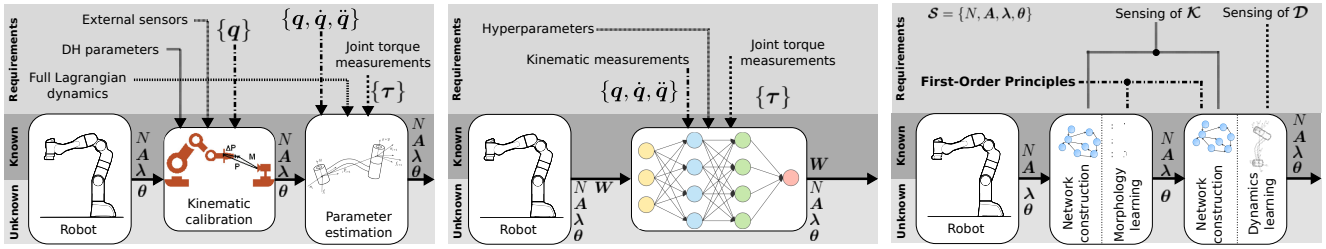
- Which measurements are required to fully automate robot kinematics and inverse dynamics learning based on knowing only the adjacency graph along with kinematic and dynamic first-order principles?
- How to transform robot system identification or end-to-end learning with meta parameter guessing into an automated learning scheme that determines both the structure and dynamical properties of the robot with minimal information?

Unlike classical system identification, we show that given only an appropriate set of proprioceptive measurements not only the inertial properties but also the morphology of the robot can be learned. In doing this, we extend the FOPnet framework introduced in [5], where *first-order principles* (FOP) distributed in a network of learning modules are used to learn the inverse dynamics of a manipulator, to the estimation of the body schema and inertial properties of general tree-structured robots with revolute joints. We demonstrate how this estimation can be separated into two partially interdependent parameter learning modules, while the network topology is constructed automatically[1]. The first is a purely kinematics-dependent problem which, based on a given robot topology, is used to learn its geometrical properties; namely, the body schema. The second learning module corresponds to the estimation of the inertial properties of

[1]Both authors are with the Chair of Robotics Science and Systems Intelligence and Munich School of Robotics and Machine Intelligence, Technische Universität München, 80797 München, Germany {fernando.diaz, haddadin}@tum.de

[1]Please note that this paper does not cover that network construction algorithm as this would go beyond the scope of the paper. A full treatment of this problem is left for future work.

978-1-5386-7282-2/18/$31.00 ©2018 IEEE




(a) (b) (c)

Fig. 1: Different schemes to model a robot: (a) classical inverse dynamics identification, (b) an example of the machine learning approach (here neural networks) to learn robot inverse dynamics, (c) our proposed FOPnet-based approach.

the now characterized multi-body system. Furthermore, we evaluate the estimation and generalization capabilities of our framework and contrast it to three purely data-based learning methods: a feedforward NN, a committee of Extreme Learning Machines and Gaussian Process Regression. We evaluate the results with a 30 degrees of freedom (DoF) humanoid and also show experimental results with a 7 DoF *Franka Emika Panda* robot manipulator.

The remainder of this paper is organized as follows. In Section II, the proposed approach for learning the self of tree structured robots is outlined. Section III addresses the proposed parameter learning scheme. Simulation and experimental results are discussed in Sec. IV, using a simulated 30-DoF ATLAS humanoid robot to compare the FOPnet performance versus that of two selected machine learning approaches. Results from an experiment on a 7 DoF manipulator are also presented in this section. Finally, conclusions are drawn in Sec. V.

## II. Learning the Self for Floating-Base Tree-Structured Robots

### A. Preliminaries

First, we define the *self* as a set

$$\mathcal{S} = \{N, \boldsymbol{A}, \boldsymbol{\lambda}, \boldsymbol{\theta}\}, \tag{1}$$

whose elements are the number of degrees of freedom (i.e. links) $N$, the adjacency matrix $\boldsymbol{A}$ of the robot defining the connectivity of the kinematic tree, the morphological description of every link in the robot $\boldsymbol{\lambda}$, and the inertial parameters of the links $\boldsymbol{\theta}$.

Figure 1 shows how elements of $\mathcal{S}$ are found in three different inverse dynamics identification/learning frameworks. Fig. 1a shows that classical system identification assumes a known calibrated kinematic model (i.e., $N$, $\boldsymbol{A}$, $\boldsymbol{\lambda}$) and mostly focuses on learning $\boldsymbol{\theta}$. In contrast, most machine learning schemes (Fig. 1b) attempt to estimate the model introducing a new set of to-be-learned parameters $\boldsymbol{W}$ that lack correlation to the elements of $\mathcal{S}$. Our approach, Fig. 1c, considers that only $N$ and $\boldsymbol{A}$ are known initially as well as what *proprioceptive* signals should be available to turn classical system identification into a learning problem, without requiring *exteroceptive* signals for further calibration. The nature of these signals enables the learning and is a direct consequence of domain-specific first-order principles. In our case they are (a) a set $\mathcal{K}$ of kinematics-related measurements and (b) a set $\mathcal{D}$ of dynamics-related measurements. From here on, the robot's body schema is identified in a first learning module exploiting $\mathcal{K}$ to learn the vector $\boldsymbol{\lambda}$, which determines the positions and orientations of the joint frames. Consequently, in the second learning module the dynamical properties $\boldsymbol{\theta}$ of the previously characterized body schema are learned, given $\mathcal{D}$. After the completion of these learning modules, an instantiated model of the robot has been found. Note that the respective kinematics and dynamics networks can be automatically created from $\boldsymbol{A}$ and the first-order principles only.

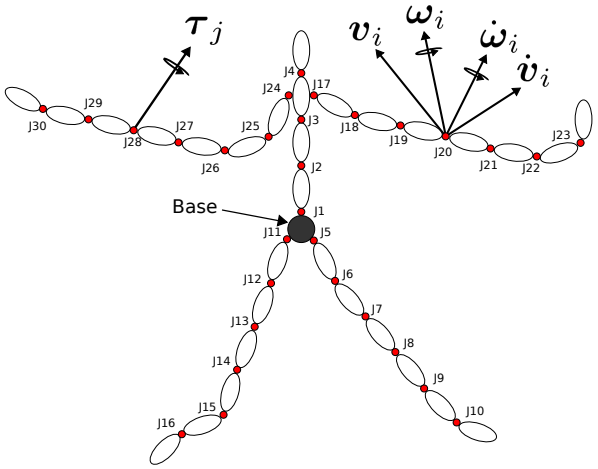



Fig. 2: General humanoid topology description including relevant measurements, joints are represented with red circles.

In the beginning, only the topology of the robot, and the relevant proprioceptive signals are available; see Fig. 2. To look for the missing elements of $\mathcal{S}$ we represent the inverse dynamics of the robot via a function parametrized by $\mathcal{S}$ of the form:

$$\boldsymbol{\tau} = f(\boldsymbol{r}, \dot{\boldsymbol{r}}, \ddot{\boldsymbol{r}}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}; \mathcal{S}), \tag{2}$$

where $\boldsymbol{r}$, $\dot{\boldsymbol{r}}$, $\ddot{\boldsymbol{r}} \in \mathbb{R}^6$ are the pose of the base link and its derivatives and $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}} \in \mathbb{R}^N$ refer to the joint angles, velocities and accelerations. Within $\mathcal{S}$, $\boldsymbol{\lambda} \in \mathbb{R}^{6N}$ is defined as

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1^\intercal & \boldsymbol{\lambda}_2^\intercal & \dots & \boldsymbol{\lambda}_j^\intercal & \dots & \boldsymbol{\lambda}_N^\intercal \end{bmatrix}^\intercal, \tag{3a}$$

$$\boldsymbol{\lambda}_j = \begin{bmatrix} \alpha_j & \beta_j & \gamma_j & {}^{(i)}\boldsymbol{p}_j{}^\intercal \end{bmatrix}^\intercal, \tag{3b}$$

where $\alpha_j$, $\beta_j$ and $\gamma_j$ are the Euler angles that determine the orientation of the $j$th-link's coordinate system and ${}^{(i)}\boldsymbol{p}_j$ is the vector that locates this frame relative to the predecessor

link $i$. Finally, $\boldsymbol{\theta} \in \mathbb{R}^{10N}$ is:

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1^\intercal & \boldsymbol{\theta}_2^\intercal & \dots & \boldsymbol{\theta}_j^\intercal & \dots & \boldsymbol{\theta}_N^\intercal \end{bmatrix}^\intercal, \tag{4a}$$

$$\boldsymbol{\theta}_j = \begin{bmatrix} m_j & mX_j & mY_j & mZ_j & XX_j & XY_j & XZ_j & YY_j & YZ_j & ZZ_j \end{bmatrix}^\intercal. \tag{4b}$$

The first element of $\boldsymbol{\theta}_j$ is the mass of link $j$, the next three elements are the first moments of mass around the coordinate system of link $j$ and the last six entries are the elements of the inertia tensor of link $j$, expressed in the frame $j$.

### B. First-order principles

The FOP defining the learning modules can be deduced from the well known Newton-Euler equations. We apply the principle of compositionality by decomposing the FOPnet into three sub-networks. The first models the robot's body schema and the other two model the robot dynamics; i.e.:

$$\boldsymbol{k} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}; N, \boldsymbol{A}, \boldsymbol{\lambda}), \tag{5a}$$

$$\boldsymbol{w} = g(\boldsymbol{q}, \boldsymbol{k}; N, \boldsymbol{A}, \boldsymbol{\lambda}, \boldsymbol{\theta}), \tag{5b}$$

$$\boldsymbol{\tau} = \boldsymbol{w} \circ \boldsymbol{k}. \tag{5c}$$

In (5a) $\boldsymbol{k}$ represents stacked Cartesian linear and angular velocities and accelerations, and (5b) and (5c) represent wrenches and torques, respectively.

### C. Characterizing the body schema

Learning $\boldsymbol{\lambda}$ assigns a specific morphology to the topology encoded in $\boldsymbol{A}$. To do this, we use the Cartesian kinematic measurements of the base link $\boldsymbol{k}_{base}$, joint positions, velocities and accelerations, as well as the twist vector for each link. Therefore, the set $\mathcal{K}$ is defined as

$$\mathcal{K} = \{\dot{\boldsymbol{v}}_{base}, \boldsymbol{\omega}_{base}, \dot{\boldsymbol{\omega}}_{base}, \boldsymbol{V}_1, \dots, \boldsymbol{V}_N, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}\}, \tag{6}$$

where

$$\boldsymbol{V}_j = \begin{bmatrix} \boldsymbol{v}_j^\intercal & \boldsymbol{\omega}_j^\intercal \end{bmatrix}^\intercal. \tag{7}$$

Learning happens as we try to reconcile the measurements $\mathcal{K}$ with the kinematic expressions $\boldsymbol{k} \in \mathbb{R}^{9N}$, whose elements are the Cartesian angular velocity $\boldsymbol{\omega}$, acceleration $\dot{\boldsymbol{\omega}}$ and linear velocity and acceleration $\boldsymbol{v}$ and $\dot{\boldsymbol{v}}$ for each link. For link $j$, with predecessor link $i$ determined by $\boldsymbol{A}$, the corresponding building blocks of (5a) are[2]:

$${}^{(j)}\boldsymbol{v}_j = {}^i_j\boldsymbol{R}^\intercal \Big\{ {}^{(i)}\boldsymbol{v}_i + {}^{(i)}\boldsymbol{\omega}_i \times {}^{(i)}\boldsymbol{p}_j \Big\}, \tag{8a}$$

$${}^{(j)}\boldsymbol{\omega}_j = {}^i_j\boldsymbol{R}^\intercal {}^{(i)}\boldsymbol{\omega}_i + \dot{q}_j \boldsymbol{z}_j, \tag{8b}$$

$${}^{(j)}\dot{\boldsymbol{v}}_j = {}^i_j\boldsymbol{R}^\intercal \Big\{ {}^{(i)}\dot{\boldsymbol{\omega}}_i \times {}^{(i)}\boldsymbol{p}_j + {}^{(i)}\boldsymbol{\omega}_i \times {}^{(i)}\boldsymbol{\omega}_i \times {}^{(i)}\boldsymbol{p}_j + {}^{(i)}\dot{\boldsymbol{v}}_i \Big\}, \tag{8c}$$

$${}^{(j)}\dot{\boldsymbol{\omega}}_j = {}^i_j\boldsymbol{R}^\intercal {}^{(i)}\dot{\boldsymbol{\omega}}_i + {}^i_j\boldsymbol{R}^\intercal {}^{(i)}\boldsymbol{\omega}_i \times \dot{q}_j \boldsymbol{z} + \ddot{q}_j \boldsymbol{z}_j. \tag{8d}$$

Here $\boldsymbol{z}_j$ is the unit z-vector of the $j$th coordinate frame. ${}^i_j\boldsymbol{R}$ denotes the rotation matrix expressing the orientation of frame $j$ in $i$ as a function of the joint angle $q_j$ and parametrized by $\boldsymbol{\lambda}_j$. It is composed of the multiplication of four basic rotation matrices around the respective $x$- and $z$-axes.

$${}^i_j\boldsymbol{R}(q_j; \boldsymbol{\lambda}_j) = \boldsymbol{R}_z(\alpha_j)\boldsymbol{R}_x(\beta_j)\boldsymbol{R}_z(\gamma_j)\boldsymbol{R}_z(q_j), \tag{9}$$

[2]Notation ${}^{(j)}\boldsymbol{\omega}_i$ represents a vector for link $i$ defined in frame $(j)$.

The *kinematics network* $\boldsymbol{k}$ consists of the following entries[3]:

$$\boldsymbol{k}_j(\boldsymbol{k}_i, q_j, \dot{q}_j, \ddot{q}_j; \boldsymbol{\lambda}_j) = \begin{bmatrix} \boldsymbol{v}_j^\intercal & \dot{\boldsymbol{v}}_j^\intercal & \boldsymbol{\omega}_j^\intercal & \dot{\boldsymbol{\omega}}_j^\intercal \end{bmatrix}^\intercal \tag{10a}$$

$$\boldsymbol{k} = \begin{bmatrix} \boldsymbol{k}_1^\intercal & \cdots & \boldsymbol{k}_j^\intercal & \cdots & \boldsymbol{k}_N^\intercal \end{bmatrix}^\intercal, \tag{10b}$$

with $\boldsymbol{k}_0 = \boldsymbol{k}_{base}$ being measurements from the base link.

### D. Representing the dynamics

The definition of $\mathcal{S}$ is completed by the determination of the dynamic parameters $\boldsymbol{\theta}$. To do this, we first look at the wrench that results from a link's own inertial properties[4]

$$\boldsymbol{w}_{j,j}(\boldsymbol{k}_j; \boldsymbol{\theta}_j) = \boldsymbol{W}_j \boldsymbol{\theta}_j. \tag{11}$$

$\boldsymbol{W}_j$ is defined as

$$\boldsymbol{W}_j = \begin{bmatrix} \dot{\boldsymbol{v}}_j & [\dot{\boldsymbol{\omega}}_j\times] + [\boldsymbol{\omega}_j\times][\boldsymbol{\omega}_j\times] & \boldsymbol{0}_{6\times 6} \\ \hline 0 & [-\dot{\boldsymbol{v}}_j\times] & [\bullet\dot{\boldsymbol{\omega}}_j] + [\boldsymbol{\omega}_j\times][\bullet\boldsymbol{\omega}_j] \end{bmatrix}, \tag{12}$$

where $[\cdot\times]$ is a skew-symmetric matrix made from the corresponding vector.

$$[\bullet\boldsymbol{\omega}] = \begin{bmatrix} \omega_x & \omega_y & \omega_x & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix}. \tag{13}$$

Consequently, the total wrench $\boldsymbol{w}_j$ acting on link $j$ results from the effects of $\boldsymbol{w}_{j,j}$ plus the effects of the wrenches $\boldsymbol{w}_{j,k}$ from its succeeding links, as depicted in eqs. (14).

$$\boldsymbol{T}_j(\boldsymbol{\lambda}_j, q_j) = \begin{bmatrix} {}^i_j\boldsymbol{R} & \boldsymbol{0}_{3\times 3} \\ \hline [\boldsymbol{p}_j(\boldsymbol{\lambda}_j)\times]{}^i_j\boldsymbol{R} & {}^i_j\boldsymbol{R} \end{bmatrix} \tag{14a}$$

$$\boldsymbol{w}_{j,k}(\boldsymbol{q}, \boldsymbol{w}_{k,k}; \boldsymbol{\lambda}, \boldsymbol{A}) = {}^j\boldsymbol{T}_{\boldsymbol{\nu}(1)}{}^{\boldsymbol{\nu}(1)}\boldsymbol{T}_{\boldsymbol{\nu}(2)} \cdots {}^{\boldsymbol{\nu}(\bar{N}-1)}\boldsymbol{T}_k \boldsymbol{w}_{k,k} \tag{14b}$$

$$\boldsymbol{w}_j(\boldsymbol{\omega}_{j,j}; \boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{i+1}, \dots, \boldsymbol{\lambda}_j) = \boldsymbol{w}_{j,j} + \sum_{k=j+1}^{P} \boldsymbol{w}_{j,k}. \tag{14c}$$

The number $P$ of succeeding links $k$ can be determined directly from the $\boldsymbol{A}$ matrix. Moreover, the vector $\boldsymbol{\nu} = f(j, k, \boldsymbol{A})$, with $\boldsymbol{\nu} \in \mathbb{R}^{\bar{N}}$, determines the arranged indices of all the $\bar{N}$ links between link $j$ and link $k$, with $\boldsymbol{\nu}(\bar{N}) = k$.

The corresponding elements of the *inter-dynamics network* are the wrenches acting on every link

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1^\intercal & \cdots & \boldsymbol{w}_j^\intercal & \cdots & \boldsymbol{w}_N^\intercal \end{bmatrix}^\intercal. \tag{15}$$

Finally, the outputs of the inverse dynamics network, for a serial chain, correspond to the joint torque vectors given by:

$$\boldsymbol{\tau} = \boldsymbol{C}\boldsymbol{w}, \tag{16}$$

with $\boldsymbol{C}$ being a matrix that selects the 6th element of the wrench vectors, i.e., the torque along the z-axis.

The learning of $\boldsymbol{\theta}$ will be driven by the measurements $\mathcal{D}$ which can be either joint torques or wrenches (so far rather uncommon); that is:

$$\mathcal{D} = \{\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_N, \boldsymbol{w}_1, \dots, \boldsymbol{w}_N\}. \tag{17}$$

[3]To ease notation we drop the left upper index, assuming that a vector for the $j$th link is defined in the $j$th frame, unless otherwise specified.

[4]Notation $\boldsymbol{w}_{j,k}$ indicates the wrench vector caused by link $k$ on link $j$.

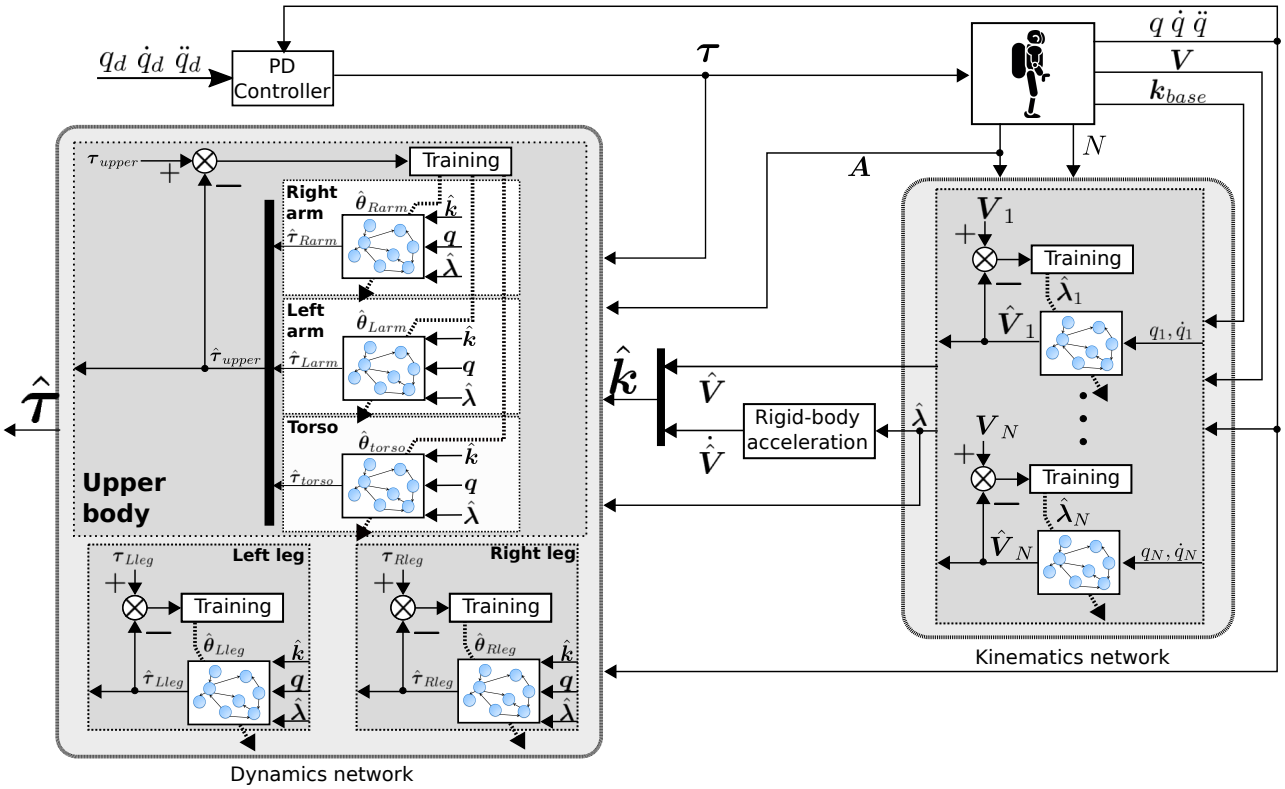



Fig. 3: Learning scheme for an $N$-DoF humanoid. Note that the different learning stages are dictated by the structure of the problem and the respective network structure (meta parameters) is not a choice but a consequence of $\boldsymbol{A}$ and the first-order principles.

## III. Parameter Learning: Instantiating the Self

Learning of the missing elements in $\mathcal{S}$ is executed according to Fig. 3. This scheme stems from the structure of the model itself, as summarized by eqs. (5). Exploitation of said structure can be appreciated in the learning of the dynamics as the topological knowledge allows the separation of the problem into sub-learning problems that are easier to solve; namely, legs and upper body. To generate information to steer the learning, random trajectories are required from the system, realized via some pure-feedback controller. The data recorded during the experiments are $\boldsymbol{\tau}$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$, as well as the twist vector $\boldsymbol{V}_j$ for each link, including the base link.

The cost function for the kinematics and dynamics learning modules is the *mean squared error* (MSE) of $m$ samples between the measured (supervisory) signals and the corresponding output estimates from the FOP networks.

### A. Learning the body schema

The inputs to the kinematics network are the measured $q_j$, $\dot{q}_j$, and $\ddot{q}_j$ for every joint $j$ in the humanoid. Additionally, $\boldsymbol{k}_{base}$ is an input for all the joints connected to the base link. Having $\boldsymbol{A}$, the morphology is learned by solving optimization problems involving a corresponding pair of joints and an estimated twist vector, eqs. (8a) and (8b) parametrized by the corresponding $\boldsymbol{\lambda}_j$. Since there are $\boldsymbol{V}_j$ available measurements for every joint, $N$ optimization problems of the form

$$\hat{\boldsymbol{\lambda}}_j^* = \underset{\hat{\boldsymbol{\lambda}}_j}{\operatorname{argmin}} \quad \frac{1}{2m}\sum_{k=1}^{m}\left(\boldsymbol{V}_j^{(k)} - \hat{\boldsymbol{V}}_j^{(k)}\right)^{\intercal}\left(\boldsymbol{V}_j^{(k)} - \hat{\boldsymbol{V}}_j^{(k)}\right) \tag{18}$$

are solved in parallel. Knowing $\hat{\boldsymbol{\lambda}}_j^*$, the remaining terms of $\boldsymbol{k}$ can be computed using eqs. (8c) and (8d).

### B. Learning the dynamics description

The dynamics FOPnet, eqs. (5b) and (5c), are trained with inputs $\hat{\boldsymbol{\lambda}}^*$, $\boldsymbol{q}$ and $\boldsymbol{k}$ and supervisory signal $\boldsymbol{\tau}$. The optimal inertial parameters are defined as:

$$\begin{aligned}\hat{\boldsymbol{\theta}}^* = \underset{\hat{\boldsymbol{\theta}}}{\operatorname{argmin}} \quad & \frac{1}{2m}\sum_{k=1}^{m}(\boldsymbol{\tau}_k - \hat{\boldsymbol{\tau}}_k)^{\intercal}(\boldsymbol{\tau}_k - \hat{\boldsymbol{\tau}}_k) \\ \text{subject to} \quad & \hat{\boldsymbol{\theta}} \in \Theta\end{aligned}. \tag{19}$$

The constraints $\Theta$ keep the parameters within physically interpretable values; e.g., the links' masses $m_i$ and diagonal elements of the inertia tensor must be strictly positive.

To ease the learning of $\hat{\boldsymbol{\theta}}$, we exploit the now known body schema of the robot. Three optimization problems are solved: the first two involve the learning of the inertial parameters for the right and left legs given the knowledge of the base kinematics. The third comprises the learning of the parameters of the right and left arms together with the torso (marked as *upper body* in Fig. 3).

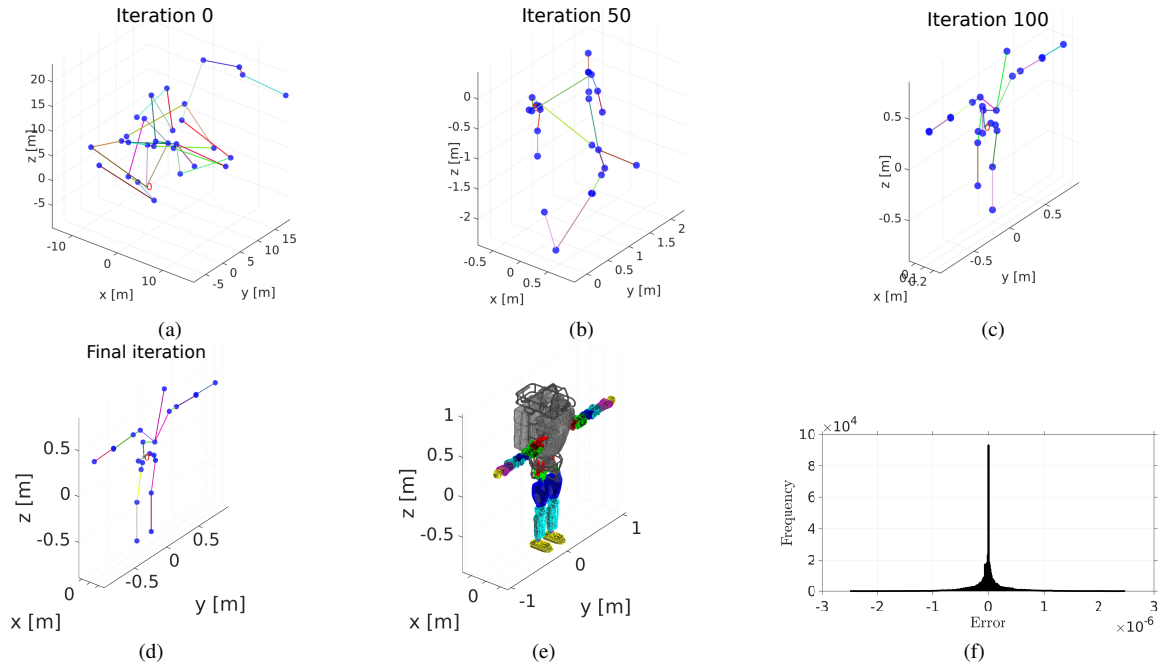



Fig. 4: Body schema estimation from the kinematics FOPnet. (a) Initial estimate (i.e. $\hat{\boldsymbol{\lambda}}_0$), (b) estimate after 50 training epochs, (c) estimate after 100 training epochs, (d) final estimated structure, (e) actual ATLAS structure, (f) kinematics estimation error ($\boldsymbol{k} - \hat{\boldsymbol{k}}$) histogram.

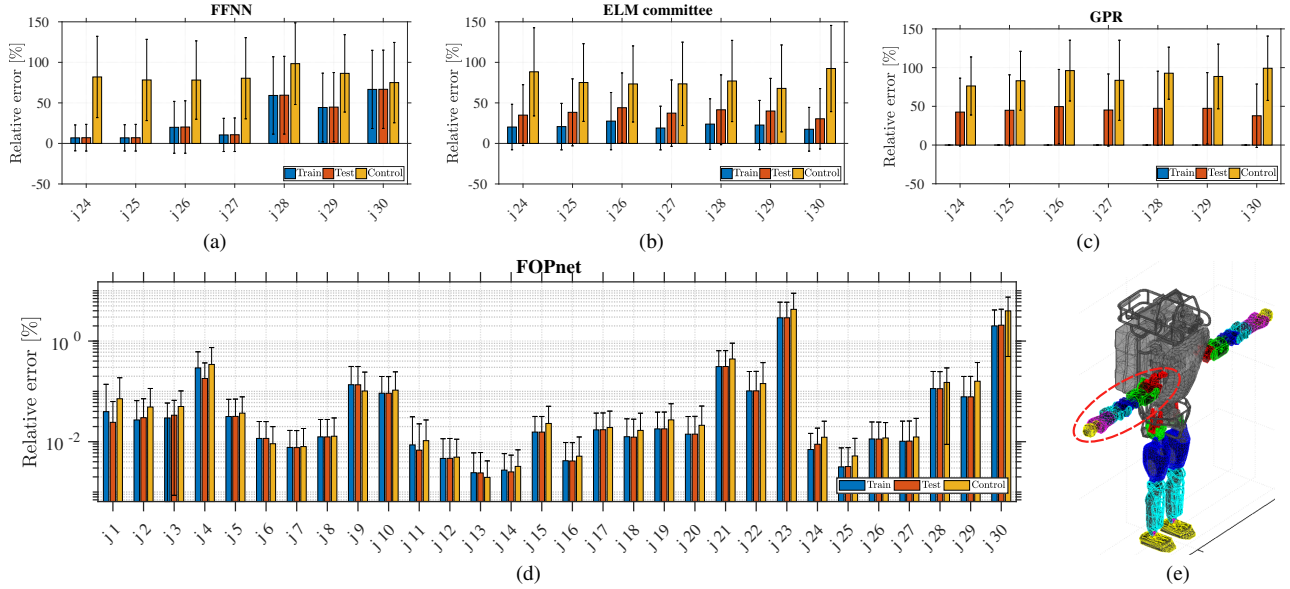



Fig. 5: Torque estimation using three machine learning algorithms and the proposed approach. Relative percentage error for the train, test, and control sets are displayed for: (a) FFNN , (b) Committee of Extreme Learning Machines, (c) Gaussian Process Regression, (d) FOPnet. (e) The machine learning techniques estimate just the right arm (highlighted).

## IV. Results and Discussion

### A. Simulation: body schema and dynamics of a humanoid

A simulated 30-DoF ATLAS humanoid robot was used to test and validate the performance of our approach. The simulation consisted of a free-falling robot, i.e. no contact forces were present. Additionally, as a means to compare the FOPnet with the performance of purely data-based learning schemes, three machine learning algorithms were used to estimate the inverse dynamics of the right arm (joints 24 to 30; see Fig. 5e). The first was a feedforward NN (FFNN) with one hidden layer and 1000 neurons with *tanh* activation functions. The second was a committee of 50 Extreme Learning Machines (ELM) [7], each with 4000 *tanh* units in their hidden layers. The total output of the network was the

averaged output of the whole committee. The third algorithm was Gaussian Process Regression (GPR) [14] with square exponential kernel functions. Random trajectories were generated for every joint as a motor-babbling of sorts to collect training data. A set of 63 distinct 10-second trajectories were sampled at 1 kHz. We then use the first 50 as a data set for training and testing, i.e. a total of 500,000 data points. The data are then shuffled to have better data distribution on the training and test sets. The remaining 13 compose the control set and are used to evaluate the generalization capabilities of all the algorithms outside the learned manifold.

We used a 75%/25% test/train set distribution. For the machine learning algorithms, the input data was scaled to zero mean and unit standard deviation. To train the FFNN, an ADAM gradient descent method was used from the Python package scikit-learn [12]. For the ELM committee, 10,000 distinct data points from the training set were used for training each of the ELMs in the committee (i.e. 50,000 total samples). Finally, only 10,000 points were used for training the GPR since more than this becomes a memory/computation challenge in a regular desktop computer. In contrast, only 1% of the training set is used to solve the training problems (18) and (19). The initial conditions for the parameters $\hat{\boldsymbol{\theta}}_0$ and $\hat{\boldsymbol{\lambda}}_0$ were randomly set. An interior point method using the MATLAB® function *fmincon* was used for training. A mini-batch of 10 samples was used and 10 training cycles, with the initial conditions at every cycle being the result of the previous cycle, were executed for both problems. As mentioned before, the kinematics network was trained first. The kinematics estimation error $\boldsymbol{k} - \hat{\boldsymbol{k}}$ for the number of samples used during training is shown in the histogram in Fig. 4f. As seen in the figure, all errors have a negligible value, implying correct estimation. The final $\hat{\boldsymbol{\lambda}}$ parameters also define an estimate of the humanoid morphology. The results of some intermediate training epochs and the actual body of the ATLAS robot are shown in Figs. 4a to 4e. Once the optimal parameters $\hat{\boldsymbol{\lambda}}^*$ were found, the dynamics network was trained. The inertial parameters were constrained to issue positive masses and positive moments of inertia. After the training cycles, they converged to steady values. The relative percentage difference, computed as:

$$err_{relative} = \frac{|\tau_j - \hat{\tau}_j|}{max(|\tau_j|, |\hat{\tau}_j|)}, \tag{20}$$

is shown in Fig. 5 for the training and test sets as well as for the FFNN, ELM committee, GPR and FOPnet control set. The FFNN shows good performance on the test set, as it belongs to the original sampled space. However, it fails to provide an acceptable performance on the control set. The situation is similar with the performance of the ELM. The GPR exhibits the worst performance in the test and control sets as the amount of data did not suffice for the estimation; feeding more data is, however, computationally challenging. Notoriously, even when the parameters $\hat{\boldsymbol{\lambda}}^*$ and $\hat{\boldsymbol{\theta}}^*$ were trained with a substantially smaller number of samples, the learned FOPnet model is able to generalize well to the test and control sets with minimal error. It is clear that, as the FOP network is based on the physical laws that dictate the behavior of the system, once the network has been trained with a small but information-rich training set its generalization capabilities are superior.

Regarding the performance of the presented machine learning methods, we understand that provided further tuning of their hyperparameters their learning power might be improved. Nonetheless, said tuning is not straight forward and in particular not generative. In addition to this, providing more data may also help improve the generalization capabilities. However, as already mentioned, what is mostly learned is a local manifold and the complexity of collecting large amounts of data that can represent the whole input-output space of a humanoid robots appears to be prohibitive.

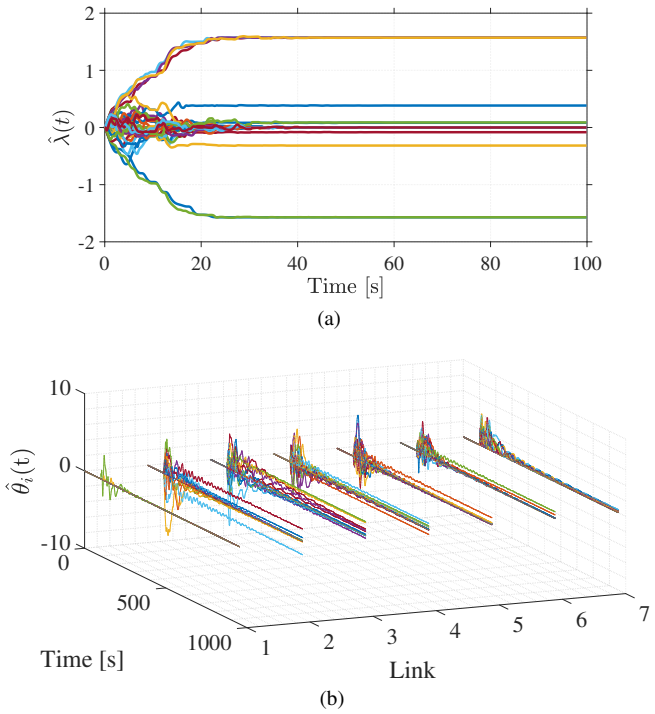



Fig. 6: Convergence of the vector (a) $\hat{\boldsymbol{\lambda}}$ and (b) $\hat{\boldsymbol{\theta}}$ for the *Franka Emika Panda* arm.

### B. Experiment: body schema and dynamics of a manipulator

As an additional test we applied our approach on a 7 DoF *Franka Emika Panda* arm. Instead of offline batch learning in this experiment we used stochastic ADAM gradient descent to enable online learning. The system ran in real time at 1 KHz. The required Cartesian linear and angular velocities and accelerations were artificially generated using the real joint noisy measurements from the robot. Fig. 6a shows the test results. Initial conditions close to zero were given to all the $\boldsymbol{\lambda}$ parameters. It can be seen that the parameters converged after 40 s. The steady-state values obtained are in agreement with those reported in the robot documentation [15]. Moreover, the learning of the inertial parameters was initialized at 100 s, with the estimated kinematics parameters together with the filtered joint torque measurements as inputs. After the cost, given in eq. (19), has converged, a set of feasible inertial parameters has been found that mimic the dynamical response of the system. The convergence of these parameters is shown in Fig. 6b.

### C. Measurement signals

As per the required proprioceptive measurement signals $\mathcal{K}$ and $\mathcal{D}$ we do emphasize that the need of sensors that provide

such signals comes by construction, in the sense that signal availability is a requirement to enable robots to automatically learn their own models and adapt them to potential changes. Moreover, noise can be dealt with, as demonstrated by the experiment, without affecting the outcome of the learning by signal filtering. Placement of the sensors and noise handling are topics that, despite its practical importance and associated complexity, have a smaller relevance to the theoretical framework presented in this work.

## V. Conclusions

This paper discussed the modeling and learning of the inverse dynamics of a humanoid represented as a network or learning modules. This approach is fundamentally different from other classical and machine learning approaches, as the topology and morphology of the robot, i.e. its body schema, is also part of the estimation. We illustrated how the FOPnet first-principles come from fundamental kinematic laws as well as the balance of force and moments acting on a multibody system. Furthermore, we showed the inherent compositionality of the problem by defining learning modules that compute the robot's kinematics and dynamics, and whose parameters correspond to inertial and kinematic parameters that define the robot morphology. A training scheme to identify these parameters was described using an off-the-shelf optimization algorithm. The simulation experiment suggests that FOP networks can accurately represent the body schema and inverse dynamics of humanoid robots with a large number of DoF with outstanding learning speed, accuracy, and generalization capabilities. In contrast, a direct comparison with three purely data-driven machine learning algorithms illustrated how purely data-driven approaches fail to generalize to regions outside the manifold represented by the training data. Future work will cover the development of a scheme to learn the number of links $N$ and the matrix $\boldsymbol{A}$.

## Acknowledgments

We gratefully acknowledge the funding of this work by the *Alfried Krupp von Bohlen und Halbach* Foundation.

## References

[1] C. G. Atkeson and S. Schaal, "Robot learning from demonstration", in *ICML*, vol. 97, 1997, pp. 12–20.

[2] K. Ayusawa, G. Venture, and Y. Nakamura, "Identification of flying humanoids and humans", in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 3715–3720.

[3] V. Bargsten, J. de Gea Fernandez, and Y. Kassahun, "Experimental robot inverse dynamics identification using classical and machine learning techniques", in *ISR 2016: 47st International Symposium on Robotics; Proceedings of*, VDE, 2016, pp. 1–6.

[4] R. Camoriano, S. Traversaro, L. Rosasco, G. Metta, and F. Nori, "Incremental semiparametric inverse dynamics learning", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 544–550.

[5] F. Diaz Ledezma and S. Haddadin, "First-order-principles-based constructive network topologies: An application to robot inverse dynamics", in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Nov. 2017, pp. 438–445. DOI: 10.1109/HUMANOIDS.2017.8246910.

[6] M. Hoffmann, H. Marques, A. Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: A review", *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, 2010.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[8] J. Mayr and H. Gattringer, "Static inertial parameter identification for humanoid robots using a torque-free support", in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, IEEE, 2014, pp. 99–104.

[9] M. Mistry, S. Schaal, and K. Yamane, "Inertial parameter estimation of floating base humanoid systems using partial force sensing", in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, IEEE, 2009, pp. 492–497.

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

[11] Y. Ogawa, G. Venture, and C. Ott, "Dynamic parameters identification of a humanoid robot using joint torque sensors and/or contact forces", in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, IEEE, 2014, pp. 457–462.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python", *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[13] A. S. Polydoros, L. Nalpantidis, and V. Krüger, "Real-time deep learning of robotic manipulator inverse dynamics", in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 3442–3448.

[14] C. E. Rasmussen and C. K. Williams, *Gaussian process for machine learning*. MIT press, 2006.

[15] *Robot and interface specifications*, `https://frankaemika.github.io/docs/control_parameters.html`, Accessed: 2018-07-18.

[16] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators", *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.

[17] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic model identification for industrial robots", *IEEE Control Systems*, vol. 27, no. 5, pp. 58–71, 2007.

[18] S. Traversaro, A. Del Prete, S. Ivaldi, and F. Nori, "Inertial parameters identification and joint torques estimation with proximal force/torque sensing", in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 2105–2110.

[19] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions", *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, 2005.

[20] C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai, "Multi-task gaussian process learning of robot inverse dynamics", in *Advances in Neural Information Processing Systems*, 2009, pp. 265–272.

[21] K. Yamane, "Practical kinematic and dynamic calibration methods for force-controlled humanoid robots", in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, IEEE, 2011, pp. 269–275.