2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Extracting Feasible Robot Parameters from Dynamic Coefficients using Nonlinear Optimization Methods

Claudio Gaz Fabrizio Flacco Alessandro De Luca

***Abstract*— We consider the problem of extracting a complete set of numerical parameters that characterize the robot dynamics, starting from the identified values of dynamic coefficients that linearly parametrize the robot dynamic equations. This information is relevant when realistic dynamic simulations have to be performed using standard packages, or when addressing the efficient numerical implementation of model-based control laws using recursive Newton-Euler algorithms. The formulated problem is highly nonlinear and is solved through the use of global optimization techniques, while imposing also physical bounds on the dynamic parameters. The identification and parameter extraction process is illustrated and experimentally validated on the link dynamics of a KUKA LWR IV+ robot.**

I. Introduction

Complete and accurate dynamic models of robot manipulators are necessary in order to design advanced motion control laws [1], perform realistic simulations [2], or implement sensorless strategies for collision detection [3] and environment interaction [4].

Identification of the dynamic model of a robot is a long standing problem that has been addressed with a variety of estimation techniques [5]. All of them exploit the fundamental property of linear dependence of the robot dynamic equations in terms of a set of ρ *dynamic coefficients*, also known as *base parameters* [6], denoted here by $\boldsymbol{\pi} \in \mathbb{R}^{\rho}$. These coefficients are suitable combinations of geometric and inertial data of the robot bodies, namely of the 10 *dynamic parameters* specifying the mass, the position of the center of mass, and the symmetric inertia matrix for each robot link. For a robot with N links, there are $10N$ such parameters, denoted here by $\boldsymbol{p} \in \mathbb{R}^{10N}$.

The dynamic parameters $\boldsymbol{p}$ can be redefined (keeping their total number constant) so that they appear linearly in the kinetic and potential energy of the robot, and thus also in its dynamic equations (the *standard* parameters mentioned in [6]). The regrouping of the $10N$ parameters into ρ dynamic coefficients, typically with $\rho < 10N$, can be expressed as defining a vector relation $\boldsymbol{\pi} = \boldsymbol{f}(\boldsymbol{p})$, see Fig. 1, and is a necessary step for guaranteeing a full rank regressor matrix in the identification problem [7]. When deriving the Lagrangian model of a robot in symbolic form, the regrouping of dynamic parameters can be performed in different ways [8], [9], so as to yield a (possibly) minimal number of dynamic coefficients —the quantities that can be identified with motion experiments (on sufficiently exciting trajectories) and that really matter in the robot dynamics. However, only a subset $\boldsymbol{p}' \subseteq \boldsymbol{p}$ of the original dynamic parameters will appear in the equations, whereas the remaining $\boldsymbol{p} \backslash \boldsymbol{p}'$ cannot be observed through any motion experiments. Moreover, the majority of the dynamic parameters $\boldsymbol{p}'$ can be identified only in groups. Only a very small number of them will appear as singletons in the coefficients $\boldsymbol{\pi}$, being thus separately identifiable.

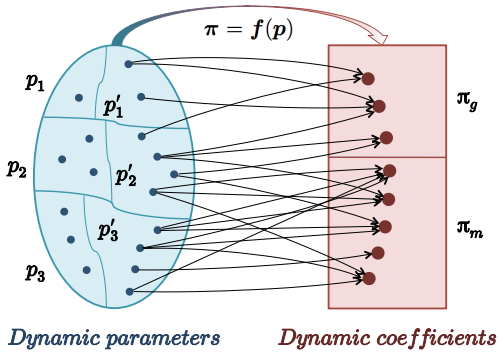


Fig. 1. Dynamic coefficients $\boldsymbol{\pi}$ are being generated as a nonlinear function $\boldsymbol{\pi} = \boldsymbol{f}(\boldsymbol{p})$ of robot dynamic parameters $\boldsymbol{p}$. Parameters are divided in three groups according to their nature (see Sec. II-B). Coefficients can be organized in two vectors $\boldsymbol{\pi}_g$ and $\boldsymbol{\pi}_m$, and only a subset $\boldsymbol{p}'_i$ of parameters $\boldsymbol{p}_i$, $i = 1, 2, 3$, contribute actually to them (see Sec. II-C). The inverse function $\boldsymbol{f}^{-1}$, which should provide dynamic parameters from dynamic coefficients, does not exist in general.

In many cases, the experimental identification (or an a priori knowledge) of a complete set of dynamic coefficients $\boldsymbol{\pi}$ will be sufficient for robot control and planning purposes. However, being able to extract from the identified dynamic coefficients some feasible numerical values for the original dynamic parameters $\boldsymbol{p}$ is a relevant objective, at least in the following two operative situations:

- If we wish to perform dynamic simulations with standard CAD-based systems like V-REP [10]. The user needs to specify explicitly the 10 dynamic parameters of each rigid body (viz., link) in some associated reference frame. Indeed, one can use (*don't care*) values for all those parameters $\boldsymbol{p} \backslash \boldsymbol{p}'$ not involved in robot motion.
- When implementing model-based control laws, e.g., feedback linearization, under hard real-time constraints. The standard solution for robots with many dofs is based on the recursive Newton-Euler (N-E) algorithm [11]. Usual N-E routines require the knowledge of the dynamic parameters of each link in the kinematic chain. Notably, the same holds true in the generalization of the N-E algorithm to robots with joint elasticity [12].

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy ({gaz,fflacco,deluca}@diag.uniroma1.it). This work is supported by the European Commission, within the FP7 ICT-287513 SAPHARI project (www.saphari.eu).

In this paper, we address the problem of recovering a complete set of values for the original robot parameters, starting from the identified dynamic coefficients. This is a nonlinear problem with infinitely many solutions, but not all having a physical significance. In order to discard such unfeasible solutions, we consider upper and lower bounds on the single components of $\boldsymbol{p}$ as well as on the total sum of the link masses. Because of the ill-conditioned nature of the space of solutions, we use global optimization techniques, such as simulated annealing [13].

While the proposed approach is general, we will illustrate the identification and extraction procedure with reference to a specific manipulator largely used in research, the 7R KUKA LWR IV+ robot, taking advantage of some features of this system. The problem of obtaining a complete and accurately identified dynamic model of this robot has generated much interest recently[1]. A few alternative approaches [9], [14], [15] to identification of the dynamic coefficients have been proposed, which may or may not exploit the numerical values of the gravity vector and of the inertia matrix returned by the KUKA Fast Research Interface (FRI) [16]. In particular, we will use our data from [9] as a starting point, recalling briefly the main aspects of the procedure for obtaining the dynamic coefficients (Sec. II). The dynamic parameter extraction problem is formulated and numerically solved in Sec. III, and the obtained results are validated with experimental data in Sec. IV. More details can be found in [17].

We finally remark that we work here only on the link-side dynamics of the KUKA LWR, neglecting therefore the presence of joint elasticity and motor inertias (as opposed, e.g., to [18]). Still, the outcome is what is actually needed by users who rely on the low-level KUKA torque control mode, which allows to bypass the motor dynamics thanks to joint torque sensing, for their dynamic simulations and for model-based control design and implementation.

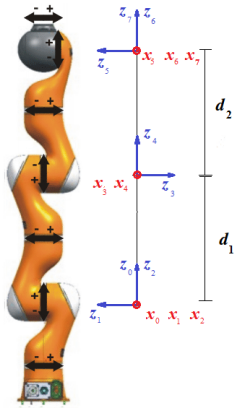
II. Preliminaries

A. KUKA LWR robot and its link dynamics

Figure 2 shows the KUKA LWR IV+ with $N = 7$ revolute joints in the zero position, together with its link frames and associated parameters chosen with the classical Denavit-Hartenberg convention. The origin of the base frame 0 and of the end-effector frame 7 are taken coincident, respectively, with the origin of the frame of link 1 and of link 6. Thus, the two lengths d_1 and d_2 as the only non-zero kinematic parameters. These frames are important in order to interpret the numerical results of the following sections.

The robot is driven by the controller unit KR C2 lr, which is able to provide, through the FRI, the link position $\boldsymbol{q}$ and joint torque $\boldsymbol{\tau}_J$ measurements at a maximum of 1 msec sampling rate, as well as the *numerical* values of the link inertia matrix $\boldsymbol{M}_{\text{num}}(\boldsymbol{q})$ and of the gravity vector $\boldsymbol{g}_{\text{num}}(\boldsymbol{q})$ at the current configuration $\boldsymbol{q}$. The controller works in two modes, generating the correct motor torque $\boldsymbol{\tau}$ in response to

[1]KUKA has never released a public version of the dynamic model of its lightweight robots.



i	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	0	q_1
2	0	$-\pi/2$	0	q_2
3	0	$-\pi/2$	d_1	q_3
4	0	$\pi/2$	0	q_4
5	0	$\pi/2$	d_2	q_5
6	0	$-\pi/2$	0	q_6
7	0	0	0	q_7

Fig. 2. Denavit-Hartenberg frames and table of parameters for the KUKA LWR IV+. All $\boldsymbol{x}$-axes point toward the viewer (frames are displaced sideways for better clarity).

the user command, which can be either a desired link position $\boldsymbol{q}_{\text{d}}$ or velocity $\dot{\boldsymbol{q}}_{\text{d}}$ (position control mode), or a joint torque $\boldsymbol{\tau}_{\text{user}}$ (torque control mode) to be applied to the links.

Although the LWR robot is equipped with harmonic drives and displays non-negligible joint elasticity, thanks to the torque control mode of the KUKA controller, it is possible to by-pass the elasticity in the dynamic model and consider only the link dynamics. This is described by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_J, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Coriolis and centrifugal vector, $\boldsymbol{g}(\boldsymbol{q})$ is the gravity vector, and $\boldsymbol{\tau}_J$ is the vector of joint torques downstream the elasticity.

B. Dynamic parameters

For a rigid robot with N links, the link masses, the position of their centers of mass, and the elements of the inertia matrices of the links build up a vector of dynamic parameters $\boldsymbol{p} \in \mathbb{R}^{10N}$. Let $N = 7$ and denote by m_i be the mass of link i, for $i = 1, \ldots, 7$. The position of the center of mass of link i with respect to the ith link frame is

$${}^i\boldsymbol{r}_{i,ci} = \begin{pmatrix} c_{ix} & c_{iy} & c_{iz} \end{pmatrix}^T, \qquad i = 1, \ldots, 7. \tag{2}$$

Similarly, the link inertia matrix relative to the center of mass of link i is the (possibly full) symmetric matrix

$${}^i\boldsymbol{I}_{\ell_i} = \begin{pmatrix} I_{ixx} & I_{ixy} & I_{ixz} \\ I_{ixy} & I_{iyy} & I_{iyz} \\ I_{ixz} & I_{iyz} & I_{izz} \end{pmatrix}, \qquad i = 1, \ldots, 7. \tag{3}$$

We can arrange all dynamic parameters into three vectors

$$\begin{aligned} \boldsymbol{p}_1 &= \begin{pmatrix} m_1 & \ldots & m_7 \end{pmatrix}^T \in \mathbb{R}^7, \\ \boldsymbol{p}_2 &= \begin{pmatrix} c_{1x} & c_{1y} & c_{1z} & \ldots & c_{7x} & c_{7y} & c_{7z} \end{pmatrix}^T \in \mathbb{R}^{21}, \\ \boldsymbol{p}_3 &= \begin{pmatrix} \mathcal{I}_1^T & \ldots & \mathcal{I}_7^T \end{pmatrix}^T \in \mathbb{R}^{42}, \end{aligned} \tag{4}$$

where

$$\mathcal{I}_i = \begin{pmatrix} I_{ixx} & I_{ixy} & I_{ixz} & I_{iyy} & I_{iyz} & I_{izz} \end{pmatrix}^T \in \mathbb{R}^6. \tag{5}$$

In (4), we can use in an equivalent way the alternative vector $\boldsymbol{p}_2 \in \mathbb{R}^{21}$ of standard parameters

$$\boldsymbol{p}_2 = \begin{pmatrix} c_{1x}m_1 & c_{1y}m_1 & c_{1z}m_1 & \dots & c_{7x}m_7 & c_{7y}m_7 & c_{7z}m_7 \end{pmatrix}^T. \tag{6}$$

C. Dynamic coefficients

In our previous work [9], we have separately identified the terms $\boldsymbol{g}(\boldsymbol{q})$ and $\boldsymbol{M}(\boldsymbol{q})$ in the dynamic equation (1), by inserting the known symbolic terms in two regressor matrices, respectively $\boldsymbol{Y}_g(\boldsymbol{q})$ and $\boldsymbol{Y}_m(\boldsymbol{q})$, and the unknown dynamic coefficients in two vectors, respectively $\boldsymbol{\pi}_g$ and $\boldsymbol{\pi}_m$. With the symbolic expression of the inertia matrix, the term $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is then computed by analytic differentiation [1]. Being $N = 7$ the number of joints, the symmetric inertia matrix has been reshaped into a stack vector $\tilde{\boldsymbol{m}}(\boldsymbol{q})$ of $N(N+1)/2 = 28$ scalar elements. Therefore, we have:

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}_g(\boldsymbol{q})\boldsymbol{\pi}_g, \qquad \tilde{\boldsymbol{m}}(\boldsymbol{q}) = \boldsymbol{Y}_m(\boldsymbol{q})\boldsymbol{\pi}_m. \tag{7}$$

At this stage, a double pruning procedure, described in part in [9] and in full detail in [17], was applied in order to obtain full (column) rank for each regressor in (7). At the end, the dimensions of $\boldsymbol{\pi}_g$ and $\boldsymbol{\pi}_m$ dropped to 12 and 52, respectively. With no end-effector tool mounted and the robot on a horizontal base, the resulting symbolic vector of dynamic coefficients related to gravity is [9]

$$\boldsymbol{\pi}_g = \begin{pmatrix} c_{7y}m_7 \\ c_{7x}m_7 \\ c_{6x}m_6 \\ c_{6z}m_6 + c_{7z}m_7 \\ c_{5z}m_5 - c_{6y}m_6 \\ c_{5x}m_5 \\ c_{5y}m_5 + c_{4z}m_4 + d_2(m_5 + m_6 + m_7) \\ c_{4x}m_4 \\ c_{4y}m_4 + c_{3z}m_3 \\ c_{2x}m_2 \\ c_{3x}m_3 \\ c_{2z}m_2 - c_{3y}m_3 + d_1(m_3 + m_4 + m_5 + m_6 + m_7) \end{pmatrix}. \tag{8}$$

We see that $\boldsymbol{\pi}_g = \boldsymbol{\pi}_g(\boldsymbol{p}'_1, \boldsymbol{p}'_2)$, with $\boldsymbol{p}'_1 \subset \boldsymbol{p}_1$ and $\boldsymbol{p}'_2 \subset \boldsymbol{p}_2$, since some parameters of the first and second links are not excited by gravity. In particular, using the parametrization (4), m_1 is missing in $\boldsymbol{p}'_1$, and c_{1x}, c_{1y}, c_{1z}, and c_{2y} are missing in $\boldsymbol{p}'_2$.

In a similar way, we obtained a vector of dynamic coefficients $\boldsymbol{\pi}_m = \boldsymbol{\pi}_m(\boldsymbol{p}'_1, \boldsymbol{p}'_2, \boldsymbol{p}'_3)$ for the inertia matrix, where $\boldsymbol{p}'_3 \subset \boldsymbol{p}_3$. Also in this case, some dynamic parameters will never be excited no matter which motion is performed. We obtained a vector $\boldsymbol{p}'_3 \in \mathbb{R}^{37}$, since the following elements do not appear in the symbolic expressions of the robot inertia matrix: I_{1xx}, I_{1xy}, I_{1xz}, I_{1yz}, and I_{1zz}.

D. Identification of dynamic coefficients

We collected $M = 1000$ sets of (numerical) gravity vectors and inertia matrix data, bringing the robot in M different static configurations. For a robot configuration $\boldsymbol{q}_k$ in the list ($1 \leq k \leq M$), we have

$$\overline{\boldsymbol{g}}_k = \overline{\boldsymbol{Y}}_{g_k}\boldsymbol{\pi}_g, \qquad \overline{\boldsymbol{m}}_k = \overline{\boldsymbol{Y}}_{m_k}\boldsymbol{\pi}_m, \tag{9}$$

where $\overline{\boldsymbol{Y}}_{g_k} = \boldsymbol{Y}_g(\boldsymbol{q}_k)$, $\overline{\boldsymbol{Y}}_{m_k} = \boldsymbol{Y}_m(\boldsymbol{q}_k)$, and the gravity vector $\overline{\boldsymbol{g}}_k$ and the inertia stack vector $\overline{\boldsymbol{m}}_k$ are both retrieved numerically from the FRI. All these quantities can be stacked in vectors and matrices with $M \times N$ rows, obtaining

$$\overline{\boldsymbol{g}} = \overline{\overline{\boldsymbol{Y}}}_g\boldsymbol{\pi}_g, \qquad \overline{\boldsymbol{m}} = \overline{\overline{\boldsymbol{Y}}}_m\boldsymbol{\pi}_m. \tag{10}$$

Finally, the two equations in (10) were solved in the least squares sense as

$$\hat{\boldsymbol{\pi}}_g = \overline{\overline{\boldsymbol{Y}}}_g^{\#}\overline{\boldsymbol{g}}, \qquad \hat{\boldsymbol{\pi}}_m = \overline{\overline{\boldsymbol{Y}}}_m^{\#}\overline{\boldsymbol{m}}, \tag{11}$$

where $^{\#}$ denotes pseudoinversion.

With this identification procedure, we obtained from (11) the following numerical values for the gravity coefficients:

$$\hat{\boldsymbol{\pi}}_g = \begin{pmatrix} 9.5457 \times 10^{-4} \\ -2.9826 \times 10^{-4} \\ 8.3524 \times 10^{-4} \\ 0.0286 \\ -0.0407 \\ -6.5637 \times 10^{-4} \\ 1.334 \\ -0.0035 \\ -4.7258 \times 10^{-4} \\ 0.0014 \\ 9.4532 \times 10^{-4} \\ 3.4568 \end{pmatrix} \quad \text{[m·kg]}. \tag{12}$$

The values $\hat{\boldsymbol{\pi}}_m$ of the inertial coefficients are found in [17].

III. Extraction of Dynamic Parameters

With the symbolic form of the dynamic coefficients $\boldsymbol{\pi}_g$ in (8) and their numerical values $\hat{\boldsymbol{\pi}}_g$ given in (12), it is possible to extract values for the dynamic parameters $\boldsymbol{p}'_1$ and $\boldsymbol{p}'_2$ by solving the nonlinear system

$$\boldsymbol{\pi}_g(\boldsymbol{p}'_1, \boldsymbol{p}'_2) = \hat{\boldsymbol{\pi}}_g, \tag{13}$$

yielding 12 equations in 23 unknowns, with the given kinematic data $d_1 = 0.4$ m and $d_2 = 0.39$ m. Similarly, for the dynamic coefficients of the inertia matrix we have

$$\boldsymbol{\pi}_m(\boldsymbol{p}'_1, \boldsymbol{p}'_2, \boldsymbol{p}'_3) = \hat{\boldsymbol{\pi}}_m, \tag{14}$$

namely a system of 52 equations in 64 unknowns[2].

Indeed, one could try to solve both (13) and (14) at the same time, but we preferred instead to attack the problem in two subsequent steps, namely: extracting first the numerical values of the dynamic parameters in the gravity vector, and computing then the extra dynamic parameters in the inertia matrix using the solution already found for parameters that are in common. Beside the reduction in complexity, there are also two other features suggesting a partitioned approach. During the identification performed in [9], the best results were obtained when the gravity vector and the inertia matrix were identified separately. Since for this robot the gravity terms have a large influence on the total joint

[2]Four out of the 5 parameters missing from $\boldsymbol{p}'_1$ in the gravity equation (13) appear back in the inertia equation (14): the only scalar parameter from $\boldsymbol{p}_1$ which is missing in (14) is in fact c_{1y}. Thus, the two $\boldsymbol{p}'_1$ used in the gravity and inertia equations are different, but we preferred to overload the notation.

torque, one is interested in having a more reliable estimation of this vector. Moreover, working separately on (13) and (14) allows introducing different possible simplifications in each subproblems. Based on the experimental data, we have in fact forced the identities

$$I_{7xz} = c_{7x} c_{7z} m_7, \qquad I_{7yz} = c_{7y} c_{7z} m_7, \tag{15}$$

reducing by 2 the number of unknowns in (14) down to 62.

For the extraction of dynamic parameters from the gravity vector equation (13), let $\boldsymbol{x} = \left(\boldsymbol{p}_1'^{\,T} \;\; \boldsymbol{p}_2'^{\,T}\right)^T \in \mathbb{R}^{23}$ be the vector of unknowns. In order to select among the infinite solutions to the underdetermined system (13), we can formulate an optimization problem, minimizing an objective function that shapes the solution and imposing constraints so as to guarantee a physical meaning to the obtained solution.

In particular, we forced a non-negative mass for each link and assumed for simplicity a common upper limit (say, 6 kg). Moreover, for each link, we easily inferred that the center of mass is located inside the smallest parallel box which includes the link geometry. In this ways, we generated the lower bounds (LB) and upper bounds (UB) in Tab. I. For comparison, we report also the numerical values assumed in the V-REP module, as described in [19]. Note that the position of the center of mass of link i is referred to the ith DH frame of Fig. 2.

Based on (13) and on the bounds in Tab. I, we define the following constrained nonlinear optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f_1(\boldsymbol{x}) = \left\| \boldsymbol{\pi}_g(\boldsymbol{x}) - \hat{\boldsymbol{\pi}}_g \right\|^2 \\ \text{s.t.} \quad & LB \leq \boldsymbol{x} \leq UB. \end{aligned} \tag{16}$$

Since the manifold generated by the cost function $f_1(\boldsymbol{x})$ contains multiple local minima [17], a global optimization method, like genetic algorithms (GA) [20] or simulated annealing (SA) [13], is mandatory to address problem (16). We have used a SA optimization algorithm [21], which finds a good approximation to the global optimum by randomly restarting the search in a different point of the feasible set. In any event, the algorithm stops when a given threshold is reached. We launched the algorithm $K = 1000$ times, applying a more sophisticated interior-point (IP) Nelder-Mead local optimization algorithm [22] at the end of each completed SA iteration, and using the solution found by SA as starting point. In order to cover the largest surface of $f_1(\boldsymbol{x})$ within the feasible set, the starting point $\boldsymbol{x}_\kappa^s$ at each search iteration κ (with $0 \leq \kappa \leq K$) is randomly selected as

$$\boldsymbol{x}_\kappa^s = LB + (UB - LB)\boldsymbol{u}, \qquad \boldsymbol{u} \sim \mathcal{U}(0,1). \tag{17}$$

The final solution is obtained in a Matlab environment using the routines `simulannealbnd` and `fmincon`, and is reported in Tab. II, where a $*$ denotes *don't care* values. In general, the reported solution cannot be qualified as the 'optimal' one, providing namely the *true* dynamic parameters related to gravity for our robot. Nonetheless, it satisfies all dynamic model equations, and as such can be safely adopted, e.g., to compute the inverse dynamics by means of a Newton-Euler algorithm. With reference to Fig. 1, we have obtained one possible inverse map going from the generated coefficients to the generating dynamic parameters.

Next, we turn to the extraction of dynamic parameters from the stacked inertia vector equation (14). Let $\boldsymbol{y} = \left(\boldsymbol{p}_1'^{\,T} \; \boldsymbol{p}_2'^{\,T} \; \boldsymbol{p}_3'^{\,T}\right)^T \in \mathbb{R}^{62}$ be the unknown vector of parameters, taking into account (15). The procedure is very similar to what was done for the gravity vector. In this case, however,

TABLE I

LOWER AND UPPER BOUNDS FOR MASSES AND CENTERS OF MASS OF THE KUKA LWR IV+ AND CORRESPONDING DATA IN V-REP

parameter	LB	UB	V-REP value	units
m_1	0	6	2.7	kg
m_2	0	6	2.7	kg
m_3	0	6	2.7	kg
m_4	0	6	2.7	kg
m_5	0	6	1.7	kg
m_6	0	6	1.6	kg
m_7	0	6	0.3	kg
c_{1x}	-0.05	0.05	0.001340	m
c_{1y}	-0.2	0.05	-0.087777	m
c_{1z}	-0.1	0.1	-0.026220	m
c_{2x}	-0.05	0.05	0.001340	m
c_{2y}	-0.1	0.1	-0.026220	m
c_{2z}	-0.05	0.2	0.087777	m
c_{3x}	-0.05	0.05	-0.001340	m
c_{3y}	-0.05	0.2	0.087777	m
c_{3z}	-0.1	0.1	-0.026220	m
c_{4x}	-0.05	0.05	-0.001340	m
c_{4y}	-0.1	0.1	0.026220	m
c_{4z}	-0.05	0.2	0.087777	m
c_{5x}	-0.05	0.05	-0.000993	m
c_{5y}	-0.2	0.05	-0.111650	m
c_{5z}	-0.1	0.1	-0.026958	m
c_{6x}	-0.05	0.05	-0.000259	m
c_{6y}	-0.1	0.1	-0.005956	m
c_{6z}	-0.1	0.1	-0.005328	m
c_{7x}	-0.05	0.05	0	m
c_{7y}	-0.05	0.05	0	m
c_{7z}	0.05	0.1	0.063	m

TABLE II

FINAL SOLUTION FOR GRAVITY PARAMETERS

(COST $= 3.2073 \times 10^{-04}$)

parameter	final solution	parameter	final solution
m_1	$*$	c_{3x}	0.0002593328
m_2	4.2996847737	c_{3y}	0.1137431845
m_3	3.658530333	c_{3z}	-0.000100257
m_4	2.3846673548	c_{4x}	-0.0014648843
m_5	1.7035567183	c_{4y}	-0.0000461
m_6	0.4000713156	c_{4z}	0.148580959
m_7	0.6501439811	c_{5x}	-0.0003791484
c_{1x}	$*$	c_{5y}	-0.0553526131
c_{1y}	$*$	c_{5z}	-0.0101255137
c_{1z}	$*$	c_{6x}	0.0020739022
c_{2x}	0.0003284751	c_{6y}	0.0586184696
c_{2y}	$*$	c_{6z}	-0.044799983
c_{2z}	0.0823647642	c_{7x}	-0.0004601303
		c_{7y}	0.0014789221
		c_{7z}	0.0715608282

TABLE III

LOWER AND UPPER BOUNDS FOR THE INERTIA MATRIX ELEMENTS OF THE KUKA LWR IV+ AND CORRESPONDING DATA IN V-REP

parameter	LB	UB	V-REP value	units
I_{1xx}	0	0.05	0.039	kg·m^2
I_{1xy}	-0.005	0.005	3.206e-04	kg·m^2
I_{1xz}	-0.005	0.005	9.415e-05	kg·m^2
I_{1yy}	0	0.05	6.887e-03	kg·m^2
I_{1yz}	-0.005	0.005	-2.681e-03	kg·m^2
I_{1zz}	0	0.05	0.03698	kg·m^2
I_{2xx}	0	0.05	0.039	kg·m^2
I_{2xy}	-0.005	0.005	9.415e-05	kg·m^2
I_{2xz}	-0.005	0.005	-3.145e-04	kg·m^2
I_{2yy}	0	0.05	0.03698	kg·m^2
I_{2yz}	-0.005	0.005	9.747e-03	kg·m^2
I_{2zz}	0	0.05	6.887e-03	kg·m^2
I_{3xx}	0	0.05	6.887e-03	kg·m^2
I_{3xy}	-0.005	0.005	3.145e-04	kg·m^2
I_{3xz}	-0.005	0.005	-9.5572e-05	kg·m^2
I_{3yy}	0	0.05	6.8872e-03	kg·m^2
I_{3yz}	-0.005	0.005	2.681e-03	kg·m^2
I_{3zz}	0	0.05	0.037	kg·m^2
I_{4xx}	0	0.05	0.037	kg·m^2
I_{4xy}	-0.005	0.005	9.415e-05	kg·m^2
I_{4xz}	-0.005	0.005	3.1455e-04	kg·m^2
I_{4yy}	0	0.05	0.037	kg·m^2
I_{4yz}	-0.005	0.005	9.747e-03	kg·m^2
I_{4zz}	0	0.05	6.887e-03	kg·m^2
I_{5xx}	0	0.05	0.032	kg·m^2
I_{5xy}	-0.005	0.005	-1.898e-04	kg·m^2
I_{5xz}	-0.005	0.005	-4.474e-05	kg·m^2
I_{5yy}	0	0.05	4.945e-03	kg·m^2
I_{5yz}	-0.005	0.005	-2.023e-03	kg·m^2
I_{5zz}	0	0.05	0.03	kg·m^2
I_{6xx}	0	0.01	0.003	kg·m^2
I_{6xy}	-0.002	0.002	-2.463e-06	kg·m^2
I_{6xz}	-0.002	0.002	-2.323e-06	kg·m^2
I_{6yy}	0	0.01	3.068e-03	kg·m^2
I_{6yz}	-0.002	0.002	-7.008e-05	kg·m^2
I_{6zz}	0	0.01	3.47e-03	kg·m^2
I_{7xx}	0	0.005	3.47e-03	kg·m^2
I_{7xy}	-0.0005	0.0005	0	kg·m^2
I_{7xz}	-0.0005	0.0005	0	kg·m^2
I_{7yy}	0	0.005	1.292e-03	kg·m^2
I_{7yz}	-0.0005	0.0005	0	kg·m^2
I_{7zz}	0	0.005	1.584e-04	kg·m^2

finding upper and lower bounds to the elements of the link inertia matrices is a bit more complex. To have reasonable estimates, the mass distribution and the position of the center of mass for each link were evaluated first with the aid of CAD tools, such as *3DS MAX* and *MeshLab*, where a model of the KUKA LWR was replicated assuming uniform mass distribution. Trading off between the desire of strict bounds, so as to limit the feasible set when searching for a candidate solution, and the need of larger bounds, not to cut off any potential good candidate, we eventually selected the bounds in Tab. III. The table reports also the nominal data assumed[3] in V-REP. Moreover, since the total weight of the KUKA LWR IV+ is approximately 16 kg [23], we imposed also two linear inequality constraints on the total robot mass:

$$13 \leq \sum_{i=1}^{7} m_i \leq 19. \tag{18}$$

Based on (14), on the bounds in Tab. III, and on (18), we define a second constrained nonlinear optimization problem:

$$\begin{aligned} \min_{\boldsymbol{y}} \; & f_2(\boldsymbol{y}) = \left\| \boldsymbol{\pi}_m(\boldsymbol{y}) - \hat{\boldsymbol{\pi}}_m \right\|^2 + g(\boldsymbol{y}) \\ \text{s.t.} \quad & LB \leq \boldsymbol{y} \leq UB, \end{aligned} \tag{19}$$

with

$$g(\boldsymbol{y}) = \begin{cases} \delta & \text{if } \max\left\{0, \sum_{i=1}^{7} y_i - 19, 13 - \sum_{i=1}^{7} y_i\right\} > 0, \\ 0 & \text{else,} \end{cases}$$

where $y_i = m_i$, for $i = 1, \ldots, 7$, and $\delta = 10$ is a (relatively) large factor in the penalty term g added to the objective function when one of the constraints in (18) is not satisfied. Keeping the gravity parameters found in Tab. II as fixed, we applied again a global numerical optimization algorithm to solve (19). The parameters extracted from the identified dynamic coefficients that parametrize the inertia matrix are given in Tab. IV, where, for compactness, *don't care* values are not reported.

IV. VALIDATION BY MEANS OF NEWTON-EULER ALGORITHM

In this section we present a validation test of the obtained dynamic parameters in Tables II and IV. These numerical values have been inserted in a Newton-Euler routine in order to compute the joint torques necessary to perform a desired trajectory in the joint space. Moreover, the following arbitrary choices have been made for *don't care* values:

$$\begin{aligned} & c_{1y} = 0.01 \text{ m} \\ & I_{1xx} = I_{1xy} = I_{1xz} = I_{1yz} = I_{1zz} = 0.01 \,\text{kg·m}^2 \end{aligned} \tag{20}$$

TABLE IV

FINAL SOLUTION FOR INERTIAL PARAMETERS (COST $= 5.2245 \times 10^{-12}$)

parameter	final solution	parameter	final solution
m_1	4.1948152162	I_{4xz}	-0.0005187982
c_{1x}	-0.0216387515	I_{4xy}	0.000000225
c_{1z}	-0.0376881829	I_{4yz}	-0.0005484476
c_{2y}	-0.0041132249	I_{5xx}	0.006322648
I_{1yy}	0.0018932828	I_{5yy}	0.0012020203
I_{2xx}	0.0474108647	I_{5zz}	0.0070806218
I_{2yy}	0.05	I_{5xy}	-0.0002163196
I_{2zz}	0.001601901	I_{5xz}	0.00000652
I_{2xy}	-0.00000621	I_{5yz}	-0.005
I_{2xz}	0.0001166457	I_{6xx}	0.0005278646
I_{2yz}	-0.0009141575	I_{6yy}	0
I_{3xx}	0.0469510749	I_{6zz}	0.0034899625
I_{3yy}	0.0008344566	I_{6xy}	0.0000483
I_{3zz}	0.05	I_{6xz}	-0.0000375
I_{3xy}	-0.000271431	I_{6yz}	-0.0010605344
I_{3xz}	4.09E-008	I_{7xx}	0
I_{3yz}	-0.000577228	I_{7yy}	0.0000323
I_{4xx}	0.0124233226	I_{7zz}	0.0001187527
I_{4yy}	0.0072708907	I_{7xy}	-0.000000577
I_{4zz}	0.0099884782		

[3]In [19], the inertia matrix of link i is computed in a reference frame placed at the center of mass and oriented like the corresponding DH frame i in Fig. 2. In our notation, the inertia matrix ${}^i\boldsymbol{I}_{\ell_i}$ is referred to the DH frame of link i. Thus, we should apply Steiner's theorem [1] in order to have a correct correspondence of parameters.

Since the robot base is fixed and there is no contact at the end-effector, we have in the N-E forward and backward initializations [1]

$$\boldsymbol{f}_8^8 = \mathbf{0},\ \boldsymbol{\mu}_8^8 = \mathbf{0}, \qquad \boldsymbol{\omega}_0^0 = \dot{\boldsymbol{\omega}}_0^0 = \mathbf{0},\ \boldsymbol{a}_0^0 = \mathbf{0}, \qquad (21)$$

where $\boldsymbol{f}_8^8$ and $\boldsymbol{\mu}_8^8$ are the external force and torque at the end-effector, and $\boldsymbol{\omega}_0^0$, $\dot{\boldsymbol{\omega}}_0^0$ and $\boldsymbol{a}_0^0$ are, respectively, the angular velocity, angular acceleration, and linear acceleration of the robot base.

The desired trajectory in the joint space has been designed so as to have both low and high joint velocities and accelerations. All joints are requested to follow a concatenation of 7 cubic trajectories interpolating 8 knots, with zero initial and final velocity at every knots, see Tab. V.

TABLE V

JOINT TRAJECTORIES USED FOR VALIDATION

knot	configuration	time
1 (start)	$\boldsymbol{q}_1 = \left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)^T$	$t_1 = 0$
2	$\boldsymbol{q}_2 = (0, 0, 0, 0, 0, 0, 0)^T$	$t_2 = 5$
3	$\boldsymbol{q}_3 = \left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)^T$	$t_3 = 8$
4	$\boldsymbol{q}_4 = -\left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)^T$	$t_4 = 15$
5	$\boldsymbol{q}_5 = \left(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}\right)^T$	$t_5 = 22$
6	$\boldsymbol{q}_6 = \left(\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}\right)^T$	$t_6 = 24$
7	$\boldsymbol{q}_7 = \left(-\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}\right)^T$	$t_7 = 25.5$
8 (end)	$\boldsymbol{q}_8 = \left(\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}\right)^T$	$t_8 = 28$

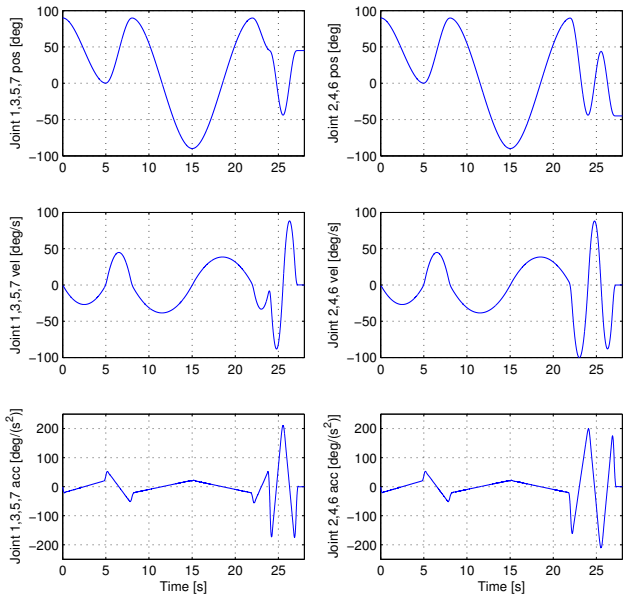


Fig. 3. Joint position (top), velocity (center) and acceleration (bottom) imposed to the KUKA LWR for the torque comparison experiment.

The obtained position, velocity, and acceleration of all seven joints are shown in Fig. 3. We have imposed this desired trajectory to the real robot and have collected the measured link positions $\boldsymbol{q}$ and joint torque $\boldsymbol{\tau}_J$ by means

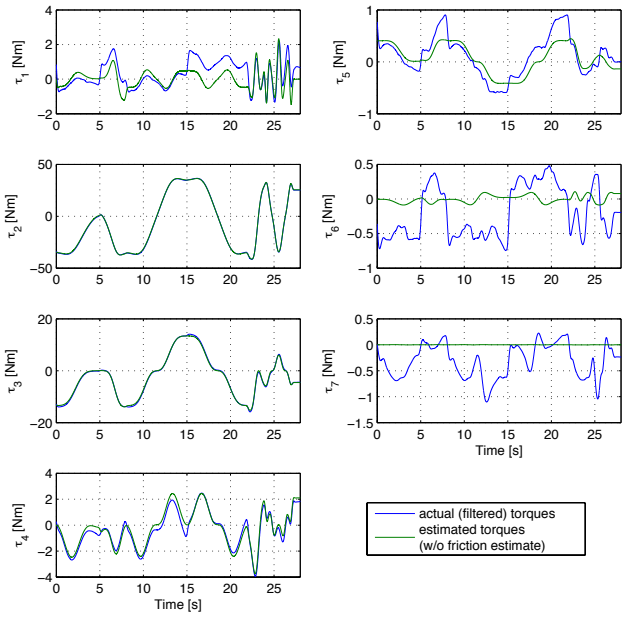


Fig. 4. Measured joint torques filtered by a low-pass filter (blue lines) and estimated joint torques (green lines) obtained with a Newton-Euler routine using the dynamic parameters in Tables II and IV.

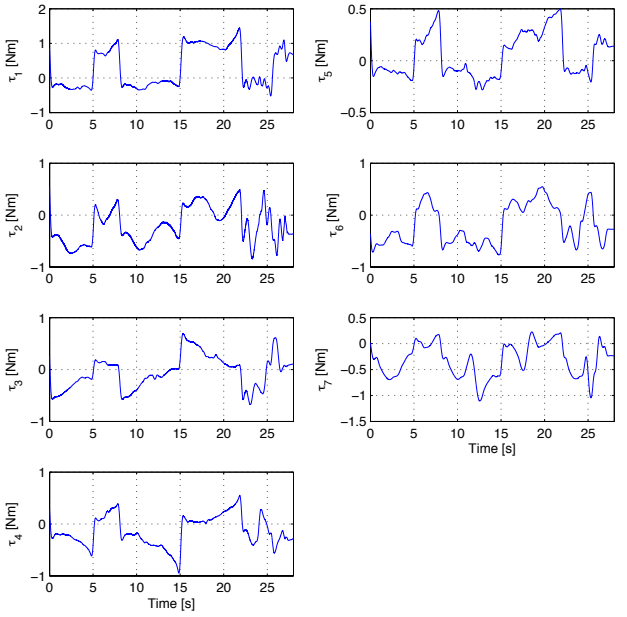


Fig. 5. Residual error torques from Fig. 4.

of the FRI routines `GetMeasuredJointPositions` and `GetMeasuredJointTorques`. Note that, in order to make the robot reach the high joint velocities toward the trajectory end, we removed the 80 W power limitation, by plugging the X15 connector, as described in the manual of the LWR controller [24]. To evaluate joint velocities and accelerations, the position data were numerically differentiated off line twice and then filtered through a 4th order zero-phase digital Butterworth filter with a cutoff frequency of 1 Hz.

These were finally substituted, together with the dynamic parameters from Tables II and IV, in the N-E routine in order to obtain an estimate $\hat{\boldsymbol{\tau}}_J$ of the joint torques.

Figure 4 shows the evolution of the filtered torque $\boldsymbol{\tau}_J$ measured by the joint torque sensors, together with the estimated joint torque $\hat{\boldsymbol{\tau}}_J$, as computed by the N-E algorithm, for all seven robot joints. While a good estimation was obtained for joints 2, 3, 4, and 5, the residual errors (always smaller than maximum 1.5 Nm, see Fig. 5) at the other joints 1, 6, and 7 suggest the presence of unmodeled dynamics, most likely static and viscous friction on the link side of the transmissions at the joints. In fact, Figure 6 shows that the results obtained with the N-E routine (which uses the extracted dynamic parameters in this paper) are comparable with those obtained with a Lagrangian approach (which uses experimentally identified dynamic coefficients) previously reported in [9]. An estimate of the friction torque at link side, neglected so far, has been carried out with good results and will be reported in an expanded version of this paper.

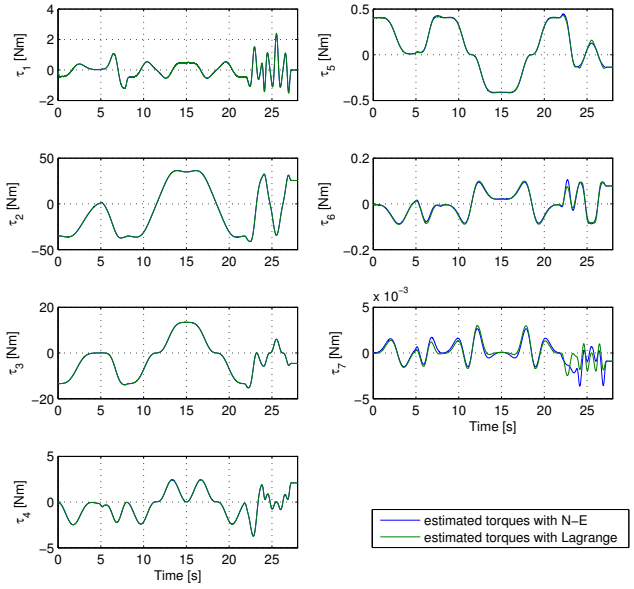


Fig. 6. Comparison between the estimation of the joint torques with the N-E routine (blue lines) and with the Lagrangian approach in [9] (green lines).

V. Conclusions

We presented a numerical approach to the solution of a special problem in identification of robot dynamic models. Namely, given a set of experimentally identified values for the dynamic coefficients that linearly parametrize the robot dynamics, find a set of individual dynamic parameters that *i)* match in the right combinations the given dynamic coefficients; *ii)* satisfy additional user-defined bounds, typically in the form of box constraints, which guarantee their physical significance (e.g., positive link masses); *iii)* allow their straightforward use in simulation tools and in recursive numerical implementation of control laws that need the complete set of robot parameters. The problem has been formulated as a constrained nonlinear optimization problem and solved through the use of global optimization techniques. The dynamic identification process has been validated with experimental data of a KUKA LWR IV+ robot moving at fast speeds.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modeling, Planning and Control*, 3rd ed. London: Springer, 2008.

[2] R. Featherstone, *Robot Dynamics Algorithms*. Kluwer, 1987.

[3] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight robot arm," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 1623–1630.

[4] E. Magrini, F. Flacco, and A. De Luca, "Control of generalized contact motion and force in physical human-robot interaction," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 2298–2304.

[5] J. Hollerbach, W. Khalil, and M. Gautier, "Model identification," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 321–344.

[6] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. Hermes Penton London, 2002.

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

[8] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[9] C. Gaz, F. Flacco, and A. De Luca, "Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 1386–1392.

[10] Coppelia Robotics. (2015) v-rep virtual robot experimentation platform. [Online]. Available: http://www.coppeliarobotics.com

[11] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69–76, 1980.

[12] G. Buondonno and A. De Luca, "A recursive Newton-Euler algorithm for robots with elastic joints and its application to control," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 5526–5532.

[13] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.

[14] M. Gautier, A. Jubien, A. Janot, and P.-P. Robet, "Dynamic identification of flexible joint manipulators with an efficient closed loop output error method based on motor torque output data," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 2949–2955.

[15] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the Kuka LightWeight robot: Comparison between actual and confidential Kuka's parameters," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2014, pp. 483–488.

[16] *KUKA.FastResearchInterface 1.0*, KUKA System Technology (KST), D-86165 Augsburg, Germany, 2011, version 2.

[17] C. Gaz, "On Dynamic Identification and Control Issues for the KUKA LWR Robot," Ph.D. dissertation, DIAG, Sapienza Università di Roma, April 2016.

[18] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the Kuka LWR robot using motor torques and joint torque sensors data," in *Proc. 19th IFAC World Congr.*, 2014, pp. 8391–8396.

[19] M. Cefalo. (2015) Notes on the KUKA LWR4 dynamic model. [Online]. Available: http://www.coppeliarobotics.com/contributions/LBR4p_dynamic_model.pdf

[20] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[22] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, vol. 7, pp. 308–313, 1965.

[23] *KUKA Lightweight Robot 4+ Specification*, KUKA Roboter GmbH, 22-12-2011, version Spez LBR 4+ V5 en.

[24] *KUKA Controller KR C2 lr Specification*, KUKA Laboratories GmbH, D-86165 Augsburg, Germany, 2012, version V5 en, pages 14–15.