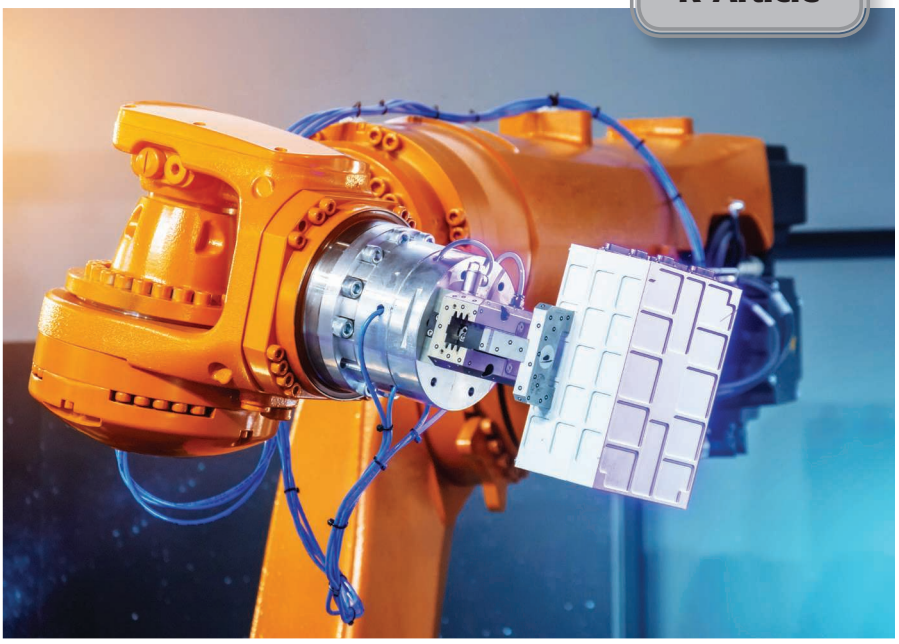
By Giacomo Golluccio, Giuseppe Gillini, Alessandro Marino, and Gianluca Antonelli



A Reproducible Comparison With Experiments on the Kinova Jaco²

Robot Dynamics Identification

Identifying the dynamic parameters of robots is a long-standing topic in robot control, and, recently, research in the area witnessed a renewal of activity. This is motivated by the high performance required for model-based control of, for example, legged or surgical robots and the need to obtain realistic simulations of such systems.

The operation of computing joint torques starting from joint positions, velocities, and accelerations and including model-based control approaches relies on the computation of inverse dynamics [1]. Direct dynamics, on the other hand, is the operation of computing joint accelerations starting from joint positions, velocities, and torques and is required, for instance, in numerical simulations, recently renamed *digital twins*, and in model-based motion planning. Both operations require a representation of the dynamic model of the robot; in addition, several other advanced functionalities, such as impact detection, also require a good knowledge of the model and of its parameters [2].

Consequently, it is of the utmost importance to derive the dynamic model and then to estimate its parameters. However, physically infeasible estimates might lead to nonpositive inertia matrices at some joint configurations or, in the worst case, in the overall joint space. Such dynamic models would lead to unrealistic simulations and negatively affect model-based control since the use of a dynamic model with a nonpositive inertial matrix might lead to an unstable system [3].

In general, robot dynamic parameters, such as mass and inertia tensors, exhibit physical restrictions that must be properly addressed to obtain meaningful estimates. However, several identification solutions, being based on regression techniques [4] (with and without considering constraints), generate nonphysical estimates due, for instance, to unavoidable modeling errors, incorrect setup of the identification experiment, and incorrect choice of parameter constraints. For all of these reasons, the community has devoted considerable effort to identification topics, resorting to classical data estimation algorithms tailored to take into account the robotic-related physical constraints [5].

Literature Analysis

Most of the identification approaches share the core methodology of

- exploiting the linear-in-the-parameter property of robot dynamics
- designing a set of experiments to collect proper data
- implementing the estimation
- validating the data.

The model identified without adding specific constraints on the unknown dynamics while minimizing the reconstruction error on the specific data [5]–[7] generally suffers from

Digital Object Identifier 10.1109/MRA.2020.3004149
Date of current version: 28 July 2020

1070-9932/21©2021IEEE

overfitting and, likely, does not exhibit important properties, such as positive link masses and positive definiteness of link inertia matrices, that constitute the physical meaningfulness of the dynamic parameters themselves, defined as *consistency* in part of the literature.

A first step in this direction was made by the authors of [8], who addressed the identification problem by finding a set of dynamic parameters that is optimal in a least-squares sense and that is closest to a set of physically consistent nominal parameters (for example, retrieved from CAD data). Since the consistency is not set as a hard constraint, this property is not guaranteed; indeed, the authors envisage the need to further modify the inertia parameters in the null space of the regressor if this property is to hold.

A constrained nonlinear optimization problem, solved via global optimization methods (like genetic algorithms), is set forth in [9], with the consistency somehow addressed by imposing lower and upper bounds on each dynamic parameter; however, even in this case, it is not guaranteed that the consistency property is met. In this regard, an extension in this direction is presented in [10], where a constrained optimization problem is solved via iterative techniques.

A rigorous discussion of the physical consistency of robot dynamic parameters can be found in [11], where it is shown that feasible solutions can be treated within the framework of linear matrix inequality (LMI) and handled by semidefinite programming (SDP) techniques (LMI–SDP), by which a global solution is found. Specifically, the considered constraints include, for each link, the nonnegativity of the mass, the positive definiteness of the link inertia matrix, and bounds on the position of the centers of mass. Remarkably, it is highlighted in [12] that the inertia matrix not only must be positive definite but also must exhibit the triangle inequality concerning its eigenvalues.

Based on this result, the work in [13] overtakes the physical semiconsistency achieved in [11] and formulates an LMI-based optimization problem to achieve full physical consistency. In addition to constraints on the link inertia matrix, the devised solution considers geometric constraints concerning the position of the center of mass and the mass distribution. In parallel, similar results were achieved in [14], in which the authors extend the set of constraints previously considered in [11] by taking into account the triangle inequality constraint; it is proven that this can still be solved in the LMI–SDP framework. In the same work, it is experimentally shown that adding meaningful physical constraints helps in preventing overfitting.

Finally, among the experimental works aimed at providing to the community a reliable dynamical model of specific robotic platforms, it is worth mentioning the work in [15], where the dynamic model of the KUKA light-weight robot (LWR) robot is identified and where both joint currents and torque sensor data, together with some insights concerning the mechanical structure of the robot, are exploited to the scope. The model is found in the framework of ordinary and weighted least-squares methods without additional hard constraints. The same robot is identified in [16] where, different from [15], joint elasticity is also taken into consideration.

In general, robot dynamic parameters, such as mass and inertia tensors, exhibit physical restrictions that must be properly addressed to obtain meaningful estimates.

In the present work, by following the guidelines of the community [17], a reproducible comparison of some of the main techniques available in the literature for the identification of the dynamic model of open-chain robots is provided. Five algorithms are taken into consideration in the comparison:

- a CAD-based estimate in which model parameters are extracted from the CAD models generally provided by robot manufacturers
- unconstrained least squares (ULS) [6], in which model parameters are extracted from data with no constraints on the robot model parameters
- constrained least squares (CLS), technique 1 (CLS-1) (inspired by [9])
- CLS, technique 2 (CLS-2) [11]
- CLS, technique 3 (CLS-3) [13], [14].

The last three methods, like the second one, extract the robot model from acquired data; however, different from it, the model is constrained to account for the physical property of the model itself. The comparison is carried out by running experiments on the Kinova Jaco^2 anthropomorphic arm with seven degrees of freedom (DoF), and a quantitative metric is adopted to validate the algorithms. Moreover, the data and the code are made available to the community for improvement and further comparison.

As a further contribution of this article, a fully physically consistent model of this robot is for the first time, to the best of our knowledge, released to the community. A version of the code without a graphical user interface (GUI) can be run online on the Code Ocean platform [23], while a version with a GUI can be freely downloaded and locally run in the MATLAB environment at [24] and [25].

The Considered Robot Model

Here, we consider open-chain manipulators composed of n rigid links connected through n rotational or prismatic joints. The equations of motion of such a system can be obtained via, e.g., the recursive Newton–Euler or Lagrange formulation [1].

In our notation, the superscript i denotes that the corresponding quantity is expressed with respect to frame i, while no superscript means it is expressed in the world reference frame. The mathematical model of the manipulator can be written in compact form as

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}_{\text{full}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\boldsymbol{\pi}_{\text{full}}, \quad (1)$$

where $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric and positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} \in \mathbb{R}^n$ denotes the Coriolis and centrifugal vector, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector, and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector collecting joint torques. Moreover, (1) also highlights that the model can be rewritten in an alternative form by taking into consideration that it is linear with respect to the dynamic parameters [5], which are $\boldsymbol{Y}_{\text{full}} \in \mathbb{R}^{n\times 10n}$ (an upper triangular matrix) and $\boldsymbol{\pi}_{\text{full}} \in \mathbb{R}^{10n}$ (the vector collecting the dynamic parameters of each link).

In detail and by referring to Table 1, the ith link is characterized by 10 parameters, which are $m_i \in \mathbb{R}$, $\boldsymbol{m}_{c_i} \in \mathbb{R}^3$ and the inertia matrix $\boldsymbol{L}_i \in \mathbb{R}^{3\times 3}$ referred to the link frame but of which only six parameters are considered due to its symmetry; that is, $\boldsymbol{l}_i = [l_{i,xx}\ \ l_{i,xy}\ \ l_{i,xz}\ \ l_{i,yy}\ \ l_{i,yz}\ \ l_{i,zz}]^T \in \mathbb{R}^6$. These parameters are stacked into the vector $\boldsymbol{\pi}_i = [m_i\ \ \boldsymbol{m}_{c_i}^{i\,T}\ \boldsymbol{l}_i^T]^T \in \mathbb{R}^{10}$, which represents the vector of the dynamic parameters relative to link i, while vector $\boldsymbol{\pi}_{\text{full}}$ in (1) is such as $\boldsymbol{\pi}_{\text{full}} = [\boldsymbol{\pi}_1^T\ \ \boldsymbol{\pi}_2^T\ \ \ldots\ \ \boldsymbol{\pi}_n^T]^T$.

It is worth noticing that the model described is written at the link side, i.e., ignoring the motor inertia and the motor friction. This is rather common for the latest generation of robots defined as lightweight, which embed a torque sensor in each of the joints at the link side [9], [15].

It is well known that, in general, not all of the dynamic parameters provide a dynamic contribution [11]. Indeed, by resorting to, for instance, the numerical procedure described in [18], they can be clustered in three groups, namely, 1) identifiable, 2) not identifiable, and 3) identifiable in linear combination. By ignoring the not-identifiable parameters since they do not contribute to the robot dynamics (e.g., the mass of the base link of fixed-base manipulators with a first rotational joint), by removing the corresponding column from $\boldsymbol{Y}_{\text{full}}$, and by properly merging together the columns of the regressor $\boldsymbol{Y}_{\text{full}}$ corresponding to parameters identifiable in linear combination, the regressor-based model can be rewritten as

$$\boldsymbol{\tau} = \boldsymbol{Y}_{\text{b}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\boldsymbol{\pi}_{\text{b}}, \quad (2)$$

with $\boldsymbol{Y}_{\text{b}} \in \mathbb{R}^{n\times n_b}$ and $\boldsymbol{\pi}_{\text{b}} \in \mathbb{R}^{n_b}$ and where the dimension $n_b\,(n_b \leq 10n)$ depends on the specific robot kinematics and will be specified for the Kinova Jaco^2 robot in the "Experimental Conditions" section. The set of all of the dynamic parameters, defined previously as *full,* is also defined as *standard* by part of the literature; the set of the dynamic parameters providing a dynamic contribution is defined as the *base parameters* or *dynamic coefficients*.

Table 1. The Denavit–Hartenberg table for the Kinova Jaco^2 robot.

Joint	a (m)	α (rad)	d (m)	θ
1	0	$\pi/2$	0.2755	θ_1
2	0	$\pi/2$	0	θ_2
3	0	$\pi/2$	−0.41	θ_3
4	0	$\pi/2$	−0.0098	θ_4
5	0	$\pi/2$	−0.3111	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.2638	θ_7

Constraints on the Robot Model

Since robots are physical systems, the vector of the dynamics parameters $\boldsymbol{\pi}_{\text{full}}$ (and $\boldsymbol{\pi}_b$) is constrained to account for the physical properties of the system. For instance, as mentioned in the introduction, it is well known that the inertia matrix $\boldsymbol{B}(\boldsymbol{q})$ is positive definite. The main physical constraints considered in the literature are briefly reviewed.

Physical Feasibility

First of all, the mass m_i and inertia matrix $\boldsymbol{L}_i$ are constrained, such as to be a positive scalar and a definitive positive matrix, respectively; $m_i > 0$ and $\boldsymbol{L}_i \succ 0$. However, these constraints are not sufficient to guarantee that matrix $\boldsymbol{B}(\boldsymbol{q})$ is positive definite since $\boldsymbol{L}_i$ is the inertia matrix about the ith link frame. The property $\boldsymbol{B}(\boldsymbol{q}) \succ 0, \forall \boldsymbol{q}$ is ensured if the following constraints are jointly considered:

$$\begin{cases} m_i > 0 \\ \boldsymbol{C}_i = \boldsymbol{L}_i - \dfrac{1}{m_i}\boldsymbol{S}(m_{c_i}^i)^T\boldsymbol{S}(\boldsymbol{m}_{c_i}^i) \succ 0 \end{cases}, \quad (3)$$

where $\boldsymbol{S}(\cdot)$ is the skew-symmetric matrix operator, and $\boldsymbol{C}_i$ is the inertia matrix about the center of mass of link i, which, based on the Huygens–Steiner theorem, is related to the inertia matrix $\boldsymbol{L}_i$.

Physical Consistency

The authors of [12] highlighted that, in addition to the constraints in (3), an additional condition needs to be considered (namely, the triangle inequality) on the eigenvalues λ_j of $\boldsymbol{C}_i$ ($j = 1, 2, 3$) to achieve the so-called physical consistency. This specific condition is proven to be a consequence of the nonnegativity of mass density. Such a condition is taken into consideration in [13] and [14] and is formulated as

$$\begin{cases} 0 < \lambda_3 < \lambda_1 + \lambda_2 \\ 0 < \lambda_2 < \lambda_1 + \lambda_3, \\ 0 < \lambda_1 < \lambda_2 + \lambda_3 \end{cases} \quad (4)$$

which actually implies the second condition of (3).

Additional constraints might exploit the knowledge of the robot's geometric structure by introducing bounded-volume limits [13], [19]. These mainly relate to the position of the center of mass of each link, which might lie in a cuboid expressed in the link frame [9], [11] as

$$\boldsymbol{m}_{c_i,\text{LB}}^i \leq \boldsymbol{m}_{c_i}^i \leq \boldsymbol{m}_{c_i,\text{UB}}^i, \quad (5)$$

with $\boldsymbol{m}_{c_i,\mathrm{LB}}$ and $\boldsymbol{m}_{c_i,\mathrm{UB}}$ as the lower and the upper bounds, respectively, and where the inequalities are intended to be componentwise.

Alternatively, by following the approach in [13], it is possible to require the center of mass of link i to lie inside an ellipsoid $\mathcal{E}_i$, with center $\boldsymbol{x}_{c_i} \in \mathbb{R}^3$ described by

$$\mathcal{E}_i = \{\boldsymbol{x} \in \mathbb{R}^3 \mid (\boldsymbol{x} - \boldsymbol{x}_{c_i}^i)^T \boldsymbol{Q}_{s_i} (\boldsymbol{x} - \boldsymbol{x}_{c_i}^i) \leq 1\}, \tag{6}$$

where $\boldsymbol{Q}_{s_i} \in \mathbb{R}^{3\times 3}$ is a positive definite matrix defining the shape and orientation of the ellipsoid in the link frame. Finally, as further constraints considered in [13], the overall mass of a link might be required to lie inside a region $\mathcal{S}_i$ (which could be an ellipsoid as well; see the "CLS-3" section). This constraint, together with the physical consistency, is addressed as $\mathcal{S}$-density realizability in [13]. As an example, Figure 1(a) reports the ellipsoid $\mathcal{S}_4$ (in orange) containing the fourth link and the ellipsoid $\mathcal{E}_4$ (in red) in which the center of mass $\boldsymbol{m}_{c_4}^4$ constraint is to lie.

Estimate of the Dynamic Parameters

In this section, we review the main identification methods listed in the "Literature Analysis" section and which will be the objects of our comparison. More mathematical details, together with instructions to run the code, can be found in the supplemental material available in [26].

Here, it is worth considering that, apart from the CAD method, the other four identification approaches require either that data be acquired or that data be reconstructed from the robot's onboard sensors, consisting of N joint configurations $(\boldsymbol{q}(t_i), \dot{\boldsymbol{q}}(t_i), \ddot{\boldsymbol{q}}(t_i))$, together with the corresponding torque vector $\boldsymbol{\tau}(t_i)$, with t_i being the generic time instant and $i \in \{1, 2, \ldots, N\}$. These data, together with (1) or (2), are generally used to build an overdetermined system of linear equations in the unknown vector of dynamic parameters $\boldsymbol{\pi}_\mathrm{b}$ in the case of the ULS method and $\boldsymbol{\pi}_\mathrm{full}$ in the cases of the CLS-1, CLS-2, and CLS-3 methods in the form of

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \boldsymbol{\tau}(t_2) \\ \vdots \\ \boldsymbol{\tau}(t_N) \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{(\cdot)}(t_1) \\ \boldsymbol{Y}_{(\cdot)}(t_2) \\ \vdots \\ \boldsymbol{Y}_{(\cdot)}(t_N) \end{bmatrix} \boldsymbol{\pi}_{(\cdot)} = \bar{\boldsymbol{Y}}_{(\cdot)} \boldsymbol{\pi}_{(\cdot)}, \tag{7}$$

where $\boldsymbol{Y}_{(\cdot)}$ and $\boldsymbol{\pi}_{(\cdot)}$ are either the base or full regressor computed in the ith configuration and the base or full vector of the dynamic parameters, depending on the identification method adopted.

CAD Estimate

The simplest way to obtain the vector of the dynamic parameters $\boldsymbol{\pi}_\mathrm{full}$ is to retrieve it from CAD data, which many robot manufacturers make available to the community. The link parameters can be automatically extracted by CAD software once, for instance, properties like the density of each component are specified. However, due to the complexity of the robot structure, the many components involved, and the uniform density assumption, which is usually made, the extracted data might be significantly different than the real data. Moreover, parameters related to friction, when relevant, are unavailable with a CAD approach. On the other hand, the physical constraints in the "Constraints on the Robot Model" section are met by construction.

ULS

The ULS method [5] is a popular method for robots and, in general, for systems identification that, in the case considered and in virtue of (7), consists of solving the following minimization problem in the unknown vector $\boldsymbol{\pi}_b$:

$$\min_{\boldsymbol{\pi}_b} (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_b \boldsymbol{\pi}_b)^\mathrm{T} (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_b \boldsymbol{\pi}_b). \tag{8}$$

In particular, identification requires proper design of the experiments, and it is important to design exciting trajectories to provide accurate and fast parameter estimation, even in the presence of measurement noise, unmodeled dynamics, and external disturbances. Most works in the field of robot identification relate the condition number of regressor $\bar{\boldsymbol{Y}}_b$ to the reliability of the data [5] and design the identification trajectory so as to minimize this condition number, as in [4].

However, despite its simplicity, the ULS approach leads to an estimate of $\boldsymbol{\pi}_b$ for which the physical properties

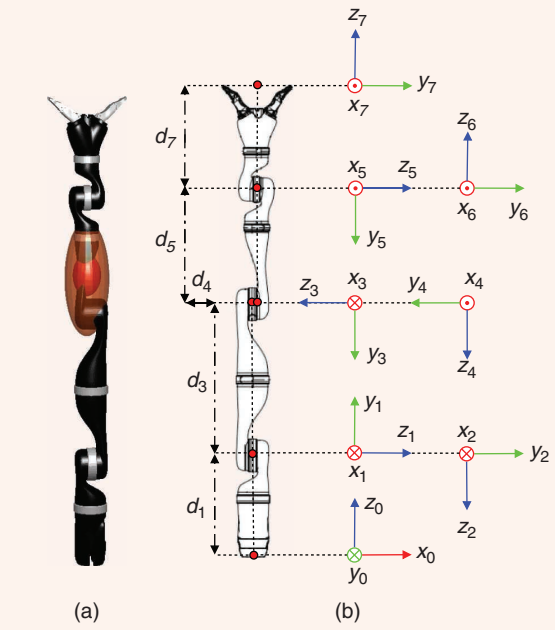


Figure 1. The Kinova Jaco^2 robot considered here as test case. (a) In orange, the external ellipsoid bounding, for example, the fourth link and, in red, the internal ellipsoid containing the center of mass. (b) The reference frames of the Kinova Jaco^2 robot according to the Denavit–Hartenberg convention.

mentioned in the "Constraints on the Robot Model" section are not guaranteed and might suffer from overfitting [13], [14]. For these reasons, in recent decades, many researchers have put effort into the constrained identification of robot manipulators, which basically differ in the number and type of constraints considered (see the "Constraints on the Robot Model" section) and in the methods adopted to solve these problems.

CLS-1

The goal of constrained identification is to extract a dynamic model maintaining, as much as possible, the physically meaning of the parameters. As seen in [8] and [9], constraints on the numerical value of the dynamic parameters in terms of bounds on the CAD values could be added. Inspired by the approach presented in [9], the method consists of estimating the vector of full parameters $\boldsymbol{\pi}_{\text{full}}$; in detail, it exhibits the following characteristics.

- The dependency of $\boldsymbol{\pi}_b$ on the full vector $\boldsymbol{\pi}_{\text{full}}$ is required; that is, $\boldsymbol{\pi}_b = \boldsymbol{\pi}_b(\boldsymbol{\pi}_{\text{full}})$. This relationship can be either linear or nonlinear, depending on the parameterization assumed.
- This dependency is necessary since constraints are set on the vector of full parameters and are in the form $\boldsymbol{\pi}_{\text{LB}} \leq \boldsymbol{\pi}_{\text{full}} \leq \boldsymbol{\pi}_{\text{UB}}$, where $\boldsymbol{\pi}_{\text{LB}}$ and $\boldsymbol{\pi}_{\text{UB}}$ are the lower and upper bounds on the full vector of dynamic parameters, respectively, mainly obtained by CAD and heuristic considerations (e.g., masses are positive).
- The solution sought is the one that, taking into account the acquired data, is closest to a vector $\hat{\boldsymbol{\pi}}_b$, which may be exactly the unconstrained solution in the "ULS" section and which meets the bounds on $\boldsymbol{\pi}_{\text{full}}$.

Moreover, it is important to highlight here that the problem formulation is slightly changed with respect to [9], but the overall approach is kept. Finally, it is worth noticing that, with this approach, neither the physical feasibility nor the physical consistency is guaranteed. Indeed, this property can only be verified a posteriori, and the bounds may eventually be modified according to a trial-and-error approach.

CLS-2

Different from the previously mentioned techniques, the approach in [11] properly takes into account the constraints in (3). In particular, the considered constrained identification aims to estimate the full vector of parameters $\boldsymbol{\pi}_{\text{full}}$ by minimizing the reconstruction error, as in (8), and where the base quantities are substituted by their full counterparts with the following constraints for each link.

- The physical feasibility is described in the "Physical Feasibility" section.
- The lower and upper bounds are considered for link mass m_i, designated $m_{i,\text{LB}}$ and $m_{i,\text{UB}}$, respectively.
- The lower and upper bounds are considered for the first moment of mass $\boldsymbol{m}_{c_i}^i$, designated $\boldsymbol{m}_{c_i,\text{LB}}^i$ and $\boldsymbol{m}_{c_i,\text{UB}}^i$, respectively.

The overall constrained problem is globally solved in the framework of LMI–SDP. Since the full vector of parameters is identified, the full regressor is used, which in general, is intrinsically numerically bad conditioned; i.e., the matrix $\bar{\boldsymbol{Y}}_{\text{full}}$ is never full rank no matter the data collected. Therefore, it is necessary to provide the algorithm with a regularization coefficient.

CLS-3

The last method we consider is presented in [13] (a similar approach is presented in [14]) and is the most complete of the constraints presented in the "Constraints on the Robot Model" section. Similarly to CLS-2, the CLS-3 technique aims to estimate the full vector of parameters $\boldsymbol{\pi}_{\text{full}}$ by minimizing the reconstruction error while considering the following constraints for each link.

- The physical consistency is described in the "Physical Consistency" section.
- The position of the first moment of mass $\boldsymbol{m}_{c_i}$ is forced to reside in an ellipsoid $\mathcal{E}_i$ described by matrix $\boldsymbol{Q}_{s_i}$ and its center $\boldsymbol{x}_{c_i}$, as in (6).
- The mass of link i is contained within a given region $\mathcal{S}_i$, also represented as an ellipsoid.

From the mathematical perspective, the solution of CLS-3 is identical to that of CLS-2, and, thus, it lays in the framework of LMI–SDP. A regularization factor is needed as well [13].

Experimental Conditions

Hardware Description

In this article, the robot Kinova Jaco^2 is considered for the purpose of experimental validation. Kinova Jaco^2 is a lightweight robot characterized by 7 DoF, the Denavit–Hartenberg (DH) table for which is reported in Table 1 and the corresponding link frames in Figure 1(b).

By applying the numerical procedure described in [18], it is possible to cluster the Kinova Jaco^2 dynamic parameters as shown in Table 2 with the linear combination reported in Table 3. The robot is equipped with joint torques mounted after the gear, common to several lightweight arms, such as, for example, the hardware described in [9] and [15]. In addition to the joint torques, the joint positions can be measured only at a sampling frequency $f_s = 100$ Hz with a Ethernet connection, resorting to the library developed by the manufacturer under the Robot Operating System [20].

Validation Methodology

Two main requirements concerning validation need to be satisfied. On one hand, the identification nature asks for a minimization of the reconstruction error along a set of data (the identification set) different from the one used for the identification itself (the validation set). On the other hand, it is required that the constraints in the "Constraints on the Robot Model" section are met, which have, as a main

consequence, that the joint-space inertia matrix is strictly positive definite ($\boldsymbol{B}(\boldsymbol{q}) \succ 0$).

The first requirement is usually verified by resorting to a proper metric, typically the reconstruction error, and, eventually, relative information such as the percentage error. In this article, the following errors are considered:

$$\varsigma = \frac{\sqrt{\chi^2}}{N}, \quad \varsigma_r = \frac{\sqrt{\chi^2}}{\|\bar{\boldsymbol{\tau}}\|}, \tag{9}$$

where N is the number of samples of the data set at hand and χ^2 is defined as

$$\chi^2 = \begin{cases} (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_b \hat{\boldsymbol{\pi}}_b)^T (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_b \hat{\boldsymbol{\pi}}_b) & \text{for ULS} \\ (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_{\text{full}} \hat{\boldsymbol{\pi}}_{\text{full}})^T (\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{Y}}_{\text{full}} \hat{\boldsymbol{\pi}}_{\text{full}}) & \text{for CAD, CLS} - 1,2,3 \end{cases}.$$

By defining $\sigma^2 = \chi^2/\nu$, with ν as the difference between the number of samples and the number of dynamic parameters identified, an estimate of the covariance matrix $\hat{\boldsymbol{M}}$ of the estimated parameters is [5]

$$\hat{\boldsymbol{M}} = \sigma^2 (\bar{\boldsymbol{Y}}_b^{\mathrm{T}} \bar{\boldsymbol{Y}}_b)^{-1} \tag{10}$$

for all of the techniques. For CLS-1, CLS-2, and CLS-3, which identify the full vector of dynamic parameters, the covariance is actually computed on the base representation of the obtained estimation.

The requirement on the positive definiteness of $\boldsymbol{B}(\boldsymbol{q})$ is actually a binary constraint and is met via construction by the CAD estimates and algorithms CLS-2 and CLS-3; however, the ULS and CLS-1 methods may or may not meet it. Since the method CLS-1 outputs the full vector of dynamic parameters, it is possible to check on their value for the fulfillment of the constraints, although this is not possible in the ULS case; in addition, since symbolic computation of the positive definiteness of $\boldsymbol{B}(\boldsymbol{q})$ is computationally intractable, numerical methods must be implemented to check this property. In this case, the numerical validation of the condition $\boldsymbol{B}(\boldsymbol{q}) \succ 0$ is valid for a specific joint configuration $\boldsymbol{q}$, and a sample-based approach is needed to check it against the joint space. Obviously, one single counterexample is sufficient to invalidate the property, while the opposite is true only in a probabilistic fashion for a large number of samples.

Identification and Validation Trajectories

Numerical conditioning on the various optimization problems to be solved needs to be properly guaranteed [5], [21], [22]. Considering the stacked regressor matrices $\bar{\boldsymbol{Y}}_b$ in (7), it is required that its condition number be kept as small as possible while increasing the minimum singular value. In the case of $\bar{\boldsymbol{Y}}_{\text{full}}$ in (7), it is, instead, required that the smallest nonzero singular value be as large as possible.

Table 2. The identifiability of the parameters.

Parameters	1	2	3	4	5	6	7
m							
$m_{c,x}$							
$m_{c,y}$							
$m_{c,z}$							
I_{xx}							
I_{yy}							
I_{zz}							
I_{yz}							
I_{xy}							
I_{xz}							

Red cells: not identifiable; blue cells: identifiable in linear combination; white cells: identifiable alone.

Table 3. The dynamic parameters in linear combinations.

Parameters
$\beta_1 = I_{1,yy} + I_{2,zz}$
$\beta_2 = m_{c_3,y} - d_3(m_4 + m_5 + m_6 + m_7) + m_{c_2,z}$
$\beta_3 = I_{2,xx} - I_{2,zz} + I_{3,zz} + d_3^2(m_4 + m_5 + m_6 + m_7)$
$\beta_4 = I_{2,yy} + I_{3,zz} + d_3^2(m_4 + m_5 + m_6 + m_7)$
$\beta_5 = m_{c_4,y} + d_4(m_6 + m_7 + m_5) + m_{c_3,z}$
$\beta_6 = I_{3,xx} - I_{3,zz} + I_{4,zz} + d_4^2(m_5 + m_6 + m_7)$
$\beta_7 = I_{3,yy} + I_{4,zz} + d_4^2(m_5 + m_6 + m_7)$
$\beta_8 = I_{3,yz} - d_3 d_4(m_5 + m_6 + m_7) - d_3 m_{c_4,y}$
$\beta_9 = m_{c_5,y} + d_5(m_6 + m_7) + m_{c_4,z}$
$\beta_{10} = I_{4,xx} - I_{4,zz} + I_{5,zz} + d_5^2(m_6 + m_7)$
$\beta_{11} = I_{4,yy} + I_{5,zz} + d_5^2(m_6 + m_7)$
$\beta_{12} = I_{4,yz} - d_4 d_5(m_6 + m_7) - d_4 m_{c_5,y}$
$\beta_{13} = m_{c_6,y} + m_{c_5,z}$
$\beta_{14} = I_{5,xx} - I_{5,zz} + I_{6,zz}$
$\beta_{15} = I_{5,yy} + I_{6,zz}$
$\beta_{16} = I_{5,yz} - d_5 m_{c_6,y}$
$\beta_{17} = m_{c_6,z} + m_{c_7,z}$
$\beta_{18} = I_{6,xx} + I_{7,yy} - I_{6,zz}$
$\beta_{19} = I_{6,yy} + I_{7,yy}$
$\beta_{20} = I_{7,xx} - I_{7,yy}$

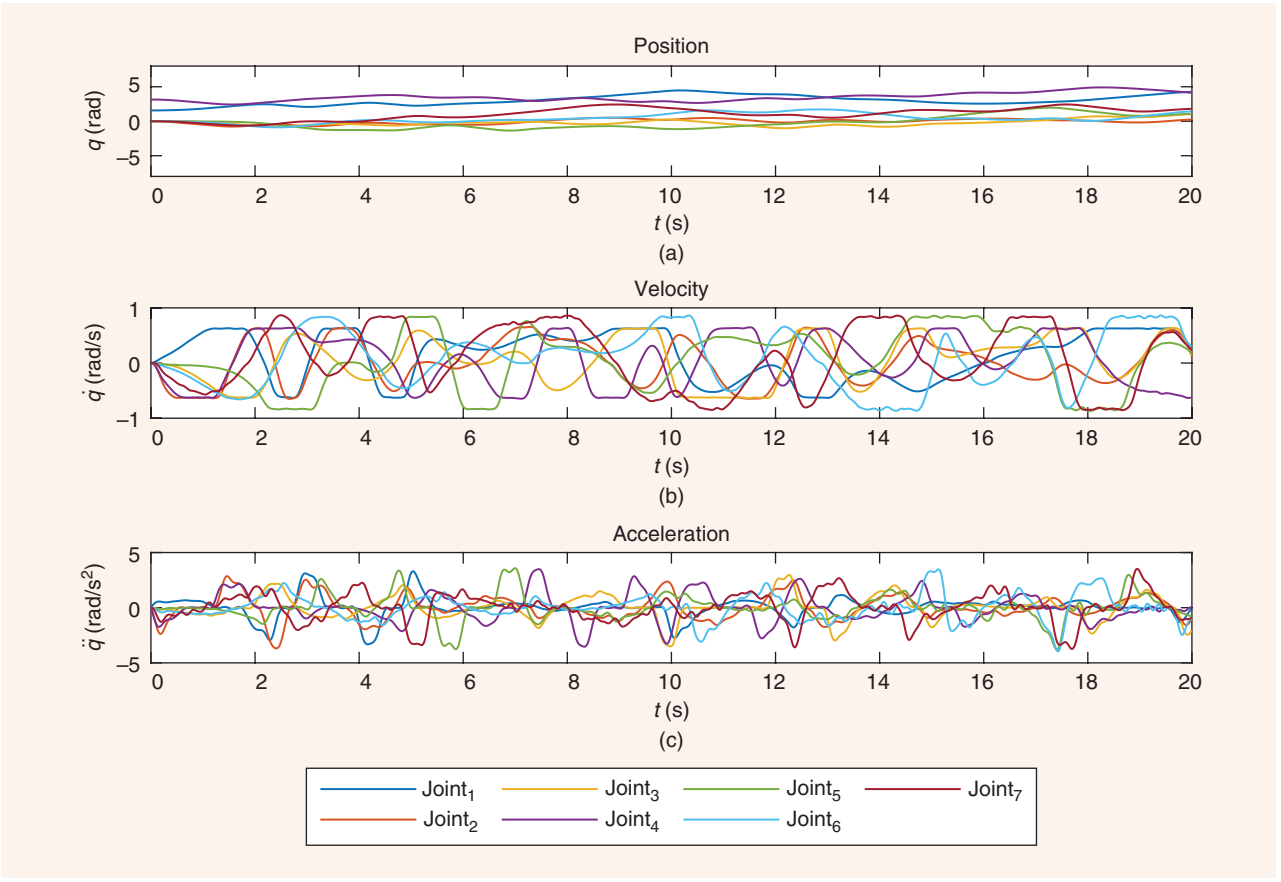


Figure 2. The (a) joint position, (b) velocity, and (c) acceleration zoom representation of the first 20 s of the third trajectory. The full version of the trajectory is used as the identification data set.

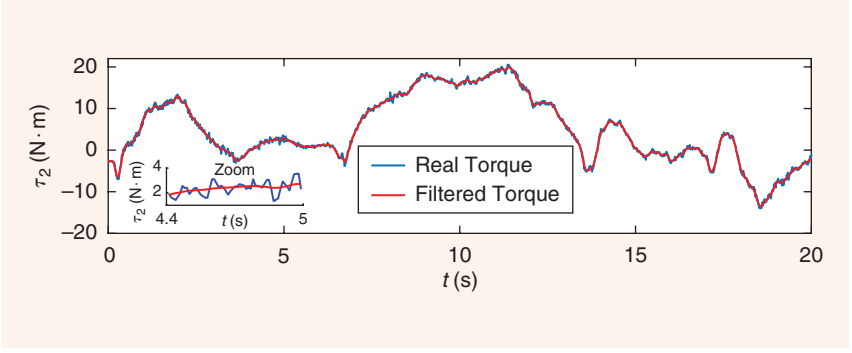


Figure 3. The low-pass filter effect on joint torque: an example for the second joint of the third trajectory. A zoom-in plot in the time interval of 4.4–5 s is reported.

Data Filtering

Since the data are spoiled from noise, it is necessary to carry out low-pass filtering on these data. A Butterworth second-order low-pass filter with a cutoff frequency equal to 5 Hz has been implemented. In the literature, a similar choice of cutoff frequency is adopted; for example, in [9], the value of 1 Hz has been used in the identification of the KUKA LWR IV. In Figure 3, an example of the effect of filtering on the torque data of joint 2 in one of trajectories is reported.

In practice, this means properly spanning the collected data within the allowed range of joint positions, velocities, and accelerations characterizing the robot under study. In our cases, five different trajectories have been considered and generated as in [4], each playing the role of identification trajectory; the remaining ones play the role of validation trajectories.

As an example, one exciting trajectory is shown in Figure 2. The corresponding base regressor is characterized by a condition number of approximately 50 with a minimal singular value of 34.

Results and Discussion

The CAD parameter values have been derived from the official repository of the manufacturer [20], properly referred to the adopted DH-compliant frames by resorting to the Huygens–Steiner theorem. Moreover, the identification algorithms described previously have been run. The identified parameters are reported in Table 4, together with the corresponding variance of the estimate (for instance, the fourth trajectory is adopted to the scope of identification, and the third trajectory is adopted for the

Table 4. The numerical values for base dynamic parameters of the algorithms on the third validation trajectory.

		ULS		CLS-1		CLS-2		CLS-3	
Parameter	**CAD**	**Value**	σ	**Value**	σ	**Value**	σ	**Value**	σ
β_1	0.003715	−0.027334	3.9e−03	0	1.7e−02	0.003664	8.3e−03	0.000824	3.9e−03
β_2	−0.956111	−1.391333	6.9e−04	−1.076	2.9e−03	−1.383625	1.5e−03	−1.390848	6.9e−04
β_3	0.413706	0.535243	9.5e−03	0.47706	4.0e−02	0.579379	2.0e−02	0.594658	9.5e−03
β_4	0.414086	0.585506	4.5e−03	0.45706	1.9e−02	0.605004	9.6e−03	0.596306	4.5e−03
β_5	−0.016199	−0.035576	4.7e−04	−0.01764	2.0e−03	−0.033551	1.0e−03	−0.03573	4.7e−04
β_6	0.000463	0.031621	6.5e−03	−0.016427	2.8e−02	0.054993	1.4e−02	0.004703	6.6e−03
β_7	0.000843	0.024736	2.8e−03	0.000573	1.2e−02	0.004193	6.0e−03	0.006755	2.8e−03
β_8	−0.006642	−0.017786	2.3e−03	−0.007232	9.8e−03	−0.013756	4.9e−03	−0.008831	2.3e−03
β_9	−0.370088	−0.572468	5.0e−04	−0.373640	2.1e−03	−0.678596	1.1e−03	−0.572292	5.1e−04
β_{10}	0.116272	0.165417	4.1e−03	0.118846	1.8e−02	0.210676	8.8e−03	0.173466	4.2e−03
β_{11}	0.116576	0.188398	2.7e−03	0.115246	1.2e−02	0.210918	5.7e−03	0.177551	2.7e−03
β_{12}	−0.00357	−0.015456	1.6e−03	−0.003603	6.6e−03	−0.006591	3.3e−03	−0.004862	1.6e−03
β_{13}	0	−0.002278	3.4e−04	0	1.5e−03	−0.000001	7.2e−04	−0.002054	3.4e−04
β_{14}	0.002554	−0.002358	3.4e−03	0.001	1.4e−02	0.013544	7.2e−03	0.000973	3.4e−03
β_{15}	0.002647	−0.006334	2.0e−03	0.001	8.4e−03	0.00373	4.2e−03	0.001497	2.0e−03
β_{16}	0	0.000519	1.3e−03	0	5.5e−03	0	2.7e−03	−0.001571	1.3e−03
β_{17}	−0.042324	0.142087	3.5e−04	−0.0436	1.5e−03	−0.043599	7.5e−04	0.142759	3.5e−04
β_{18}	0.003917	0.036578	2.4e−03	−0.001	1.0e−02	0.114899	5.0e−03	0.017273	2.4e−03
β_{19}	0.00401	0.021991	1.5e−03	0	6.5e−03	0.105452	3.3e−03	0.017606	1.5e−03
β_{20}	0	−0.018514	1.6e−03	0	6.7e−03	−0.00852	3.3e−03	0.000107	1.6e−03
$m_{c_2,x}$	0	0.006347	4.0e−04	0	1.7e−03	0	8.4e−04	0.006854	4.0e−04
$I_{2,xy}$	0	0.000908	4.1e−03	0	1.7e−02	0	8.7e−03	0	4.1e−03
$I_{2,xz}$	0	0.041782	4.1e−03	0	1.8e−02	0	8.8e−03	0	4.2e−03
I_{2yz}	0	−0.006675	2.4e−03	0	1.0e−02	0	5.0e−03	0	2.4e−03
$m_{c_3,x}$	0	0.013107	5.3e−04	0	2.3e−03	−0.000001	1.1e−03	0.01237	5.3e−04
$I_{3,xy}$	0	−0.003376	2.1e−03	0	9.0e−03	0	4.5e−03	0	2.1e−03
$I_{3,xz}$	0	−0.022552	3.1e−03	0	1.3e−02	0	6.5e−03	0	3.1e−03
$m_{c_4,x}$	0	0.028568	3.8e−04	0	1.6e−03	0	8.1e−04	0.02833	3.8e−04
$I_{4,xy}$	0	0.016842	1.6e−03	0	6.9e−03	0	3.4e−03	0	1.6e−03
$I_{4,xz}$	0	0.008521	1.8e−03	0	7.8e−03	0	3.9e−03	0	1.8e−03
$m_{c_5,x}$	0	0.002895	4.2e−04	0	1.8e−03	0.000001	9.0e−04	0.003247	4.3e−04
$I_{5,xy}$	0	0.003167	1.1e−03	0	4.7e−03	0	2.3e−03	0	1.1e−03
$I_{5,xz}$	0	−0.017816	1.5e−03	0	6.3e−03	0	3.1e−03	0	1.5e−03
$m_{c_x,6}$	0	0.005368	3.6e−04	0	1.5e−03	0.000001	7.7e−04	0.004566	3.6e−04
$I_{6,xy}$	0	0.006955	9.9e−04	0	4.2e−03	0	2.1e−03	0	9.9e−04
$I_{6,xz}$	0	0.003348	9.6e−04	0	4.1e−03	0	2.0e−03	0	9.6e−04
$I_{6,yz}$	0	0.010184	1.0e−03	0	4.4e−03	0	2.2e−03	0	1.0e−03
$m_{c_x,7}$	0	0.000949	2.9e−04	0	1.2e−03	0	6.2e−04	0	2.9e−04
$m_{c_y,7}$	0	−0.001331	2.8e−04	0	1.2e−03	0	6.0e−04	−0.000298	2.9e−04
$I_{7,xy}$	0	0.004124	7.6e−04	0	3.3e−03	0	1.6e−03	0	7.7e−04
$I_{7,xz}$	0	0.00202	7.2e−04	0	3.1e−03	0	1.5e−03	0	7.3e−04
$I_{7,yz}$	0	0.003968	7.4e−04	0	3.2e−03	0	1.6e−03	0	7.4e−04
$I_{7,zz}$	0.000582	0.00047	1.1e−03	0.00047	4.6e−03	0.000004	2.3e−03	0.000107	1.1e−03

validation). As already mentioned, since ULS identifies only the base vector $\boldsymbol{\pi}_b$, while the other algorithms identify the full vector $\boldsymbol{\pi}_{full}$, the linear combinations of the latter have been considered to compare them with the former. Since a graphical visualization of the errors is always meaningful, Figures 4–8 report the reconstruction errors as a time history of measured versus estimated torques for the different algorithms using trajectory 3 for validation.

Table 5 reports an overall comparison of the various techniques among different identification and validation trajectories. Several interesting comments can be made.

- The ULS always exhibits the smallest error along the identification trajectory. This superior performance, however, is not confirmed along the validation trajectories.
- This confirms the observation made in [14], i.e., that the ULS somehow overfits the identification trajectory, while other techniques, taking into account the physical constraints, finally better perform on the validation trajectories.
- The reconstruction made by resorting to the CAD values is polarized to be the worst among the various techniques, confirming the importance of the identification process.
- The CLS-3 method almost always exhibits the smallest error; for a few samples, it is overperformed by ULS and, in one case, by CLS-1. It is worth noticing that the displacement of the first moment of inertia is numerically small and that the errors are numerically very close one each other.
- The last two lines of Table 5 show the binary conditions resulting in the requirement to satisfy the physical constraints in the "Constraints of the Robot Model" section: the sole CAD and CLS-3 satisfy all of them. However, if the reconstruction error is considered, the performance of CLS-3 among the five methods is clearly superior.

As a final consideration, since the robot at hand is equipped with sensors at the link side, motor inertia and joint friction parameters at the motor side are not taken into consideration, because the torque measures are insensitive to these parameters. The identifications discussed have been run as well, including the friction terms at the link side; consistent with the literature, the result was that the errors were almost invariant, while the variances corresponding to those parameters, computed as in (10), were much higher than the rest of the parameters.

Conclusions

In this article, a reproducible comparison of some of the main dynamic identification algorithms of open-chain manipulators is reported. The overall objective of the identification process is to minimize the reconstruction error while meeting physical constraints on the identified parameters. The dynamic parameters extracted from CAD data

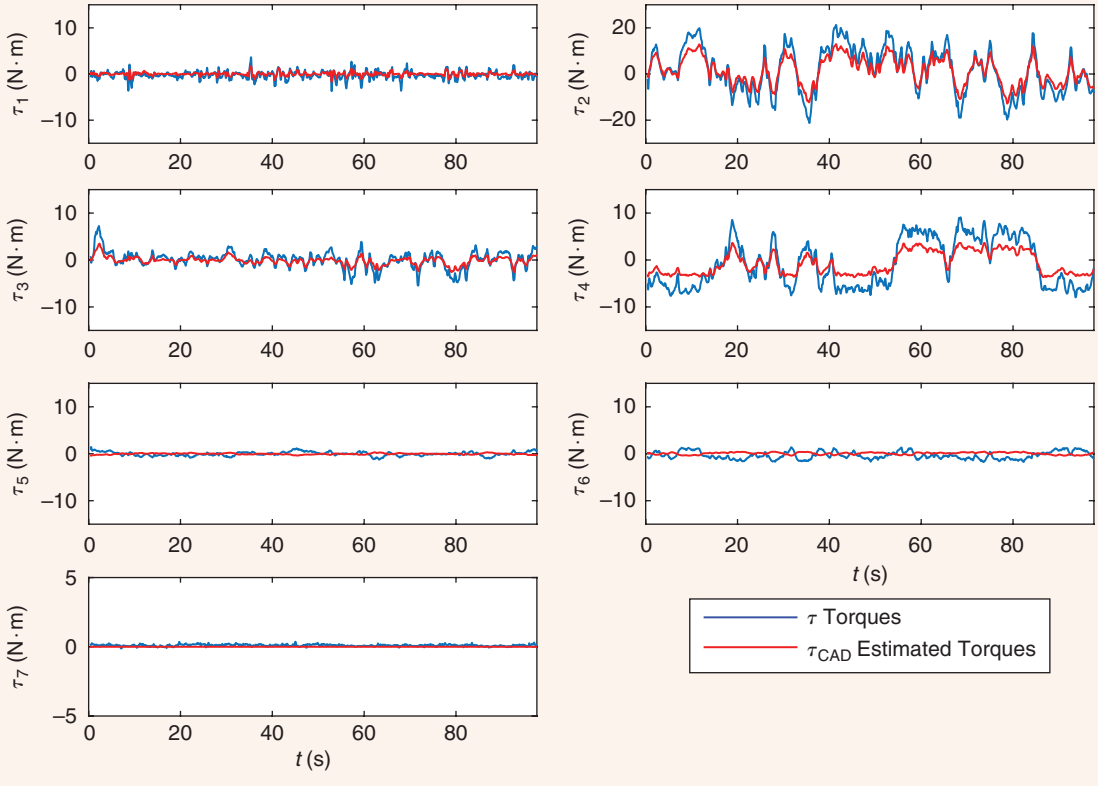


Figure 4. The CAD reconstruction errors along the third trajectory.

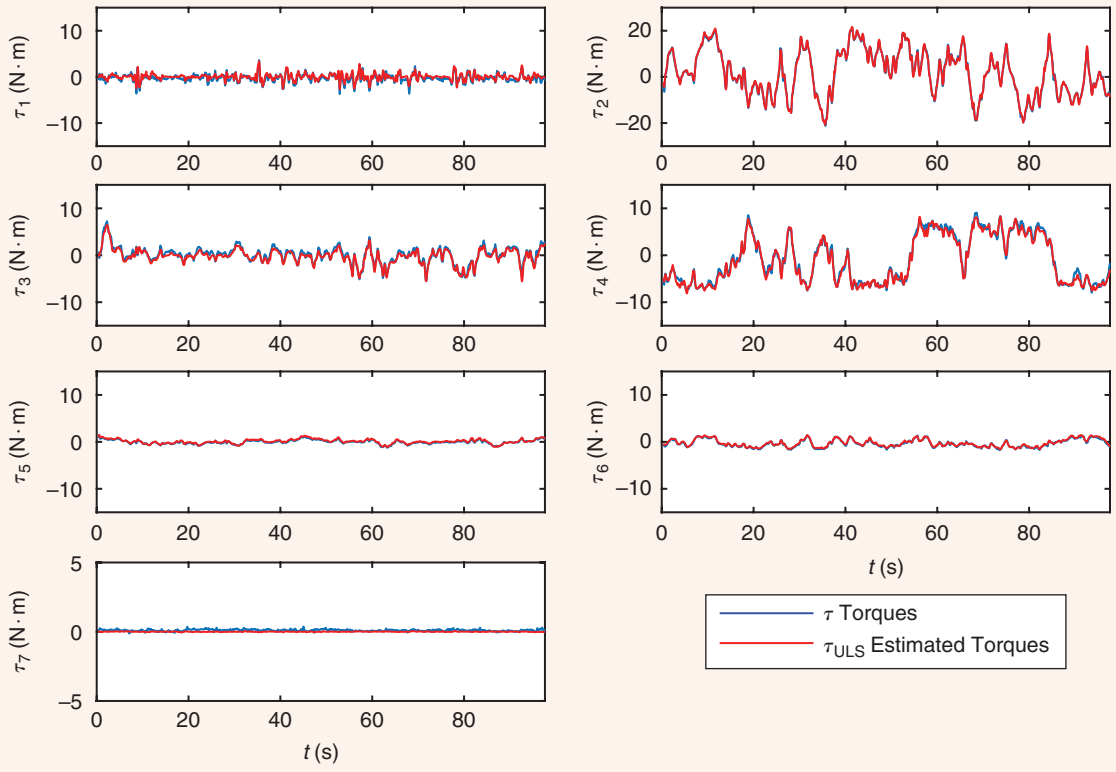


Figure 5. The ULS reconstruction errors along the third trajectory.

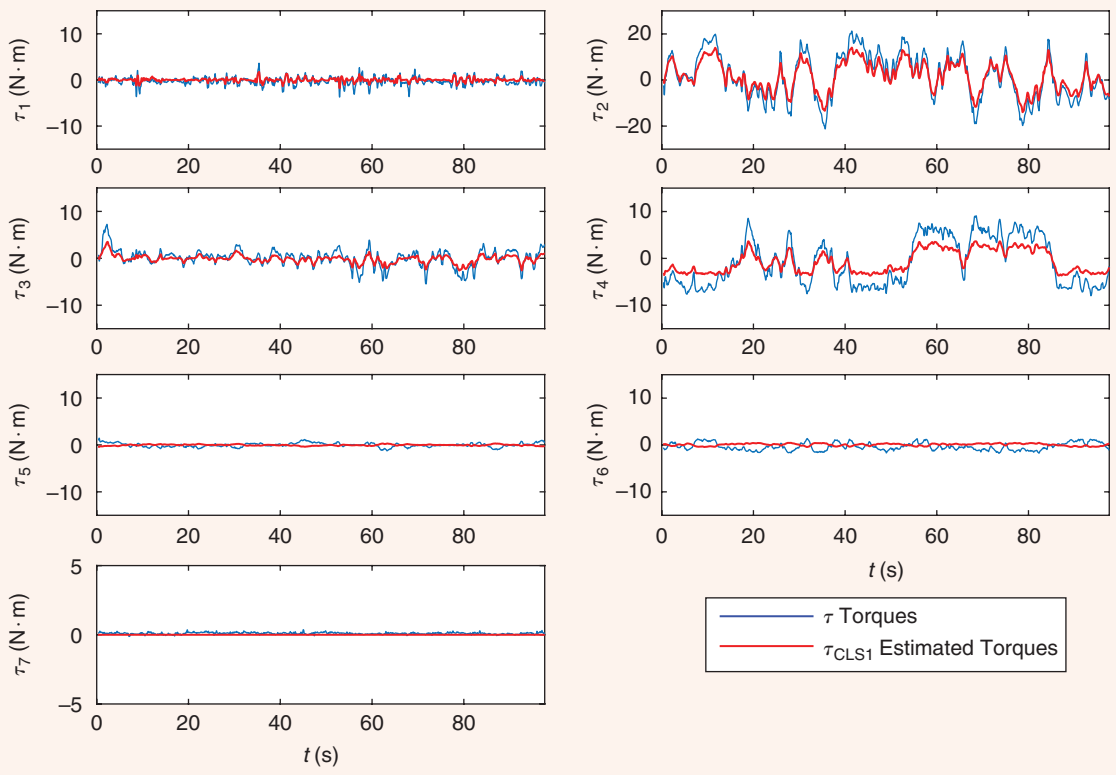


Figure 6. The CLS-1 reconstruction errors along the third trajectory.

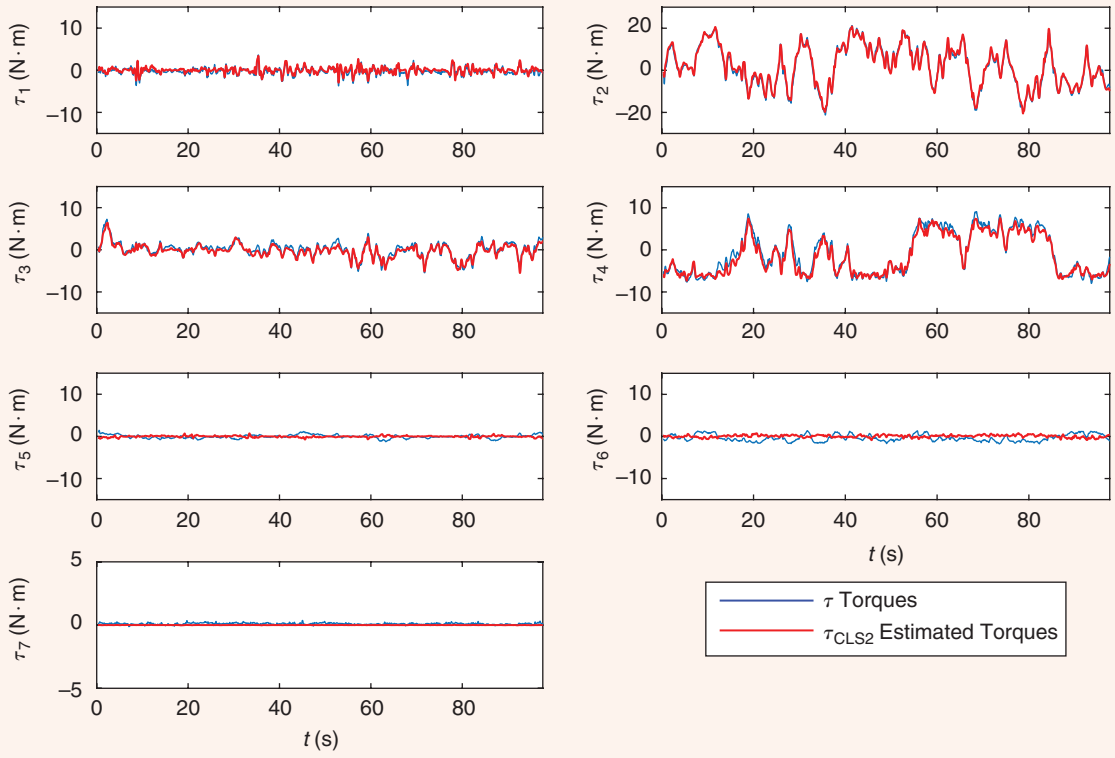


Figure 7. The CLS-2 reconstruction errors along the third trajectory.

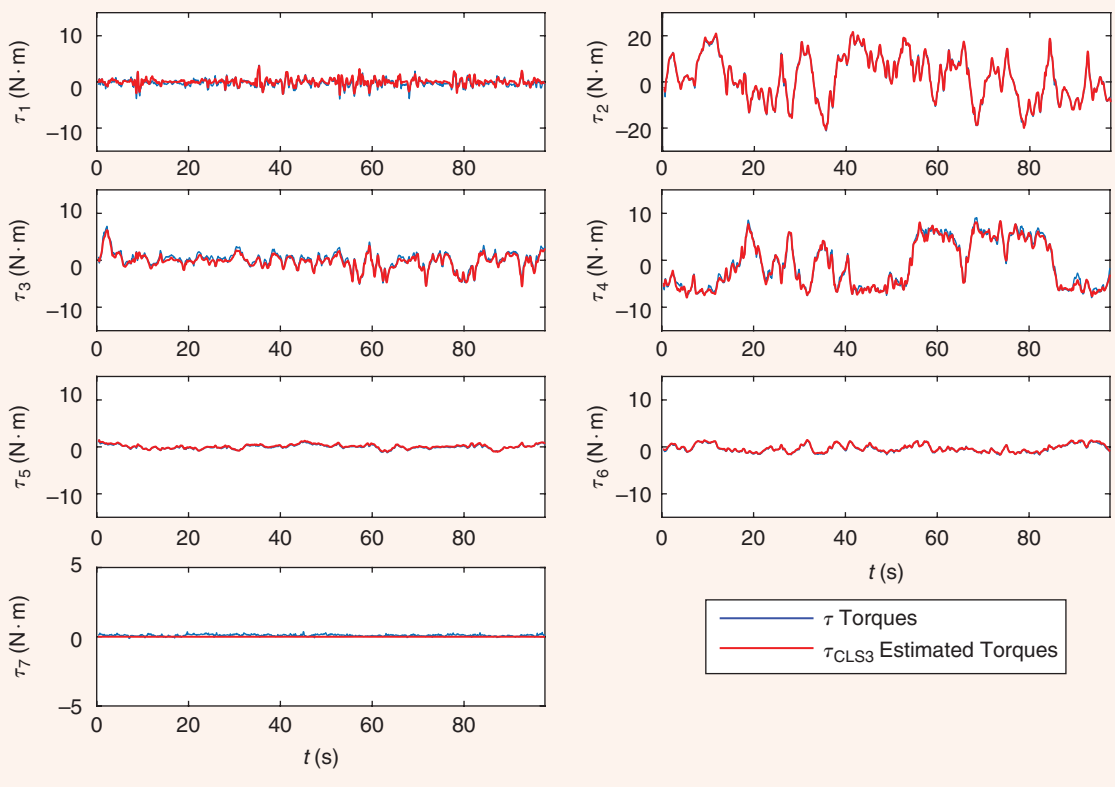


Figure 8. The CLS-3 reconstruction errors along the third trajectory.

are physically consistent but fail in providing an acceptable reconstruction error. The ULS method guarantees a small reconstruction error on the identification trajectories but suffers from overfitting and, in general, does not meet the physical constraints on the parameters.

For these reasons, constrained identification algorithms were considered. The CLS-1 algorithm explicitly considers bounds on the vectors of dynamic parameters; however, this does not guarantee that the physical consistency property is met. CLS-2 sets constraints on the sign of the inertia matrix and on the first moment of mass and outperforms CLS-1 concerning the reconstruction error. Finally, CLS-3 considers the full set of constraints while generally minimizing the reconstruction error. Moreover, the overall code used to carry out the comparison of the five algorithms considered in this article is made available to the community, together with the dynamic model of the Kinova Jaco^2 robot.

Table 5. A summary of the results.

	CAD		ULS		CLS-1		CLS-2		CLS-3	
Trajectory	ς [Nm]	ς_r	ς [Nm]	ς_r	ς [Nm]	ς_r	ς [Nm]	ς_r	ς [Nm]	ς_r
Identification: Traj 1	0.01694	0.39842	0.00302	0.07097	0.01465	0.34454	0.00724	0.17023	0.0031	0.07297
Validation: Traj 2	0.01947	0.39431	0.00537	0.10872	0.0169	0.34224	0.00665	0.13477	0.00526	0.10647
Validation: Traj 3	0.007	0.43991	0.00213	0.13407	0.00648	0.40725	0.00298	0.1874	0.00203	0.12791
Validation: Traj 4	0.00556	0.42289	0.00166	0.12647	0.00505	0.38421	0.00265	0.20172	0.00159	0.1206
Validation: Traj 5	0.00348	0.41734	0.0009	0.10735	0.0031	0.37147	0.00133	0.15957	0.00089	0.1071
Identification: Traj 2	0.01947	0.39431	0.00292	0.05909	0.01689	0.34222	0.00585	0.11841	0.00307	0.0621
Validation: Traj 1	0.01694	0.39842	0.00468	0.11017	0.01464	0.3443	0.00797	0.18741	0.00408	0.09599
Validation: Traj 3	0.007	0.43991	0.00166	0.10448	0.00648	0.40728	0.00287	0.18031	0.00154	0.09691
Validation: Traj 4	0.00556	0.42289	0.0015	0.11404	0.00505	0.3842	0.00256	0.19432	0.00129	0.09794
Validation: Traj 5	0.00348	0.41734	0.00078	0.09305	0.0031	0.37148	0.00134	0.16051	0.00071	0.08475
Identification: Traj 3	0.007	0.43991	0.00143	0.08963	0.00647	0.40677	0.00277	0.17403	0.00144	0.0903
Validation: Traj 1	0.01694	0.39842	0.00378	0.08884	0.01462	0.3439	0.00781	0.18363	0.0038	0.08928
Validation: Traj 2	0.01947	0.39431	0.00328	0.06651	0.01688	0.34183	0.0062	0.12555	0.00326	0.06609
Validation: Traj 4	0.00556	0.42289	0.00123	0.09381	0.00505	0.38369	0.00254	0.19269	0.00121	0.09225
Validation: Traj 5	0.00348	0.41734	0.00065	0.07827	0.00065	0.07827	0.00133	0.15934	0.00066	0.07852
Identification: Traj 4	0.00556	0.42289	0.00118	0.08982	0.00505	0.38382	0.00251	0.19105	0.00119	0.09044
Validation: Traj 1	0.01694	0.39842	0.00476	0.08842	0.01464	0.34425	0.00783	0.18409	0.00373	0.08766
Validation: Traj 2	0.01947	0.39431	0.00329	0.06663	0.01688	0.34195	0.00607	0.12303	0.0033	0.06678
Validation: Traj 3	0.007	0.43991	0.00148	0.09291	0.00647	0.40691	0.00281	0.17648	0.00146	0.09193
Validation: Traj 5	0.00348	0.41734	0.00066	0.07911	0.0031	0.37138	0.00134	0.16069	0.00066	0.07902
Identification: Traj 5	0.0048	0.41734	0.00062	0.07412	0.0031	0.37152	0.00132	0.1582	0.00064	0.07676
Validation: Traj 1	0.01694	0.39842	0.0053	0.12469	0.01465	0.34457	0.00744	0.17503	0.00372	0.08743
Validation: Traj 2	0.01947	0.39431	0.00429	0.08688	0.0169	0.34238	0.00636	0.12886	0.00323	0.0655
Validation: Traj 3	0.007	0.43991	0.00198	0.12472	0.00648	0.4074	0.00286	0.18003	0.00148	0.09326
Validation: Traj 4	0.00556	0.42289	0.0018	0.13693	0.00506	0.38436	0.00259	0.19691	0.00123	0.09333
Additional Criteria										
Physical feasibility (3)	Yes		No		No		Yes		Yes	
Triangle inequality (4)	Yes		No		No		No		Yes	
First moment (5) or (6)	Yes		No		Yes		Yes		Yes	

Cells with a red background indicate the largest error for a specific trajectory; those with a green background indicate the smallest. Traj: trajectory.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. New York: Springer-Verlag, 2009.

[2] E. Magrini, F. Flacco, and A. De Luca, "Estimation of contact forces using a virtual force sensor," in *Proc. 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 2126–2133. doi: 10.1109/IROS.2014.6942848.

[3] K. Yoshida and W. Khalil, "Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters," *Int. J. Robot. Res.*, vol. 19, no. 5, pp. 498–510, 2000. doi: 10.1177/02783640022066996.

[4] G. Antonelli, F. Caccavale, and P. Chiacchio, "A systematic procedure for the identification of dynamic parameters of robot manipulators," *Robotica*, vol. 17, no. 4, pp. 427–435, 1999. doi: 10.1017/S026357479900140X.

[5] J. Hollerbach, W. Khalil, and M. Gautier, *Model Identification*. Cham: Springer-Verlag, 2016, pp. 113–137.

[6] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans. Robot. Autom.*, vol. 13, no. 5, pp. 730–740, Oct. 1997. doi: 10.1109/70.631234.

[7] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 362–375, 1992. doi: 10.1177/027836499201100408.

[8] M. Gautier and G. Venture, "Identification of standard dynamic parameters of robots with positive definite inertia matrix," in *Proc. 2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 5815–5820. doi: 10.1109/IROS.2013.6697198.

[9] C. Gaz, F. Flacco, and A. De Luca, "Extracting feasible robot parameters from dynamic coefficients using nonlinear optimization methods," in *Proc. 2016 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2075–2081. doi: 10.1109/ICRA.2016.7487356.

[10] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4147–4154, 2019. doi: 10.1109/LRA.2019.2931248.

[11] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *Int. J. Robot. Res.*, vol. 33, no. 6, pp. 931–944, 2014. doi: 10.1177/0278364913514870.

[12] S. Traversaro, S. Brossette, A. Escande, and F. Nori, "Identification of fully physical consistent inertial parameters using optimization on manifolds," in *Proc. 2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 5446–5451. doi: 10.1109/IROS.2016.7759801.

[13] P. M. Wensing, S. Kim, and J.-J. E. Slotine, "Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 60–67, 2018. doi: 10.1109/LRA.2017.2729659.

[14] C. D. Sousa and R. Cortesão, "Inertia tensor properties in robot dynamics identification: A linear matrix inequality approach," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 1, pp. 406–411, Feb. 2019. doi: 10.1109/TMECH.2019.2891177.

[15] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the Kuka LWR robot using motor torques and joint torque sensors data," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 8391–8396, 2014. doi: 10.3182/20140824-6-ZA-1003.01079.

[16] C. Gaz, F. Flacco, and A. De Luca, "Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach," in *Proc. 2014 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 1386–1392. doi: 10.1109/ICRA.2014.6907033.

[17] F. Bonsignorio, J. Hallam, and A. del Pobil, "GEM guidelines: Euron GEM SIG report," Euron GEM Sig, Tech. Rep., 2008. [Online]. Available: http://www.heronrobots.com/EuronGEMSig/downloads/GemSigGuidelinesBeta.pdf

[18] M. Gautier, "Numerical calculation of the base inertial parameters of robots," *J. Robot. Syst.*, vol. 8, no. 4, pp. 485–506, 1991. doi: 10.1002/rob.4620080405.

[19] J. Jovic, A. Escande, K. Ayusawa, E. Yoshida, A. Kheddar, and G. Venture, "Humanoid and human inertia parameter identification using hierarchical optimization," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 726–735, June 2016. doi: 10.1109/TRO.2016.2558190.

[20] "Official ROS packages for Kinova robotic arms," Kinova, Boisbriand, Canada, 2019. [Online]. Available: https://github.com/Kinovarobotics/kinova-ros

[21] B. Armstrong, "On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics," *Int. J. Robot. Res.*, vol. 8, no. 6, pp. 28–48, 1989. doi: 10.1177/027836498900800603.

[22] C. Presse and M. Gautier, "New criteria of exciting trajectories for robot identification," in *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, pp. 907–912. doi: 10.1109/ROBOT.1993.292259.

[23] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli, "Robot dynamics identification: A reproducible comparison with experiment on KINOVA Jaco2," Code Ocean. Accessed on: 2020. [Online]. Available: https://codeocean.com/capsule/31dcd5d6-7a11-4002-87f5-4d904d13b98a

[24] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli, "Robot dynamics identification," IEEE *DataPort*. Accessed on: 2020. [Online]. Available: https://ieee-dataport.org/documents/robot-dynamics-identification

[25] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli, "Robot dynamics identification." Accessed on: 2020. [Online]. Available: http://webuser.unicas.it/lai/robotica/papers/RAMAdditionalMaterial.zip

[26] G. Golluccio, G. Gillini, A. Marino, and G. Antonelli, "Robot dynamics identification: A reproducible comparison with experiments on the Kinova Jaco2: Supplemental Information." Accessed on: 2020. [Online]. Available: http://webuser.unicas.it/lai/robotica/papers/IEEERAMSupplMaterial.pdf

Giacomo Golluccio, Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. Email: giacomo.golluccio@unicas.it.

Giuseppe Gillini, Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. Email: giuseppe.gillini@unicas.it.

Alessandro Marino, Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. Email: al.marino@unicas.it.

Gianluca Antonelli, Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. Email: antonelli@unicas.it.