Sequential semidefinite optimization for physically and statistically consistent robot identification

Alexandre Janot, Patrick M. Wensing

Sequential Semidefinite Optimization For Physically and Statistically Consistent Robot Identification

Alexandre Janot and Patrick M. Wensing

***Abstract*—This work considers the problem of dynamic identification for robotic mechanisms given noisy measurements of configuration variables and applied torques. Conventionally, this problem is solved via least-squares, exploiting linearity properties of the inverse dynamics model for rigid-body systems. However, the nonlinear dependency of this model on configurations and velocities gives rise to bias in the resultant estimates when using noisy or even filtered data. Further, these biases can cause parameters of best fit to be non-physical, potentially leading to an ill-posed forward dynamic model. The main contribution of this paper is to propose a sequential semidefinite optimization procedure to both 1) ensure the physical consistency of the identified model and 2) maintain the statistical consistency of the estimator. The new method validates both a direct and inverse dynamic identification model (DIDIM), and also ensures that intermediate iterates of the algorithm remain physically valid. Due to these favorable properties, the method is named a Physically-Consistent DIDIM (PC-DIDIM) approach. Recent statistical hypothesis tests for instrumental variable approaches are generalized for application with a PC-DIDIM approach. Experimental results with a six-degree-of-freedom industrial robot supported by Monte Carlo simulations show the effectiveness of the new method and robustness benefits in comparison to conventional least-squares and the vanilla DIDIM method.**

***Index Terms*—Calibration and identification, rigid-body dynamics, semidefinite programming.**

I. Introduction

System identification for the dynamic parameters of a robot is a classical problem, with results spanning recent decades, [1]–[7]. Dynamic identification methods traditionally assume that a kinematic model is known a priori, such that identification may focus on determination of inertial parameters (mass, center of mass, and rotational inertia), friction parameters, etc. that influence the relationship between applied forces and resultant accelerations. Recent years have witnessed a renew of interest in this problem [6], [8]–[12], due in part to a rapid increase in robotic hardware platforms capable of accurate model-based control (see e.g., [13]–[18]), and force-controlled actuators [19], [20].

Conventional approaches to the identification problem make use of the linearity of the inverse dynamic model (IDM) with respect to the unknown parameters. This allows identification to be formulated as a least-squares problem, [7]. The most popular identification method is referred to as the Inverse Dynamic Identification Model with Least-Squares estimation, (IDIM-LS method). The main challenge within this approach comes from characteristics of input data (e.g., motor position, velocity, acceleration, and torque), which are inherently noisy, discretely sampled, and often quantized. While proper data filtering may be applied to the data to minimize these effects, the resulting estimates remain biased (see, e.g., [21], [5], and references therein). Instrumental variable (IV) techniques, introduced by Reiersøl in the 1940's [22], offer a framework within which to pursue unbiased linear regression. IV techniques were originally applied for time-series prediction in Econometrics, but have gradually made their way into the robotics identification literature [23], [24] with more recent developments in [5].

Conventional least-squares methods only fit the inverse dynamic model (IDM) of the robot. While one may be tempted to claim this provides the forward model as well, since the two are mathematically related by the inversion of the mass matrix (see e.g. [7]), this is not always the case for inaccurate parameter estimates. Particularly in the case of noisy or incomplete data, it is possible that the identified model may be non-physical and violate this invertibility assumption. The IV approach developed in [5] is able to provide statistically consistent estimates in the case of noisy data and validates both the direct and inverse dynamic models. However, since the statistical consistency of the estimator is an asymptotic property, the physical consistency of the output is not guaranteed for any finite sample size. IV methods also require measurements of the joint angles from experiments and require the use of so-called base inertial parameters, and this may increase their barrier to entry.

As an alternative, Gautier and colleagues [21], [25] developed a method using both direct (forward) and inverse dynamics identification models (DIDIM). This DIDIM approach has close connections to IV approaches and performs comparably in many cases [26]. The main advantage of the DIDIM method over the IV approach lies in the fact it does not require the measurements of the joint kinematics, which means that the DIDIM method is a suitable framework when joint position, velocity, and/or acceleration measurements are noisy. Although measured torque data is also usually noisy, this noise represents less of an issue since zero mean torque noise does not bias the estimates. By contrast, noisy position data enters the estimation problem in a nonlinear manner, leading to estimation biases and presenting a fundamentally different challenge. Despite promising results, state-of-the-art DIDIM methods do not guarantee that the direct dynamic model (DDM) will be well-posed during its iterations.

Recent progress in the development of physical consistency constraints provides opportunities to address this deficiency for the DIDIM method. The set of all possible inertial param-

[1]A. Janot is with the French Aerospace Laboratory, Toulouse Cedex 4 31055, France Alexandre.Janot@onera.fr

[2]P. M. Wensing is with the Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA pwensing@nd.edu

Digital Object Identifier (DOI): To be assigned

eters ensuring a positive definite mass matrix is known to be convex [27], [28]. More recently, these conditions were shown to admit formulation as a set of 6×6 linear matrix inequalities (LMIs) posed over the 10 inertial parameters of each rigid body [29], [30]. However, these conditions alone are not sufficient to ensure a physically plausible set of parameters [6], [31]. Instead, a necessary and sufficient condition for physical plausibility was recently described using 4×4 LMIs [6] posed over the 10 inertial parameters of each rigid body. Owing to the Riemannian structure of the positive definite matrices [32], this connection has subsequently enabled novel regularization techniques to strategically bias parameter estimates, [10], [33]. The general approach of combining IDIM-LS and physical constraints is called the Physically-consistent IDIM-LS method (PC-IDIM-LS). Despite this recent activity, statistical properties of PC-IDIM-LS methods have thus far remained an open question in the literature.

The main contribution of the current work is to leverage sequential semidefinite optimization toward a DIDIM method with guarantees of physical consistency at both its intermediate iterates as well as in its final result. This physically consistent DIDIM method (PC-DIDIM) represents an identification strategy for robots that ensures both the physical and statistical consistency of the framework. In comparison to IV methods, the approach also has the benefit of being able to be used without requiring the use of base parameters, further lowering the barrier of adoption. As a second contribution, the statistical properties of the PC-IDIM-LS and PC-DIDIM methods are theoretically studied and validated through Monte Carlo simulations.

The remainder of the paper is structured as follows. Section II details background preliminaries for the identification of rigid-body robots. Section III provides a deep theoretical analysis of PC-IDIM-LS method while Section IV details our new Physically Consistent Direct and Inverse Dynamics Model (PC-DIDIM) identification method. Section V provides results obtained by running Monte Carlo simulations while Section VI provides details of an experimental validation of this method using a TX-40 industrial robot with results showing the performance of the PC-DIDIM method in comparison to previous approaches. Section VII provides concluding remarks.

II. Preliminaries

A. Notation

The equations of motion of a manipulator with n degrees of freedom can be written in the form [6], [34],

$$\mathbf{H}(\mathbf{q},\boldsymbol{\pi})\,\ddot{\mathbf{q}}+\mathbf{c}(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\pi})+\mathbf{g}(\mathbf{q},\boldsymbol{\pi})+\boldsymbol{\tau}_f(\dot{\mathbf{q}},\boldsymbol{\pi})=\boldsymbol{\tau}_{idm}\,, \quad (1)$$

where $\mathbf{q}\in\mathbb{R}^n$ the vector of configuration variables; $\mathbf{H}\in\mathbb{R}^{(n\times n)}$ the mass matrix; $\mathbf{c}\in\mathbb{R}^n$ a vector of Coriolis and centripetal terms; $\mathbf{g}\in\mathbb{R}^n$ the vector of gravity terms; $\boldsymbol{\tau}_f\in\mathbb{R}^n$ the vector of viscous and coulomb friction terms; and $\boldsymbol{\tau}_{idm}\in\mathbb{R}^n$ the vector of actuator torques. The vectors $\mathbf{c}$, $\mathbf{g}$ and $\boldsymbol{\tau}_f$ are conveniently regrouped in $\mathbf{n}(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\pi}) = \mathbf{c}(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\pi})+\mathbf{g}(\mathbf{q},\boldsymbol{\pi})+\boldsymbol{\tau}_f(\dot{\mathbf{q}},\boldsymbol{\pi})$. Assuming that each joint is a single degree of freedom, the vector $\boldsymbol{\pi}\in\mathbb{R}^{13n}$ collects dynamic parameters for each link

$$\boldsymbol{\pi}=[\boldsymbol{\pi}_1^T,\,\ldots,\,\boldsymbol{\pi}_n^T]^T\,, \quad (2)$$

where the dynamic parameters for each link are given by $\boldsymbol{\pi}_j = [m, MX, MY, MZ, XX, XY, XZ, YY, YZ, ZZ, I_a, f_v, f_c]_j^T$ with m its mass, $\boldsymbol{MS}_j=[MX,MY,MZ]^T$ its first mass moments, XX, XY, etc. its mass moments and products of inertia as entries of its inertia tensor $\boldsymbol{I}_j$, I_a the total rotational inertia of the rotor and gears for the preceding actuator, and f_v and f_c the viscous and Coulomb friction terms for the motor. The equations of motion (1) can be written in a linear form [1]

$$\boldsymbol{\tau}_{idm}=\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,\boldsymbol{\pi}\,, \quad (3)$$

where $\mathbf{Y}\in\mathbb{R}^{(n\times13n)}$ is the classical regressor matrix.

Not all of the dynamic parameters in $\boldsymbol{\pi}$ are identifiable, as some parameters do not appear in (1), or only appear in linear combinations with other parameters [1]. Any maximal linearly independent set of identifiable parameter combinations constitutes a choice of base parameters for a mechanism. A set of base parameters can be calculated analytically in some cases [35], numerically through SVD or QR decompositions applied to samples of $\mathbf{Y}$ [1], [36], or using recursive methods without sampling [37]. A vector specifying numeric values for the b base parameters $\underline{\boldsymbol{\pi}}\in\mathbb{R}^b$ can be used to fully specify the IDM with an associated base regressor $\underline{\mathbf{Y}}\in\mathbb{R}^{(n\times b)}$ according to

$$\boldsymbol{\tau}_{idm}=\underline{\mathbf{Y}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,\underline{\boldsymbol{\pi}}\,. \quad (4)$$

Appendix A details a method to form a base parameter set and develops the associated relationship between $\boldsymbol{\pi}$ and $\underline{\boldsymbol{\pi}}$. For any choice of base parameters, there exists a matrix $\overline{\mathbf{K}}\in\mathbb{R}^{(b\times13n)}$ relating the full and base parameters through $\underline{\boldsymbol{\pi}}=\overline{\mathbf{K}}\,\boldsymbol{\pi}$. Furthermore, it is assumed, without loss of generality, that $\boldsymbol{\pi}$ can be partitioned as $\boldsymbol{\pi}=\begin{bmatrix}\boldsymbol{\pi}_1^T & \boldsymbol{\pi}_2^T\end{bmatrix}^T$ such that $\underline{\boldsymbol{\pi}}=\boldsymbol{\pi}_1+\mathbf{K}\boldsymbol{\pi}_2$ for some matrix $\mathbf{K}\in\mathbb{R}^{(b\times(13n-b))}$. Details on the construction of $\overline{\mathbf{K}}$ and $\mathbf{K}$ are given in Appendix A.

Because of uncertainties (modeling errors and noise), the actual joint torques $\boldsymbol{\tau}\in\mathbb{R}^n$ differ from $\boldsymbol{\tau}_{idm}$ by a vector error $\boldsymbol{e}\in\mathbb{R}^n$. We obtain the Inverse Dynamic Identification Model (IDIM) given by

$$\boldsymbol{\tau}=\underline{\mathbf{Y}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,\underline{\boldsymbol{\pi}}+\boldsymbol{e}\,. \quad (5)$$

It is noted that in the automatic control community, relation (5) is often called *the data generating system*, see, e.g., [38] and [39]. The assumptions made on $\boldsymbol{e}$ are given later.

B. Data acquisition and controls

For safety considerations, robots are often identified under closed-loop control using linear controllers (e.g., PD, PID, etc.). The torque delivered by each motor τ_j is then given by

$$\tau_j=C_j(s)(q_{r_j}-q_{m_j})\,, \quad (6)$$

where $C_j(s)$ is the transfer function of the controller for joint j; q_{r_j} its reference; q_{m_j} the measured angle of the joint position; and s is the Laplace variable. From this process, the

data available for identification includes the measured angles $\mathbf{q}_m \in \mathbb{R}^n$ and input torques $\boldsymbol{\tau}$.

C. Robot Identification

In practical applications of identification, estimates $\hat{\mathbf{q}}$, $\hat{\dot{\mathbf{q}}}$, and $\hat{\ddot{\mathbf{q}}}$ are obtained by filtering measured angles $\mathbf{q}_m$ through a non-causal delay-free discrete filter. Details of this filtering process vary from application to application. Further, to eliminate high-frequency noise in $\boldsymbol{\tau}$, a subsequent decimation procedure is often used to filter and downsample both $\boldsymbol{\tau}$ and $\underline{\mathbf{Y}}$ (e.g., as in [21]).

After data acquisition and decimation down to n_e samples, the following overdetermined linear system is obtained

$$\mathbf{y}(\boldsymbol{\tau}) = \underline{\mathbf{X}}(\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})\, \underline{\boldsymbol{\pi}} + \boldsymbol{\varepsilon}\,, \tag{7}$$

where $\mathbf{y} \in \mathbb{R}^r$ is the vector of measurements built from decimated samples of the measured torques, with and $r = n \cdot n_e$. The matrix $\underline{\mathbf{X}} \in \mathbb{R}^{(r \times b)}$ is the observation matrix built from decimated samples of $\underline{\mathbf{Y}}(\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})$, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$ is a vector characterizing uncertainty resulting from the sampling of $\boldsymbol{e}$. Errors are assumed serially uncorrelated and with finite variance such that the covariance $\boldsymbol{\Omega}$ can be partitioned as

$$\boldsymbol{\Omega} = \operatorname{diag}\left(\sigma_1^2 \mathbf{I}_{n_e}, \ldots, \sigma_n^2 \mathbf{I}_{n_e}\right)\,. \tag{8}$$

Following the procedure in [21], each σ_j is estimated from the standard deviation of the error in a least-squares fit to τ_j alone.

Following this construction, the IDIM-LS estimates and their covariance are given by

$$\hat{\underline{\boldsymbol{\pi}}}_{LS} = \boldsymbol{\Sigma}_{LS}\, \underline{\mathbf{X}}^T \boldsymbol{\Omega}^{-1} \mathbf{y} \quad \text{and} \tag{9}$$

$$\boldsymbol{\Sigma}_{LS} = \left(\underline{\mathbf{X}}^T \boldsymbol{\Omega}^{-1} \underline{\mathbf{X}}\right)^{-1}\,. \tag{10}$$

In [40], the authors have considered noise models that are more complex than white noise. They suggest to carry out a pre-whitening process as it is usually done in system identification. They have also shown that the use of the decimating filter whitens the noise and is, therefore, equivalent to the process suggested in other work, see, e.g., [38], [39] and [41].

The IDIM-LS estimates are consistent if (see, e.g., [42])

$$E[\underline{\mathbf{X}}^T \boldsymbol{\varepsilon}] = \mathbf{0}\,, \tag{11}$$

where $E[\cdot]$ is the expectation operator. In the case when this condition holds, the least-squares estimator is a consistent estimator as is detailed explicitly below.

Definition 1 (Statistical Consistency of Base-Parameter Estimators)**.** *Consider a robot model* (4) *with base parameters* $\underline{\boldsymbol{\pi}}$*. Consider further an estimator producing a random variable* $\hat{\underline{\boldsymbol{\pi}}}_r$ *for* r *noisy samples of input data* $\mathbf{q}$ *and* $\boldsymbol{\tau}$*. The estimator is said to be statistically consistent if* $\hat{\underline{\boldsymbol{\pi}}}_r$ *converges to* $\underline{\boldsymbol{\pi}}$ *in probability as* $r \to \infty$*.*

In [43], the authors have shown that (11) leads to the following condition

$$\underline{\mathbf{X}} = \underline{\mathbf{X}}_{nf}\,, \tag{12}$$

where $\underline{\mathbf{X}}_{nf} \in \mathbb{R}^{(r \times b)}$ is a noise-free version of the observation matrix. $\underline{\mathbf{X}}_{nf}$ results from the sampling of $\underline{\mathbf{Y}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. Condition (12) is consistent with the result exposed in [41, pp. 153] and explains why researchers in robotics have developed several techniques of data filtering suitable for robot identification, see e.g., [2] and [21]. However, since robots are identified under closed-loop control, errors in measurements of $\mathbf{q}$ will appear in $\boldsymbol{\tau}$ via the feedback law (6), and thus, $\underline{\mathbf{X}}$ is most often correlated with $\boldsymbol{\varepsilon}$. Alternate methods may be pursued to overcome this bias-inducing correlation.

D. The standard DIDIM method

The direct and inverse dynamics identification method (DIDIM) is a closed-loop-input-error method that aims to remove bias from identification through the use of an offline noise-free simulation. The method aims to minimize the following quadratic criterion

$$J_{DIDIM}(\underline{\boldsymbol{\pi}}) = \sum_{i=1}^{r} \left\| \boldsymbol{\tau}(t_i) - \boldsymbol{\tau}_s(t_i, \underline{\boldsymbol{\pi}}) \right\|^2, \tag{13}$$

that captures the difference between the measured torques $\boldsymbol{\tau}$ and simulated torques $\boldsymbol{\tau}_s$, at a set of sample times $t_1, \ldots, t_r$. The method is completely detailed in [21], with the main steps summarized herein.

DIDIM is an iterative method, producing a sequence of estimates, with $\hat{\underline{\boldsymbol{\pi}}}^{(k)}$ denoting the estimate used for simulation at the k-th iteration of the algorithm. Because we focus on rigid robot identification, and since each simulation operates under closed-loop control, the sensitivity of simulation output to parameters $\underline{\boldsymbol{\pi}}$ is low, as shown in [21]. At each iteration, this implies that the approximation

$$\frac{\partial \boldsymbol{\tau}_s^{(k)}}{\partial \hat{\underline{\boldsymbol{\pi}}}^{(k)}} \approx \underline{\mathbf{Y}}\left(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \ddot{\mathbf{q}}_s^{(k)}\right)\,, \tag{14}$$

where $\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \ddot{\mathbf{q}}_s^{(k)}$ are the vectors of simulated joint positions, velocities and accelerations, respectively. Simulated accelerations $\ddot{\mathbf{q}}_s^{(k)}$ are obtained via the DDM

$$\ddot{\mathbf{q}}_s^{(k)} = \mathbf{H}\left(\mathbf{q}_s^{(k)}, \hat{\underline{\boldsymbol{\pi}}}^{(k)}\right)^{-1} \left[\boldsymbol{\tau}_s^{(k)} - \mathbf{n}(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \hat{\underline{\boldsymbol{\pi}}}^{(k)})\right]\,, \tag{15}$$

while the simulated torque at joint j is calculated with

$$\tau_{s_j} = C_j(s)(q_{r_j} - q_{s_j})\,. \tag{16}$$

Simulated joint positions and velocities are obtained via the numerical integration of (15). Note that $C_j(s)$ is usually known in robot identification, [21], but it can be identified by running the methodology described in [40] when necessary.

After data sampling and decimation, the following overdetermined system is obtained

$$\mathbf{y}(\boldsymbol{\tau}) = \underline{\mathbf{X}}_s(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \ddot{\mathbf{q}}_s^{(k)})\underline{\boldsymbol{\pi}} + \boldsymbol{\varepsilon}_{DIDIM}\,, \tag{17}$$

where $\underline{\mathbf{X}}_s(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \ddot{\mathbf{q}}_s^{(k)}) \in \mathbb{R}^{(r \times b)}$ is the observation matrix constructed with the simulated states resulting from the sampling of $\underline{\mathbf{Y}}\left(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \ddot{\mathbf{q}}_s^{(k)}\right)$; and $\boldsymbol{\varepsilon}_{DIDIM}$ accounts for both the approximation error from (14), as well as the measurement noise in $\boldsymbol{\tau}$. After simulation, a LS solution of (17) provides

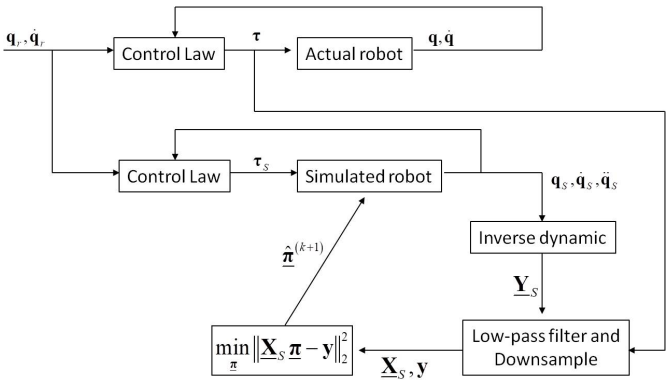

Fig. 1. Diagram for one iteration of the standard DIDIM method

parameters and their covariance matrix for the subsequent iteration

$$\begin{aligned} \hat{\underline{\boldsymbol{\pi}}}^{(k+1)} &= \left(\underline{\mathbf{X}}_s^T \boldsymbol{\Omega}^{-1} \underline{\mathbf{X}}_s\right)^{-1} \underline{\mathbf{X}}_s^T \boldsymbol{\Omega}^{-1} \mathbf{y}\,, \\ \boldsymbol{\Sigma}_{DIDIM} &= (\underline{\mathbf{X}}_s^T \boldsymbol{\Omega}^{-1} \underline{\mathbf{X}}_s)^{-1}\,. \end{aligned} \tag{18}$$

This process is iterated, with a single iteration diagrammed in Fig. 1. Because DIDIM is a LS-like identification method and according to (11), its estimates are consistent if $E[\underline{\mathbf{X}}_s^T \boldsymbol{\varepsilon}_{DIDIM}] = \mathbf{0}$. In Appendix B, we show that this relation holds if there is no modeling error and turns to $\underline{\mathbf{X}}_s = \underline{\mathbf{X}}_{nf}$. It is noted, that this DIDIM simulation procedure (15) requires the mass matrix $\mathbf{H}$ to be invertible along the simulated trajectory. The iteration procedure (18) does not, however, guarantee this requirement because it was not formally proven that the estimates of base parameters are physically consistent.

III. Theoretical analysis of PC-IDIM-LS method

A. Introduction of the physical consistency

In the standard IDIM-LS method, noise may bias parameters to the point that they are no longer physically meaningful. For instance, least-squares may assign a negative mass or rotational inertia that is not positive definite, see, e.g., [40]. Such results occur when the data filtering is not appropriate and/or data does not had adequate accuracy. Recent results [6] have provided linear matrix inequality (LMI) constraints that are both necessary and sufficient for the physical plausibility of inertial parameters $\boldsymbol{\pi}_j$. These results hinge on the so-called pseudo-inertia matrix $\boldsymbol{J}(\boldsymbol{\pi}_j) \in \mathbb{R}^{(4\times4)}$ of a rigid body as defined by :

$$\boldsymbol{J}(\boldsymbol{\pi}_j) = \begin{bmatrix} \frac{1}{2}\mathrm{tr}(\boldsymbol{I}_j)\mathbf{I}_3 - \boldsymbol{I}_j & \boldsymbol{MS}_j \\ \boldsymbol{MS}_j^T & m_j \end{bmatrix}, \tag{19}$$

where $\mathbf{I}_3$ is the (3×3) identity matrix, and again, $\boldsymbol{MS}_j$ represents the first-mass moment described after (2).

Definition 2 (Physical Consistency)**.** *A set of body dynamic parameters $\boldsymbol{\pi}_j \in \mathbb{R}^{13}$ is said to be physically consistent if there exists a rigid body with mass m_j, first mass moment $\boldsymbol{MS}_j$, and inertia tensor $\boldsymbol{I}_j$.*

Proposition 1 (An LMI for Physical Consistency [6])**.** *The parameters $\boldsymbol{\pi}_j$ are physically consistent if and only if*

$$\boldsymbol{J}(\boldsymbol{\pi}_j) \succ \mathbf{0}\,. \tag{20}$$

The expression "$\succ \mathbf{0}$" is used to indicate that the matrix is positive definite. Because the constraint (20) is only valid with the standard parameters $\boldsymbol{\pi}$, we deal now with them instead of the base parameters $\underline{\boldsymbol{\pi}}$.

This development allows a Physically-Consistent IDIM-LS problem (PC-IDIM-LS) to be formulated as a semidefinite program (SDP)

$$\min_{\hat{\boldsymbol{\pi}}} \ \|\mathbf{X}\hat{\boldsymbol{\pi}} - \mathbf{y}\|_2^2 \tag{21}$$

$$\begin{aligned} \text{s.t. } & \boldsymbol{J}(\hat{\boldsymbol{\pi}}_j) \succ \mathbf{0}, I_a(\hat{\boldsymbol{\pi}}_j) > 0\,, \\ & f_v(\hat{\boldsymbol{\pi}}_j) > 0, f_c(\hat{\boldsymbol{\pi}}_j) > 0\,, \quad \forall j\,, \end{aligned} \tag{22}$$

where $\mathbf{X}$ is the corresponding IDIM for the full parameters resulting from the sampling of $\mathbf{Y}$, and, again, f_c and f_v are Coulomb and viscous friction terms for the motor. The PC-IDIM-LS solution is denoted $\hat{\boldsymbol{\pi}}_{PC-IDIM-LS}$, and the final residual from the PC-IDIM-LS method is given by $\hat{\boldsymbol{\varepsilon}}_{PC-IDIM-LS} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\pi}}_{PC-IDIMLS}$. The optimization problem (21) can be solved with state-of-the-art primal-dual interior-point algorithms (e.g., [44]) that utilize a path-following paradigm. A complete description of the path-following paradigm is provided in [44]. Here, the optimization problem (21) is specified in the form above using CVX [45] and solved using its integrated semidefinite optimization packages. The following subsections consider optimality criteria that must be satisfied at the solution to this problem.

B. Characterization of semidefiniteness

Let $\boldsymbol{M} \in \mathbb{R}^{(m\times m)}$ be a positive semidefinite matrix, denoted as $\boldsymbol{M} \succeq \mathbf{0}$. The first characterization is that its eigenvalues are nonnegative, i.e., $\lambda_i \geq 0$ for $i = 1, \cdots, m$. Likewise, $\boldsymbol{M} \succ \mathbf{0}$ if its eigenvalues are positive, i.e., $\lambda_i > 0$ for $i = 1, \cdots, m$. Another fact is that any positive semidefinite matrix $\boldsymbol{M}$ can be decomposed as

$$\boldsymbol{M} = \boldsymbol{L}_M \boldsymbol{D}_M \boldsymbol{L}_M^T\,, \tag{23}$$

where $\boldsymbol{L}_M \in \mathbb{R}^{(m\times m)}$ is a lower triangular matrix; and $\boldsymbol{D}_M \in \mathbb{R}^{(m\times m)}$ is a non-negative diagonal matrix whose i-th diagonal element is denoted d_i. We recall that $\boldsymbol{D}_M$ is unique for a positive semidefinite matrix while $\boldsymbol{L}_M$ is unique only when $\boldsymbol{M}$ is positive definite, see [46]. It follows that if a Non-Linear Programming (NLP) algorithm involves semi-positive definite matrix $\boldsymbol{M}$, then the following constraints $d_i(\boldsymbol{M}) \geq 0$ for $i = 1, \cdots, m$, can be used in place of $\boldsymbol{M} \succeq \mathbf{0}$ [46]. In other words, a positive semidefinite constraint can be expressed using standard inequality constraints. It follows that for robot identification one obtains

$$\boldsymbol{J}(\boldsymbol{\pi}_j) \succ \mathbf{0} \Leftrightarrow \boldsymbol{h}_{\boldsymbol{J}_j}(\boldsymbol{\pi}_j) \geq \mathbf{0}\,, \text{ for } j = 1, \cdots, n\,, \tag{24}$$

where $\boldsymbol{h}_{\boldsymbol{J}_j} \in \mathbb{R}^4$ is a vector of continuous functions of $\boldsymbol{\pi}_j$ that are twice differentiable and concave, see [46]. Note that we have $\boldsymbol{h}_{\boldsymbol{J}_j} \in \mathbb{R}^4$ because one has $\boldsymbol{J}(\boldsymbol{\pi}_j) \in \mathbb{R}^{(4\times4)}$. However, for ease and clarity, we introduce the following matrix

$\boldsymbol{D_J} = \text{diag}\,(\boldsymbol{J}(\boldsymbol{\pi}_1), f_{v_1}, f_{c_1}, I_{a_1}, \cdots, \boldsymbol{J}(\boldsymbol{\pi}_n), f_{v_n}, f_{c_n}, I_{a_n})$ and the following constraint $h_i(\boldsymbol{\pi}) = \lambda_i \geq 0$, where λ_i is the i-th diagonal element of $\boldsymbol{D}(\boldsymbol{D_J})$ resulting from the LDL^T decomposition of $\boldsymbol{D_J}$. Since each λ_i is a function of $\boldsymbol{\pi}$, the following constraint will be used for characterizing the semidefiniteness of the n pseudo-inertia matrices $\boldsymbol{J}(\boldsymbol{\pi}_j)$, as well as the positiveness of the n viscous coefficients, Coulomb coefficients, and actuators inertias

$$h_i(\boldsymbol{\pi}) \geq 0\,, \tag{25}$$

where $i = 1, ..., p$ with $p = 7 \cdot n$.

C. Problem formulation

According to relations (21) and (24), the problem of physically consistent robot identification is equivalent to an inequality constrained programming problem formulated as

$$\begin{aligned} &\text{minimize} \quad f_0(\boldsymbol{\pi})\,, \\ &\text{subject to} \quad h_i(\boldsymbol{\pi}) \geq 0\,, \ \text{for } i = 1, \cdots, p\,, \end{aligned} \tag{26}$$

where $f_0(\boldsymbol{\pi}) = (1/2r)||\boldsymbol{\varepsilon}||_2^2 = (1/2r)||\mathbf{y} - \mathbf{X}\boldsymbol{\pi}||_2^2 = (1/2)\boldsymbol{\pi}^T\boldsymbol{Q}_{XX}\boldsymbol{\pi} - \boldsymbol{c}^T\boldsymbol{\pi} + (1/2)\alpha$; $\boldsymbol{Q}_{XX} = (1/r)\mathbf{X}^T\mathbf{X} \succeq \mathbf{0}$; $\boldsymbol{c} \in \mathbb{R}^c = (1/r)\mathbf{X}^T\mathbf{y}$; $c = 13n$; and $\alpha = (1/r)\mathbf{y}^T\mathbf{y}$ is a scalar. Compared to (21), the objective is scaled by $1/r$ such that $\boldsymbol{Q}_{XX}$ and $\boldsymbol{c}$ remain bounded in the limit as the number of samples $r \to \infty$. The Karush Kuhn Tucker (KKT) conditions for this problem can be stated as follows: if $\boldsymbol{\pi}^*$ is an optimizer for (26), there exists a multiplier $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that

$$\begin{aligned} \boldsymbol{Q}_{XX}\boldsymbol{\pi}^* - \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\mu}^* &= \mathbf{0}\,, \\ h_i(\boldsymbol{\pi}^*) &\geq 0\,, \ \text{for } i = 1 \cdots p\,, \\ \mu_i^* h_i(\boldsymbol{\pi}^*) &= 0 \ \text{ for } i = 1 \cdots p\,, \\ \mu_i^* &\geq 0 \ \text{ for } i = 1 \cdots p\,, \end{aligned} \tag{27}$$

where $\boldsymbol{A} \in \mathbb{R}^{(p \times c)}$ is the Jacobian matrix of $\boldsymbol{h}(\boldsymbol{\pi}) = \left[h_1^T \cdots h_p^T\right]^T$. It is assumed that $\text{rank}(\boldsymbol{A}) = p$. By introducing the slack variable $\boldsymbol{s} \geq \mathbf{0}$ with $\boldsymbol{s} \in \mathbb{R}^p$, the above conditions can be equivalently formulated as follows

$$\begin{aligned} \boldsymbol{Q}_{XX}\boldsymbol{\pi} - \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\mu} &= \mathbf{0}\,, \\ \boldsymbol{h}(\boldsymbol{\pi}) - \boldsymbol{s} &= \mathbf{0}\,, \\ s_i\mu_i &= 0 \ \text{ for } i = 1 \cdots p\,, \\ s_i, \mu_i &\geq 0 \ \text{ for } i = 1 \cdots p\,. \end{aligned} \tag{28}$$

An interior-point strategy is adopted to solve the constraint satisfaction problem described by (27),. This choice is motivated by the fact that such methods are very popular and are easily accessible though multiple solvers used by CVX [45].

D. Problem resolution

The strategy is to find the optimal triplet $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{s})$ satisfying the KKT conditions given by (28) by running an interior-point method whose main principles are sketched here to support the subsequent statistical analysis. First, the nonnegativity constraints are replaced with logarithmic barriers such that $f_0(\boldsymbol{\pi})$ becomes $f_0(\boldsymbol{\pi}) - \zeta \sum_{i=1}^p \log(s_i)$, where ζ is the barrier parameter that has a small value. Second, we write the Lagrangian

$$L_\zeta(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{s}) = f_0(\boldsymbol{\pi}) - \zeta \sum_{i=1}^p \log(s_i) - \sum_{i=1}^p \mu_i(h_i(\boldsymbol{\pi}) - s_i)\,, \tag{29}$$

which plays an important role in the optimality conditions

$$\begin{aligned} \nabla_{\boldsymbol{\pi}} L_\zeta = \partial L_\zeta / \partial \boldsymbol{\pi} &= \boldsymbol{Q}_{XX}\boldsymbol{\pi} - \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{\mu} = \mathbf{0}\,, \\ \nabla_{\boldsymbol{\mu}} L_\zeta = \partial L_\zeta / \partial \boldsymbol{\mu} &= \boldsymbol{h}(\boldsymbol{\pi}) - \boldsymbol{s} = \mathbf{0}\,, \\ \nabla_{\boldsymbol{s}} L_\zeta = \partial L_\zeta / \partial \boldsymbol{s} &= -\zeta \boldsymbol{D}_s^{-1}\mathbf{1}_p + \boldsymbol{\mu} = \mathbf{0}\,, \end{aligned} \tag{30}$$

where $\boldsymbol{D}_s \in \mathbb{R}^{(p \times p)}$ is a diagonal matrix defined by $\boldsymbol{D}_s = \text{diag}(s_1, \cdots, s_p)$; and $\mathbf{1}_p \in \mathbb{R}^p$ is a vector of ones defined by $\mathbf{1}_p = (1, \cdots, 1)^T$. Note that $\boldsymbol{D}_s$ is positive definite since one has $s_i > 0$ for compatibility with the logarithmic barrier. Third, by left multiplying $\nabla_{\boldsymbol{s}} L_\zeta$ by $\boldsymbol{D}_s$, the following transformed criteria are obtained

$$F_\zeta\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{s}\right) = \begin{bmatrix} \boldsymbol{Q}_{XX}\boldsymbol{\pi} + \boldsymbol{A}^T\boldsymbol{\mu} - \boldsymbol{c} \\ \boldsymbol{h}(\boldsymbol{\pi}) - \boldsymbol{s} \\ \boldsymbol{D}_s\boldsymbol{D}_\mu\mathbf{1}_p - \zeta\mathbf{1}_p \end{bmatrix} = \mathbf{0}\,, \tag{31}$$

where $\boldsymbol{D}_\mu \in \mathbb{R}^{(p \times p)}$ is a diagonal matrix defined by $\boldsymbol{D}_\mu = \text{diag}(\mu_1, \cdots, \mu_p)$. Note that the difference between (31) and (28) is the presence of $\zeta\mathbf{1}_p$ such that the last equation in (31) represents a perturbed complementarity condition. Further, the Lagrange multipliers are given by $\mu_i^* = \zeta / s_i$ for $i = 1, \cdots, p$.

The idea is to apply Newton's method to (31) to compute the optimal triplet $(\boldsymbol{\pi}_\zeta, \boldsymbol{\mu}_\zeta, \boldsymbol{s}_\zeta)$ on the central path, and automatically compute ζ with $\zeta = \varsigma\kappa$ where $\varsigma \in [0, 1]$ is a parameter chosen by the algorithm, and κ is a duality measure introduced later, [47]. At iteration k, the Newton increments $(\Delta\boldsymbol{\pi}^{(k)}, \Delta\boldsymbol{\mu}^{(k)}, \Delta\boldsymbol{s}^{(k)})$ are the solution of the linear system

$$\begin{bmatrix} \boldsymbol{H}_{kkt} & \boldsymbol{A}^T & \mathbf{0} \\ \boldsymbol{A} & \mathbf{0} & \boldsymbol{I}_p \\ \mathbf{0} & \boldsymbol{D}_s & \boldsymbol{D}_\mu \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\pi}^{(k)} \\ \Delta\boldsymbol{\mu}^{(k)} \\ \Delta\boldsymbol{s}^{(k)} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{r}_c \\ -\boldsymbol{r}_b \\ \boldsymbol{g}_s \end{bmatrix} \tag{32}$$

where $\boldsymbol{H}_{kkt}(\boldsymbol{\pi}^{(k)}, \boldsymbol{\mu}^{(k)}) = \boldsymbol{Q}_{XX} + \sum_{i=1}^p \mu_i \nabla_{\boldsymbol{\pi}}^2 h_i(\boldsymbol{\pi}^{(k)})$ where $\nabla_{\boldsymbol{\pi}}^2 h_i(\boldsymbol{\pi}^{(k)}) \in \mathbb{R}^{(c \times c)}$ is the Hessian matrix of $h_i(\boldsymbol{\pi})$ evaluated at $\boldsymbol{\pi}^{(k)}$; $\boldsymbol{r}_c = \boldsymbol{Q}_{XX}\boldsymbol{\pi}^{(k)} + \boldsymbol{A}^T\boldsymbol{\mu}^{(k)} - \boldsymbol{c}$; $\boldsymbol{r}_b = \boldsymbol{h}(\boldsymbol{\pi}^{(k)}) - \boldsymbol{s}^{(k)}$; and $\boldsymbol{g}_s = \boldsymbol{D}_s\boldsymbol{D}_\mu\mathbf{1}_p - \varsigma\kappa\mathbf{1}_p$.

To compute κ in (32), it is recalled that the set of points $(\boldsymbol{\pi}_\zeta, \boldsymbol{\mu}_\zeta, \boldsymbol{s}_\zeta)$ satisfying

$$F_\zeta(\boldsymbol{\pi}_\zeta, \boldsymbol{\mu}_\zeta, \boldsymbol{s}_\zeta) = \mathbf{0}\,, \tag{33}$$

(for some ζ) is called the central path, [47]. The new iterate $(\boldsymbol{\pi}_\zeta^{(k+1)}, \boldsymbol{\mu}_\zeta^{(k+1)}, \boldsymbol{s}_\zeta^{(k+1)})$ is thus determined by means of $(\boldsymbol{\pi}_\zeta^{(k+1)}, \boldsymbol{\mu}_\zeta^{(k+1)}, \boldsymbol{s}_\zeta^{(k+1)}) = (\boldsymbol{\pi}_\zeta^{(k)}, \boldsymbol{\mu}_\zeta^{(k)}, \boldsymbol{s}_\zeta^{(k)}) + \alpha(\Delta\boldsymbol{\pi}^{(k)}, \Delta\boldsymbol{\mu}^{(k)}, \Delta\boldsymbol{s}^{(k)})$, with α chosen such that $(\boldsymbol{\pi}_\zeta^{(k+1)}, \boldsymbol{\mu}_\zeta^{(k+1)}, \boldsymbol{s}_\zeta^{(k+1)})$ stays feasible. Given a feasible iterate $(\boldsymbol{\pi}_\zeta^{(k)}, \boldsymbol{\mu}_\zeta^{(k)}, \boldsymbol{s}_\zeta^{(k)})$, the duality measure κ is calculated according to $\kappa = \frac{1}{p}\sum_{i=1}^p s_\zeta^{(k)}(i)\mu_\zeta^{(k)}(i) = \frac{(\boldsymbol{s}_\zeta^{(k)})^T\boldsymbol{\mu}_\zeta^{(k)}}{p}$.

We now proceed to derive the iteration step. Since $\boldsymbol{D}_\mu$ is a nonsingular diagonal matrix by definition, the increment $\Delta\boldsymbol{s}^{(k)}$ in the slack variable can be easily eliminated with

$\boldsymbol{D}_s\Delta\boldsymbol{\mu}^{(k)} + \boldsymbol{D}_\mu\Delta\boldsymbol{s}^{(k)} = \boldsymbol{g}_s$, yielding

$$\Delta\boldsymbol{s}^{(k)} = \boldsymbol{D}_\mu^{-1}\boldsymbol{g}_s - \boldsymbol{D}_\mu^{-1}\boldsymbol{D}_s\Delta\boldsymbol{\mu}^{(k)}\,, \tag{34}$$

resulting in

$$\begin{bmatrix}\boldsymbol{H}_{kkt} & \boldsymbol{A}^T\\ \boldsymbol{A} & -\boldsymbol{D}_\mu^{-1}\boldsymbol{D}_s\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{\pi}^{(k)}\\ \Delta\boldsymbol{\mu}^{(k)}\end{bmatrix} = \begin{bmatrix}-\boldsymbol{r}_c\\ -\boldsymbol{r}_b - \boldsymbol{D}_\mu^{-1}\boldsymbol{g}_s\end{bmatrix}. \tag{35}$$

Then, $\boldsymbol{D}_s$ being nonsingular by definition, $\Delta\boldsymbol{\mu}^{(k)}$ can be eliminated with

$$\Delta\boldsymbol{\mu}^{(k)} = \boldsymbol{D}_s^{-1}\boldsymbol{D}_\mu\boldsymbol{r}_b + \boldsymbol{D}_s^{-1}\boldsymbol{g}_s + \boldsymbol{D}_s^{-1}\boldsymbol{D}_\mu\boldsymbol{A}\Delta\boldsymbol{\pi}^{(k)}\,, \tag{36}$$

in order to obtain

$$\Delta\boldsymbol{\pi}^{(k)} = -\boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\left(\boldsymbol{r}_c + \boldsymbol{r}_{feas} + \boldsymbol{r}_{cent}\right), \tag{37}$$

with $\boldsymbol{N}_{\boldsymbol{\pi}} \in \mathbb{R}^{(c\times c)} = \boldsymbol{H}_{kkt} + \boldsymbol{A}^T\boldsymbol{D}_s^{-1}\boldsymbol{D}_\mu\boldsymbol{A}$; $\boldsymbol{r}_{feas} = (\boldsymbol{A}^T\boldsymbol{D}_s^{-1}\boldsymbol{D}_\mu\boldsymbol{r}_b)$; and $\boldsymbol{r}_{cent} = (\boldsymbol{A}^T\boldsymbol{D}_s^{-1}\boldsymbol{g}_s)$. Note that $\boldsymbol{N}_{\boldsymbol{\pi}}$ is positive definite as follows. First, $\boldsymbol{D}_s$ and $\boldsymbol{D}_\mu$ are positive definite while $\boldsymbol{A}$ is full rank yielding $\boldsymbol{A}^T\boldsymbol{D}_s^{-1}\boldsymbol{D}_\mu\boldsymbol{A} \succeq \boldsymbol{0}$. Then, with $\boldsymbol{Q}_{XX} \succeq \boldsymbol{0}$ and $\sum_{i=1}^{p}\mu_i\nabla^2_{\boldsymbol{\pi}}\boldsymbol{h}_i(\boldsymbol{\pi}^{(k)}) \succ \boldsymbol{0}$, one finally obtains $\boldsymbol{N}_{\boldsymbol{\pi}} \succ \boldsymbol{0}$. Interestingly, we note that $\Delta\boldsymbol{\pi}$ is composed of three terms. The first component, $-\boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\boldsymbol{r}_c$, is the element of the step direction, the second one $-\boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\boldsymbol{r}_{feas}$ aims towards feasibility while the last one $-\boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\boldsymbol{r}_{cent}$ provides centering.

Working backwards through (37), (36) and (34), we solve for $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\mu}$ and $\Delta\boldsymbol{s}$. This process can be iterated until its convergence. It follows that this iterative algorithm must be initialized with values that keep the properties of physical consistency. This can be accomplished by choosing the Computer-Aided-Design (CAD) values of $\boldsymbol{\pi}$.

E. Addressing the statistical consistency of PC-IDIM-LS

This subsection now proceeds to analyze the behavior of the previous iteration process in the case of noisy data, and compares it to the noise free case. We consider a single iteration of the algorithm and compare the iteration with and without noise. We note by $\hat{\boldsymbol{\pi}}^{(k)}$ and $\boldsymbol{\pi}^{(k)}$ the iterate of the algorithm with noisy and noise-free data respectively, and we consider the case where these estimates start equivalent, and likewise that noise free and noisy estimates for $\boldsymbol{\mu}$ and $\boldsymbol{s}$ start equivalent. Denote by $\Delta\hat{\boldsymbol{\pi}}^{(k)}$ and $\Delta\boldsymbol{\pi}^{(k)}$ the change in variables over the iteration. We leave $\boldsymbol{\pi}$ itself to represent the true parameters and $\hat{\boldsymbol{e}}_{\boldsymbol{\pi}}^{(k)} = \hat{\boldsymbol{\pi}}^{(k)} - \boldsymbol{\pi}$ to represent the parameter error with noisy data and $\boldsymbol{e}_{\boldsymbol{\pi}}^{(k)} = \boldsymbol{\pi}^{(k)} - \boldsymbol{\pi}$ with noise-free data. We assume that the data is maximally exciting [2] with $\mathrm{rank}(\mathbf{X}^T\mathbf{X}) = b$. Noting that it is ultimately only the base parameters that affect a dynamic model, this assumption implies that noise-free data provides $\lim_{k\to\infty}\overline{\mathbf{K}}\boldsymbol{\pi}^{(k)} = \underline{\boldsymbol{\pi}}$. The main goal of this section will be to derive conditions under which $\hat{\boldsymbol{e}}_{\boldsymbol{\pi}}^{(k+1)}$ converges in probability to the noise free version $\boldsymbol{e}_{\boldsymbol{\pi}}^{(k+1)}$. This result will then imply that $\hat{\boldsymbol{\pi}} = \lim_{k\to\infty}\hat{\boldsymbol{\pi}}^{(k)}$ provides a statistically consistent estimate of the base parameters.

To proceed with the statistical analysis, we consider a single iteration from k to $k+1$ of the above algorithm such that dependence of $\boldsymbol{N}_{\boldsymbol{\pi}}$ on the iteration is dropped for simplicity of notation. We assume that the noisy version $\hat{\boldsymbol{N}}_{\boldsymbol{\pi}}$ converges in probability to the finite constant $E[\hat{\boldsymbol{N}}_{\boldsymbol{\pi}}] = \boldsymbol{N}_{\boldsymbol{\pi}} \in \mathbb{R}^{(c\times c)}$ as the number of samples $r \to \infty$. We denote this convergence in probability via

$$\operatorname*{plim}_{r\to\infty}\hat{\boldsymbol{N}}_{\boldsymbol{\pi}} = \boldsymbol{N}_{\boldsymbol{\pi}}$$

Hence, by applying the continuous mapping theorem, [41], one obtains that

$$\operatorname*{plim}_{r\to\infty}\hat{\boldsymbol{N}}_{\boldsymbol{\pi}}^{-1} = E[\hat{\boldsymbol{N}}_{\boldsymbol{\pi}}]^{-1} = \boldsymbol{N}_{\boldsymbol{\pi}}^{-1}.$$

These assumptions are reasonable in robot identification for the following reasons. First, as robots are often position controlled, the deterministic components of joint positions, velocities, accelerations are bounded by definition. Second, when robots are operating in normal conditions, the variances of the signals are finite. Finally, the signals can be considered as quasi-stationary over the duration of the identification process so that Theorem 2.3 in [38, page 43] applies. A formal proof would be more rigorous, but such analysis is beyond the scope of this paper and will be dealt with in a later publication.

Finding the conditions ensuring the consistency of (37) is not as easy as with usual linear systems. First, we start with a linear-in-parameter system to finally deal with a nonlinear system that has been linearized around the current estimates.Second, the problem is complicated by the fact that many $\boldsymbol{\pi}$ produce the same data. Indeed, all $\boldsymbol{\pi}$ resulting in the same data lie in an affine subspace of $\mathbb{R}^c$ denoted

$$\mathcal{D} = \{\boldsymbol{\pi}_d \in \mathbb{R}^c \mid \underline{\boldsymbol{\pi}} = \overline{\mathbf{K}}\boldsymbol{\pi}_d\},$$

which is also the subspace of parameters that leads to the true base parameters, see [48]. Thus, we shall only analyze the convergence of base parameters of the estimated parameters. Note that this problem is similar to the one tackled in [38, Chapter 8].

Let us focus now on relation (37). First, we note that with noisy data (37) gives

$$\hat{\boldsymbol{N}}_{\boldsymbol{\pi}}\Delta\hat{\boldsymbol{\pi}}^{(k)} = -\left(\boldsymbol{r}_c + \boldsymbol{r}_{feas} + \boldsymbol{r}_{cent}\right). \tag{38}$$

Then, with $\boldsymbol{c} = \frac{1}{r}\mathbf{X}^T\mathbf{y} = \frac{1}{r}\mathbf{X}^T\mathbf{X}\boldsymbol{\pi} + \frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon} = \boldsymbol{Q}_{XX}\boldsymbol{\pi} + \frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}$, one has

$$\boldsymbol{r}_c = \boldsymbol{Q}_{XX}\boldsymbol{e}_{\boldsymbol{\pi}}^{(k)} + \boldsymbol{A}^T\boldsymbol{\mu}^{(k)} - \frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}\,, \tag{39}$$

By introducing $\boldsymbol{r}_{c0} = \boldsymbol{Q}_{XX}\boldsymbol{e}_{\boldsymbol{\pi}}^{(k)} + \boldsymbol{A}^T\boldsymbol{\mu}^{(k)}$, by assuming that $\boldsymbol{r}_{c0}$, $\boldsymbol{r}_{feas}$ and $\boldsymbol{r}_{cent}$ are noise-independent, we obtain a relation we are familiar with

$$\mathbf{y}_{nls} = \hat{\boldsymbol{N}}_{\boldsymbol{\pi}}\Delta\hat{\boldsymbol{\pi}}^{(k)} - \frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}\,, \tag{40}$$

with $\mathbf{y}_{nls} = -\boldsymbol{r}_{c0} - \boldsymbol{r}_{feas} - \boldsymbol{r}_{cent}$. Finally, considering (37) under noise-free data, and inserting (40) from noisy data, it follows that

$$\Delta\hat{\boldsymbol{\pi}}^{(k)} = \Delta\boldsymbol{\pi}^{(k)} - \hat{\boldsymbol{N}}_{\boldsymbol{\pi}}^{-1}\left(\frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}\right). \tag{41}$$

By application of the Slutsky's theorem, [42], we can write

$$\operatorname*{plim}_{r\to\infty}\Delta\hat{\boldsymbol{\pi}}^{(k)} = \Delta\boldsymbol{\pi}^{(k)} - \boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\operatorname*{plim}_{r\to\infty}\left(\frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}\right), \tag{42}$$

and by multiplying (42) by $\overline{\mathbf{K}}$ with $\hat{\underline{\boldsymbol{e}}}_{\boldsymbol{\pi}}^{(k)} = \hat{\underline{\boldsymbol{\pi}}}^{(k)} - \underline{\boldsymbol{\pi}}$, one obtains

$$\operatorname*{plim}_{r\to\infty}\left(\hat{\underline{\boldsymbol{e}}}_{\boldsymbol{\pi}}^{(k+1)}\right) = \underline{\boldsymbol{e}}_{\boldsymbol{\pi}}^{(k+1)} - \overline{\mathbf{K}}\,\boldsymbol{N}_{\boldsymbol{\pi}}^{-1}\operatorname*{plim}_{r\to\infty}\left(\frac{1}{r}\mathbf{X}^T\boldsymbol{\varepsilon}\right), \quad (43)$$

yielding that if $E\left[\mathbf{X}^T\boldsymbol{\varepsilon}\right] = \mathbf{0}$ then $\hat{\underline{\boldsymbol{e}}}_{\boldsymbol{\pi}}^{(k+1)}$ converges in probability to the constant $\underline{\boldsymbol{e}}_{\boldsymbol{\pi}}^{(k+1)}$ as the sample size $r \to \infty$, see, e.g., [42, chapter 4, pp. 103]. In this case, the step $\Delta\hat{\boldsymbol{\pi}}^{(k)}$ is consistent with the noise-free step $\Delta\boldsymbol{\pi}^{(k)}$. Further, since $\boldsymbol{r}_c$ is the only location where noise enters (32), it follows that $\Delta\hat{\boldsymbol{\mu}}^{(k)}$ and $\Delta\hat{\boldsymbol{s}}^{(k)}$ result in steps that are consistent with the noise free ones in this case as well. Proceeding inductively, consistency from iteration-to-iteration and the convergence of the noise-free iterations to the correct base parameters together implies the following main proposition

Proposition 2 (Statistical and Physical Consistency of PC-IDIM-LS)**.** *The PC-IDIM-LS algorithm is a statistically consistent base parameter estimator for $\underline{\boldsymbol{\pi}}$ if*

$$E[\mathbf{X}^T\boldsymbol{\varepsilon}] = \mathbf{0}. \quad (44)$$

The algorithm produces a physically consistent result even when this condition does not hold.

This result is important because constrained LS estimation is not able to ensure consistent estimates when the system is identified under closed-loop control since (44) does not hold. One notable exception occurs when proper tailor-made data filtering is applied, as we shall see later. Indeed, as mentioned in Section II, relation (44) is violated because of the feedback which induces a correlation between $\mathbf{X}$ and $\boldsymbol{\varepsilon}$. Filtering data is a way to *break* this correlation, but the results strongly depend on the quality of data and expertise of the user, and this is the reason why it is preferable to choose an identification method robust against a violation of (44). One interesting approach is the DIDIM method previously presented. The following section shows how the constraints (20) can be used to enforce the physical consistency of DIDIM estimates.

F. Conclusion: the key results

In this section, a thorough analysis of the PC-IDIM-LS method has been presented. The resolution of the constrained LS problem, as well as the statistical properties, have been tackled. The key results that will be utilized and/or emphasized in the following sections are the results of Proposition 1 (i.e., (20)) and Proposition 2 (i.e., (44)), the former establishing the physical consistency of the full parameters, the later considering statistical consistency of the base parameters.

IV. Sequential Semidefinite Optimization for Physical and Statistical Consistency

A. Introduction

As it has been theoretically shown in the previous sections, while the work in [6] guaranteed physical consistency of the IDIM-LS estimates via LMI constraints (22), the method naturally suffers from the same challenges as standard IDIM-LS in terms of bias from noise. By comparison, the DIDIM method is a statistically consistent estimator when the model form is valid but does not guarantee the physical consistency of the output. The following subsections consider a unified framework to leverage the relative advantages of each of these previous methods. The new physically-consistent DIDIM (PC-DIDIM) method leverages sequential semidefinite optimization toward guaranteeing the physical consistency of the output and maintaining the statistical consistency of the framework.

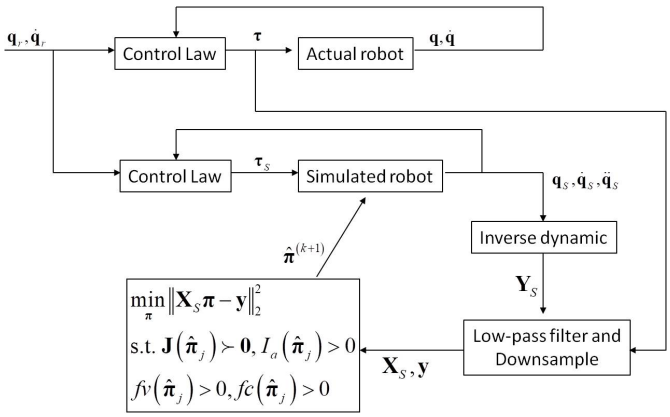

Fig. 2. Diagram for one iteration of the PC-DIDIM algorithm

B. The PC-DIDIM method

The standard DIDIM method from Section II-D is improved by enforcing physical consistency constraints on the inertial parameters at each iterate. Unlike the original DIDIM method, physical consistency requires consideration of the full parameter set. This feature is desirable and lowers the barrier to adoption of the new method. At each iteration k, the PC-DIDIM estimates $\hat{\boldsymbol{\pi}}^{(k)}$ are used to simulate the DDM to obtain the simulated joint positions, velocities and accelerations for the construction of $\mathbf{X}_s^{(k)}$, i.e.,

$$\ddot{\mathbf{q}}_s^{(k)} = \mathbf{H}\left(\mathbf{q}_s^{(k)}, \hat{\boldsymbol{\pi}}^{(k)}\right)^{-1}\left[\boldsymbol{\tau}_s^{(k)} - \mathbf{n}(\mathbf{q}_s^{(k)}, \dot{\mathbf{q}}_s^{(k)}, \hat{\boldsymbol{\pi}}^{(k)})\right]. \quad (45)$$

Estimates for the next iteration are then calculated by solving an SDP that enforces physical consistency of the full parameters:

$$\hat{\boldsymbol{\pi}}^{(k+1)} = \operatorname*{argmin}_{\hat{\boldsymbol{\pi}}} \|\mathbf{X}_s^{(k)}\,\hat{\boldsymbol{\pi}} - \mathbf{y}\|_2^2 \quad (46)$$

$$\text{s.t. } \boldsymbol{J}(\hat{\boldsymbol{\pi}}_j) \succ 0,\ I_a(\hat{\boldsymbol{\pi}}_j) > 0\,, \quad (47)$$

$$f_v(\hat{\boldsymbol{\pi}}_j) > 0,\ f_c(\hat{\boldsymbol{\pi}}_j) > 0,\ \forall j \quad (48)$$

This process is iterated, with a single iteration diagrammed in Fig. 2. This sequential semidefinite optimization procedure is run until convergence. Convergence is considered with a relative change criterion:

$$\max \frac{|\hat{\boldsymbol{\pi}}^{(k+1)}(i) - \hat{\boldsymbol{\pi}}^{(k)}(i)|}{|\hat{\boldsymbol{\pi}}^{(k)}(i)|} \le tol_{cv}\,, \quad (49)$$

where $\hat{\boldsymbol{\pi}}^{(k)}(i)$ is the i-th component of $\hat{\boldsymbol{\pi}}^{(k)}$; and tol_{cv} is a user-defined threshold.

Algorithm 1: PC-DIDIM Algorithm

Run the controller on the physical robot to obtain samples of $\boldsymbol{\tau}$ and $\mathbf{q}$ (Note: $\mathbf{q}$ is only used in validation with the DWH test)

Form $\mathbf{y}$ as described in Section II, equation (7)

Set $\hat{\boldsymbol{\pi}}^{(0)}$ from CAD

for $k = 0, \ldots,$ `max_iter` **do**

  Simulate the forward-dynamic model with robot parameters $\hat{\boldsymbol{\pi}}^{(k)}$ to obtain $\mathbf{q}_s^{(k)}$, $\dot{\mathbf{q}}_s^{(k)}$, $\ddot{\mathbf{q}}_s^{(k)}$

  Construct $\mathbf{X}_s^{(k)}$ as explained in Section IV-B

  Solve the SDP:

$$\hat{\boldsymbol{\pi}}^{(k+1)} = \underset{\hat{\boldsymbol{\pi}}}{\operatorname{argmin}} \|\mathbf{X}_s^{(k)}\hat{\boldsymbol{\pi}} - \mathbf{y}\|_2^2$$
$$\text{s.t. } \boldsymbol{J}(\hat{\boldsymbol{\pi}}_j) \succ 0, I_a(\hat{\boldsymbol{\pi}}_j) > 0$$
$$f_v(\hat{\boldsymbol{\pi}}_j) > 0, f_c(\hat{\boldsymbol{\pi}}_j) > 0 \ \forall j$$

  if $\max \frac{|\hat{\boldsymbol{\pi}}^{(k+1)}(i) - \hat{\boldsymbol{\pi}}^{(k)}(i)|}{|\hat{\boldsymbol{\pi}}^{(k)}(i)|} < tol_{cv}$ **then**

    Break

  end

  Run the revised DWH-test described in [43] to validate the model form.

end

A summary of the PC-DIDIM method is given in Algorithm 1. Note that measurements of $\mathbf{q}$ and $\dot{\mathbf{q}}$ from the robot are not required for an iteration of the PC-DIDIM algorithm since the simulated $\mathbf{q}_s$ and $\dot{\mathbf{q}}_s$ are used. It should also be stressed that initializing this process is not difficult because Computer-Aided-Design (CAD) values of the parameters can be used to provide a seed $\hat{\boldsymbol{\pi}}^{(0)}$. Since these CAD values are computed with software that makes use of physical laws, these initial parameters are physically consistent.

C. The properties of PC-DIDIM method

Let $\hat{\boldsymbol{\pi}}_{PC-DIDIM}$ be the solution obtained at the last iteration, and $\hat{\boldsymbol{\varepsilon}}_{PC-DIDIM} = \mathbf{y} - \mathbf{X}_s\hat{\boldsymbol{\pi}}_{PC-DIDIM}$ the final residual from the PC-DIDIM method. Errors are assumed serially uncorrelated such that the covariance matrix $\boldsymbol{\Omega}$ can be partitioned as (8), where each σ_j is estimated with $\hat{\sigma}_j^2 \approx \|\hat{\boldsymbol{\varepsilon}}_{PC-DIDIM_j}\|_2^2/n_e$, $\hat{\boldsymbol{\varepsilon}}_{PC-DIDIM_j}$ being the joint j residual of the PC-DIDIM method. Let $\bar{\hat{\boldsymbol{\varepsilon}}}_{PC-DIDIM} = \boldsymbol{\Omega}^{-1}\hat{\boldsymbol{\varepsilon}}_{PC-DIDIM}$ be the normalized PC-DIDIM error.

With the assumptions mentioned above, $\bar{\hat{\boldsymbol{\varepsilon}}}_{PC-DIDIM}$ must be independent and identically distributed (i.i.d.). The whiteness of $\bar{\hat{\boldsymbol{\varepsilon}}}_{PC-DIDIM}$ can be assessed with appropriate statistical tests such as the autocorrelation and partial autocorrelation functions (ACF - PACF) commonly used in practice (see e.g. [41]). It is recalled that autocorrelation (or serial correlation) is the correlation of a signal with a delayed copy of itself as a function of delay while the partial autocorrelation function gives the partial correlation of a time series with its own lagged values, controlling for the values of the time series at all shorter lags. These two functions play an important role to determine the lag p in Autoregressive (AR) models. If the tests reject the hypothesis that $\bar{\hat{\boldsymbol{\varepsilon}}}_{PC-DIDIM}$ is white, then it means that either the statistical assumptions made are violated, or a modeling error remains. In the former case, there are no consequences on the consistency of the PC-DIDIM estimates, whereas there are in the latter case [42]. Rigorous justification for this claim is given in Appendix B . For a practitioner to discriminate between these two possibilities, the original Durbin-Wu-Hausman test (DWH-test) presented in [49] has been extended to robot identification in [43]. This modified version is called the *Revised DWH-test*, and the main idea is briefly summarized here. If the model form of the IDM is perfectly known, then we have the following equality

$$\underline{\mathbf{X}}_s = \underline{\mathbf{X}}_{nf}\,. \tag{50}$$

Relation (50) therefore suggests that $\underline{\mathbf{X}}$ differs from $\underline{\mathbf{X}}_s$ by a $\mathbb{R}^{(r\times b)}$ matrix of error from noise denoted $\underline{\mathbf{V}} \in \mathbb{R}^{(r\times b)}$ i.e.

$$\underline{\mathbf{X}} = \underline{\mathbf{X}}_s + \underline{\mathbf{V}}\,, \tag{51}$$

where $E[\underline{\mathbf{V}}] = \mathbf{0}$. Then, the basic idea of this test is simple: each column of $\underline{\mathbf{X}}_s$ denoted $\mathbf{x}_s(i)$ is projected onto the space spanned by the columns of $\underline{\mathbf{X}}$, and if the hypothesis H_0 : $\underline{\mathbf{x}}(i) = \mathbf{x}_s(i) + \underline{\mathbf{v}}(i)$, where $\underline{\mathbf{x}}(i)$, $\mathbf{x}_s(i)$ and $\underline{\mathbf{v}}(i)$ are the i-th columns of $\underline{\mathbf{X}}$, $\underline{\mathbf{X}}_s$ and $\underline{\mathbf{V}}$, respectively, is not rejected, then one concludes that both relations (50) and (51) hold. This means that there is no significant modeling error. In this case, we have the following proposition

Proposition 3 (Consistency of the PC-DIDIM method)**.** *If there is no modeling error, then one has $\mathbf{X}_s = \mathbf{X}_{nf}$, and the PC-DIDIM estimates of the base parameters are statistically consistent.*

Proof. The argument is very similar to the one in Appendix B combined with previous development. If there is no modeling error, then one has $\mathbf{X}_s = \mathbf{X}_{nf}$. With this equality inserted into relation (42), one obtains

$$\operatorname*{plim}_{r\to\infty} \Delta\hat{\boldsymbol{\pi}}^{(k)} = \Delta\boldsymbol{\pi}^{(k)} - \boldsymbol{N}_{\boldsymbol{\pi}}^{-1} \operatorname*{plim}_{r\to\infty} \left(\frac{1}{r}\mathbf{X}_s^T\boldsymbol{\varepsilon}\right)$$

Yet, since $\mathbf{X}_s = \mathbf{X}_{nf}$ is deterministic, and $E[\boldsymbol{\varepsilon}] = 0$ with entries having finite variance, it follows that

$$\operatorname*{plim}_{r\to\infty} \Delta\hat{\boldsymbol{\pi}}^{(k)} = \Delta\boldsymbol{\pi}^{(k)} \tag{52}$$

As a result, in the limit as the number of samples goes to infinity, the PC-DIDIM algorithm takes the same steps under noisy torque data as in the case of having noise-free torque data. A case with noise-free torque data provides

$$\lim_{k\to\infty} \overline{\mathbf{K}}\boldsymbol{\pi}^{(k)} = \underline{\boldsymbol{\pi}}.$$

Thus, combing this behavior with (52), it follows that PC-DIDIM provides a statically consistent base parameter estimate. □

D. Covariance matrix of PC-DIDIM estimates

Until now, the calculation of the covariance matrix of PC-DIDIM estimates was not addressed. Note further that this problem was not tackled in [29], [31] and [6]. From the point

of view of system identification, it is incomplete to identify parameters without dealing with their deviations since they are used to assess the quality of estimation, see [39], [38] and [41] among others. If we can always compute the covariance matrix of the DIDIM estimates given by (18), the calculation of the covariance matrix of $\hat{\boldsymbol{\pi}}_{PC-DIDIM}$ cannot be treated by simply using the standard approaches presented in the previous references. Indeed, $\mathbf{X}_s$ is rank deficient while $\boldsymbol{\pi}$ is estimated by means of a nonlinear method. This combination of these two difficulties may explain why [6], [29], [31], [50], [51] did not address this critical point. In this subsection, we present a methodology to estimate the covariance matrix of $\hat{\boldsymbol{\pi}}_{PC-DIDIM}$.

The topic of rank-deficient observation and covariance matrices has been investigated, see [52] and [39, chapter 4, pages 88-90]. If it was a classical unconstrained LS problem, i.e., $\mathbf{y} = \mathbf{X}_s\boldsymbol{\pi} + \boldsymbol{\varepsilon}$, an elegant way to circumvent the issue of rank-deficient observation and covariance matrices is to use the following relation

$$\boldsymbol{\Sigma}_{PC-DIDIM_2} = \mathbf{X}_{\boldsymbol{\Omega}}\boldsymbol{\Omega}\mathbf{X}_{\boldsymbol{\Omega}}^T\,, \tag{53}$$

which is the covariance matrix of the LS solution $\hat{\boldsymbol{\pi}}_{pinv} = \mathbf{X}_s^{\dagger}\mathbf{y}$, where $\mathbf{X}_{\boldsymbol{\Omega}} = (\mathbf{X}_s^T\tilde{\boldsymbol{\Omega}}^{\dagger}\mathbf{X}_s)^{\dagger}\mathbf{X}_s^T\tilde{\boldsymbol{\Omega}}^{\dagger} \in \mathbb{R}^{(r\times r)}$; $\tilde{\boldsymbol{\Omega}} = \mathbf{X}_s\mathbf{X}_s^T + \boldsymbol{\Omega}$; and the upper-script $\dagger$ denotes the Moore-Penrose inverse. The properties of the pseudo-inverse matrix are given in [53]. However, relation (53) does not account for the constraints that affect the calculation of the covariance matrix (see an example in [39], chapter 4, p.85).

In our case, by considering (40) with $\mathbf{X}$ replaced by $\mathbf{X}_s$, i.e.,

$$\mathbf{y}_{nls} = \hat{\boldsymbol{N}}_{\boldsymbol{\pi}}\Delta\boldsymbol{\pi}^{(k)} - \frac{1}{r}\mathbf{X}_s^T\boldsymbol{\varepsilon}\,, \tag{54}$$

by considering the probability limit, see e.g. [42] chapter 5, and if we consider that $\boldsymbol{N}_{\boldsymbol{\pi}}$ is full-rank, whereas $\mathbf{X}_s$ is not, the calculations in [39, chapter 4], recalled in Appendix C, show that the optimal covariance matrix of the PC-DIDIM estimates is given by

$$\boldsymbol{\Sigma}_{PC-DIDIM_1} = \left(\boldsymbol{N}_{\boldsymbol{\pi}}^T\tilde{\boldsymbol{\Omega}}_{\varepsilon}^{-1}\boldsymbol{N}_{\boldsymbol{\pi}}\right)^{-1} - \mathbf{I}_c\,, \tag{55}$$

with $\tilde{\boldsymbol{\Omega}}_{\varepsilon} = \boldsymbol{N}_{\boldsymbol{\pi}}\boldsymbol{N}_{\boldsymbol{\pi}}^T + \mathbf{X}_s\boldsymbol{\Omega}\mathbf{X}_s^T$. Interestingly, we note that the covariance matrix given by (55) is also the covariance matrix of the increment (37). To be complete, it should be noticed that these explanations are consistent with the calculations presented in [39, pages 212-213]. The main lines of the proof are briefly recalled in Appendix D.

Let $\boldsymbol{\Sigma}_{PC-DIDIM_1}$ be partitioned such that $\boldsymbol{\Sigma}_{PC-DIDIM_1} = \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22}\end{bmatrix}$ where $\boldsymbol{\Sigma}_{11} \in \mathbb{R}^{(b\times b)}$ (resp. $\boldsymbol{\Sigma}_{22} \in \mathbb{R}^{(c-b)\times(c-b)}$) is the covariance matrix of $\hat{\boldsymbol{\pi}}_1$ (resp. $\hat{\boldsymbol{\pi}}_2$); and $\boldsymbol{\Sigma}_{12} \in \mathbb{R}^{b\times(c-b)}$ is the correlation matrix between $\hat{\boldsymbol{\pi}}_1$ and $\hat{\boldsymbol{\pi}}_2$. According to the relation linking $\underline{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}$, one obtains $\boldsymbol{\Sigma}_{red}$, the covariance matrix of the base parameters from the one of standard parameters (55)

$$\boldsymbol{\Sigma}_{red} = \boldsymbol{\Sigma}_{11} + \mathbf{K}\boldsymbol{\Sigma}_{22}\mathbf{K}^T + \mathbf{K}\boldsymbol{\Sigma}_{12}^T + \boldsymbol{\Sigma}_{12}\mathbf{K}^T\,, \tag{56}$$

and since $\operatorname{plim}_{r\to\infty}\hat{\boldsymbol{\pi}} \in \mathcal{D}$, we obtain $\boldsymbol{\Sigma}_{red} \to \boldsymbol{\Sigma}_{DIDIM}$ as $r \to \infty$. Note that if $\mathbf{X}_s$ is replaced by $\underline{\mathbf{X}}_s$ in (53), we retrieve (18), see the calculations given in [39, chapter 4].

We conclude now this section with further comments that are of interest.

E. Further comments on the equality constraints

In [29], the authors suggest that some additional constraints can be added to get more accurate results. If this idea sounds good and is well founded, the user must be wary of equality constraints. For instance, we might add the following equality constraints $\sum_{j=1}^{n} m_j = m_{tot}$, where m_{tot} is the total mass of the robot since it is sufficient to weight it to get m_{tot}. Note that this constraint can be written as a vectorial form with $\boldsymbol{a}_m^T\boldsymbol{\pi} = m_{tot}$, where $\boldsymbol{a}_m \in \mathbb{R}^c$. However, m_{tot} being not perfectly known, an error will be introduced in constraints that will bias the final results. The proof is straightforward if we consider the term $\boldsymbol{r}_b$ in (40) with the error becoming $\boldsymbol{\varepsilon}_{rc} = \mathbf{X}_s^T\boldsymbol{\varepsilon} + \boldsymbol{a}_m^T\delta m_{tot}$, where δm_{tot} is the error on m_{tot}. It is clear that $E[\boldsymbol{\varepsilon}_{rc}] = \mathbf{0}$ if and only $E[\boldsymbol{a}_m^T\delta m_{tot}] = 0$ i.e., $\delta m_{tot} = 0$. Although this error will remain very small in most of cases, it is preferable to use inequality constraints in order to take possible small errors into account. For instance, we can use $l_m < m_{tot} < u_m$, where l_m and u_m are the lower and upper bounds of the total mass, respectively.

It could also be tempting to use the CAD values of non-identifiable parameters (i.e., $\boldsymbol{\pi}_2$) as equality constraints with the aim of obtaining more accurate result. Again, this idea seems well-founded since CAD values are physically consistent by definition and are not far from the real values in most cases. But in doing so, we introduce some errors because the CAD values are not perfect, and $\hat{\boldsymbol{\pi}}_1$ will differ from what we expect. One remaining question is: what is the impact of such errors on the estimates of base parameters? Let us assume that the errors introduced by the use of CAD values of $\boldsymbol{\pi}_2$ do not make the problem infeasible, i.e., these errors remain reasonable and $E[\mathbf{X}_s^T\boldsymbol{\varepsilon}] = \mathbf{0}$ holds. $E[\hat{\underline{\boldsymbol{\pi}}}] = \underline{\boldsymbol{\pi}}$ by definition, and let $\boldsymbol{\pi}_{10}$ and $\boldsymbol{\pi}_{20}$ be the true values of $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, respectively. We can write $\underline{\boldsymbol{\pi}} = \boldsymbol{\pi}_{10} + \mathbf{K}\boldsymbol{\pi}_{20}$ and $\hat{\underline{\boldsymbol{\pi}}} = \boldsymbol{\pi}_1 + \mathbf{K}\boldsymbol{\pi}_2$. Let $\delta\boldsymbol{\pi}_1$ and $\delta\boldsymbol{\pi}_2$ the bias and the errors caused by the use of CAD values of $\boldsymbol{\pi}_2$. One has $\hat{\underline{\boldsymbol{\pi}}} = \boldsymbol{\pi}_{10} + \mathbf{K}\boldsymbol{\pi}_{20} + \delta\boldsymbol{\pi}_1 + \mathbf{K}\delta\boldsymbol{\pi}_2$, i.e., $\hat{\underline{\boldsymbol{\pi}}} = \underline{\boldsymbol{\pi}} + \delta\boldsymbol{\pi}_1 + \mathbf{K}\delta\boldsymbol{\pi}_2$. Since $E[\hat{\underline{\boldsymbol{\pi}}}] = \underline{\boldsymbol{\pi}}$, it follows that $E[\hat{\underline{\boldsymbol{\pi}}}] = \underline{\boldsymbol{\pi}} + E[\delta\boldsymbol{\pi}_1] + E[\mathbf{K}\delta\boldsymbol{\pi}_2] = \underline{\boldsymbol{\pi}} + \delta\boldsymbol{\pi}_1 + \mathbf{K}\delta\boldsymbol{\pi}_2 \implies \delta\boldsymbol{\pi}_1 + \mathbf{K}\delta\boldsymbol{\pi}_2 = \mathbf{0} \implies \delta\boldsymbol{\pi}_1 = -\mathbf{K}\delta\boldsymbol{\pi}_2$ because $\delta\boldsymbol{\pi}_1$ and $\delta\boldsymbol{\pi}_2$ are deterministic by definition. Loosely speaking, the errors introduced by the imperfect CAD values of $\boldsymbol{\pi}_2$ will be compensated by the regroupings provided that problem remains feasible and the observation matrix is not correlated with the vector of errors. This result will be validated through Monte Carlo simulations.

To conclude this section, it is of critical importance to assume that the trajectories applied to the robot are exciting enough, i.e., one has $\operatorname{rank}(\mathbf{X}) = \operatorname{rank}(\underline{\mathbf{X}}) = b$. If this assumption is violated, then all the theoretical analysis made in this section no longer holds because $\operatorname{rank}(\boldsymbol{Q}_{XX}) = \operatorname{rank}(\mathbf{X}^T\mathbf{X}) = \operatorname{rank}(\mathbf{X}) < b$, which implies that all the base parameters cannot be identified from the data. In such cases, experimental protocol must be re-planed or some regularization methods such as [33] must be considered.

F. Conclusion: the main results

In this section, a thorough study of the PC-DIDIM approach has been presented. The algorithm, as well as its statistical properties, have been introduced. The main results that will be emphasized in the following sections are Proposition 3 and the calculation of the covariance matrix of the PC-DIDIM estimates via (55).

V. Monte Carlo simulations results

A. Introduction

This section presents all the results obtained by running Monte Carlo Simulations (MCS). After a short description of the TX-40 robot, the convergence and robustness of PC-DIDIM and PC-IDIM-LS are investigated. Then, the influence of LMI constraints on the estimates is studied, followed by the validation of the computation of the covariance matrix (55) and an analysis of the impact of equality constraints.

All the simulations are executed on a laptop equipped with an Intel Core i5 processor, 8 GB of RAM (DDR4 SDRAM technology), and a capacity of 2 TB [an ACER aspire 3]. MATLAB version 2020-A was used.

B. Presentation of the TX40 robot

The TX40 robot has a serial structure with six rotational joints and is characterized by a coupling between the joints 5 and 6, see [5]. This coupling adds two additional parameters: the viscous coupling friction coefficient of motor 6; and the dry coupling friction coefficient of motor 6. The TX40 robot has 60 base dynamic parameters, with complete modeling as given in [25]. The robot is controlled by a cascade controller, which consists of a P control of the inner velocity loop, and a P control of the outer position loop. τ_j is given by $\tau_j = k_{p_j} k_{v_j}(q_{r_j} - q_{m_j}) - k_{v_j}\dot{q}_{m_j}$, where k_{p_j} is the proportional gain of the outer position loop in Nm/rad, k_{v_j} is the proportional gain of the inner velocity loop in Nm/(rad/s), $\dot{q}_{m_j}$ is the velocity calculated from the differentiation of q_{m_j}. The bandwidth of the first (resp. last) three position closed-loops is 10Hz (resp. 20 Hz).

To fully identify the dynamic parameters, exciting reference trajectories $\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r$ were applied to the TX40 robot. The trajectories used provided a conditioning number of 200 for $\mathbf{X}$ to avoid numerical issues [54]. To evaluate the identification methods involved in this study, all the data are stored with a sampling rate $f_m = 5kHz$. In order to validate the estimates, cross-validations were carried out with three fifth-order polynomials passing through points that are different from those of trajectories involved in the identification methods. All cross-validation data are stored with a different sampling rate given by $f_m^{cv} = 1kHz$, and the relative errors are calculated with the estimates and with these trajectories (see [5] for the technical details).

C. Convergence and robustness of the PC-DIDIM method

The convergence and the robustness against an inappropriate data filtering of PC-DIDIM and PC-IDIM-LS are investigated by running Monte Carlo simulations. Three scenarios are studied: appropriate data filtering and accurate data, inappropriate data filtering and accurate data, and appropriate data filtering and poor data. The case of inappropriate data filtering and poor data is not investigated since it is known that LS estimation fails to provide consistent results in this case, see, e.g., [40]. The assumed ground truth values of the base parameters are their CAD values given in [48] with $\underline{\boldsymbol{\pi}}_{CAD} = \overline{\mathbf{K}}\boldsymbol{\pi}_{CAD}$.

The joint positions are corrupted by zero-mean Gaussian noise with a resolution of 32000 (resp. 1000) counts per revolution for accurate (resp. poor) data. The PC-IDIM-LS method is first carried out with 1) a fourth-order Butterworth filter for position $\hat{\mathbf{q}}$, 2) central differences of this filtered $\hat{\mathbf{q}}$ to generate estimates $\hat{\dot{\mathbf{q}}}$ and $\hat{\ddot{\mathbf{q}}}$, and 3) parallel decimation with a lowpass Tchebyshef filter. The cut-off frequency of the Butterworth (resp. Tchebyshef) filter is 40 Hz (resp. 10 Hz) when the data filtering is appropriate (see the rules given in [21]), whereas the cut-off frequency of the Butterworth (resp. Tchebyshef) filter is 200 Hz (resp. 100 Hz) when the data filtering is inappropriate. For the PC-DIDIM method, only the parallel decimation is carried out, and CAD values of the standard parameters are used for its initialization.

To have an interpretation of results provided by Monte Carlo simulations, 300 simulations are executed, the mean value, as well as the deviation of each parameter, is calculated over the 300 estimations obtained. It is recalled that the mean value is computed with

$$\overline{\underline{\boldsymbol{\pi}}}(i) = \frac{1}{N_{\text{mcs}}} \sum_{l=1}^{N_{\text{mcs}}} \hat{\underline{\boldsymbol{\pi}}}^{[l]}(i)\,, \tag{57}$$

where N_{mcs} is the number of Monte Carlo simulations, while the deviation is evaluated as

$$\overline{\sigma}_{\hat{\underline{\boldsymbol{\pi}}}(i)} = \frac{1}{N_{\text{mcs}} - 1} \sum_{l=1}^{N_{\text{mcs}}} (\hat{\underline{\boldsymbol{\pi}}}^{[l]}(i) - \overline{\underline{\boldsymbol{\pi}}}(i))^2\,. \tag{58}$$

If the mean value is close to the true value with a small deviation, then the parameter is considered as consistently estimated, otherwise it is considered as biased. Note that the mean value is compared with $\underline{\boldsymbol{\pi}}_{CAD}$.

For the first scenario, PC-DIDIM converges in 3 or 4 iterations only; the average number of iterations for convergence over the 300 runs being 3.4. Regarding the computation time, for each iteration of the PC-DIDIM method, the SDP programming needs 18 iterations in 1 second to converge while the simulation of the DDM takes only 5 seconds for an 8-second trajectory. This implies that PC-DIDIM takes 6 seconds only for one iteration and so an average time of 20.4 seconds to converge. For the last two scenarios, the average number of iterations for convergence over the 300 runs being 5.3 while for each iteration of PC-DIDIM, the SDP programming needs 23 iterations in 10 seconds to converge. PC-DIDIM takes an average time of 53 seconds to converge. This increase is simply caused by the fact that the cut-off frequency of the decimate filter is higher which implies that we have more samples to deal with. These results show that the simulation of the DDM and resolution of (22) is not a

TABLE I
MCS RESULTS: PC-DIDIM AND PC-IDIM-LS ESTIMATES, PROPER DATA FILTERING

params	units	PC-DIDIM	PC-IDIM-LS	*true* values
ZZ_1	kg m^2	1.2496 (0.0051)	1.2494 (0.0052)	1.2501
MX_2	kg m	2.2108 (0.0183)	2.2106 (0.0187)	2.2107
XX_3	kg m^2	0.1186 (0.0021)	0.1185 (0.0022)	0.1206
ZZ_4	kg m^2	0.0071 (0.0012)	0.0070 (0.0012)	0.0063
I_{a_6}	kg m^2	0.0113 (0.0002)	0.0115 (0.0003)	0.0114
f_{v_6}	N m s	0.7042 (0.0040)	0.7045 (0.0039)	0.7000

TABLE II
MCS RESULTS: PC-DIDIM AND PC-IDIM-LS ESTIMATES, IMPROPER DATA FILTERING

params	units	PC-DIDIM	PC-IDIM-LS	*true* values
ZZ_1	kg m^2	1.2485 (0.0053)	0.9530 (0.0003)	1.2501
MX_2	kg m	2.2119 (0.0191)	2.3794 (0.0269)	2.2107
XX_3	kg m^2	0.1177 (0.0029)	0.0860 (0.0000)	0.1206
ZZ_4	kg m^2	0.0068 (0.0007)	0.0149 (0.0000)	0.0063
I_{a_6}	kg m^2	0.0110 (0.0005)	0.0000 (0.0001)	0.0114
f_{v_6}	N m s	0.7049 (0.0045)	0.5170 (0.0031)	0.7000

real burden. The mean values of estimated base parameters being very close to their real ones with small deviations, PC-DIDIM converges asymptotically to an element of possible representations of $\boldsymbol{\pi}$, and it can be thus concluded that the PC-DIDIM estimates are both physically and statistically consistent. In Table I the estimates of some base parameters with their variances in parentheses are provided, the detailed results being given in the extra file. Interestingly, we can notice that the estimates of the standard parameters converge to the CAD values. This result can be explained by the fact that CAD values are one representation of possible solutions in $\mathcal{D}$, and PC-DIDIM was initialized with them. The effect of initialization is further studied in the following section.

In contrast, PC-IDIM-LS only provides consistent results when data are accurate enough and associated with appropriate data filtering (i.e., the first scenario). Indeed, for the last two scenarios, the average values of estimated parameters differ significantly from their true values with small deviations which implies that the bias is persisting. This result is simply explained by the fact that relation (11) is violated because data are not accurate enough and/or the data filtering is no longer sufficient to deal with noises that are significant. In Table II the estimates of some base parameters with their variances in parentheses are provided while the detailed results being given in the extra file.

All these results are in agreement with the theoretical approach developed in Sections III-E and IV-B.

D. Influence of LMI constraints and initial values

Until now, only the LMI constraints given by (20) have been considered while others have been proposed in the literature. In [50], the authors have experimentally evaluated the influence of the LMI constraints employed in [29] and [6], i.e., (20), and they concluded that better results are obtained with (20). However, they did not consider the possible correlation between $\mathbf{X}$ and $\boldsymbol{\varepsilon}$. Note that the constraints used in [51] are the same as ours. We recall that the constraints imposed in [29] are simply the positiveness of the mass and the positive definiteness of the rotational inertia matrix about the CoM; these constraints are termed *physical semi-consistency constraints*. In [6], [31], the authors impose that so-called triangle-inequalities must be satisfied as well; these constraints are termed *full physical consistency* constraints. The impact of the LMI constraints on the PC-IDIM-LS and PC-DIDIM estimates is now investigated by imposing the *physical semi-consistency* and *full physical consistency* constraints, and running 300 MCS accordingly to the conditions given in the previous section.

The obtained results being very similar to those exposed in the previous Section V-C, they are not reported here; all the detailed results are given in the extra file. It is clear that the LMI constraints imposed by the user have little impact on the estimates of the PC-DIDIM method coupled or not with proper data filtering. Indeed, the PC-DIDIM estimates are very close to their true values while the number of iterations needed to converge does not significantly vary. Finally, PC-IDIM-LS provides only consistent estimates if proper data filtering is carried out regardless of the LMI constraints imposed. Again, this result is in agreement with the theoretical study introduced in Sections III-E and IV-B.

Concerning the initialization, we normally use the CAD values since they are usually known. In contrast to black-box identification methods developed by the community of system identification, there is thus little need to *estimate* initial values as it is usually done, see [38], [39], [41] among others. Nevertheless, if the CAD values can be considered as *good enough* in most of cases, the impact of initialization must be investigated because the CAD values may be not accurate enough for various reasons. To evaluate the effect of initialization on the DIDIM and PC-DIDIM estimates, the two methods are initialized with the CAD values perturbed by zero-mean Gaussian noise with a relative deviation varying between 10% to 50%, that is to say $\boldsymbol{\pi}^{(0)}(i) = \mathcal{N}\left(\boldsymbol{\pi}_{CAD}(i),\ (0.1\ |\boldsymbol{\pi}_{CAD}(i)|)^2\right)$ for a relative error of 10%. It has been checked that these *disturbed* CAD values are still physically consistent, i.e., they fulfill eq. (20). The obtained

results show that PC-DIDIM succeeds to converge to the true values regardless the initialization whereas DIDIM fails to converge when the relative error exceeds 30%. In fact, if DIDIM is improperly initialized, the estimates may be no longer physically consistent providing a non-positive definite inertia matrix. In such case, the simulation is aborted. In contrast to DIDIM, PC-DIDIM always provides estimates that are physically consistent provided that the problem remains feasible. This result is particularly interesting because it shows that PC-DIDIM is more robust against an improper initialization than the standard DIDIM method, and this proves to be a real advantage when the CAD values are not accurate enough.

E. Calculation of the covariance matrix of PC-DIDIM estimates

The estimation of the covariance matrix of PC-DIDIM estimates is now investigated by executing Monte Carlo simulations and since it has been proved that the PC-DIDIM estimates are consistent, we can deal with the estimation of their covariance matrix. The calculations have been made with accurate data and appropriate data filtering while the matrix $\mathbf{\Omega}$ is estimated according to the procedure described in [21].

It is recalled that the covariance matrix of PC-DIDIM estimated over N_{mcs} simulations is given by (see [39], page 270)

$$\mathbf{\Sigma}_{MCS} = \frac{1}{N_{\text{mcs}}} \sum_{l=1}^{N_{\text{mcs}}} \left(\hat{\boldsymbol{\pi}}^{[l]} - \overline{\hat{\boldsymbol{\pi}}}\right) \left(\hat{\boldsymbol{\pi}}^{[l]} - \overline{\hat{\boldsymbol{\pi}}}\right)^T . \qquad (59)$$

The covariance matrices given by (53) and (55) are compared with $\mathbf{\Sigma}_{MCS}$. For completeness, we compare the covariance matrix given by (18) to the *reduced* covariance matrix $\mathbf{\Sigma}_{red}$ given by (56) by using (55). In Table III, some parameters are gathered while the detailed results are given in the extra file for further reference.

The results tend to show that there is a good agreement between the deviations estimated with MCS and those with (55); only differences that are almost negligible remain. Surprisingly, despite the fact that (53) is a rough approximation because the constraints are not accounted for, the estimated deviations are *acceptable* although some significant differences remain. This tends to show that relation (53) can be used as a rough approximation which will be acceptable for practical purposes if $\boldsymbol{N}_{\boldsymbol{\pi}}$ is not available.

Concerning the standard deviations of base parameters calculated with $\mathbf{\Sigma}_{red}$ based on (55), they match these estimated with (18), see the detailed results given in the extra file. In addition, if (53) is used with $\mathbf{X}_s$ replaced by $\underline{\mathbf{X}}_s$, then we retrieve (18). These Monte Carlo results are in agreement with the theoretical approach developed in Section IV-B and presented in [39, chapter 4, pages 88-90].

F. Effects of errors in the equality constraints

To conclude this part devoted to Monte Carlo simulations, we present now the results obtained when errors are introduced in equality constraints. Interested readers are referred to the detailed results that are given in the supplementary file included with this paper.

TABLE III
MCS RESULTS: EVALUATION OF PC-DIDIM VARIANCES

params	units	deviation with (53)	deviation with (55)	*true* deviation (59)
ZZ_1	kg m^2	0.0067	0.0092	0.0090
I_{a_1}	kg m^2	0.0076	0.0077	0.0080
MX_2	kg m	0.0512	0.0561	0.0565
I_{a_2}	kg m^2	0.0057	0.0049	0.0045
XX_3	kg m^2	0.0063	0.0066	0.0065
YY_3	kg m^2	0.0068	0.0059	0.0061
ZZ_4	kg m^2	0.0006	0.0003	0.0004
M_5	kg	0.0104	0.0097	0.0093
I_{a_6}	kg m^2	0.0012	0.0009	0.0009
f_{v_6}	N m s	0.0108	0.0099	0.0093

To do so, the parameters that are not identifiable, i.e., $\boldsymbol{\pi}_2$ are fixed with their CAD values given in [48] (i.e., $\boldsymbol{\pi}_2 = \boldsymbol{\pi}_{2_{CAD}}$), and every $\boldsymbol{\pi}_2(i)$ is perturbed by zero-mean Gaussian noise with a relative deviation of 5%, that is to say $\boldsymbol{\pi}_2(i) = \mathcal{N}\left(\boldsymbol{\pi}_{2_{CAD}}(i),\ (0.05\ |\boldsymbol{\pi}_{2_{CAD}}(i)|)^2\right)$. Note that $\boldsymbol{\pi}_2 = \boldsymbol{\pi}_{2_{CAD}}$ can be written as $\boldsymbol{\pi}_{2_{CAD}} = \boldsymbol{A}_{ec}\boldsymbol{\pi}$, where $\boldsymbol{A}_{ec} \in \mathbf{R}^{(b-c)\times c}$ is an appropriate constraint matrix. In this case, we simply have $\boldsymbol{A}_{ec} = \begin{bmatrix}\mathbf{0}_{(c-b)\times b} & \mathbf{I}_{(c-b)\times(c-b)}\end{bmatrix}$. Again, the joint positions are corrupted by zero-mean Gaussian noise with a deviation of 32000 counts per revolution, and PC-DIDIM is carried out with the parallel decimation whose the cut-off frequency is 10 Hz. The mean values of PC-DIDIM estimates over the N_{mcs} simulations are calculated, as well their dispersion as explained in the previous sections. The base parameters are estimated with $\hat{\boldsymbol{\pi}}_1$, $\boldsymbol{\pi}_2$ and the regroupings, and they are compared with their true values. Again, interested readers are referred to the detailed results that are given in the supplementary file included with this paper.

Concerning the standard parameters, the results show that $\hat{\boldsymbol{\pi}_1}$ differs from its real value with small deviation. This result, which is consistent with the theoretical analysis provided in Section IV-E, was expected. Indeed, since the equality constraints are corrupted by errors and since they are involved in the estimation of $\boldsymbol{\pi}_1$, it is expected that the estimation of $\boldsymbol{\pi}_1$ is also affected by these errors.

Regarding the base parameters, the results prove that they are not affected by the errors in equality constraints since the estimates stick to their true values with small deviations, i.e., around 1%, see Table IV. This result validates the theoretical analysis made in Section IV-E. It actually indicates that the errors in equality constraints are somehow compensated while estimating $\boldsymbol{\pi}_1$. However, it is critical to understand that the estimates of base parameters must be consistent to get this result because if they are not, then it no longer holds.

G. Conclusion: key facts

This section has presented the results obtained by conducting MCS. They have emphasized that, unlike PC-IDIM-LS, PC-DIDIM is robust against improper data filtering and is more robust against a bad initialization than the former DIDIM method. Also, the number of iterations needed to

TABLE IV
MCS RESULTS: STANDARD DIDIM ESTIMATES AND CONSTRAINED PC-DIDIM

params	units	standard DIDIM	constrained PC-DIDIM	*true* values
ZZ_1	kg m^2	1.2496 (0.0051)	1.2495 (0.0052)	1.2501
MX_2	kg m	2.2108 (0.0183)	2.2109 (0.0184)	2.2107
XX_3	kg m^2	0.1186 (0.0021)	0.1186 (0.0022)	0.1206
ZZ_4	kg m^2	0.0071 (0.0012)	0.0072 (0.0012)	0.0063

converge remains reasonable, i.e., less than 10. It has been shown that the LMI constraints have little impact on the estimates, and this tends to prove that only the correlation between the observation matrix and the vector of errors matters. Finally, imposing equality constraints does not improve the final results.

VI. EXPERIMENTAL RESULTS

A. Introduction

This penultimate section gathers the experimental results obtained with the TX-40 robot. The first subsection presents a comparison between PC-IDIM-LS and PC-DIDIM in the case of proper data-filtering, while the second one deals with such a comparison in the case of improper data filtering. Finally, the last subsection studies the impact of LMI constraints.

B. Comparison of the PC-DIDIM and PC-IDIM-LS methods: Case of Appropriate Data Filtering

The PC-IDIM-LS method is carried out with a filtered position $\hat{\mathbf{q}}$ calculated with a 40 Hz fourth-order Butterworth filter while the parallel decimation is carried out with a 10 Hz Tchebyshef filter. For the PC-DIDIM method, only the parallel decimation is carried out, and CAD values of the standard parameters are used for its initialization.

The PC-DIDIM converges in only 3 iterations, with the SDP programming providing physically consistent estimates. Regarding the computation time, at each iteration of the PC-DIDIM method, the SDP programming needs 18 iterations in 1 second to converge while the simulation of the DDM takes only 5 seconds for an 8-second trajectory. This implies that each iteration of the PC-DIDIM method takes 6 seconds. To assess the statistical consistency of the PC-DIDIM estimates, the construction of $\mathbf{X}_s$ is validated with the revised DWH-test described in [43]. The results obtained proved that (50) holds which means that the PC-DIDIM estimator is statistically consistent. Thus, the estimator provides both a physically and statistically consistent estimation scheme for the TX-40.

With proper data filtering, the PC-IDIM-LS estimates are nearly identical to those provided by the PC-DIDIM method. According to the theory of Hausman [49], the PC-IDIM-LS estimates can be considered as statistically consistent. This result is not surprising, as an appropriate implementation of data filtering should provide $\mathbf{X} \approx \mathbf{X}_s \approx \mathbf{X}_{nf}$, which means that the condition for a consistent LS estimate (11) holds.

TABLE V
RELATIVE ERRORS OBTAINED WITH DIRECT COMPARISONS FOR THE PC-IDIM-LS AND THE PC-DIDIM METHODS - APPROPRIATE DATA FILTERING

Joint j	$e_{PC-IDIM-LS}$	$e_{PC-DIDIM}$
1	5.1%	5.3%
2	4.9%	5.1%
3	5.0%	5.0%
4	5.4%	5.5%
5	7.2%	7.1%
6	7.0%	7.2%

TABLE VI
RELATIVE ERRORS OBTAINED WITH CROSS-VALIDATION, THE PC-IDIM-LS AND THE PC-DIDIM ESTIMATES - APPROPRIATE DATA FILTERING

Joint j	$e_{PC-IDIM-LS}$	$e_{PC-DIDIM}$
1	6.0%	6.1%
2	6.4%	6.6%
3	6.8%	6.9%

Direct comparisons have been performed with the following relative errors:

$$e_{PC-IDIM-LS} = \|\hat{\boldsymbol{\varepsilon}}_{PC-IDIM-LS}\|/\|\mathbf{y}\| \quad \text{and} \tag{60}$$

$$e_{PC-DIDIM} = \|\hat{\boldsymbol{\varepsilon}}_{PC-DIDM}\|/\|\mathbf{y}\| \tag{61}$$

for the PC-IDIM-LS method and PC-DIDIM method, respectively. Relative error values for each joint are given in Table V. These results suggest a satisfactory identification since they are below 10%. Furthermore, in Fig. 3, the portrayal result provided by the ACF function suggests that $\bar{\hat{\boldsymbol{\varepsilon}}}_{PC-DIDIM}$ can be considered white because there are no significant correlations between the samples, all the spikes being confined within the 5% limits plotted in red-dotted lines, see e.g., [41]. In this case, the decimating process succeeded to whiten the final residuals by completely removing the serial correlation. In Fig. 4, the PC-DIDIM estimates are used for a cross-validation, with estimated torques plotted in the same figure. The torque reconstruction matches the measured one. The relative errors on this validation dataset are given in Table VI and demonstrate comparable performance to the training dataset.

In comparison to the original DIDIM algorithm [21], the PC-DIDIM algorithm can work with the full parameter set and does not require computation of the base parameters. Yet, to further verify the PC-DIDIM approach, base parameters values $\underline{\boldsymbol{\pi}}$ were computed from the full parameter output $\boldsymbol{\pi}$ of the algorithm and the QR decomposition of $\mathbf{X}$. The identified base parameter values were found to be in agreement with the estimates of the original DIDIM algorithm. This experimental result is in agreement with the theoretical approach presented in Section IV and those obtained by Monte Carlo simulations. It many appear to the reader that DIDIM and PC-DIDIM

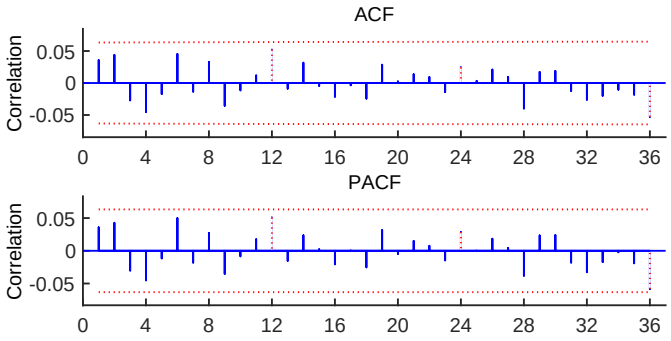

Fig. 3. Portrayal result of the ACF function for the PC-DIDIM method error

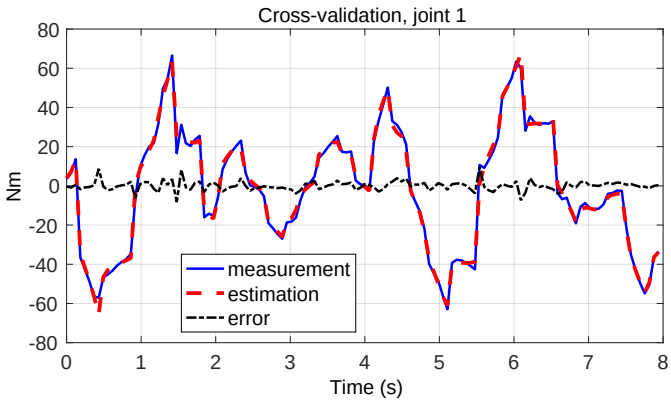

Fig. 4. Cross validation for joint 1 using the PC-DIDIM method.

have similar performance, which would mean that PC-DIDIM would not be very useful for an expert in robot identification. Note, however, that PC-DIDIM has two marked advantages. First, its iterates are guaranteed to be physically consistent such that the forward dynamic model is well defined. Second, its use of the full parameters eliminates the need for a practitioner to compute a base parameter set, thus lowering the barrier to entry for accurate system identification. Finally, as we shall see later, PC-DIDIM has better robustness against an improper initialization than DIDIM.

C. Robustness of the PC-DIDIM method to data filtering

In this section, the robustness of the methods is studied with respect to implementation details of the data filtering. The PC-IDIM-LS method was carried out using 1) a 400 Hz fourth-order Butterworth filter for position $\hat{\mathbf{q}}$ 2) central differences of this filtered $\hat{\mathbf{q}}$ to generate estimates $\dot{\hat{\mathbf{q}}}$ and $\ddot{\hat{\mathbf{q}}}$, and 3) parallel decimation with a 100 Hz cut-off lowpass Tchebyshef filter. For the PC-DIDIM method, only the parallel decimation is carried out, and CAD values of the standard parameters are used for its initialization.

In this case, the high cut-off frequencies lead to noisy regressors and torques that impact the methods. The PC-DIDIM method converges in 5 iterations in this case, guaranteeing physically consistent estimates. Regarding the computation time, at each iteration of the PC-DIDIM method, the SDP programming needs 23 iterations in 11 seconds to converge while the simulation of the DDM still takes only 5 seconds for an 8-second trajectory. This data implies that each iteration of the PC-DIDIM method takes 16 seconds. The increase of the computation time is mainly due to the length of data that has been not sufficiently decimated, as we shall see later. To assess the statistical consistency of the PC-DIDIM estimates, the construction of $\mathbf{X}_s$ is again validated with the revised DWH-test described in [43]. Note that the presence of noise within $\mathbf{y}$ will necessarily increase the relative error, explaining the increase in this case. Overall, the PC-DIDIM estimator still provides a physically and statistically consistent estimation scheme with noisy data.

TABLE VII
RELATIVE ERRORS OBTAINED WITH DIRECT COMPARISONS FOR THE PC-IDIM-LS AND THE PC-DIDIM METHODS - INAPPROPRIATE DATA FILTERING

Joint j	$e_{PC-IDIM-LS}$	$e_{PC-DIDIM}$
1	51.0%	11.0%
2	52.0%	11.0%
3	49.0%	14.0%
4	55.0%	14.0%
5	63.0%	17.0%
6	63.0%	16.0%

However, in Fig. 5, the portrayal result provided by the ACF function suggests that $\bar{\hat{\varepsilon}}_{PC-DIDIM}$ cannot be considered white because there are some remaining significant correlations between the samples: there is significant lag 11 autocorrelation; and significant partial autocorrelation up to lag 8 possibly 9. Indeed, for these lags the spikes are not confined in the 5% limits plotted in red-dotted lines. In other words, the residual is colored. This is mainly explained by the fact that the cut-off frequency of the decimate filter is too large compared to the bandwidth of the position closed-loop. Indeed, samples containing no information (i.e., between 20Hz and 100Hz) have not been removed by the decimation process inducing a more significant serial correlation that can be only coped with by increasing n_d. Finally, it is also conceivable that the heteroscedasticity existing in some columns of $\underline{\mathbf{V}}$ plays a role in the coloration of the residual. This result is consistent with the physical interpretation of the bandpass filtering described in [21] and those presented in [40].

In contrast, the PC-IDIM-LS estimates are strongly biased by the presence of noise in the filtered measurements. Table VII shows relative errors on the order of 50% in this case. Note that for the PC-IDIM-LS scheme an improper filter leads to noise on the decimated regressor $\underline{\mathbf{X}}$ in addition to the measurements $\mathbf{y}$. This causes the estimator to be asymptotically biased, and prevents it from being statistically consistent. This result is explained by the fact that because of inappropriate data filtering, one obtains $\mathbf{X} \neq \mathbf{X}_s \approx \mathbf{X}_{nf}$, which means that relation (11) no longer holds.

Again, cross-validation tests were performed using the output of the two estimators. The relative errors are described in Table VIII, with torque reconstruction illustrated in Fig. 6 for PC-DIDIM. These cross-validation results confirm inaccuracy of the PC-IDIM-LS methods, as its torque reconstruction

TABLE VIII
RELATIVE ERRORS OBTAINED WITH CROSS-VALIDATION, THE PC-IDIMLS AND THE PC-DIDIM ESTIMATES - INAPPROPRIATE DATA FILTERING

Joint j	$e_{PC-IDIM-LS}$	$e_{PC-DIDIM}$
1	73.0%	15.0%
2	71.0%	16.0%
3	73.0%	15.0%

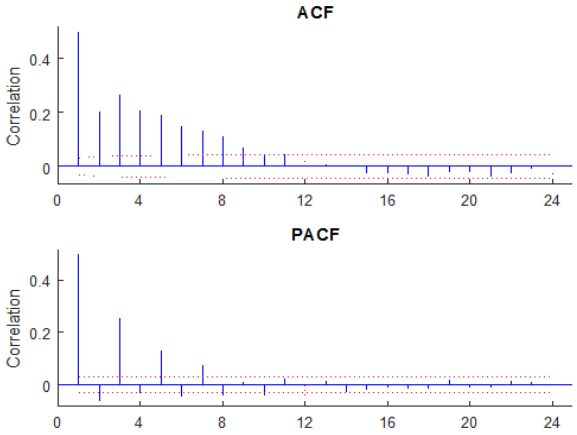

Fig. 5. Portrayal result of the ACF function for PC-DIDIM method error with improper filtering.

clearly shows mismatch with relative errors greater than 50%. This experimental result supports the theoretical study and the results of Monte Carlo simulations.

To definitively validate the PC-DIDIM approach, $\underline{\boldsymbol{\pi}}$ was computed from the full parameter output $\boldsymbol{\pi}$ of the algorithm and the QR decomposition of $\mathbf{X}$. Again, the identified base parameter values were found to be in agreement with the output of the original DIDIM algorithm.

D. Impact of LMI constraints and initialization on the estimates

As a final study, the effect of constraints and the initialization are now investigated, as done in Section V-D.

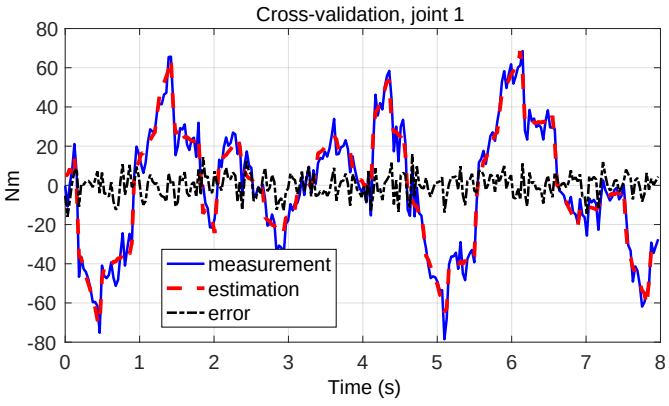

Fig. 6. Cross validation for joint 1 using the PC-DIDIM method with improper filtering.

Regarding the effect of constraints on the PC-IDIM-LS and PC-DIDIM estimates, the *physical semi-consistency* and *full physical consistency* constraints are now imposed. The obtained results being very similar to those exposed in sections VI-B and VI-C, they are not reported here, and all the detailed results are given in the extra file. The LMI constraints imposed by the user have, indeed, little impact on the base-parameter estimates calculated by the PC-DIDIM method coupled or not with a proper data filtering: the construction of $\underline{\mathbf{X}}_s$ is still validated by the revised DWH test described in [43] which implies that the PC-DIDIM estimates can be considered as consistent; the number of iterations needed to converge does not vary significantly; and the relative errors calculated with the cross-validations are very close to those given in Table VI and Table VIII. In contrast, PC-IDIM-LS provides only consistent estimates when coupled with a proper data filtering, regardless of the LMI constraints imposed. This result is consistent with the theoretical study presented in Sections III-E and IV-B.

Regarding the initialization of DIDIM and PC-DIDIM, the two methods are now initialized with the CAD values of the standard parameters perturbed by zero-mean Gaussian noise with a relative deviation varying between 10% to 50%, as done in Section V-D. Again, it has been checked that the initial values fulfill (20). Interestingly, the results obtained and given in the extra file show that PC-DIDIM succeeds to converge, whereas DIDIM fails when the relative error exceeds 20%. This result is consistent with the one presented in Section V-D and enlightens the fact that DIDIM is more sensitive to initialization than PC-DIDIM. This can be due to the fact that PC-DIDIM always provides estimates that are physically consistent provided the problem remains feasible. This robustness against an improper initialization is a clear advantage of PC-DIDIM over DIDIM.

E. Conclusion: key outcomes

The experimental results are in line with those obtained by running MCS. They have confirmed that PC-DIDIM is robust against improper data filtering or noisy data and is more robust against a bad initialization than DIDIM. This last fact is a clear advantage over the former DIDIM method. In addition, the LMI constraints have no impact on the final estimates of the base parameters.

VII. CONCLUSION

This paper has discussed a new framework for the estimation of the inertial parameters for industrial manipulators from noisy data. The proposed Physically Consistent Direct and Inverse Dynamics Identification Method (PC-DIDIM) validates both a simulator of the robot and its inverse dynamics model through iterative semidefinite optimization. By leveraging the noise-free data of a simulation, the approach provides a statistically consistent model estimation scheme while ensuring that its output with any finite dataset provides a physically plausible output.

Extensive results with Monte Carlo simulations and an industrial TX-40 manipulator show that, like the DIDIM

approach, PC-DIDIM is robust against improper data filtering and/or noisy data whereas PC-IDIM-LS must be coupled with a proper data filtering to provide statistically consistent base-parameter estimates, the type of LMI constraints has little impact on the base-parameter estimates, and a way to estimate the covariance matrix of the standard parameters has presented and validated. Finally, PC-DIDIM is found to be more robust against an improper initialization than the former DIDIM method.

Future works concern the use of LMI constraints with an Instrumental Variable approach to conduct a comparison with PC-DIDIM. They will also focus on studying different identification methods of industrial robots to unify them.

APPENDIX

A. Computation of the base parameters

The calculation of the base parameters makes use of the QR decomposition of $\mathbf{X} \in \mathbb{R}^{(r\times c)}$ which is given by

$$\mathbf{Q}^T\mathbf{X} = \begin{bmatrix}\mathbf{R}\\ \mathbf{0}_{(r-c)\times c}\end{bmatrix},$$

where $\mathbf{Q} \in \mathbb{R}^{(r\times r)}$ is an orthogonal matrix, $\mathbf{R} \in \mathbb{R}^{(c\times c)}$ is an upper triangular matrix and $\mathbf{0}_{(r-c)\times c} \in \mathbb{R}^{(r-c)\times c}$ is a matrix of zeros. Noting the b independent columns of $\mathbf{R}$, the same columns of $\mathbf{X}$ are collected in the matrix $\mathbf{X}_1$ and the corresponding parameters be collected in the vector $\boldsymbol{\pi}_1$. The other columns and parameters are collected in $\mathbf{X}_2$ and $\boldsymbol{\pi}_2$ such that

$$\mathbf{X}\,\boldsymbol{\pi} = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\pi}_1\\ \boldsymbol{\pi}_2\end{bmatrix}.$$

Since $\mathbf{X}_1$ is full column rank, the matrix $\mathbf{X}_2$ can be written in terms of $\mathbf{X}_1$ as

$$\mathbf{X}_2 = \mathbf{X}_1\mathbf{K}.$$

Consequently

$$\mathbf{X}\,\boldsymbol{\pi} = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix}\begin{bmatrix}\underline{\boldsymbol{\pi}}\\ \mathbf{0}\end{bmatrix} = \underline{\mathbf{X}}\,\underline{\boldsymbol{\pi}},$$

where $\underline{\mathbf{X}} = \mathbf{X}_1$ and the base parameter vector $\underline{\boldsymbol{\pi}}$ is given by

$$\underline{\boldsymbol{\pi}} = \boldsymbol{\pi}_1 + \mathbf{K}\boldsymbol{\pi}_2.$$

The matrix $\mathbf{K}$ allows us obtaining the grouping equations of the parameters $\boldsymbol{\pi}_2$ with $\boldsymbol{\pi}_1$.

In order to determine $\mathbf{K}$, the QR decomposition of the matrix $\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix}$ is used and one obtains

$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2\\ \mathbf{0}_{(r-b)\times b} & \mathbf{0}_{(r-b)\times(c-b)}\end{bmatrix},$$

where $\mathbf{R}_1 \in \mathbb{R}^{(b\times b)}$ is a full-rank upper triangular matrix, and $\mathbf{R}_2 \in \mathbb{R}^{b\times(c-b)}$. It follows that

$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_1\mathbf{R}_1 & \mathbf{Q}_1\mathbf{R}_2\end{bmatrix},$$
$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1\begin{bmatrix}\mathbf{I}_b & \mathbf{R}_1^{-1}\mathbf{R}_2\end{bmatrix},$$
$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \mathbf{X}_1\begin{bmatrix}\mathbf{I}_b & \mathbf{R}_1^{-1}\mathbf{R}_2\end{bmatrix},$$
$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \mathbf{X}_1\begin{bmatrix}\mathbf{I}_b & \mathbf{K}\end{bmatrix},$$

$$\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2\end{bmatrix} = \mathbf{X}_1\overline{\mathbf{K}},$$

with

$$\overline{\mathbf{K}} = \begin{bmatrix}\mathbf{I}_b & \mathbf{K}\end{bmatrix},$$
$$\mathbf{K} = \mathbf{R}_1^{-1}\mathbf{R}_2,$$

and thus

$$\underline{\boldsymbol{\pi}} = \overline{\mathbf{K}}\boldsymbol{\pi},$$
$$\underline{\boldsymbol{\pi}} = \begin{bmatrix}\mathbf{I}_b & \mathbf{R}_1^{-1}\mathbf{R}_2\end{bmatrix}\boldsymbol{\pi}.$$
$$\underline{\boldsymbol{\pi}} = \boldsymbol{\pi}_1 + \mathbf{R}_1^{-1}\mathbf{R}_2\boldsymbol{\pi}_2.$$

B. The effect of modeling errors on the base regressor $\underline{\mathbf{X}}$

Let us now consider an error in the model. In econometrics, a bias due to a modeling error is often considered as a bias due to omitted parameters [42]. It is therefore convenient to introduce the following term $\underline{\mathbf{X}}_{s_{oc}}\underline{\boldsymbol{\pi}}_{ov}$ where $\underline{\boldsymbol{\pi}}_{ov} \in \mathbb{R}^{oc}$ is the vector of omitted parameters in the model, $\underline{\mathbf{X}}_{s_{oc}} \in \mathbb{R}^{(r\times oc)}$ the omitted columns in $\underline{\mathbf{X}}_s$ associated with $\underline{\boldsymbol{\pi}}_{ov}$, and $\underline{\mathbf{V}} \in \mathbb{R}^{(r\times b)}$ a matrix of error from noise. In this case, one obtains

$$\boldsymbol{\varepsilon}_{DIDIM} = \underline{\mathbf{V}}\,\underline{\boldsymbol{\pi}} + \boldsymbol{\varepsilon} + \underline{\mathbf{X}}_{s_{oc}}\underline{\boldsymbol{\pi}}_{ov},$$

yielding

$$E[\underline{\mathbf{X}}_s^T\boldsymbol{\varepsilon}_{DIDIM}] = \underline{\mathbf{X}}_s^T\underline{\mathbf{X}}_{s_{oc}}\underline{\boldsymbol{\pi}}_{ov},$$

where $\boldsymbol{\varepsilon}_{DIDIM}$ is the DIDIM error. This follows since $\underline{\mathbf{X}}_s$, $\underline{\mathbf{X}}_{s_{oc}}$ and $\underline{\boldsymbol{\pi}}_{ov}$ are deterministic by definition. Finally, since $\underline{\boldsymbol{\pi}}_{ov} \neq \mathbf{0}$, it comes that $E[\underline{\mathbf{X}}_s^T\boldsymbol{\varepsilon}_{DIDIM}] \neq \mathbf{0}$ which means that the DIDIM estimates are biased.

To conclude this subsection, we show that an error within $\mathbf{X}_s$ also corrupts $\underline{\mathbf{X}}_s$. According to the regrouping formula

$$\underline{\boldsymbol{\pi}} = \boldsymbol{\pi}_1 + \mathbf{R}_1^{-1}\mathbf{R}_2\boldsymbol{\pi}_2$$

it is clear that $E(\hat{\underline{\boldsymbol{\pi}}}) \neq \underline{\boldsymbol{\pi}}$ if $\hat{\boldsymbol{\pi}} \notin \mathcal{D}$. Further, because $\hat{\boldsymbol{\pi}}$ is used to simulate the DDM in order to get $\ddot{\mathbf{q}}_s$, the vector of simulated accelerations. It follows that $\ddot{\mathbf{q}}_s \neq \ddot{\mathbf{q}}_{nf}$, the vector of noise-free accelerations. This leads to $\dot{\mathbf{q}}_s \neq \dot{\mathbf{q}}_{nf}$ and $\mathbf{q}_s \neq \mathbf{q}_{nf}$ by integration of the DDM where $\dot{\mathbf{q}}_s$ (resp. $\dot{\mathbf{q}}_{nf}$) is the vector of simulated (resp. noise-free) velocities, and $\mathbf{q}_s$ (resp. $\mathbf{q}_{nf}$) is the vector of simulated (resp. noise-free) joint positions. Finally, we obtain $\underline{\mathbf{X}}_s \neq \underline{\mathbf{X}}_{nf}$ which leads to $\mathbf{X}_s \neq \mathbf{X}_{nf}$ since $\mathbf{X}_s = \begin{bmatrix}\underline{\mathbf{X}}_s & \mathbf{X}_{s_2}\end{bmatrix}$ by construction and with the notations previously used i.e. $\underline{\mathbf{X}}_s = \mathbf{X}_{s_1}$.

C. Calculation of the covariance matrix of $\Delta\hat{\boldsymbol{\pi}}$

Let us condider (40)

$$\mathbf{y}_{nls} = \hat{\boldsymbol{N}}_{\boldsymbol{\pi}}\Delta\boldsymbol{\pi}^{(k)} - \frac{1}{r}\mathbf{X}_s^T\boldsymbol{\varepsilon},$$

We recall that the covariance matrix of $\Delta\hat{\boldsymbol{\pi}}$, denoted $\boldsymbol{\Sigma}_{\boldsymbol{\pi}}$, is given by, see e.g. [42] chapter 5,

$$\boldsymbol{\Sigma}_{\boldsymbol{\pi}} = \lim_{r\to\infty}\left(r\,E\left[\Delta\tilde{\boldsymbol{\pi}}^{(k)}\Delta\tilde{\boldsymbol{\pi}}^{(k)^T}\right]\right),$$

with $\Delta\tilde{\boldsymbol{\pi}}^{(k)} = \Delta\boldsymbol{\pi}^{(k)} - \Delta\hat{\boldsymbol{\pi}}^{(k)}$. By considering the probability limit and the result given in [39], complement 4.3 page 88,

we have

$$\boldsymbol{\Sigma_\pi} = \left(\boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi}\right)^{-1} \boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \cdots \boldsymbol{\Omega}_\varepsilon \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi} \left(\boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi}\right)^{-1},$$

where

$$\tilde{\boldsymbol{\Omega}}_\varepsilon = \boldsymbol{N_\pi} \boldsymbol{N_\pi}^T + \mathbf{X}_s^T \boldsymbol{\Omega} \mathbf{X}_s \,,$$

and

$$\boldsymbol{\Omega}_\varepsilon = \mathbf{X}_s^T \boldsymbol{\Omega} \mathbf{X}_s \,.$$

Now, with

$$\boldsymbol{\Omega}_\varepsilon = \mathbf{X}^T \boldsymbol{\Omega} \mathbf{X} = \tilde{\boldsymbol{\Omega}}_\varepsilon - \boldsymbol{N_\pi} \boldsymbol{N_\pi}^T \,,$$

one obtains

$$\boldsymbol{\Sigma_\pi} = \left(\boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi}\right)^{-1} \boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \cdots \left(\tilde{\boldsymbol{\Omega}}_\varepsilon - \boldsymbol{N_\pi} \boldsymbol{N_\pi}^T\right) \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi} \left(\boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi}\right)^{-1},$$

yielding

$$\boldsymbol{\Sigma_\pi} = \left(\boldsymbol{N_\pi}^T \tilde{\boldsymbol{\Omega}}_\varepsilon^{-1} \boldsymbol{N_\pi}\right)^{-1} - \mathbf{I}_c \,.$$

Interestingly, if $\underline{\mathbf{X}}_s$ and $\boldsymbol{N_\pi}$ are full-rank, then $\boldsymbol{\Sigma_\pi}$ reduces to the well-known formula

$$\boldsymbol{\Sigma_\pi} = \boldsymbol{N_\pi}^{-1} \boldsymbol{\Omega}_\varepsilon \boldsymbol{N_\pi}^{T^{-1}} \,.$$

D. Calculation of the covariance matrix of the increment

In this appendix, we recall the calculation of the covariance matrix of the increment. The complete proof is given in [42], chapter 6. First, let us the following general nonlinear relation

$$\mathbf{y} = f(\boldsymbol{\theta}) + \boldsymbol{\varepsilon} \,,$$

where $\mathbf{y} \in \mathbb{R}^m$ is the vector of measurements; $\boldsymbol{\theta} \in \mathbb{R}^b$ is the vector of parameters to identify; $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ is the vector of errors; and f is a nonlinear function. Let $\hat{\boldsymbol{\theta}}$ be an estimate of $\boldsymbol{\theta}$. A first-order development around $\hat{\boldsymbol{\theta}}$ gives

$$\mathbf{y} = \mathbf{y}(\hat{\boldsymbol{\theta}}) + D_f(\hat{\boldsymbol{\theta}}) \Delta\boldsymbol{\theta} + \boldsymbol{r}_{ho} + \boldsymbol{\varepsilon} \,,$$

where $D_f(\hat{\boldsymbol{\theta}}) \in \mathbb{R}^{(m \times b)}$ is the Jacobian matrix of f with respect to $\boldsymbol{\theta}$ evaluated at $\hat{\boldsymbol{\theta}}$; $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is the increment; and $\boldsymbol{r}_{ho}$ is the errors due to the higher orders. With the approximation $\boldsymbol{r}_{ho} \approx \mathbf{0}$, and $\boldsymbol{r}(\hat{\boldsymbol{\theta}}) = \mathbf{y} - \mathbf{y}(\hat{\boldsymbol{\theta}})$, one obtains the following LS estimates of the increment

$$\Delta\hat{\boldsymbol{\theta}} = \left(D_f(\hat{\boldsymbol{\theta}})^T D_f(\hat{\boldsymbol{\theta}})\right)^{-1} D_f(\hat{\boldsymbol{\theta}})^T \boldsymbol{r}(\hat{\boldsymbol{\theta}}) \,,$$

whose the covariance matrix is

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}} = \left(D_f(\hat{\boldsymbol{\theta}})^T \boldsymbol{\Omega}^{-1} D_f(\hat{\boldsymbol{\theta}})\right)^{-1} ,$$

with $\boldsymbol{\Omega}$ the covariance matrix of $\boldsymbol{\varepsilon}$. Now, if $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$, i.e. what we actually expect, then the nonlinear relation and its linearized have the same residuals since $\boldsymbol{r}(\boldsymbol{\theta}) = \mathbf{y} - \mathbf{y}(\boldsymbol{\theta}) = \mathbf{y} - f(\boldsymbol{\theta}) = \boldsymbol{\varepsilon}$. This explains why the final solution $\hat{\boldsymbol{\theta}}$ and $\hat{\Delta\boldsymbol{\theta}}$ have the same covariance matrix. Note that $\boldsymbol{r}_{ho} \approx \mathbf{0}$ is standard approximation while employing the Gauss-Newton algorithm, and this approximation is often met in practice, see [42, chapter 6].

REFERENCES

[1] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[2] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. V. Brussel, "Optimal robot excitation and identification," *IEEE Trans. on Robotics*, vol. 13, no. 5, pp. 730–740, Oct 1997.

[3] M. M. Olsen, J. Swevers, and W. Verdonck, "Maximum Likelihood Identification of a Dynamic Robot Model: Implementation Issues," *The International Journal of Robotics Research*, vol. 21, no. 2, pp. 89–96, 2002.

[4] N. Ramdani and P. Poignet, "Robust dynamic experimental identification of robots with set membership uncertainty," *IEEE/ASME Trans. on Mechatronics*, vol. 10, no. 2, pp. 253–256, April 2005.

[5] A. Janot, P. O. Vandanjon, and M. Gautier, "A generic instrumental variable approach for industrial robot identification," *IEEE Trans. on Control Systems Technology*, vol. 22, no. 1, pp. 132–145, Jan 2014.

[6] 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 Robotics and Automation Letters*, vol. 3, no. 1, pp. 60–67, Jan 2018.

[7] M. G. John Hollerbach, Wisama Khalil, "Chapter 6: Model identification," in *Springer Handbook of Robotics*, 2nd ed., B. Siciliano and O. Khatib, Eds. New York: Springer, 2016.

[8] K. Ayusawa, G. Venture, and Y. Nakamura, "Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems," *Int. J. of Robotics Research*, vol. 33, no. 3, pp. 446–468, 2014.

[9] J. Jovic, A. Escande, K. Ayusawa, E. Yoshida, A. Kheddar, and G. Venture, "Humanoid and human inertia parameter identification using hierarchical optimization," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 726–735, June 2016.

[10] T. Lee and F. C. Park, "A geometric algorithm for robust multibody inertial parameter identification," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–1, 2018.

[11] J. Jin and N. Gans, "Parameter identification for industrial robots with a fast and robust trajectory design approach," *Robotics and Computer-Integrated Manufacturing*, vol. 31, pp. 21 – 29, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584514000441

[12] K. Dolinský and S. Čelikovský, "Application of the method of maximum likelihood to identification of bipedal walking robots," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–8, 2017.

[13] R. Miranda-Colorado and J. Moreno-Valenzuela, "An efficient on-line parameter identification algorithm for nonlinear servomechanisms with an algebraic technique for state estimation," *Asian Journal of Control*, vol. 19, no. 6, pp. 2127–2142, 2017, asjc.1511. [Online]. Available: http://dx.doi.org/10.1002/asjc.1511

[14] A. Montazeri, C. West, S. D. Monk, and C. J. Taylor, "Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm," *International Journal of Control*, vol. 90, no. 4, pp. 661–683, 2017. [Online]. Available: https://doi.org/10.1080/00207179.2016.1230231

[15] A. Valera, M. Díaz-Rodríguez, M. Valles, E. Oliver, V. Mata, and A. Page, "Controller–observer design and dynamic parameter identification for model-based control of an electromechanical lower-limb rehabilitation system," *International Journal of Control*, vol. 90, no. 4, pp. 702–714, 2017. [Online]. Available: https://doi.org/10.1080/00207179.2016.1215529

[16] L. Simoni, M. Beschi, G. Legnani, and A. Visioli, "Modelling the temperature in joint friction of industrial manipulators," *Robotica*, pp. 1–22, 2017.

[17] R. Miranda-Colorado and J. Moreno-Valenzuela, "Experimental parameter identification of flexible joint robot manipulators," *Robotica*, vol. 36, no. 3, pp. 313–332, 2018.

[18] E. Villagrossi, L. Simoni, M. Beschi, N. Pedrocchi, A. Marini, L. M. Tosatti, and A. Visioli, "A virtual force sensor for interaction tasks with conventional industrial robots," *Mechatronics*, vol. 50, pp. 78 – 86, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415818300163

[19] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, "Towards versatile legged robots through active impedance control," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 2015. [Online]. Available: http://ijr.sagepub.com/content/34/7/1003.abstract

[20] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[21] M. Gautier, A. Janot, and P. O. Vandanjon, "A new closed-loop output error method for parameter identification of robot dynamics," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 428–444, March 2013.

[22] O. Reiersøl, "Confluence analysis by means of lag moments and other methods of confluence analysis," *Econometrica*, vol. 9, no. 1, pp. 1–24, 1941.

[23] S. C. Puthenpura and N. K. Sinha, "Identification of continuous-time systems using instrumental variables with application to an industrial robot," *IEEE Transactions on Industrial Electronics*, vol. IE-33, no. 3, pp. 224–229, Aug 1986.

[24] K. Yoshida, N. Ikeda, and H. Mayeda, "Experimental study of the identification methods for an industrial robot manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Jul 1992, pp. 263–270.

[25] M. Gautier, P. O. Vandanjon, and A. Janot, "Dynamic identification of a 6 dof robot without joint position data," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 234–239.

[26] M. Brunot, A. Janot, F. Carrillo, and H. Garnier, "Comparison between the idim-iv method and the didim method for industrial robots identification," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2017, pp. 571–576.

[27] W. Li and J.-J. E. Slotine, "An indirect adaptive robot controller," *Systems & Control Letters*, vol. 12, no. 3, pp. 259–266, 1989.

[28] K. Yoshida and W. Khalil, "Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters," *Int. J. of Robotics Research*, vol. 19, no. 5, pp. 498–510, 2000.

[29] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *Int. J. of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014.

[30] D. Jung, J. Cheong, D. Park, and C. Park, "Backward sequential approach for dynamic parameter identification of robot manipulators," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, pp. 1–10, 2018.

[31] S. Traversaro, S. Brossette, A. Escande, and F. Nori, "Identification of fully physical consistent inertial parameters using optimization on manifolds," in *IEEE/RSJ Int. Conf. on Intelligent Rob. and Sys.*, Oct 2016, pp. 5446–5451.

[32] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 3, pp. 735–747, 2005. [Online]. Available: https://doi.org/10.1137/S0895479803436937

[33] T. Lee, P. M. Wensing, and F. C. Park, "Geometric robot dynamic identification: A convex programming approach," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 348–365, 2019.

[34] R. Featherstone and D. Orin, "Chapter 2: Dynamics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer, 2008.

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

[36] M. Gautier, "Numerical calculation of the base inertial parameters," *J Robotics Syst.*, vol. 8, pp. 485–506, 1991.

[37] P. M. Wensing, G. Niemeyer, and J.-J. E. Slotine, "Observability in Inertial Parameter Identification," 2017. [Online]. Available: http://arxiv.org/abs/1711.03896

[38] L. Ljung, *System Identification: Theory for the User (2nd Edition)*. Prentice Hall, 1999.

[39] T. Soderstrom and P. Stoica, *System Identification*, ser. Series in Systems and Control Engineering. Prentice Hall, 1989.

[40] M. Brunot, A. Janot, P. Young, and F. Carrillo, "An improved instrumental variable method for industrial robot model identification," *Control Engineering Practice*, vol. 74, pp. 107–117, May 2018.

[41] P. Young, *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer Verlag Berlin, 2011.

[42] R. Davidson and J. G. MacKinnon, *Estimation and Inference in Econometrics*, ser. OUP Catalogue. Oxford University Press, 1993, no. 9780195060119.

[43] A. Janot, P.-O. Vandanjon, and M. Gautier, "A revised durbin-wu-hausman test for industrial robot identification," *Control Engineering Practice*, vol. 48, pp. 52 – 62, 2016.

[44] R. H. Tütüncü, K. Toh, and M. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming*, vol. 95, no. 2, p. 189217, 2003.

[45] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[46] H. Y. Benson and R. J. Vanderbei, "Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming," *Mathematical Programming*, vol. 95, no. 2, pp. 279–302, Feb 2003.

[47] L. Vandenberghe, "The cvxopt linear and quadratic cone program solvers," *Documentation*, March 2010. [Online]. Available: http://cvxopt.org/documentation/coneprog.pdf

[48] M. Gautier, S. Briot, and G. Venture, "Identification of consistent standard dynamic parameters of industrial robots," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2013, pp. 1429–1435.

[49] J. Hausman, "Specification tests in econometrics," *Econometrica*, vol. 46, no. 6, pp. 1251–1271, 1978.

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

[51] 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 Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, Oct 2019.

[52] H. J. Werner, "More on blu estimation in regression models with possibly singular covariances," *Linear Algebra and its Applications*, vol. 67, pp. 207–2, June 1985.

[53] J. Barata and M. Hussein, "The moorepenrose pseudoinverse: A tutorial review of the theory," *Brazilian Journal of Physics*, vol. 42, no. 1-2, pp. 146–165, April 2012.

[54] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," *The International Journal of Robotics Research*, vol. 11, no. 4, pp. 362–375, 1992.