Physical Feasibility of Robot Base Inertial Parameter Identification: a Linear Matrix Inequality Approach

Cristóvão D. Sousa* Rui Cortesão*

January 20, 2014

Postprint of doi:10.1177/0278364913514870

Abstract

Identification of robot dynamics is a key issue to boost the performance of model-based control techniques, having also a key role in realistic simulation. Robot dynamic parameters have physical meaning, hence parameter estimations must correspond to physically feasible values. Since it is only possible to identify linear combinations of parameters — the base parameters — the physical feasibility of such combinations cannot be directly asserted. In this paper we show that feasibility conditions define a convex set and can be written as a linear matrix inequality (LMI), suitable for semidefinite programming (SDP) techniques. We propose three methods based on LMI–SDP to deal with the feasibility of base parameters. The first method checks if a given base parameter estimation has physical meaning. The second method corrects infeasible estimations, finding the closest feasible base parameters. These two methods can be applied to existing regression techniques. The third method performs parameter identification through ordinary least squares regression constrained to the feasible space, guaranteeing the optimal solution. Experiments with a 7-link robot manipulator are provided, and efficiency and scalability are discussed.

*Institute of Systems and Robotics – University of Coimbra, Portugal; crisjss@gmail.com, cortesao@isr.uc.pt

1 Introduction

The knowledge of robot dynamics is essential to boost model-based control, enabling high performance motions and controlled force interactions. Moreover, accurate dynamic models are also important for realistic robot simulation. Robot dynamics depends on both geometry and rigid-body parameters, including link inertias, and friction. The identification of dynamic parameters can be done through CAD software, by individual experimental tests of robot parts, or by dynamic model regression methods. Since the robot dynamic model is linear with respect to inertial parameters, linear regression techniques can be applied, although some parameters can only be estimated in linear combinations, known as base parameters (Gautier and Khalil, 1990). There is a rich literature on parameter identification, focusing on regression uncertainties and excitation trajectories — see (Wu et al., 2010) for an overview. Notwithstanding all the research, the physical feasibility of estimated parameters is often disregarded. Dynamic parameters have physical meaning and are therefore bounded to physical values. Nevertheless, the verification of physical consistency of base parameters is not straightforward. Physically infeasible estimations lead to non-positive inertia matrix at some, or even all, joint positions. Consequently, the use of a dynamic model with non-positive inertia matrix leads to unrealistic simulation and unstable model-based control (Yoshida and Khalil, 2000). It can be argued that adequate experiments can lead to esti-

mations that are accurate enough to be physically feasible. However, no matter how good the parameter excitation and regression techniques are, a mathematical proof that guarantees physical feasibility of robot base inertia parameter estimation is necessary. Moreover, a regression method constrained to physically feasible solutions can be of interest. Yoshida and Khalil (2000) proposed a trial and error recursive method to check if a base parameter estimation is feasible using two-dimensional graphs. It is unclear if this method is suited for automatic implementation or if it is capable of coping with a high number of robot links. Mata et al. (2005) proposed a method for parameter identification taking feasibility into account. The method firstly finds an unconstrained base parameter estimation using classical regression. Then, a quadratic programming optimization constrained to feasible values is done over the total inertial parameters, minimizing both the regression error and the distance to the base parameters previously found. The method presents a variable to be tuned between two optimization objectives, and the convergence to the global solution that minimizes regression error is not discussed. Ting et al. (2011) proposed a nonlinear Bayesian parameter identification method with physical feasibility assessment. Feasibility is guaranteed by searching a solution in a feasible virtual parameter space which has a nonlinear projection onto the inertial parameter space. This solution is the one closest to the unconstrained solution, where the distance is weighted by regression data. Ayusawa and Nakamura (2010) proposed an identification method for high number of link robots addressing physical feasibility. By representing robot links as a finite number of mass points they replace the feasibility conditions by linear inequalities. In this way the optimization is simplified, although the quality of the approximation depends on the number of mass points. Díaz-Rodríguez et al. (2010) proposed an approach where less significant parameters are pruned from the model until the identified ones are feasible. Feasibility is checked by searching the overall discretized solution space. Gautier et al. (2013) proposed a method where the regression estimation is pushed towards an a priori known feasible solution obtained from CAD data. The feasibility is then checked by taking into account a tolerance to accommodate regression data measurement errors.

In our paper, we show that physical feasibility conditions can be written in a semidefinite programming (SDP) perspective, as a linear matrix inequality (LMI). This implies that the set defined by physical feasibility conditions is necessarily convex, leading to global optimums in optimization problems. It follows that base parameters can also be written as LMIs. This allows the development of SDP problems to treat physical feasibility constraints. SDP techniques are known for being effective and fast, allowing problems with a high number of parameters to be efficiently solved. We firstly present a succinct SDP based method which verifies base parameter feasibility in automatic and scalable fashion. Then, we devise a method to correct infeasible parameter solutions to the closest feasible one. Furthermore, we propose a regression method which finds the optimal physically feasible base parameter solution that best fits regression data.

The paper is organized as follows. Section 2 introduces the conditions for physical feasibility of inertial parameters. In Section 3 we reformulate such conditions as an LMI. It is also shown how additional constraints can be included in the LMI formulation. In Section 4 we devise the feasibility test and correction methods for base parameters. Additionally, optimal base parameter feasible estimation through constrained regression is proposed. In Section 5, two experiments are analyzed. The first one addresses the feasibility test in a 3-link robot and the second one assesses the three proposed methods in a 7-link WAM$^{\mathrm{TM}}$ robot. Section 6 concludes the paper.

2 Physical Feasibility of Inertial Parameters

Rigid-body robot manipulator dynamics relates joint positions, velocities and accelerations ($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$), with the joint generalized force/torque $\boldsymbol{\tau}$,

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

where $\boldsymbol{M}(\boldsymbol{q})$ is the mass matrix, which is symmetric and positive definite, $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$ represents Coriolis and

centripetal forces, and $\boldsymbol{g}(\boldsymbol{q})$ is the gravity term. All these terms depend on geometric and dynamic parameters. Each robot link contributes with a set of inertial parameters, composed by mass, center of mass, and inertia tensor. Other dynamic parameters can also be taken into account, such as joint frictions and actuator inertias. Dynamic parameters have physical restrictions which should be addressed in the estimation problem. Link masses are positive and inertia tensors are symmetric and positive definite. The set of conditions for each link k which guarantees physical feasibility is given by[1] (Yoshida and Khalil, 2000)

$$\begin{cases} m_k > 0 \\ \boldsymbol{I}_k \succ 0 \end{cases} , \tag{2}$$

where m_k is the link mass, and $\boldsymbol{I}_k$ is the link inertia tensor about its center of mass,

$$\boldsymbol{I}_k \equiv \begin{bmatrix} I_{k,xx} & I_{k,xy} & I_{k,xz} \\ I_{k,xy} & I_{k,yy} & I_{k,yz} \\ I_{k,xz} & I_{k,yz} & I_{k,zz} \end{bmatrix} . \tag{3}$$

Inertial parameters are often estimated using linear regression techniques, therefore the inertia tensor calculated about the link frame k, $\boldsymbol{L}_k$,

$$\boldsymbol{L}_k \equiv \begin{bmatrix} L_{k,xx} & L_{k,xy} & L_{k,xz} \\ L_{k,xy} & L_{k,yy} & L_{k,yz} \\ L_{k,xz} & L_{k,yz} & L_{k,zz} \end{bmatrix} , \tag{4}$$

has to be used instead of $\boldsymbol{I}_k$ (Khalil and Dombre, 2004). Considering that link and center of mass frames are parallel, the relation between $\boldsymbol{L}_k$ and $\boldsymbol{I}_k$, given by the Huygens–Steiner theorem, is

$$\boldsymbol{L}_k = \boldsymbol{I}_k + m_k \, \mathbf{S}(\boldsymbol{r}_k)^T \, \mathbf{S}(\boldsymbol{r}_k) , \tag{5}$$

where $\mathbf{S}(\cdot)$ is the skew-symmetric matrix operator,

$$\mathbf{S}(\boldsymbol{x}) \equiv \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad \text{for} \quad \boldsymbol{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} . \tag{6}$$

The vector given by

$$\boldsymbol{r}_k \equiv \begin{bmatrix} r_{k,x} \\ r_{k,y} \\ r_{k,z} \end{bmatrix} \tag{7}$$

is the center of mass relative to the link frame k. The matrix $\boldsymbol{L}_k$ is positive definite since it is the sum of a positive definite matrix with a positive semidefinite matrix. Furthermore, in the linear to parameters form, the first moment of inertia vector, $\boldsymbol{l}_k$,

$$\boldsymbol{l}_k \equiv \begin{bmatrix} l_{k,x} \\ l_{k,y} \\ l_{k,z} \end{bmatrix} = m_k \, \boldsymbol{r}_k = \begin{bmatrix} m_k \, r_{k,x} \\ m_k \, r_{k,y} \\ m_k \, r_{k,z} \end{bmatrix} , \tag{8}$$

has to be used instead of $\boldsymbol{r}_k$ (Khalil and Dombre, 2004). From (8) and (5), and knowing that $m_k > 0$, we get

$$\boldsymbol{L}_k = \boldsymbol{I}_k + m_k \, \mathbf{S}\left(\frac{\boldsymbol{l}_k}{m_k}\right)^T \mathbf{S}\left(\frac{\boldsymbol{l}_k}{m_k}\right) . \tag{9}$$

Since $\mathbf{S}(\cdot)$ is a linear operator, (9) can be rewritten as

$$\boldsymbol{I}_k = \boldsymbol{L}_k - \frac{1}{m_k} \mathbf{S}(\boldsymbol{l}_k)^T \, \mathbf{S}(\boldsymbol{l}_k) . \tag{10}$$

The set of all physical feasibility constraints for each link k given by (2) can then be written as

$$\begin{cases} m_k > 0 \\ \boldsymbol{L}_k - \dfrac{1}{m_k} \mathbf{S}(\boldsymbol{l}_k)^T \, \mathbf{S}(\boldsymbol{l}_k) \succ 0 \end{cases} . \tag{11}$$

The vector $\boldsymbol{\delta}$ of all inertial parameters of a serial robot with N links is given by

$$\boldsymbol{\delta} \equiv \begin{bmatrix} \boldsymbol{\delta}_1^T & \boldsymbol{\delta}_2^T & \cdots & \boldsymbol{\delta}_k^T & \cdots & \boldsymbol{\delta}_N^T \end{bmatrix}^T , \tag{12}$$

where, for each link k,

$$\boldsymbol{\delta}_k \equiv [L_{k,xx} \; L_{k,xy} \; L_{k,xz} \; L_{k,yy} \; L_{k,yz} \; L_{k,zz} \; l_{k,x} \; l_{k,y} \; l_{k,z} \; m_k]^T . \tag{13}$$

The vector $\boldsymbol{\delta}$ is the *link inertial parameter* vector[2] (Yoshida and Khalil, 2000) and, from (12) and (13), has size $n = 10N$. The set of all *physically feasible link parameter* vectors, $\mathcal{D}$, can then be defined as

$$\mathcal{D} \equiv \{ \boldsymbol{\delta} \in \mathbb{R}^n : m_k > 0, \boldsymbol{L}_k - m_k^{-1} \mathbf{S}(\boldsymbol{l}_k)^T \, \mathbf{S}(\boldsymbol{l}_k) \succ 0 \; | \; k = 1, \ldots, N \} . \tag{14}$$

Making use of *Sylvester's criterion*, the constraints of (14) can be written as a system of polynomial inequalities in $\boldsymbol{\delta}$. Therefore, $\mathcal{D}$ is a semialgebraic set.

[1] The notation $\boldsymbol{A} \succ 0$ means that $\boldsymbol{A}$ is positive definite.

[2] *Link parameter* vector for short.

3 Physical Feasibility of Inertial Parameters as a Linear Matrix Inequality

This section addresses physical feasibility of robot links, showing that $\mathcal{D}$ is a convex set since it can be defined in the framework of LMIs, a common representation in control theory and in SDP.

3.1 Physical Feasibility of Each Robot Link

For each robot link, the second inequality of (11) can be written as a quadratic matrix inequality,

$$\boldsymbol{L}_k - \mathbf{S}(\boldsymbol{l}_k)^T \, (m_k \mathbf{1})^{-1} \, \mathbf{S}(\boldsymbol{l}_k) \succ 0 \ , \tag{15}$$

where $\mathbf{1}$ is the identity matrix of proper size (size 3, in this case). Through the *Schur complement* condition for positive definite matrices (Boyd et al., 1994; Goldman and Ramana, 1995), we can state that constraint (11) is equivalent[3] to

$$\begin{bmatrix} \boldsymbol{L}_k & \mathbf{S}(\boldsymbol{l}_k)^T \\ \mathbf{S}(\boldsymbol{l}_k) & m_k \mathbf{1} \end{bmatrix} \succ 0 \ . \tag{16}$$

Defining the matrix $\bar{\mathbf{D}}_k(\boldsymbol{\delta}_k)$ as

$$\bar{\mathbf{D}}_k(\boldsymbol{\delta}_k) \equiv \begin{bmatrix} \boldsymbol{L}_k & \mathbf{S}(\boldsymbol{l}_k)^T \\ \mathbf{S}(\boldsymbol{l}_k) & m_k \mathbf{1} \end{bmatrix} =$$

$$= \begin{bmatrix} L_{k,xx} & L_{k,xy} & L_{k,xz} & 0 & l_{k,z} & -l_{k,y} \\ L_{k,xy} & L_{k,yy} & L_{k,yz} & -l_{k,z} & 0 & l_{k,x} \\ L_{k,xz} & L_{k,yz} & L_{k,zz} & l_{k,y} & -l_{k,x} & 0 \\ 0 & -l_{k,z} & l_{k,y} & m_k & 0 & 0 \\ l_{k,z} & 0 & -l_{k,x} & 0 & m_k & 0 \\ -l_{k,y} & l_{k,x} & 0 & 0 & 0 & m_k \end{bmatrix} , \tag{17}$$

(16) is a *linear matrix inequality* (LMI) since it can be written as

$$\bar{\mathbf{D}}_k(\boldsymbol{\delta}_k) \succ 0 \ , \tag{18}$$

[3] Applying *Sylvester's criterion* to the block matrices of (16), $\boldsymbol{L}_k \succ 0$ and $\boldsymbol{L}_k \, m_k \mathbf{1} - \mathbf{S}(\boldsymbol{l}_k)^T \mathbf{S}(\boldsymbol{l}_k) \succ 0$.

with

$$\bar{\mathbf{D}}_k(\boldsymbol{\delta}_k) \equiv \boldsymbol{D}_0 + \sum_{j=1}^{10} \boldsymbol{D}_{k,j} \, \delta_{k,j} \ , \tag{19}$$

where $\boldsymbol{D}_0$ and $\boldsymbol{D}_{k,j}$ are 6×6 symmetric matrices, and each $\delta_{k,j}$ is one element of $\boldsymbol{\delta}_k$ (see (13)). The matrix $\bar{\mathbf{D}}_k(\boldsymbol{\delta}_k)$ is equivalent to the *generalized inertia* of the recursive formulation of robot dynamics using Lie Groups proposed by Park et al. (1995). There are similar representations in other dynamic formulations related to screw theory and using *6-D vector* forms of robot motion and forces, such as the spatial vectors of Featherstone (2010). Equation (18) is a *strict* LMI since $\bar{\mathbf{D}}_k$ is strictly positive definite. In a practical approach we can say that (18) is equivalent to the *nonstrict* LMI[4]

$$\mathbf{D}_k(\boldsymbol{\delta}_k) \succeq 0 \ , \tag{20}$$

with

$$\mathbf{D}_k(\boldsymbol{\delta}_k) \equiv \bar{\mathbf{D}}_k(\boldsymbol{\delta}_k) - \varepsilon \mathbf{1} \ , \tag{21}$$

where ε is an infinitesimally small positive scalar. We will use the nonstrict LMI formulation since it is required for semidefinite programming.

3.2 Physical Feasibility of a Robot with N Links

For a robot with N links, the feasibility condition can be expressed as

$$\mathbf{D}(\boldsymbol{\delta}) \succeq 0 \ , \tag{22}$$

where $\mathbf{D}(\boldsymbol{\delta})$ is a single block-diagonal matrix given by

$$\mathbf{D}(\boldsymbol{\delta}) \equiv \begin{bmatrix} \mathbf{D}_1(\boldsymbol{\delta}_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2(\boldsymbol{\delta}_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_N(\boldsymbol{\delta}_N) \end{bmatrix}_{6N \times 6N} . \tag{23}$$

The set $\mathcal{D}$ of (14) can then be rewritten as

$$\mathcal{D} = \{ \boldsymbol{\delta} \in \mathbb{R}^n : \mathbf{D}(\boldsymbol{\delta}) \succeq 0 \} \ . \tag{24}$$

This set is a *spectrahedron* since it is representable by a nonstrict LMI. Like all spectrahedra, $\mathcal{D}$ is convex.

[4] The notation $\boldsymbol{A} \succeq 0$ means that $\boldsymbol{A}$ is positive semidefinite.

Spectrahedra are the solution spaces of semidefinite programs, which remarkably are guaranteed to converge to the global optimum and can be solved very efficiently (Vandenberghe and Boyd, 1996).

3.3 Additional Constraints and Additional Dynamic Parameters

The set of conditions discussed above guarantee a physical feasible dynamic model. However, the defined feasibility region can be further shrunk to only include solutions respecting particular robot specifications. For example, one can consider upper and lower limits for link masses, $m_{k,l}$ and $m_{k,u}$ respectively,

$$\begin{cases} m_k - m_{k,l} \geq 0 \\ -m_k + m_{k,u} \geq 0 \end{cases} . \tag{25}$$

Constraints in the sum of link masses can also be added. Furthermore, since link center of masses have to be within link convex hulls, one can consider the constraints

$$\begin{cases} \boldsymbol{l}_k - m_k \boldsymbol{r}_{k,l} \geq \boldsymbol{0} \\ -\boldsymbol{l}_k + m_k \boldsymbol{r}_{k,u} \geq \boldsymbol{0} \end{cases} , \tag{26}$$

for a cuboid shaped link, or even constraints for other shapes (e.g., cylindrical). Moreover, it is usual to include additional dynamic effects in the model, such as driver chain frictions and inertias. For each joint k, friction is often modeled as

$$f_k = f_{vk}\, \dot{q}_k + f_{ck}\, \mathrm{sgn}\,(\dot{q}_k) + f_{ok} \ , \tag{27}$$

where f_{vk} and f_{ck} are positive constant parameters for viscous and Coulomb frictions, respectively, and f_{ok} is the Coulomb friction offset which can also include a motor current offset. The drive chain inertia seen by joint k can be represented by a positive constant I_{ak}. When taken into account, these parameters are included into the $\boldsymbol{\delta}$ vector, and positiveness conditions can be included in the feasibility set. Adding more constraints is as simple as appending additional LMI blocks to the diagonal of $\mathbf{D}(\boldsymbol{\delta})$. For example, for friction and drive inertia positiveness constraints, one can define an LMI matrix as

$$\mathbf{D}_{\text{ext}}(\boldsymbol{\delta}) = \mathrm{diag}\big(\mathbf{D}(\boldsymbol{\delta}),\, f_{v1},\, f_{c1},\, I_{a1},\, \ldots \\ \ldots,\, f_{vN},\, f_{cN},\, I_{aN}\big) \ . \tag{28}$$

Indeed, a large range of simple and complex conditions can be easily cast into the LMI formulation (Boyd et al., 1994).

4 Physical Feasibility of Base Inertial Parameters

Given some $\boldsymbol{\delta}$ estimate, $\hat{\boldsymbol{\delta}}$, the verification of its physical feasibility is a matter of checking if $\hat{\boldsymbol{\delta}}$ is in $\mathcal{D}$. This is done in straightforward way by numerically checking the positive definiteness of $\mathbf{D}(\hat{\boldsymbol{\delta}})$. Since the parameter estimation has to be done in base inertial parameters (Gautier and Khalil, 1990; Mayeda et al., 1990) rather than in the link inertial parameter space $\boldsymbol{\delta}$, further methodologies are necessary. In the sequel, we propose a new method taking advantage of semidefinite programming, yielding easy, fast and automatic implementation.

4.1 Base Inertial Parameter Formulation

It is well know that robot dynamics is linear to $\boldsymbol{\delta}$,

$$\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\, \boldsymbol{\delta} = \boldsymbol{\tau} \ , \tag{29}$$

where the matrix $\boldsymbol{H}$ has such a structure that some of its columns are always null and others are always linearly dependent (Khalil and Dombre, 2004). Hence, (29) can be reduced to a minimum number, n_b, of linearly independent columns. Reordering, (29) can be written as

$$\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_b^T & \boldsymbol{\delta}_d^T \end{bmatrix}^T = \boldsymbol{\tau} \ , \tag{30}$$

where $\boldsymbol{H}_b$ has n_b linearly independent columns of $\boldsymbol{H}$, and $\boldsymbol{H}_d$ has the n_d remaining null and dependent columns, with

$$n = n_b + n_d \ . \tag{31}$$

The matrix $\boldsymbol{H}_d$ can be written as a linear combination of $\boldsymbol{H}_b$,

$$\boldsymbol{H}_d = \boldsymbol{H}_b \, \boldsymbol{K}_d \ , \tag{32}$$

where $\boldsymbol{K}_d$ is a constant matrix. Vectors $\boldsymbol{\delta}_b$ and $\boldsymbol{\delta}_d$ are reordered link inertial parameters according to $\boldsymbol{H}_b$ and $\boldsymbol{H}_d$. Equations (30) and (29) are related by

$$\begin{aligned} \boldsymbol{H}_b &\equiv \boldsymbol{H} \, \boldsymbol{P}_b \ , \\ \boldsymbol{H}_d &\equiv \boldsymbol{H} \, \boldsymbol{P}_d \ , \\ \boldsymbol{\delta}_b &\equiv \boldsymbol{P}_b^T \, \boldsymbol{\delta} \ , \\ \boldsymbol{\delta}_d &\equiv \boldsymbol{P}_d^T \, \boldsymbol{\delta} \ , \end{aligned} \tag{33}$$

where $\boldsymbol{P}_b$ and $\boldsymbol{P}_d$ are, respectively, the n_b first columns and the n_d last columns of a permutation matrix $\boldsymbol{P}$ which verifies

$$\boldsymbol{H} \, \boldsymbol{P} = \begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_d \end{bmatrix} \ . \tag{34}$$

The column selection for the basis $\boldsymbol{H}_b$ is not unique, but it is usual to choose the first independent columns of $\boldsymbol{H}$ starting from the left. Equation (30) can then be written as

$$\boldsymbol{H}_b \, \boldsymbol{\beta} = \boldsymbol{\tau} \ , \tag{35}$$

where $\boldsymbol{\beta}$ is the *base inertial parameter* vector (of size n_b),

$$\boldsymbol{\beta} \equiv \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{n_b} \end{bmatrix}^T \ , \tag{36}$$

which is equal to a linear combination of link parameters $\boldsymbol{\delta}$,

$$\boldsymbol{\beta} = \boldsymbol{\delta}_b + \boldsymbol{K}_d \, \boldsymbol{\delta}_d \ . \tag{37}$$

The matrices $\boldsymbol{P}_b$, $\boldsymbol{P}_d$ and $\boldsymbol{K}_d$ can be obtained either by rule based (Gautier and Khalil, 1990; Mayeda et al., 1990) or numerical based methods (Gautier, 1991). From (33), (37) can also be written as

$$\boldsymbol{\beta} = \boldsymbol{K} \, \boldsymbol{\delta} \ , \tag{38}$$

with

$$\boldsymbol{K} = \boldsymbol{P}_b^T + \boldsymbol{K}_d \, \boldsymbol{P}_d^T \ . \tag{39}$$

4.2 Base Inertial Parameter Feasible Set

Equation (38) defines a linear map from $\boldsymbol{\delta}$ ($\mathbb{R}^n$) to $\boldsymbol{\beta}$ ($\mathbb{R}^{n_b}$) spaces. This is not a bijective map, therefore for each $\hat{\boldsymbol{\beta}}$ vector picked from the $\boldsymbol{\beta}$ space, there is more than one solution in the $\boldsymbol{\delta}$ space. Yoshida and Khalil (2000) called this set of $\boldsymbol{\delta}$ solutions the *virtual parameters* of $\hat{\boldsymbol{\beta}}$, which can be defined by $\mathcal{V}_{\hat{\boldsymbol{\beta}}}$,

$$\mathcal{V}_{\hat{\boldsymbol{\beta}}} \equiv \{ \boldsymbol{\delta} \in \mathbb{R}^n : \boldsymbol{K} \, \boldsymbol{\delta} = \hat{\boldsymbol{\beta}} \} \ . \tag{40}$$

Yoshida and Khalil have shown that if there is *at least one* feasible $\boldsymbol{\delta}$ vector in the virtual parameter set $\mathcal{V}_{\hat{\boldsymbol{\beta}}}$ (i.e., that $\mathcal{V}_{\hat{\boldsymbol{\beta}}}$ intersects $\mathcal{D}$), then the respective $\hat{\boldsymbol{\beta}}$ defines a positive definite robot inertia matrix, meaning that $\hat{\boldsymbol{\beta}}$ is physically feasible. Therefore we can define the set of all *physically feasible base parameters*, $\mathcal{B}$, as

$$\mathcal{B} \equiv \{ \boldsymbol{\beta} \in \mathbb{R}^{n_b} : \exists \, \boldsymbol{\delta} \in \mathcal{D} \mid \boldsymbol{K} \, \boldsymbol{\delta} = \boldsymbol{\beta} \} \ . \tag{41}$$

This definition cannot be directly used to verify if a given $\hat{\boldsymbol{\beta}}$ vector is in $\mathcal{B}$ since $\boldsymbol{K}$ is not square, and there is an existential quantifier ($\exists$) over a continuous set. Bellow, we reformulate (41) in a semidefinite representation to overcome this problem. Although $\boldsymbol{\beta} = \boldsymbol{K} \, \boldsymbol{\delta}$ is not bijective, it is possible to define a bijective mapping. A new map $\mathfrak{m}$ with $\boldsymbol{\delta}$ as domain and $(\boldsymbol{\beta}, \boldsymbol{\delta}_d)$ as codomain can be defined by

$$\begin{aligned} \mathfrak{m} : \quad & \mathbb{R}^n \to \mathbb{R}^n \\ & \boldsymbol{\delta} \to (\boldsymbol{\beta}, \boldsymbol{\delta}_d) \end{aligned} \ , \tag{42}$$

where

$$\mathfrak{m}(\boldsymbol{\delta}) \equiv \boldsymbol{G} \, \boldsymbol{\delta} \ , \tag{43}$$

and

$$\boldsymbol{G} \equiv \boldsymbol{K}_G \, \boldsymbol{P}^T \ , \tag{44}$$

with

$$\boldsymbol{K}_G \equiv \begin{bmatrix} \boldsymbol{1} & \boldsymbol{K}_d \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \ . \tag{45}$$

Multiplying $\boldsymbol{G}$ by $\boldsymbol{\delta}$,

$$\begin{aligned} \boldsymbol{G} \, \boldsymbol{\delta} &= \boldsymbol{K}_G \, \boldsymbol{P}^T \, \boldsymbol{\delta} = \\ &= \begin{bmatrix} \boldsymbol{1} & \boldsymbol{K}_d \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_b \\ \boldsymbol{\delta}_d \end{bmatrix} = \\ &= \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta}_d \end{bmatrix} \equiv (\boldsymbol{\beta}, \boldsymbol{\delta}_d) \ . \end{aligned} \tag{46}$$

Since $\boldsymbol{P}^T$ is a permutation matrix and $\boldsymbol{K}_G$ is upper triangular with non zero diagonal elements, it follows that $\boldsymbol{G}$ is invertible, hence the map $\mathfrak{m}$ is bijective. Therefore, there is a one-to-one correspondence between *extended base parameters* $(\boldsymbol{\beta}, \boldsymbol{\delta}_d)$, and link parameters $\boldsymbol{\delta}$. Since Goldman and Ramana (1995) proved that a spectrahedral set remains spectrahedral if a bijective affine transformation is applied, a new *physically feasible extended base parameter* set, $\mathcal{D}_\beta$, can be define as

$$\mathcal{D}_\beta \equiv \{(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \in \mathbb{R}^{n_b+n_d} : \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0\} \ , \quad (47)$$

where $\mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0$ is the LMI of $\mathcal{D}_\beta$. The new matrix function $\mathbf{D}_\beta$ is obtained from (23),

$$\mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \equiv \mathbf{D}\left(\mathfrak{m}^{-1}(\boldsymbol{\beta}, \boldsymbol{\delta}_d)\right) \ , \quad (48)$$

where

$$\begin{aligned} \mathfrak{m}^{-1}(\boldsymbol{\beta}, \boldsymbol{\delta}_d) &= \boldsymbol{G}^{-1} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta}_d \end{bmatrix} \\ &= \boldsymbol{P}^{T^{-1}} \boldsymbol{G}_K^{-1} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta}_d \end{bmatrix} \\ &= \boldsymbol{P} \begin{bmatrix} \mathbf{1} & -\boldsymbol{K}_d \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta}_d \end{bmatrix} \ . \end{aligned} \quad (49)$$

From (47), (41) can then be written as

$$\mathcal{B} = \{\boldsymbol{\beta} \in \mathbb{R}^{n_b} : \exists\, \boldsymbol{\delta}_d \in \mathbb{R}^{n_d} \mid \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0\} \ . \quad (50)$$

Equation (50) still includes an existential quantifier over $\boldsymbol{\delta}_d$. Since $\mathbf{D}_\beta$ is positive definite and semialgebraic, by *Tarski's Theorem* we know that $\mathcal{B}$ is also semialgebraic and its closed formula exists, not requiring the computation of $\boldsymbol{\delta}_d$. To obtain the closed formula, quantifier elimination can be done using the *cylindrical algebraic decomposition* (CAD) method (Collins, 1975). However, this algorithm has double exponential complexity associated to the space dimension. For small planar robots, where the number of inertial parameters can be cut down to 4 per link, the CAD algorithm is capable of finding the closed-form conditions. However, for a common robot manipulator, the number of parameters grows to several dozens, rendering a computationally intractable CAD problem. Nonetheless, as we will show in the next section, (50) can still be used to verify base parameter feasibility of general robots, recurring not to CAD but to SDP.

4.3 Verification of Physical Feasibility

From (47) and (50) we can infer that $\mathcal{B}$ is a projection of $\mathcal{D}_\beta$ onto the $\boldsymbol{\beta}$ space. Projections of spectrahedra are known to be convex and semialgebraic, but, unlike bijective affine transformations, they do not necessarily conserve a spectrahedral form (Goldman and Ramana, 1995). Even so, such projections, known as *semidefinite representable* (or *SDP representable*) sets, still benefit from all the advantages of semidefinite programming techniques (Nesterov and Nemirovskii, 1987). In this context, (50) is a semidefinite representation of $\mathcal{B}$, $\mathcal{D}_\beta$ is an *SDP lift* of $\mathcal{B}$, and $\mathbf{D}_\beta$ is a *lifted LMI* for $\mathcal{B}$.

4.3.1 SDP Method for Base Parameter Feasibility Test (BPFT)

A given $\hat{\boldsymbol{\beta}}$ estimation is feasible if it exists at least one solution in the space of $\boldsymbol{\delta}_d$, which in conjunction to $\hat{\boldsymbol{\beta}}$ belongs to $\mathcal{D}_\beta$ (see (47) and (50)). This can be represented as an *SDP feasibility problem*, i.e.,

$$\begin{aligned} \text{find} \quad & \boldsymbol{\delta}_d \\ \text{subject to} \quad & \mathbf{D}_\beta(\hat{\boldsymbol{\beta}}, \boldsymbol{\delta}_d) \succeq 0 \ . \end{aligned} \quad (51)$$

Problem (51) is easily solved with any standard SDP software. The LMI $\mathbf{D}_\beta(\hat{\boldsymbol{\beta}}, \boldsymbol{\delta}_d) \succeq 0$ defines a set in the space of $\boldsymbol{\delta}_d$. If such set is not empty, the SDP software finds one $\boldsymbol{\delta}_d$ solution, implying that $\hat{\boldsymbol{\beta}}$ is feasible. If the set is empty, the SDP software finds a certificate of infeasibility. In Figure 1, an example of feasible and infeasible vectors is given, showing sets and their relations.

4.3.2 SDP Method for Base Parameter Feasibility Correction (BPFC)

In addition to feasibility check, we can extend the method to find the feasible base parameter vector $\boldsymbol{\beta}'$ which is closest to a given base parameter solution $\hat{\boldsymbol{\beta}}$. Such problem is then formulated as

$$\begin{aligned} \underset{(\boldsymbol{\beta}, \boldsymbol{\delta}_d)}{\text{minimize}} \quad & \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\| \\ \text{subject to} \quad & \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0 \ , \end{aligned} \quad (52)$$

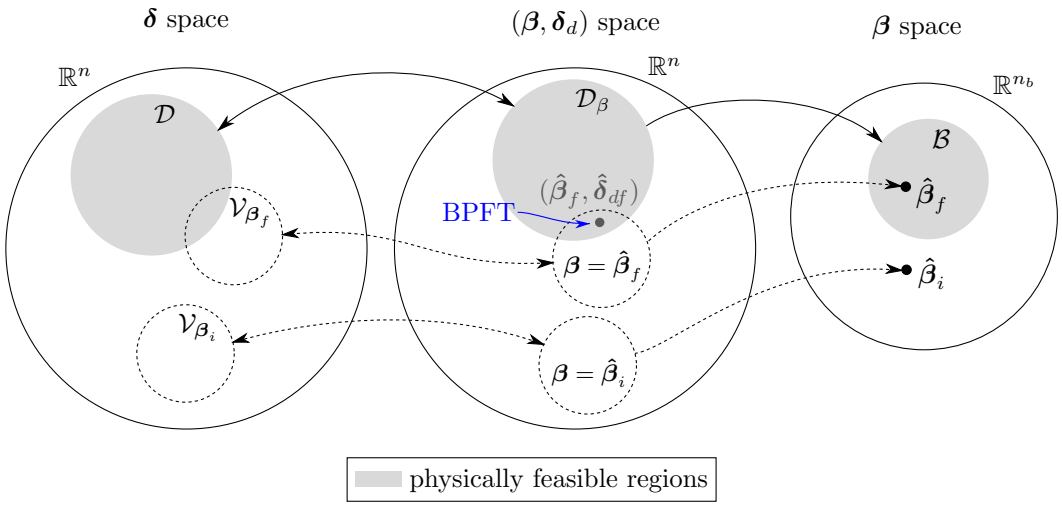

Figure 1: Relation between feasible sets in $\boldsymbol{\delta}$, $(\boldsymbol{\beta}, \boldsymbol{\delta}_d)$ and $\boldsymbol{\beta}$ spaces (2D illustration). $\hat{\boldsymbol{\beta}}_f$ is a feasible parameter vector and $\hat{\boldsymbol{\beta}}_i$ is infeasible. The BPFT algorithm applied to $\hat{\boldsymbol{\beta}}_f$ is able to find a $\hat{\boldsymbol{\delta}}_{df}$ for which $(\hat{\boldsymbol{\beta}}_f, \hat{\boldsymbol{\delta}}_{df})$ is inside the feasibility set $\mathcal{D}_\beta$.

which is not in SDP form. Nevertheless, problem (52) is equivalent to

$$\begin{aligned} \underset{(u,\boldsymbol{\beta},\boldsymbol{\delta}_d)}{\text{minimize}} \quad & u \\ \text{subject to} \quad & u \geq \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2 \\ & \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0 \quad , \end{aligned} \tag{53}$$

where u is an upper bound of the square distance. Using *Schur complement* condition for positive definite matrices, the u inequality of (53) can be written as an LMI,

$$\begin{aligned} & \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2 \leq u \\ \Leftrightarrow \quad & u - (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \geq 0 \\ \Leftrightarrow \quad & \begin{bmatrix} u & (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \\ \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} & \mathbf{1} \end{bmatrix} \succeq 0 \\ \Leftrightarrow \quad & \mathbf{U}_{\hat{\boldsymbol{\beta}}}(u, \boldsymbol{\beta}) \succeq 0 \; . \end{aligned} \tag{54}$$

Therefore, (52) can be reformulated as an SDP problem, i.e.,

$$\begin{aligned} \underset{(u,\boldsymbol{\beta},\boldsymbol{\delta}_d)}{\text{minimize}} \quad & u \\ \text{subject to} \quad & \mathbf{F}_{\hat{\boldsymbol{\beta}}}(u, \boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0 \quad , \end{aligned} \tag{55}$$

where $\mathbf{F}_{\hat{\boldsymbol{\beta}}}(u, \boldsymbol{\beta}, \boldsymbol{\delta}_d)$ is a block-diagonal matrix given by

$$\mathbf{F}_{\hat{\boldsymbol{\beta}}}(u, \boldsymbol{\beta}, \boldsymbol{\delta}_d) = \begin{bmatrix} \mathbf{U}_{\hat{\boldsymbol{\beta}}}(u, \boldsymbol{\beta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \end{bmatrix} \; . \tag{56}$$

Being $(u', \boldsymbol{\beta}', \boldsymbol{\delta}'_d)$ the solution of (55), $\boldsymbol{\beta}'$ is the physically feasible base parameter vector closest to $\hat{\boldsymbol{\beta}}$, and u' is the respective squared distance. If u' is zero, then $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}'$ meaning that $\hat{\boldsymbol{\beta}}$ is feasible.

4.4 Optimal Estimation of Base Parameters

In this section we propose an SDP solution to find the base parameters within the feasible space which best fits regression data. The SDP approach guarantees fast and accurate convergence. One of the

methods to identify the dynamic model is to move the robot along an identification trajectory (i.e., an excitation trajectory used for parameter identification), doing regression over joint and torque data (Khalil and Dombre, 2004). Having a trajectory with s points (with $s \gg 1$), the motion data is collected into a regression matrix $\boldsymbol{W}$ and a vector $\boldsymbol{\omega}$, where

$$\boldsymbol{W} \equiv \begin{bmatrix} \boldsymbol{H}_b(\boldsymbol{q}_1, \dot{\boldsymbol{q}}_1, \ddot{\boldsymbol{q}}_1) \\ \boldsymbol{H}_b(\boldsymbol{q}_2, \dot{\boldsymbol{q}}_2, \ddot{\boldsymbol{q}}_2) \\ \vdots \\ \boldsymbol{H}_b(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s, \ddot{\boldsymbol{q}}_s) \end{bmatrix} , \tag{57}$$

and

$$\boldsymbol{\omega} \equiv \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \vdots \\ \boldsymbol{\tau}_s \end{bmatrix} . \tag{58}$$

The estimated base parameters are the ones that minimize the residuals $\boldsymbol{\epsilon}$ in

$$\boldsymbol{W}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \boldsymbol{\omega} . \tag{59}$$

Such minimization can be done using ordinary least squares (OLS), minimizing the sum of squared residuals given by

$$\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta}\|^2 . \tag{60}$$

The $\boldsymbol{\beta}$ estimation is the solution of the problem

$$\begin{aligned} \underset{(u,\boldsymbol{\beta})}{\text{minimize}} \quad & u \\ \text{subject to} \quad & u \geq \|\boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta}\|^2 \quad , \end{aligned} \tag{61}$$

which, using *Schur complement* in a similar way to (54), can be written as an SDP problem as

$$\begin{aligned} \underset{(u,\boldsymbol{\beta})}{\text{minimize}} \quad & u \\ \text{subject to} \quad & \mathbf{U}_{\boldsymbol{\omega}}(u, \boldsymbol{\beta}) \succeq 0 \quad , \end{aligned} \tag{62}$$

where

$$\mathbf{U}_{\boldsymbol{\omega}}(u, \boldsymbol{\beta}) \equiv \begin{bmatrix} u & (\boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta})^T \\ \boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta} & \mathbf{1} \end{bmatrix} . \tag{63}$$

The square matrix $\mathbf{U}_{\boldsymbol{\omega}}$ has $sN + 1$ rows, which can be too big for SDP solutions.

4.4.1 SDP Method for Feasible Base Parameter Estimation with Ordinary Least Squares (FBPE-OLS)

Given the QR decomposition of the rectangular matrix $\boldsymbol{W}$,

$$\boldsymbol{W} = \boldsymbol{Q}\,\boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1 \\ \mathbf{0} \end{bmatrix} = \boldsymbol{Q}_1\,\boldsymbol{R}_1 , \tag{64}$$

it is known that

$$\|\boldsymbol{Q}^T\boldsymbol{\epsilon}\|^2 = (\boldsymbol{Q}^T\boldsymbol{\epsilon})^T(\boldsymbol{Q}^T\boldsymbol{\epsilon}) = \boldsymbol{\epsilon}^T\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = \|\boldsymbol{\epsilon}\|^2 , \tag{65}$$

since $\boldsymbol{Q}$ is orthogonal. From (60), (64), and (65),

$$\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{Q}^T\boldsymbol{\omega} - \boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{\beta}\|^2 = \left\| \begin{bmatrix} \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_2^T \end{bmatrix} \boldsymbol{\omega} - \begin{bmatrix} \boldsymbol{R}_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{\beta} \right\|^2 . \tag{66}$$

Defining

$$\boldsymbol{\rho}_1 \equiv \boldsymbol{Q}_1^T\boldsymbol{\omega} , \tag{67}$$

$$\boldsymbol{\rho}_2 \equiv \boldsymbol{Q}_2^T\boldsymbol{\omega} \tag{68}$$

and

$$\boldsymbol{\rho} \equiv \begin{bmatrix} \boldsymbol{\rho}_1 \\ \boldsymbol{\rho}_2 \end{bmatrix} , \tag{69}$$

we get the equivalence

$$\|\boldsymbol{\epsilon}\|^2 = \left\| \begin{bmatrix} \boldsymbol{\rho}_1 \\ \boldsymbol{\rho}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{\beta} \right\|^2 = \|\boldsymbol{\rho}_2\|^2 + \|\boldsymbol{\rho}_1 - \boldsymbol{R}_1\boldsymbol{\beta}\|^2 , \tag{70}$$

where the term $\|\boldsymbol{\rho}_2\|^2$ is not dependent on $\boldsymbol{\beta}$. From (60), (61), and (70),

$$u - \|\boldsymbol{\rho}_2\|^2 \geq \|\boldsymbol{\rho}_1 - \boldsymbol{R}_1\boldsymbol{\beta}\|^2 . \tag{71}$$

Hence, the SDP problem of (62) is in fact equivalent to

$$\begin{aligned} \underset{(u,\boldsymbol{\beta})}{\text{minimize}} \quad & u \\ \text{subject to} \quad & \mathbf{U}_{\boldsymbol{\rho}}(u, \boldsymbol{\beta}) \succeq 0 \quad , \end{aligned} \tag{72}$$

where

$$\mathbf{U}_{\boldsymbol{\rho}}(u, \boldsymbol{\beta}) \equiv \begin{bmatrix} u - \|\boldsymbol{\rho}_2\|^2 & (\boldsymbol{\rho}_1 - \boldsymbol{R}_1\,\boldsymbol{\beta})^T \\ \boldsymbol{\rho}_1 - \boldsymbol{R}_1\,\boldsymbol{\beta} & \mathbf{1} \end{bmatrix} . \tag{73}$$

The new LMI matrix $\mathbf{U}_{\boldsymbol{\rho}}$ is square with $n_b + 1$ rows, being much smaller than $\mathbf{U}_{\boldsymbol{\omega}}$, and therefore computationally tractable.

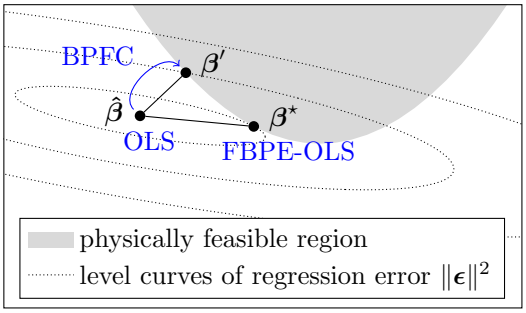

Figure 2: Regression with a non-feasible $\hat{\boldsymbol{\beta}}$ (2D illustration). The always feasible and optimal solution $\boldsymbol{\beta}^\star$ has lower regression error than the feasible solution $\boldsymbol{\beta}'$ closer to $\hat{\boldsymbol{\beta}}$.

By including the constraint $\mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0$ into problem (72), the physically feasible base parameter vector $\boldsymbol{\beta}^\star$ which minimizes the sum of squared residuals is given by

$$\begin{aligned} \underset{(u,\boldsymbol{\beta},\boldsymbol{\delta}_d)}{\text{minimize}} \quad & u \\ \text{subject to} \quad & \mathbf{F}_{\boldsymbol{\rho}}(u, \boldsymbol{\beta}, \boldsymbol{\delta}_d) \succeq 0 \quad , \end{aligned} \tag{74}$$

where

$$\mathbf{F}_{\boldsymbol{\rho}}(u, \boldsymbol{\beta}, \boldsymbol{\delta}_d) = \begin{bmatrix} \mathbf{U}_{\boldsymbol{\rho}}(u, \boldsymbol{\beta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\beta(\boldsymbol{\beta}, \boldsymbol{\delta}_d) \end{bmatrix} . \tag{75}$$

The solution to this problem is $(u^\star, \boldsymbol{\beta}^\star, \boldsymbol{\delta}_d^\star)$, where $u^\star$ is the sum of squared residuals of (59) with $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$. If $\boldsymbol{\beta}^\star$ is equal to the unconstrained estimation $\hat{\boldsymbol{\beta}}$ of (72) then $\hat{\boldsymbol{\beta}}$ is feasible. In this case, $\boldsymbol{\beta}^\star$ is also equal to the solution $\boldsymbol{\beta}'$ given by (55). If $\hat{\boldsymbol{\beta}} \neq \boldsymbol{\beta}^\star$, then $\hat{\boldsymbol{\beta}}$ is infeasible, and the feasible solution $\boldsymbol{\beta}'$ is not necessarily equal to $\boldsymbol{\beta}^\star$, as illustrated in Figure 2. The fact that the regression errors of $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}^\star$ are always bigger than the $\hat{\boldsymbol{\beta}}$ error (when $\hat{\boldsymbol{\beta}}$ is infeasible), it does not necessarily mean that such estimations entail worse models. Indeed, the question of whether these estimations are worse or better than physically infeasible ones can be related to the kind of desired model — either a *prediction* model, seeking for the lowest prediction error, or a *structural* model, seeking for a meaningful physical system description (Hollerbach et al., 2008). Although physically infeasible estimations can be acceptable for prediction purposes, such estimations do not meet physical requirements of structural models, being unsuitable for control and simulation.

4.5 Parameter Space Issues

The methods BPFT, BPFC and FBPE-OLS have been developed on the base parameter space $\boldsymbol{\beta}$ plus the nullspace $\boldsymbol{\delta}_d$. With straightforward algebraic manipulation, it is possible to directly write these methods on the link parameter space $\boldsymbol{\delta}$. The drawback of $\boldsymbol{\delta}$ solutions is that they are a mix of identifiable values and nullspace values, which contain redundant and arbitrary information. With $(\boldsymbol{\beta},\ \boldsymbol{\delta}_d)$ solutions, informative and uninformative terms are split into $\boldsymbol{\beta}$ and $\boldsymbol{\delta}_d$, respectively. One can still map $(\boldsymbol{\beta},\ \boldsymbol{\delta}_d)$ to the $\boldsymbol{\delta}$ space through the inverse mapping $\mathfrak{m}^{-1}$ given by (49). Regarding the non-unique choice of base parameter combinations, feasibility/infeasibility is maintained when mapping between different base parameter spaces. Two base parameter vectors on two different base spaces are equivalent when they share the same virtual parameter set (see (40)). Since feasibility is dependent on the virtual parameter set, both base parameter vectors also share the feasibility/infeasibility property. Therefore, the BPFT method is invariant with respect to the choice of parameter grouping. Since the regression error of (60) is not affected by base parameter regrouping (a transformation in $\boldsymbol{\beta}$ implies canceling operations in $\boldsymbol{W}$), the FBPE-OLS method gives equivalent solutions for different $(\boldsymbol{\beta}, \boldsymbol{\delta_d})$ spaces. Solutions of the BPFC method are however dependent on parameter grouping. This is due to the fact that the BPFC method is based on measured distances in base parameter space which are in general not linearly mapped between different spaces. Nevertheless, the BPFC method can be applied to correct an unfeasible solution for which regression data is no longer available. Moreover, the $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}'\|$ distance given by BPFC can be used as a measurement of infeasibility.

5 Experimental Tests

Two applications are presented in this section. The first one addresses the feasibility test of a 3-link robot using the BPFT method. The second one analyzes the three methods proposed in this paper using the 7-link WAM$^{\text{TM}}$ robot. The proposed methods were deployed in *Python* and rely on open source solutions, only. The LMI is firstly defined by a symbolic matrix using the *Python SymPy* symbolic library. This allows easy mapping from the $\boldsymbol{\delta}$ space to the $(\boldsymbol{\beta}, \boldsymbol{\delta}_d)$ space. Then, recurring to the *PyLMI-SDP* package[5] developed by the authors, the coefficient matrices are extracted from the symbolic LMI and saved into the SDPA sparse file format. Such format is accepted by a large number of SDP solvers which directly exploit the sparsity of LMI matrices. For the BPFT method, the "find" problem of (51) is implemented by setting an objective function equal to zero. In our experiments, the SDP optimization is performed by the *DSDP5* solver (Benson and Ye, 2008), which is based on the *interior-point* method as many other SDP solvers are. The *DSDP5* solver has a tolerance parameter to define the solution accuracy. We set such parameter to 10^{-7}. Additionally, LMI strictness is enforced with a safe margin of $\varepsilon = 10^{-6}$ (see (21)).

5.1 Base Parameter Feasibility Test of a 3-Link Robot

We applied our BPFT and BPFC methods to the example included in Yoshida and Khalil (2000). Yoshida and Khalil introduced a base parameter combination $\boldsymbol{\beta}$ for a 3-link robot and presented two examples of base parameter vector values, $\hat{\boldsymbol{\beta}}_{t1}$ and $\hat{\boldsymbol{\beta}}_{t2}$ (see Table 1). They have shown that $\hat{\boldsymbol{\beta}}_{t1}$ is feasible and $\hat{\boldsymbol{\beta}}_{t2}$ is not. We double checked $\hat{\boldsymbol{\beta}}_{t1}$ and $\hat{\boldsymbol{\beta}}_{t2}$ with the BPFT method. For $\hat{\boldsymbol{\beta}}_{t1}$, a $\boldsymbol{\delta}_d$ solution is found, which implies physical feasibility. In the case of $\hat{\boldsymbol{\beta}}_{t2}$ a certificate of infeasibility is issued by the SDP solver. These results corroborate Yoshida and Khalil ones. Furthermore, we applied the BPFC method to $\hat{\boldsymbol{\beta}}_{t2}$ for which the closest feasible parameter vector $\boldsymbol{\beta}'_{t2}$ is found (see Table 1). The distance from $\boldsymbol{\beta}'_{t2}$ to $\hat{\boldsymbol{\beta}}_{t2}$, measured in $\boldsymbol{\beta}$ space, is given by $\sqrt{u'} = 1.65 \times 10^{-3}$. As expected, $\boldsymbol{\beta}'_{t2}$ passes the BPFT feasibility test. For this example no regression data is available, so the FBPE-OLS method could not be tested.

Table 1: Base parameter estimations of the 3-link robot example provided by Yoshida and Khalil (2000)

$\boldsymbol{\beta}(\boldsymbol{\delta})$	$\hat{\boldsymbol{\beta}}_{t1}$	$\hat{\boldsymbol{\beta}}_{t2}$	$\boldsymbol{\beta}'_{t2}$
$L_{1yy} + L_{2yy} + L_{3yy} + m_3$	6.4	**6.2**	6.200951
$L_{2xx} - L_{2yy} - m_3$	−5.48	−5.48	−5.479049
L_{2xy}	0.072	0.072	0.071966
$L_{2xz} - l_{3z}$	−0.087	−0.087	−0.086967
L_{2yz}	0.051	0.051	0.050999
$L_{2zz} + m_3$	5.6	5.6	5.600000
$l_{2x} + m_3$	6.5	6.5	6.500000
l_{2y}	−0.00075	−0.00075	−0.000750
$L_{3xx} - L_{3yy}$	−0.72	−0.72	−0.719049
L_{3xy}	−0.0098	−0.0098	−0.009819
L_{3xz}	−0.0098	−0.0098	−0.009817
L_{3yz}	−0.00045	−0.00045	−0.000450
L_{3zz}	0.72	0.72	0.720000
l_{3x}	0.95	0.95	0.949999
l_{3y}	0.015	0.015	0.014966

5.2 Parameter Identification of the 7-Link WAM$^{\text{TM}}$ Arm

We performed experiments with a 7-link robot, the Barrett WAM$^{\text{TM}}$ Arm, to test our LMI-SDP methods[6]. The WAM geometry is described by the Denavit-Hartenberg parameters presented in Table 2. For identification purposes we consider a dynamic model including joint friction in the form of (27) and drive chain inertia as presented in (Khalil and Dombre, 2004). The dynamic model is derived using the author's *SymPy*-

[5] `https://github.com/cdsousa/PyLMI-SDP`

[6] Experimental code and data developed for the 7-link WAM robot are available online at `https://github.com/cdsousa/wam7_dyn_ident`.

Table 2: Denavit-Hartenberg parameters of the 7-link WAM Arm.

Joint k	α_k	a_k	d_k	θ_k
1	$-\pi/2$	0	0	q_1
2	$\pi/2$	0	0	q_2
3	$-\pi/2$	0.045	0.55	q_3
4	$\pi/2$	-0.045	0	q_4
5	$-\pi/2$	0	0.3	q_5
6	$\pi/2$	0	0	q_6
7	0	0	0.06	q_7

Botics software[7], an open source *Python* package for symbolic robot dynamics, based on *SageRobotics* (Sousa and Cortesão, 2012). With such software, the base inertial parameter combination shown in Table 3 is obtained, yielding 69 base parameters ($\boldsymbol{\beta}$) out of 98 link parameters ($\boldsymbol{\delta}$). To obtain dynamic model regression data, a periodic identification trajectory is performed by the robot.

5.2.1 Computed Torque Control

Unlike common industrial robots, the WAM robot is controlled directly in torque. The motors are commanded in current by internal current controllers which receive torque set points. The robot has no torque sensors and does not provide access to motor currents. Thus, we assume good torque to current conversion and good current tracking control. Torque command allows nonlinear feedback linearization and computed torque control. The trajectory is performed through a proportional and derivative (PD) position controller over the linearized robot plant, as shown in Figure 3. The commanded torque $\boldsymbol{\tau}_c$ is computed by

$$\boldsymbol{\tau}_c = \hat{\boldsymbol{g}}(\boldsymbol{q}) + \hat{\boldsymbol{c}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{M}}(\boldsymbol{q})\, \boldsymbol{\alpha} \ , \qquad (76)$$

where $\hat{\boldsymbol{g}}(\boldsymbol{q})$, $\hat{\boldsymbol{c}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\hat{\boldsymbol{M}}(\boldsymbol{q})$ are estimations of $\boldsymbol{g}(\boldsymbol{q})$, $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ based on datasheet dynamic parameters, and $\boldsymbol{\alpha}$ is the desired acceleration. For control purposes we do not considered friction or drive inertia. Neglecting dynamic model errors, the plant for

[7] `https://github.com/cdsousa/SymPyBotics`

Table 3: Base parameters of the 7-link WAM Arm.

$\boldsymbol{\beta}$	corresponding linear combination
β_1	$L_{1yy} + Ia_1 + L_{2zz}$
β_2	fv_1
β_3	fc_1
β_4	fo_1
β_5	$L_{2xx} - L_{2zz} + L_{3zz} - 1.1\, l_{3y} + 0.300475\,(m_3 + m_4 + m_5 + m_6 + m_7)$
β_6	L_{2xy}
β_7	L_{2xz}
β_8	$L_{2yy} + Ia_2 + L_{3zz} - 1.1\, l_{3y} + 0.300475\,(m_3 + m_4 + m_5 + m_6 + m_7)$
β_9	L_{2yz}
β_{10}	l_{2x}
β_{11}	$l_{2z} - l_{3y} + 0.55\,(m_3 + m_4 + m_5 + m_6 + m_7)$
β_{12}	fv_2
β_{13}	fc_2
β_{14}	fo_2
β_{15}	$L_{3xx} - L_{3zz} + 0.002025\, m_3 + L_{4zz}$
β_{16}	$L_{3xy} - 0.045\, l_{3y}$
β_{17}	L_{3xz}
β_{18}	$L_{3yy} - 0.002025\, m_3 + L_{4zz} - 0.00405\,(m_4 + m_5 + m_6 + m_7)$
β_{19}	L_{3yz}
β_{20}	$l_{3x} + 0.045\,(m_3 + m_4 + m_5 + m_6 + m_7)$
β_{21}	$l_{3z} + l_{4y}$
β_{22}	Ia_3
β_{23}	fv_3
β_{24}	fc_3
β_{25}	fo_3
β_{26}	$L_{4xx} - L_{4zz} + 0.002025\, m_4 + L_{5zz} - 0.6\, l_{5y} + 0.092025\,(m_5 + m_6 + m_7)$
β_{27}	$L_{4xy} + 0.045\, l_{4y}$
β_{28}	L_{4xz}
β_{29}	$L_{4yy} - 0.002025\, m_4 + L_{5zz} - 0.6\, l_{5y} + 0.087975\,(m_5 + m_6 + m_7)$
β_{30}	L_{4yz}
β_{31}	$l_{4x} - 0.045\,(m_4 + m_5 + m_6 + m_7)$
β_{32}	$l_{4z} - l_{5y} + 0.3\,(m_5 + m_6 + m_7)$
β_{33}	Ia_4
β_{34}	fv_4
β_{35}	fc_4
β_{36}	fo_4
β_{37}	$L_{5xx} - L_{5zz} + L_{6zz}$
β_{38}	L_{5xy}
β_{39}	L_{5xz}
β_{40}	$L_{5yy} + L_{6zz}$
β_{41}	L_{5yz}
β_{42}	l_{5x}
β_{43}	$l_{5z} + l_{6y}$
β_{44}	Ia_5
β_{45}	fv_5
β_{46}	fc_5
β_{47}	fo_5
β_{48}	$L_{6xx} - L_{6zz} + L_{7yy} + 0.12\, l_{7z} + 0.0036\, m_7$
β_{49}	L_{6xy}
β_{50}	L_{6xz}
β_{51}	$L_{6yy} + L_{7yy} + 0.12\, l_{7z} + 0.0036\, m_7$
β_{52}	L_{6yz}
β_{53}	l_{6x}
β_{54}	$l_{6z} + l_{7z} + 0.06\, m_7$
β_{55}	Ia_6
β_{56}	fv_6
β_{57}	fc_6
β_{58}	fo_6
β_{59}	$L_{7xx} - L_{7yy}$
β_{60}	L_{7xy}
β_{61}	L_{7xz}
β_{62}	L_{7yz}
β_{63}	L_{7zz}
β_{64}	l_{7x}
β_{65}	l_{7y}
β_{66}	Ia_7
β_{67}	fv_7
β_{68}	fc_7
β_{69}	fo_7

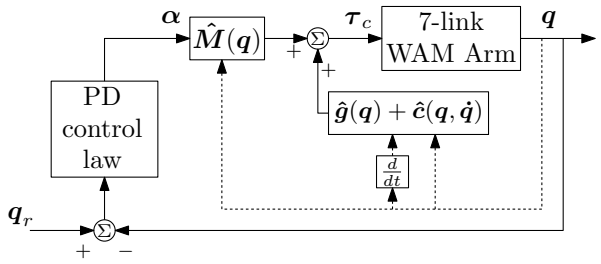

Figure 3: PD joint position control scheme of the 7-link WAM Arm with feedback linearization. $\boldsymbol{q}_r$ is the joint reference position.

Table 4: PD joint position control gains for the 7-link WAM Arm.

Joint k	$K_{p,k}$	$K_{d,k}$
1	400.0	40.0
2	204.1	28.6
3	625.0	50.0
4	400.0	40.0
5	10000.0	200.0
6	10000.0	200.0
7	20408.2	285.7

each joint k in free space is given by (see (1) and (76))

$$\ddot{q}_k = \alpha_k \ . \tag{77}$$

This is equivalent to a double integrator over which the PD controller is designed. PD gains ($K_{p,k}$ and $K_{d,k}$, respectively) given in Table 4 are tuned for critically damped response.

5.2.2 Trajectory Generation and Data Processing

The identification trajectory is generated and optimized using the techniques proposed by Swevers et al. (1997). Each k-th joint trajectory is defined as a function of time t by

$$q_{r,k}(t) = \sum_{l=1}^{L} \frac{a_{k,l}}{\omega_f \, l} \sin(\omega_f \, l \, t) - \frac{b_{k,l}}{\omega_f \, l} \cos(\omega_f \, l \, t) + q_{k,0} \ , \tag{78}$$

Table 5: Identification trajectory parameters for the 7-link WAM Arm.

Joint k	$a_{k,1}$	$a_{k,2}$	$a_{k,3}$	$a_{k,4}$	$a_{k,5}$
1	0.05	−0.29	0.48	0.55	0.65
2	0.03	0.29	−0.23	0.32	0.82
3	−0.07	0.40	0.45	0.40	−0.03
4	0.14	−0.35	0.15	0.11	0.93
5	0.21	0.35	0.16	−0.02	0.03
6	−0.11	0.28	0.36	−0.06	0.33
7	−0.01	0.24	0.37	−0.45	0.75

k	$b_{k,1}$	$b_{k,2}$	$b_{k,3}$	$b_{k,4}$	$b_{k,5}$	$q_{k,0}$
1	0.19	−0.40	−0.18	0.63	−0.46	−0.29
2	0.09	−0.08	0.05	−0.02	0.65	0.11
3	−0.49	0.32	−0.26	−0.63	0.06	−0.02
4	−0.14	0.06	−0.13	−0.14	−0.03	1.67
5	−0.51	0.14	0.37	−0.15	−0.17	−2.41
6	0.13	0.07	0.67	−0.15	−0.22	0.20
7	0.24	0.24	−0.22	−0.50	−0.52	0.58

where, in our experiment,

$$\omega_f = 0.1\pi \tag{79}$$

and

$$L = 5 \ , \tag{80}$$

entailing a period of 20 seconds. Trajectory parameters $a_{k,l}$, $b_{k,l}$ and $q_{k,0}$ are set to the values shown in Table 5. The reference joint positions for one whole period are shown in Figure 4. The robot performs the identification trajectory for almost 1 minute. At a sampling time of 1 millisecond, a total of $s = 57656$ trajectory points (joint position and torque) are recorded. Position data comes from encoder readings and torque data comes from the computed torque $\boldsymbol{\tau}_c$ (see (76)). Joint position signals are filtered with third order low-pass Butterworth filters with cut frequency $10\omega_f L/2\pi$. This process is done offline, therefore phase distortion is compensated. First and second order derivatives of the position are computed through second order central differences. Applying the corresponding $\boldsymbol{H}_b$ matrix to the data, the regression matrix $\boldsymbol{W}$ and the torque vector $\boldsymbol{\omega}$ are obtained, both with $sN = 403592$ rows. The regression matrix has a condition number of 99, therefore it can be considered well conditioned (Gautier and Khalil, 1992).

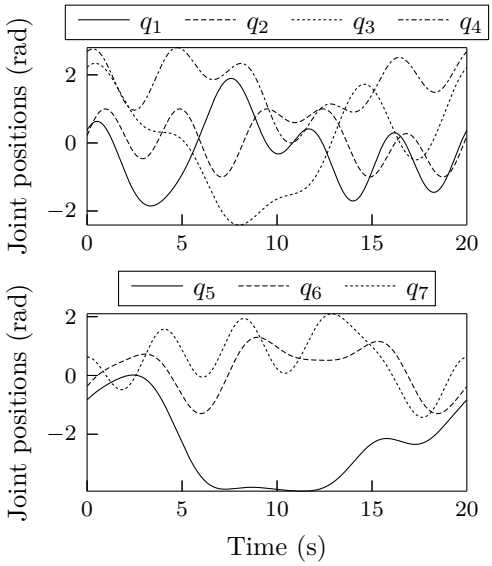

Figure 4: Identification trajectory for the 7-link WAM Arm. Joint position reference for the first period.

5.2.3 Classical Base Parameter Estimation

Using the identification trajectory data, $\boldsymbol{W}$ and $\boldsymbol{\omega}$, the classical OLS solution $\hat{\boldsymbol{\beta}}$ given in Table 6 is obtained. Datasheet parameters used in the controller, $\boldsymbol{\beta}_{\text{CAD}}$, are also shown in Table 6. The vector of $\hat{\boldsymbol{\beta}}$ standard deviations is computed by the formula (Khalil and Dombre, 2004)

$$\hat{\sigma}_{\hat{\boldsymbol{\beta}}} = \sqrt{\text{diag}\left(\hat{\sigma}_{\boldsymbol{\epsilon}}^2 \left(\boldsymbol{W}^T \boldsymbol{W}\right)^{-1}\right)} \,, \tag{81}$$

where $\hat{\sigma}_{\boldsymbol{\epsilon}}^2$ is the variance of the estimation error $\boldsymbol{\epsilon}$ (see (59)) given by

$$\hat{\sigma}_{\boldsymbol{\epsilon}}^2 = \frac{\|\boldsymbol{\omega} - \boldsymbol{W}\hat{\boldsymbol{\beta}}\|^2}{sN - n_b} \,. \tag{82}$$

The percentage of relative standard deviation of the $\hat{\boldsymbol{\beta}}$ k-th parameter, given by

$$\%\hat{\sigma}_{\hat{\boldsymbol{\beta}}_k} = \frac{\hat{\sigma}_{\hat{\boldsymbol{\beta}}_k}}{|\hat{\boldsymbol{\beta}}_k|} \cdot 100\% \,, \tag{83}$$

is presented in Table 6. The solution $\hat{\boldsymbol{\beta}}$ is tested with

Table 6: Estimated base parameters of the 7-link WAM Arm.

		OLS		FBPE-OLS
	$\boldsymbol{\beta}_{\text{CAD}}$	$\hat{\boldsymbol{\beta}}$	$\%\hat{\sigma}_{\hat{\boldsymbol{\beta}}}$	$\boldsymbol{\beta}^{\star\text{e}}$
β_1	$Ia_1 + 0.1328$	0.3381	0.53	0.3049
fv_1	—	1.63	0.28	1.648
fc_1	—	1.601	0.26	1.605
fo_1	—	0.01037	18	0.007658
β_5	1.181	1.017	0.29	1.04
L_{2xy}	$1.347 \cdot 10^{-5}$	−0.0505	3.2	−0.05561
L_{2xz}	0.000117	−0.02539	4.8	−0.02304
β_8	$Ia_2 + 1.189$	1.451	0.23	1.446
L_{2yz}	$-3.659 \cdot 10^{-5}$	−0.03322	4.3	−0.038
l_{2x}	−0.009183	0.08154	1.7	0.08391
β_{11}	2.333	2.358	0.027	2.36
fv_2	—	2.372	0.21	2.364
fc_2	—	0.533	0.71	0.5332
fo_2	—	0.898	1.2	0.9296
β_{15}	0.003856	−0.01168	20	0.0183
β_{16}	$-2.509 \cdot 10^{-5}$	0.03663	2.9	0.02517
L_{3xz}	$-5.098 \cdot 10^{-6}$	0.002625	45	−0.00827
β_{18}	−0.006954	−0.06614	2.2	0.001251
L_{3yz}	$5.295 \cdot 10^{-6}$	−0.04278	2.5	−0.03598
β_{20}	0.1476	0.1194	0.44	0.1255
β_{21}	−0.000491	−0.002107	18	−0.0009105
Ia_3	—	0.1139	1.3	0.06988
fv_3	—	0.8554	0.52	0.8111
fc_3	—	0.3454	1.1	0.3718
fo_3	—	−0.09566	2.1	−0.08749
β_{26}	0.1149	0.165	0.9	0.1385
β_{27}	$-3.944 \cdot 10^{-5}$	0.01292	5.4	0.009043
L_{4xz}	$-8.193 \cdot 10^{-5}$	−0.02431	2.4	−0.01386
β_{29}	0.1057	0.1213	1.1	0.1239
L_{4yz}	$9.417 \cdot 10^{-5}$	−0.01128	6.2	−0.005383
β_{31}	−0.1235	−0.1057	0.35	−0.101
β_{32}	0.501	0.5209	0.088	0.5174
Ia_4	—	−0.005825	27	$1.009 \cdot 10^{-6}$
fv_4	—	0.6153	0.8	0.6005
fc_4	—	0.6291	0.57	0.6417
fo_4	—	0.05353	5.4	0.03402
β_{37}	0.0005648	−0.008932	12	−0.01554
L_{5xy}	$-2.564 \cdot 10^{-7}$	0.005089	7.4	−0.002231
L_{5xz}	$1.882 \cdot 10^{-9}$	0.0198	2.4	0.01646
β_{40}	0.000653	0.006071	11	0.007886
L_{5yz}	$8.326 \cdot 10^{-7}$	−0.0092	4.5	−0.002773
l_{5x}	$1.104 \cdot 10^{-5}$	0.006003	4.9	0.006159
β_{43}	−0.00658	−0.01351	2	−0.01388
Ia_5	—	−0.0009694	81	$1.004 \cdot 10^{-6}$
fv_5	—	0.1457	3.3	0.1387
fc_5	—	0.03597	7.7	0.03627
fo_5	—	0.004681	43	0.001217
β_{48}	0.0006155	0.003965	20	0.009329
L_{6xy}	$-1.483 \cdot 10^{-6}$	0.001507	22	−0.002418
L_{6xz}	$2.005 \cdot 10^{-6}$	−0.01062	3.4	−0.003334
β_{51}	0.0007576	0.01942	3	0.01731
L_{6yz}	0.0002216	0.00495	7.2	0.004267
l_{6x}	$-5.125 \cdot 10^{-5}$	−0.007862	3.2	−0.006727
β_{54}	0.01421	0.009031	2.4	0.009508
Ia_6	—	−0.008871	10	$1.003 \cdot 10^{-6}$
fv_6	—	0.1437	3.2	0.1344
fc_6	—	0.06428	4.7	0.07144
fo_6	—	−0.007354	26	−0.007547
β_{59}	$-3.286 \cdot 10^{-7}$	−0.01584	3.2	−0.01357
L_{7xy}	$1.909 \cdot 10^{-7}$	−0.002082	12	−0.0009831
L_{7xz}	$-1.771 \cdot 10^{-8}$	0.0002248	61	$6.76 \cdot 10^{-5}$
L_{7yz}	$3.623 \cdot 10^{-8}$	0.0002694	52	0.0007734
L_{7zz}	$7.408 \cdot 10^{-5}$	−0.0005387	29	0.0003626
l_{7x}	$-5.474 \cdot 10^{-6}$	0.00782	2.4	0.007006
l_{7y}	$1.12 \cdot 10^{-5}$	−0.0003508	52	−0.0003549
Ia_7	—	0.0009977	27	0.0001039
fv_7	—	0.05763	5.6	0.05625
fc_7	—	0.0118	27	0.01253
fo_7	—	−0.002192	84	−0.002699

Table 7: Additional constraints: spatial limits of center of masses relative to link frames.

k	$\boldsymbol{r}_{k,l}$	$\boldsymbol{r}_{k,u}$
1	$(-0.14, -0.174, -0.084)$	$(0.14, 0.174, 0.346)$
2	$(-0.084, -0.174, -0.084)$	$(0.084, 0.174, 0.17)$
3	$(-0.09, -0.55, -0.045)$	$(0.04, 0.04, 0.045)$
4	$(-0.045, -0.045, -0.05)$	$(0.095, 0.045, 0.83)$
5	$(-0.045, -0.02, -0.045)$	$(0.045, 0.1, 0.045)$
6	$(-0.045, -0.06, -0.02)$	$(0.045, 0.045, 0.06)$
7	$(-0.045, -0.045, -0.018)$	$(0.045, 0.045, 0.001)$

the BPFT method, set with LMI conditions on both link inertia and drive parameters (see (28)). The SDP solver issues an infeasibility certificate, showing that $\hat{\boldsymbol{\beta}}$ is physically infeasible. For this solution, the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is not positive definite for some random $\boldsymbol{q}$ values, which corroborates physical infeasibility. Moreover, in this particular case, the infeasibility can be directly inferred since some drive inertia parameters are negative.

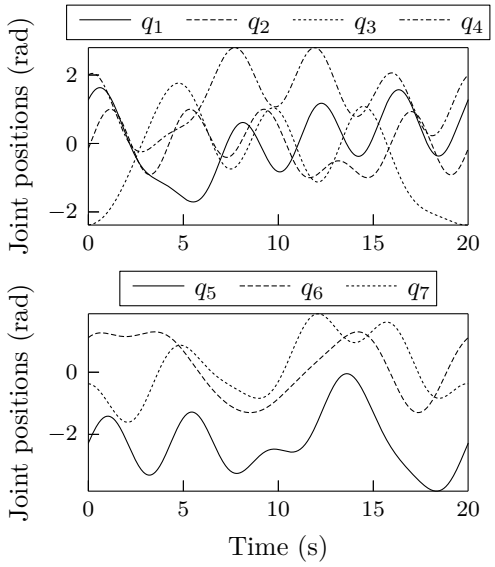

Figure 5: Validation trajectory A for the 7-link WAM Arm. Joint position reference for the first period.

5.2.4 Feasible Base Parameter Estimations

Since the $\hat{\boldsymbol{\beta}}$ solution is not feasible, its closest feasible vector, $\boldsymbol{\beta}'$, is computed by the BPFC method. We also applied the FBPE-OLS method to the same regression data obtaining the optimal solution $\boldsymbol{\beta}^\star$. Based on datasheet information, we added extra constraints to the LMI on center of masses and total mass. Each center of mass constraint is of form (26) with bounds given by the smallest cuboid containing link's body (see Table 7). The sum of link masses is constrained to a maximum of 27 Kg. The solution $\boldsymbol{\beta}^\star$ does not pass the feasibility test (BPFT) if these additional constraints are taken into account. Therefore, we reapplied the FBPE-OLS method with the extra constraints obtaining the $\boldsymbol{\beta}^{\star\mathrm{e}}$ solution shown in Table 6. The feasibility of each of these solutions is successfully double checked by analyzing eigenvalue positiveness of the respective LMI matrices. As expected, the inertia matrix evaluated with these solutions at random $\boldsymbol{q}$ is always positive definite.

Regarding computing time performance, the SDP solver running on a mid-end laptop shows to be very fast, taking less than two seconds to find each solutions[8].

5.2.5 Estimation Assessment

To validate estimated parameters, three validation trajectories, named A, B and C, are generated by the same techniques used for the identification trajectory, although with different initial conditions. As result, validation trajectories have the same harmonic frequencies but with unrelated amplitudes (e.g., Figure 5 shows the joint reference for the validation trajectory A). The robot performs validation trajectories for about 1 minute, and the recorded data is processed in the same way as for the identification trajectory. Regressor matrices for validation trajectories A, B and C show good condition numbers of 94, 106 and 101, respectively. Torque prediction is computed for all four solutions ($\hat{\boldsymbol{\beta}}$, $\boldsymbol{\beta}'$, $\boldsymbol{\beta}^\star$ and $\boldsymbol{\beta}^{\star\mathrm{e}}$). Table 8 presents, in percentage, the relative torque

[8]For a 20-link robot, the SDP solver takes less than 15 seconds to obtain the FBPE-OLS solution.

Table 8: Percentage of relative error ($100\% \cdot \|\boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta}\|/\|\boldsymbol{\omega}\|$) of predicted torque for identification and validation trajectories.

	OLS	BPFC	FBPE-OLS	
	$\hat{\boldsymbol{\beta}}$	$\boldsymbol{\beta}'$	$\boldsymbol{\beta}^\star$	$\boldsymbol{\beta}^{\star \mathrm{e}}$
identification	6.60	6.68	6.64	6.65
validation A	7.16	7.14	7.01	6.98
validation B	8.47	8.49	8.44	8.40
validation C	6.81	6.79	6.76	6.74

error given by

$$\epsilon_r = \frac{\|\boldsymbol{\omega} - \boldsymbol{W}\boldsymbol{\beta}\|}{\|\boldsymbol{\omega}\|} \ , \tag{84}$$

for both identification and validation trajectories. For the identification trajectory (first row of Table 8), all four solutions present values of relative error below 7%, indicating good parameter estimation. As expected, physically feasible constrained estimations show higher regression errors than the unconstrained one $\hat{\boldsymbol{\beta}}$. Also as expected, $\boldsymbol{\beta}^\star$ has less error than $\boldsymbol{\beta}'$, and $\boldsymbol{\beta}^{\star \mathrm{e}}$ has a higher regression error w.r.t. $\boldsymbol{\beta}^\star$. Even so, differences between solutions are small ($< 0.1\%$). The relative prediction errors of validation trajectories (bottom three rows of Table 8) are below 8.5%, indicating also good torque prediction. Feasible estimations present slightly smaller errors w.r.t. the classical OLS solution, showing good performance outside the identification set. Figure 6 shows plots of measured torque, $\boldsymbol{\beta}^{\star \mathrm{e}}$ predicted torque and respective error for trajectory A. It can be inferred that for the first four joints predicted torque tracks well the measured one. For the last three joints some degradation occurs due to smaller torque values, since inertial parameters are relatively small. Looking to Table 6, we can also infer that for the last three joints some $\hat{\boldsymbol{\beta}}$ parameters have big $\%\hat{\sigma}_{\hat{\boldsymbol{\beta}}}$, probably indicating bad estimations. This can suggest that a sequential or a weighted least squares (WLS) approach can improve the identification (Gautier, 1997), although it does not guarantee physical feasibility. In this case, the FBPE-OLS can be easily modified to accommodate a WLS regression.

6 Conclusion

In this paper we have presented a systematic approach to address physical feasibility of estimated robot base inertial parameters. Feasibility conditions have been written with LMIs, ready to be used in the SDP framework. Moreover, feasible sets can be further restricted to cope with additional constraints (e.g., spacial limits of the center of mass and bounded masses) and extended to include additional dynamic parameters (e.g., frictions).

We have proposed three LMI–SDP based methods addressing physical feasibility of base parameters. The BPFT method tests if a given parameter estimation is physically feasible or not. The BPFC method computes the closest physically feasible solution for a given (infeasible) solution. Finally, the FBPE-OLS method performs optimal estimation of base parameters (for a given data set) through least squares regression constrained to feasible solutions. These methods benefit from SDP advantages, noticeably the efficient convergence to the global optimum. LMI–SDP is well suited to deal with the feasibility problem, since it is a convex optimization technique which exploits positive semidefinite constraints. The FBPE-OLS method does not require a feasibility cost function definition and deals with base parameter space issues in a straightforward way. The FBPE-OLS is pertinent when model approximation and measurement errors entail infeasible solutions by classical techniques. All three methods address the issue of physically consistent estimations for a given data set and for a given robot dynamic model. Obviously, better robot models and better data sets entail better solutions. The proposed methods have been assessed using 3-link and 7-link robot arms. 7-link robot feasible estimations have been successfully achieved within the LMI-SDP framework, showing good performance for both identification and validation trajectories. These methods have been straightforwardly implemented with open source software, being computationally efficient.

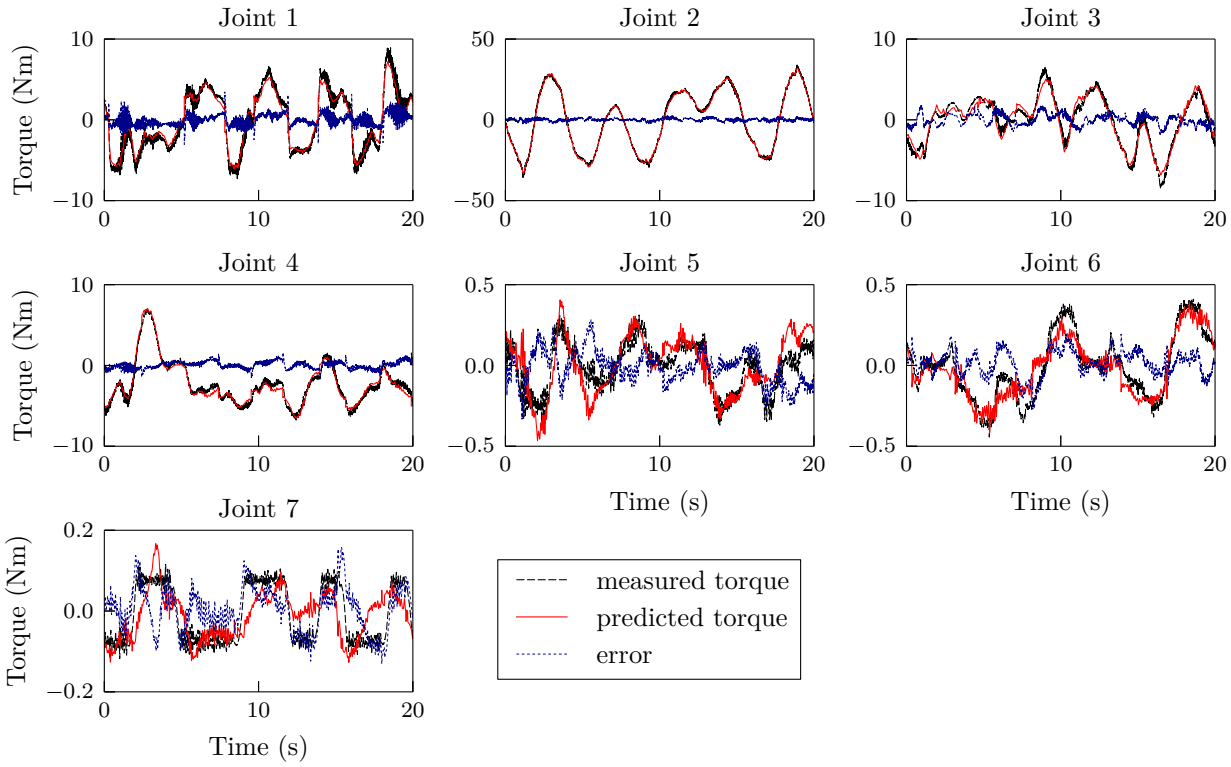

Figure 6: Validation trajectory A for the 7-link WAM Arm. Measured torque vs. predicted torque of the $\boldsymbol{\beta}^{\star \mathrm{e}}$ solution.

Funding

This work was supported in part by the Portuguese Science and Technology Foundation (FCT) [project number PTDC/EEA-CRO/110008/2009; grant number SFRH/BD/61553/2009].

References

Ayusawa, K. and Nakamura, Y. (2010). Identification of standard inertial parameters for large-DOF robots considering physical consistency. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 6194–6201. IEEE.

Benson, S. J. and Ye, Y. (2008). DSDP5-Software for Semidefinite Programming. *ACM Transactions on Mathematical Software*, 34(3):1–20.

Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia.

Collins, G. E. (1975). Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In H Brakhage, editor, *Proceedings Second GI Conference on Automata Theory and Formal Languages*, volume 33, pages 134–183. Springer-Verlag.

Díaz-Rodríguez, M., Mata, V., Valera, A., and Page, A. (2010). A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. *Mechanism and Machine Theory*, 45(9):1337–1356.

Featherstone, R. (2010). A beginner's guide to 6-D vectors (Part 1). *IEEE Robot. Automat. Mag.*, 17(3):83–94.

Gautier, M. (1991). Numerical calculation of the base inertial parameters of robots. *Journal of Robotic Systems*, 8(4):485–506.

Gautier, M. (1997). Dynamic identification of robots with power model. In *Proc. IEEE Int. Conf. Robot. Autom.*, volume 3, pages 1922–1927. IEEE.

Gautier, M., Briot, S., and Venture, G. (2013). Identification of consistent standard dynamic parameters of industrial robots. *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1429–1435.

Gautier, M. and Khalil, W. (1990). Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Trans. Robot. Autom.*, 6(3):368–373.

Gautier, M. and Khalil, W. (1992). Exciting Trajectories for the Identification of Base Inertial Parameters of Robots. *Int. J. Robotics Research*, 11(4):362–375.

Goldman, A. J. and Ramana, M. (1995). Some geometric results in semidefinite programming. *Journal of Global Optimization*, 7(1):33–50.

Hollerbach, J. M., Khalil, W., and Gautier, M. (2008). Model Identification. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*, chapter 14, pages 321–344. Springer.

Khalil, W. and Dombre, E. (2004). *Modeling, identification & control of robots*. Butterworth-Heinemann.

Mata, V., Benimeli, F., Farhat, N., and Valera, A. (2005). Dynamic parameter identification in industrial robots considering physical feasibility. *Journal of Advanced Robotics*, 19(1):101–120.

Mayeda, H., Yoshida, K., and Osuka, K. (1990). Base parameters of manipulator dynamic models. *IEEE Trans. Robot. Autom.*, 6(3):312–321.

Nesterov, Y. and Nemirovskii, A. (1987). *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM.

Park, F., Bobrow, J., and Ploen, S. (1995). A Lie Group Formulation of Robot Dynamics. *Int. J. Robotics Research*, 14(6):609–618.

Sousa, C. D. and Cortesão, R. (2012). SageRobotics: Open Source Framework for Symbolic Computation of Robot Models. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12*, pages 262–267. ACM Press.

Swevers, J., Ganseman, C., Tukel, D., de Schutter, J., and Van Brussel, H. (1997). Optimal robot excitation and identification. *IEEE Trans. Robot. Autom.*, 13(5):730–740.

Ting, J.-A., D'Souza, A., and Schaal, S. (2011). Bayesian robot system identification with input and output noise. *Neural Networks*, 24(1):99–108.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

Wu, J., Wang, J., and You, Z. (2010). An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5):414–419.

Yoshida, K. and Khalil, W. (2000). Verification of the Positive Definiteness of the Inertial Matrix of Manipulators Using Base Inertial Parameters. *Int. J. Robotics Research*, 19(5):498–510.