Contribution to the parametric identification of dynamic models : application to collaborative robotics

Fabio Ardiani

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

ONERA
THE FRENCH AEROSPACE LAB

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

délivré par
l'Institut Supérieur de l'Aéronautique et de l'Espace

A être présenté et soutenu par

Fabio ARDIANI

le 10/01/2023

Contribution to the Parametric Identification of Dynamic Models : Application to Collaborative Robotics.

École doctorale et discipline ou spécialité :
EDSYS (École Doctorale Systèmes) : Robotique et Automatique

Unité de recherche :
Office National d'Etudes et de Recherches Aérospatiales (ONERA)
Laboratoire Génie de Production de l'École Nationale d'Ingénieurs de Tarbes (LGP-ENIT)

Thèse dirigée par
M. Alexandre JANOT et M. Mourad BENOUSSAAD

Membres du jury :

M. Yannick AOUSTIN	Professeur - Université de Nantes	Rapporteur
M. Andrea CHERUBINI	Professeur - Université de Montpellier	Rapporteur
M. Mehdi CHERIF	Professeur - ENSAM Bordeaux	Examinateur
Mme Marina INDRI	Associate Professor - Politecnico di Torino	Examinatrice
M. Hugues GARNIER	Professeur - Polytech Nancy	Examinateur
M. Mourad BENOUSSAAD	Maître de conférences - ENI de Tarbes	Co-directeur
M. Alexandre JANOT	Ingénieur de recherches - ONERA	Directeur

Contents

Table of Contents i

List of Figures vii

List of Tables xi

Nomenclature xiii

General introduction 1

1 Robotic manipulators and identification procedure 5

1.1 Introduction 5

1.2 Modeling basics of serial manipulators 6

1.2.1 Rigid robot model 6

1.2.2 Friction model 11

1.2.3 Model of motors and gearboxes 14

1.2.4 Flexible joint model 14

1.2.5 Load and external force model 15

1.3 Identification procedure 16

1.3.1 Model selection 17

1.3.2 Model reduction 18

1.3.3 Trajectory selection 23

1.3.4 Robot programming and measurements 26

1.3.5 Data processing 28

1.3.6 Model validation 29

1.4 Parameter identification methods 31

1.4.1 Terminology and classification 32
1.4.2 The IDIM-LS method 36
1.4.3 The IDIM-RLS method 37
1.4.4 The PC-IDIM-LS method 37
1.5 Conclusion 38

2 Offline en-bloc identification **39**
2.1 Introduction 39
2.2 Problem statement and related works 40
2.3 The new PC-IDIM-IV method 43
2.3.1 The IDIM-IV method 44
2.3.2 An appropriate IV criterion for the PC problem 45
2.3.3 Resolution 47
2.3.4 Statistical analysis 48
2.3.5 Algorithm 49
2.3.6 Experimental results 49
2.3.7 Conclusion 61
2.4 Friction identification 62
2.4.1 The IDIM-SLS method 63
2.4.2 Experimental comparison 64
2.4.3 Conclusion 68
2.5 Final conclusion 69

3 Offline recursive identification **71**
3.1 Introduction 71
3.2 Problem statement and related works 72
3.3 The IDIM-RIV method 75

3.3.1 Formulation . . . 75
3.3.2 Construction of Z . . . 76
3.3.3 Justification of the construction of Z . . . 78
3.3.4 Algorithm . . . 78
3.3.5 Miscellaneous remarks . . . 79
3.3.6 Experimental results . . . 79
3.3.7 Conclusion . . . 87
3.4 Line-by-line coupled identification . . . 89
3.4.1 Preliminaries . . . 90
3.4.2 The IDIM-CLS method . . . 90
3.4.3 The S-IDIM-RLS method . . . 92
3.4.4 Experimental results . . . 93
3.4.5 Conclusion . . . 94
3.5 Final conclusion . . . 95

4 Online identification **97**

4.1 Introduction . . . 97
4.2 Problem statement and related works . . . 98
4.3 The IDIM-CIV method . . . 102
4.3.1 Algorithm . . . 102
4.4 Adding the forgetting factor . . . 103
4.5 Experimental framework . . . 105
4.6 Case study I: simple trajectory . . . 106
4.7 Case study II: movement with constant payload . . . 109
4.8 Case study III: movement with changing payload . . . 112
4.9 Case study IV: physical Human Robot Interaction . . . 116
4.10 Final conclusion . . . 118

General conclusion & perspectives **121**

List of publications **125**

Appendix A Least-squares estimator **127**

A.1 Derivation 127

A.2 BLUE estimator 128

Appendix B Collaborative robot platform **131**

B.1 The components 131

B.1.1 The robot 131

B.1.2 The load 133

B.1.3 The Force-Torque sensor 134

B.1.4 The computer 134

B.2 The framework 135

B.3 Experimental preliminaries 136

B.3.1 Model selection and reduction 136

B.3.2 Exciting trajectories choice 137

B.3.3 Data processing 137

B.3.4 Analysis of signals 139

Appendix C MESSII dataset **145**

C.1 Introduction 145

C.2 Related work 147

C.3 Robotic platform 147

C.4 The dataset 148

C.4.1 Structure 148

C.4.2 Data collection 149

C.4.3 Data processing 151

C.4.4 Calibration 152

C.5 Applications and challenges 153

C.5.1 Parameter identification 153

C.5.2 Human interaction 154

C.6 Conclusion and future work 157

Appendix D PC-IDIM-IV on TX40 manipulator 159

D.1 Introduction 159

D.2 Robotic manipulator, exciting trajectories and data acquisition 160

D.3 Experimental results 161

D.3.1 First scenario: actual data 161

D.3.2 Second scenario: downgraded data 163

D.3.3 Third scenario: modeling errors 164

D.4 Conclusion 164

Appendix E Model and identification of an unbalanced two-wheeled differential drive mobile robot 167

E.1 Introduction 167

E.2 Unbalanced differential drive mobile robot model 168

E.2.1 Kinematic equations of motion 170

E.2.2 Dynamic model 171

E.3 Parameter identification 174

E.3.1 Methods 174

E.3.2 Trajectory selection 175

E.4 Simulation and results 176

E.5 Conclusion 179

Appendix F Identification of multiple KUKA iiwa manipulators **181**

F.1 Signal analysis 182

F.2 Estimated parameters 183

F.3 Torque reconstruction 183

F.4 Conclusion 188

Bibliography **189**

List of Figures

1.1 Serial manipulator. 6

1.2 Summary of mathematical models of a manipulator. 7

1.3 Modified Denavit-Hartenberg (MDH) notation. 8

1.4 Friction models. 13

1.5 Modeling of joint flexibility. 15

1.6 Iterative process of parameter identification. 17

1.7 General closed-loop block scheme of a robotic system. 28

1.8 Structure of different identification methods. 35

2.1 PC-IDIM-IV algorithm. 50

2.2 Real and reconstructed torque using the model in Table 2.1. 53

2.3 Torque reconstruction with state-of-the-art models and the one derived in Table 2.3. 57

2.4 Two friction identification procedures. 63

2.5 Friction identification trajectory. 64

2.6 External torque reconstruction of the joints' friction models obtained for *Model B* during the training trajectories. 67

3.1 Evolution of ZZR_2, XXR_2, MYR_2, ZZR_4, XXR_4, MYR_4, ZZR_6 and XXR_6 estimates in *Situation 1*. 83

3.2 Comparison between: sensed torque (blue) and reconstructed torque (orange) with estimates from Table 3.1 using Traj. 1 in *Situation 1*. 84

3.3 Evolution of some base parameters of joint 4 using IDIM-RLS with and without a priori information. 86

3.4 Evolution of some base parameters of joint 4 using IDIM-RLS and IDIM-RIV methods. 88

3.5 Schematic diagram of the CLS algorithm. 91

3.6 Schematic diagram of the S-IDIM-RLS algorithm. 93

4.1 Framework for online estimation of paramters of the KUKA iiwa manipulator 105

4.2 Evolution of some base parameters of joint 4 using the online IDIM-CIV without payload. 108

4.3 Evolution of M_L and MZ_L in online identification of a constant fixed payload. 113

4.4 a) Wooden box and b) extra payloads. 114

4.5 Steps of the trajectory for online identification of changing payload. 115

4.6 Evolution of varying payload mass during online identification with different forgetting factors strategies and speed of joint 4. 117

4.7 Forces applied by the human on the end effector. 119

4.8 Sensed and reconstructed torques during human interaction. 120

B.1 KUKA LBR iiwa 14 R820. 132

B.2 Industrial collaborative manipulator's structure. 133

B.3 Steel cylinder fixed to the tip of the robot used as payload. 134

B.4 Force-Torque sensor fixed to the tip of the robot. 135

B.5 Communication framework of the experimental setup of the KUKA iiwa manipulator. 136

B.6 MDH parameters and frames of the KUKA LBR iiwa 14 R820. 138

B.7 Example of movement of end-effector during global trajectory. 139

B.8 Example of position signals and their respective error. 140

B.9 Example of velocity signals and their respective error. 141

B.10 Example of acceleration signals and their respective error. 142

B.11 Example of torque signals and their respective error. 143

C.1 IIWA robotic manipulator, payload and force-torque sensor used to build the dataset. 148

C.2 Structure of the *MESSII Dataset*. 149

C.3 Sub-folders description of the *MESSII dataset*. 150

C.4 Position, speed and acceleration signals of joint 4 used for identification. . . . 155

C.5 Comparison between actual and reconstructed torque of the trajectory in Figure C.4 with the identified essential parameters of Table C.1. 155

C.6 Comparison between reconstructed forces and the real ones obtained with the F/T sensor. 156

D.1 TX40 manipulator. 160

D.2 Cross-validation, second joint and first trajectory, actual data. 162

D.3 Cross-validation, second joint and first trajectory, downgraded data. 166

E.1 Differential drive mobile robot. 170

E.2 Inputs of Fourier-based trajectory. 177

E.3 Position followed by the robot when applying inputs of Figure E.2. 177

E.4 Inputs obtained as a combination of *sins*. 177

E.5 Inputs obtained as a combination of *steps*. 177

F.1 Three KUKA iiwa manipulators. 182

F.2 *Commanded position* for the three KUKA iiwa manipulators. 184

F.3 *Sensed torque* for the three KUKA iiwa manipulators. 185

F.4 Some of the identified parameters and its standard deviation for the three KUKA iiwa manipulators over four trajectories. 186

List of Tables

2.1 Set of essential parameters of the manipulator's model integrated in the controller. 52

2.2 Torque reconstruction error with model of Table 2.1. 53

2.3 Set of essential parameters using the IDIM-LS, IDIM-IV and NF-IDIM-IV methods. 56

2.4 Torque reconstruction error with state-of-the-art models and the one from Table 2.3. 58

2.5 Set of standard parameters identified with the PC-IDIM-IV method. 60

2.6 Set of base parameters identified with the PC-IDIM-IV method equivalent to Table 2.5. 60

2.7 Torque reconstruction error of models identified with the PC-IDIM-LS, PC-IDIM-IV, NF-PC-IDIM-LS and NF-PC-IDIM-IV methods. 61

2.8 Identified essential inertial parameters and their respective relative standard deviation. 65

2.9 Identified essential friction parameters and their respective relative standard deviation. 66

2.10 Percent error [%] of the sensed torque reconstruction of global trajectories for Model A and Model B. 66

3.1 IDIM-RIV set of base parameters estimates averaged over the last 50 measurements and their respective relative standard deviation. 82

3.2 Percent error [%] of torque reconstruction using the IDIM-RIV estimates of Table 3.1. 85

3.3 Percent error [%] of torque reconstruction of models identified with the IDIM-RLS, S-IDIM-RLS, IDIM-CLS and the method in [De Wit and Aubin, 1990]. 94

4.1 Features of the evolution of some parameters of joint 4 with different initial values. 107

B.1 Summary of transformation between standard and base parameters for the KUKA LBR iiwa manipulator. 137

B.2 Regrouped base parameters of the KUKA LBR iiwa following the MDH convention stated in Figure B.6. 138

C.1 Identified essential inertial parameters and their respective relative standard deviation of joint 4. 154

D.1 Relative errors obtained with direct comparisons for the PC-IDIM-LS, the PC-IDIM-IV, IDIMIV and PC-DIDIM methods - Actual data 162

D.2 Relative errors obtained with cross-validation for the PC-IDIM-LS, the PC-IDIM-IV, IDIMIV and PC-DIDIM estimates - Actual data 162

D.3 Relative errors obtained with direct comparisons for the PC-IDIM-LS, the PC-IDIM-IV, IDIM-IV and PC-DIDIM methods - Downgraded data 163

D.4 Relative errors obtained with cross-validation, the PC-IDIM-LS, the PC-IDIM-IV, IDIM-IV and PC-DIDIM estimates - Downgraded data 163

D.5 Relative errors obtained for the PC-IDIM-IV, IDIMIV and PC-DIDIM methods - Modeling errors 164

D.6 Relative errors obtained with cross-validation, the PC-IDIM-IV, IDIMIV and PC-DIDIM estimates - Modeling errors 164

E.1 Symbols. 169

E.2 Fourier-based trajectory parameters. 176

E.3 Estimates' values. 178

E.4 Cross-validation: fitness value (one's complement of NRMSE) between real and estimated trajectories [in %]. 178

E.5 Cross-validation with external perturbation: fitness value (complement of NRMSE) between real and estimated trajectories [in %]. 179

F.1 Percent error [%] in torque reconstruction for the three KUKA iiwa manipulators. 187

Nomenclature

Acronyms

BLUE	Best Linear Unbiased Estimate.
CAD	Computer Aided Design
CLIE	Closed-Loop Input-Error.
CLOE	Closed-Loop Output-Error.
DDIM	Direct Dynamic Identification Model.
DDM	Direct Dynamic Model.
DIDIM	Direct and Inverse Dynamic Identification Models.
dof	Degrees of freedom.
FRI	*Fast Robot Interface*.
IDIM	Inverse Dynamic Identification Model.
IDM	Inverse Dynamic Model.
IRLS	Iteratively Re-Weighted Least-Squares.
IV	Instrumental Variable.
KKT	Karush – Kuhn – Tucker.
LE	Lagrange-Euler.
LMI	Linear Matrix Inequality.
LC	Least-Squares.
LS	Least-Squares.
MDH	Modified Denavit-Hartenberg.
MESSII	Manipulator Experimental SyStem Identification and Interaction.
ML	Maximum-Likelihood.
NF	No-Filter.
NLP	Nonlinear Programming.
OE	Output-Error.

PC	Physically-Consistent.
pHRI	Physical Human Robot Interaction.
PID	Proportional, integrative, derivative
PTP	Point-to-point.
RLS	Recursive-Least-Squares.
RMS	Root Mean Square.
ROS	Robot Operating System.
SDP	Semi-Definite Programming.
SLS	Separable Least-Squares.
SPL	Spline.
SQP	Sequential Quadratic Programming.
SRIV	Simple Refined Instrumental Variable.
SVD	Singular Value Decomposition.
TLS	Total Least-Squares.
WLS	Weighted Least-Squares.

Operators

C.	cos(.)
CC.	$cos(.) \times cos(.)$
cond(.)	Condition number.
$\mathbb{E}(.)$	Expectation operator.
$. \succ 0$	Positive definiteness.
$sgn(.)$	Sign function.
S.	sin(.)
SS.	$sin(.) \times sin(.)$
$tr(.)$	Trace.
$Var(.)$	Variance operator.

Symbols

$\boldsymbol{\beta_{st}^{j}}$	Vector of standard parameters of link j.

$\boldsymbol{\beta}$	Vector of base parameters.
$\hat{\boldsymbol{\beta}}$	Vector of estimates of the base parameters.
$\mathbf{e}$	Vector of errors.
f_c	Coulomb friction coefficient.
$\mathbf{f_e}$	Wrench that the manipulator exerts on the environment.
f_{nlv}, α_f	Nonlinear viscous friction coefficients.
f_{sc}, δ	Friction coefficients related to the Stribeck effect.
f_v	Viscous friction coefficient.
$\mathbf{G}, \mathbf{G}(\mathbf{q})$	Gravity torques vector.
$\mathbf{H}, \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$	Coriolis and centrifugal effect matrix.
I_{a_j}	Equivalent inertia of the motor on the link side of link j.
$\mathbb{I}_i$	Identity matrix of order i.
$\mathbf{I}^j$	Inertia tensor of link j considered from the origin of the link.
$\mathbf{IDM_{st}}$	IDM with respect to the standard parameters.
$\mathbf{IDM}$	IDM with respect to base parameters.
$\mathbf{J}, \mathbf{J}(\mathbf{q})$	Jacobian matrix.
$\mathbb{J}$	Pseudo Inertia Matrix.
K_v	Coefficient related to the approximation of the sign function.
$\mathbf{M}, \mathbf{M}(\mathbf{q})$	Inertia matrix.
m	Number of measurements.
M_j	Mass of link j.
$\alpha_j, d_j, \theta_j, r_j$	MDH parameters of link j.
$\mathbf{MS^j}$	Vector of the first moment of inertia of link j about its origin.
MX_j, MY_j, MZ_j	Components of the first moment of inertia of link j $\mathbf{MS^j}$.
$\mathbf{N}, \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$	Vector of Coriolis, centrifugal and gravitational torques.
n	Number of dof of the serial manipulator.
n_b	Number of base parameters of a model.
n_e	Number of essential parameters of a model.

n_{st}	Number of standard parameters of a model.
$\boldsymbol{O_j}, \boldsymbol{x_j}, \boldsymbol{y_j}, \boldsymbol{z_j}$	Origin and coordinate system of frame j.
$\mathbf{0}_{i\times j}$	Matrix of size $i \times j$ full of zeros.
$\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}}$	$n \times 1$ vectors representing the filtered joint angles, velocities and accelerations of a robotic manipulator of n dof, respectively.
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	$n \times 1$ vectors representing the joint angles, velocities and accelerations of a robotic manipulator of n dof, respectively.
r	Defined as amount of measurements m and amount of joints n.
$\hat{\rho}$	Vector of residuals.
ρ	Vector of sampled errors.
$\%\sigma_{\hat{\beta}_j}$	Relative standard deviation of estimate of parameter j.
$\hat{\boldsymbol{\tau}}$	Vector of size $n \times 1$ of filtered joint torques.
$\boldsymbol{\tau}$	Vector of size $n \times 1$ of joint torques.
$\boldsymbol{\tau_f}$	Vector of $n \times 1$ friction torques.
τ_{off}	Offset torque.
$\mathbf{W}$	Sampled IDM with respect to base parameters or IDIM.
$\mathbf{X}$	Position and orientation of the tip of the robot.
XX_j, YY_j, ZZ_j	Diagonal elements of the inertia tensor $\boldsymbol{I}^j$.
XY_j, XZ_j, YZ_j	Products of inertia of the inertia tensor $\boldsymbol{I}^j$.
$\hat{\boldsymbol{y}}$	Vector of estimated torques.
$\boldsymbol{y}$	Vector of sampled torques.
$\mathbf{Z}$	Instrumental Matrix.

General introduction

Robotics has played an important role on industrial changes over the last decades, allowing to reduce costs, improve quality, increase productivity, alleviate strenuous tasks and complete missions that are risky for a human to undertake. Even though robots were first designed to work in industrial scenarios inside cages, there has been a tendency over the last years to delete these barriers and allow the robots not only to share their workspace with humans, but also their tasks and objectives [Ajoudani et al., 2018b]. This has widened their application area whose result is an increasingly frequent appearance of robots in our daily lives and a necessity to master the robot in a better way [Siciliano et al., 2008]. Whether it is needed to design a control law, simulate the system to predict future states, or perform fault or collision detection, an accurate and reliable model is mandatory [Craig et al., 1987, Östring, 2002].

There are two kind of approaches to develop a mathematical description of the system of interest. The first one is a theoretical approach called *first-principles method*, which is based on the fundamental laws of physics that govern the process. The second one is an empirical approach based on the analysis of the observations of the inputs and outputs of the system (*data-driven*) called *identification*. On the one hand, the first approach has the advantage that the obtained model is directly relatable to the physics of the process, which is the reason why it is also called *white-box* modeling. This allows the model to be easily extrapolated to different working conditions. However, it requires a deep knowledge of the system, which becomes more difficult as the system gets more complex. They roughly exist in the real world because it is difficult to precisely model all phenomena that interact with a system. Noise, errors and uncertainties will usually be present. On the other hand, the data-driven approach is more suitable for complex systems as it does not require the full knowledge of it. This method is also called *black-box* modeling because the physical meaning of the model is not known. However, it has the drawback that the performance of the obtained model will usually depend on the conditions in which the data was obtained. Moreover, it lacks the physical interpretation of the system, which can be useful to validate, understand, scale and optimize the process of identifying a useful model. For these reasons, combining the strength of these two approaches, particularly, the knowledge of the physical laws and properties of the process and the power of experimental data, gives rise to a useful third approach called *grey-box* modeling. This method has the advantages of being less opaque with respect to the physics of the process than the black-box method, thus being more scalable, and of being applicable to more complex processes than white-box methods, as it does not require the complete knowledge of the system.

Gray-box modeling consists in determining the relations between inputs and outputs thanks to the physical knowledge of the system, and then, in identifying the parameters of this model through measurements and proper mathematical treatment. This technique has been widely applied in industrial robotics for years [Khalil and Dombre, 2002], in order to provide the information of the usually not disclosed CAD model, and avoid the disassembly of the robot to carry out mechanical experiences on each of the individual parts of the robot.

This is usually enough precise to be able to design a model-based controller, which would compensate for the possible uncertainties, noise and external effects. However, the increasingly complexity of the systems and its surroundings, and the need to redefine the model while the system is running, bring these techniques back into the spotlight of research. Being able to track the evolution of the parameters on real-time while the robot is carrying out its task allows to adapt the control to the possible not-minor variations of the model, understand the interactions of the manipulator with the environment, modify its behavior with respect to them, and be able to predict possible failures.

In the context of this thesis, the focus is made on the development and enhancement of parameter estimation methods to be applied in the identification of the dynamic model of robotic systems. The attention is centered on collaborative robotics: manipulators that work together with humans. The need to identify the model and interactions in real-time becomes an evident necessity, as the safety of humans could be compromised. Even though we validate the methods mostly on the KUKA iiwa collaborative manipulator, they are general and applicable to a wide variety of robots. The thesis is organized as follows:

- **Chapter 1:** it introduces the kinematic and dynamic modeling techniques of collaborative serial manipulators, the several steps of the process of identification, and the state-of-the-art parameter estimation methods based on the least-squares theory. It lays the theoretical groundwork for the following chapters.

- **Chapter 2:** we tackle two issues in offline en-bloc estimation methods. As real systems are complex being almost impossible to model all its interactions, as measurements are inherently noisy, and as models used in robotics usually have a physical meaning, the results of estimation may lead to parameters that are physically inconsistent. This means that although the estimated model may correspond to the training data, it is not likely to be in accord with the real parameters. The first proposed method tackles these issues as is robust against noise and ensures physical consistency of the estimates. Furthermore, the second proposed method addresses two other hindrances: the facts that manufacturers of commercial robots often hide important information and measurements to the users due to copyright and safety reasons; and the important but not yet fully understood friction phenomenon. The method identifies the model that the manufacturer has included in their controller in a reverse-engineering process, and uses it to estimate the friction parameters.

- **Chapter 3:** on the other hand, in systems that change during time or interact with an unknown dynamically changing environment, the estimation of the model and its parameters must be updated in an online basis. For this purpose, the recursive variants of the methods must be analyzed as they treat one measurement at a time, although in this chapter is still studied in an offline way. Aspects as algorithm initialization, stability of estimates and computing time gain importance. In this context, we first develop a new method that yields consistent estimates with noisy measurements, its robust against initial values and does not require an external simulation of the system. Second, with the objective of making recursive algorithms faster, we develop two new methods to carry

out a sequential identification which do not need matrices inversion. They present good features for online application, being less complex, and providing a tool to select which parameters to update depending on the information that measurements bring.

- **Chapter 4:** furthermore, some of these methods are tested in several online scenarios: the manipulator moving freely, with a fixed constant payload on the end-effector, with a changing payload on the end-effector and including human interaction. These scenarios allow to study the possible impact that these methods could have in real-time problems.
- **General conclusion & perspectives:** finally, a conclusion is given to summarize and highlight the contributions of this thesis. Advantages, disadvantages and limitations of the methods and scenarios are discussed. Moreover, this analysis leaves the place to consider the perspectives. From possible future works in order to improve and test the methods proposed, to the deliberation of open remarks that this work may leave to readers.
- **Appendices:** last but not least, this work presents several appendices which are based on original work and which are worth reading. They mainly show information that support the experimental tests that are discussed during the thesis, and results of the proposed methods in other systems, which are not collaborative robots, to show the wide application range of our developed methods.

Chapter 1

Robotic manipulators and identification procedure

Contents

1.1 Introduction . . . 5
1.2 Modeling basics of serial manipulators . . . 6
1.2.1 Rigid robot model . . . 6
1.2.2 Friction model . . . 11
1.2.3 Model of motors and gearboxes . . . 14
1.2.4 Flexible joint model . . . 14
1.2.5 Load and external force model . . . 15
1.3 Identification procedure . . . 16
1.3.1 Model selection . . . 17
1.3.2 Model reduction . . . 18
1.3.3 Trajectory selection . . . 23
1.3.4 Robot programming and measurements . . . 26
1.3.5 Data processing . . . 28
1.3.6 Model validation . . . 29
1.4 Parameter identification methods . . . 31
1.4.1 Terminology and classification . . . 32
1.4.2 The IDIM-LS method . . . 36
1.4.3 The IDIM-RLS method . . . 37
1.4.4 The PC-IDIM-LS method . . . 37
1.5 Conclusion . . . 38

1.1 Introduction

This chapter presents the main concepts of grey-box modeling in robotics which will lay the groundwork for the following chapters. In the first place, the introduction of the well-known formalisms used for deriving the equations that govern the kinematics and dynamics of collaborative robotic manipulators is done. In the second place, the procedure to identify

parameters is explained, detailing the main concepts and challenges of each of the steps. Finally, the basic notions and classification of parameter identification methods are given.

1.2 Modeling basics of serial manipulators

A serial manipulator is an open kinematic chain of bodies (i.e. *links*) interconnected by means of articulations (i.e. *joints*) extending from the base to the end-effector (see Figure 1.1). The movement is made by the joints, which can be either prismatic or revolute, and that determine its number n of degrees of freedom (*dof*). To express its behavior mathematically, several models relating different quantities are used (see an overview in Figure 1.2). For instance, the kinematic model describes the motion of the manipulator without taking into account the forces that cause the movement, whereas the dynamic model considers these forces. Extensive description of these models can be found in robotics textbooks [Khalil and Dombre, 2002, Craig, 2005, Siciliano et al., 2010]. In this section, a short overview of the main concepts will be presented. First, the so-called rigid robot model is presented, and then, some important additional phenomena that take place on real robots are introduced.

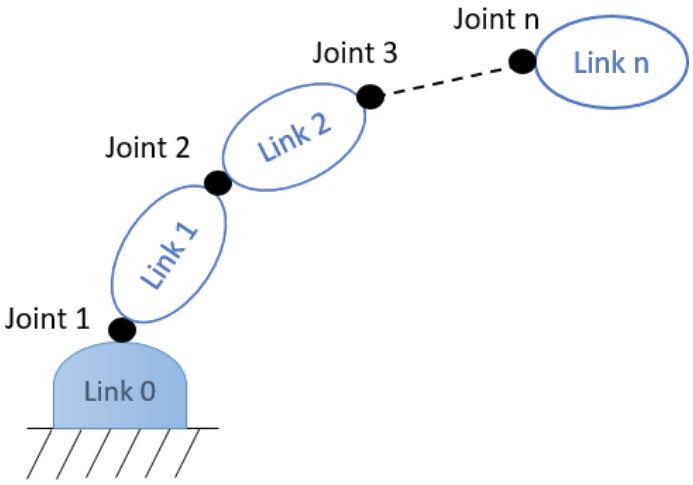


Figure 1.1: Serial manipulator.

1.2.1 Rigid robot model

In classical robotic manipulators, links are considered to be rigid bodies, meaning that their deformation during time is assumed to be negligible, and the manipulator's base is considered to be fixed. Thus, neither flexible link robots nor mobile robots will be addressed in this part. Additionally, for simplification purposes, we consider in this subsection that manipulators are actuated by motors which gearboxes are ideal with no backlash nor elasticity. However, these phenomena will be included in Section 1.2.4 deriving the flexible joint robot model. Meanwhile, in the current subsection, we will present the kinematic and dynamic models of serial rigid manipulators.

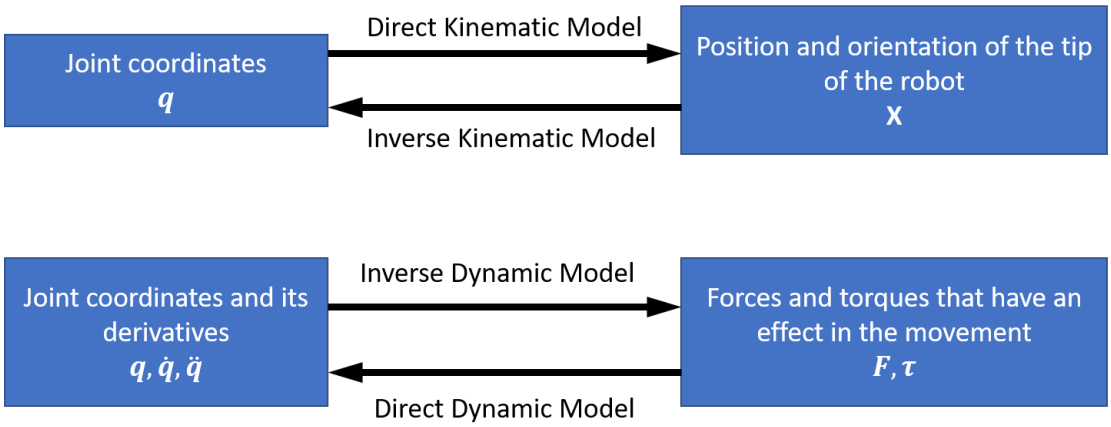


Figure 1.2: Summary of mathematical models of a manipulator.

1.2.1.1 Kinematic model

The direct kinematic model provides the location of the end-effector in terms of the n joint variables and the geometric parameters of the links. Although the solution to this problem is unique, there are many valid ways to express these relations. To standardize this systematic process, several methods and notations have been proposed, e.g. the Denavit-Hartenberg convention [Denavit and Hartenberg, 1955], the Sheth-Uicker method [Sheth and Uicker Jr, 1971] and the Modified Denavit-Hartenberg (MDH) convention [Khalil and Kleinfinger, 1986]. The later gives a unified description for all mechanical articulated systems, not just serial robots, with a minimum number of parameters and is the one that will be adopted all along this manuscript.

The first step of the MDH convention is numbering the $n+1$ links and n joints as shown in Figure 1.1, being link 0 the motionless base and link n the end-effector. Joint j connects link $j-1$ to link j and its variable is denoted by q_j. Next, the respective frames defined by the origin $\boldsymbol{O_j}$ and the coordinate systems defined by the unit vectors ($\boldsymbol{x_j}$,$\boldsymbol{y_j}$,$\boldsymbol{z_j}$) are located according to the following convention:

- the $\boldsymbol{z_j}$ axis is set along the axis of joint j,
- the $\boldsymbol{x_j}$ is aligned with the common normal between $\boldsymbol{z_j}$ and $\boldsymbol{z_{j+1}}$ and its orientation is arbitrarily selected,
- the intersection between $\boldsymbol{x_j}$ and $\boldsymbol{z_j}$ defines the origin $\boldsymbol{O_j}$ of the frame,
- $\boldsymbol{y_j}$ is chosen in the way that completes an orthonormal basis,
- if the first joint is revolute, $\boldsymbol{z_1}$ is chosen to be aligned with $\boldsymbol{z_0}$ and origins $\boldsymbol{O_0}$ and $\boldsymbol{O_1}$ to be coincident.

Once frames are located, homogeneous transformations are used to express the position and orientation of one link with respect to another. Homogeneous coordinates allow to state all

transformations applied to a vector by means of matrices multiplications, which is a desirable characteristic for simplicity and computational reasons. Following the MDH convention, the homogeneous transformation matrix ${}^{j-1}\boldsymbol{T_j}$ between two successive frames $j-1$ and j can be expressed as [Khalil and Dombre, 2002]:

$$ {}^{j-1}\boldsymbol{T_j} = \mathbf{Rot}(\boldsymbol{x_{j-1}}, \alpha_j)\mathbf{Trans}(\boldsymbol{x_{j-1}}, d_j)\mathbf{Rot}(\boldsymbol{z_j}, \theta_j)\mathbf{Trans}(\boldsymbol{z_j}, r_j), \tag{1.1} $$

where $\mathbf{Trans}(\boldsymbol{x_{j-1}}, d_j)$ and $\mathbf{Trans}(\boldsymbol{z_j}, r_j)$ are 4×4 matrices denoting a translation of d_j and r_j along the axis of the unit vectors $\boldsymbol{x_{j-1}}$ and $\boldsymbol{z_j}$, respectively; $\mathbf{Rot}(\boldsymbol{x_{j-1}}, \alpha_j)$ and $\mathbf{Rot}(\boldsymbol{z_j}, \theta_j)$ are 4×4 matrices denoting a rotation of α_j and θ_j about $\boldsymbol{x_{j-1}}$ and $\boldsymbol{z_j}$, respectively; and the four parameters α_j, d_j, θ_j and r_j are shown in Figure 1.3 [Khalil and Dombre, 2002] and described by:

- α_j is the angle between $\boldsymbol{z_{j-1}}$ and $\boldsymbol{z_j}$ about $\boldsymbol{x_{j-1}}$;
- d_j is the distance between $\boldsymbol{z_{j-1}}$ and $\boldsymbol{z_j}$ along $\boldsymbol{x_{j-1}}$;
- θ_j is the angle between $\boldsymbol{x_{j-1}}$ and $\boldsymbol{x_j}$ about $\boldsymbol{z_j}$ and the joint variable for revolute joints;
- r_j is the distance between $\boldsymbol{x_{j-1}}$ and $\boldsymbol{x_j}$ along $\boldsymbol{z_j}$ and the joint variable for prismatic joints.

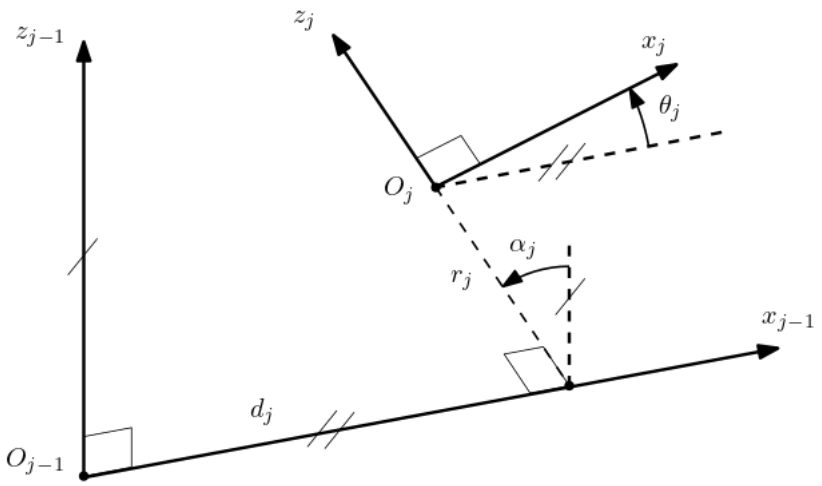


Figure 1.3: Modified Denavit-Hartenberg (MDH) notation.

Then, ${}^{j-1}\boldsymbol{T_j}$ can be expressed as:

$$ {}^{j-1}\boldsymbol{T_j} = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1.2} $$

where the notation $C. \triangleq cos(.)$ and $S. \triangleq sin(.)$ is adopted. By simple analysis, this expression

can be written in the form:

$$^{j-1}\boldsymbol{T_j} = \begin{bmatrix} ^{j-1}\boldsymbol{R_j}(\mathbf{q}) & ^{j-1}\boldsymbol{p_j}(\mathbf{q}) \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix}, \tag{1.3}$$

where the 3×3 matrix $^{j-1}\boldsymbol{R_j}(\mathbf{q})$ is the rotation matrix from frame $\boldsymbol{O_{j-1}}$ to frame $\boldsymbol{O_j}$, being $\mathbf{q} \in \mathbb{R}^n$ the vector of joint positions; $^{j-1}\boldsymbol{p_j}(\mathbf{q})$ is the 3×1 vector corresponding to the respective translation transformation from frame $\boldsymbol{O_{j-1}}$ to frame $\boldsymbol{O_j}$; and $\mathbf{0_{1\times 3}}$ is a vector of size 1×3 full of zeros.

Finally, the model that relates the base frame with the end-effector is built by multiplication of successive homogeneous transformations:

$$^{0}\boldsymbol{T_n} = {}^{0}\boldsymbol{T_1}{}^{1}\boldsymbol{T_2}...^{n-1}\boldsymbol{T_n}. \tag{1.4}$$

This relation corresponds to the direct kinematics of a manipulator, and allows to express the Cartesian coordinates of the end-effector (or any other link) with respect to the base (or any other link) by knowing the joint coordinates $\mathbf{q}$ and the kinematic parameters (structural dimensions).

Contrary, the inverse kinematic model consists of obtaining which are the joint variables that correspond to a given end-effector position and orientation. This problem is usually more complex than the direct kinematics, as it may be difficult to obtain a closed-form solution and there may exist multiple, infinite or not admissible solutions. Closed-form solutions can be obtained for relatively simple manipulators, using, for example, the Pieper method. However, if more complex geometries are considered, iterative numerical techniques are required and are usually applied [Pieper, 1969].

1.2.1.2 Differential kinematic model

Analogously to the kinematic model, the direct differential kinematic model gives the relation between the speed of the end-effector $\dot{\mathbf{X}}$ and the speed of the joints $\dot{\mathbf{q}}$. This is expressed by the Jacobian matrix $\mathbf{J}$ in the form:

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{1.5}$$

where $\mathbf{X}$ is the vector regrouping the position and orientation of the end-effector.

The Jacobian can be partitioned into 3×1 column vectors, $\mathbf{J}_{P_j}$ relating the contribution of the joint velocities $\dot{\mathbf{q}}$ to the end-effector linear velocity and $\mathbf{J}_{O_j}$ its contribution to the end-effector angular velocity, in the form:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{P_1} & ... & \mathbf{J}_{P_n} \\ \mathbf{J}_{O_1} & ... & \mathbf{J}_{O_n} \end{bmatrix}, \tag{1.6}$$

where, for revolute joints, $\mathbf{J}_{Pj}$ and $\mathbf{J}_{Oj}$ can be expressed as:

$$\begin{bmatrix} \mathbf{J}_{Pj} \\ \mathbf{J}_{Oj} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z_j} \times \boldsymbol{L_{j,n}} \\ \boldsymbol{z_j} \end{bmatrix}, \tag{1.7}$$

where $\boldsymbol{L_{j,n}}$ denotes the position vector connecting the frame $\boldsymbol{O_j}$ to the end-effector frame $\boldsymbol{O_n}$ ($\boldsymbol{L_{j,n}} = {}^{\mathbf{0}}\boldsymbol{p_n} - {}^{\mathbf{0}}\boldsymbol{p_j}$); and the symbol $\times$ denotes the vector product.

On the contrary, the inverse differential kinematic model gives the joint velocities $\dot{\mathbf{q}}$ for a desired end-effector velocity $\dot{\boldsymbol{X}}$. As well as for the inverse kinematic model, for complex or redundant manipulators, it is usually solved by numerical techniques where optimization methods are applied.

The Jacobian is an important concept in robotics and can be used for many purposes, e.g. to numerically compute the inverse kinematic model, to calculate the motor torques that are needed to exert specified forces in task space on the environment, to determine and predict singularities and to analyze the workspace.

1.2.1.3 Dynamic model

When we add the dynamic model, we firstly model the inverse dynamic model (IDM). The IDM relates the joint forces (in prismatic joints) and the joint torques (in revolute joints) with respect to the joint positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, accelerations $\ddot{\mathbf{q}}$ and the vector of forces and torques that the robot exerts on the environment $\mathbf{f_e}$:

$$\boldsymbol{\tau} = f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f_e}), \tag{1.8}$$

where $\boldsymbol{\tau}$ is the vector of joint torques and forces. This model is called the IDM because it defines the manipulator's inputs as a function of the outputs.

Alternatively, the direct dynamic model (DDM) relates the joint accelerations with respect to the joint positions, velocities, torques/forces and external forces:

$$\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{f_e}). \tag{1.9}$$

In this work, $\boldsymbol{\tau}$ will refer just to joint torques as only revolute joints are considered.

There are several ways to arrive to an expression of Equation (1.8), e.g. the Newton-Euler formulation and its recursive method [Orin et al., 1979, Luh et al., 1980, Craig, 2005], the Kane's method [Kane and Levinson, 1983], the Gibbs–Appell equations, and the Lagrange-Euler (LE) formulation [Uicker, 1965, Hollerbach, 1980, Khalil and Dombre, 2002]. Although all the proposed formulations arrive to a mathematical expression of the model which has the same output, they differ in their structure and computational efficiency.

In this work, we will make use of the LE formulation to derive the model. The method

describes the system in terms of energies in the form of:

$$\boldsymbol{\tau} = \frac{\mathrm{d}}{\mathrm{dt}} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}}, \tag{1.10}$$

where $L = E - U$ is called the Lagrangian; E is the kinetic energy and U is the potential energy of the manipulator.

After evaluating L in a serial manipulator, the general expression of the IDM can be described by n coupled nonlinear second order differential equations:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau_f} + \boldsymbol{\tau_e}, \tag{1.11}$$

where $\boldsymbol{\tau_e} \in \mathbb{R}^n$ is the vector of torques necessary to exert a given wrench on the environment $\mathbf{f_e}$; $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravitational torques defined as: $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$, where $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ stands for the gravity torques and $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is the vector of the Coriolis and centrifugal forces; $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric and positive definite inertia matrix; and $\boldsymbol{\tau_f} \in \mathbb{R}^n$ regroups the friction torques. The IDM is useful to select the actuators when designing a manipulator, to compute the actuator torques needed to achieve a desired motion and to identify its dynamic parameters.

Consequently to Equation (1.11), the DDM takes the form:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\tau_f} - \boldsymbol{\tau_e}). \tag{1.12}$$

The DDM is useful for simulation as it describes the motion of the system when a set of assigned joint torques is applied to the manipulator. However, the inertia matrix is configuration-dependent and its inversion is a time and resource consuming task, sometimes leading to instability of the method. Due to this, recursive methods are usually used to solve this problem [Khalil and Dombre, 2002].

1.2.2 Friction model

Friction is a complex phenomenon which plays a fundamental role in the modeling of robotic manipulators, and that has been studied for a long time in literature [Armstrong-Hélouvry et al., 1994, Bona and Indri, 2005, Van Geffen, 2009, Pennestrì et al., 2016]. Its modeling varies from a simple expression linear with respect to the joint velocity, to large and complex non-linear relations that depend on multiple physical parameters as the lubrication, the properties of the materials in contact or the temperature. Up to the moment, there is no available model that represents a general answer for all type of scenarios, and each of them has advantages and disadvantages with respect to each other in specific situations.

In robotics, the most used model is the one that includes the Coulomb friction and the viscous friction for each of the joints j. Its behavior is depicted in Figure 1.4(a) and mathe-

matically described by:

$$\tau_{f_j} = f_{v_j}\dot{q}_j + f_{c_j} sgn(\dot{q}_j), \tag{1.13}$$

where f_{v_j} is the viscous friction coefficient of joint j; f_{c_j} is its respective Coulomb coefficient; and $sgn(.)$ denotes the sign function.

However, this friction model is sometimes an over simplification of the process and does not depict reality adequately. For instance, the Coulomb friction coefficient can be different depending on the direction that the joint is turning. For that, another parameter τ_{off_j}, that regroups the asymmetrical Coulomb friction coefficient and other offsets, as those introduced by sensors and amplifiers, is included [Hamon et al., 2010]:

$$\tau_{f_j} = f_{v_j}\dot{q}_j + f_{c_j} sgn(\dot{q}_j) + \tau_{\text{off}_j}. \tag{1.14}$$

Figure 1.4(b) depicts the input/output behavior of this model.

Additionally, the viscous friction can also be considered asymmetrical, meaning that the constant of f_{v_j} will depend on the turning direction (see Figure 1.4(c)) [Canudas et al., 1987, Armstrong, 1988]. These models may be enough for joint speeds that are not around zero velocity. However, at low speeds, the Coulomb friction is not enough to explain the dry fiction phenomena. For that, its static counterpart called stiction, or breakaway friction, needs to be introduced. It is the static friction that needs to be overcome to enable relative motion of two stationary objects that are in contact, and it may be much bigger than the Coulomb friction (see Figure 1.4(d) for its input/output behavior) [Olsson et al., 1998]. In other words, the stiction blocks the movement of the system until the force applied in the direction of movement is higher than the force of stiction. In addition, the *Stribeck* effect may be of significant importance (see Figure 1.4(e)) [Canudas de Wit et al., 1991]. It corresponds to a negative slope at low speeds, and is due to the fact that the velocity dependence in the transition between static and viscous friction is continuous, and not discontinuous as in the previously described models (Figures 1.4(a-d)). These phenomena can be expressed in different ways, e.g. in [Afrough and Hanieh, 2019], the authors express the friction torques as a polynomial (see Figure 1.4(f) for a third order polynomial); and in [Olsson et al., 1998], the authors expressed it using an exponential function. Because most of these models are still discontinuous at zero velocity due to the *sgn* function, in [Indri et al., 2013, Indri and Trapani, 2020], the authors proposed an alternative expression:

$$\tau_{f_j} = f_{c_j} S_0 + f_{sc_j} \arctan\left(\dot{q}_j \delta_j\right) + f_{v_j}\dot{q}_j + \tau_{\text{off}_j} + f_{\text{nlv}_j} |\dot{q}_j|^{\alpha_{f_j}} S_0, \tag{1.15}$$

with

$$S_0 = \frac{2}{\pi} \arctan\left(\dot{q}_j K_v\right), \tag{1.16}$$

where f_{sc_j} is the coefficient related to the Stribeck effect; f_{nlv_j} is the viscous nonlinear friction coefficient; α_{f_j} is the power of the viscous non-linear term; and δ_i defines the shape of the Stribeck effect. S_0 provides a continuous approximation of the *sgn* function, where K_v is the compression factor which needs to be tuned in order to find the desired trade-off between the necessity of approximating the *sgn* function and avoiding abrupt changes around 0 velocity.

This is not the only way to approximate the sign function, since, for example, the authors in [Vantilt et al., 2015] approximated it with a hyperbolic tangent.

Last but not least, the friction may also depend on other factors besides velocity, as the temperature, the load and the position of the shaft [Bittencourt et al., 2010, Hamon et al., 2011, Carlson et al., 2015, Simoni et al., 2015, Gao et al., 2017, Simoni et al., 2019, Nevmerzhitskiy et al., 2019, Raviola et al., 2021, Tadese et al., 2021]. These may make the friction coefficients function of some variables instead of constants as shown in the previous equations.

All these expressions refer to, the so-called, static friction forces because they depend just on the joint speed. Another type of models are the dynamic ones, which add at least one more *dof* or state per joint to explain other phenomena as hysteresis, friction lag and varying breakaway force [Indri and Trapani, 2020]. For this, the Dahl model [Dahl, 1968], the LuGre model [De Wit et al., 1995] and the Leuven integrated friction model [Swevers et al., 2000] are known models addressing some of these behaviours.

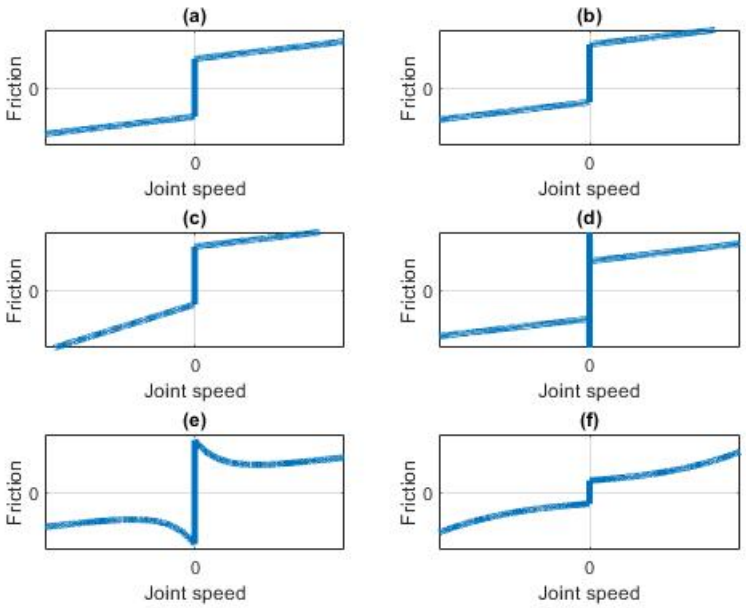


Figure 1.4: Friction models.

As the influence of friction on the manipulator is usually not negligible, obtaining a model that explains the real system with a good trade-off between precision and computational effort is a key topic in modeling of manipulators and it will be treated in more detail in following chapters.

1.2.3 Model of motors and gearboxes

The effect of the motors and gearboxes can be included in the inertia matrix $\mathbf{M}(\mathbf{q})$. I_{m_j} being the inertia of the rotor and transmission of the actuator of joint j, and N_j the respective transmission ratio, which is defined as the relation between the rotor velocity $\dot{q}_{m_j}$ and the link-side velocity $\dot{q}_j$, the equivalent inertia $I_{a_j} = N_j^2 I_{m_j}$ referred to the joint velocity is added to the element M_{jj} of the diagonal of the mass matrix $\mathbf{M}(\mathbf{q})$ in Equation (1.8). Moreover, the motor losses and friction can be regrouped with other friction terms shown previously.

This model neglects the gyroscopic effects of the rotors when the actuator is also moving with the link, which can be considered an acceptable simplification when there are high gear transmission ratios [Chedmail et al., 1986]. If this is not the case, more complex models are derived as in [Sciavicco et al., 1994].

1.2.4 Flexible joint model

Classical industrial robots are designed to work in structured known environments, to repeat the same tasks at high speeds, and to be robust, accurate and precise. However, if the manipulator is intended to work in unstructured and, a priori, unpredictable environments, where humans can be part of it, a different mechanical approach is adopted. For instance, the manipulator should be of low mass in comparison with the payload in order to increase the mobility, versatility and reduce the own inertia, the risk for humans and its energy consumption. To do so, lightweight structures, high-energy motors and high load to weight ratio gears are used. In addition, compliance is introduced by design in the gearbox, usually by means of harmonic drives, to mechanically decouple the motor from the link. These characteristics serve to reduce damages caused to the environment and the robot itself in the case of a collision. All these characteristics bring new challenges in terms of control and identification that have been widely addressed lately in literature [Hirzinger et al., 2001, Pham et al., 2001, De Luca et al., 2005, De Luca et al., 2006, Albu-Schäffer et al., 2007a, Albu-Schäffer et al., 2007b, Albu-Schaffer et al., 2008, Haddadin et al., 2009, Bischoff et al., 2010, Grebenstein et al., 2011].

The most well-known way to model the joint flexibility is presented in [Spong, 1987] and shown in Figure 1.5. It is an elastic lumped parameter model, where the rigid bodies are connected by torsional spring-damper pairs. Considering as if the friction acts just on the link side, the model in Equation (1.11) can be redefined as:

$$\begin{aligned} \boldsymbol{\tau}_{spring} &= \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_f + \boldsymbol{\tau}_e, \\ \boldsymbol{\tau}_{spring} &= \boldsymbol{K}_f(\mathbf{q_m} - \mathbf{q}) + \boldsymbol{D}_f(\dot{\mathbf{q}}_\mathbf{m} - \dot{\mathbf{q}}), \\ \boldsymbol{\tau} - \boldsymbol{\tau}_{spring} &= \boldsymbol{M}_m \ddot{\mathbf{q}}_\mathbf{m}, \end{aligned} \tag{1.17}$$

where, $\mathbf{q_m}, \dot{\mathbf{q}}_\mathbf{m}, \ddot{\mathbf{q}}_\mathbf{m} \in \mathbb{R}^n$ are the respective position, velocity and accelerations of the motor reflected through the gear ratios; $\boldsymbol{\tau}_{spring} \in \mathbb{R}^n$ is the vector of gearbox output torques; $\boldsymbol{K}_f \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D}_f \in \mathbb{R}^{n\times n}$ are the stiffness and damping diagonal matrices that describe

the flexibility, respectively; and $\boldsymbol{M_m} \in \mathbb{R}^{n \times n}$ is the diagonal matrix gathering inertias of motors I_{a_j}. It can be seen that the number of variables in these equations has been doubled, having the motor and link sides coordinates coupled by the flexibility.

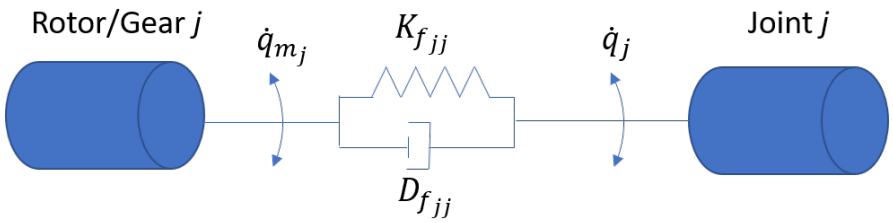


Figure 1.5: Modeling of joint flexibility.

Nevertheless, this model of flexibility may not represent the phenomenon in a satisfactory way, and more complex terms may be needed. For the derivation of Equations (1.17), the inertial coupling between the motors and the rigid links was neglected. This assumption is reasonable if the gear ratio is high [Spong, 1987]. However, if this coupling is not neglected, a more complex model is obtained where terms in the the kinetic energy of the rotor of motors due to the movement of the other links will appear besides the one of its relative rotation [Tomei, 1991]. Moreover, to model other phenomena, as the elasticity of bearings and bending and torsion of links, each rigid body can be considered to be connected with two torsional spring-damping pairs [De Luca, 2000]. Furthermore, nonlinear damping and hysteresis in the flexibility can also be included [Ruderman, 2012].

1.2.5 Load and external force model

Either because they need to push an object from one place to another, carry a load or tool, or work alongside with humans, all manipulators are intended to interact somehow with the environment. This influence can be modeled in two ways. First, if the manipulator is carrying a payload, it can be regarded as a structural modification of the last link. This means that, in the model, the last link would be redefined to include also the load. The second possibility is to consider the presence of the payload as if it was an external force applied to the tip of the robot. This method is applicable, not only to payloads, but to any surface that the tip of the robot is in contact with. The torque $\boldsymbol{\tau_e}$ (already included in Equation (1.11)) needed to exert a given wrench $\mathbf{f_e}$ on the environment is obtained by [Khalil and Dombre, 2002]:

$$\boldsymbol{\tau_e} = \boldsymbol{J}^T \mathbf{f_e}. \tag{1.18}$$

These situations will be covered in depth in Chapter 4.

1.3 Identification procedure

Either because a model-based controller is going to be implemented, simulation and prediction of future states are required, or diagnostics or fault and collision detection are planned, knowing the numerical value of the parameters of the model of the manipulator is crucial. In robotics, there are three approaches to determine the value of the parameters needed to complete the model in Equation (1.11) [Khalil and Dombre, 2002]:

- The first one is by means of physical experiments carried out to each link [Armstrong et al., 1986]. This process should be done by the manufacturer before assembling the manipulator, as, in most of the cases, disassembling the manipulator is not advisable because it can damage the robot if not done carefully, and its posterior assembly is not an easy task. Besides, the experiments needed to determine the parameters may be tedious and long, and may require of special and accurate measuring devices.

- The second approach is by means of computer aided design (CAD) techniques. In this case, by knowing the geometric and material characteristics of each link, their parameters can be estimated using computational tools without the need to disassemble the robot. However, the precision of the link model will determine the accuracy of the parameters. Indeed, the geometry of each link is complicated to define, and parts as bearing, bolts and wires should also be modeled, making the process also long and complicated. Besides, some parameters as those of the friction are not provided by manufacturers and are not possible to obtain via this technique.

- The third option is to apply identification techniques since both of the previous approaches often lead to inaccurate models. Provided the necessary sensors, this approach uses the analysis of the inputs and outputs of the manipulator to estimate parameters. The main advantages are that it can be directly applied to the manipulator without the need to disassemble it and that there is no need to know the precise construction features of each link.

Parameter identification is an iterative process composed by several steps with different tuning features which will ultimately depend on whether the model is going to be used for design, estimation, control, monitoring or other application, and on which is the expected performance. Figure 1.6 shows the corresponding steps for a typical parameter identification procedure (applied to a robotic system, although it can be extrapolated to other kind of systems), also called grey-box modeling. Extensive studies of system identification can be found in textbooks such as [Söderström and Stoica, 1989, Ljung, 1998, Tangirala, 2018], and overviews of parameter identification in robotics in [Wu et al., 2010, Leboutet et al., 2021]. This section will present the main concepts of the mentioned identification process for robotics systems, without making focus on the parameter identification step, which is the topic of Section 1.4.

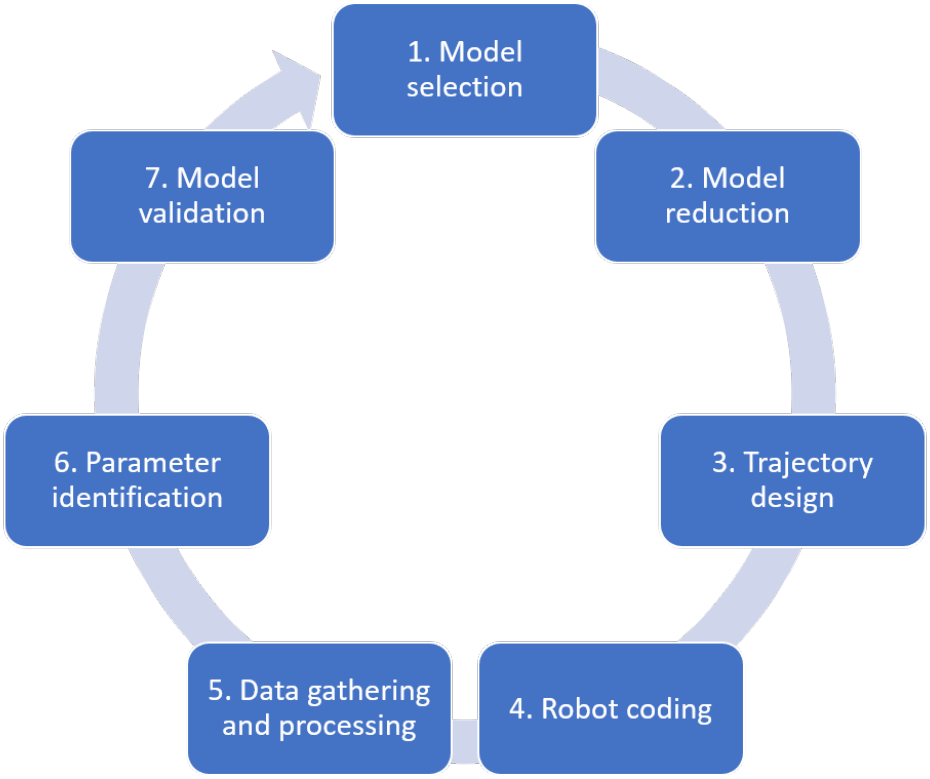


Figure 1.6: Iterative process of parameter identification.

1.3.1 Model selection

First of all, which and how a physical phenomenon is modeled is a user decision based on its knowledge of the system. For instance, besides the explicit dynamic model shown in Section 1.2, there are other formulations that allow to obtain a model with the same set of parameters to be identified, such as the energy model [Gautier and Khalil, 1988] and the power model [Gautier, 1997]. They may provide advantages as, for example, no need to calculate accelerations, which will reduce the impact of noise and the difficulty of the trajectory design, or reduce the computation time. However, in this work, we will focus on the IDM of Equation (1.11) as it is the model usually used for control purposes.

No matter which of the formulations is chosen, the dynamics of each link j can be described with 10, so-called, *standard parameters*, defined by its mass M_j; the 6 components (XX_j, XY_j, XZ_j, YY_j, YZ_j and ZZ_j) of the symmetric inertia tensor $\boldsymbol{I^j}$, considered from the origin of the link and defined by:

$$\boldsymbol{I^j} = \begin{bmatrix} XX_j & XY_j & XZ_j \\ XY_j & YY_j & YZ_j \\ XZ_j & YZ_j & ZZ_j \end{bmatrix}; \tag{1.19}$$

and, the 3 components (MX_j, MY_j and MZ_j) of its first moment of inertia $\mathbf{MS^j}$ about the origin of the link:

$$\mathbf{MS^j} = [MX_j \; MY_j \; MZ_j]^T. \tag{1.20}$$

Then, the standard parameters of joint j, without considering flexibility, friction nor the presence of motors, can be regrouped in the vector $\boldsymbol{\beta}_{st}^{j}$:

$$\boldsymbol{\beta}_{st}^{j} = [XX_j \; XY_j \; XZ_j \; YY_j \; YZ_j \; ZZ_j \; MX_j \; MY_j \; MZ_j \; M_j]^T. \tag{1.21}$$

Furthermore, the IDM of a serial manipulator is known to be linear with respect to this set of dynamic parameters [Siciliano et al., 2010]. Therefore, the model can be expressed in the form:

$$\boldsymbol{\tau} = \mathbf{IDM}_{st}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\beta}_{st}, \tag{1.22}$$

where $\boldsymbol{\beta}_{st} \in \mathbb{R}^{n_{st}}$ is the vector of the n_{st} standard parameters describing the manipulator and defined by:

$$\boldsymbol{\beta}_{st} = [{\boldsymbol{\beta}_{st}^{1}}^T {\boldsymbol{\beta}_{st}^{2}}^T ... {\boldsymbol{\beta}_{st}^{n}}^T]^T, \tag{1.23}$$

being $\boldsymbol{\beta}_{st}^{i}$ the vector of *standard parameters* of joint i; and $\mathbf{IDM}_{st}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the matrix of the IDM with respect to the standard parameters which will depend on the position, velocity and acceleration of the joints and on the geometric parameters of the links. In this study, geometric parameters are considered to be known and/or given by the manufacturer. If unknown or known but not precisely enough, an identification procedure can be applied called kinematic calibration [Mooring et al., 1991, Khalil et al., 2000, Santolaria and GinéS, 2013].

1.3.2 Model reduction

The standard parameters have a direct physical meaning related to each link. However, there are two other set of parameters that are important to introduce. The first one is called the set of *base parameters*, and is used to describe the complete dynamic model. They are deduced from the set of *standard parameters* by eliminating those that have no effect on the dynamic model and by grouping some others. They are an important concept in robotics, as they constitute the minimum set of inertial parameters that are needed to compute the dynamic model. Hence, they reduce the computational cost and they constitute the only identifiable parameters. The other set of parameters is called the *essential parameters*, and is obtained from the set of base parameters by eliminating some of them which are insignificant. It defines a simplified model which reduces computational burden and improves noise immunity of the estimation process, even though it is not theoretically complete [Pham and Gautier, 1991]. This section will describe an analytical and a numerical way to obtain the set of base parameters, and, afterwards, some criteria used to determine the set of essential parameters.

1.3.2.1 Base parameters

A. Analytical method. The work in [Gautier and Khalil, 1990, Khalil and Bennis, 1994] presents a direct method to determine the base parameters of serial robots by means of closed-form relations obtained from the energy model formulation. These rules are not unique, and the convention in [Khalil and Dombre, 2002] is used throughout this work. The algorithm

to determine all the parameters which can be eliminated or grouped for serial manipulators having just rotational joints is summed up as follows:

- *Grouping relations*: it is chosen to group YY_j, MZ_j and M_j with the other parameters. They can be grouped with the parameters of link j and $j-1$, yielding the following algorithm:

  for $j = n, ..., 1$:

$$\begin{aligned}
XXR_j &= XX_j - YY_j, \\
XXR_{j-1} &= XX_{j-1} + YY_j + 2\ r_j\ MZ_j + r_j^2\ M_j, \\
XYR_{j-1} &= XY_{j-1} + d_j\ S\alpha_j\ MZ_j + d_j\ r_j\ S\alpha_j\ M_j, \\
XZR_{j-1} &= XZ_{j-1} - d_j\ C\alpha_j\ MZ_j - d_j\ r_j\ C\alpha_j\ M_j, \\
YYR_{j-1} &= YY_{j-1} + CC\alpha_j\ YY_j + 2\ r_j\ CC\alpha_j\ MZ_j + (d_j^2 + r_j^2\ CC\alpha_j)M_j, \\
YZR_{j-1} &= YZ_{j-1} + CS\alpha_j\ YY_j + 2\ r_j\ CS\alpha_j\ MZ_j + r_j^2\ CS\alpha_j\ M_j, \\
ZZR_{j-1} &= ZZ_{j-1} + SS\alpha_j\ YY_j + 2\ r_j\ SS\alpha_j\ MZ_j + (d_j^2 + r_j^2\ SS\alpha_j)M_j, \\
MXR_{j-1} &= MX_{j-1} + d_j\ M_j, \\
MYR_{j-1} &= MY_{j-1} - S\alpha_j\ MZ_j - r_j\ S\alpha_j\ M_j, \\
MZR_{j-1} &= MZ_{j-1} + C\alpha_j\ MZ_j + r_j\ C\alpha_j\ M_j, \\
MR_{j-1} &= M_{j-1} + M_j,
\end{aligned} \tag{1.24}$$

  where $CC. \triangleq \cos(.) \times \cos(.)$, $SS. \triangleq \sin(.) \times \sin(.)$ and $CS. \triangleq \cos(.) \times \sin(.)$; and XXR_j, XYR_j, XZR_j, YYR_j, YZR_j, ZZR_j, MXR_j, MYR_j, MZR_j and MR_j are regrouped parameters. The parameters on the right side of the equations take their regrouped form if they were calculated on a previous iteration. Moreover, the three parameters YY_j, MZ_j and M_j, or their regrouped form, are then deleted from the model as their effect is now included in the regrouped parameters. In this way, there are $3 \times n$ parameters that have been reduced from the model by relating them to other dynamic parameters through geometric values.

- *Parameters having no effect on the dynamic model:* given the restricted motion of the first links of the manipulator, there are some parameters that have no effect on the dynamic model:
  - for the first revolute joint, and all those immediately subsequent joints whose axis are parallel to the first one, parameters XX_j, XY_j, XZ_j and YZ_j (or their regrouped form) are eliminated;
  - if the axis of the first joint is parallel to gravity, and for all other immediately subsequent joints which rotational axis is along the axis of the first one, the parameters MX_j and MY_j (or they regrouped form) are also eliminated.

Taking into account these rules, the number of minimum inertial parameters or *base parameters* for a manipulator with n rotational joints is $n_b \leq 7n - 4 - 2n_{1g}$, being n_{1g} equal

to 1 if the first joint is parallel to gravity, and 0 otherwise. If the second joint axis is not along the axis of the first joint, this equation gives the exact number of base parameters for serial manipulators with revolute joints. Then, the model in Equation (1.22) can be expressed as:

$$\boldsymbol{\tau} = \mathbf{IDM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\beta}, \tag{1.25}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n_b}$ is the vector of the n_b base parameters describing the manipulator; and $\mathbf{IDM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the matrix of the IDM with respect to the base parameters.

After m measurements, the problem of parameter identification is to find an estimate of $\boldsymbol{\beta}$ (denoted as $\hat{\boldsymbol{\beta}}$) of the over-determined linear system with respect to the parameters, called IDIM (Inverse Dynamic Identification Model), expressed as:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\beta} + \boldsymbol{\rho}, \tag{1.26}$$

where $\boldsymbol{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{r \times n_b}$ is the regressor or observation matrix obtained from evaluating $\mathbf{IDM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ m times being $r = n \times m$, and written as follows:

$$\boldsymbol{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \begin{bmatrix} \mathbf{IDM}(\mathbf{q}(1), \dot{\mathbf{q}}(1), \ddot{\mathbf{q}}(1)) \\ \mathbf{IDM}(\mathbf{q}(2), \dot{\mathbf{q}}(2), \ddot{\mathbf{q}}(2)) \\ \dots \\ \mathbf{IDM}(\mathbf{q}(m), \dot{\mathbf{q}}(m), \ddot{\mathbf{q}}(m)) \end{bmatrix}, \tag{1.27}$$

where $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$ are the vectors of joint positions, velocities and accelerations at time t, respectively; $\boldsymbol{y}(\boldsymbol{\tau}) \in \mathbb{R}^r$ is the vector of joint torques:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{\tau}(1) \\ \boldsymbol{\tau}(2) \\ \dots \\ \boldsymbol{\tau}(m) \end{bmatrix}; \tag{1.28}$$

and $\boldsymbol{\rho} \in \mathbb{R}^r$ is the residual error term built as:

$$\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{e}(1) \\ \boldsymbol{e}(2) \\ \dots \\ \boldsymbol{e}(m) \end{bmatrix}, \tag{1.29}$$

being $\boldsymbol{e}(i)$ the vector of errors at the measurements i that would appear in Equation (1.25).

B. Numerical method. Alternatively to the analytical method, from a linear algebra point of view, the problem of getting the set of base parameters from the standard parameters is a rank deficiency problem which can be solved by numerical methods [Gautier, 1991]. The two most used techniques are the QR and SVD decomposition. It is known that the first one is computationally cheaper than the second one and it will be explained in this section [Golub and Van Loan, 2013].

After m measurements, the IDIM model in Equation (1.22) can be expressed as:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{W_{st}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\beta_{st} + \boldsymbol{\rho}, \tag{1.30}$$

where $\boldsymbol{W_{st}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{r \times n_{st}}$ is the regressor matrix built with all the observations with respect to the standard parameters.

If $\boldsymbol{W_{st}}$ is rank deficient, then $n_b < n_{st}$, where n_b is the rank of $\boldsymbol{W_{st}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ and the number of base parameters. In this case, there is not a unique QR decomposition. However, a permutation matrix $\boldsymbol{P}$ (i.e. identity matrix with its columns permuted) can be defined such that $\boldsymbol{W_{st}P}$ has a unique decomposition:

$$\boldsymbol{W_{st}P} = [\boldsymbol{W_1 W_2}], \tag{1.31}$$

where, $\boldsymbol{W_1}$ is a set of n_b independent columns and $\boldsymbol{W_2}$ is a set of $n_{st} - n_b$ columns to be deleted.

By QR decomposition:

$$[\boldsymbol{W_1 W_2}] = \boldsymbol{QR} = [\boldsymbol{Q_1 Q_2}] \begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{R_2} \\ \mathbf{0}_{(r-n_{st})\times n_b} & \mathbf{0}_{(r-n_{st})\times(n_{st}-n_b)} \end{bmatrix}, \tag{1.32}$$

where $\boldsymbol{Q} \in \mathbb{R}^{r \times r}$ is an orthogonal matrix, what means that $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$; $\boldsymbol{R} \in \mathbb{R}^{r \times n_{st}}$ is an upper triangular matrix; $\boldsymbol{R_1} \in \mathbb{R}^{n_{st} \times n_b}$ is a full-rank upper-triangular matrix; $\boldsymbol{R_2} \in \mathbb{R}^{n_{st} \times (n_{st}-n_b)}$; $\boldsymbol{Q_1} \in \mathbb{R}^{r \times n_{st}}$ is the matrix related with $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$; and $\boldsymbol{Q_2} \in \mathbb{R}^{r \times (r-n_{st})}$ is the matrix related with $\mathbf{0}$.

From this equation it follows that $\boldsymbol{W_1} = \boldsymbol{Q_1 R_1}$ and $\boldsymbol{W_2} = \boldsymbol{Q_1 R_2}$. Combining these two expressions, we obtain that $\boldsymbol{W_2} = \boldsymbol{W_1 R_1}^{-1} \boldsymbol{R_2}$, which expresses the columns of $\boldsymbol{W_2}$ as linear combinations of the independent columns of $\boldsymbol{W_1}$.

Then, Equation (1.30) can be rewritten as:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{W_{st} P P}^T \beta_{st} + \boldsymbol{\rho} = [\boldsymbol{W_1 W_2}] \begin{bmatrix} \boldsymbol{\beta_1} \\ \boldsymbol{\beta_2} \end{bmatrix} + \boldsymbol{\rho} \\ &= \boldsymbol{Q_1 R_1 \beta_1} + \boldsymbol{Q_1 R_2 \beta_2} + \boldsymbol{\rho} \\ &= \boldsymbol{Q_1 R_1}(\boldsymbol{\beta_1} + \boldsymbol{R_1}^{-1} \boldsymbol{R_2 \beta_2}) + \boldsymbol{\rho} \\ &= \boldsymbol{W \beta} + \boldsymbol{\rho} \end{aligned} \tag{1.33}$$

where $\boldsymbol{\beta} = \boldsymbol{\beta_1} + \boldsymbol{R_1}^{-1} \boldsymbol{R_2 \beta_2}$ is the vector of n_b parameters that will be identified; and $\boldsymbol{W} = \boldsymbol{Q_1 R_1} \in \mathbb{R}^{r \times n_b}$ is the full column rank observation matrix associated with the base parameters. Note that $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ result of permuting β_{st} with $\boldsymbol{P}^T$. $\boldsymbol{P}$ is chosen in a way that, as in the analytical method, parameters of links with larger subscript, thus of joints that are nearer to the end-effector, are regrouped on those of the previous links (to follow the same convention that for the analytical method). It is called a numerical method because it needs of real measurements to be calculated and no analytical expression is derived.

Finally, it is important to notice, that choosing both the numerical or the analytical reduction method, will lead to a system with the same parameters to be identified in the form of Equation (1.26). The analytical method has the advantage of being insensitive to numerical problems and of having a direct relation with the physical meaning, whereas the numerical method presents the advantage of being a general solution, thus there is no need to change the algorithm for different manipulators.

1.3.2.2 Essential parameters

As some of these base parameters may only have a minor influence on the robot dynamics, they can be neglected in practice, creating the set of the so-called *essential parameters* [Pham and Gautier, 1991]. Its low influence can be due to two reasons: a bad trajectory selection leading to an insufficient excitation, which is the topic of the following section, or because the model is not enough sensitive with respect to these parameters, meaning that its contribution is negligible. Either of these reasons will lead to poor estimates with low practical identifiability [Ljung, 1998, Tangirala, 2018]. Identifiability is a widely used term in the identification community, and it refers to whether the model selection and the input has generated the information required to distinguish between two possible models and whether the estimation method is capable of estimating the true parameters if infinite samples are available, thus yielding consistent estimates.

One of the tools to determine if a parameter has a significant influence on the model or not, is the analysis of the statistical distribution of the estimate, or in other words, its variance. It is a measure of the precision of an estimator, i.e. the lower the variance, the more precise is the estimate. Let's consider that the estimates $\hat{\beta}_i$ follow a statistical distribution with standard deviation $\sigma_{\hat{\beta}_i}$ and relative standard deviation defined as:

$$\%\sigma_{\hat{\beta}_i} = \frac{100\ \sigma_{\hat{\beta}_i}}{|\hat{\beta}_i|}, \text{for} \hat{\beta}_i \neq 0. \tag{1.34}$$

The criterion to select the essential parameters is proposed by the user with the objective to obtain a simplified dynamic model with good precision and more robust against noise. For instance, the authors in [Pham and Gautier, 1991] proposed a step-wise regression method in which a selection of the essential parameters is done, and by analyzing the standard deviations of the estimates and the ability to predict the data of this new model, they determined if those parameters could be effectively removed from the model. Moreover, in [Khalil and Dombre, 2002] it is considered that if the relative standard deviation of a parameter is greater than ten times the minimum relative standard deviation value, the parameter can be considered poorly identified. If the same result is obtained with different trajectories, and if the value of the parameter is relatively small with respect to the others, then the authors cancelled this parameter. Furthermore, in [Gautier et al., 2013a], the authors calculated the essential parameters in an iterative procedure starting from the base parameters estimation. At each step the base parameter with the largest relative standard deviation is cancelled, and a new

estimation of the parameters is made and their respective new relative error standard deviation calculated. The procedure ends when the relation between the maximum and the minimum relative error standard deviation of the estimated parameters is lower than a chosen ratio proposed between 10 and 30. In [Janot et al., 2013b], the authors said that parameters with relative standard deviation greater than 20% or 30% can be cancelled.

A mathematically more elegant method, taken from econometrics [Davidson et al., 1993], has been presented in [Janot et al., 2013b] using the F-statistic. The F-statistic is known to be a tool for testing hypotheses about several parameters jointly, thus can be used to validate or invalidate model reduction hypotheses. We denote $\bar{\boldsymbol{\rho}}$ the error $\boldsymbol{\rho}$ of Equation (1.26) using the set of base parameters when it follows a multivariate normal distribution with zero mean, when samples are considered to be independent of each other and when it is homoskedastic, thus when it presents a constant covariance matrix $\sigma^2\mathbb{I}$. Then the F-statistic can be obtained by:

$$\hat{\boldsymbol{F}} = \frac{(||\bar{\boldsymbol{\rho}}_{\boldsymbol{e}}||^2 - ||\bar{\boldsymbol{\rho}}||^2)}{||\bar{\boldsymbol{\rho}}||^2} \frac{r - n_b}{n_b - n_e}, \tag{1.35}$$

where n_e refers to the amount of essential parameters being tested and $\bar{\boldsymbol{\rho}}_{\boldsymbol{e}}$ is the error when using the n_e essential parameters. It follows that if $\hat{\boldsymbol{F}}$ is less than $\boldsymbol{F}_{(1-\alpha),(n_b-n_e),(r-n_b)}$ (being the first term in parenthesis the critical value, and the following two the degrees of freedom) the model reduction is accepted, otherwise it is rejected. α refers to a threshold value used to judge whether a test statistic is statistically significant, usually being 0.01, 0.05 and 0.1 as acceptable values. The process is also iterative: first, the parameters with largest relative deviations are eliminated and it continues successively until the F-statistic fails.

Other ways to obtain the essential parameters are based on numerical methods as the QR or SVD decomposition [Pham and Gautier, 1991]. In the case of QR decomposition, the number of n_e essential parameters will be given by the rank of $\boldsymbol{R_1}$ in Equation (1.32). In order to build this matrix, a round-off approximation is considered in the elements R_{jj}. Hence, if they are lower than a threshold they are approximated to zero, and can be deleted from the model.

1.3.3 Trajectory selection

As identification uses real measurements to estimate parameters, it seems obvious that results will depend on the quality of the measured data. Therefore, one factor that has a direct influence on this quality, is the design of the experiment [Pukelsheim, 2006]. The amount of information in the data about a specific parameter depends almost entirely on the identifiability of the parameter and the selection of the trajectory applied to the manipulator. Thus, to improve the convergence rate and precision of the identification algorithms, a, so-called, persistently exciting trajectory needs to be designed [Ljung, 1998, Swevers et al., 2007]. This problem can be divided in three parts which will be introduced in this section: first, the selection of a cost-function to be able to compare the performance of two different trajectories; second, the selection on how to express the trajectory; and third, the selection of the constraints and the optimization method.

Most of the indexes used in bibliography to determine if the trajectories and the measured data are sufficiently exciting are based, either on the *Fisher information* or the condition of the observation matrix $\boldsymbol{W}$. The Fisher information is a well-known tool used in statistics and econometrics and it is a criterion of the amount of information provided by the measurements on the unknown parameters. Meanwhile, if the identification problem can be written in a linear form, as in Equation (1.26), the mentioned analysis can be done throughout criteria derived from the condition of the observation matrix $\boldsymbol{W}$. The condition number measures how much the output value can change for a small change in the input argument. A big condition number means that even a small perturbation in the signals will produce a big perturbation in the parameters, or, in other words, that the data contains little or no information of these parameters [Driels and Pathre, 1990, Khalil and Dombre, 2002]. Hence, the choice of the trajectory can be defined as a non-linear optimization problem, with linear and non-linear constraints and the selected criterion as the cost function.

Some of the cost-functions used in bibliography to optimize the trajectory are:

- *Cost function 1:*
$$C_1 = \text{cond}(\boldsymbol{W}) = \frac{\sigma_{max}}{\sigma_{min}}, \tag{1.36}$$
where σ_{max} and σ_{min} are the maximum and minimum singular values of $\boldsymbol{W}$. The optimization problem consists to find the trajectory that makes C_1 the closest possible to 1 [Gautier and Khalil, 1992, Gautier, 1992].

- *Cost function 2:*
$$C_2 = \text{cond}(\boldsymbol{W}) + k_1 \frac{1}{\sigma_{min}}, \tag{1.37}$$
where k_1 is a weighting scalar. It has been shown in [Presse and Gautier, 1993] that the higher the singular values are, the lower the standard deviation of the parameters is. This cost function prevents having a low condition number but with low singular values when $\boldsymbol{W}$ is badly scaled [Armstrong, 1989].

- *Cost function 3:*
$$C_3 = \text{cond}_{fro}(\boldsymbol{W}) = \sqrt{tr(\boldsymbol{W}^T\boldsymbol{W})tr((\boldsymbol{W}^T\boldsymbol{W})^{-1})}, \tag{1.38}$$
where $tr(.)$ is the operator giving the trace of the matrix. Here, the condition of the observation matrix is obtained using the Frobenius norm which maximizes the determinant of the Fischer Matrix [Vandanjon et al., 1995]. This is a different condition number of the previous ones which were base on the euclidean L-2 norm.

There are others criteria used. For instance, in [Presse and Gautier, 1993], the authors used a priori information about the order of magnitude of each of the dynamic parameters to balance their contribution on the cost function. Moreover, the D-optimality criterion does not depend on the condition number of the regressor matrix, and searches to maximize the determinant of the information matrix. Several works used it, as [Calafiore et al., 2001, Park, 2006, Vantilt et al., 2015] or derivations of it in [Villagrossi et al., 2014]. Furthermore, the

authors in [Jin and Gans, 2015] proposed a new cost function based on Hadamard's inequality which leads to a faster and more robust trajectory design approach than the mentioned ones

The variables that are optimized given one of the cost-functions mentioned above will depend on how the trajectory is expressed. In the first works in this area, researchers used to design a trajectory by interpolation and derivation/integration of points in the workspace, and subsequently check if constraints were satisfied for that specific trajectory [Armstrong, 1989]. This method has a large amount of *dof*: each of the points building the trajectory. Moreover, it may be difficult to satisfy the constraints (e.g. joint limits of position, velocity, acceleration and torque, restrictions on the position of the end-effector, and initial and final conditions) with this trial-and-error technique, which can end up being time consuming and not assuring optimal results.

On the one hand, one way to reduce the computational burden is to divide this problem in one of finding several simpler trajectories. Each of them will lead to identify a sub-set of the set of parameters to be identified (refer to Section 1.4.1.3 for the difference between global and sequential identification). This division can lead to design trajectories just by having a good knowledge of the system and of how to excite the sub-set of parameters that are searched to be identified [Vandanjon et al., 1995].

On the other hand, another approach to simplify this optimization process consists in the parameterization of the trajectory. The amount of *dof* is highly reduced because it will not longer depend on the amount of points to be specified. It also allows to include constraints while optimizing the trajectory, leading to optimal (or sub-optimal) solutions. A well known parameterization for the trajectory of joint i is presented in [Swevers et al., 1997a, Swevers et al., 1997b] based on finite *Fourier* series:

$$\begin{cases} q_i(t) = \sum_{l=1}^{N_i} \frac{a_{l,i}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{l,i}}{\omega_f l} \cos(\omega_f l t) + q_{0_i} \\ \dot{q}_i(t) = \sum_{l=1}^{N_i} a_{l,i} \cos(\omega_f l t) + b_{l,i} \sin(\omega_f l t) \\ \ddot{q}_i(t) = \sum_{l=1}^{N_i} -a_{l,i}\, \omega_f l \sin(\omega_f l t) + b_{l,i}\, \omega_f l \cos(\omega_f l t) \end{cases} , \tag{1.39}$$

where ω_f is the fundamental pulsation of the Fourier series; and N_i indicates the order of the harmonic. There are $2N_i + 1$ parameters corresponding to the amplitudes of the sine and cosine functions $a_{l,i}$ and $b_{l,i}$ respectively, and q_{0_i} is the initial position around which the joint will oscillate. This approach has proven to have interesting features:

- it guarantees periodic trajectories which is advantageous for data filtering,
- it allows the specification of the bandwidth of excitation that can be a design criteria to excite or not different phenomena,
- the calculation of the velocity and acceleration is done in an analytical way which is

more accurate than applying filters and numerical differentiation.

Although this parameterization has been widely used in literature [Vantilt et al., 2015], there are other parameterizations proposed, i.e. a combination of Fourier series and polynomial functions [Park, 2006], optimized B-splines [Rackl et al., 2012, Bonnet et al., 2016], a modified Fourier series [Wu et al., 2012] and a finite sum of harmonic sine functions [Calafiore and Indri, 1998, Calafiore et al., 2001].

After the optimization problem is selected (cost-function and the way to express trajectories), it can be solved by any known algorithm for nonlinear optimization problems with linear and non-linear constraints. Algorithms based on the Sequential Quadratic Programming (SQP) Method [Boggs and Tolle, 1995] are efficient to solve this kind of problems, but there exists other methods as the implementation of genetic algorithms [Calafiore et al., 2001, Villagrossi et al., 2013, Villagrossi et al., 2014].

In case where it is not possible to build an exciting trajectory using an optimization procedure, a random trajectory can be designed and, after application, verify the corresponding cost criterion [Jubien et al., 2014a].

1.3.4 Robot programming and measurements

The typical closed-loop block diagram of a robotic system is shown in Figure 1.7. The controller compares the reference signal (user's choice with an eventual intermediate step of a path planner or trajectory generator) with the feedback signal coming from the sensors, and adequately modifies the robot's input, in order to match, as close as possible, the desired and real robot's behavior. Having already described the robot model, we will shortly describe the other building blocks of the system in this subsection.

The reference input is given either by the user, by a motion planner or by an intermediate stage carried out by an interpolator, and its nature will depend on how the controller is built and what it intends to control. Usually, industrial controllers will ask for points in the joint space $\boldsymbol{q}_{\mathbf{ref}}$ or in the operational space $\boldsymbol{X}_{\mathbf{ref}}$, and by using the kinematic models, it can switch between the two of them. Then, the desired velocity and acceleration profiles are calculated, which will then be transformed in the adequate signals to send to the actuators. There are also other features that the user could provide to the controller, e.g. he/she could specify the desired force that the manipulator should exert on the environment.

Control theory for robotics is a wide topic, presenting numerous techniques: computed-torque control, adaptive control, robust control, force control, etc. The reader is directed to robotics textbooks for more information [Siciliano et al., 2010]. Nevertheless, two important remarks are going to be done. First, although a well-known decentralized PID controller in each of the joints is a low-cost solution simple to implement, its performance will vary depending on the configuration of the robot, and it will have poor accuracy when considering high speeds. If this is not suitable for the system in study, then, it has been proven that a model-based controller outperforms a controller which does not have knowledge of the

system [Khalil and Dombre, 2002]. Generally, the more precise the model is, the better the performance of the controller will be. Second, since the dynamic model is often not exactly known before execution, or it may change during run-time, adaptive control has been investigated in literature and searches to adapt on-line the computational dynamic model [Hsia, 1986, Khalil and Dombre, 2002]. It requires to identify the set of parameters while the robot is carrying out its task.

Equally important for a good performance is the measuring equipment. Data can come from two types of sensors: proprioceptive sensors, which measure the internal state of the manipulator, and exteroceptive sensors, that measure a characteristic from the environment that surrounds the manipulator. On the one hand, proprioceptive sensors measure joint positions, velocities and torques and are essential to control the manipulator. Encoders, potentiometers, resolvers and tachometers are usually used to measure the position and/or velocity, and by future integration and/or derivation, the other internal variables are derived. On the other hand, exteroceptive sensors are usually needed to measure and perceive its situation in the surrounding environment. They can go from force and torque sensors to tactile sensors, proximity sensors, range sensors, vision sensors as cameras, sound and ultra-sound sensors, pressure sensors and temperature sensors. Force sensors are of special importance in collaborative robotics, as they can sense internal variables as well as external effects on the manipulator. These are normally strain gauges, in which the force is indirectly sensed by measuring the strain induced to an extensible element by the force. They can measure the force in one direction, as the ones that are used to measure the link-side torque in collaborative robots [Albu-Schäffer et al., 2007a]. They can also be used to measure the force and the torque at the end-effector in their 3 respective axis in Cartesian space, also called 6-axis force/torque sensor.

Moreover, as these mentioned sensors include an analog to digital converter, thus, they transform a continuous signal into a discrete one, they will unavoidably lead to information losses. To reduce the impact of these losses, it is important to select the sampling interval. The study of sampling is better presented in the frequency-domain, being the theorem of Shannon-Nyquist one of its pillars [Shannon, 1948]. This theorem states that if a system uniformly samples an analog signal at a rate that exceeds the signal's highest frequency by at least a factor of two, the original analog signal can be perfectly recovered from the discrete sampled values without the loss of information. Being T the sampling interval, thus $\omega_s = \frac{2\pi}{T}$ the sampling frequency, then $\omega_n = \frac{\omega_s}{2}$ will be the Nyquist frequency. This frequency is the minimum sampling frequency needed to keep information. Signals which frequency is higher than ω_n will be indistinguishable, phenomenon known as *aliasing*. If the cost of acquisition of data within execution time is not an issue, then, it is highly recommended to sample as fast as possible, and then treat the information as desired [Ljung, 1998].

In this work, the following vocabulary will be used:

- *commanded signal*: it is the reference, the desired behavior. It is what the robot is asked to do. In that way, the *commanded position* will be the trajectory the robot is

asked to follow;

- *sensed signal*: it is the measured or real value coming from the sensors;
- *external signal*: it is a term used for the residual value, e.g. the *external torque* will be the difference between the *sensed torque* and the *commanded torque*.

Finally, having introduced all the building blocks of Figure 1.7, it is important to mention the interface between the human user and an industrial robot, which is different for each robot. Some industrial robots present a high-level interface, where the user just needs to write the desired points. For others, it is necessary to use a programming language in order to code the desired behaviour. There are numerous programming languages usually used in robotics, e.g. C/C++, Java, Fortran, Python, etc. The most popular one is probably C/C++, which enables a low-level hardware interface and real-time performance, although Python is expanding on the area. Moreover, Robot Operating System (ROS) is an open-source set of software libraries and tools that are specially useful to communicate the different modules needed to build robot applications. In terms of simulation, Gazebo and MATLAB are two widely used software. The second one offers a wide variety of tools for robotics based on well-known toolboxes [Corke, 1996, Corke and Khatib, 2011], which have recently been published for Python [Corke and Haviland, 2021].

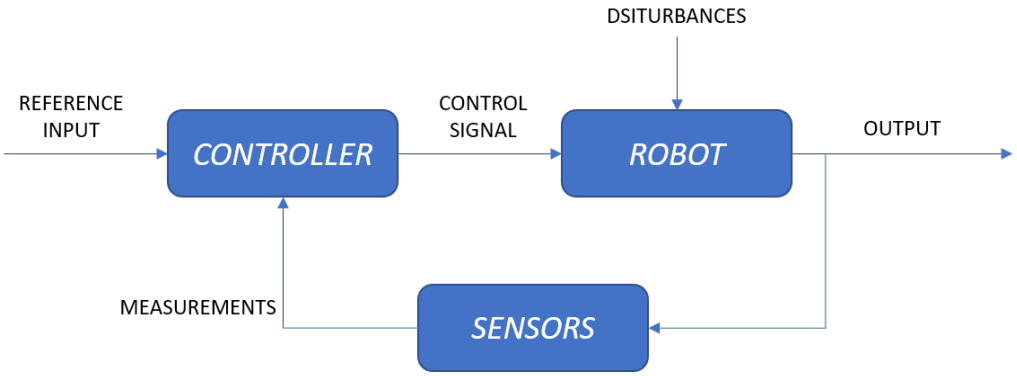


Figure 1.7: General closed-loop block scheme of a robotic system.

1.3.5 Data processing

The acquired data coming from the analog/digital converter of the sensors is in its raw form. The presence of possible outliers, border effects, noise, missing data, drifts, trends and other non-stationary effects has to be dealt before applying the identification method, otherwise, this can lead to poor estimates. Information on how to design filters and process data can be found on any textbook related to digital signal processing [Oppenheim et al., 1997, Anderson and Moore, 2012].

Moreover, integration and/or differentiation of data may be needed for the construction of the regressor matrix. To do so, numerical differentiation is usually carried out. Hence,

the order and type (backward, forward or central differentiation) have to be selected. For instance, the first and second order derivatives in their central and backwards form can be obtained by the Taylor expansion and can be respectively approximated by:

$$\begin{cases} f'(x) = \dfrac{f(x+dt) - f(x-dt)}{2dt} \\ f''(x) = \dfrac{f(x+dt) - 2f(x) + f(x-dt)}{dt^2} \end{cases}, \tag{1.40}$$

$$\begin{cases} f'(x) = \dfrac{f(x) - f(x-dt)}{dt} \\ f''(x) = \dfrac{f(x) - 2f(x-dt) + f(x-2dt)}{dt^2} \end{cases}, \tag{1.41}$$

where dt is the time between two consecutive measurements.

After the processing done in this stage, the sampled model is described by m_f ($m_f \leq m$ depending on amount of deleted points, although they are going to be used indistinctly throughout this manuscript) equations in the form of:

$$\hat{\boldsymbol{\tau}}(t) = \mathbf{IDM}(\hat{\mathbf{q}}(t), \hat{\dot{\mathbf{q}}}(t), \hat{\ddot{\mathbf{q}}}(t))\boldsymbol{\beta} + \boldsymbol{e}(t), \tag{1.42}$$

which can be expressed in its en-bloc form as:

$$\boldsymbol{y}(\hat{\boldsymbol{\tau}}) = \boldsymbol{W}(\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})\boldsymbol{\beta} + \boldsymbol{\rho}, \tag{1.43}$$

where $\hat{\boldsymbol{\tau}}$, $\hat{\mathbf{q}}$, $\hat{\dot{\mathbf{q}}}$ and $\hat{\ddot{\mathbf{q}}}$ refer to the filtered signals.

It is at this moment when the parameter identification method can be applied to obtain the estimates $\hat{\boldsymbol{\beta}}$, which will be the topic of discussion of Section 1.4. Once the parameter identification is done, the validation of results should be carried out. This will be the focus of discussion of the following subsection.

1.3.6 Model validation

"Essentially, all models are wrong, but some are useful." - George E. P. Box.

The goal of identification is not necessarily to develop the true perfect model, but is rather to build a good and useful working one. For this, the purpose and future use of model needs to be defined. The requirements may be different if the model is going to be used for simulation, control, prediction or fault detection.

There are many available tools to validate the model. Usually, the objective is to know how effectively the model fits the output given a specific input (either with training data, or in the cross-validation tests). The smaller the prediction error and the higher the precision of the estimates are, the better the model is. However, other aspects should be analyzed. For

instance, a compromise between fitting the training data and fitting the testing data has to be found, in order to not over-fit the model when carrying out the identification (meaning that the model will predict the output in a very good way for one trajectory, but fail to predict it for others). For this, three tests can be performed:

- *Analysis of residuals*: the residual $\hat{\boldsymbol{\rho}}$ is defined as:

$$\hat{\boldsymbol{\rho}} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}, \tag{1.44}$$

  where $\hat{\boldsymbol{y}}$ is the predicted torque vector. The analysis searches to validate all of the assumptions that are made on the error $\boldsymbol{\rho}$. For instance, verify if residuals are uncorrelated with each other and/or with past inputs can be of importance. They are also called the *whiteness* and *independence test criteria*.

- *Analysis of estimates*: this requires to analyse the vector of estimates $\hat{\boldsymbol{\beta}}$ and their respective statistical distribution. For instance, poor estimates will have high variance in comparison with the mean value, which was explained in Section 1.3.2.2. Besides this, taking advantage of the physical meaning of the estimates, it can be verified if the solution is physically feasible. It is unreasonable to identify a negative mass. This has brought a lot of attention in the latest years and will be introduced in Section 1.3.6.1.

- *Analysis of model fit*: it is used to determine the degree of fit of the prediction with respect to the real measurements. Some metrics are usually needed, such as the percent error of the torque:

$$\%\rho_{y_i} = 100 \left| \frac{\boldsymbol{y_i} - \hat{\boldsymbol{y}}_{\boldsymbol{i}}}{\hat{\boldsymbol{y}}_{\boldsymbol{i}}} \right|, \tag{1.45}$$

  and the root mean square error (RMS):

$$\rho_{\mathrm{RMS},i} = \sqrt{\frac{1}{m}\sum_{h=1}^{m}(y_i(h) - \hat{y}_i(h))^2}. \tag{1.46}$$

1.3.6.1 Physical feasibility

Dynamical parameters have physical meaning and are therefore bounded by physical values. Failing to estimate a physically feasible set of parameters can lead to unrealistic simulations (e.g. negative mass parameters) and unstable model-based control (e.g. not leading to a positive definite inertia matrix). Lately, there has been much work done on this topic [Yoshida et al., 1996, Díaz-Rodríguez et al., 2010, Gautier et al., 2013a, Sousa, 2015, Wensing et al., 2017a, Stürz et al., 2017, Sousa and Cortesao, 2019, Gaz et al., 2019, Janot and Wensing, 2021]. Different ways of approaching the problem have been treated, all based somehow in the verification of the positive definite inertia matrix. In [Yoshida and Khalil, 2000], the authors proposed a trial and error method to verify the feasibility of the numerical value of estimates. In order to automate this process, in [Mata et al., 2005], the authors proposed an unconstrained identification method followed by quadratic programming optimization with

physical constraints. Moreover, the authors in [Ayusawa and Nakamura, 2010] proposed to replace the mentioned constraints for linear inequalities simplifying the optimization problem. Then, [Sousa and Cortesao, 2014] proposed to write the physical feasibility conditions in a semi-definite programming (SDP) perspective as a linear matrix inequality (LMI). This means that the set of conditions is necessarily convex, which will lead to global optimums in the optimization problem. The LMIs are based in the so-called pseudo inertia matrix $\mathbb{J}$ of a rigid body:

$$\mathbb{J}(\boldsymbol{\beta}_{\boldsymbol{st}}^{\boldsymbol{j}}) = \begin{bmatrix} \frac{1}{2}tr(\boldsymbol{I}^{\boldsymbol{j}})\mathbb{I}_{\mathbf{3}} - \boldsymbol{I}^{\boldsymbol{j}} & \mathbf{MS^j} \\ \mathbf{MS^j}^T & M_j \end{bmatrix}. \tag{1.47}$$

The parameters $\boldsymbol{\beta}_{\boldsymbol{st}}^{\boldsymbol{j}}$ are physically consistent if and only if, the pseudo inertia matrix is positive definite:

$$\mathbb{J}(\boldsymbol{\beta}_{\boldsymbol{st}}^{\boldsymbol{j}}) \succ 0, \tag{1.48}$$

where the notation $\mathbb{J} \succ 0$, means that $\mathbb{J}$ is a positive definite matrix.

As having estimates with a real physical meaning is critical for the a posterior utilisation of the model, it is a topic of study that will be treated in depth in Section 2.3.

1.4 Parameter identification methods

Once the model is built and the necessary information is gathered and processed, the parameter identification can be carried out. There are a lot of methods available in literature and their applicability will depend on the characteristics of the system to be identified (type of system, available measurements, etc.) and the expected results. This section starts by making a short summary of the main classification and concepts of identification methods that sets and explains the vocabulary used during the whole work. These aspects, which will be detailed later, can be summarized as:

- linear and non-linear models with respect to parameters,
- identification of open-loop and closed-loop systems,
- global and sequential identification,
- en-bloc and recursive identification,
- online and offline identification.

After these classifications are detailed, we devote Section 1.4.2, Section 1.4.3 and Section 1.4.4 to three identification methods that set the bases to understand methods that will be developed and used during this work:

- the Least-Squares (LS) method based on the IDIM, known as IDIM-LS,

- the Recursive LS (RLS) method based on the IDIM, known as IDIM-RLS,
- the Physically-Consistent LS (PC-LS) method based on the IDIM, known as PC-IDIM-LS.

These three methods are based on the LS solution. As it will be explained later, the IDIM-LS solution is the Best Linear Unbiased Estimator (BLUE) under specific conditions (refer to Appendix A). For example, some assumptions have to me made on the noise, that may not be entirely satisfied or difficult to assure, leading to possible biased estimations. Many methods have been proposed in literature (see second section from each of the following chapters) to deal with this problem. In this work, we will mainly focus on methods based on Instrumental Variables (IV). As it will be explained in Chapter 2, these methods yield estimates needing less assumptions on the noise and making use of an external simulation of the system.

1.4.1 Terminology and classification

1.4.1.1 Linear vs. non-linear with respect to parameters

As said in Section 1.3.1, the IDM of serial manipulators has the important property of being linear with respect to the set of dynamic parameters. A model is considered to be linear with respect to the parameters if it can be expressed in the form of Equation (1.25). The advantage of working with linear models is that the associated mathematics are simpler, more convenient and tractable, being more transparent and easy to comprehend. However, the system can be non-linear with respect to parameters, if, for example, the friction model in Equation (1.15) is included. Hence, the system becomes non-linear with respect to the parameters δ_j and α_{f_j}, and the system will be expressed as:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\beta}) + \boldsymbol{\rho}. \tag{1.49}$$

This non-linearity does not allow the system to be expressed in the linear form, and no closed-form solution exists, thus it needs a numerical solver that will probably lead to a local optima [Tangirala, 2018].

1.4.1.2 Open-loop vs closed-loop

Most of the identification theory considers that the system in study works in open-loop. The outputs and the inputs of the system are related through the dynamics of it, but the output does not have any influence on the future inputs of the system. However, it can be seen that many systems in reality, as shown for robotic manipulators in Figure 1.7, are closed-loop processes. Either because this process is unstable in open-loop or because it is asked to follow a specific reference, the feedback loop is necessary. The fundamental issue with closed-loop data is that the noise and the inputs will be correlated due to the feedback [Gustavsson et al., 1977, Van den Hof, 1997, Forssell and Ljung, 1999].

There are three usual ways to identify a system in closed-loop [Ljung, 1998]:

- *The Direct Approach:* it identifies the open-loop system by using the measurements of the input (output of the controller) and output of the system, by ignoring the feedback and by having a good noise model. Moreover, it can be proven that if the controller has good noise rejection properties and the input of the system (trajectory) is exciting enough (see Section 1.3.3), the bias produced by not knowing the model of noise in this method is reduced considerably.
- *The Indirect Approach:* it consists of two steps. First, it identifies the closed-loop system using the reference signal and its output, as if it was an open-loop system. Then, it identifies the process from the results of the first step and the knowledge of the controller. No knowledge of the noise is needed.
- *The Joint Input-Output Approach:* it considers as if the system was driven by the reference signal and noise (which should be measured), and identifies both the controller and the process without previous knowledge of them.

The choice of which method apply when working with a system in closed-loop almost entirely depends on the available measured signals and on the level of knowledge of the system.

1.4.1.3 Global vs sequential

Besides the nature of the system, the identification can be carried out considering the whole system at once, which means that all inputs and outputs are used at the same time (called global identification), or not, where the data are used sequentially (called sequential identification). The first one has the advantage that the whole set of parameters is identified at the same time, grasping all the coupled effects of moving joints simultaneously. Despite this, it has the drawback that the trajectory selection is a more complicated optimization problem because it involves more variables and a more complex model.

On the contrary, the sequential identification method divides the identification procedure in different steps and different type of trajectories are used to excite particular parameters. For instance, it is possible to move one joint at a time while locking the others. This can be advantageous for identification of certain parameters like the friction ones, as it is an effect usually produced by the rotation of individual joints. The knowledge of the identified parameters in a previous step can be used to identify the following ones. The main advantage of a sequential identification technique is that the trajectory selection is easier and that the identification equations are simpler. However, the identification errors and uncertainties may be accumulated from one identification step to the next one being generally less precise than global identification. Moreover, some parameters are only excitable if more than one joint move simultaneously. Therefore, in [Mayeda et al., 1984, Vandanjon et al., 1995], the authors proposed to generate several types of trajectories moving several joints at the same time,

but with the objective of exciting different physical phenomena: inertial effect, centrifugal coupling, inertial coupling and gravity effect.

1.4.1.4 En-bloc vs recursive

The identification method can be also classified as non-recursive or recursive. Non-recursive methods, also called en-bloc identification, is depicted in Figure 1.8(a), and refers to those methods in which, given a specific trajectory, all the data are used at the same time. They generally have a closed-form expression of the solution, and they can be applied in models which are linear with respect to the parameters, as explained in Section 1.4.1.1. Conversely, a recursive method (see Figure 1.8(c)) treats each measurement once at a time [Young, 2011]. They usually need a first guess or initialization of the parameters to be identified, and the identified parameters at a certain step k will be used and updated at step $k+1$.

The reader should not confuse the terms iterative and recursive, which are sometimes, not adequately defined and used with the same meaning. To highlight this difference, Figure 1.8(b) shows an iterative en-bloc procedure. This iterative method uses the whole data several times and needs a convergence criteria in order to stop the process when a specific condition is fulfilled. The number of iterations needed will depend on the method, the convergence criteria and the information used. Both, en-bloc and recursive methods can be iterative.

1.4.1.5 Offline vs online

Finally, the identification can be carried out while the manipulator is carrying out its trajectory (see Figure 1.8(d) for online identification) or after execution, when all the measurements are already available (see Figure 1.8.(a-c) for offline identification). Online and offline identification are thus differentiated.

On the one hand, offline identification is usually a good approach to make a first description of the system or to identify those parameters that will probably not change during the operation of the system. For example, the parameters of the structure of the links of the manipulator may not change during run-time, so identifying it once, in an offline way, can be enough.

On the other hand, online identification is interesting in the case that the manipulator is intended to work with different loads in unknown and dynamically changing environments, or for fault or collision detection applications [Östring, 2002]. For all of them, it is essential to know how the parameters vary during run-time in order to adapt the control and the behavior of the system depending on the external stimulus. It is evident that, as measurements are not all available at once, when carrying an online identification a recursive method is needed. This implementation has the challenge that all the tasks, such as the data gathering, data processing, parameter identification and further change on the behavior of the system, have to be carried out while the manipulator is performing its task. Thus, the computational cost

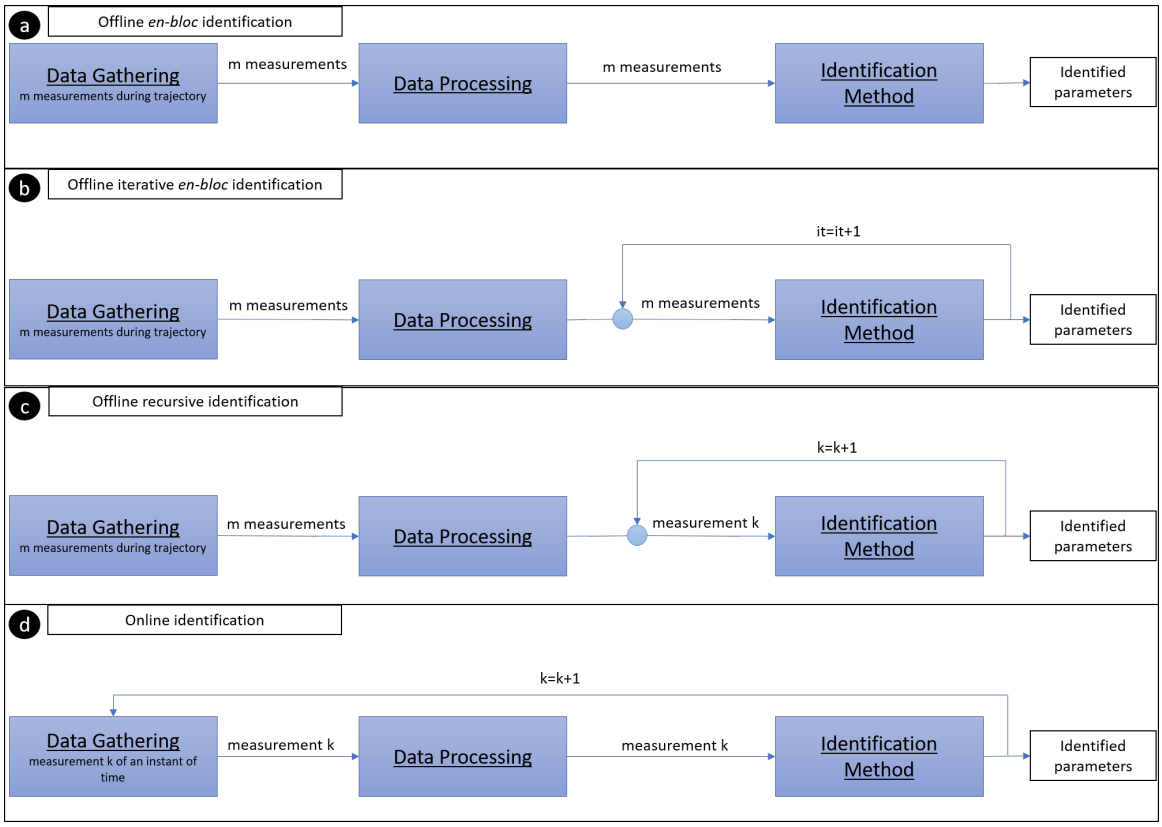


Figure 1.8: Different structures of identification methods: (a) offline en-bloc identification, (b) offline iterative en-bloc identification, (c) offline recursive identification and (d) online identification.

of each of these steps becomes crucial.

A mixture of these two methodologies (online and offline) is called batch identification. This approach, although is applied during the execution of the manipulator (or in between different tasks), does not process the measurements one at a time as a classical online application, but gathers measurements from a certain time range in order to process them all together [Boeren et al., 2017].

All in all, there are many ways to identify the parameters, and it is up to the user to select which method to apply based on many criteria as the objective of the identification, the desired computation time and the available movements that the manipulator is allowed to do. These selections will also influence on the mathematical method used to actually obtain the set of estimates. The next three subsections present some known techniques based on the LS estimation.

1.4.2 The IDIM-LS method

The simplest way to solve the identification problem when treating with models that are linear with respect to the parameters is the Least-Squares (LS) solution. This solution $\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}}$ minimizes the squared 2-norm of the prediction error from the IDIM of Equation (1.43), formulated as:

$$\min_{\boldsymbol{\beta}} ||\hat{\boldsymbol{e}}(t,\boldsymbol{\beta})||_2^2 = \min_{\boldsymbol{\beta}} ||\boldsymbol{\tau}(t) - \mathbf{IDM}(\hat{\mathbf{q}}(t),\hat{\dot{\mathbf{q}}}(t),\hat{\ddot{\mathbf{q}}}(t))\boldsymbol{\beta}||_2^2, \tag{1.50}$$

and can be expressed in its closed form as:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{y}. \tag{1.51}$$

Appendix A shows that, under the assumptions that the error terms have zero conditional mean, they are homoscedastic (constant finite variance for all variables), they are not auto-correlated (diagonal covariance matrix), and the regressor matrix does not present multi-collinearity, the LS estimator is the Best Linear Unbiased Estimate (BLUE). It is mathematically proven that, under the mentioned assumptions, it is not possible to find a better linear unbiased estimate than the LS (tests should be made to validate these assumptions, as the ones made in [Janot et al., 2013a] and others mentioned in Section 1.3.6). Besides showing these proofs and the mathematical steps to get the expression of the estimate, Appendix A shows that the covariance matrix of the estimates can be calculated by:

$$\boldsymbol{C}_{\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}}} = \hat{\sigma}_\rho^2(\boldsymbol{W}^T\boldsymbol{W})^{-1}, \tag{1.52}$$

knowing that the unbiased estimation of the variance is:

$$\hat{\sigma}_\rho^2 = \frac{||\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}}||}{r - n_b}. \tag{1.53}$$

Then, analogically to Equation (1.34), the relative standard deviation of the i^{th} estimate $\hat{\beta}^i_{LS}$ from the set of estimates $\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}}$ is given by:

$$\%\hat{\sigma}_{\hat{\beta}^i_{LS}} = 100\frac{\sqrt{C^{ii}_{\hat{\beta}_{LS}}}}{|\hat{\beta}^i_{LS}|}, \tag{1.54}$$

where $C^{ii}_{\hat{\beta}_{LS}}$ are the diagonal coefficients of $\boldsymbol{C}_{\hat{\boldsymbol{\beta}}_{\boldsymbol{LS}}}$.

This simple method is the base for many others. In the following sections some of them will be explained.

1.4.3 The IDIM-RLS method

The IDIM-RLS is the recursive version of the IDIM-LS. The RLS solution $\hat{\boldsymbol{\beta}}_{RLS}$ is given by the following algorithm [Young, 2011]:

$$\begin{aligned} \boldsymbol{K}(k) &= \boldsymbol{P}(k-1)\mathbf{IDM}^T(k)(\mathbb{I}_n + \mathbf{IDM}(k)\boldsymbol{P}(k-1)\mathbf{IDM}^T(k))^{-1} \\ \boldsymbol{P}(k) &= \boldsymbol{P}(k-1) - \boldsymbol{K}(k)\mathbf{IDM}(k)\boldsymbol{P}(k-1) \\ \hat{\boldsymbol{\beta}}_{RLS}(k) &= \hat{\boldsymbol{\beta}}_{RLS}(k-1) + \boldsymbol{K}(k)(\boldsymbol{\tau}(k) - \mathbf{IDM}(k)\hat{\boldsymbol{\beta}}_{RLS}(k-1)) \end{aligned} \tag{1.55}$$

where $\boldsymbol{P}(k) \in \mathbb{R}^{n_b \times n_b}$ is the covariance matrix, $\boldsymbol{K}(k) \in \mathbb{R}^{n_b \times n}$ is the estimator gain matrix, $\mathbf{IDM}(k) \in \mathbb{R}^{n \times n_b}$ is the observation matrix defined in Equation (1.25), $\boldsymbol{\tau}(k) \in \mathbb{R}^{n \times 1}$ is the vector of torques also defined in Equation (1.25), and $\hat{\boldsymbol{\beta}}_{RLS}(k) \in \mathbb{R}^{n_b \times 1}$ is the vector of RLS estimates respectively, at the instant k, with $k = 1, ..., m$, being m the amount of measurements, and $\mathbb{I}_n$ is the identity matrix of order n.

As regards the implementation of the RLS algorithm, when no a priori information is available, $\hat{\boldsymbol{\beta}}_{RLS}$ should be started at zero and $\boldsymbol{P}$ as a diagonal matrix with large elements (10^6 in general [Young, 2011]). However, in robotics, this initialization brings numerical problems (inversion of matrices as the inertia matrix may not be possible), and an alternative is to initialize them with low values and the adequate sign. The choice of these initial values affects the stability and speed convergence of the algorithm [Gautier and Poignet, 2001]. Therefore, the best way to initialize values is with proper a priori knowledge, either by having the CAD model or by carrying out another identification process before applying the recursive algorithm.

1.4.4 The PC-IDIM-LS method

The PC-IDIM-LS (Physically-Consistent IDIM-LS) problem can be formulated as a semi-definite program (SDP) that minimizes Equation (1.50) subject to the constraints of Equation (1.48) and others corresponding to other phenomena as the friction or the motor (e.g. $I_{a_j} > 0$, $f_{v_j} > 0$, $f_{c_j} > 0$ for $j = 1, ..., n$). In this subsection, just the two parameters of viscous and Coulomb friction will be considered for explanation, although the method can be

scaled to include more constraints.

The condition of Equation (1.48) holds only if the eigenvalues of the matrix $\mathbb{J}$ are positive. Moreover, any positive semi-definite matrix can be decomposed by the LDL method (variant of the Cholesky decomposition) in the form:

$$\mathbb{J} = \boldsymbol{L}_{\mathbb{J}} \boldsymbol{D}_{\mathbb{J}} \boldsymbol{L}_{\mathbb{J}}^{T}, \tag{1.56}$$

where all matrices have the same size than $\mathbb{J}$, i.e. 4x4; $\boldsymbol{L}_{\mathbb{J}}$ is a lower triangular matrix; and $\boldsymbol{D}_{\mathbb{J}}$ is a diagonal matrix whose *ith* diagonal element is denoted $d_{\mathbb{J}_i}$. It has been proven that the condition of Equation (1.48) can be expressed using standard inequality constraints in the form of $d_{\mathbb{J}_i} \geq 0$ [Benson and Vanderbei, 2003].

If we introduce the following extended matrix $\mathbb{J}_e = diag(\mathbb{J}(\boldsymbol{\beta}_{st}^1), f_{c1}, f_{v1}, ..., \mathbb{J}(\boldsymbol{\beta}_{st}^n), f_{cn}, f_{vn})$, then the constraints can be expressed in the form of $h_i(\boldsymbol{\beta_{st}}) = \lambda_i \geq 0$, where $h_i(\boldsymbol{\beta_{st}}) = \lambda_i$ is the i^{th} diagonal of the $D_{\mathbb{J}_e}$ matrix obtained after the LDL decomposition of $\mathbb{J}_e$.

In [Janot and Wensing, 2021], the authors show that the constraints from Equation (1.48) can be then expressed using these inequality constraints, turning the PC-IDIM-LS problem formulation into:

$$\begin{aligned} &\text{minimize} \quad f_0(\boldsymbol{\beta}_{st}), \\ &\text{subject to} \quad h_i(\boldsymbol{\beta_{st}}) \geq 0, \end{aligned} \tag{1.57}$$

where i is the size of the extended matrix $\mathbb{J}_e$; and f_0 is Equation (1.50). They also proved that the PC-IDIM-LS estimates are physically and statistically consistent if $\underset{r\to\infty}{\text{plim}}(\boldsymbol{W}^T\boldsymbol{\rho}) = 0$, where *plim* is the limit in probability as r tends to ∞.

1.5 Conclusion

In this section, the kinematic and dynamic modeling for robotic serial manipulators was described. Then, the main concepts and tools of the parameter identification process were shortly explained, which lays the groundwork for the following sections, where some of the concepts are deepen and practical applications are shown. In the last sections of this chapter, three identification methods based on the LS solution were presented. In the following chapters, we will make use of the knowledge of them to derive other methods, as those based on the IV.

Chapter 2

Offline en-bloc identification

Contents

2.1 Introduction . . . 39
2.2 Problem statement and related works . . . 40
2.3 The new PC-IDIM-IV method . . . 43
2.3.1 The IDIM-IV method . . . 44
2.3.2 An appropriate IV criterion for the PC problem . . . 45
2.3.3 Resolution . . . 47
2.3.4 Statistical analysis . . . 48
2.3.5 Algorithm . . . 49
2.3.6 Experimental results . . . 49
2.3.7 Conclusion . . . 61
2.4 Friction identification . . . 62
2.4.1 The IDIM-SLS method . . . 63
2.4.2 Experimental comparison . . . 64
2.4.3 Conclusion . . . 68
2.5 Final conclusion . . . 69

2.1 Introduction

Chapter 1 presented the basics of robotic manipulators modeling and of the parameter identification process. The current chapter will cover the issue of offline en-bloc identification, where we assume that all the measurements are available prior to the parameter estimation process. Consequently, no study on the computational time and amount of resources to carry out the whole process is needed.

The chapter starts with describing the problem, and making a brief state-of-the-art of methods used in offline *en-bloc* identification of robotic manipulators. Then, there are two main contributions:

- First, a new Instrumental Variables (IV) method including physical consistency is derived, called the PC-IDIM-IV method. For that, the IDIM-IV method is explained,

and then, the mathematical steps to solve the PC problem are derived. Finally, the algorithm is presented and the experimental validation on the KUKA iiwa manipulator is shown.

- Second, a procedure to identify a non-linear friction model on collaborative robots is derived based on the Sequential LS (SLS) identification method. Experimental validation is done on the mentioned manipulator.

2.2 Problem statement and related works

The offline en-bloc identification methods. Offline en-bloc identification is carried out after all the measurements were made and presents a non-recursive solution. A lot of methods have been used in bibliography to solve this problem. The most common approach is to identify the parameters of the IDM via the LS method (both concepts were presented in Chapter 1). The method is called IDIM-LS (Inverse Dynamic Identification Model with Least-Squares) [Gautier, 1986, Caccavale and Chiacchio, 1994, Gautier, 1997, Gautier and Poignet, 2001], and its accuracy lays in the fact that the model is considered to be linear with respect to the parameters to be identified and on some noise assumptions mentioned in Chapter 1. There are several variants that have been proposed to improve this method, as the IDIM-WLS (IDIM - Weighted Least-Squares) [Gautier, 1997, Swevers et al., 2007, Gautier and Briot, 2013] where the data is weighted with the inverse of the estimated covariance of the torque measurements to make a difference in the confidence level between accurate and inaccurate measurements; the IDIM-TLS (IDIM - Total Least-Squares) where errors not only in the torque measurements but also in the regressor matrix (thus in the position measurements) are considered (see [Golub and Van Loan, 1980, Markovsky and Van Huffel, 2007] for an overview of the method, [Briot and Gautier, 2015] for its application in robotics, and [Gautier and Briot, 2012] for its weighted version (IDIM-WTLS)); and the IDIM-IRLS (Iteratively Re-Weighted Least-Squares) presented in [Janot et al., 2009b] (a sequential approach is presented in [Han et al., 2020]) which decreases the influence of outliers. However, all these methods rely on a proper noise rejection and on a good conditioning of the exciting trajectory. Mainly due to the facts that robots work in closed-loop and that it is difficult to ensure noise conditions during the whole execution of a task, the estimates that these methods yield may be biased.

Another type of methods rely on the Maximum-Likelihood (ML) approach. The ML estimation can provide unbiased estimates with minimal uncertainty regardless of the spectrum of the measurement noise [Swevers et al., 1997b, Olsen and Petersen, 2001, Olsen et al., 2002], but at the expense of needing more computational resources to compute the estimates and of a good selection of the initial guess. It is usually solved by nonlinear optimization or a search method which can lead to sub-optimal solutions which may be still biased if the initialization is not done correctly.

Even if a good data filter is applied and good initial conditions are selected, we can doubt if the previous methods yield unbiased estimates or not. This is one of the reasons that other methods have been developed, specially the ones based on Instrumental Variables (IV). These

methods are based on correlation analysis and a realisation that the asymptotic bias caused in the LS estimation by the non-fulfilled assumptions on the noise can be eliminated just by the introduction of special variables called *instruments* [Young, 1985]. A generic IV approach in robotics, named IDIM-IV, was introduced in [Janot et al., 2013b]. There, the *instrument set* is the IDM built from simulated data calculated from the simulation of the DDM. The method also presents the variant which treats noise also in the observation matrix which is named the Simple Refined IV (SRIV) [Vandanjon et al., 2007, Janot et al., 2009c, Janot et al., 2009a]. These methods are attractive since they tend to remove the bias due to noise, they are statistically optimal, they are usually not sensitive to initial conditions and they present a rapid convergence. However, the construction of a valid *instrumental matrix* requires special attention and they generally require more computational effort than the IDIM-LS method.

All the previous methods consider that both the inputs and the outputs of the manipulator are available. In the contrary, there are other methods which are designed to be applied when not all the signals are accessible to the user. This is the case of the so-called Output-Error (OE) methods. Their objective is to minimize, by means of non-linear optimization, a quadratic error between the actual output and a simulated output of the system, assuming both systems (real and simulated) have the same input [Brunot et al., 2020], e.g. the Closed-Loop Output Error (CLOE), proposed in [Gautier et al., 2012], and the DIDIM (Direct and Inverse Dynamic Identification Models) used in [Gautier et al., 2013b, Janot et al., 2014, Brunot et al., 2017]. These methods do not require the measurement of the torques. Inversely to the OE methods, the Closed-Loop Input Error (CLIE) method requires just the measurement of the torques and no measurement of the positions and derivatives [Brunot et al., 2020].

Furthermore, as said in Section 1.3.6.1, the physical feasibility of the estimates has been the focus of much attention in last years [Sousa and Cortesao, 2014]. Therefore, the addition of constraints to achieve this objective gives raise to new methods (the letters PC, for Physically-Consistent, are added to the acronym of the method), as the PC-IDIM-LS [Mata et al., 2005, Vuong and Ang Jr, 2009] and the PC-IDIM-WLS [Bahloul et al., 2018].

Because of the importance of having unbiased reliable estimates with physical meaning, we will present, in the next section of this chapter, a new identification method called PC-IDIM-IV. This algorithm has the good noise rejection properties of an IV method, and it also includes PC constraints on the standard parameters that leads to a physically feasible solution.

The friction identification. Friction modelling has been addressed in Section 1.2.2, and many of the works cited therein are focused in its identification. If the model is linear with respect to the parameters as the model in Equation (1.14), then, the estimates of the friction can be obtained as if they were like any other dynamic parameter by methods as the ones already mentioned. If the model is chosen to be non-linear with respect to the parameters, as in Equation (1.15), then, other methods may be needed. In [Indri and Trapani, 2020], the authors proposed a method in which they fix the parameters that generate the non-linearity in order to obtain a model linear with respect to the remaining parameters and

apply the IDIM-LS method. To do that, they used a minimum search algorithm where they tested several values of the parameters causing the non-linearities and they selected the set of parameters that minimized the error. Although this method has proven to be efficient and not computationally heavy for the mentioned model of friction, it does not ensure optimal results.

Because of this reason, in this chapter, in Section 2.4, we will propose solving the identification problem with the SLS (Separable LS) method [Golub and Pereyra, 1973]. This is a method which is applied to systems which are non-linear with respect to some of the parameters to be estimated and linear with respect to the others, and allows to divide the problem into a two-part process: one in which an optimization is carried out to find an optimal value for the parameters which produce the non-linearity, and another step where the IDIM-LS method is applied to the remaining model considering the parameters obtained in the first part as known.

The robots. The advent of collaborative robots and physical human robot interaction (pHRI) brought new challenges. Notably, the presence of non-negligible flexibility in the joints. This has been treated in works as [Albu-Schaffer and Hirzinger, 2001, Janot et al., 2011, Dumas et al., 2012, Ni et al., 2019], and many manipulators have been used for identification purposes, e.g., [Taghbalout et al., 2019] applied IDIM-LS on the ABB IRB14000 (YuMi) robot, [Bargsten et al., 2013, Jubien et al., 2014b] applied the IDIM-LS on the KUKA LWR-IV, [Gaz et al., 2019] used a PC-IDIM-LS method on the increasingly popular Panda manipulator manufactured by Franka Emika, [Kovincic et al., 2019, Raviola et al., 2021] applied several methods on the UR3 and UR5 manipulators and [Stürz et al., 2017, Xu et al., 2020] applied IDIM-LS and PC-IDIM-LS on the KUKA IIWA. In addition to the higher mechanical design complexity compared to the classical industrial manipulators, the fact that commercial collaborative robots hide crucial information about the incorporated controller, model and measurements, makes the identification task more difficult. Works as [Jubien et al., 2014a, Gaz et al., 2014] on the KUKA LWR-IV have retrieved the *confidential* parameters of the model that the controller has of the robot. This is of advantage as it derives the model that the manufacturer had of the manipulator, which replaces the often not-available CAD model of the manipulator.

Even though the methods presented on this thesis are general and can be applied to any kind of robot, the experimental results are mostly shown by analyzing the KUKA iiwa manipulator. This manipulator has gain a certain popularity and many works have been based on its use [Besset et al., 2016, Chawda and Niemeyer, 2017a, Chawda and Niemeyer, 2017b, Sellami and Respall, 2020, Mujica et al., 2023]. Besides the two works mentioned on the previous paragraph which present a complete model of the manipulator ([Stürz et al., 2017, Xu et al., 2020]), [Hennersperger et al., 2016] also presents a set of parameters of the KUKA iiwa, but without an identification procedure.

Furthermore, two appendixes are included in this work to allow the total reproduction of almost every experimental validation carried out. On the one hand, the framework used for the experiments on the KUKA iiwa robot is shown in Appendix B. The description of the

manipulator, the load, the computer, the added sensors and the way they are communicated and coded is made. Moreover, we describe the experimental preliminaries: the way to obtain the set of base parameters from the set of standard parameters, the selection of the trajectories and of the data processing. Finally, the analysis of the signals of the manipulator is made in order to support the hypothesis made regarding the model-based characteristic of the KUKA controller. On the other hand, Appendix C presents the *MESSII dataset*, which stores many of the trajectories that are used all along this manuscript. The willing to make science as open and achievable for society as possible, lead to the creation of this dataset. It is a way to give the research community and society in general the tools to both: verify publications based on the dataset either because it may have experiments that are difficult, tedious, or impossible to replicate, and also to give the possibility to fellow multidisciplinary researchers to continue the research work and deepen the knowledge on one topic from different points of views.

Despite this, methods derived in this chapter and following ones can also be implemented on other robots or systems, as it is shown in Appendix E, which is based on our work of identification in [Ardiani et al., 2021], and in Appendix D. These appendices are not included in the main part of the text to maintain a balance in the length of the chapters and not to overflow the chapter with experimental results. In the first one, besides obtaining a new model of unbalanced two-wheeled differential drive mobile robots, we compared the IDIM-LS and IDIM-IV methods on simulation showing the potential advantage of the non-exploited IV methods on mobile robotics. The second Appendix validates the new PC-IDIM-IV method on the Stäubli TX40 6-*dof* manipulator. We compare its results with those of the PC-IDIM-LS, the IDIM-IV and the PC-DIDIM methods, showing its advantages.

2.3 The new PC-IDIM-IV method

As already mentioned, the LS estimates of Equation (1.51) may be biased because robots are identified with closed-loop procedures and noisy measurements that induce correlations, putting into doubt the validity of one of the stated conditions needed for an unbiased LS estimator:

$$\mathbb{E}(\boldsymbol{W}^T\boldsymbol{\rho}) = 0, \tag{2.1}$$

where $\mathbb{E}(.)$ is the expectation operator. This is why it is interesting to study the IV method. It was probably first introduced by the works in [Reiersøl, 1941, Reiersøl, 1945], although it was simultaneously developed by Geary (an interesting paper discussing these first works can be found in [Aldrich, 1993]).

In this section, we will start by describing the known IDIM-IV method, in order to afterwards derive the new PC-IDIM-IV method. For that, we will demonstrate through mathematical steps the validity of the proposed instrument and we will derive an algorithm to solve the problem. Finally, we validate the proposed algorithm with experimental results.

2.3.1 The IDIM-IV method

The IV method is able to remove the bias of LS estimation produced by the endogeneity of the independent variables by using an instrumental matrix denoted $\boldsymbol{Z} \in \mathbb{R}^{r \times n_b}$ [Young, 1985]. This matrix must possess two specific properties to ensure consistent estimates [Young, 2011, Janot et al., 2012]:

- $\boldsymbol{Z}$ must be correlated with $\boldsymbol{W}$ (sampled IDM) so that $\boldsymbol{Z}^T\boldsymbol{W}$ is invertible. In other words: $\mathbb{E}(\boldsymbol{Z}^T\boldsymbol{W})$ exists, is finite, and of full rank n_b.
- $\boldsymbol{Z}$ must be uncorrelated with the noise $\boldsymbol{\rho}$, thus $\mathbb{E}(\boldsymbol{Z}^T\boldsymbol{\rho}) = 0$.

Then, the simple IV estimator provides unbiased estimates by [Ljung, 1998]:

$$\hat{\boldsymbol{\beta}}_{IV} = (\boldsymbol{Z}^T\boldsymbol{W})^{-1}\boldsymbol{Z}^T\boldsymbol{y}. \tag{2.2}$$

Notice that if we replace $\boldsymbol{y}$ with Equation (1.26), we take the expectation operator and we consider the two previously mentioned assumptions, we prove the consistency of the estimates by:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}_{IV}) = \mathbb{E}((\boldsymbol{Z}^T\boldsymbol{W})^{-1}\boldsymbol{Z}^T\boldsymbol{W}\boldsymbol{\beta}) + \mathbb{E}((\boldsymbol{Z}^T\boldsymbol{W})^{-1}\boldsymbol{Z}^T\boldsymbol{\rho}) = \boldsymbol{\beta}. \tag{2.3}$$

The way how to build the instrumental matrix $\boldsymbol{Z}$ has drawn a lot of attention over the years. According to [Garnier et al., 2008, Young, 2011], a good way to build it is by means of simulated data only. This data is produced by a mathematical model of the system to be identified, which is inherently noise-free. Thus, in this case, we consider $\boldsymbol{Z} = \boldsymbol{W}_{nf}$, being $\boldsymbol{W}_{nf}$ the noise free observation matrix defined as:

$$\boldsymbol{W} = \boldsymbol{W}_{nf} + \boldsymbol{V}. \tag{2.4}$$

The system is then expressed in the form of:

$$\boldsymbol{y} = \boldsymbol{W}_{nf}\boldsymbol{\beta} + \boldsymbol{e}_{\boldsymbol{y}}, \tag{2.5}$$

where $\boldsymbol{V} \in \mathbb{R}^{r \times n_b}$ is a matrix of error terms; and $\boldsymbol{\rho} = \boldsymbol{e}_{\boldsymbol{y}} - \boldsymbol{V}\boldsymbol{\beta}$. The matrix $\boldsymbol{V}$ is uncorrelated with $\boldsymbol{W}_{nf}$, thus $\mathbb{E}(\boldsymbol{V}^T\boldsymbol{W}_{nf}) = 0$, and with the error $\boldsymbol{e}_{\boldsymbol{y}}$, thus $\mathbb{E}(\boldsymbol{V}^T\boldsymbol{e}_{\boldsymbol{y}}) = 0$. Notice that $\boldsymbol{W}$ is correlated with $\boldsymbol{\rho}$ because of $\boldsymbol{V}$, which has already been said to be the reason that leads to biased LS estimates, and the motivation for the implementation of the IV method.

To verify that $\boldsymbol{Z} = \boldsymbol{W}_{nf}$ is a valid instrument choice, we have to validate the two mentioned properties. First, replacing the relations on the first condition and using properties of the expectation operator:

$$\mathbb{E}(\boldsymbol{Z}^T\boldsymbol{W}) = \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{W}_{nf}) + \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{V}) = \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{W}_{nf}). \tag{2.6}$$

Then, the condition can be verified by analyzing $\boldsymbol{W}_{nf}$.

The second condition is verified because:

$$\mathbb{E}(\boldsymbol{Z}^T\boldsymbol{\rho}) = \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{\rho}) = \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{e_y}) - \mathbb{E}(\boldsymbol{W}_{nf}^T\boldsymbol{V}) = 0. \tag{2.7}$$

With these, it has been shown that the noise-free observation matrix $\boldsymbol{W}_{nf}$ is a valid instrumental matrix [Janot et al., 2013b].

Moreover, it has also been shown in [Janot et al., 2013b] that the relation $\boldsymbol{Z} \approx \boldsymbol{W}_{nf}$ is obtained by simulating the DDM using the previous estimation $\hat{\boldsymbol{\beta}}_{IV}^{it-1}$, and considering the same reference trajectory and control law for the actual and simulated robot, obtaining $\boldsymbol{Z}^{it} = \boldsymbol{W}_{nf}^{it} = \boldsymbol{W}(\boldsymbol{q_s}, \dot{\boldsymbol{q}}_{\boldsymbol{s}}, \ddot{\boldsymbol{q}}_{\boldsymbol{s}}, \hat{\boldsymbol{\beta}}_{IV}^{it-1})$, where $\boldsymbol{q_s}$, $\dot{\boldsymbol{q}}_{\boldsymbol{s}}$, $\ddot{\boldsymbol{q}}_{\boldsymbol{s}}$ are the simulated joint positions, velocities and accelerations, respectively. Then, the consistent IV estimates at iteration it are given by:

$$\hat{\boldsymbol{\beta}}_{IV}^{it} = ({\boldsymbol{Z}^{it}}^T\boldsymbol{W})^{-1}{\boldsymbol{Z}^{it}}^T\boldsymbol{y}. \tag{2.8}$$

It is an iterative process because the estimates obtained in one iteration are used to generate the instrumental matrix of the next iteration. The process ends when a criterion is full-filed, e.g. when the relative difference between two successive estimates is lower than a chosen threshold.

However, it has been shown in [Janot et al., 2012] that even though IV algorithms are robust against initialization, a bad choice of the initial values may lead to algorithm divergence due to the closed-loop influence. Their solution was to modify the controller of the simulation at each iteration considering the estimated parameters on the previous step, to match the same performance than that of the real controller.

In the next subsection, we will focus on showing the validity of the IV selection for the PC-IDIM-IV problem.

2.3.2 An appropriate IV criterion for the PC problem

The transformation of the PC-IDIM-LS method introduced in Section 1.4.4 to its IV version is not straightforward. To begin with, before inserting the PC constraints, we must choose an appropriate IV criterion. The diverse IV criteria considered in the literature are not satisfying since they deal with the identifiable parameters $\boldsymbol{\beta}$ while we will consider the standard parameters $\boldsymbol{\beta_{st}}$. This choice is done because the PC constraints are directly applicable to them [Young, 2011, Gilson et al., 2011]. So, bridging the gap between using classical IV approaches and inserting physical constraints is not straight. In Econometrics, they usually employ the following IV criterion for over-determined cases where there are more instruments than covariates in the equation of interest [Davidson et al., 1993, Wooldridge, 2015]:

$$f_0(\beta_{st}) = ||\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{\rho}||_2^2 = ||\mathbf{P}_{\boldsymbol{Z}}(\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\beta})||_2^2 = |\mathbf{P}_{\boldsymbol{Z}}(\boldsymbol{y} - \boldsymbol{W}_{st}\boldsymbol{\beta}_{st})||_2^2\,, \tag{2.9}$$

where $\mathbf{P}_{\boldsymbol{Z}} = \boldsymbol{Z}(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T$, with $\mathbf{P}_{\boldsymbol{Z}} \in \mathbb{R}^{r\times r}$, is the orthogonal projector onto the space generated by the columns of $\boldsymbol{Z}$ [Davidson et al., 1993]. Note that we still consider $\boldsymbol{Z}$ to build $\mathbf{P}_{\boldsymbol{Z}}$ and not $\boldsymbol{Z}_{st} = \boldsymbol{W}_{st}(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s, \ddot{\boldsymbol{q}}_s)$ because $\boldsymbol{Z}$ is full rank whereas $\boldsymbol{Z}_{st}$ is not. Further, this matrix has two important properties that can be easily proved with some basic algebra: the matrix is symmetric, thus $\mathbf{P}_{\boldsymbol{Z}}^T = \mathbf{P}_{\boldsymbol{Z}}$, and the matrix is idempotent, thus $\mathbf{P}_{\boldsymbol{Z}}^2 = \mathbf{P}_{\boldsymbol{Z}}$ (which has the logical explanation that projecting for a second time into the same space does not affect the result of the first projection).

Expanding (2.9) yields:

$$\begin{aligned} f_0(\boldsymbol{\beta}_{st}) &= ||\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{\rho}||_2^2 \\ &= \boldsymbol{\rho}^T\mathbf{P}_{\boldsymbol{Z}}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{\rho} \\ &= \boldsymbol{\rho}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{\rho} \\ &= (\boldsymbol{y} - \boldsymbol{W}_{st}\boldsymbol{\beta}_{st})^T\mathbf{P}_{\boldsymbol{Z}}(\boldsymbol{y} - \boldsymbol{W}_{st}\boldsymbol{\beta}_{st}) \\ &= \boldsymbol{y}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} - \boldsymbol{y}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st}\boldsymbol{\beta}_{st} - \boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} + \boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st}\boldsymbol{\beta}_{st} \\ &= \boldsymbol{y}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} - 2\boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} + \boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st}\boldsymbol{\beta}_{st} \end{aligned} \tag{2.10}$$

It seems that we cannot pursue further since we do not know what the projection of $\boldsymbol{W}_{st}$ onto the space generated by the columns of $\boldsymbol{Z}$ gives $(\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st})$.

To answer this, from Section 1.3.2, we know that the base parameters are a linear combination of the set of standard parameters which can be expressed in the form [Gautier and Khalil, 1990]:

$$\boldsymbol{\beta} = \boldsymbol{\beta}_I + \mathbf{R}_{\mathbf{t}}\boldsymbol{\beta}_{NI}\,, \tag{2.11}$$

where $\boldsymbol{\beta}_I \in \mathbb{R}^{n_b}$ is the vector of identifiable standard parameters, meaning those standard parameters that will not be reason of regrouping or deletion ($\boldsymbol{\beta}_{\mathbf{1}}$ in Equation (1.33) if the QR decomposition is used); $\boldsymbol{\beta}_{NI} \in \mathbb{R}^{n_{st}-n_b}$ is the vector of non-identifiable standard parameters regrouped with $\boldsymbol{\beta}_I$ ($\boldsymbol{\beta}_{\mathbf{2}}$ in Equation (1.33) if the QR decomposition is used); and $\mathbf{R}_{\mathbf{t}} \in \mathbb{R}^{n_b\times(n_{st}-n_b)}$ is the matrix of linear regroupings (which is equal to $\boldsymbol{R_1}^{-1}\boldsymbol{R_2}$ from Equation (1.33) if the QR decomposition is used). We then have that:

$$\boldsymbol{\beta}_{st} = \begin{bmatrix} \boldsymbol{\beta}_I \\ \boldsymbol{\beta}_{NI} \end{bmatrix}. \tag{2.12}$$

Since $\boldsymbol{y} = \boldsymbol{W}_{st}\boldsymbol{\beta}_{st} = \boldsymbol{W}\boldsymbol{\beta}$, it follows straight that $\boldsymbol{y} = \boldsymbol{W}_{st}\boldsymbol{\beta}_{st} = \boldsymbol{W}\boldsymbol{\beta}_I + \boldsymbol{W}\mathbf{R}_{\mathbf{t}}\boldsymbol{\beta}_{NI} = [\boldsymbol{W} \;\; \boldsymbol{W}\mathbf{R}_{\mathbf{t}}]\begin{bmatrix}\boldsymbol{\beta}_I^T & \boldsymbol{\beta}_{NI}^T\end{bmatrix}^T = [\boldsymbol{W} \;\; \boldsymbol{W}\mathbf{R}_{\mathbf{t}}]\,\boldsymbol{\beta}_{st}$.

Moreover, with $\boldsymbol{W} = \boldsymbol{Z} + \boldsymbol{V}$, one obtains $\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W} = \mathbf{P}_{\boldsymbol{Z}}\boldsymbol{Z} + \mathbf{P}_{\boldsymbol{Z}}\boldsymbol{V} = \boldsymbol{Z}$. Indeed, the space generated by the columns of $\boldsymbol{V}$ is orthogonal to the one generated by the columns of $\boldsymbol{Z}$ by definition, i.e., $\boldsymbol{Z}^T\boldsymbol{V} = \mathbf{0}$ [Janot et al., 2013b], and the projection of $\boldsymbol{Z}$ into its own space is $\boldsymbol{Z}$ itself. Now, with $\boldsymbol{W} = \boldsymbol{Z} + \boldsymbol{V}$, and $\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{Z} = \boldsymbol{Z}$, one has $\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st} = \mathbf{P}_{\boldsymbol{Z}}\,[\boldsymbol{W} \;\; \boldsymbol{W}\mathbf{R}_{\mathbf{t}}] = [\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{Z} \;\; \mathbf{P}_{\boldsymbol{Z}}\boldsymbol{Z}\mathbf{R}_{\mathbf{t}}] + [\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{V} \;\; \mathbf{P}_{\boldsymbol{Z}}\boldsymbol{V}\mathbf{R}_{\mathbf{t}}] = [\boldsymbol{Z} \;\; \boldsymbol{Z}\mathbf{R}_{\mathbf{t}}] = \boldsymbol{Z}_{st}$, which will be the chosen instrumental matrix for the PC-IDIM-IV problem as it is directly related to the standard parameters. We have proven that the projection of $\boldsymbol{W}_{st}$ onto the space generated by the columns of $\boldsymbol{Z}$ is the

noise free regressor matrix built with the IDM with respect to the standard parameters.

Finally, continuing with the derivation of Equation (2.10) and replacing with the new definition of the instrumental matrix $\boldsymbol{Z}_{st} = \mathbf{P}_{\boldsymbol{Z}}\boldsymbol{Z}$:

$$\begin{aligned} f_0(\boldsymbol{\beta}_{st}) &= ||\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{\rho}||_2^2 \\ &= \boldsymbol{y}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} - 2\boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} + \boldsymbol{\beta}_{st}^T\boldsymbol{W}_{st}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{W}_{st}\boldsymbol{\beta}_{st} \\ &= \boldsymbol{y}^T\mathbf{P}_{\boldsymbol{Z}}\boldsymbol{y} - 2\boldsymbol{\beta}_{st}^T\boldsymbol{Z}_{st}^T\boldsymbol{y} + \boldsymbol{\beta}_{st}^T\boldsymbol{Z}_{st}^T\boldsymbol{W}_{st}\boldsymbol{\beta}_{st}, \end{aligned} \tag{2.13}$$

which gives the expression of the function to minimize in the PC-IDIM-IV problem. Its resolution will be explained in the following subsection, which is based in the minimization of $f_0(\boldsymbol{\beta}_{st})$.

2.3.3 Resolution

We can express the PC-IDIM-IV problem, in the same way as the PC-IDIM-LS in Section 1.4.4, as an inequality constrained programming problem formulated as:

$$\begin{aligned} &\text{minimize} \quad f_0(\boldsymbol{\beta}_{st})\,, \\ &\text{subject to} \quad h_i(\boldsymbol{\beta}_{st}) \geq 0\,, \ \text{for } i = 1, \cdots, p \end{aligned} \tag{2.14}$$

where p is the amount of constraints coming from the positive definiteness of the equivalent inertia matrix and from other parameters as the friction, explained in Section 1.3.6.1 and Section 1.4.4. Now, $f_0(\boldsymbol{\beta}_{st})$ takes the form given by (2.13).

We will make use of the Karush – Kuhn – Tucker (KKT) conditions to solve Equation (2.14) [Wright et al., 1999]. They are first derivative tests to resolve, in an optimal way, optimization problems which have either nonlinear constraints or a nonlinear objective function, also called nonlinear programming (NLP). Our problem can be stated as follows: if $\boldsymbol{\beta}_{st}^*$ is an optimizer for Equation (2.14), there exists a multiplier $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that:

$$\begin{aligned} 2\boldsymbol{Z}_{st}^T\boldsymbol{W}_{st}\boldsymbol{\beta}_{st}^* - 2\boldsymbol{Z}_{st}^T\boldsymbol{y} + \mathbf{A}^T\boldsymbol{\mu}^* &= \mathbf{0}\,, \\ h_i(\boldsymbol{\beta}_{st}^*) &\geq 0\,, \ \text{for } i = 1 \cdots p\,, \\ \mu_i^* h_i(\boldsymbol{\beta}_{st}^*) &= 0\,, \ \text{for } i = 1 \cdots p\,, \\ \mu_i^* &\geq 0\,, \ \text{for } i = 1 \cdots p\,, \end{aligned} \tag{2.15}$$

where $\boldsymbol{A} \in \mathbb{R}^{(p \times n_{st})}$ is the Jacobian matrix of $\boldsymbol{h}(\boldsymbol{\beta}_{st}) = [h_1 \cdots h_p]^T$; and $\boldsymbol{\mu^*} = \left[\mu_1^* \cdots \mu_p^*\right]^T$. It is assumed that $rank(\boldsymbol{A}) = p$. Then, by introducing the slack variable $\boldsymbol{s} \geq \mathbf{0}$ with $\boldsymbol{s} \in \mathbb{R}^p$, the above inequalities conditions can be equivalently formulated as equality conditions as follows:

$$\begin{aligned} 2\boldsymbol{Z}_{st}^T\boldsymbol{W}_{st}\boldsymbol{\beta}_{st} - 2\boldsymbol{Z}_{st}^T\boldsymbol{y} + \boldsymbol{A}^T\boldsymbol{\mu}^* &= \mathbf{0}\,, \\ \boldsymbol{h}(\boldsymbol{\beta}_{st}^*) - \boldsymbol{s} &= \mathbf{0}\,, \\ s_i\mu_i^* &= 0\,, \ \text{for } i = 1 \cdots p\,, \\ s_i, \mu_i^* &\geq 0\,, \ \text{for } i = 1 \cdots p\,. \end{aligned} \tag{2.16}$$

An interior-point strategy is adopted to solve the constraint satisfaction problem described by Equation (2.15). This choice is motivated by the fact that such methods are very popular and are easily accessible though multiple solvers used by CVX [Grant and Boyd, 2014]. The triplet $\boldsymbol{s}$, $\boldsymbol{\mu}$, $\boldsymbol{\beta}_{st}$ is calculated by executing a nonlinear programming algorithm such as the Gauss-Newton. All the details are given in [Janot and Wensing, 2021], being sufficient to replace $\boldsymbol{W}_{st}^T\boldsymbol{W}_{st}$ by $\boldsymbol{Z}_{st}^T\boldsymbol{W}_{st}$.

2.3.4 Statistical analysis

Like in [Janot and Wensing, 2021], finding the conditions to get consistent estimates is complicated by the fact that many set of standard parameters $\boldsymbol{\beta}_{st}$ produce the same data. Indeed, all $\boldsymbol{\beta}_{st}$ resulting in the same data lie in an affine subspace of $\mathbb{R}^{n_{st}}$ denoted:

$$\mathcal{D} = \{\boldsymbol{\beta}_{st_d} \in \mathbb{R}^{n_{st}} \mid \boldsymbol{\beta} = [\mathbb{I}_{n_b} \quad \mathbf{R_t}]\,\boldsymbol{\beta}_{st_d}\}, \tag{2.17}$$

which is also the subspace of parameters that leads to the true base parameters [Gautier et al., 2013a]. Thus, we shall only analyze the convergence of the set of estimated base parameters [Ljung, 1998]. Hence, by noting $\hat{\boldsymbol{\beta}}_{st_{IV}}$ the PC-IDIM-IV estimates, and if $\hat{\boldsymbol{\beta}}_{st_{IV}} \in \mathcal{D}$ given by Equation (2.17), we have $\underset{r\to\infty}{\text{plim}}\left([\mathbb{I}_{n_b} \quad \mathbf{R_t}]\,\hat{\boldsymbol{\beta}}_{st_{IV}}\right) = \underset{r\to\infty}{\text{plim}}\left(\hat{\boldsymbol{\beta}}_{IV}\right) = \boldsymbol{\beta}$, where $\underset{r\to\infty}{\text{plim}}$ is the limit in probability as r tends to ∞. Thus, the estimates of the standard parameters will tend to the real base parameters as r gets bigger.

After similar calculations than those conducted in [Janot and Wensing, 2021] (see from Equation (38) to Equation (44) in that publication), the PC-IDIM-IV estimates are consistent if $\underset{r\to\infty}{\text{plim}}\left(\boldsymbol{Z}_{st}^T\boldsymbol{\rho}\right) = \mathbf{0}$. If $\boldsymbol{Z}_{st}$ is a valid instrumental matrix, then at the last iteration we have:

$$\boldsymbol{Z}_{st} = \boldsymbol{W}_{st_{nf}} = [\boldsymbol{W}_{nf} \quad \boldsymbol{W}_{nf}\boldsymbol{R_t}]\,. \tag{2.18}$$

This yields

$$\boldsymbol{Z}_{st}^T\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{Z}^T\boldsymbol{\rho} \\ \boldsymbol{R_t}^T\boldsymbol{Z}^T\boldsymbol{\rho} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{nf}^T\boldsymbol{\rho} \\ \boldsymbol{R_t}^T\boldsymbol{W}_{nf}^T\boldsymbol{\rho} \end{bmatrix}, \tag{2.19}$$

giving

$$\underset{r\to\infty}{\text{plim}}\left(\boldsymbol{Z}_{st}^T\boldsymbol{\rho}\right) = \underset{r\to\infty}{\text{plim}}\left(\begin{bmatrix} \boldsymbol{Z}^T\boldsymbol{\rho} \\ \boldsymbol{R_t}^T\boldsymbol{Z}^T\boldsymbol{\rho} \end{bmatrix}\right) = \underset{r\to\infty}{\text{plim}}\left(\begin{bmatrix} \boldsymbol{W}_{nf}^T\boldsymbol{\rho} \\ \boldsymbol{R_t}^T\boldsymbol{W}_{nf}^T\boldsymbol{\rho} \end{bmatrix}\right) = \mathbf{0}, \tag{2.20}$$

since $\mathbb{E}(\boldsymbol{\rho}) = \mathbf{0}$ by definition. This proves the consistency of the estimates.

Moreover, it is useful to clarify that, in Equation (2.16), if $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are quasi-stationary signals whose the first four moments are finite, then, $\underset{r\to\infty}{\text{plim}}\left(\frac{1}{r}\boldsymbol{Z}_{st}^T\boldsymbol{W}_{st}\right)$ exists and is finite [Ljung, 1998] .

Regarding the covariance matrix of the PC-IDIM-IV estimates, we could do as suggested in [Janot and Wensing, 2021], where the authors considered $\hat{\boldsymbol{\beta}}_{IV}$ instead of $\hat{\boldsymbol{\beta}}_{st_{IV}}$. If this way is feasible, it is better to consider $\hat{\boldsymbol{\beta}}_{IV}$ since we used it to make the statistical analysis of

PC-IDIM-IV. It follows that the covariance matrix of $\hat{\boldsymbol{\beta}}_{IV}$ is

$$\mathbf{C}_{\hat{\boldsymbol{\beta}}_{\boldsymbol{IV}}} = \left(\boldsymbol{Z}^T \boldsymbol{\Omega}^{-1} \boldsymbol{Z}\right)^{-1}, \tag{2.21}$$

where errors are assumed serially uncorrelated and with finite variance such that the covariance $\boldsymbol{\Omega}$ can be partitioned as

$$\boldsymbol{\Omega} = \text{diag}\left(\sigma_1^2 \mathbb{I}_m, \ldots, \sigma_n^2 \mathbb{I}_m\right). \tag{2.22}$$

Following the procedure in [Gautier et al., 2012], each σ_j is estimated from the standard deviation of the error between the real torque and the reconstructed one, similarly to Equation (1.53).

In this subsection, we showed the existence and consistency of the estimates obtained with the proposed PC procedure. We also described the way to obtain the statistical properties of the estimates, which is important to determine their relevance and derive the set of essential parameters.

2.3.5 Algorithm

In this subsection, we summarize the PC-IDIM-IV algorithm, shown in Figure 2.1:

Step 0: collect $\boldsymbol{q}$ and $\boldsymbol{\tau}$, compute $\hat{\boldsymbol{q}}, \hat{\dot{\boldsymbol{q}}}, \hat{\ddot{\boldsymbol{q}}}$, and then construct $\boldsymbol{y}$ and $\boldsymbol{W}_{st}\left(\hat{\boldsymbol{q}}, \hat{\dot{\boldsymbol{q}}}, \hat{\ddot{\boldsymbol{q}}}\right)$. Initialize the PC-IDIM-IV algorithm with a priori values of $\boldsymbol{\beta}_{st}$.

Step it: construct $\boldsymbol{Z}_{st}$ by simulating the DDM with $\hat{\boldsymbol{\beta}}_{st_{IV}}^{it-1}$, being the PC-IDIM-IV estimates calculated at the previous step. Then, compute $\hat{\boldsymbol{\beta}}_{st_{IV}}^{it}$ by minimizing $f_0(\boldsymbol{\beta}_{st})$ defined by Equation (2.9) subject to the constraints given in Section 1.4.4. Run this iterative algorithm until convergence.

Final step: consider $\hat{\boldsymbol{\beta}}_{IV} = [\mathbb{I}_{n_b} \;\; \mathbf{R_t}]\, \hat{\boldsymbol{\beta}}_{st_{IV}}$, and compute the covariance matrix of the PC-IDIM-IV estimates with Equation (2.21).

The PC-IDIM-IV consists of two nested iterative algorithms: an outer one that is the standard IDIM-IV approach and an inner one that accounts for the physical constraints solved by a Gauss-Newton algorithm.

2.3.6 Experimental results

In this section, the validation of the recently presented new PC-IDIM-IV method on the KUKA IIWA manipulator is done. The model and experimental setup are presented and explained in Appendix B and Appendix C. The validation is carried out in three steps:

- First, we carry out an IDIM-LS identification using the *commanded signals* (see Ap-

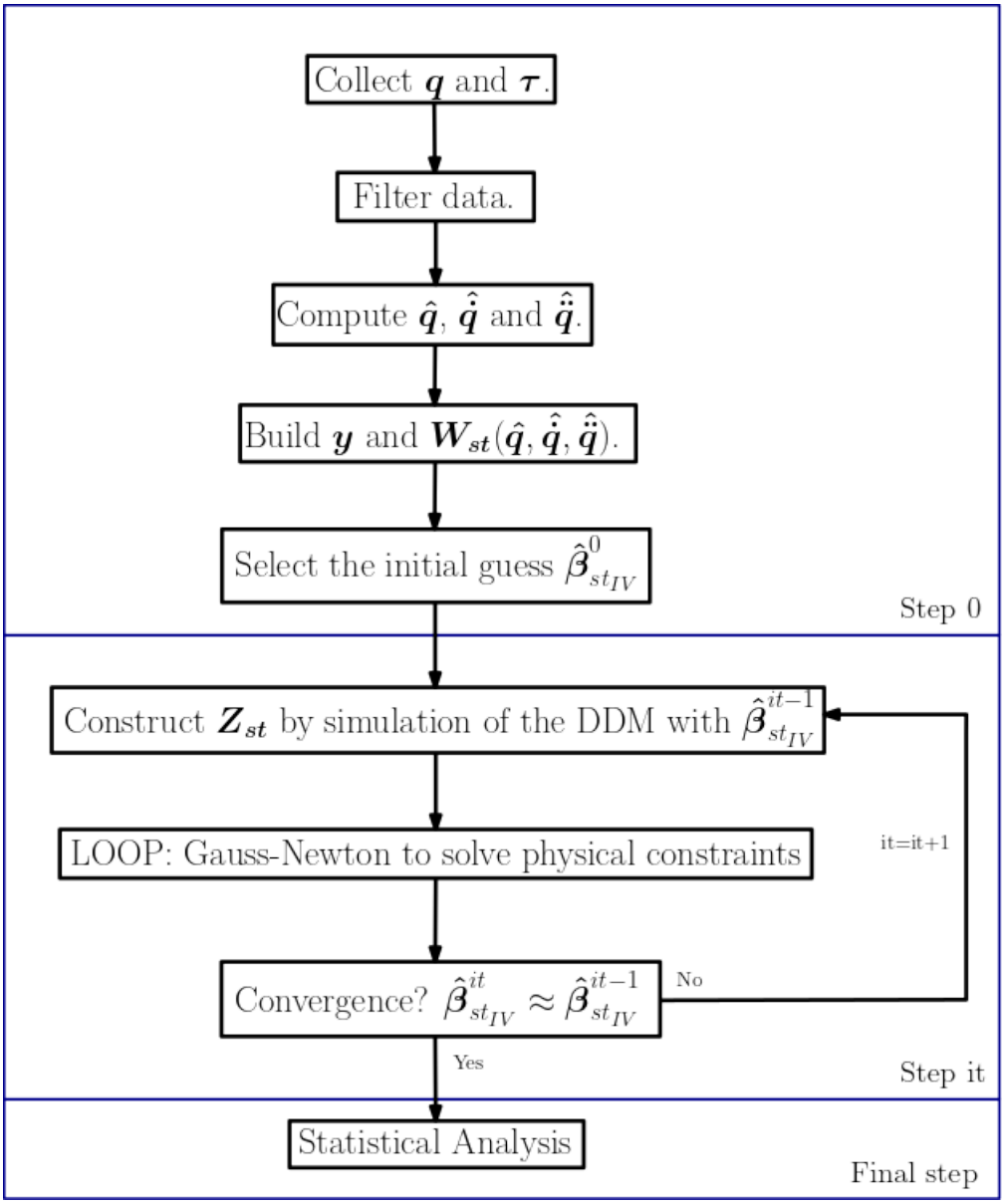


Figure 2.1: PC-IDIM-IV algorithm.

pendix B, specially Figure B.2 for the definition of these signals) in order to identify the manipulator's model that the manufacturer has introduced in their own controller, and verify that these signals are indeed related by this dynamic model. This study, together with the analysis of the measurements done in Appendix B, is made in order to validate the use of these signals to build $\boldsymbol{Z}$ and $\boldsymbol{Z_{st}}$ in following steps.

- Second, we compare the identification results of the real manipulator using the IDIM-LS and IDIM-IV methods to highlight the advantage of using IV based methods.
- Finally, we show the results of carrying out the new PC-IDIM-IV method.

As said in Section 2.2, although in this chapter the methods are validated on just one robotic system, they are general and they can also be applied on other type of systems:

- In Appendix D, we validate the PC-IDIM-IV method on the 6 dof Stäubli TX40 manipulator, and we compare its results with the ones of the PC-IDIM-LS, the IDIM-IV and the PC-DIDIM methods. We show the advantage of using the PC-IDIM-IV method over the others when there is noise and model uncertainties.
- Based on [Ardiani et al., 2021], in Appendix E, we derive a new dynamic model for two-wheeled differential drive mobile robots in which a totally decentralized center of mass is considered, and we compare the IDIM-LS and IDIM-IV methods on simulation showing the potential advantage of IV methods on mobile robots.

2.3.6.1 IDIM-LS to identify manufacturer's model from commanded signals

In this section, the identification of the model that the manufacturer configured inside their industrial controller is carried out. Besides the analysis done of the signals in Appendix B, in order to verify that the *commanded signals* are indeed related by the dynamic model that the manufacturer has given to the controller, an identification of this model is done in this section. It is a reverse engineering procedure, in which the final objective is to find the model that the manufacturer implemented in their model-based controller. For this, we apply the IDIM-LS method to the model of the 64 base parameters (43 dynamic parameters and 21 friction parameters from Equation (1.14)) shown in Appendix Section B.3.1. We use the *Global-PTP_2* trajectory from the *MESSII dataset* (see Appendix C) to identify a set of essential parameters, and then we validate the model by means of the reconstructed torques. It is important to say that no filter was applied. We suppose these signals come from an internal simulation of the dynamic model, thus they are noise free. Just a decimation of factor 20 is applied to reduce computational efforts.

Results. The 14 identified essential parameters are shown in Table 2.1 (parameters with relative standard deviation higher than 20% are not considered as essential). Moreover, Table 2.2 shows the percent error on the torque reconstruction using the training trajectory and

the percent error of the cross-validation with four other trajectories. Furthermore, Figure 2.2 shows the real commanded torque and the reconstructed one for the *Global-SPL_1* trajectory on the seven joints.

Discussion. From the observation of these results, some conclusions can be drawn:

- As no friction parameter was identified, it can be concluded that friction is not included in the manufacturer's model. Its effect is considered as an external perturbation, thus it will be regrouped under the effects of the *external signal*.
- The torque of joint 7 is much lower than those of the other joints. This explains the fact that no parameter corresponding to the seventh link was identified, leading to a percent error of 100%. In order to solve this issue, a regularization of the data could be done previous to the identification [Ljung, 1998].
- The parameters ZZR_i, MYR_i and XXR_i are the predominant inertial parameters. Given the morphology of the links and the way the frames were defined, it can be seen that neither the products of inertias nor the components MX_i will be of great importance in the dynamics of the manipulator.
- The central joints (from the 2^{nd} to the 6^{th}) present an error bounded below 5% which shows a good reconstruction and validates the obtained model.
- The higher relative error in the first joint (joint with vertical axis) can be explained by the fact that it is more difficult to excite it than the others. This is because it presents a big mass, low speeds and accelerations, and no gravity effect. Moreover, it can be seen in the signals that the torque shape is sharper than the others, presenting some peaks and being small during most of the trajectory. These reasons make it's parameters more difficult to be identified in a good way.

Table 2.1: Identified set of essential parameters and their respective relative standard deviation of the manipulator's model integrated in the controller.

Param.	Value	%σ	Param.	Value	%σ
ZZR_1	0.0368	9.21	ZZR_4	0.7590	0.21
XXR_2	2.2346	0.26	MYR_4	2.3156	0.03
ZZR_2	2.2176	0.16	ZZR_5	0.0099	11.15
MYR_2	-5.8227	0.02	MYR_5	-0.0715	0.56
ZZR_3	0.0692	2.71	XXR_6	0.0129	7.53
MYR_3	-0.0164	3.61	ZZR_6	0.0124	7.03
XXR_4	0.7483	0.27	MYR_6	-0.1234	0.34

These results, adding up to the analysis of the signals from Appendix B, lead to think that the controller is indeed model-based and that the set of essential parameters identified in

Table 2.2: Percent error [%] of the *commanded torque* reconstruction using the model in Table 2.1.

Trajectory	Joint						
	1	2	3	4	5	6	7
Global-PTP_2	7.57	1.59	2.31	1.52	3.10	2.66	100.00
Global-PTP_1	21.53	1.93	2.63	1.75	2.90	2.40	100.00
Global-SPL_1	25.52	2.51	3.46	1.46	2.86	2.18	100.00
Global-SPL_2	3.84	1.69	1.68	2.15	4.75	4.87	100.00
Global-DS	22.06	2.46	2.74	2.31	4.23	2.81	100.00

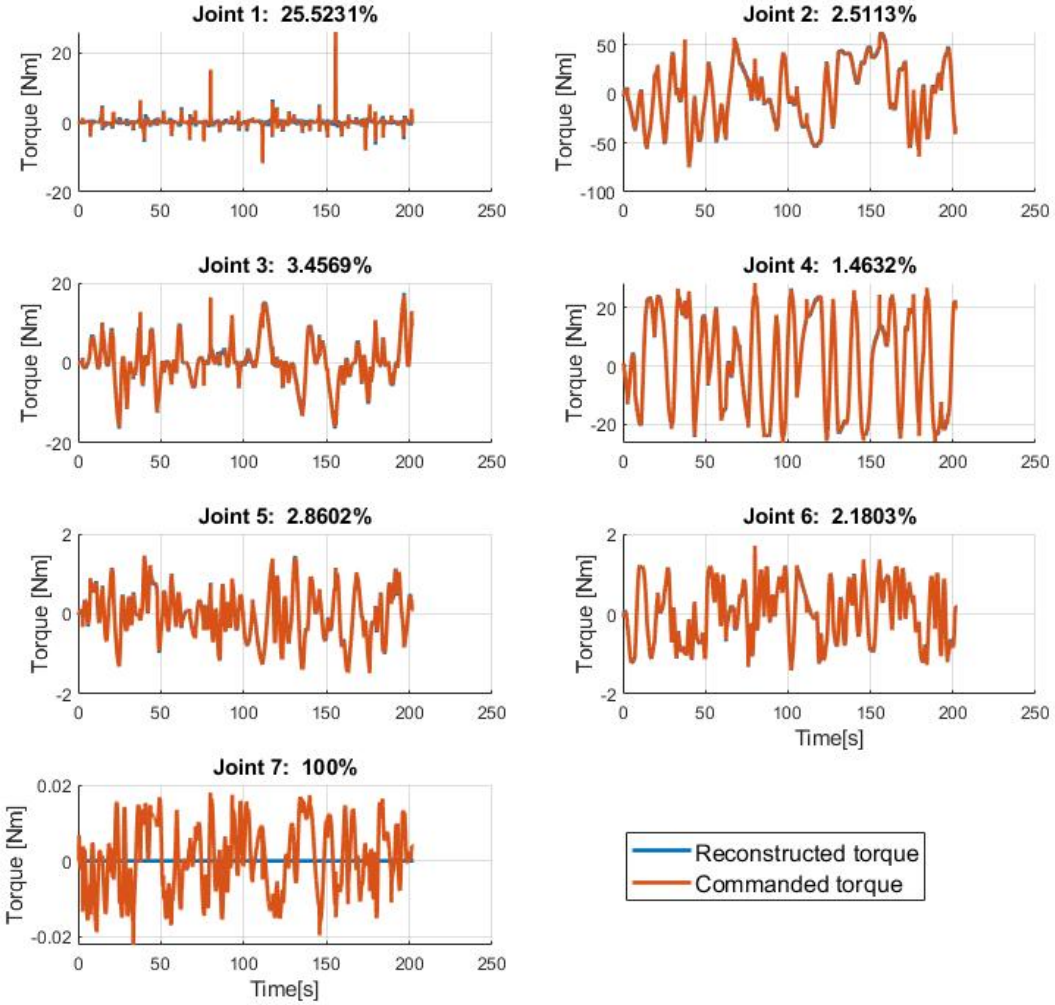


Figure 2.2: Reconstructed and real torques for the *Global-SPL_1* trajectory and the identified model in Table 2.1.

this section corresponds to the model that the manufacturer included inside their controller. This also validates the fact that the *commanded signals* come from their simulation and are related through this identified dynamic model.

The robot's model integrated in the controller can be used as a tool to check that the parameters obtained in the identification using *sensed signals* (see next section) are close to the manufacturer's ones, and as a priori knowledge for identification methods where an initial guess is needed. Moreover, the set of essential parameters obtained using measured data may not be complete and may not allow the inversion of the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ for the DDM. The parameters identified using *commanded signals* can be used to complete the set.

2.3.6.2 IDIM-LS vs IDIM-IV

In this section, we will compare the results of identifying the dynamic parameters of the KUKA iiwa manipulator using:

- the IDIM-LS method with filtered data,
- the IDIM-IV method with filtered data,
- the IDIM-IV method without filtering, called NF-IDIM-IV, where NF stands for non-filtered signals.

We will compare these results between each other and with the state-of-the-art models of the KUKA iiwa manipulator available in literature ([Hennersperger et al., 2016, Stürz et al., 2017, Xu et al., 2020]). The instrumental matrix for the IDIM-IV method is built by evaluating the regressor matrix with the *commanded position* signals and its derivatives. In Section 2.3.6.1, it has been shown that the *commanded position* and the *commanded torque* signals are indeed related via an internal model that the controller has of the manipulator, which is a priori used for the model-based control law and to compute the external forces/perturbations. They are product of a simulation of the model of the manipulator, which makes them a valid instrument for the IV methods. However, due to the fact that we cannot modify the parameters of the simulation (the manufacturer does not allow to change the values of the parameters of the model that the controller has of the robot), the non-iterative IV method is used.

The chosen model in this section is the one shown in Appendix B, together with the 3-parameters friction model from Equation (1.14). The experimental setup and filtering process are also explained in Appendix B and the trajectories generation in Appendix C. In this section, we will consider filtered and non-filtered (NF) signals. The NF signals are the raw signals with a downsample process applied to reduce computational effort. The trajectory *Global-PTP_2* is used for identification, and the trajectories *Global-DS* and *Global-SPL_1* from the *MESSII data-set* are used for validation.

Results. Table 2.3 shows the set of essential parameters values with their respective relative standard deviation, result of three different identification methods. In the 2^{nd} and 3^{rd} columns, the results of applying the IDIM-LS method with filtered signals are shown; in the 4^{th} and 5^{th} columns the ones related to the IDIM-IV method also with filtered signals are depicted; and, finally, in the last two columns the results of applying the IDIM-IV method without filtering the signals, called the NF-IDIM-IV method, are displayed. From the $n_b = 64$ base parameters (obtained from $n_{st} = 91$ standard parameters), 33 essential parameters are identified using the IDIM-LS method, 29 using the IDIM-IV method with filtered signals and 13 using the NF-IDIM-IV method.

On the other hand, Table 2.4 and Figure 2.3 show the respective results of the torque reconstruction using the sensed signals and the set of essential parameters obtained in Table 2.3. They also depict the torque reconstruction results using the set of parameters in Stürz [Stürz et al., 2017] and iiwa Stack [Hennersperger et al., 2016] for comparison and validation purposes.

Discussion. Numerical results in Table 2.3 show that, in general, viscous friction f_{v_i} has less effect on the model than static friction f_{c_i} and torque offset τ_{off_i}. Just one parameter corresponding to the viscous friction (f_{v_i}) is identified, whereas almost all the parameters of each of the other two effects (f_{c_i} and τ_{off_i}) are identified, in both of the models that used filtered signals. Particularly, it is interesting to notice that for the NF-IDIM-IV method, links 1 and 7 can be accurately approximated just by knowing their friction model. For link 7, it is due to the small values of mass and inertia. For link 1, it may happen for two reasons: that it is under-excited, as the accelerations are limited by the manufacturer for safety reasons, and/or, due to the fact that the value of inertia ZZR_1 is not affected by gravity, its value is relatively small even if it is the biggest link in size.

Not all main inertial parameters are identified for every link. The standard deviations and estimated values in Table 2.3 show that the parameters that have more influence on the behavior of the robot are MYR_i, XXR_i and ZZR_i from links that deal with higher gravity effects (2, 4 and 6). Moreover, it can be seen that not all the parameters are physically consistent, specially those from the friction model. This is highly dependant on the chosen trajectory and noise, and becomes a "lucky" activity to find a physically consistent set of parameters if no constraints are specified. This fact emphasises the need to use an optimization that ensures the physical consistency, and that is why in the next section, results from the PC-IDIM-IV method will be shown.

Moreover, Table 2.4 shows that the three models of Table 2.3 have an error of less than 30% between the *sensed torque* and the reconstructed torque for the first 6 joints, being a little higher for the last joint in the second validation trajectory. As well as in the previous section, the error is lower in central joints where the inertia and gravity effects are higher, being less than 4% for joints 2, 3 and 4.

The NF-IDIM-IV model has two advantages: similar results are obtained without filtering the signals, which is quite convenient for a possible online application not needing the filtering

Table 2.3: Identified essential parameters and their respective relative standard deviation with the IDIM-LS, IDIM-IV and NF-IDIM-IV methods.

Param.	IDIM-LS		IDIM-IV		NF-IDIM-IV	
	Value	$\%\sigma$	Value	$\%\sigma$	Value	$\%\sigma$
ZZR_1	0.0759	5.45	0.0509	6.19	-	-
f_{c_1}	0.9970	0.54	0.9782	0.43	1.0468	5.70
τ_{off_1}	0.2804	1.93	0.2854	1.47	-	-
XXR_2	2.3262	0.31	2.2492	0.24	2.4262	2.71
ZZR_2	2.5995	0.18	2.4129	0.14	2.4358	2.93
YZ_2	0.0365	8.75	0.0505	4.62	-	-
MYR_2	-5.8921	0.03	-5.9126	0.02	-5.9396	0.45
f_{c_2}	1.2275	0.47	1.2195	0.75	1.0957	10.40
f_{v_2}	-	-	-0.1658	7.36	-	-
τ_{off_2}	-0.3666	1.73	-0.3607	1.38	-	-
XZ_3	0.0131	14.29	-	-	-	-
YZ_3	-0.0147	12.50	-	-	-	-
MX_3	0.0156	6.47	0.0122	5.18	-	-
MYR_3	-0.0358	2.50	-0.0332	1.79	-	-
f_{c_3}	0.0947	10.73	0.1432	2.97	-	-
f_{v_3}	0.0576	16.38	-	-	-	-
XXR_4	0.8324	0.28	0.8151	0.22	0.8357	2.07
ZZR_4	0.8531	0.31	0.8209	0.25	0.8615	3.35
MYR_4	2.3721	0.03	2.3724	0.03	2.3644	0.42
f_{c_4}	0.1426	3.82	0.1131	3.76	-	-
τ_{off_4}	0.1167	4.75	0.1186	3.64	-	-
XXR_5	0.0191	11.58	0.0205	8.22	-	-
ZZR_5	0.0210	6.27	0.0216	4.58	-	-
XZ_5	0.0055	15.71	-	-	-	-
YZ_5	-0.0061	17.44	-0.0062	12.99	-	-
MYR_5	-0.0766	0.76	-0.0756	0.60	-0.0878	7.21
ZZR_6	0.0130	11.68	0.0177	6.13	-	-
MYR_6	-0.1164	0.45	-0.1189	0.33	-0.1306	4.82
f_{c_6}	0.3618	1.51	0.3626	1.18	0.3748	2.07
τ_{off_6}	-0.1937	2.83	-0.1937	2.21	-0.1907	3.50
XXR_7	0.0077	12.60	-	-	-	-
MX_7	-0.0035	11.99	-0.0034	9.36	-	-
f_{c_7}	0.2324	2.37	0.2328	1.84	0.2207	0.88
τ_{off_7}	0.0618	8.90	0.0609	7.04	-	-
Other non-essential parameters: f_{v_1}; XY_2; XZ_2; MX_2; XXR_3; ZZR_3; XY_3; τ_{off_3}; XY_4; XZ_4; YZ_4; MX_4; f_{v_4}; XY_5; MX_5; f_{c_5}; f_{v_5}; τ_{off_5}; XXR_6; XY_6; XZ_6; YZ_6; MX_6; f_{v_6}; ZZ_7; XY_7; XZ_7; YZ_7; MY_7; f_{v_7}.						

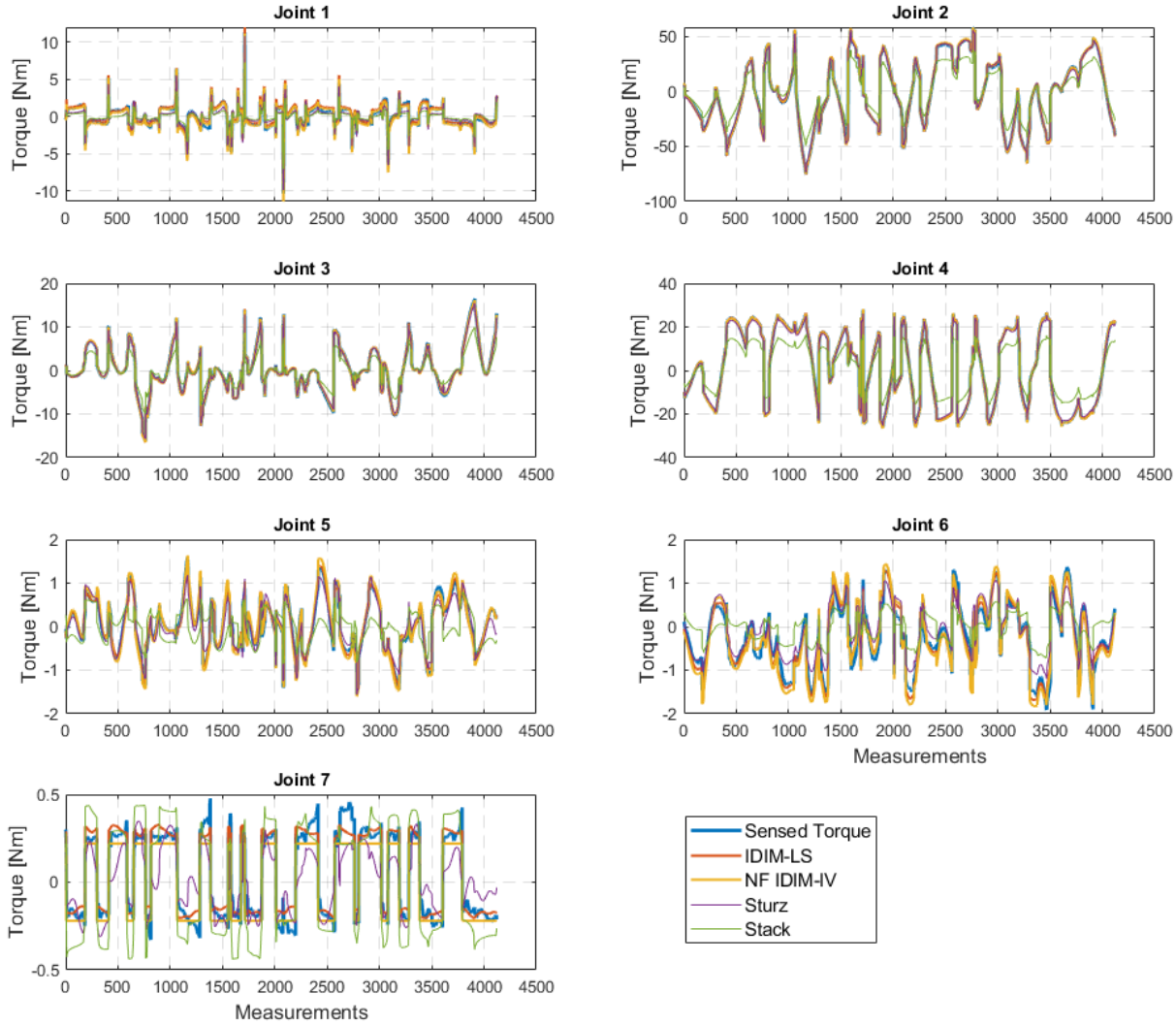


Figure 2.3: Torque reconstruction using models from Table 2.3, [Stürz et al., 2017] and [Hennersperger et al., 2016] and the *Global-SPL_1* trajectory.

Table 2.4: Percent error [%] of torque reconstruction of models from Table 2.3, and the ones derived in [Stürz et al., 2017] and [Hennersperger et al., 2016].

Joint	1	2	3	4	5	6	7
Global-SPL_1							
IDIM-LS	22.65	2.47	3.45	0.78	12.23	21.31	24.39
IDIM-IV	21.95	2.57	3.42	0.78	11.95	22.07	24.26
NF-IDIM-IV	23.45	3.33	4.05	1.34	15.74	26.74	28.86
[Stürz et al., 2017]	42.04	5.22	6.52	4.63	27.92	38.33	55.27
[Hennersperger et al., 2016]	56.51	34.46	40.27	39.41	88.88	69.65	55.84
Global-DS							
IDIM-LS	29.62	1.71	2.65	1.08	17.59	17.46	35.24
IDIM-IV	28.90	1.76	2.64	1.13	17.10	18.19	34.99
NF-IDIM-IV	28.80	3.12	3.84	1.85	17.00	22.76	54.66
[Stürz et al., 2017]	39.87	3.11	6.37	4.46	33.30	32.54	71.59
[Hennersperger et al., 2016]	55.41	33.69	41.25	39.40	72.31	63.53	118.21

stage, and it presents similar performance with a considerable less amount of parameters. It shows that with just 13 parameters, it is possible to predict the robot's movement quite accurately. This is also shown in Figure 2.3, where it can be graphically seen that there is no significant difference between the three methods compared in this section.

Furthermore, in Figure 2.3 and Table 2.4, it is shown that the torque reconstruction using our identification results is better than those carried out from models in bibliography [Stürz et al., 2017] and [Hennersperger et al., 2016]. The difference may be due to a different experimental setup. For a thorough description of the experimental setup used in our work, see Appendix B and Appendix C. In [Stürz et al., 2017], the authors do not specify whether a flange is being used or not. In [Hennersperger et al., 2016], the authors do specify that a flange is being used, but not which one. This is also the reason why the results of [Xu et al., 2020] are not shown. There, besides not specifying the flange used, the authors use a not-specified significant load on the tip. Although they explain that the contribution of the payload could be removed, they do not detail the load values. That is why to allow the reproducibility of the present work and fair future comparisons, the experimental setup is thoroughly explained. Besides, it is known that numerical values, especially the ones of friction parameters, are not exactly the same from one robot manipulator to another, and that may be the reason why our results give slighlty better results than those of [Stürz et al., 2017]. To support this statement, in Appendix F, we compare the identification of three KUKA iiwa manipulators, and we show, that even tough the experimental setup is the same for the three robots, the identified model is slightly different. It is not possible to assure that all of the manipulators have the same characteristics, even though each link should have the same inertial values in all of the manipulators. The assembly process probably causes the difference, not being exactly the same for each of the manipulators. However, choosing a more complete friction model than those in [Stürz et al., 2017], [Xu et al., 2020] and [Hennersperger et al., 2016]

leads to better results in expense of having 7 more parameters (τ_{off_i}, with $i = 1, \ldots, 7$) to identify.

Moreover, the performance indicator used in [Stürz et al., 2017] and [Xu et al., 2020] for cross-validation (RMS) is directly dependent on the amplitude of the trajectory not allowing a fair comparison of results. To overcome this issue, in this work, the percent error (relative indicator with respect to the amplitude of the trajectory) was chosen.

2.3.6.3 PC-IDIM-IV

Now, we will analyze the results of identifying the model with the new method developed in this chapter: the PC-IDIM-IV method. In the same way as in the previous two subsections, the validation is done on the KUKA iiwa manipulator presented in Appendix B. The model is completed with the friction model of Equation (1.14). The filtering stage and the trajectories used in this subsection are explained in Appendix B and Appendix C, respectively.

Moreover, the comparison of the results of the PC-IDIM-IV and the PC-IDIM-LS (see Section 1.4.4) is done with filtered and non-filtered data. It seems more interesting to analyze the performance of the method against other PC methods, and not with methods which do not require an optimization regarding the physical feasibility of the parameters. For example, it is logical to think that the torque reconstruction of the estimated model obtained with the IDIM-IV is equal or better than the one of the PC-IDIM-IV. Indeed, the PC adds constraints limiting the possible solutions to a set of physically feasible parameters.

In Appendix D, the validation is also carried out on the 6-*dof* Stäubli manipulator TX-40, showing the generalization of the method to other type of robots. There, we also compare the PC-IDIM-IV method's performance with the PC-IDIM-LS, the IDIM-IV and the PC-DIDIM methods.

Results. Table 2.5 shows the set of standard parameters identified with the PC-IDIM-IV method, and Table 2.6 shows its respective transformation to the set of base parameters. Here, 0.0000 means that the identified value is positive and smaller than $1e^{-4}$ and -0.0000 means that the value is negative and bigger than $-1e^{-4}$.

Moreover, Table 2.7 shows the percent error of the torque reconstruction of the model obtained via the PC-IDIM-IV method showed in the previous tables over 2 trajectories: *Global-SPL_1* and *Global-DS*. Here, we also show the validation of the models obtained with the PC-IDIM-LS method, and with both methods but without filtering, called NF-PC-IDIM-IV and NF-PC-IDIM-LS, respectively.

Discussion. Numerical results from Table 2.5 show that the identified set of standard parameters is indeed physically feasible, as well as the friction parameters. However, it can be seen that there are some parameters that are hardly possible to be near the true value (e.g.

Table 2.5: Set of standard parameters identified with the PC-IDIM-IV method.

Param.	Value	Param.	Value	Param.	Value	Param.	Value
ZZ_1	0.0000	YY_3	13.9613	τ_{off_4}	0.1100	ZZ_6	0.0097
f_{c_1}	0.9774	ZZ_3	0.0005	M_5	5.0604	XY_6	0.0016
f_{v_1}	0.0000	XY_3	0.0062	MX_5	0.0004	XZ_6	-0.0011
τ_{off_1}	0.2853	XZ_3	0.0087	MY_5	-0.0862	YZ_6	0.0001
MX_2	0.0072	YZ_3	-0.0000	MZ_5	-0.4907	f_{c_6}	0.3617
MY_2	-0.2459	f_{c_3}	0.1035	XX_5	0.0659	f_{v_6}	0.0000
XX_2	0.1669	f_{v_3}	0.0443	YY_5	0.0591	τ_{off_6}	-0.1941
YY_2	0.0643	τ_{off_3}	0.0059	ZZ_5	0.0118	M_7	1.1130
ZZ_2	0.2291	M_4	0.2188	XY_5	-0.0009	MX_7	-0.0032
XY_2	-0.0079	MX_4	0.0012	XZ_5	0.0043	MY_7	0.0008
XZ_2	0.0081	MY_4	-0.0004	YZ_5	-0.0045	MZ_7	0.0347
YZ_2	0.0485	MZ_4	0.0011	f_{c_5}	0.0203	XX_7	0.0088
f_{c_2}	1.1113	XX_4	0.0044	f_{v_5}	0.0233	YY_7	0.0077
f_{v_2}	0.0000	YY_4	0.0065	τ_{off_5}	-0.0203	ZZ_7	0.0013
τ_{off_2}	-0.3147	ZZ_4	0.0083	M_6	0.9919	XY_7	-0.0000
M_3	86.4409	XY_4	-0.0020	MX_6	-0.0008	XZ_7	-0.0007
MX_3	0.0142	XZ_4	-0.0026	MY_6	-0.0818	YZ_7	-0.0002
MY_3	-0.0348	YZ_4	0.0039	MZ_6	0.0100	f_{c_7}	0.2318
MZ_3	-33.7425	f_{c_4}	0.1100	XX_6	0.0094	f_{v_7}	0.0000
XX_3	13.9618	f_{v_4}	0.0099	YY_6	0.0026	τ_{off_7}	0.0601

Table 2.6: Set of base parameters identified with the PC-IDIM-IV method equivalent to Table 2.5.

Param.	Value	Param.	Value	Param.	Value	Param.	Value
ZZR_1	0.0643	XY_3	0.0062	f_{v_4}	0.0099	YZ_6	0.0001
f_{c_1}	0.9774	XZ_3	0.0087	τ_{off_4}	0.1100	MX_6	-0.0008
f_{v_1}	0.0000	YZ_3	-0.0000	XXR_5	0.0094	MYR_6	-0.1166
τ_{off_1}	0.2853	MX_3	0.0142	ZZR_5	0.0145	f_{c_6}	0.3617
XXR_2	2.2711	MYR_3	-0.0359	XY_5	-0.0009	f_{v_6}	0.0000
ZZR_2	2.3975	f_{c_3}	0.1030	XZ_5	0.0043	τ_{off_6}	-0.1941
XY_2	-0.0079	f_{v_3}	0.0443	YZ_5	-0.0045	XXR_7	0.0010
XZ_2	0.0081	τ_{off_3}	0.0059	MX_5	0.0004	ZZ_7	0.0013
YZ_2	0.0485	XXR_4	0.8110	MYR_5	-0.0762	XY_7	-0.0000
MX_2	0.0072	ZZR_4	0.8213	f_{c_5}	0.0203	XZ_7	-0.0007
MYR_2	-5.9099	XY_4	-0.0020	f_{v_5}	0.0233	YZ_7	-0.0002
f_{c_2}	1.1113	XZ_4	-0.0026	τ_{off_5}	-0.0203	MX_7	-0.0032
f_{v_2}	0.0000	YZ_4	0.0039	XXR_6	0.0145	MY_7	0.0008
τ_{off_2}	-0.3147	MX_4	0.0012	ZZR_6	0.0175	f_{c_7}	0.2318
XXR_3	0.0070	MYR_4	2.3750	XY_6	0.0016	f_{v_7}	0.0000
ZZR_3	0.0070	f_{c_4}	0.1100	XZ_6	-0.0011	τ_{off_7}	0.0601

Table 2.7: Percent error [%] of torque reconstruction of models identified with the PC-IDIM-LS, PC-IDIM-IV, NF-PC-IDIM-LS and NF-PC-IDIM-IV methods.

Joint	1	2	3	4	5	6	7
Global-SPL_1							
PC-IDIM-IV	22.14	2.71	3.40	0.79	10.55	21.45	23.95
PC-IDIM-LS	22.16	2.71	3.40	0.79	10.53	21.46	23.93
NF-PC-IDIM-IV	22.82	2.60	3.56	0.92	10.51	23.57	26.91
NF-PC-IDIM-LS	30.22	3.38	6.80	3.37	45.53	48.42	50.26
Global-DS							
PC-IDIM-IV	29.24	1.68	2.64	1.16	11.88	17.75	34.63
PC-IDIM-LS	29.27	1.68	2.64	1.15	11.83	17.77	34.61
NF-PC-IDIM-IV	30.89	1.34	2.58	1.10	13.88	20.28	39.33
NF-PC-IDIM-LS	36.43	1.65	5.68	4.40	40.77	41.86	50.88

$M_3 = 86.4409$ kg). This is indeed because the optimization problem just searches the feasibility of the solution, without any a priori knowledge. This means that it searches for the global minimum between all the possible physically feasible solutions, which may not correspond to the real values due to external effects as noise, uncertainties and level of excitation. Other works as [Stürz et al., 2017] and [Xu et al., 2020] include constraints regarding an apriori range for the parameters, reducing the possible solutions. This is a simple way to find standard parameters which are closer to reality. However, this may not change the estimation of the set of base parameters of Table 2.6, thus of the torque reconstruction result. In fact, there are infinite set of standard parameters that correspond to the same set of base parameters, and each of these sets will have the same performance and torque reconstruction. Furthermore, the results in Table 2.6 can be compared with the ones in Table 2.3. It can be noticed that the estimates using the PC-IDIM-IV and the IDIM-IV methods are very similar.

Moreover, Table 2.7 shows the performance of the methods. The PC-IDIM-IV does not present any clear advantage over the PC-IDIM-LS in terms of performance when the data is correctly filtered. However, when data is not filtered in a good way, the PC-IDIM-IV method largely outperforms its LS version. This is the most attractive characteristic of the derived method and presents good perspectives for further applications. Its performance when data is filtered or not, does not varies, which is a characteristic of the IV methods in general.

2.3.7 Conclusion

In this section, we derived the new PC-IDIM-IV method. It is a method which yields physically consistent estimates using the IV theory. It is usually preferable against the PC-IDIM-LS method because it is more robust against noise than the later. We proposed an algorithm to solve the problem, which can be of interest to the robotics community.

2.4 Friction identification

In this section, we are going to tackle the issue of friction identification in robotic manipulators. Friction identification and its proper compensation has an important role for control of manipulators. A good compensation of it, improves the transient performance, reduces the steady-state tracking error, and ensures a smooth control signal with low feedback gains [Kermani et al., 2007].

The difficulty of this process is that the friction force has various forms, which are not yet fully understood, that may change during run-time and that will depend on several other conditions and features. Some of the models of friction usually used in robotics were presented in Section 1.2.2, and its identification and compensation is widely treated in the works therein cited.

In this section, we will compare two methods to identify a non-linear friction model (see Equation (1.15)). Taking advantage of the knowledge of the system and of the manipulator's model included in the controller, Figure 2.4 shows the two procedures that will be used to identify the following models:

- *Model A*: it is obtained via a global approach, where the friction and dynamic parameters are identified from one trajectory using position and torque measurements. In order to make the friction model in Equation (1.15) linear with respect to the parameters and allow to get a solution with the LS method, the parameters K_v, δ_i and α_{f_i} need to be fixed to a certain value. We applied an approach similar to the one presented in [Indri et al., 2016, Indri and Trapani, 2020], where K_v and α_{f_i} were fixed, and δ_i was varying inside a range, searching for the value that minimizes the error between the experimental and the reconstructed data.

- *Model B*: it is obtained from two sources. On the one hand, the dynamic parameters are supposed to be available or to be identified from the internal model that the controller has of the robot, as shown in the previous section. On the other hand, the friction parameters are obtained via a joint-by-joint sequential approach, in which they are identified from the *external torque* of trajectories that move one joint at a time. Although it is a sequential identification, the error propagation is insignificant because the friction parameters identified of one joint will not affect the identification of the friction parameters of the other joints.

The method mentioned in *Model A*, although effective as shown in [Indri and Trapani, 2020], is a naive search of the optimal solution as it will depend on the selected possible values of the parameters. Because of that, to obtain *Model B*, we used the Separable Least-Squares (SLS) method, which will be explained in the following subsection. It consists of a two step process, one in which the optimization of the non-linear parameters δ_i and α_{f_i} is made based on the projection of the column space of the regressor, and another in which a IDIM-LS solution is obtained, considering the other parameters as known.

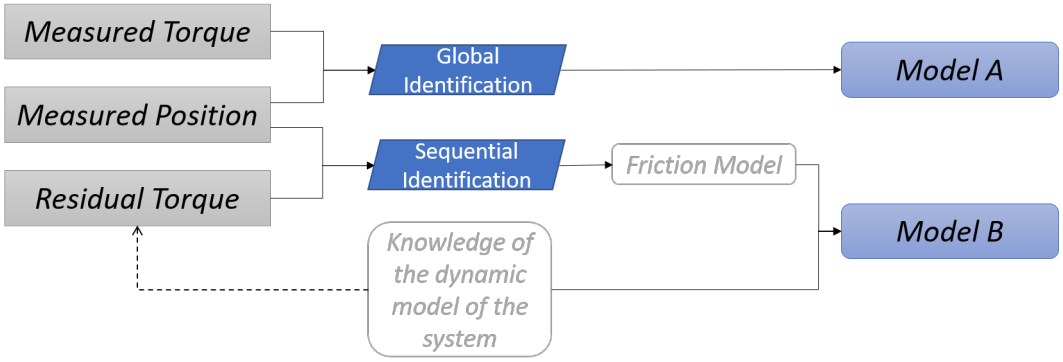


Figure 2.4: Two friction identification procedures.

This section starts by explaining the IDIM-SLS method, and it continues with the experimental validation and comparison of the two mentioned methodologies.

2.4.1 The IDIM-SLS method

The Separable Least-Squares estimation (SLS), or variable projection method, is a technique to be applied when the model is linear with respect to some parameters and nonlinear with respect to others. It is an alternative to the application of nonlinear optimization techniques, and it has been proven to be more efficient and accurate in several applications [Bruls et al., 1999]. It has been successfully applied to robotics in [Wernholt and Gunnarsson, 2006, Hashemi and Werner, 2009], and will be presented in this section. See [Golub and Pereyra, 1973, Golub and Pereyra, 2003] for a thorough treatment of the theory behind the method.

If we consider a vector $\boldsymbol{\alpha}$ of n_{nl} amount of parameters whose relation with respect to the independent variables is non-linear, the whole system of Equation (1.26) could be considered as non-linear with respect to the parameters to be identified:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})\boldsymbol{\beta} + \boldsymbol{\rho}. \tag{2.23}$$

The idea of the method is to divide the solution in two steps: one in which a reduced nonlinear iterative minimization with respect to the new vector $\boldsymbol{\alpha}$ is done, and another one where the simple LS solution is done to obtain $\boldsymbol{\beta}$ considering $\boldsymbol{\alpha}$ as known.

If we note the Moore-Penrose generalized inverse, thus the solution to the LS problem, as $\boldsymbol{W}^{+}$, we get that:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})^{+}\boldsymbol{y}(\boldsymbol{\tau}). \tag{2.24}$$

If we replace this value on Equation (2.23), then we get that the error can be expressed as:

$$\hat{\boldsymbol{\rho}} = \left[\mathbb{I}_r - \boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})\boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})^{+}\right]\boldsymbol{y}(\boldsymbol{\tau}). \tag{2.25}$$

Then, by minimizing the LS functional of the error, we can express:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}||\hat{\boldsymbol{\rho}}||^2 = \min_{\boldsymbol{\alpha}} \frac{1}{2} \left|\left| \left[\mathbb{I}_r - \boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}}) \boldsymbol{W}(\boldsymbol{\alpha}, \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}})^+ \right] \boldsymbol{y}(\boldsymbol{\tau}) \right|\right|^2 . \tag{2.26}$$

In this expression, as $\hat{\mathbf{q}}$, $\hat{\dot{\mathbf{q}}}$, $\hat{\ddot{\mathbf{q}}}$ and $\boldsymbol{\tau}$ are known, $\boldsymbol{\alpha}$ is the only unknown. Then, a non-linear optimization is carried out on this last equation, with for example the SQP method. Finally, after the optimized $\boldsymbol{\alpha}$ is obtained, Equation (2.24) can be solved to obtain the solution of $\hat{\boldsymbol{\beta}}$.

This method presents a main advantage with respect to applying a non-linear optimization to the whole system: the iterative nonlinear algorithm used for the first step of the SLS method works in a reduced space. This means that it requires less initial guesses, less time and less computational effort, reducing the possibilities of failing to find a solution.

2.4.2 Experimental comparison

The KUKA iiwa manipulator was used to compare the mentioned friction identification methods. Both, the robot's model and the experimental setup are mentioned in Appendix B. The chosen non-linear friction model is the one of Equation (1.15). For *Model A* a global trajectory is used, while for *Model B*, sequential trajectories are designed [Hao et al., 2021]. These sequential trajectories are composed by a single joint moving at a time in all its position range and at different constant velocities. An example is shown in Figure 2.5.

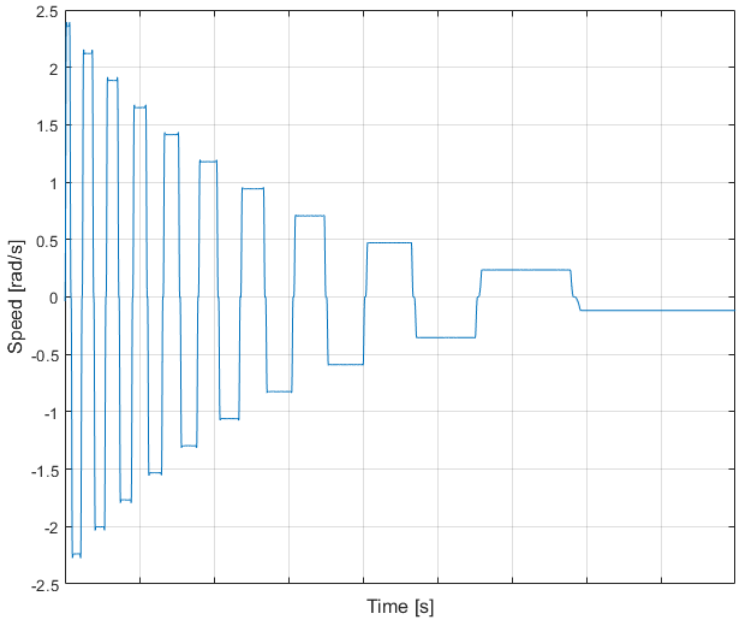


Figure 2.5: Friction identification trajectory.

In [Indri et al., 2016, Indri and Trapani, 2020], the authors suggested to fix K_v to 1000,

Table 2.8: Identified essential inertial parameters and their respective relative standard deviation.

Param.	*Model A*		*Model B*		Param.	*Model A*		*Model B*	
	Value	%σ	Value	%σ		Value	%σ	Value	%σ
ZZR_1	0.04	8.12	0.04	2.64	MX_4	0.01	5.87	-0.00	6.31
XXR_2	2.32	0.24	2.28	0.15	MYR_4	2.37	0.02	2.31	0.00
ZZR_2	2.50	0.14	2.31	0.11	XXR_5	-	-	0.01	16.04
YZ_2	0.04	6.39	-	-	ZZR_5	0.02	5.23	0.01	4.88
MX_2	0.03	8.78	-	-	YZ_5	-0.01	12.99	-0.00	6.27
MYR_2	-5.86	0.02	-5.81	0.00	MYR_5	-0.08	0.58	-0.07	0.08
ZZR_3	-	-	0.07	1.23	XXR_6	-	-	0.01	3.78
XZ_3	0.01	9.73	-0.01	9.05	ZZR_6	0.01	10.23	0.02	2.75
YZ_3	-0.01	12.24	-	-	MX_6	-	-	-0.00	5.22
MX_3	0.01	7.29	-0.00	9.40	MYR_6	-0.12	0.32	-0.12	0.04
MYR_3	-0.04	1.79	-0.02	0.47	ZZ_7	-0.01	8.59	0.00	16.02
XXR_4	0.83	0.17	0.76	0.13	XZ_7	-0.00	18.89	-	-
ZZR_4	0.84	0.19	0.77	0.11	MX_7	-0.00	7.39	0.00	2.67
YZ_4	-	-	-0.01	7.17					
Other non-essential parameters: XY_2, XZ_2, XXR_3, XY_3, XY_4, XZ_4,XY_5, XZ_5, MX_5, XY_6, XZ_6,YZ_6, XXR_7, XY_7, YZ_7, MY_7									

α_{f_i} to 2 and to vary δ_i from 0 to 100 with a step of 0.1 searching for the value that minimizes the error between the experimental and the reconstructed data. The found values of δ_i in [Indri et al., 2016] were almost all lower than 10. In this work, for *Model A* a similar approach is adopted. K_v is fixed to 100 as it seems a good choice for the approximation of the *sign* function. In a first step, for each joint, α_{f_i} is fixed to 2 and δ_i varies from 0 to 10 with a step of 0.001. Once found the best δ_i, as it may occur that the exponential part of the nonlinear viscous friction term is not exactly 2, α_{f_i} will vary from 1.5 to 2.5 with an step of 0.001. For this model, α_{f_i} and δ_i are the same for all joints as a global approach is applied.

Results. Table 2.8 shows the set of identified dynamic essential parameters for both models, and Table 2.9 shows the identified essential friction parameters. Moreover, Figure 2.6 shows the *external torque* reconstruction of the training trajectory of the seven joints and *Model B*, together with the respective percent error. Finally, Table 2.10 depicts the percent error of the torque reconstruction of the sensed torque of both models during two global trajectories (seven joints moving at the same time).

Discussion. From the 43 base inertial parameters and 35 friction parameters, *Model A* consists of 22 inertial and 23 friction essential parameters, whereas, *Model B* of 23 inertial and 35 friction essential parameters. Most of the predominant inertial parameters (ZZR_i and MYR_i) are identified and are similar in both models, what was already shown and verified in

Table 2.9: Identified essential friction parameters and their respective relative standard deviation.

Param.	*Model A*		*Model B*		Param.	*Model A*		*Model B*	
	Value	%σ	Value	%σ		Value	%σ	Value	%σ
f_{c_1}	0.1770	15.85	1.4894	1.02	f_{c_5}	-	-	-	-
f_{sc_1}	0.8400	5.17	-0.6213	5.60	f_{sc_5}	-	-	-	-
f_{v_1}	-1.3224	10.53	0.2727	13.48	f_{v_5}	-	-	0.0528	7.54
τ_{off_1}	0.1286	3.25	0.0425	12.50	τ_{off_5}	0.0224	18.56	0.0139	5.29
f_{nlv_1}	0.9278	10.09	-	-	f_{nlv_5}	-	-	-0.0263	12.41
δ_1	10		3.4510		δ_5	-		-	
α_{f_1}	1.5		-		α_{f_5}	-		1.5000	
f_{c_2}	2.1200	0.99	2.3349	1.62	f_{c_6}	0.3912	1.15	0.4683	0.83
f_{sc_2}	-1.2318	3.11	-1.0632	3.38	f_{sc_6}	-	-	-0.1258	5.90
f_{v_2}	1.4631	11.06	-	-	f_{v_6}	-	-	-	-
τ_{off_2}	-0.7297	2.76	-0.9631	1.32	τ_{off_6}	-0.2162	1.94	-0.1821	1.235
f_{nlv_2}	-0.6156	19.47	-	-	f_{nlv_6}	-	-	-	-
δ_2	10		-		δ_6	-		0.8129	
α_{f_2}	1.5		-		α_{f_6}	-		-	
f_{c_3}	0.1498	3.02	0.1095	1.95	f_{c_7}	0.2561	1.76	0.2762	1.22
f_{sc_3}	-	-	-0.0264	8.57	f_{sc_7}	-	-	-1.4620	8.22
f_{v_3}	-	-	-	-	f_{v_7}	-	-	1.8715	9.14
τ_{off_3}	0.1397	3.01	0.0986	0.87	τ_{off_7}	0.0315	13.46	0.0722	0.87
f_{nlv_3}	-	-	0.0000	19.84	f_{nlv_7}	-	-	-0.7597	9.52
δ_3	-		3.7254		δ_7	-		1.0381	
α_{f_3}	-		35.1636		α_{f_7}	-		1.5000	
f_{c_4}	0.1695	2.72	0.3758	5.67					
f_{sc_4}	-	-	-0.1665	9.20					
f_{v_4}	-	-	-	-					
τ_{off_4}	0.3774	1.46	0.3833	1.03					
f_{nlv_4}	-	-	-0.0000	16.99					
δ_4	-		16.0014						
α_{f_4}	-		32.4732						

Table 2.10: Percent error [%] of the sensed torque reconstruction of global trajectories for Model A and Model B.

Joint	1	2	3	4	5	6	7
Validation Trajectory 1							
Model A	5.91	1.81	2.76	2.44	13.95	26.74	46.94
Model B	7.90	3.83	3.62	4.48	14.43	29.03	42.21
Validation Trajectory 2							
Model A	25.26	1.69	2.37	1.23	11.84	31.46	33.36
Model B	21.88	3.07	4.42	2.33	10.56	32.16	35.01

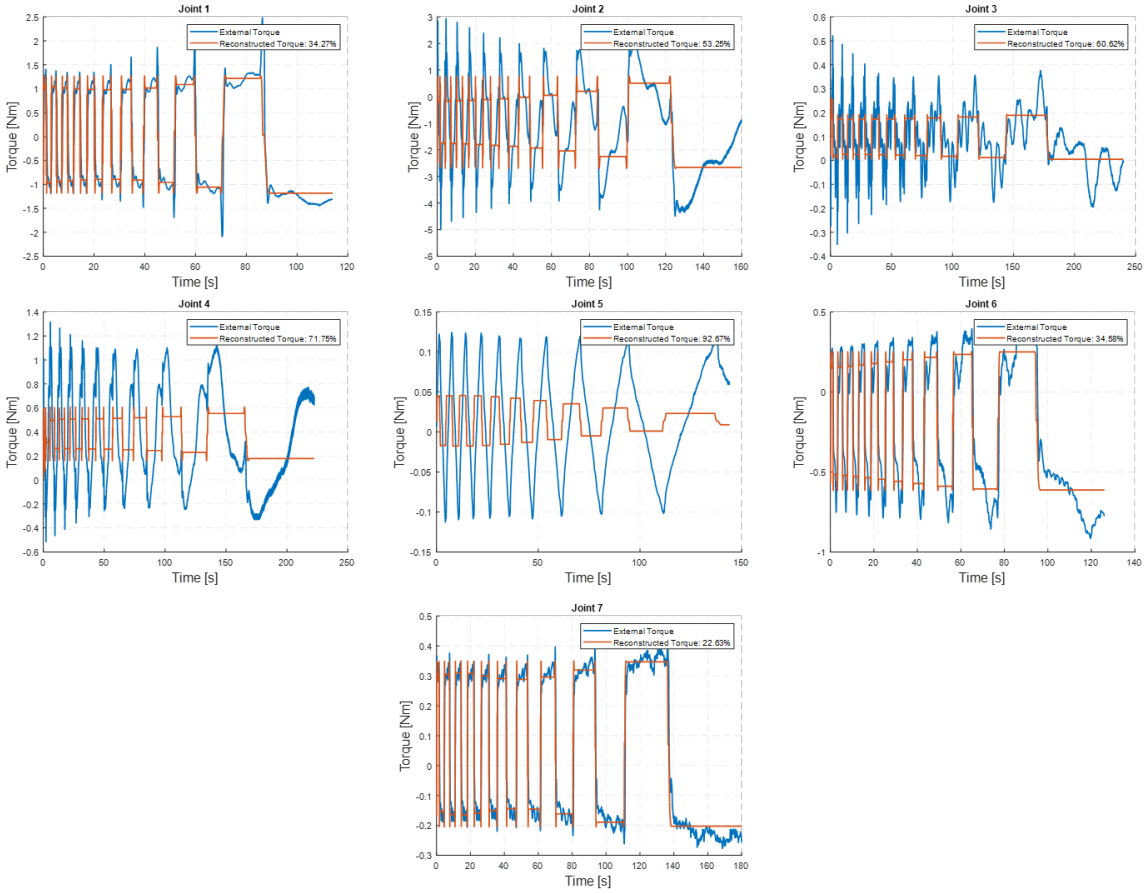


Figure 2.6: External torque reconstruction of the joints' friction models obtained for *Model B* during the training trajectories.

the previous section.

Numerical results in Table 2.9 show that the Coulomb friction parameters (f_{c_i} and τ_{off_i}) are predominant over the other effects. *Model B* succeeds in identifying better the *Stribeck* effect due to the use of trajectories specially designed for friction identification, as well as the presence of independent coefficients for each joint. Individual coefficients δ_i and α_i should be included in *Model A* for a fair comparison. However, the addition of all these parameters would make the method applied in *Model A*, inspired from [Indri and Trapani, 2020], tedious and time-consuming (the SLS method could also be tested). Moreover, the fact that the viscous friction terms are not well identified can be explained because joints limits do not allow to make movements at high speeds during a long time.

Table 2.10 shows that, for the first 6 joints the percent error between the *sensed torque* and the reconstructed torque is lower than 35% for both proposed models. The error is higher in joints 6 and 7 due to low inertia, and lower than 5% for joints 2, 3 and 4 where the inertia and gravity effects are high. Moreover, the second validation trajectory is more demanding in terms of acceleration what can explain the big difference in the torque reconstruction of joint 1.

The results from *Model A* are slightly better than those of *Model B*. This is an expected result: the KUKA's model inside the controller is not perfect, thus, there are also inertial effects on the *external torque*, which in *Model B* are not grasped. The *external torque* will account not only for the friction but also for the effect of external forces and for model uncertainties and simplifications made by the manufacturer. Thus, the second part of *Model B* identification procedure taking into account just friction parameters may be far from perfect, as it is shown in the reconstruction torque of joint 5 in Figure 2.6. In contrast, joint 7 can be almost purely defined with the friction model, as its effect is much higher than the inertial part. However, trajectories for friction identification in *Model B* are easier to design and apply (lower accelerations and displacements are required) and are less prone to errors due to noise.

2.4.3 Conclusion

In this section, we developed two identification procedures for a non-linear friction model inspired in the work of [Indri and Trapani, 2020]. On the one hand, *Model A* uses the same technique but in a global way. It has the property of leading to better overall results at the expense of using just 2 parameters to explain the non-linear effect of friction of all the joints (to reduce the computational complexity of the algorithm). On the other hand, *Model B* uses 14 parameters (2 per joint) to explain the non-linear part of the friction model, which leads, of course, to a better friction identification. It applies the same sequential approach than the one done in the mentioned work but with one main difference: we used the SLS method instead of the minimum search algorithm proposed there. This method theoretically leads to optimal results, and does not depend on the discretization and range of the search algorithm. With proper apriori knowledge of the dynamic parameters, or by identification of the model

that is inside the robot's controller if available and the use of the *external torque*, the method succeeds in finding the friction parameters with sequential trajectories and predicting the global behavior of the manipulator.

There are many possible ways to improve the results shown in this section. For instance, better trajectories could be designed (if the manipulator allows it) built with movements at lower velocities without exciting the inertial parameters. Furthermore, as said during the section, the SLS method could be applied over the global trajectory, considering independent friction models for each joint. Moreover, although the SLS method finds an optimal result of the problem, there are other methods that could be tested, as the Separable IV (SIV) presented in [Janot, 2021] that will lead to a more robust against noise procedure.

2.5 Final conclusion

In this chapter, the topic of offline en-bloc identification was covered. As explained in Chapter 1, all the measurements are available and processed before the identification method is applied, thus there is no computational time constraint. Usually, these methods are applied to obtain a first model of the system and be able to design a controller. It can be an accurate model for those systems which will not change during run-time.

During the whole chapter, two problems were mainly tackled. The first one is about the physical feasibility of the estimates of the identification methods. A novel approach, called PC-IDIM-IV, was developed, which ensures the physical feasibility of the solution on the IDIM-IV problem. For this, mathematical proofs of the validity of the instrumental matrix were shown. Several experimental validation tests were carried out obtaining different models. It has been shown that the estimates will be trajectory dependant and that, even though the new method will yield physically feasible results, in order to obtain the real value of the standard parameters, further constraints should be added. As any other IV-based method, the PC-IDIM-IV yields better results than its LS counterpart (PC-IDIM-LS) when a bad filtering is done or when the assumptions necessary for the LS estimate to be unbiased are not assured. This method can be extended, in order to be able to, for example, identify non-linear models, or to include non-linear models of friction.

The second problem is about the identification of a non-linear friction model on collaborative robots. We have proposed a method that making use of a priori knowledge (or taking advantage of the model that the manufacturer included on the robot controller), the friction can be identified from the residual torque using the SLS method. This method divides the problem in a non-linear and a linear optimization, searching for the best values. There are plenty of models, methods and ways to identify the friction. In this chapter, two ways were successfully carried out and compared.

These methods were applied in an offline way and are useful to characterize the systems. However, if the manipulator's model is intended to change during run-time, as for example in a pick and place action, these methods may not be suitable. The fact that in online

applications measurements are made at the same time than estimations, makes mandatory the use of a recursive identification method. The next chapter will go a little further into this idea, and it will be focused on recursive methods, although still in an offline way.

Chapter 3

Offline recursive identification

Contents

3.1 Introduction 71
3.2 Problem statement and related works 72
3.3 The IDIM-RIV method 75
3.3.1 Formulation 75
3.3.2 Construction of Z 76
3.3.3 Justification of the construction of Z 78
3.3.4 Algorithm 78
3.3.5 Miscellaneous remarks 79
3.3.6 Experimental results 79
3.3.7 Conclusion 87
3.4 Line-by-line coupled identification 89
3.4.1 Preliminaries 90
3.4.2 The IDIM-CLS method 90
3.4.3 The S-IDIM-RLS method 92
3.4.4 Experimental results 93
3.4.5 Conclusion 94
3.5 Final conclusion 95

3.1 Introduction

Chapter 2 focused on the offline *en-bloc* estimation of the dynamic parameters of robotic manipulator models. Although these methods are widely used to characterize robots and generate a first approximation of the system, in general, they are not applicable in a real-time basis. The reason is that *en-bloc* methods require the knowledge of all measurements beforehand. In online scenarios, the measurements are progressively available while the robot is executing its task, and the estimations need to be computed simultaneously. For this reason, Chapter 3 goes a little further in this process, focusing on recursive identification methods. The methods herein mentioned and derived are suitable for online application, as they treat one measurement at a time. However, in this chapter, the experimental validation is still

done in an offline way (it is assumed that data is already obtained and processed), in order to avoid issues related to the real-time implementation of the algorithms and to emphasize the analysis on the method per se.

This chapter is divided in three parts:

- The first section briefly focuses on related work present in literature. It sets the state-of-art of methods used for recursive identification in robotics.
- Second, a new Recursive Instrumental Variable Method (IDIM-RIV) is introduced. We justify the selection of the instrumental variable and we complement the contribution done on the previous chapter regarding the PC-IDIM-IV method. Experimental validation on the KUKA iiwa is carried out to validate the method.
- Third, we propose two new recursive methods/algorithms to be applied in robotics: the IDIM-CLS (IDIM - Coupled Least Squares) and the S-IDIM-RLS (Sequential - IDIM - RLS). These methods have attractive features for real-time application: they do not need any matrix inversion to compute the estimates leading to less computational efforts and more stable algorithms; and they present a tool to avoid updating specific parameters depending on the measurements at each time step. This can reduce even more the computational time and partially solve the problem of not having exciting trajectories in real-time scenarios. A short experimental validation is also carried out on the KUKA iiwa.

3.2 Problem statement and related works

The offline recursive identification methods. Recursive estimation methods are used in many areas of science, and the theory behind many of them can be studied from textbooks as [Ljung and Söderström, 1983, Ljung, 1998, Young, 2011]. This chapter will focus on its application on robotics.

One of the reasons these methods are popular and are needed in robotics is because they allow the application of methodologies and techniques that depend on the parameters variation during time, as the adaptive control (see [Hsia, 1986, Craig et al., 1987, Landau et al., 2011] for more information). The most used method was introduced in Section 1.4.3 and is the RLS (Recursive Least-Squares). There are plenty of works using this method in literature. For instance, the authors in [Sidhom et al., 2010] used a RLS method with forgetting factor to estimate the parameters of a 2 *dof* SCARA manipulator; in [Neuman and Khosla, 1986], the authors applied it on a 3 *dof* Stanford manipulator on simulation; [Middletone and Goodwin, 1986] built an adaptive control based on it; and the authors in [Liu et al., 2016] proposed a recursive differential evolution algorithm based on the RLS in order to estimate parameters while signals are being measured, and validated it on offline simulation on a 2 *dof* serial manipulator.

Moreover, there are other works that evaluated different versions of this method and compared them with others. In [Kubus et al., 2007, Kubus et al., 2008b], the authors proposed a RTLS (Recursive Total Least-Squares) and compared it with the RLS and RIV (Recursive IV) methods. They identified the load parameters using a Stäubli RX60 industrial manipulator equipped with a wrist-mounted force-torque and acceleration sensor. They showed that the RTLS is superior to the other two methods because of the inclusion of errors both in the torque vector and in the regressor matrix, although it seems to take more time to converge to the real value than the RIV method. Furthermore, in [Kubus et al., 2008a] they used a RIV method to estimate the load parameters and compensate non-contact forces due to it on the model. Furthermore, [De Souza et al., 2021] presents a hybrid identification method based on the RLS optimized via the Kalman Filter (KF). The method is called RLS-KF, and the authors compared it with the RLS and the Extended RLS (ERLS). They used these methods to identify the respective transfer functions of a cylindrical 3 *dof* manipulator.

An alternative method, more common in the automatic control community, is the use of the Extended Kalman Filter (EKF) algorithm for identification [Lightcap and Banks, 2009]. Although this method is more robust against noise than the LS solution, it is highly sensitive to initial conditions. In [Gautier and Poignet, 2001], the authors compared the EKF method with the non-recursive WLS on a 2 *dof* SCARA type robot. They showed that the EKF is usually computationally more costly than its LS counterpart.

Besides the RLS and the EKF, there are other methods that have been applied in literature. The work in [Östring and Gunnarsson, 2004] uses a recursive prediction error algorithm to identify some parameters that are subject to changes (motor torque constant, viscous friction coefficients and moments of inertia) by moving the first axis of the 6 *dof* ABB IRB 1400 industrial manipulator. In [Urrea and Pascal, 2017, Urrea and Pascal, 2018], the authors test different methods on a 3 *dof* SCARA, namely LS, Adaline (Adaptive Linear Neuron) neural networks, Hopfield neural networks (HNNs), EKF and genetic algorithms. Even though they do not explicitly treat the recursive identification, they showed that the HNNs can be used for estimating parameters on-line, as they present a recursive formulation.

As well as in offline en-bloc identification, there are a lot of methods that can be applied. Its choice will depend on the characteristics of the system, of the measurements and of the expected estimates. Nevertheless, methods based on the LS are more frequently used in recursive estimation of robotic manipulators over the other options mentioned in this section. However, the potential of IV based methods have not yet been fully exploited. The advantage of applying them has been shown in the previous chapter, and Section 3.3 is aimed at developing its recursive version: the IDIM-RIV method. The RIV method has been extensively studied in literature in other areas of science [Young, 2011], and it was first formally proposed for robotics applications in [Brunot and Janot, 2018]. In the mentioned section, we will present an extension of this work.

The sequential identification methodology. Sequential identification arose as an alternative to global identification for situations where the trajectory generation and/or execution in order to identify all the parameters at the same time becomes a difficult task. As said in

Section 1.4.1.3, it divides the problem in a set of sub-problems which have a less amount of parameters to estimate each. It is an attractive method because it facilitates the generation of exciting trajectories, making the technique less sensitive to noise measurements and error modelling (due to a better parameter excitation). Although the division can be done in different ways, the basic one involves moving one joint while the others are locked. In [Khalil and Dombre, 2002], the authors stated:

"...This technique simplifies the identification equations. However, an accumulation of errors may occur since the values of some estimated parameters will be assumed to be known in subsequent identification."

This is the reason why global identification is usually preferred over sequential identification: dividing the process in different steps leads to an accumulation of errors, resulting, in general, in less efficient estimates than global approaches [Gautier and Presse, 1991].

However, one could wonder that if this loss of efficiency comes from the fact of considering as known parameters that were identified in previous steps, the process could be improved by considering the uncertainties related to these parameters. Usually, identification methods yield estimates in the form of statistical distributions, where the measurement of their dispersion may not be negligible. However, up to the moment, the applied sequential identification techniques consider as totally known (thus no variability) the parameters estimated in previous steps. For instance, in [De Wit and Aubin, 1990], the authors applied a sequential line-by-line (also called joint-by-joint because the identification is carried out in different steps, where each of them correspond to the movement of one joint) RLS estimation in the TAM robot 4-*dof* general purpose tele-manipulator. The authors in [Jung et al., 2018] used the same approach of dividing the IDM matrix by rows and included physical feasible constraints, in what they called a *backward sequential approach*. Furthermore, in [Curtelin et al., 1991, Yoshida et al., 1992, Qin et al., 2010], the authors used different techniques where they generated particular trajectories to excite some specific phenomenon once at a time (friction, gravity, inertia, etc.), which reduces the error propagation, as the parameters identified in one trajectory will have low or none effect on the others.

Although this chapter is not focused on sequential identification, the previously mentioned issue and the application to real-time scenarios in Chapter 4, motivated the development of Section 3.4. There, we propose two methods called the IDIM-CLS (IDIM - Coupled Least Squares) and the S-IDIM-RLS (Sequential IDIM - Recursive Least Squares). As well as in sequential identification, these two methods divide the system to be identified in different subsystems. The division with respect to its joints (thus with respect to the rows of the **IDM** matrix) leads to methods where no matrix inversion is required. This is a good characteristic for online application, as matrix inversion can be computationally costly and lead to algorithm instability. Moreover, it allows to decide which of the subsystems to update in each iteration, reducing its computing time drastically, and, thanks to their recursive nature, the uncertainties in the regressor matrix are propagated from one step to another.

3.3 The IDIM-RIV method

In this section, we present and improve the IDIM-RIV method. This method is more robust against noise than IDIM-RLS (which is a common characteristic of IV-based methods), but has the problem of how to adequately choose $\boldsymbol{Z}$ for optimal results. As mentioned in [Janot et al., 2013b] and in Chapter 2, in robotics, a valid auxiliary model for the generation of the instrumental matrix is the simulation of the DDM. However, its computational cost and possible numerical issues, makes this choice not suitable for online applications. To avoid the real-time simulation of the robot's model, the authors in [Brunot and Janot, 2018], inspired by the work in [Boeren et al., 2017], proved that the instrumental matrix can be also built with the reference trajectories of position, velocity and acceleration sent to the robot, which are usually computed before the robot's movement. There, the authors validated the approach offline on a 2-*dof* planar SCARA robot.

In the present section, we will enhance the work in [Brunot and Janot, 2018] and focus on the following question: how can we construct a valid instrumental matrix without the simulation of the DDM? In that sense, we present three improvements of the IDIM-RIV method. First, we no longer need to know the controller because we filter the reference signals with the identification of the closed-loop position transfer function of each joint. Interestingly, since at low frequencies the closed-loop transfer function acts as a unitary gain, we can construct the instrumental matrix with the references directly if only low frequencies are excited. This is a highly likely scenario, since the manipulator's trajectory are generally designed to be smooth. Second, we evaluate the robustness of IDIM-RIV against data filtering and initialization. Third, we execute IDIM-RIV offline to identify a bigger number of base dynamic parameters of a 7-dof industrial collaborative robotic manipulator.

The section is structured as follows:

- First, the formulation of the IDIM-RIV is made.
- Second, the issue of the construction of the instrumental variable is tackled.
- Third, we present an algorithm to run the IDIM-RIV method, followed by some remarks regarding its application.
- Finally, the experimental validation is carried out on the KUKA iiwa manipulator.

3.3.1 Formulation

The RIV estimate $\hat{\boldsymbol{\beta}}_{RIV}$ is closely related to the RLS estimate $\hat{\boldsymbol{\beta}}_{RLS}$ presented in Section 1.4.3 [Young, 2011]:

$$\begin{aligned} \boldsymbol{K}(k) &= \boldsymbol{P}(k-1)\boldsymbol{Z}^T(k)(\mathbb{I}_n + \mathbf{IDM}(k)\boldsymbol{P}(k-1)\boldsymbol{Z}^T(k))^{-1} \\ \boldsymbol{P}(k) &= \boldsymbol{P}(k-1) - \boldsymbol{K}(k)\mathbf{IDM}(k)\boldsymbol{P}(k-1) \\ \hat{\boldsymbol{\beta}}_{RIV}(k) &= \hat{\boldsymbol{\beta}}_{RIV}(k-1) + \boldsymbol{K}(k)(\boldsymbol{\tau}(k) - \mathbf{IDM}(k)\hat{\boldsymbol{\beta}}_{RIV}(k)), \end{aligned} \tag{3.1}$$

where most of the terms were already defined, but are remembered for better readability: $\boldsymbol{P}(k) \in \mathbb{R}^{n_b \times n_b}$ is the covariance matrix, $\boldsymbol{K}(k) \in \mathbb{R}^{n_b \times n}$ is the estimator gain matrix, $\mathbf{IDM}(k) \in \mathbb{R}^{n \times n_b}$ is the observation matrix defined in Equation (1.25), $\boldsymbol{\tau}(k) \in \mathbb{R}^{n \times 1}$ is the vector of torques also defined in Equation (1.25), $\boldsymbol{Z}(k) \in \mathbb{R}^{n \times n_b}$ is the instrumental matrix, and $\hat{\boldsymbol{\beta}}_{RIV}(k) \in \mathbb{R}^{n_b \times 1}$ is the vector of RIV estimates respectively, at the iteration k ($k = 1, ..., m$) being m the amount of measurements, and $\mathbb{I}_n$ is the identity matrix of order n.

3.3.2 Construction of Z

Having presented the basic formulation of RIV, in this subsection we will focus on answering the question presented in the introduction of this section: how can we construct a valid instrumental matrix without the simulation of the DDM? Although the context was different, [Boeren et al., 2017] presented an elegant way to answer this question by considering the closed-loop transfer function between the reference and output position. This result led the authors in [Brunot and Janot, 2018] to construct $\boldsymbol{Z}$ as follows. First, they considered the DDM as n decoupled linear models since the gains of the controller are tuned so that the influence of the nonlinear coupling of joint j proves to be negligible [Khalil and Dombre, 2002, Siciliano et al., 2010]. This nonlinear coupling p_j of joint j is obtained from Equation (1.11) (without considering the friction torque and the vector of external torques) and it is given by:

$$p_j = -\sum_{i=1; i \neq j}^{n} M_{ji}(\boldsymbol{q})\ddot{q}_i - N_j(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{3.2}$$

p_j decouples the IDM equation, making the torque of joint j depend only of its acceleration and the respective component of the diagonal of the inertia matrix. The joint j decoupled model is then approximated with:

$$G_j(s) = \frac{1}{J_{max_j} s^2}, \tag{3.3}$$

where s is the Laplace's variable and J_{max_j} is the maximum joint j inertia defined by:

$$J_{max_j} = ZZ_j + I_{a_j} + \max_{\mathbf{q}}(M_{jj}(\mathbf{q}) - ZZ_j - I_{a_j}). \tag{3.4}$$

J_{max_j} is the maximum value with respect to $\mathbf{q}$ of the inertia around the joint axis $\boldsymbol{z_j}$. This gives the smallest damping value and the smallest stability margin of the closed-loop second order transfer function while $\mathbf{q}$ varies (see [Gautier et al., 2012]).

Second, in robotics the gains within the controller are tuned such that the influence of p_j are negligible in order to obtain a good tracking. Thus, below the bandwidth of the joint j position closed loop, the following relation holds [Khalil and Dombre, 2002, Brunot and Janot, 2018]:

$$q_{nf_j} \approx H_j(s) q_{r_j}, \tag{3.5}$$

where q_{r_j} is the joint j reference, q_{nf_j} is the noise free part of q_j, and $H_j(s)$ is the joint j

position closed-loop transfer function given by:

$$H_j(s) = \frac{C_j(s)G_j(s)}{1 + C_j(s)G_j(s)}, \tag{3.6}$$

where $C_j(s)$ is the joint j controller's transfer function and $G_j(s)$ was defined in Equation (3.3).

Third, with $\hat{q}_{r_j} = H_j(s)q_{r_j}$ being the filtered references of position, they chose $\boldsymbol{Z} = \mathbf{IDM}\left(\hat{\boldsymbol{q}}_r, \dot{\hat{\boldsymbol{q}}}_r, \ddot{\hat{\boldsymbol{q}}}_r\right)$, where $\hat{\boldsymbol{q}}_r, \dot{\hat{\boldsymbol{q}}}_r, \ddot{\hat{\boldsymbol{q}}}_r \in \mathbb{R}^n$, are the vectors of *filtered* references of positions, velocities and accelerations, respectively.

Even if it is a valid and correct approach, this construction requires knowledge of $C_j(s)$. Knowing that industrial robot companies are overcautious regarding their controllers, it is a severe limitation. One possibility is to identify the structure of $C_j(s)$, and estimate the parameters to get $\hat{C}_j(s)$, the estimate of $C_j(s)$. Then, we use $\hat{C}_j(s)$ to reconstruct $H_j(s)$, with

$$\hat{H}_j(s) = \frac{\hat{C}_j(s)G_j(s)}{1 + \hat{C}_j(s)G_j(s)}, \tag{3.7}$$

and calculate $\hat{q}_{r_j}, \dot{\hat{q}}_{r_j}, \ddot{\hat{q}}_{r_j}$. If reconstructing $H_j(s)$ in this way is appealing, why not identifying $H_j(s)$ directly? Indeed, many today's toolboxes, such as CAPTAIN [Young, 2011], do that very well, and identifying a transfer function is not a burden. So, unlike in [Boeren et al., 2017] and [Brunot and Janot, 2018], we suggest to get an estimate of $H_j(s)$, denoted $\hat{H}_j(s)$, and construct $\hat{q}_{r_j}, \dot{\hat{q}}_{r_j}, \ddot{\hat{q}}_{r_j}$ with $\hat{q}_{r_j} = \hat{H}_j(s)q_{r_j}$. We can easily obtain $\hat{H}_j(s)$, for example, by executing the *rivc*, continuous-time refined instrumental variable, function of the CAPTAIN toolbox with q_{r_j} and q_j as input and output, respectively. It comes that we choose:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}(1) \\ \dots \\ \boldsymbol{z}(k) \\ \dots \\ \boldsymbol{z}(m) \end{bmatrix}, \tag{3.8}$$

where $\boldsymbol{z}(k) = \mathbf{IDM}(\hat{\boldsymbol{q}}_r(k), \dot{\hat{\boldsymbol{q}}}_r(k), \ddot{\hat{\boldsymbol{q}}}_r(k)) \in \mathbb{R}^{n \times n_b}$ is the IDM fed with the *filtered* references and their derivatives at time k.

This construction of $\boldsymbol{Z}$ leads to a simpler RIV approach since we do not need to simulate the DDM, and, therefore, to know the controller. Compared to [Brunot and Janot, 2018], this is a major improvement. Besides, if the robot has to track trajectories for planned tasks, we can construct $\boldsymbol{Z}$ before execution. It is a potentially useful approach for real-time implementation. Otherwise, we simply construct $\boldsymbol{z}(k)$ with the IDM and filtered reference signals at each time k. This enhanced recursive instrumental variable algorithm is called IDIM-RIV.

3.3.3 Justification of the construction of Z

In this subsection, we show that $\boldsymbol{Z}$ defined in Equation (3.8) is a valid instrumental matrix.

We note that $q_{nf_j} = \hat{q}_{r_j} = H_j(s)q_{r_j}$, where q_{nf_j} is the noise-free part of q_j, and q_{r_j} is noise-free by definition. This is a *reasonable* assumption because it implicitly assumes that the control rejects well p_j. If this assumption is not fulfilled, it would mean that the controller does not have a good performance because it does not reject well the perturbations, which is rarely the case for industrial robots. Therefore, we have to show that:

$$\underset{m\to\infty}{\text{plim}}\, \hat{H}_j(s) = H_j(s), \tag{3.9}$$

where $\underset{m\to\infty}{\text{plim}}$ is the limit in probability as m tends to infinity. To do so, we assume that q_j and q_{r_j} are quasi-stationary signals with finite bounded fourth moments (see [Ljung, 1998]). This is a fair assumption because if the first four moments are not finite, then the robot is not working in usual operations [Janot et al., 2013b]. With these assumptions and the use of the *rivc* function of CAPTAIN toolbox, $\hat{H}_j(s)$ is a consistent estimate of $H_j(s)$ according to [Gilson et al., 2011, Young, 2011]. It follows straight that, for all j:

$$\underset{m\to\infty}{\text{plim}}\,(\hat{H}_j(s)q_{r_j}) = \underset{m\to\infty}{\text{plim}}\,(\hat{H}_j(s))q_{r_j} = H_j(s)q_{r_j} = \hat{q}_{r_j} = q_{nf_j}, \tag{3.10}$$

implicating $\left(\hat{\boldsymbol{q}}_r, \hat{\dot{\boldsymbol{q}}}_r, \hat{\ddot{\boldsymbol{q}}}_r\right) = \left(\boldsymbol{q}_{nf}, \dot{\boldsymbol{q}}_{nf}, \ddot{\boldsymbol{q}}_{nf}\right)$, leading to $\boldsymbol{Z} = \boldsymbol{W}\left(\boldsymbol{q}_{nf}, \dot{\boldsymbol{q}}_{nf}, \ddot{\boldsymbol{q}}_{nf}\right) = \boldsymbol{W}_{nf}$, which has already been proven to be a valid instrumental matrix in [Young, 2011, Janot et al., 2013b] and in Section 2.3.1. To complete this short analysis, we must emphasize that $\boldsymbol{Z} = \boldsymbol{W}_{nf}$ induces $\boldsymbol{z}(k) = \mathbf{IDM}_{nf}(\boldsymbol{q}_{nf}(k), \dot{\boldsymbol{q}}_{nf}(k), \ddot{\boldsymbol{q}}_{nf}(k)) \in \mathbb{R}^{n\times n_b}$. It comes that $\boldsymbol{z}(k)$ is also a valid instrumental matrix.

3.3.4 Algorithm

In the current section, we present the IDIM-RIV algorithm. We note $\hat{\boldsymbol{\beta}}_{\text{RIV}}(k)$, the IDIM-RIV estimate at time k with $k = 1, ..., m$. The algorithm is executed as follows:

Before execution: Compute the symbolic expression of the IDM with the SYMORO+ software [Khalil and Creusot, 1997] or other software that allows symbolic computation as MATLAB. If $C_j(s)$ is not known, estimate each $H_j(s)$ with, for example, the *rivc* function of CAPTAIN toolbox to get $\hat{H}_j(s)$ using the reference and measured positions. Compute $\hat{q}_{r_j} = \hat{H}_j(s)q_{r_j}$ and its derivatives for all joints j to obtain $(\hat{\boldsymbol{q}}_r(k), \hat{\dot{\boldsymbol{q}}}_r(k), \hat{\ddot{\boldsymbol{q}}}_r(k))$. If the tasks are planned before execution, then choose $\boldsymbol{Z} = \left[\boldsymbol{z}^T(1)\cdots\boldsymbol{z}^T(k)\cdots\boldsymbol{z}^T(m)\right]^T$. Otherwise construct $\boldsymbol{z}(k) = \mathbf{IDM}(\hat{\boldsymbol{q}}_r(k), \hat{\dot{\boldsymbol{q}}}_r(k), \hat{\ddot{\boldsymbol{q}}}_r(k)) \in \mathbb{R}^{n\times n_b}$ at time k with the current filtered references.

Initialization: $\hat{\boldsymbol{\beta}}_{\text{RIV}}(0)$ takes usually the CAD values if available, or is estimated with a non-recursive identification method beforehand, as the ones showed in Chapter 2.

Step k: At time k, the IDIM-RIV solution, $\hat{\boldsymbol{\beta}}_{RIV}(k)$, its covariance matrix, $\boldsymbol{P}(k) \in \mathbb{R}^{n_b \times n_b}$, and the estimator gain matrix, $\boldsymbol{K}(k) \in \mathbb{R}^{n_b \times n}$, are calculated with Equation (3.1).

3.3.5 Miscellaneous remarks

For this method, some important remarks need to be given:

Remark 1: In this section, we have considered $C_j(s)$ as a linear controller. If the control law is the computed torque, then $C_j(s)$ is no longer linear, and identify it turns to identifying the IDM itself [Gaz et al., 2019]. That said, we can still identify $H_j(s)$ since after the compensation, it is a second-order transfer function in this case [Khalil and Dombre, 2002, Siciliano et al., 2010].

Remark 2: At low frequencies, one usually has $H_j(s) = 1$, and it implies that we can construct $\boldsymbol{Z}$ directly with the references since one has $(\hat{\boldsymbol{q}}_r, \hat{\dot{\boldsymbol{q}}}_r, \hat{\ddot{\boldsymbol{q}}}_r) = (\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)$. Note that $H_j(s) = 1$ may also occur with the computed torque control since this approximation is what we expect when we implement this type of control law [Khalil and Dombre, 2002].

Remark 3: Constructing $\mathbf{IDM}(k)$ and $\boldsymbol{z}(k)$ is not a burden since a symbolic expression of the IDM is obtained beforehand with an according software as SYMORO+ or MATLAB [Khalil and Dombre, 2002]. Moreover, $(\hat{\boldsymbol{q}}(k), \hat{\dot{\boldsymbol{q}}}(k), \hat{\ddot{\boldsymbol{q}}}(k))$ and $(\hat{\boldsymbol{q}}_r(k), \hat{\dot{\boldsymbol{q}}}_r(k), \hat{\ddot{\boldsymbol{q}}}_r(k))$ are calculated from measurements and references with standard digital filters that are online implementable [Brunot and Janot, 2018].

Remark 4: In Equation (3.1), we do not account for the covariance matrix of $\boldsymbol{\tau}(k)$ denoted $\boldsymbol{R_w}$ assumed to be constant as in [Gautier et al., 2012, Janot et al., 2013b]. To improve the statistical efficiency, i.e., to get minimum variances, we can adopt a weighted RIV method, called IDIM-WRIV. To do so, we introduce a weighting matrix $\boldsymbol{W_w} = \boldsymbol{R_w}^{-1/2}$, and substitute $\mathbf{IDM}(k)$, $\boldsymbol{z}(k)$ and $\boldsymbol{\tau}(k)$ by $\boldsymbol{W_w}, \mathbf{IDM}(k)$, $\boldsymbol{W_w z}(k)$ and $\boldsymbol{W_w \tau}(k)$, respectively, in Equation (3.1). However, using IDIM-WRIV requires an estimate of $\boldsymbol{R_w}$, and it implies running *en-bloc* methods before executing IDIM-WRIV [Young, 2011].

3.3.6 Experimental results

In this subsection, we will validate the method by estimating a set of base parameters of the KUKA iiwa manipulator. The working framework, data gathering and processing are presented in Appendix B and Appendix C. The validation is made in two parts.

- First, the IDIM-RIV is tested in two scenarios to assess its robustness against noise: one in which measurements are filtered and another where they are not.
- Second, we compare the IDIM-RLS with the IDIM-RIV. Filtered data and proper initialization parameters are used. We also show the importance of a correct choice of

initial values in the IDIM-RLS method. This subsection is part of the work presented in [Ardiani et al., 2022].

3.3.6.1 IDIM-RIV

Exciting trajectories. Online parameter identification is designed so that it captures the relevant information while the robot is doing a specific task not necessarily developed for identification. Thus, as mentioned in Appendix C and in Section 1.3.3, we are not going to use a specially designed trajectory as that proposed by [Swevers et al., 1997b]. The trajectories used in this section are 50 points randomly selected in the whole work-space of the robot, from which the KUKA Sunrise OS generates the reference position, velocity, and acceleration profiles [kuka, 2017]. Although these trajectories are not designed for a specific task, there is no optimization carried out to assure a proper excitation of the parameters. This method of generating trajectories yields similar results than those in [Swevers et al., 1997b] for this kind of manipulators (see [Jubien et al., 2014a]) and guarantees the excitation of all the base parameters.

Note that we could secure the physical feasibility of estimates with further constraints included in the optimization process as in [Sousa and Cortesao, 2014, Sousa and Cortesao, 2019, Janot and Wensing, 2021], and the PC-IDIM-IV method introduced in Chapter 2. However, including such constraints in recursive algorithms is not easy, and [Janot and Wensing, 2021] showed that the physical feasibility is secured if the estimates are consistent; the reciprocity being not true.

Scenarios. We consider two situations. First, *Situation 1* refers to the case when a digital filter with cutoff frequency of 3.5 Hz is applied to the data and $\hat{\boldsymbol{\beta}}_{\text{RIV}}$ is initialized with the results of carrying out the IDIM-LS identification of the robot's model that is integrated into the industrial controller (see Section 2.3.6.1). This is done by building $\boldsymbol{W}$ with the commanded position and its derivatives, and $\boldsymbol{y}$ with the commanded torque. Second, *Situation 2* refers to the case when no filter is applied to data and $\hat{\boldsymbol{\beta}}_{\text{RIV}}$ is initialized with no *a priori* knowledge. In both cases, points where any of the joint's velocity is less than 0.01 rad/s are not considered to avoid problems due to the discontinuity generated by the non-smooth around zero model of friction in Equation (1.14), see [Gautier et al., 2012]. The initialization of $\boldsymbol{P}$ is set to big values (10-100) when no *a priori* information is available and small values (0.01) when there is [Gautier and Poignet, 2001].

Results and discussion. Now, we show the results of IDIM-RIV using the *sensed position* and its derivatives to build $\mathbf{IDM}(k)$, the *sensed torque* to build $\boldsymbol{\tau}(k)$, and the *commanded position* and its derivatives to build $\boldsymbol{z}(k)$. We have identified each $H_j(s)$ with the *rivc* function of CAPTAIN toolbox. For the seven joints, $\hat{H}_j(s)$ is a second-order transfer function, and the fitting between q_j and q_{r_j} is practically perfect since it reaches 99.9%. Furthermore, at low frequencies, we have $\hat{H}_j(s) = 1.0$ for all j. It follows straight that we have $(\hat{\boldsymbol{q}}_r, \hat{\dot{\boldsymbol{q}}}_r, \hat{\ddot{\boldsymbol{q}}}_r) =$

$(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)$, and it implies that $\boldsymbol{z}(k)$ can be directly constructed with the references.

The average value of the IDIM-RIV estimates of the 64 base parameters over the last 100 samples, and their respective relative deviations $\%\sigma_i$ are shown in Table 3.1 for both mentioned situations. We take the average to reduce the effect of possible small variations that happen during the evolution of the estimates. The relative deviation of each parameter i is calculated as: $\%\sigma_i = 100 \times \sqrt{P_{i,i}}/|\hat{\beta}_{\mathrm{RIV}_i}|$, being $P_{i,i}$ the diagonal elements of the covariance matrix of the IDIM-RIV in Equation (3.1). Even though many base parameters are considered as non-essential, i.e., $\%\sigma_i > 30\%$, for this trajectory, the main inertial parameters of links capturing the influence of inertia and gravity, i.e., MYR_j, XXR_j and ZZR_j, with $j = 2, 4, 6$, are well identified. Furthermore, as the comparison of the estimates' results of both situations shows very small differences, the robustness against noise and initialization of $\hat{\boldsymbol{\beta}}_{\mathrm{RIV}}$ of the IDIM-RIV is proved. Moreover, the relative standard deviation value of the different parameters may be significantly different, because it is a relative attribute that will depend on the value of the estimate. Notice that, the parameters that are well identified in one situation, thus they have a small relative standard deviation, are also well identified on the other situation.

These two results suggest that IDIM-RIV needs less *a priori* knowledge of the system, and requires minimal data processing. IDIM-RIV is then robust against initialization and noise. That said, we must highlight that the price to pay is the time of convergence: the less the data is processed and the less information is given before execution, the more it will take to converge.

Moreover, Figure 3.1 shows the evolution of some base parameters of *Situation 1*. *Situation 2* is not shown because its variability over the first 500-1000 measurements is very high in comparison to those shown in Figure 3.1. The same phenomena, at a lower scale, is seen for the estimates of *Situation 1*, where after a first abrupt variation of the estimates in the first seconds of execution, the values converge. We explain this variability with the fact that, at the beginning, the number of samples is not enough to identify the 64 parameters correctly. If we had executed IDIM-IV, the observation matrix would have been rank-deficient for around the 1000 first samples; the information is not sufficient.

To validate the estimated models in Table 3.1, Table 3.2 shows the percent error of the torque reconstruction relative to the *sensed torque* $\boldsymbol{\tau}_j$ over three validation trajectories calculated as: $100|\frac{\boldsymbol{\tau}_j - \boldsymbol{\tau}_{jr}}{\boldsymbol{\tau}_{jr}}|$, being $\boldsymbol{\tau}_{jr}$ the reconstructed torque for joint j. Figure 3.2 shows the reconstruction torques and *sensed torques* over the first validation trajectory using the results from *Situation 1*. For the central links 2, 3, and 4, the percent error is lower than 4%, being bounded to 17% on the first five links. The higher relative percent errors of joints 6 and 7 are explained by the fact that the influence of the inertia parameters is practically insignificant compared to the friction and noise effects due to the low payload. Even so, we can conclude that the identified model succeeds in predicting the behavior of the manipulator. Interestingly, although the context is a bit different, these results are in line with those exposed in [Gaz et al., 2014]. In addition, by analyzing the close results in Table 3.2 of both situation, we can again validate the robustness against noise and initialization of the method.

Table 3.1: IDIM-RIV set of base parameters estimates averaged over the last 50 measurements and their respective relative standard deviation.

Param.	*Situation 1*		*Situation 2*		Param.	*Situation 1*		*Situation 2*	
	Value	%σ	Value	%σ		Value	%σ	Value	%σ
ZZR_1	0.045	21.5	0.061	14.8	XXR_5	0.006	110.5	0.009	69.3
f_{c_1}	1.002	3.0	1.026	3.0	ZZR_5	0.000	1360.1	0.038	114.4
f_{v_1}	0.016	184.2	0.016	177.1	XY_5	0.000	12520.4	-0.012	23.4
τ_{off_1}	0.139	10.5	0.074	19.8	XZ_5	-0.002	143.8	-0.008	34.8
XXR_2	2.227	1.4	2.150	1.4	YZ_5	-0.014	24.5	-0.016	19.7
ZZR_2	2.271	1.2	2.493	1.1	MX_5	0.002	88.1	0.012	18.5
XY_2	0.003	390.2	0.004	237.2	MYR_5	-0.088	2.0	-0.081	2.1
XZ_2	-0.047	36.2	0.037	42.0	f_{c_5}	0.022	122.6	0.049	56.5
YZ_2	0.065	12.1	0.087	8.1	f_{v_5}	0.017	126.9	0.002	912.8
MX_2	0.023	77.8	0.063	28.5	τ_{off_5}	-0.025	60.4	-0.029	52.3
MYR_2	-5.862	0.1	-5.879	0.1	XXR_6	0.020	25.1	0.014	33.4
f_{c_2}	1.868	1.3	1.847	1.3	ZZR_6	0.018	24.2	-0.002	177.5
f_{v_2}	-0.901	8.1	-0.764	9.6	XY_6	0.003	75.0	0.005	40.5
τ_{off_2}	-1.185	13.5	-0.856	18.7	XZ_6	0.004	54.8	-0.006	40.5
XXR_3	-0.042	64.3	-0.069	38.2	YZ_6	0.006	47.8	0.004	65.5
ZZR_3	0.002	504.1	-0.012	61.4	MX_6	-0.007	26.8	-0.013	13.5
XY_3	-0.001	2538.5	0.016	75.9	MYR_6	-0.118	1.3	-0.118	1.3
XZ_3	-0.009	75.3	0.004	150.8	f_{c_6}	0.368	6.0	0.376	5.8
YZ_3	-0.041	17.1	-0.070	9.3	f_{v_6}	-0.009	310.6	-0.017	158.6
MX_3	0.002	245.6	0.021	17.5	τ_{off_6}	-0.245	6.7	-0.273	6.0
MYR_3	-0.045	5.4	-0.040	5.9	XXR_7	-0.002	159.4	0.006	56.4
f_{c_3}	0.093	26.7	0.103	24.2	ZZ_7	-0.000	687.5	-0.001	186.2
f_{v_3}	0.022	118.6	0.041	62.2	XY_7	0.002	69.7	-0.002	69.1
τ_{off_3}	0.016	91.0	0.018	81.8	XZ_7	0.002	106.8	-0.001	155.6
XXR_4	0.813	1.1	0.857	1.0	YZ_7	-0.002	97.5	0.001	179.4
ZZR_4	0.853	0.9	0.876	0.8	MX_7	0.001	150.2	-0.000	2346.1
XY_4	0.003	124.4	0.002	261.7	MY_7	-0.001	101.3	-0.005	23.3
XZ_4	-0.003	115.8	-0.015	22.9	f_{c_7}	0.250	10.0	0.264	9.5
YZ_4	0.009	56.5	-0.015	32.9	f_{v_7}	-0.008	284.7	-0.018	117.4
MX_4	-0.005	58.7	-0.017	18.4	τ_{off_7}	0.057	25.4	0.057	25.8
MYR_4	2.369	0.09	2.375	0.1					
f_{c_4}	0.124	24.6	0.161	19.0					
f_{v_4}	0.067	48.5	0.048	68.8					
τ_{off_4}	0.351	6.5	0.409	5.6					

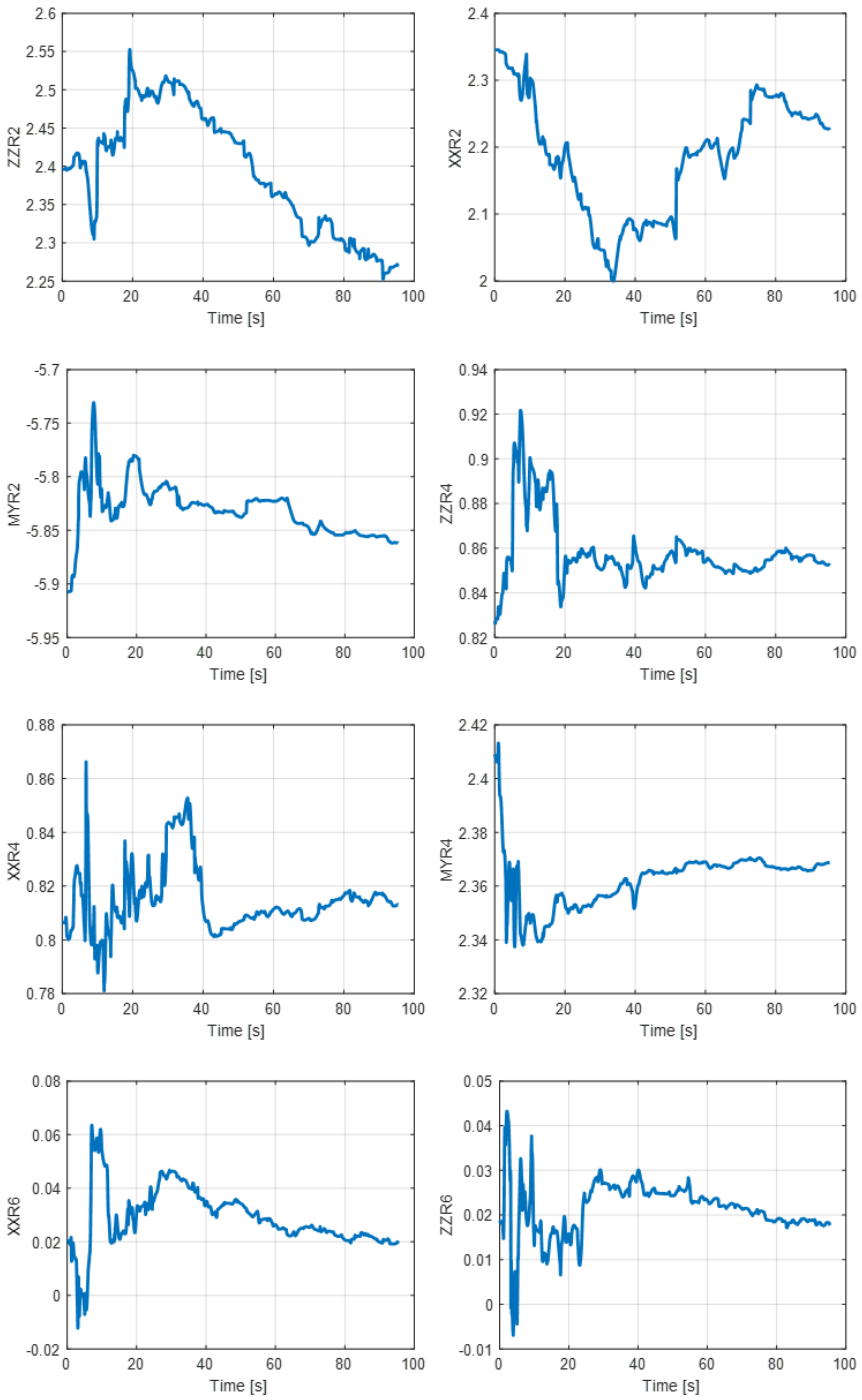


Figure 3.1: Evolution of ZZR_2, XXR_2, MYR_2, ZZR_4, XXR_4, MYR_4, ZZR_6 and XXR_6 estimates in *Situation 1*.

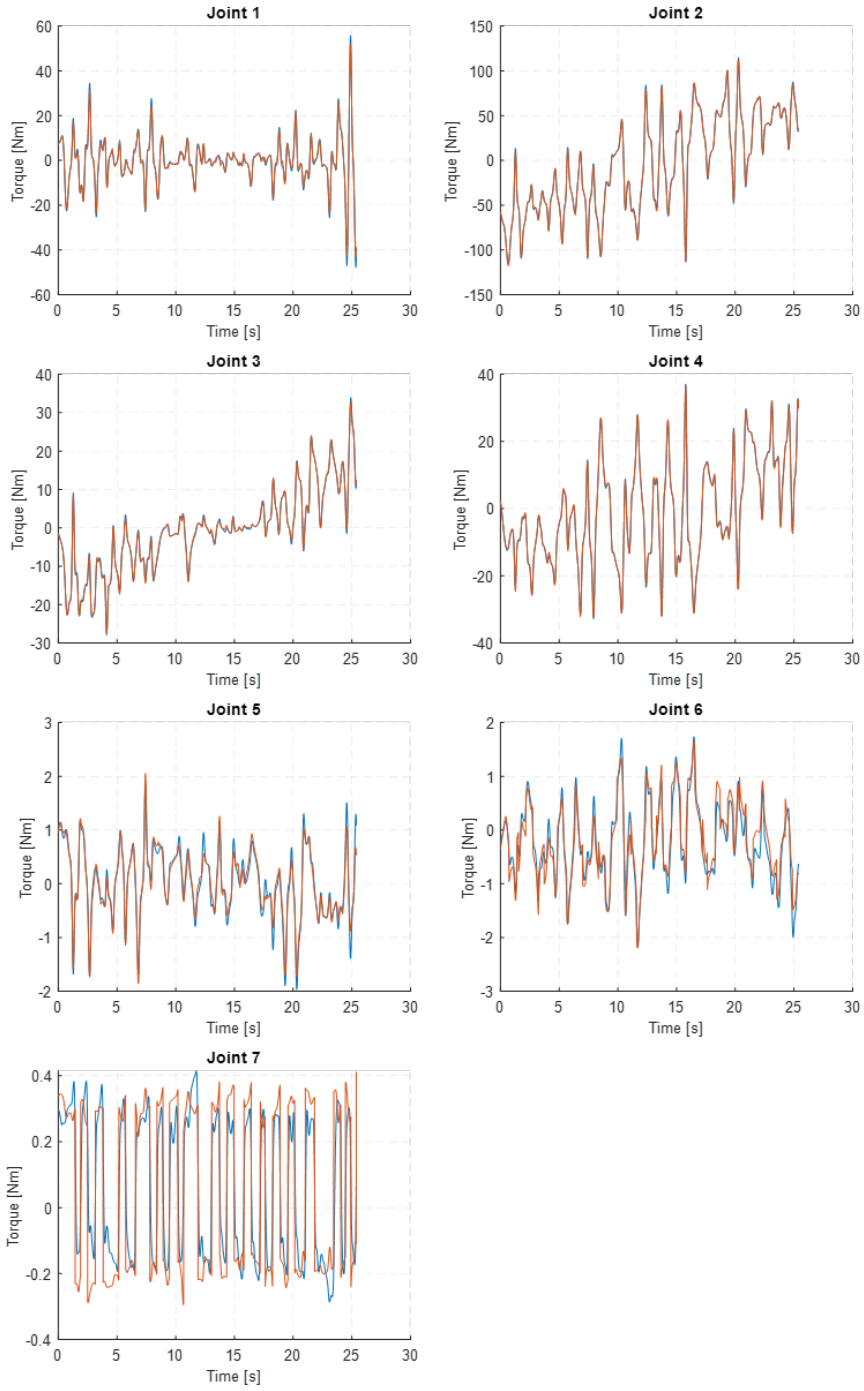


Figure 3.2: Comparison between: sensed torque (blue) and reconstructed torque (orange) with estimates from Table 3.1 using Traj. 1 in *Situation 1*.

Table 3.2: Percent error [%] of torque reconstruction using the IDIM-RIV estimates of Table 3.1.

Situation 1							
Joint	1	2	3	4	5	6	7
Traj. 1	7.52	2.90	3.54	2.73	16.69	25.76	19.37
Traj. 2	9.77	1.89	3.01	1.46	11.58	21.02	21.85
Traj. 3	8.41	1.61	2.67	1.21	12.96	23.52	18.14
Situation 2							
Joint	1	2	3	4	5	6	7
Traj. 1	8.85	2.34	3.90	3.06	21.20	31.70	26.75
Traj. 2	9.77	1.89	3.01	1.46	11.58	21.02	21.85
Traj. 3	8.41	1.61	2.67	1.21	12.96	23.52	18.14

3.3.6.2 IDIM-RIV vs IDIM-RLS

The objective of this subsection is to compare the state of art IDIM-RLS method and the IDIM-RIV presented in this chapter. As in the previous section, we highlight the importance of initial values now by analyzing the results of the IDIM-RLS method. As in the previous section, we will initialize recursive methods using the identified set of essential parameters of the model inside the industrial controller obtained in Section 2.3.6.1, and we will build the instrumental matrix of the IDIM-RIV method with the *commanded signals* made available by the controller. In addition, the trajectories are designed in the same way than in the previous subsection. For the commanded signals and the sensed signals when applying IDIM-RIV just a down-sample is applied without filtering. For the IDIM-RLS, the filter mentioned in the previous subsection is applied, as the necessary noise assumptions cannot be ensured.

Parameter initialization effect on IDIM-RLS. To evaluate the effect of parameters initialization in IDIM-RLS two situations are compared: when no a priori information is available and when the set of essential parameters from Table 2.1 is considered. Figure 3.3 shows the evolution of some identified parameters of joint 4 for these two situation. In the case of initializing the parameters with no a priori information, all values were started at 0 with a 100 in the diagonal of the matrix $\boldsymbol{P}$. When having a priori information, the essential parameters were started as in Table 2.1, and all the non-essential parameters (inertial and friction) at 0. Variance values were started at 0.01 for essential parameters, at 1 for the non-essential inertial parameters and at 100 for friction, answering to the different degrees of parameters' knowledge. The choice of the magnitude of the variance locks or lets free the parameters' variation [Gautier and Poignet, 2001].

In between 1000 and 2000 measurements, both methods converge to the same results. As it was expected, main inertial parameters' estimations converge faster when initializing the parameters with the identified model from Section 2.3.6.1. This is a substantial advantage

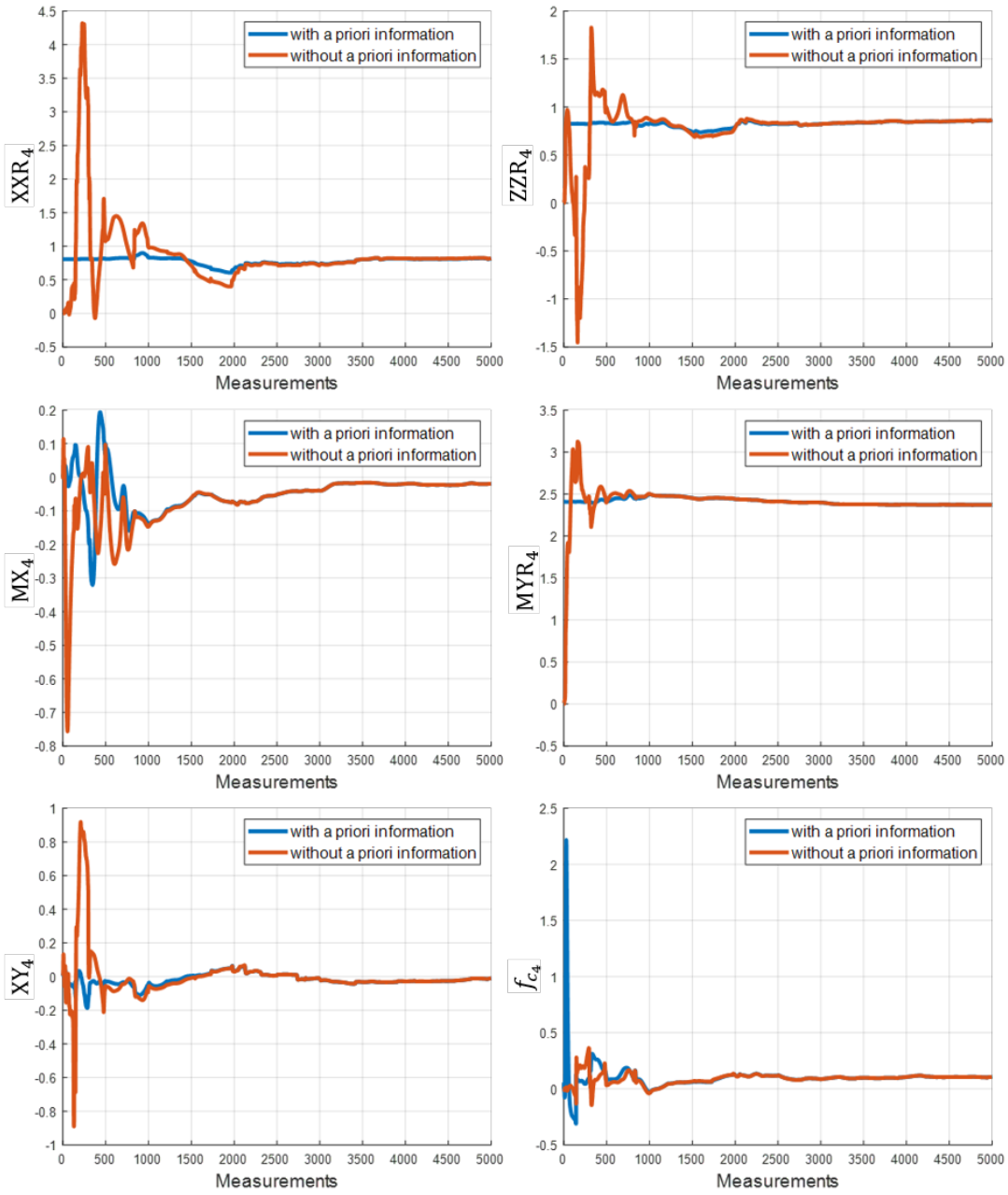


Figure 3.3: Evolution of some base parameters of joint 4 using IDIM-RLS with and without a priori information.

for the beginning of the execution, reducing the risk of having a non-computable dynamic model. However, friction parameters (f_{c_4} in Figure 3.3) vary significantly more, as they are initialized with much higher variance than the other parameters. Nevertheless, it cannot cause problems in terms of calculation because this issue comes from the non-invertibility of the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ in Equation (1.12) which friction does not affect.

These results show two things. First, initializing parameters in recursive algorithms with the values identified of the model that is integrated in the controller is better than initializing them with no a priori knowledge for an online identification context. This method also avoids the need of CAD models, which are often not provided by manufacturers. Furthermore, although the real robot parameters are not exactly the same as the ones inside the model of the controller, they are a good approximation, and they can be used to control the robot.

Comparison between IDIM-RIV and IDIM-RLS. To compare with IDIM-RLS results obtained in the previous subsection, IDIM-RIV results are analyzed. Parameter identification using IDIM-RLS with filtered sensed signals, and using IDIM-RIV with non-filtered sensed signals and commanded position for the instrumental matrix generation, are shown in Figure 3.4. For simplicity, some of the parameters of joint 4 shown in the previous subsection are depicted. Initial values for both methods are the ones shown in Table 2.1 obtained in Section 2.3.6.1.

Results are very close for both methods, and it is not possible to demonstrate with this study which is more precise, as the real values are unknown. However, the fact that results have a difference of less than 2% between each other, and that no filtering process was made to the signals in IDIM-RIV, presents a huge advantage of this method over others for online application.

3.3.7 Conclusion

In this section, a recursive instrumental variable method for industrial robots, called IDIM-RIV, has been presented and validated offline on the 7-*dof* industrial collaborative robot KUKA LBR iiwa 14 R820, identifying its 64 base parameters. We have shown that this method does not require the simulation of the direct dynamic model and is robust against noise and initialization. The construction of the instrumental matrix uses the inverse dynamic model and the references filtered by the closed-loop transfer function of position. We no longer need to know the controller since we identify the closed-loop transfer function with toolboxes, such as CAPTAIN. In the case of having the trajectory defined before execution, one calculates the instrumental matrix beforehand. Besides this transfer function acting as a unitary gain at low frequencies, we can construct the instrumental matrix directly with the references calculated online by the robot's controller (study which is complementary with the analysis done in Section 2.3.6.1 and Section B.3.4). Moreover, the identified parameters of the model that is inside the controller were used to initialize values in IDIM-RLS and IDIM-RIV methods, and it was shown that this approach is a proper alternative to arbitrary or

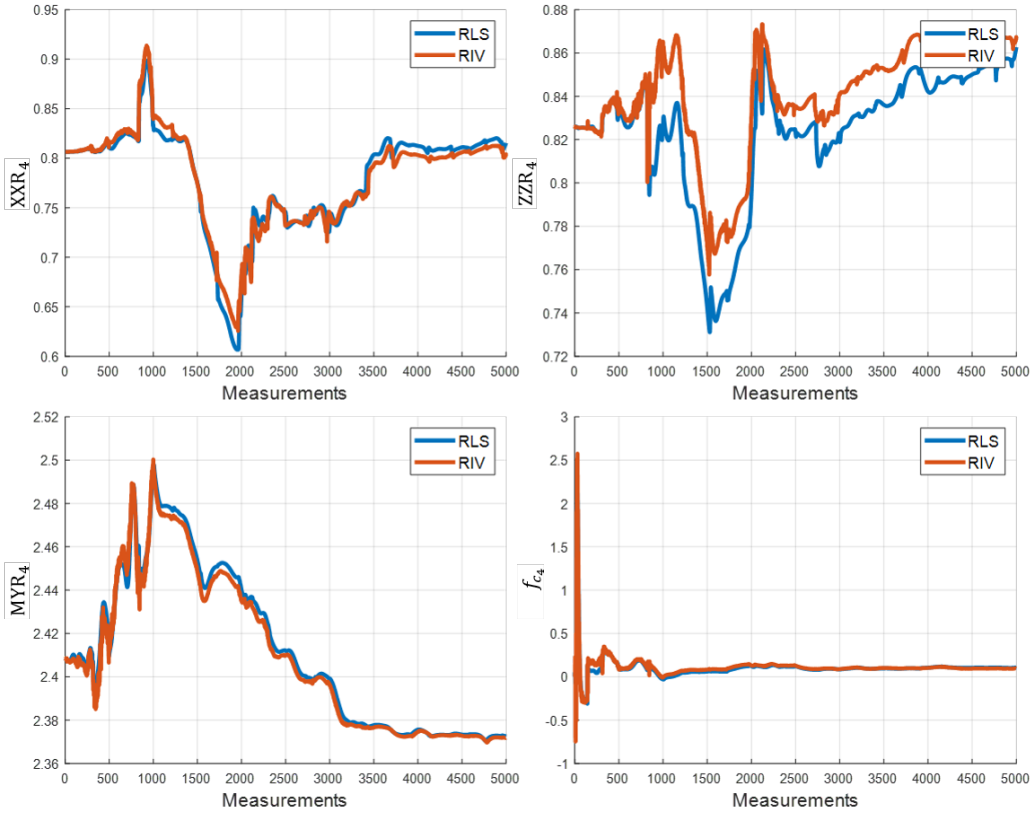


Figure 3.4: Evolution of some base parameters of joint 4 using IDIM-RLS and IDIM-RIV methods.

CAD-based initialization of parameters. Hence, this recursive instrumental variable method is relevant for online applications and tasks where trajectories change during execution or are unknown beforehand.

3.4 Line-by-line coupled identification

Up to the moment, all methods presented need at least one matrix inversion to compute the estimates. Inverting matrices is computationally complex (increasing with the amount of *dof* n) and may lead to instabilities in the algorithm. These two characteristics may not cause issues when carrying out offline identification, but can have a huge impact in online identification. In this section, we are going to use the line-by-line separation of the system used in sequential identification [De Wit and Aubin, 1990] and the parameter initialization tool of recursive methods to allow a propagation of information from one subsystem to another. Therefore, the current section is divided in four parts:

- In the preliminaries, we will present useful notation used in this section that adds up to the notation presented in Chapter 1.

- The next two parts present the new methods: the IDIM-CLS (IDIM - Coupled LS) and the S-IDIM-RLS (Sequential - IDIM - RLS). They are inspired on the works of [Ding, 2013] and [De Wit and Aubin, 1990]. In [Ding, 2013], the authors proposed the CLS (Coupled Least Squares) method. It is composed of two loops which allow a sequential identification by dividing the whole manipulator in different subsystems with their own input-output relation, including the variance propagation and yielding the same results than the RLS applied in a global way, requiring less computationally efforts (see the mentioned paper for more information). Furthermore, in [De Wit and Aubin, 1990], the authors presented a sequential identification process for robotics, called *line-by-line estimation*. Taking advantage of the fact that the regressor matrix is diagonal, they divided the system in n subsystems, one per joint (it has already been explained in Section 1.4.1.3 that there are other possible methods to divide the system to carry out sequential identification), leading to use only one row of the **IDM** matrix in each subsystem.

- Finally, we validate the methods experimentally on the KUKA iiwa manipulator.

3.4.1 Preliminaries

The regressor or observation matrix **IDM** in Equation (1.25) is known to be upper triangular, being able to re-write this equation in the form:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ ... \\ \tau_n \end{bmatrix} = \begin{bmatrix} \mathbf{idm}_{1,1} & \mathbf{idm}_{1,2} & \mathbf{idm}_{1,3} & ... & \mathbf{idm}_{1,n} \\ \mathbf{0} & \mathbf{idm}_{2,2} & \mathbf{idm}_{2,3} & ... & \mathbf{idm}_{2,n} \\ \mathbf{0} & \mathbf{0} & \mathbf{idm}_{3,3} & ... & \mathbf{idm}_{3,n} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ... & ... \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{idm}_{n,n} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta^1} \\ \boldsymbol{\beta^2} \\ \boldsymbol{\beta^3} \\ ... \\ \boldsymbol{\beta^n} \end{bmatrix}, \tag{3.11}$$

and,

$$\mathbf{idm}_i = \begin{bmatrix} \mathbf{0} & ... & \mathbf{idm}_{i,i} & \mathbf{idm}_{i,i+1} & ... & \mathbf{idm}_{i,n} \end{bmatrix}, \tag{3.12}$$

where τ_i is the torque value of joint i; $\mathbf{idm}_i$ (of components $\mathbf{idm}_{i,j}$) is the corresponding row vector of the observation matrix **IDM**; and $\boldsymbol{\beta^i}$ is the vector the parameters related with joint i, respectively. The sizes of these vectors will depend on the selected model. These equations show that the torque τ_i will depend on the vector of parameters $\boldsymbol{\beta^j}$, with $j = i, i+1, ..., n$. In other words, the torque in joint i will depend on the parameters of joint i and of higher-index joints.

3.4.2 The IDIM-CLS method

The coupled-least-squares (CLS) identification algorithm was first proposed in [Ding, 2013] for the purpose of avoiding the matrix inversion in the multivariable RLS algorithm (see Section 1.4.3). In this paper, it is shown that the CLS algorithm does not involve matrix inversion, requires less computational effort than the RLS method, and yields consistent estimates which converge to the true value as the amounts of measurements increase. In this section, we will shortly present its new version for robotic manipulators: the IDIM-CLS method.

The algorithm defined in Equations (27-32) in [Ding, 2013] is adapted to our notation of robotic manipulators and depicted in Figure 3.5 and in the following set of equations:

for $t = 1, ..., m$:

for $i = n, ..., 1$:

if $i == n$:

$$\begin{aligned} \boldsymbol{k}_n(t) &= \frac{\boldsymbol{P}_1(t-1)\mathbf{idm}_n^T(t)}{1 + \mathbf{idm}_n(t)\boldsymbol{P}_1(t-1)\mathbf{idm}_n^T(t)} \\ \boldsymbol{P}_n(t) &= \boldsymbol{P}_1(t-1) - \boldsymbol{k}_n(t)\mathbf{idm}_n(t)\boldsymbol{P}_1(t-1) \\ \hat{\boldsymbol{\beta}}_{CLS_n}(t) &= \hat{\boldsymbol{\beta}}_{CLS_1}(t-1) + \boldsymbol{k}_n(t)\left[\boldsymbol{\tau}_n(t) - \mathbf{idm}_n(t)\hat{\boldsymbol{\beta}}_{CLS_1}(t-1)\right], \end{aligned} \tag{3.13}$$

else:

$$\begin{aligned} \boldsymbol{k}_i(t) &= \frac{\boldsymbol{P}_{i+1}(t)\mathbf{idm}_i^T(t)}{1+\mathbf{idm}_i(t)\boldsymbol{P}_{i+1}(t)\mathbf{idm}_i^T(t)} \\ \boldsymbol{P}_i(t) &= \boldsymbol{P}_{i+1}(t) - \boldsymbol{k}_i(t)\mathbf{idm}_i(t)\boldsymbol{P}_{i+1}(t) \\ \hat{\boldsymbol{\beta}}_{CLS_i}(t) &= \hat{\boldsymbol{\beta}}_{CLS_{i+1}}(t) + \boldsymbol{k}_i(t)\left[\tau_i(t) - \mathbf{idm}_i(t)\hat{\boldsymbol{\beta}}_{CLS_{i+1}}(t)\right], \end{aligned} \tag{3.14}$$

where the matrices used here were already defined in Section 1.4.3; $\boldsymbol{k}_i(t) \in \mathbb{R}^{n_b \times 1}$ is the estimator gain vector; $\mathbf{idm}_i(t)$ is the *ith* row of the matrix $\mathbf{IDM}(t)$ obtained at time t; and $\hat{\boldsymbol{\beta}}_{CLS_{i+1}}(t)$ is the whole vector of estimates obtained with the use of the measurements of time t and subsystem $i+1$ (see Figure 3.5). The algorithm is composed of two loops: one outer loop depending on the measurement number $t = 1, ..., m$, and an inner loop depending on the subsystem number $i = n, ..., 1$. Note that inverting the loop from $i = n, ..., 1$ to $i = 1, ..., n$ and making the corresponding changes on the algorithm will yield the same final result. We have chosen to use the letter n in descending order because of two reasons: first, n is the same letter than the amount of *dof* of the manipulator, as it can be used to divide the system by joints, but other separations of the system can be used (as explained in Section 1.4.1.3); second, when carrying out sequential identification, authors in literature usually chose to start from joint n and compute the estimates until joint 1 taking advantage of the triangular feature of matrix **IDM**.

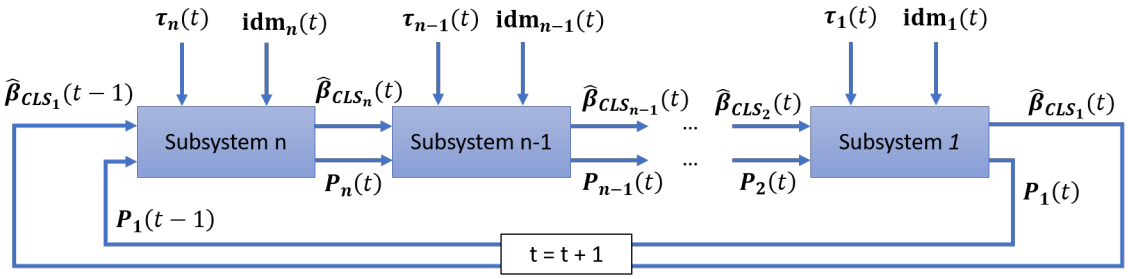


Figure 3.5: Schematic diagram of the CLS algorithm.

As said, the IDIM-CLS algorithm does not require matrix inversion for the computing of $\boldsymbol{k}_i(t)$, which is a major advantage over the IDIM-RLS (where the inverse of $(\mathbb{I}_n + \mathbf{IDM}(t)\boldsymbol{P}(t-1)\mathbf{IDM}^T(t))$ is necessary to compute the matrix of estimator gains $\boldsymbol{K}_i(t)$, see Equation (1.55)). As any other recursive method it requires of parameter initialization, and it has been shown in [Ding, 2013, Huang and Ding, 2017], that it yields the same results than the IDIM-RLS algorithm.

Furthermore, the IDIM-CLS is a method that can be applied online, having one main advantage: the user can choose which subsystem to update at each instant of time t. This means that the computational burden can be reduced by ignoring some specific subsystems, which are believed not to add new information at instant t.

3.4.3 The S-IDIM-RLS method

The IDIM-CLS treats the measurements made at instant t for all subsystems before treating the measurements of instant $t+1$. This method may not be suitable for situations as the one in *Model B* of Section 2.4 where the trajectories are made sequentially (one joint moves at the time). Here, the data obtained during the trajectory where joint i moves will just have an influence on the equations of subsystem i, and low or none effect on the others. This can be solved by changing the order of the two loops of the method. We then propose the S-IDIM-RLS (Sequential - IDIM - Recursive Least Squares) algorithm: the outer loop will now be due to the subsystems, and the inner loop will depend on the measurements. The method is equivalent to the IDIM-CLS and the IDIM-RLS yielding the same estimates. It uses all the measurements of subsystem i before using the measurements of subsystem $i-1$ and it allows to propagate the variance between different subsystems.

The algorithm scheme is depicted in Figure 3.6, and it is described as follows:

for $i = n, ..., 1$:

for $t = 1, ..., m$:

if $t == 1$:

$$\begin{aligned} \boldsymbol{k}_i(1) &= \frac{\boldsymbol{P}_{i+1}(m)\mathbf{idm}_i^T(1)}{1+\mathbf{idm}_i(1)\boldsymbol{P}_{i+1}(m)\mathbf{idm}_i^T(1)} \\ \boldsymbol{P}_i(1) &= \boldsymbol{P}_{i+1}(m) - \boldsymbol{k}_i(1)\mathbf{idm}_i(1)\boldsymbol{P}_{i+1}(m) \\ \hat{\boldsymbol{\beta}}_{SRLS_i}(1) &= \hat{\boldsymbol{\beta}}_{SRLS_{i+1}}(m) + \boldsymbol{k}_i(1)\left[\boldsymbol{\tau}_i(1) - \mathbf{idm}_i(1)\hat{\boldsymbol{\beta}}_{SRLS_{i+1}}(m)\right], \end{aligned} \tag{3.15}$$

else:

$$\begin{aligned} \boldsymbol{k}_i(t) &= \frac{\boldsymbol{P}_i(t-1)\mathbf{idm}_i^T(t)}{1+\mathbf{idm}_i(t)\boldsymbol{P}_i(t-1)\mathbf{idm}_i^T(t)} \\ \boldsymbol{P}_i(t) &= \boldsymbol{P}_i(t-1) - \boldsymbol{k}_i(t)\mathbf{idm}_i(t)\boldsymbol{P}_i(t-1) \\ \hat{\boldsymbol{\beta}}_{SRLS_i}(t) &= \hat{\boldsymbol{\beta}}_{SRLS_i}(t-1) + \boldsymbol{k}_i(t)\left[\boldsymbol{y}_i(t) - \mathbf{idm}_i(t)\hat{\boldsymbol{\beta}}_{SRLS_i}(t-1)\right], \end{aligned} \tag{3.16}$$

In Figure 3.6, $\hat{\boldsymbol{\beta}}_{\mathrm{SRLS_{init}}}$ and $\boldsymbol{P}_{\mathrm{init}}$ refer to the initial values, the observation matrices are defined as: $\boldsymbol{W_i} = \left[\mathbf{idm}_i^T(1)\ \mathbf{idm}_i^T(2)\ ...\ \mathbf{idm}_i^T(m)\right]^T$ and the vector of torques $\boldsymbol{y}_i = [\tau_i(1)\ \tau_i(2)\ ...\ \tau_i(m)]^T$.

Both of the presented methods can be applied online. The choice between the use of the S-IDIM-RLS and the IDIM-CLS will just depend on how the measurements are made. For instance, if the trajectory is defined in order to first excite subsystem n, then subsystem n-1 and subsequently, then the S-IDIM-RLS method is more adequate. On the other hand, if all subsystems are being excited at the same time, the IDIM-CLS would be the choice.

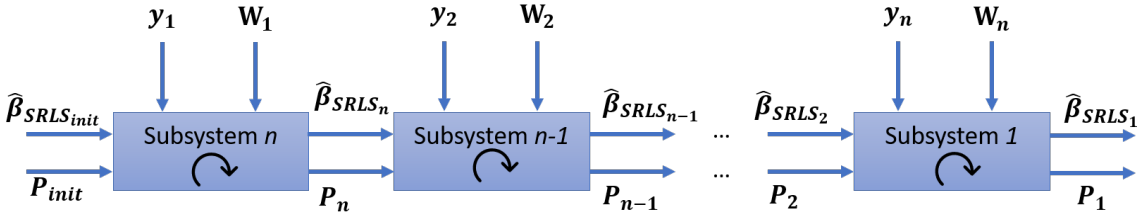


Figure 3.6: Schematic diagram of the S-IDIM-RLS algorithm.

3.4.4 Experimental results

For the experimental validation, the KUKA iiwa manipulator is used. The robot, its framework and data processing steps are explained in Appendix B and Appendix C. The model chosen is presented in the mentioned appendixes and it is completed with the 3 parameters friction model of Equation (1.14). The *Global_PTP_2* filtered trajectory is used for identification, and the filtered *Global_DS* and *Global_Spline_1* trajectories for validation.

Table 3.3 shows the results of the torque reconstruction of several methods: S-IDIM-RLS, IDIM-CLS, IDIM-RLS, the method in [De Wit and Aubin, 1990] and the S-IDIM-RLS-B. This last method is a version of the S-IDIM-RLS where in each subsystem just the parameters related to that subsystem are updated (for subsystem i, just $\boldsymbol{\beta}^i$ is updated). It is similar to the method in [De Wit and Aubin, 1990] with the difference that the matrix $\boldsymbol{P}$ is obtained as in the S-IDIM-RLS method and propagated throughout the whole process. Moreover, the method presented in the mentioned paper is equivalent to carrying out a sequential LS from joint 7 up to the first 1 considering the estimates obtained in previous steps as fully known.

We can observe that there is no much difference between all the methods' results. Although the S-IDIM-RLS, IDIM-CLS and IDIM-RLS are slightly better, we can say that there is no difference in performance between all the methods. This happens because the chosen trajectory is well-conditioned and will lead to precise estimates. Notice that the trajectories are global, thus all joints are moving at the same time, but the identification is carried out sequentially (except in IDIM-RLS that is carried out globally). Thus, the accumulation error in sequential identification produced by method [De Wit and Aubin, 1990] and S-IDIM-RLS-B because of considering the parameters as totally known for subsequent identification is minimal. The difference between these two methods is that, in S-IDIM-RLS-B, we propagate the values of matrix $\boldsymbol{P}$. As we are using filtered signals and the regressor matrix is well-conditioned, it does not have a big impact on the result.

Second, note that the S-IDIM-RLS, IDIM-CLS and IDIM-RLS have the same result (which was already verified in [Ding, 2013] for the CLS method). However, the first two methods have the advantage of not needing a matrix inversion. Moreover, they divide the system in different subsystems which is of advantage for online applications. This can allow to decide which subsystem to update given the measurements of time k. For example, if joint 7 is not moving nor accelerating at time k, but all the other joints are moving, then, it is possible to delete from the loop at time k the update of subsystem 7. This can have a good impact on

real-time application. First, it gives a way not to update parameters that are being under-excited which can lead to wrong results and algorithm instability. Second, not updating several subsystems at each loop, can reduce the computational time abruptly.

Table 3.3: Percent error [%] of torque reconstruction of models identified with the IDIM-RLS, S-IDIM-RLS, IDIM-CLS and the method in [De Wit and Aubin, 1990].

Joint	1	2	3	4	5	6	7
Global_Spline_1							
S-IDIM-RLS, IDIM-CLS and IDIM-RLS	22.02	2.56	3.40	0.78	10.60	21.38	22.83
S-IDIM-RLS-B	22.54	2.65	3.56	0.84	12.36	19.98	20.62
Method in [De Wit and Aubin, 1990]	22.78	2.64	3.55	0.84	12.71	20.02	20.62
Global_DS							
S-IDIM-RLS, IDIM-CLS and IDIM-RLS	29.94	1.71	2.65	1.14	11.18	17.70	31.35
S-IDIM-RLS-B	30.66	1.92	2.93	1.08	11.92	15.20	30.85
Method in [De Wit and Aubin, 1990]	30.44	1.83	2.79	1.10	12.61	15.23	30.85

3.4.5 Conclusion

In this section, we have developed two different methods for robotic identification: the IDIM-CLS and S-IDIM-RLS. Both of them are based on the CLS method, and are applicable in different situations. They have appealing features for online application. They divide the system in different subsystems, avoiding the matrix inversion, and allowing the possibility to choose which of them to update depending on the input and output measurements. These two characteristics can reduce the computational efforts of the method and make it more stable. More studies regarding statistical and performance analysis of the methods are to be done.

We notice that the loss of precision of sequential methods is insignificant if the trajectory is well-conditioned. This leaves two open remarks. First, it leads to think that the problem of sequential identification is not the method per se, but the poor excitation of parameters when carrying out sequential trajectories. One solution is to use trajectories as the ones in [Mayeda et al., 1984, Vandanjon et al., 1995], to decouple effects of the manipulator, and analyze how the proposed methods work and compare them with global methods. Second, the other problem of sequential identification is considering as known estimates of previous estimations, while, in fact, they come in the form of statistical distribution. Once the information of how well the estimates predict the input on a previous step is included in the method, then, theoretically, the only issue in sequential identification is the mentioned previously. This is not addressed in this work, although future studies are planned.

3.5 Final conclusion

In this chapter, the topic of offline recursive identification was covered. As explained in Chapter 1, all the measurements are available and processed before the identification method is applied, thus there is no computational time limit. However, the formulation of these methods is suitable for online application which will be the subject of Chapter 4.

We developed three new methods/algorithms for offline recursive identification. First, the IDIM-RIV was developed and enhanced. It is an algorithm more robust to noise than the IDIM-RLS and posses some interesting properties for the online application. We tackled the issue of the construction of the IV matrix avoiding the simulation of the DDM presented in Chapter 2, which may be time consuming. Moreover, we proposed a closed-loop method in which we estimated the transfer function of the controller, in order to be able to estimate the model of the manipulator using the reference signal and the measurements. This is an attractive method specially to be applied to industrial manipulators, where usually no much information about the controller and the model of the manipulator is given, and not all needed measurements are available for the user. Second, two different algorithms were derived (the IDIM-CLS and S-IDIM-RLS). They divide the whole system in different subsystems in order to carry out the identification. They can be applied both for sequential and global trajectories, and are applicable in an online basis. The idea behind these algorithms was motivated by the fact that the use of sequential estimation methods is usually discouraged against the global estimation methods due to the accumulation of error in subsequent estimations.

As said, these methods are suitable to be applied online, and they have a lot of utility in robotics as they allow the implementation of features as the adaptive control, fault/collision detection, and interaction estimations. The next chapter intends to apply some of the methods derived in this section in different special cases.

Chapter 4

Online identification

Contents

4.1 Introduction 97
4.2 Problem statement and related works 98
4.3 The IDIM-CIV method 102
4.3.1 Algorithm 102
4.4 Adding the forgetting factor 103
4.5 Experimental framework 105
4.6 Case study I: simple trajectory 106
4.7 Case study II: movement with constant payload 109
4.8 Case study III: movement with changing payload 112
4.9 Case study IV: physical Human Robot Interaction 116
4.10 Final conclusion 118

4.1 Introduction

Chapter 3 focused on the development of recursive estimation methods because they are mandatory for online estimation of the model. However, in the previous chapter they were tested and validated in an offline way. Chapter 4 studies their performance on different real online scenarios using the KUKA LBR iiwa 14 R820 collaborative manipulator. The chapter is presented as follows:

- First, similar to previous chapters, the problem and related works are discussed. Three topics are considered: online identification, payload identification and human interaction.
- Second, a new identification method is presented: the IDIM-CIV. This method is directly derived by combining the IDIM-RIV and IDIM-CLS shown in Chapter 3. It has interesting properties of noise rejection and online computation. Moreover, the forgetting factor is introduced, as it is necessary to allow variations on the estimates when the system is under changes and/or interactions.

- Third, the experimental framework is described. It is a short extension for online applications of the framework presented in Appendix B.
- Finally, several scenarios are studied:
  - *Case study I*: it is a simple scenario where the robotic manipulator moves without load, as in the trajectories that were used all along the manuscript.
  - *Case study II*: a fixed constant payload is attached to the tip of the manipulator.
  - *Case study III*: during the movement, the payload placed on the tip of the robot changes, simulating what would happen in a pick and place or transportation application.
  - *Case study IV*: during the movement, there is a human interacting with the manipulator.

4.2 Problem statement and related works

The online identification. For any application where it is needed to know how the model, or part of it, evolves during time, online methods are needed. Examples of these in robotics is the use of adaptive control [Balestrino et al., 1984, Slotine and Li, 1987]. There are several challenges that arise with the real-time implementation. First, that not all the measurements are available at the same time, limiting the choice of possible filtering procedures. Related to this, the second issue is that it brings up the challenge of limited computation time and algorithm stability. The algorithm needs to be computed fast enough to be able to track different changes, at a rate that will depend on the application, allowing to take suitable decisions before the system fails to do what it is supposed to. Third, the robot is usually following a trajectory that is not intended for identification, which leads to under-excitation of parameters and ill-conditioned regressors. If this is not treated correctly, the estimated model may significantly differ from reality and fail to grab the general behavior of the manipulator.

One possible technique for online applications is batching [Barfoot, 2017]. This technique takes the measurements done in a period of time, and applies either en-bloc or recursive methods to estimate the parameters. Although it can be effective, it is a solution proposed in order to avoid issues due to treating measurements one at a time and in order to loosen the computation time constraint. The issue with this technique is that estimates may not be computed fast enough for certain applications.

Another technique is to apply recursive methods to treat one measurement at a time in real-time. In [Kubus et al., 2007, Kubus et al., 2008b, Kubus et al., 2008a], the authors applied the RLS, RIV and RTLS methods in an online way on the Stäubli RX60 industrial manipulator, equipped with a force/torque sensor, in order to identify the payload parameters with a pre-designed trajectory. In [Farsoni et al., 2018], they applied the same method to estimate the payload on a 6-*dof* FANUC LR Mate 209iD/7L including a quaternion-based KF to estimate velocities and accelerations. In [Kurdas et al., 2022], the authors proposed an online payload identification approach based on momentum observers via the RLS and

RTLS methods on the collaborative Franka Emika Panda manipulator equipped with a two-finger gripper. In [Hu et al., 2020], the authors estimated a set of essential parameters of the robot in an offline way, in order to design an adaptive robust control and generate a momentum observer capable of estimating the payload's parameters in real-time. They tested it experimentally on the 6-*dof* COMAU-RACER3 industrial robot.

However, in all these works, the authors just identify a constant payload in real-time while they consider the model of the manipulator as completely known before-hand. This would be a suitable method for situations where there is no other change but a constant load on the end-effector. If the payload is meant to change during execution, these algorithms may fail to predict it. Nevertheless, even in the constant payload situation, we can doubt if parameters as the ones related to the friction or the joint flexibility change when the payload is modified. Furthermore, for situations where humans interact with the manipulator, where payloads are not placed on the end-effector of the manipulator or for carrying fault/collision detection algorithms, the knowledge of how all the parameters of the dynamic model of the manipulator (or the coupled model between the manipulator and the other system which is interacting with it) evolve during time may be useful.

In this chapter, we are going to present the IDIM-CIV (IDIM - Coupled IV) method in order to track the evolution of the parameters in an online basis. This method is a priori attractive for several reasons: it is robust against noise as any other IV approach, which leads to a less restrictive filtering stage, thus reducing the computation time; it does not require matrix inversion typical of coupled algorithms, which makes it more stable; and it divides the system in several subsystems, which allows not to compute certain estimates when it is considered that measurements related to them do not add information, making it even faster.

Moreover, as this method is going to be applied on a system whose model may change over time, the forgetting factor needs to be introduced [Young, 2011]. It allows the method to track changes and not to be stacked on values that were previously identified. It equips the algorithm with a certain *memory*, in which old measurements have less influence than new ones. It is an hyper-parameter that the user needs to choose. An interesting work related to this topic is the one presented in [Vahidi et al., 2005]. During this chapter, the influence of different forgetting factors on the algorithm's performance will be studied.

Identification of payload. For almost every application in robotics, the manipulator will be asked to carry a tool or a load to execute its task (both terms will be regrouped under the word *payload*). Sometimes this payload can be quite significant in relation to the manipulator's structure, and not including it in the controller may lead to a huge loss of precision or even to failure. If the payload's parameters are known, then its addition to the model is straightforward. However, if they are unknown and it is not possible to carry out mechanical tests on it before-hand, an identification process should be carried out. The survey in [Mavrakis and Stolkin, 2020] summarises several works on how to estimate the object's inertial parameters in robotic grasping and manipulation. There, the authors divide the methods in three: visual methods, where cameras are needed; exploratory methods, where the object is estimated by different interactions with the manipulator as pushing, poking and tilting;

and fixed-object methods, where the object is fixed to the manipulator and included in its dynamic model. Although all of them are interesting, the later is the scenario which is going to be tested in this chapter. The fact of adding a payload to the tip of the manipulator will lead to changes in the dynamic model, to larger joint torques and to changes in the volume of the manipulator which will lead to different safety characteristics changes [Hamad et al., 2019]. Knowing the payload's features is essential to improve the manipulator's performance.

Load identification has been treated in several works like those in [Atkeson et al., 1986, Kawasaki and Nishimura, 1988, Kozlowski and Dutkiewicz, 1996, Swevers et al., 2002, Gaz and De Luca, 2017, Bahloul et al., 2018, Dong et al., 2018]. The work done in [Khalil et al., 2007] summarizes the techniques used in most of these papers and enumerates four methods to estimate the inertial parameters of a payload fixed on the terminal link of the robot. The *en-bloc* model in Equation (1.26) can be re-described as:

$$\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\beta} + \boldsymbol{W_L}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\beta_L} + \boldsymbol{\rho}, \tag{4.1}$$

where $\boldsymbol{W_L}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the observation matrix corresponding to the 10 payload's inertial parameters $\boldsymbol{\beta}_L$. The four methods, which are regrouped in three categories, can be summarized as:

- *Making use of the values estimated without payload:* if $\boldsymbol{\beta}$ is already known, the payload parameters can be estimated in two ways. The first is directly obtained from Equation (4.1), in the form of the LS solution as:

$$\hat{\boldsymbol{\beta}}_L = \boldsymbol{W_L^+}(\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}). \tag{4.2}$$

  The second one consists in identifying the whole system as done in Chapter 2 and Chapter 3, obtaining the payload values by simple analysis of the equations of the base parameters and subtracting the already known values of the manipulator:

$$\Delta\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{wl} - \hat{\boldsymbol{\beta}}, \tag{4.3}$$

  where $\Delta\hat{\boldsymbol{\beta}}$ is the variation of inertial parameters from which $\hat{\boldsymbol{\beta}}_L$ can be obtained; and $\hat{\boldsymbol{\beta}}_{wl}$ is the set of base parameters estimated when the robot is carrying a load.

- *Using the difference between the joint torques before and after loading the robot on the same trajectory:* if we consider that the control system is efficient enough, there is no need to estimate the parameters of the manipulator. It is sufficient to measure the torque in the same trajectory with ($\boldsymbol{y}_{wl}$) and without ($\boldsymbol{y}_{wol}$) load:

$$\hat{\boldsymbol{\beta}}_L = \boldsymbol{W_L^+}(\boldsymbol{y}_{wl} - \boldsymbol{y}_{wol}). \tag{4.4}$$

- *Global identification of the robot parameters and the load parameters:* this method is presented to avoid sequential identification methods due to its supposedly accumulation of errors already discussed in Chapter 3. It needs to carry out trajectory a without payload and trajectory b with payload, and regroups them in one system of equations

in the form:

$$\begin{bmatrix} \boldsymbol{y_a} \\ \boldsymbol{y_b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W_a} & \boldsymbol{0} \\ \boldsymbol{W_b} & \boldsymbol{W_{bL}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\beta}_L \end{bmatrix}. \tag{4.5}$$

If we consider that parameters as the friction or joint flexibility are dependant on the payload, the methods need to be redefined to include them. Moreover, several works as [Atkeson et al., 1985, Dutkiewicz et al., 1993, Kubus et al., 2008b] added a force-torque sensor on the tip of the robot to identify the payload. This usually facilitates the task by adding more equations to the system in the form of:

$$\begin{bmatrix} \boldsymbol{f_{ft}} \\ \boldsymbol{\tau_{ft}} \end{bmatrix} = \boldsymbol{W_{pl}} \boldsymbol{\beta}_L, \tag{4.6}$$

where $\boldsymbol{f_{ft}}$ and $\boldsymbol{\tau_{ft}}$ are the measured forces and torques by the force-torque sensor; and $\boldsymbol{W_{pl}}$ is the corresponding regressor matrix that relates the measured linear and angular velocities and accelerations with the 10 inertial parameters of the payload.

In this work, we will make use of the values estimated without payload from Chapter 2 and Chapter 3, and of the method in Equation (4.3). The calculation of the physical parameters of the payload will be useful to validate the methods as we can measure some of them to know their true value.

Human interaction. Besides a changing payload, collaborative robots are expected to work alongside with humans. Although the interaction is usually expected to happen in the tip of the robot, it may also happen in any part of the manipulator in unexpected ways. This brings up several questions: how to differentiate between a wanted contact and an undesirable contact or collision? how to differentiate between a human force and the effect produced by a payload? is it possible to model the human? Works as [De Luca et al., 2006, Haddadin et al., 2008, Haddadin et al., 2009, De Luca and Flacco, 2012, Haddadin et al., 2017] treat the issue of collision detection for collaborative robots based on physical quantities such as total energy and generalized momentum of the manipulator, by estimating the external torque via observers. This estimation of forces of interaction is a common activity in collaborative robotics, and is also a tool that is being investigated to try to predict what is the human intending to do [Erden and Tomiyama, 2010, Dumora et al., 2013]. This is useful in applications such as human assistance, where the robot needs to help the human, and knowing what he/she wants to do before-hand is of vital importance. It is also of importance in co-manipulation tasks, which are also being thoroughly studied [Hayashibara et al., 1999, Dumora et al., 2012, Lawitzky et al., 2010, Mujica et al., 2023].

Nowadays, regarding identification, much attention is being paid on how to model the human in the human-robot interaction. In [Artemiadis et al., 2010], the authors estimated 27 parameters of the upper limb impedance characteristics of a human interacting with a 7-*dof* manipulator by means of the LS method. The works in [Erden and Billard, 2014, Ajoudani et al., 2018a] studied the end-point impedance and stiffness values of human arms for applications as welding and drilling. In [Gomi and Kawato, 1997], the authors studied

the human multi-joint arm stiffness parameters by making the person move a robotic arm on a horizontal plane. The authors in [Takagi et al., 2017] estimated a model to explain human behaviours in order to be applied in collaborative robots that provide human-like assistance.

Although this manuscript is not focused on pHRI (physical Human Robot Interaction) and does not intend to answer all the previously mentioned open questions, nor modeling the human side, a scenario where there is interaction with the human is included. This is done to analyze the parameter identification method developed in this chapter and determine if the information obtained could be potentially used in cobotics.

4.3 The IDIM-CIV method

Having introduced the IDIM-CLS method in Section 3.4.2, and the IDIM-RIV in Section 3.3, we can deduce the new IDIM-CIV (IDIM - Coupled IV) method. It takes the form of:

for $t = 1, ..., m$:

for $i = n, ..., 1$:

if $i == n$:

$$\begin{aligned} \boldsymbol{k}_n(t) &= \frac{\boldsymbol{P}_1(t-1)\boldsymbol{z}_n^T(t)}{1 + \mathbf{idm}_n(t)\boldsymbol{P}_1(t-1)\boldsymbol{z}_n^T(t)} \\ \boldsymbol{P}_n(t) &= \boldsymbol{P}_1(t-1) - \boldsymbol{k}_n(t)\mathbf{idm}_n(t)\boldsymbol{P}_1(t-1) \\ \hat{\boldsymbol{\beta}}_{CIV_n}(t) &= \hat{\boldsymbol{\beta}}_{CIV_1}(t-1) + \boldsymbol{k}_n(t)\left[\tau_n(t) - \mathbf{idm}_n(t)\hat{\boldsymbol{\beta}}_{CIV_1}(t-1)\right], \end{aligned} \tag{4.7}$$

else:

$$\begin{aligned} \boldsymbol{k}_i(t) &= \frac{\boldsymbol{P}_{i+1}(t)\boldsymbol{z}_i^T(t)}{1 + \mathbf{idm}_i(t)\boldsymbol{P}_{i+1}(t)\boldsymbol{z}_i^T(t)} \\ \boldsymbol{P}_i(t) &= \boldsymbol{P}_{i+1}(t) - \boldsymbol{k}_i(t)\mathbf{idm}_i(t)\boldsymbol{P}_{i-1}(t) \\ \hat{\boldsymbol{\beta}}_{CIV_i}(t) &= \hat{\boldsymbol{\beta}}_{CIV_{i+1}}(t) + \boldsymbol{k}_i(t)\left[\tau_i(t) - \mathbf{idm}_i(t)\hat{\boldsymbol{\beta}}_{CIV_{i+1}}(t)\right]. \end{aligned} \tag{4.8}$$

This method will take the good properties of IV approaches regarding noise rejection and fewer requirements on the filtering stage, and the characteristics of coupled approaches, where the system is divided in several sub-systems and there is no matrix inversion needed. These features make it an attractive approach for online application.

4.3.1 Algorithm

The online IDIM-CIV method can be applied by the following algorithm:

- **Step 0:** initialize $\mathbf{P}$ and $\hat{\boldsymbol{\beta}}_{CIV}$. Use a priori values if available. Otherwise, initialize with zeros the parameters and high values for the covariance matrix.

- **Step *it*:** take the measurements of instant it, compute the velocities and accelerations with numerical differentiation and apply the following algorithm:

  for $i = n, ..., 1$:

  Compute the vectors $\mathbf{idm}_i$ and $\boldsymbol{z}_i$ and obtain the estimates with:

$$\begin{aligned} \boldsymbol{k} &= \frac{\boldsymbol{P}\boldsymbol{z}_i^T}{1 + \mathbf{idm}_i \boldsymbol{P}\boldsymbol{z}_i^T} \\ \boldsymbol{P} &= \boldsymbol{P} - \boldsymbol{k}\ \mathbf{idm}_i \boldsymbol{P} \\ \hat{\boldsymbol{\beta}}_{CIV} &= \hat{\boldsymbol{\beta}}_{CIV} + \boldsymbol{k}\left[\tau_i - \mathbf{idm}_i \hat{\boldsymbol{\beta}}_{CIV}\right] \end{aligned} \tag{4.9}$$

Conditions on *Step it* can be included in order to avoid the computation of certain iterations of the for loop if it is considered that the measurements at that iteration do not provide important information.

4.4 Adding the forgetting factor

The forgetting factor is important for online applications as it prevents the algorithm from getting stacked on values that were already identified. In the classical RLS, the covariance matrix vanishes to zero with time, losing its capability to keep track of changes in the parameters. The fact of adding a forgetting factor, slows down the fading out of the covariance matrix. It is an hyper-parameter to be chosen by the user, and it can be a way to indicate how quick some parameters are desired to react against changes [Vahidi et al., 2005].

In all recursive methods that were presented during this manuscript, the forgetting factor λ can be included in the same way [Young, 2011]. For example, if we add λ to the IDIM-RLS method, presented in Section 1.4.3, it becomes:

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_{k-1}\mathbf{IDM}_k^T(\lambda \mathbb{I}_n + \mathbf{IDM}_k \boldsymbol{P}_{k-1}\mathbf{IDM}_k^T)^{-1} \\ \boldsymbol{P}_k &= \frac{1}{\lambda}(\boldsymbol{P}_{k-1} - \boldsymbol{K}_k \mathbf{IDM}_k \boldsymbol{P}_{k-1}) \\ \hat{\boldsymbol{\beta}}_{RLS}^k &= \hat{\boldsymbol{\beta}}_{RLS}^{k-1} + \boldsymbol{K}_k(\boldsymbol{\tau}_k - \mathbf{IDM}_k \hat{\boldsymbol{\beta}}_{RLS}^{k-1}) \end{aligned} \quad . \tag{4.10}$$

The forgetting factor defined in this way works as an algorithmic memory that dies away into the past in an exponential fashion. If $\lambda = 1$, the infinite-memory RLS algorithm showed in Chapter 1 is obtained. Furthermore, it has been shown that, in practice, it is better to replace the constant λ by a variable $\lambda(k)$ which is initially smaller than λ but approaches it asymptotically [Young, 2011]. In this way, the memory at the beginning is small, and it grows as more samples are processed. This is helpful to improve convergence speed, as the

first estimations may present important variations due to the small number of measurements and the effect of initial conditions. One possible definition for the forgetting factor is:

$$\lambda(k) = \alpha_0\lambda(k-1) + (1-\alpha_0)\lambda, \tag{4.11}$$

where $\lambda(k)$ is the forgetting factor at iteration k; $\lambda(0)$ and α_0 are coefficients to be defined by the user (they have typical values of 0.95 and 0.99 respectively); and λ is the expected final value (note that if the value is too low, it leads to unstable estimators).

RLS with forgetting factor has been widely used in estimation and tracking of time-varying parameters. However, as said in [Vahidi et al., 2005], whenever the excitation of the system is poor, this scheme can lead to the covariance *wind-up* problem or estimator *blow-up*. During the moments that the system is under poor excitation, old information is progressively forgotten, while the new measurements with no dynamic information comes in. This may lead to an exponential grow of the covariance matrix, which results in the estimates to be extremely sensitive and prone to errors. Moreover, the method assumes that all the parameters vary with similar rates, not allowing the algorithm to realize if the error is due to one or more parameters. For example, in the case of a drift of one parameter, the correction applied to all of the parameters is of the same order as the one that is growing exponentially, which may lead to a *blow-up* of all the estimators. Works have proposed different techniques to avoid this problem: by introducing bounds to the values of the covariance matrix, adopting time variant forgetting factors (close to 1 in moments of poor excitation) in [Fortescue et al., 1981, Rao Sripada and Grant Fisher, 1987], resetting the covariance matrix during low excitation in [Salgado et al., 1988], or using the concept of directional forgetting in [Hägglund, 1985].

The mentioned issues can occur when we are estimating multiple parameters where some vary at different rates or are being excited differently. A single forgetting factor may not be able to track all the changes. One solution is to assign different forgetting factors to different parameters. This method is called vector-type forgetting or selective forgetting [Saelid and Foss, 1983, Sælid et al., 1985, Parkum et al., 1990, Fraccaroli et al., 2015], and it may be suitable for our application in robotics. For example, if the robot is going to interact just by its end-effector, we already know which are the parameters of the model that should change. Thus, by modifying the forgetting factor related to each parameter, we can equip the algorithm with this information. The equations of the IDIM-RLS become (from [Sælid et al., 1985, Yoshitani and Hasegawa, 1998]):

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_{k-1}\mathbf{IDM}_k^T(\mathbb{I}_n + \mathbf{IDM}_k\boldsymbol{P}_{k-1}\mathbf{IDM}_k^T)^{-1} \\ \boldsymbol{P}_k &= \Lambda^{-1}(\boldsymbol{P}_{k-1} - \boldsymbol{K}_k\mathbf{IDM}_k\boldsymbol{P}_{k-1})\Lambda^{-1} \\ \hat{\boldsymbol{\beta}}_{RLS}^k &= \hat{\boldsymbol{\beta}}_{RLS}^{k-1} + \boldsymbol{K}_k(\boldsymbol{\tau}_k - \mathbf{IDM}_k\hat{\boldsymbol{\beta}}_{RLS}^{k-1}) \end{aligned}, \tag{4.12}$$

where $\Lambda \in \mathbb{R}^{n_b \times n_b}$ is a diagonal matrix with the values of the forgetting factors of each parameter $\lambda_1, \lambda_2, ..., \lambda_{n_b}$ in the diagonal.

4.5 Experimental framework

This section briefly describes the framework used for the real-time estimation of parameters using the KUKA iiwa manipulator. The basis of this framework is thoroughly described and discussed in Appendix B for offline applications. We will duplicate some explanations for reading clarity.

Analogically to Figure B.5 (see Appendix B), Figure 4.1 shows the framework for the online application of the algorithm on the KUKA iiwa. As in any other application for this manipulator, the user must code a JAVA program on the KUKA controller, which operates on the SUNRISE operative system (also created by KUKA), with the desired behavior and trajectory to follow. This JAVA application communicates with a C/C++ application (named *master application*) running on the external computer (see Appendix B.1.4), via a channel performed using the *Fast Robot Interface* (FRI) (a version of the one presented in [Schreiber et al., 2010]), which is an open-source library for remote control of some KUKA robots. It allows deterministic access to information on joint position and axis torque at a rate of 1-100 ms, and also provides the tools to direct path manipulation at a rate of 1-4 ms. Moreover, another C/C++ application, running on the same computer, is in charge of the communication with the F/T sensor, and writes the respective measurements on a shared space memory. The *master application* is in charge of writing the measurements, recieved from the controller, on the shared memory space, as well as saving a log file with the measurements from the robot and the F/T sensor (see Appendix B.1.1 and Appendix B.1.3). Finally, a Python Application is included in order to estimate in real-time the parameters. This application makes use of the measurements available on the shared memory, of well-known libraries as *numpy* and *sysv_ipc*, and of a symbolic definition of the regressor matrix obtained from MATLAB (or other software capable of working in a symbolic way). This application shows in real-time how parameters change, and saves these values on a log for posterior treatment.

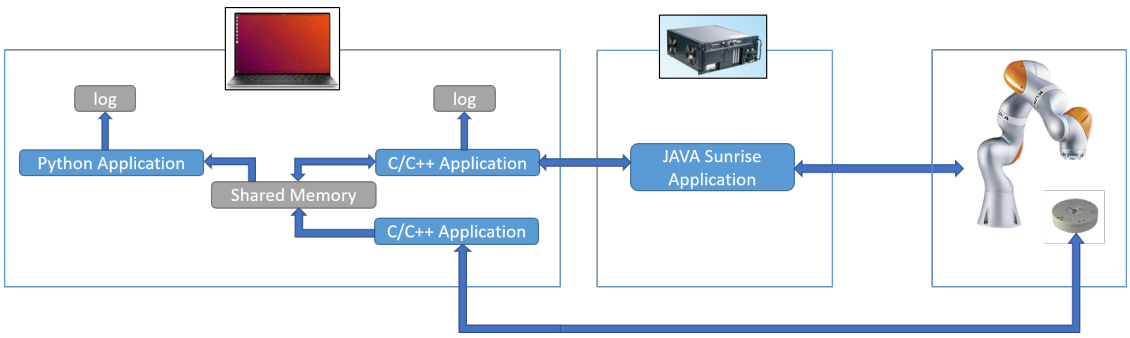


Figure 4.1: Framework for online estimation of paramters of the KUKA iiwa manipulator.

Besides initial definitions, the Python Application is mainly a loop that repeats until the manipulator's application stops. The loop is composed by the main following steps, which may slightly change depending on the application:

- Reading of the measured signals (*sensed torque*, *sensed position*, *commanded torque* and *commanded position*) from the Shared Memory Space.
- Calculating velocities and accelerations by numerical differentiation of measurements (after the filtering stage if needed)
- If the IDIM-CIV is being implemented: computing an updating in a *for* loop $\mathbf{idm}_i$, $\boldsymbol{z}_i$ and the estimates.
- Writing results in a *log* file.

4.6 Case study I: simple trajectory

The first situation that will be studied is when the collaborative robot does not interact with the environment and does not have any payload. It is the same situation that has been treated all along the manuscript, but this time is analyzed in an online way applying the IDIM-CIV. The objective of this section is to show how the estimates of the parameters evolve during time, and if the algorithm is fast enough to keep up with the real-time constraint. For length reasons, just some of the parameters are shown. The manipulator is asked to follow the *Global-PTP_1* trajectory, and no filtering stage is applied. In this section, three aspects will be addressed:

- First, the influence of the initial values will be analyzed. This study is equivalent to the one done in Section 3.3.6.2, but now in an online way. We will use three types of initial values:
  - *Case A*: the set of base parameters obtained with the PC-IDIM-IV method from Table 2.6 and the covariance matrix filled with 0.01 on the diagonal;
  - *Case B*: the set of essential parameters from Table 2.1 obtained by applying the IDIM-LS method with the commanded signals to identify the model of the robot that is integrated in the controller, with 0.01 on the diagonal of $\boldsymbol{P}$ for those identified parameters and 1 for the others.
  - *Case C*: no a priori information (thus all the parameters initialized at 0) with $\boldsymbol{P}$ being a diagonal matrix filled with 1.

  The forgetting factor is set to 1, and the suppression of the iterations of the IDIM-CIV internal *for* loop of those joints whose velocity is lower than 0.1 rad/s is done. This is performed in order to avoid problems due to the non-continuous model of friction around 0 velocity of Equation (1.14).
- Second, we analyze the impact of deleting or not those iterations corresponding to velocities lower than 0.1 rad/s. We will carry out the same experiment as *Case B*, but without deleting points due to low speed, called *Case D*.

- Third, during the mentioned experiments, we also study the time the parameters take to settle and converge, as well as the algorithm's computation time.

Due to the facts that the applied algorithm is the so-called infinite memory algorithms (the forgetting factor is set to 1), and that the coupled methods have been proven to be identical to the recursive methods, the final numerical result should be the same as if the offline recursive method is applied.

Effect of initial values. Figure 4.2 and Table 4.1 show the curves and some features of the evolution of some parameters of joint 4 (the same as those of Figure 3.3, where we analyzed the effect of initial values on the IDIM-RLS method offline, to keep a coherence). In the mentioned table, we show the maximum (*Max.*) and minimum (*Min.*) values that the parameters take since the start of the experiment and the time (t_{set}) it takes for the parameter to stay between ±50% of the final value (calculated as the mean of the last 50 estimations) for the three cases. Even though this index seems to be too high for big parameters and very low for parameters near zero, it will allow the analysis of the convergence time.

Table 4.1: Features of the evolution of some parameters of joint 4 with different initial values.

Param.	*Case A*			*Case B*			*Case C*		
	Max.	Min.	t_{set} [s]	Max.	Min.	t_{set} [s]	Max.	Min.	t_{set} [s]
XXR_4	0.91	0.75	0	0.93	0.73	0	2.20	-0.22	37.40
ZZR_4	0.91	0.68	0	0.87	0.67	0	1.21	-1.00	18.37
MX_4	0.03	-0.03	68.28	0.04	-0.07	56.04	0.58	-0.61	67.44
MYR_4	2.43	2.34	0	2.46	2.33	0	2.79	-0.08	6.32
XY_4	0.03	-0.11	-	0.11	-0.07	-	1.40	-1.50	-
f_{c_4}	0.24	0.11	0	0.63	-0.16	-	0.63	-0.77	19.10

The analysis of Table 4.1 is straight forward. It is logical to think that the main inertial parameters (XXR_4, ZZR_4 and MYR_4) from *Case A* and *Case B* do not take time to settle, as they already start with a good estimation of the final value, result of other identification processes. Moreover, we know that the other inertial parameters may not affect the dynamic model significantly so the fact that they take more than one minute to settle, a priori, does not bring any problem. In fact, they may oscillate around 0, which leads to a large t_{set}. What it is important to highlight, although it is already a common knowledge, is that if no initial values are supplied to the algorithm, it will take some time to settle, which in online applications is not recommended. Usually, the estimation needs to be reliable from the first moment in order to use it and react against possible changes. Because of that, there are two possible solutions: either we have a good a priori knowledge of the system, or while applying the online identification, the model obtained in the first seconds of execution is discarded and not used for other purposes. *Case C* depicts this fact: it takes more than 35 seconds for all the three main inertial parameters to settle presenting a bigger variation between the maximum

and minimum values. Different results are obtained by incrementing the initial values of the covariance matrix, which may lead to a faster convergence but with higher variability.

Case C is not included in Figure 4.2 for clarity purposes. It can be observed that the evolution is quite similar in all of the cases shown, with less variation in general for *Case A*, due to the fact that the initial values are more precise.

Effect of speed deletion. Figure 4.2 also shows the difference between deleting low speeds and not. There is no much difference between *Case B* and *Case D* except for around the fifth second of execution, where *Case D* has a peak in all of its estimations due to the sudden change in friction parameters. This shows that it would be better to delete speeds around 0, or to consider a continuous friction model, because it makes the algorithm more stable in general.

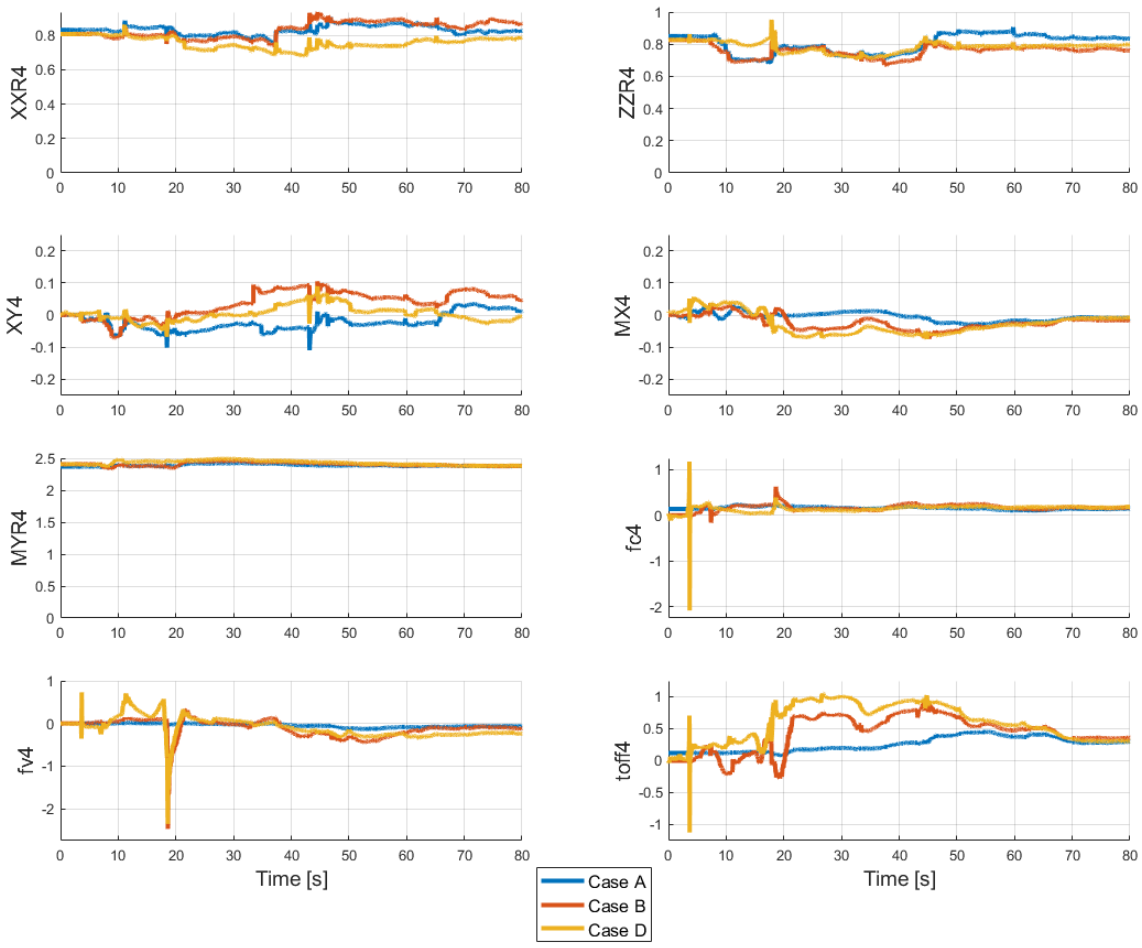


Figure 4.2: Evolution of some base parameters of joint 4 using the online IDIM-CIV without payload.

Computation time. The algorithm runs in the computer described in Appendix B.1.4. Regarding its computation time, *Case A* ran at 43.92 Hz as mean during the whole trajectory, *Case B* at 43.08 Hz, *Case C* at 43.18 Hz and *Case D* at 24.49 Hz. In all cases the algorithm runs fast enough for online identification. *Case D* is the worse case because it has to compute the estimates of all joints in every iteration, but still, updating the model around 25 times per second in a not powerful computer is an excellent result. Moreover, this information shows how the Coupled approach can help in computation time. The computation speed was almost doubled just by deleting the computation from the loop of those joints, which had low velocity at that moment. This shows the advantage of using a Coupled approach, instead of their non-coupled recursive counterpart. This way of selecting measurements that brings important information for the process is naive, and can be improved by selecting more sophisticated indexes.

4.7 Case study II: movement with constant payload

The second case study is when a constant payload is attached to the manipulator . This emulates a manipulator carrying a load or equipped with a tool that it is not known. Besides the computation time, two things will be analyzed:

- First, using the theory explained at the beginning of this chapter, the payload is going to be estimated (see Appendix B.1.2 for a description of the load used during this section). The accuracy of the method as well as the time it takes to determine its value will be analyzed.
- Second, we will compare different forgetting factors to study its effect on the estimates. It is important to know that, if the infinite memory algorithms are used, the online and offline applications should yield the same result if the same experimental features are used. However, if a forgetting factor is included, then the algorithm will account for the influence of a certain amount of measurements, yielding a different final result depending on the last part of the trajectory.

The experimental setup is the following: we use *Global-PTP_2* trajectory at 60% of its maximum speed, we delete all those values of speed lower than 0.05 rad/s, we apply the IDIM-CIV method, and we compare the following cases:

- *Case E1* and *Case E2*: forgetting factor equals to 1 (infinite memory algorithm) and matrix $\boldsymbol{P}$ full of 0.01 in the diagonal (difference between the two cases is explained afterwards).
- *Case F*: forgetting factor equals to 1 and matrix $\boldsymbol{P}$ full of 0.01 in the diagonal except for those parameters that are expected to change if a load is attached on the tip of the robot which will have 0.1.

- *Case G*: same than *Case F*, but with forgetting factor equals to 0.999. This value was selected by knowing that if it gets much smaller than that, then the algorithm may become unstable.

It is important to highlight that the trajectory used is not specially derived for load identification, so, the speed of convergence of the parameters regarding the load may be faster or slower depending on the trajectory. This choice is done to emulate a real task, where it is sometimes not possible to perform a trajectory specially designed for identification. The study done is then comparative.

Preliminaries. Before showing results, the effect of a payload in the model is analyzed by using the method in Equation (4.3). If the identified model is needed for control, then there is no special need to know which are the specific parameters of the load. Determining the base parameters of the model (manipulator and payload as one system) is enough. However, for validation purposes, we will isolate the effect of the load in the manipulator. From the model of the KUKA iiwa manipulator in Appendix B and Table B.2, we can derive the following relations:

$$\begin{aligned} MYR_{2_{wl}} &= MYR_2 + d_3 M_L \\ ZZR_{2_{wl}} &= ZZR_2 + d_3^2 M_L \\ XXR_{2_{wl}} &= XXR_2 + d_3^2 M_L \\ MYR_{4_{wl}} &= MYR_4 + d_5 M_L \\ ZZR_{4_{wl}} &= ZZR_4 + d_5^2 M_L \\ XXR_{4_{wl}} &= XXR_4 + d_5^2 M_L \end{aligned} \tag{4.13}$$

$$\begin{aligned} ZZR_{6_{wl}} &= ZZR_6 + YY_L \\ XXR_{6_{wl}} &= XXR_6 + YY_L \end{aligned} \tag{4.14}$$

$$XXR_{7_{wl}} = XXR_7 + XX_L - YY_L \tag{4.15}$$

$$\begin{aligned} MYR_{6_{wl}} &= MYR_6 - MZ_L \\ XY_{7_{wl}} &= XY_7 + XY_L \\ XZ_{7_{wl}} &= XZ_7 + XZ_L \\ YZ_{7_{wl}} &= YZ_7 + YZ_L \\ ZZ_{7_{wl}} &= ZZ_7 + ZZ_L \\ MX_{7_{wl}} &= MX_7 + MX_L \\ MY_{7_{wl}} &= MY_7 + MY_L \end{aligned} \tag{4.16}$$

where the sub-index wl refers to the parameters obtained in the trajectory where the manipulator is carrying a load; and sub-index L refers to the payload's parameters. These equations are applicable just to the model corresponding to the model of the KUKA iiwa obtained in Appendix B. They need to be redefined for each manipulator.

From these equations, and knowing the estimates of the robot when it is not carrying a load, the 10 inertial parameters of the payload can be estimated:

- From Equation (4.16), the values of MZ_L, XY_L, XZ_L, YZ_L, ZZ_L, MX_L and MY_L are

determined by simple replacement.

- From Equation (4.14), the estimate of YY_L is obtained by the LS solution of the system of equations.
- From the previous result and Equation (4.15), it is possible to determine the value of XX_L by simple replacement .
- Finally, from Equation (4.13), the value of M_L is determined by the LS solution of the system of equations. Here, we make the difference between *Case E1* and *Case E2*. *Case E2* will use the 6 equations to determine the mass, while *Case E1* will use the 3 equations corresponding to the parameters of joint 4. The subsequent cases (*Case F* and *Case G*) just take the three equations corresponding to joint 4.

The numerical values of the parameters of the load are given in Appendix B.1.2. However, there, the parameters are given with respect to a frame that is located on the base of the load. If the notation of Figure B.6 is used, then, the parameters need to be referenced to the frame of joint 7, which means a translation of 152 mm (due to link 7 and the flange used, see Appendix B). Using the theorem of Steiner for the inertias, the approximated parameters to be estimated are theoretically as follows:

- Mass: $M_L = 4.092\ kg$.
- First moments of inertia:
  - $MX_L = MY_L = 0\ kg.m$,
  - $MZ_L = 0.69\ kg.m$.
- Inertia tensor:
  - $XX_L = 0.10\ kg.m^2$,
  - $YY_L = 0.10\ kg.m^2$,
  - $ZZ_L = 0.01\ kg.m^2$,
  - $XY_L = XZ_L = YZ_L = 0\ kg.m^2$.

Results and discussion. The evolution of M_L and MZ_L for the 4 cases during the first 20 seconds of execution is shown in Figure 4.3. These two parameters are studied, because they are the two most significant values of the payload. Note that the 64 base parameters are being estimated on real-time. Several things can be analyzed from this figure:

- First, we see that the values approximately converge to the true parameter. After a first quick raise, then, they slowly approach or oscillate around the previously mentioned real values. This is an expected behavior, as we want the estimate to react quick to changes and, then, approach with enough time to the most precise model possible.

- The difference between *Case E1* and *Case E2* is clear when analyzing M_L. The parameters of joint 4 seem to stabilize quicker near the real value. It takes around 2 seconds for *Case E1* to yield a value bigger than 3 kg, while, for *Case E2* it takes more than 5 seconds. Moreover, 20 seconds are not enough for *Case E2* to arrive to a value near 4 kg. This happens because usually, in this kind of manipulators, joint 4 will move more than joint 2, and it is more adequate to rely on the changes of joint 4 than those of joint 2. Of course, this will depend on the trajectory, but it is usually the case.

- Comparing the rising times from *Case E1* and *Case F*, we can observe the effect of choosing different variances related to the base parameters that will change with the load. The estimates converge faster for *Case F* where the variances for those parameters are bigger. This technique is just applicable for cases where we know that the change in the robot will happen on its tip. The technique must be revisited if changes are expected on other links. Making the difference between those parameters that are known to variate with a change of payload, and those that do not, makes the rising time of the parameters much faster. This is a desired characteristic: we want the estimates to react as fast and good as possible.

- *Case G* shows how the parameters variate more when adding a forgetting factor different of 1. Notice the difference between *Case F* and *Case G* in the estimate of M_L after the 10^{th} second. Of course, we desire a behavior as stable as possible as in *Case F*, but this is the cost to pay if we want the algorithm to react against changes (see next section).

Finally, the frequency of computation of all these experiments range between 47 HZ and 54 Hz, which shows that adding a load has no effect on the computation time, and is a totally admissible rate for real-time interaction.

4.8 Case study III: movement with changing payload

For the third case study, we will consider a changing payload on the tip of the robot. This scenario represents a lot of possible tasks a manipulator can do, as the well-known pick and place task. Although just the payload's parameters are going to be analyzed, as well as in the previous section, all the 64 base parameters are identified on-line.

Figure 4.4 shows the payloads used in this scenario. A wooden box (see Figure 4.4.a) is rigidly attached to the robot at the end-effector level (same as the one used in [Mujica et al., 2023]). Its dimensions ($0.15m$ of height, $0.2m$ of width, $0.3m$ of length and width of wooden-walls $0.01m$) and its mass were measured and used to estimate the set of inertial parameters. They were computed with respect to a center of mass placed at the geometrical center, without considering the floor of the box (which is the thinnest and lightest wall). Moreover, given the shape of the object we approximated the non-diagonal inertia parameters to zero. The approximated parameters once referenced to the frame of joint 7 of Figure B.6 are (the sub-index L is still used to make reference to any payload on the end-effector):

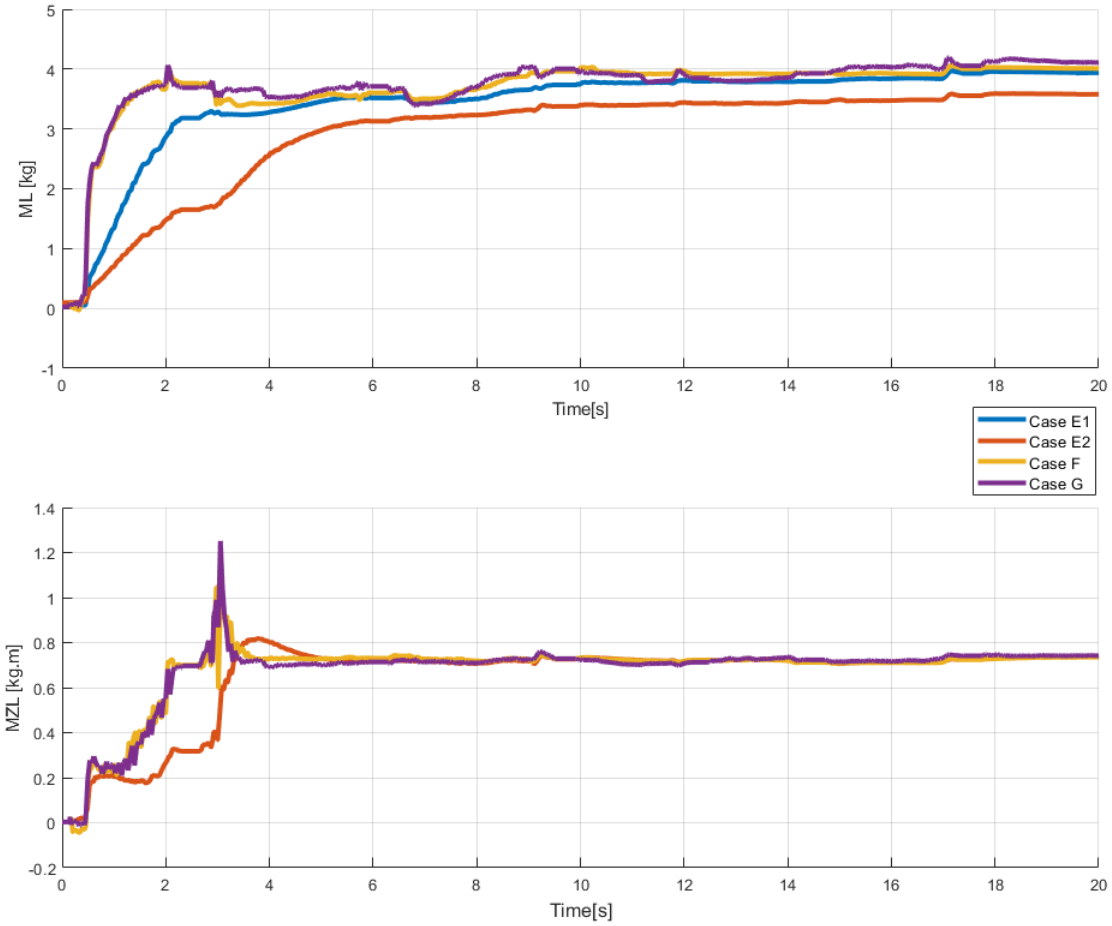


Figure 4.3: Evolution of M_L and MZ_L in online identification of a constant fixed payload.

- Box mass: $M_L = 0.941\ kg$.
- Box first moments of inertia:
  - $MX_L = -0.024\ kg.m$,
  - $MY_L = 0\ kg.m$,
  - $MZ_L = 0.425\ kg.m$.
- Box inertia tensor:
  - $XX_L = 0.104\ kg.m^2$,
  - $YY_L = 0.100\ kg.m^2$,
  - $ZZ_L = 0.009\ kg.m^2$,
  - $XY_L = XZ_L = YZ_L = 0\ kg.m^2$.

Figure 4.4.b shows the extra load that is added inside the wooden box while the manipulator is executing its task. They are weights of $0.5kg$ each (they will be used either alone or grouped in a pack of 4, resulting in a $2kg$ load). Due to the fact that, in this framework, it is not easy to calculate all the inertial parameters of the payload once these extra charges are included, we will focus our study for validation purposes on the mass. It is the only value that we have an accurate estimate, as the values of inertia and moments will depend on how the loads are placed in the box, which will be different in each of the executions.

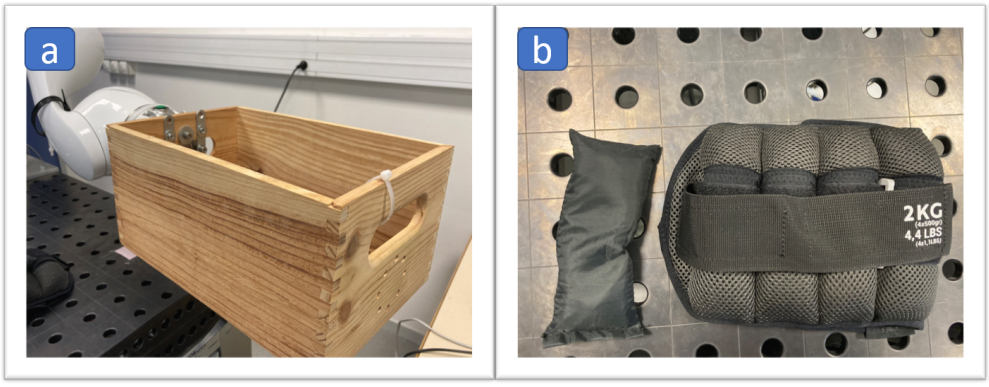


Figure 4.4: a) Wooden box and b) extra payloads.

Furthermore, the trajectories used in this section are not the same as the ones used all along the manuscript. Up to the moment, all the trajectories used for identification were not specially designed for it: they were a trajectory generated by interpolating different points on the workspace of the manipulator. Not many constraints besides joint limits to avoid collisions and velocity limits when using the payload in the previous section were used. In this section, we will add one more constraint to make the trajectory even more similar to a real task: the payload must not fall (or move abruptly) once inside the box. In order to make it a more real scenario, we will test a naive approach where we command three positions to the robot and allow the controller to interpolate between them (using the PTP motion,

see Appendix B). This scenario is inspired in the work done in [Mujica et al., 2023]. There, the manipulator works alongside with two persons carrying loads from one workstation to another. Figure 4.5 shows the three mentioned positions, and the transitions that are carried out on a loop. In this scenario, all the interaction (adding or taking out payloads) is done while the robot is standing still (it waits 3 seconds when arriving to each workstation, thus the first and last posture of Figure 4.5). The trajectory is an example of what could be a real task where no optimization regarding identification is done. The excitation of the parameters may be compromised.

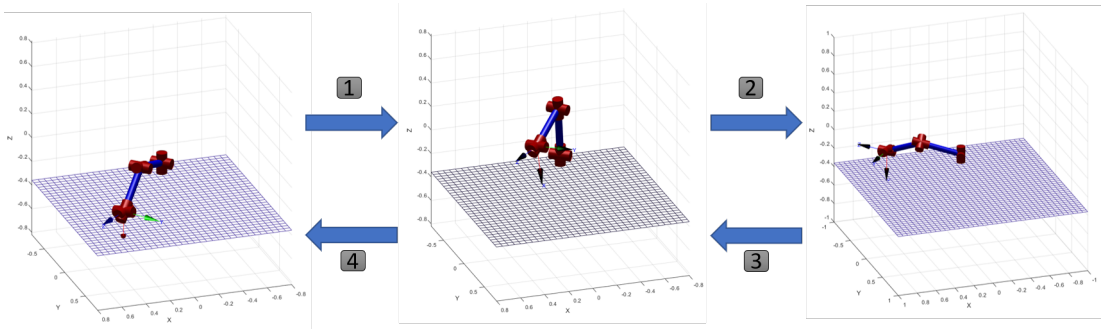


Figure 4.5: Steps of the trajectory for online identification of changing payload.

Several experiences were tested. The initial values of all of them are the same of *Case F* of the previous section. They differ from each other due to the forgetting factor choice:

- *Ex-1:* $\lambda = 0.999$,
- *Ex-2:* $\lambda = 0.9995$,
- *Ex-3:* $\lambda = 0.9999$,
- *Ex-4:* it corresponds to the technique in Equation (4.12) where different forgetting factors are set to each of the parameters. $\lambda = 0.999$ is chosen for parameters of joints 2,4,6 and 7, and $\lambda = 0.99901$ for the others,
- *Ex-5:* it corresponds to the technique in Equation (4.12) where different forgetting factors are set to each of the parameters. $\lambda = 0.9999$ is chosen for parameters of joints 2,4,6 and 7, and $\lambda = 0.999901$ for the others.

Results and discussion. Figure 4.6 shows the evolution of the estimation of M_L during the five mentioned situations, as well as the speed of joint 4, to show the reader in which part of the task the robot is.

The first thing to highlight is that the trajectory is clearly a task-like one. It consists of several repetitive movements with trapezoidal profiles of velocity. Due to this, the expected results are not to be as precise as in previous sections, as there may be many parameters that are not excited at all. Note also that, when the manipulator stands still, no update of parameters is done, as the speeds are lower than the set threshold.

From the three first experiences we can analyze the effect of different values of λ. The bigger it is, the more reluctant to changes are the estimates. Note that, for example, for *Ex-1*, it takes one round-trip between the two workstations to arrive to a precise estimate. Even after going from one work-station to the other, the estimates are fairly good, showing that in few seconds it can quickly predict the changes in the payload.

Furthermore, note how *Ex-4* differs from *Ex-1* and *Ex-5* from *Ex-3*. These experiences are comparable with each other as they use the same forgetting factor for the parameters of the joints that are related with the payload's parameters. Just by slightly indicating to the algorithm which are the parameters that need to change in a faster rate than others, the estimate changes drastically. *Ex-4* becomes too sensitive, indicating that the choice of λ is not good; while *Ex-5* seems to be a better choice over *Ex-3*.

Even if a task is being carried out, we could use the redundancy of the manipulator in order to excite as many parameters as possible in the best way. There are many ways to do this. For example, an optimization process, as the one mentioned in Chapter 1, can be carried out with an additional constraint being the final trajectory of the end-effector. Moreover, while the robot is standing still, we could make all the movements move or "dance" maintaining the end-effector in its position and orientation. This could help the optimization process as much more information would be added to the identification process.

Moreover, the choice of the forgetting factor is not straight forward. We have shown in this section how different forgetting factors affect one of the 64 estimated parameters. Studies regarding how is the optimal way to select these values are planned for the future.

In addition, the torque reconstruction of the whole trajectory could be analyzed (as it is going to be done in the next section), and the study of which estimates are being well-estimated during the movement can be done. This could lead to determine a reduced model of the manipulator corresponding to the specific task. This will reduce the computation time of the model, and make the process more robust against noise.

4.9 Case study IV: physical Human Robot Interaction

Real-time estimation of models has been done during several years for adaptive control. However, this kind of controllers are not recommended whenever unmodelled dynamic effects or external disturbances are present. They usually require that the dynamic model varies slowly in time, which cannot be ensured in many scenarios. As the pHRI field keeps growing, the need to identify the models of the manipulator while interacting with humans, gains importance. In this section, we will tackle this scenario. This work does not intend to give a final solution to the problem of estimating the interactions, but just open the perspectives of estimating the complete model of the manipulator while it is interacting with the environment.

The experimental setup consists of the robot equipped with the F/T sensor, described in Appendix B (to be able to show the ground-truth of forces applied by the human at the

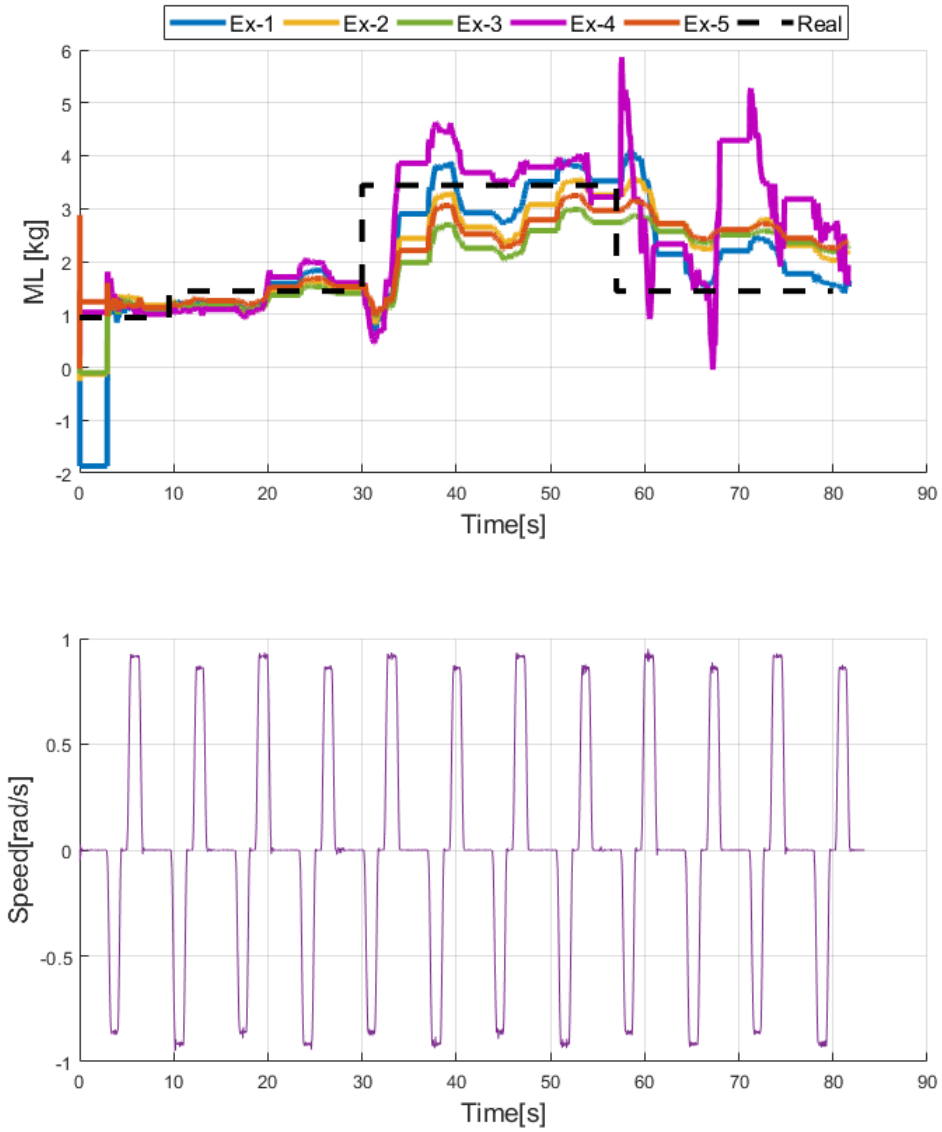


Figure 4.6: Evolution of varying payload mass during online identification with different forgetting factors strategies and speed of joint 4.

end-effector), and a trajectory in which the end-effector carries 6 times the same loop (done by indicating 4 points and interpolating via SPL trajectories). During the trajectory, the axis $\boldsymbol{z}_7$ remains parallel to the floor, and it moves on a perpendicular plane to the floor, simulating what could be a manufacturing task. The human interaction forces are sensed and shown in Figure 4.7. The forces are obtained with respect to an axis placed on the F/T sensor with the same orientation than the frame of joint 7 in Figure B.6. During the first and third trajectory, the human makes a force in the direction of gravity, and during the fifth cycle, he/she makes a force in the direction of axis $\boldsymbol{z}_7$ pushing the sensor. This can be seen in the F_x and F_z signals from the figure. There are also measurements on the other components because the force applied is not precise due to the fact that the robot is moving at the same time.

The IDIM-CIV method was used to estimate in real-time the 64 base parameters of the model. Knowing that all the interactions of the robot can be translated to moments on the joints and forces and moments in the base, we could imagine the method as identifying the parameters of a "virtual" manipulator which accounts for the real manipulator and the human. The torques produced in each joint can be considered as if they were produced by a manipulator with different links' inertial parameters. However, if the axis of the joint is parallel to that of the gravity, then, it may not be possible to find a combination of inertial parameters that explain the external force. This effect should be added to the model. Note also, that the physical meaning of the parameters may be lost, as know, we are identifying a system composed by the human and the manipulator.

Figure 4.8 shows the sensed torque, the reconstructed torque with the model obtained during Chapter 2 and the reconstructed torque with the model that is being estimated online (at each instant t, the estimated model at that iteration yields the torque shown in the figure). Even-though the results are not exact, they show promising results. By identifying the whole set of base parameters of the robot, we are able to predict the interaction of the human. Note that at the beginning, during the first seconds of the trajectory the reconstruction is almost perfect, showing the potential of the method. The results get worse afterwards because the tuning of the algorithm is not optimal. Better results are expected with an optimal choice of the hyper-parameters of the algorithm.

4.10 Final conclusion

In this chapter, the topic of online identification was treated. We introduced the new IDIM-CIV method, and several tools needed for time-varying models. We validated the methods in 4 different scenarios. First, when the robot is moving without any payload, which is the case that has been treated all along the manuscript. Second, we added a payload, and studied the scenarios where the payload is constant and when it is varying. Given a good tuning of the algorithm, the results are promising, being able to track the changing manipulator's model through time. Although the methods were tested adding a payload on the end-effector, the same results can be obtained if the payload is added somewhere else on the manipulator. The inertial parameters of the redefined manipulator's structure are obtained. Finally, we

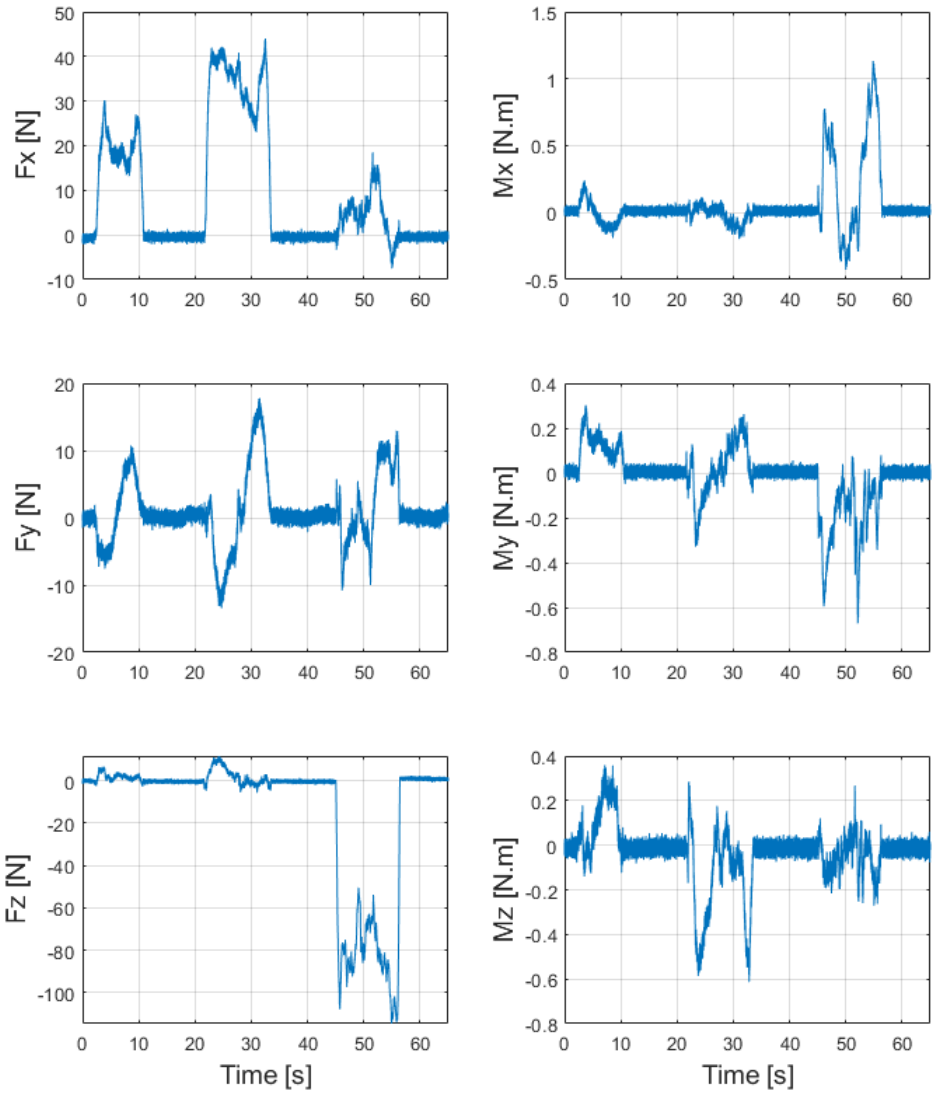


Figure 4.7: Forces applied by the human on the end effector.

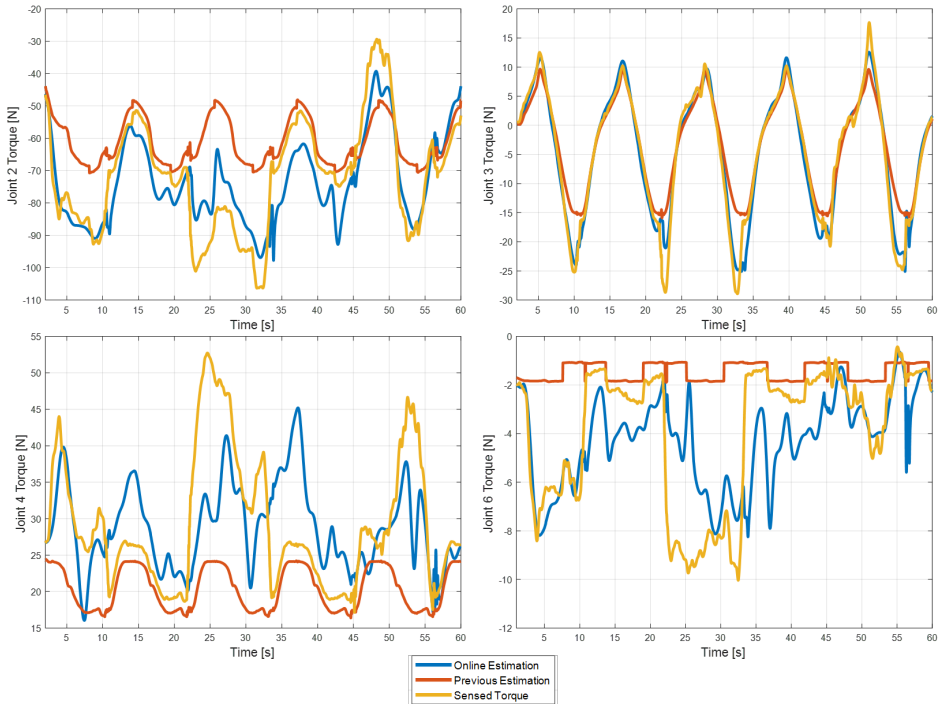


Figure 4.8: Sensed and reconstructed torques during human interaction.

also show the case where there is human interaction. Although, the results are not highly accurate, it is shown that, given a good tuning, it is possible to predict the joint torques while the manipulator is interacting with the human.

This chapter has opened several important questions and perspectives. It has been proven that the real-time identification of the whole model of the manipulator is feasible, and can be used and combined with other well-known techniques (observers, controllers, etc.) to improve the overall performance of the system. One of the main challenges is how to tune in the best way the hyper-parameters in order to track all the changes as good and as fast as possible. Besides this, the issue of how to excite the parameters during a task (either alone or including human interaction) is opened. New constraints regarding the task could be added to the generation of trajectories for identification to obtain the best possible condition of the regressor. Even if this is done, all the base parameters may not be enough excited, and a reduced manipulator's model could be searched. It would be enough to predict the behavior of the manipulator during a specific task. Finally, the human side could also be modeled and added to the manipulator to obtain a more complete model.

General conclusion & perspectives

Conclusion

This thesis manuscript presents the development and contributions done in the area of robotics identification. In a context where robots are being applied in more and more dynamically changing scenarios and environments, with more requirements in terms of precision, accuracy, performance and safety, part of the spotlight is back to identification techniques. Therefore, this thesis contributes to widen the knowledge of parameter identification of dynamic models in robotics.

The layout of this thesis is a logical pyramidal structure, where each of the chapters is supported by the previous ones, arriving to the final scenario where real-time identification of the model of the robot is done while a human is interacting with it. **Chapter 1** is an introduction to modeling and to the identification process in collaborative robotics. The terms, classification and state-of-the-art methods therein described, lay the theoretical groundwork for the following chapters.

Chapter 2 deals with *en-bloc* identification techniques. Many state-of-the-art techniques are based on the well-known Least-Squares solution. It has been proven that it yields the best estimates if several assumptions made on the system and on the noise signals are valid. However, many of them are not possible to be ensured in robotics, mainly because they are systems working in closed loop, thus there is an inherent correlation between the output and the inputs of the system. Moreover, due to unmodelled effects, uncertainties, external perturbations, noise and not enough exciting trajectories, between other phenomena, the estimates of the methods may not be physically feasible. The usually used methods yield solutions that minimize a specific cost-function without considering the signification of the parameters being estimated, which leads to parameters that are not likely to be the real values. To respond to these mentioned issues, in the first part of the chapter, and one of the main contributions of the thesis, the new PC-IDIM-IV (Physically Consistent Inverse Dynamic Identification Model Instrumental Variables) method is introduced and mathematically formulated. The method is based on the IV technique, which has been proven to be more robust against noise than the LS-based techniques, and it includes constraints regarding the physically consistency of the parameters. The results therein depicted, and supported by those in Appendix D, show the validity of the formulation and the superiority of the method against the state-of-the-art ones. The second part of the chapter, deals with another controversial aspect of robot modelling which is still source of attention: the friction. It is a complex phenomenon which depends on a lot of aspects. It ranges from a simple two terms linear equation to complex non-linear relations with time-varying coefficients. In this chapter, a methodology to identify a non-linear model of friction is proposed based on the Separable Least Squares. After determining the inertial parameters of the manipulator, e.g. by a reverse-engineering process to identify the model that is inside the model-based controller of industrial manipulators, the friction is

identified using sequential trajectories. The method is composed of two stages: one in which a non-linear optimization is done to obtain an optimal value of those parameters that make the model of friction to be non-linear, and another step, where considering the already identified parameters as known, the IDIM-LS is applied.

For real-time identification of models, *en-bloc* methods are usually not desired. Recursive methods are suitable for this kind of applications as they treat one measurement at a time. **Chapter 3** focuses on this, although testing the methods in an offline way. First, the IDIM-RIV (Inverse Dynamic Identification Model Recursive Instrumental Variables) method is presented, derived and enhanced. We show that by making a black-box identification of the closed-loop joint position, it is possible to use the filtered reference as the instrument of the method, which avoids the simulation of the dynamic model to fulfill this purpose. We also compare this method to state-of-the-art ones to show its performance. Moreover, we develop another two methods called IDIM-CLS (Inverse Dynamic Identification Model Coupled Least Squares) and S-IDIM-RLS (Sequential Inverse Dynamic Identification Model Recursive Least Squares). These methods divide the system in different subsystems, taking advantage of the triangular feature of the regressor matrix, becoming a computationally more stable algorithm. They yield the same result than the IDIM-RLS but without the need of any matrix inversion.

Finally, **Chapter 4** focuses on real-time identification of the whole manipulator model. A new method is introduced called the IDIM-CIV (Inverse Dynamic Identification Model Coupled Instrumental Variables) which has the good properties of noise rejection of IV-based methods and the algorithm stability of coupled methods. Four different scenarios are tested: a simple scenario where the manipulator moves freely; a scenario where it moves in the same way but with a constant load fixed on the end-effector; a third scenario where the manipulator performs a task-like trajectory with varying load; and a last case where the manipulator moves in a task-like trajectory while a human is interacting with it. Results shown in this chapter, show the potential use of the algorithm in real-time identification. With a proper tuning of the algorithm, the changes in payload can be accurately estimated and the model can follow the changes that are produced due the unexpected interaction of a human.

Last but not least, the work is supported by several complementary appendixes which are based on original work. **Appendix A** is a theoretical support and **Appendix B** describes the experimental framework used during most of this manuscript. Moreover, supporting the open-science movement, **Appendix C** describes a dataset where most of the trajectories used during the manuscript are available for users. **Appendix D** validates the PC-IDIM-IV method on a 6-*dof* manipulator and shows its advantage against state-of-the-art methods. In addition **Appendix E**, shows the possible advantages of using IV-based methods on mobile robotics, while also deriving a new model for unbalanced two-wheels differential drive mobile robots. Finally, **Appendix F** describes the results obtained by identifying three identical manipulators, and to show that the torque reconstruction obtained with one of them may differ significantly with respect to the others.

This thesis had for goal to study the identification methods of collaborative manipulators. Even though it is a field that has been studied for several decades, based on the developments carried out and results obtained, we showed that there are still many open challenges

in robotics identification. Particularly, it has been shown that real-time identification of the whole manipulator, provided a good algorithm tuning, can be a very powerful tool. Nevertheless, during this work, few scenarios were tested and further studies should be carried out in order to generalize the proposed solutions. For those reasons, the limitations are acknowledged and the possible improvements and perspectives are presented in the following section.

Perspectives

The results obtained in this thesis showed the potential application of the proposed contributions. However, several perspectives can be conceived, in order to enhance the methods proposed. In this section, some of them are going to be mentioned.

First, during this work, we proposed a new method called the PC-IDIM-IV, which is based on the instrumental variables and yields physically consistent estimates. We derived the mathematical formulation and validated it offline using a model that is linear with respect to the parameters. Further mathematical developments could be done to generalize this method to identify models that are non-linear with respect to some or all parameters. This would, eventually, allow to use non-linear models of friction which may also depend on factors as the load and temperature. In addition, although the optimization regarding the physically feasibility of the parameters is time-consuming, the way how to add this information to real-time identification could be explored.

Second, we showed that for many industrial manipulators, much information related to it, is hidden from users by the manufacturers for copyright and safety issues. Due to this, the structure and parameters of the industrial controller is usually unknown. In order to obtain a model of the whole system (controller and manipulator), a black-box model can be derived for the controller, and a gray-box model for the manipulator, if the necessary measurements are available. However, in the case no signal *in between* these systems is available, then it is not possible to identify them individually, and the whole system becomes as a black-box because we have no knowledge of the controller. However, several techniques could be explored in order to still be able to include the physical knowledge of the manipulator in the model and retrieve the real value of the parameters, while also identifying the controller. One possible solution is what was named as *dark-grey box modeling* in [Ljung et al., 2004] by means of defining custom regressors for the black-box identification problem. Another method could be to search for support in a relatively new type of neural networks called PINNs (Physically Informed Neural Networks) [Raissi et al., 2019, Karniadakis et al., 2021, Nicodemus et al., 2022]. These networks allow to include the physical knowledge of the system in the neurons, and in this way retrieve the physical meaning of the parameters. Together with colleagues, we are currently studying this topic.

Third, the real-time application of the methods open a lot of questions. One of the main issues is the possible sub-excitation of the parameters when carrying out a task, which will lead to bad estimates. Usually manipulators will be asked to perform a trajectory which is

related to a specific task and not specially designed for identification. Taking advantage of the possible kinematic redundancy of the manipulators, there are many joint trajectories that yield the same end-effector path. One possible solution to the mentioned issue could be carry out an optimization process to select between them, that which yields a better condition number of the regressor matrix. Another possible solution is to obtain a reduced model of the manipulator corresponding to a specific task, and identify it instead of considering the whole manipulator. We know that when carrying a trajectory there will be some parameters that are not sufficiently excited, meaning that, given the treated measurements, they do not have a big influence on the model. If the manipulator is intended to carry out a repetitive task, we could imagine a method to find this reduced model that will account just for the parameters that are important to it. This could be of big utility in real time implementation: it would improve the computation speed, the algorithm stability and the immunity to noise.

Furthermore, in the context of a continuously growing field of pHRI, several perspectives considering the human interaction can be conceived for identification. For instance, in a cooperation task, where the manipulator carries a load together with an operator, one issue is how to differentiate between the forces applied by the human and those due to the load. Identifying the correct model of the payload while there is interaction, could be of much help for the controller. In addition to this, as previously said, an even more challenging problem, is how to generate trajectories that are suitable for parameter identification, while the cooperative activity is going on. This brings up requirements related to human's safety and of its experience satisfaction. Moreover, a current challenge being investigated in the community is how to model the human to be able to predict its behavior. For instance, its effect could be included in the model of the manipulator, and in that way identify the behavior of the coupled system instant by instant. To do that, we could consider as the existence of a "virtual manipulator", in which the links' dynamic parameters would account for the real manipulator and the human effect, together.

Finally, from a general point of view, all the methods presented in this manuscript could be compared between each other and with state-of-the-art methods and generalised to different real-time scenarios and systems. BFor instance, they could be extended to: mobile manipulators [Kim et al., 2020], manipulators working cooperatively [Gan et al., 2014], flexible manipulators [Yoshikawa et al., 2001], or modular robots [Deremetz et al., 2021]. All of them present their own challenges, but being able to identify their model with physically feasible estimates in real time, may lead to a better performance. Moreover, we could consider different kind of payloads: non-rigid objects, liquid containers and wires (as in the scenario showed in the last chapter).

List of publications

In the context of this thesis, several publications were carried out and some more are planned for the near future. The accepted and published works, as well as those that are currently under review, are listed below.

Published papers

- [Ardiani et al., 2021]: Ardiani, F.; Benoussaad, M.; and Janot, A. (2021). **Comparison of least-squares and instrumental variables for parameters estimation on differential drive mobile robots.** *IFAC-PapersOnLine, 54(7):310–315.* Presented at SYSID 2021.
- [Ardiani et al., 2022]: Ardiani, F.; Benoussaad, M.; and Janot, A. (2022). **Improving Recursive Dynamic Parameter Estimation of Manipulators by knowing Robot's Model integrated in the Controller**. *IFAC-PapersOnLine, 55(20), 223-228.* Presented at MATHMOD 2022.

Accepted papers

- Ardiani, F.; Janot, A.; and Benoussaad, M. (2022). **Industrial Robot Parameter Identification using a Constrained Instrumental Variable Method.** Presented at IROS 2022. To appear.
- Ardiani, F.; Benoussaad, M.; and Janot, A. (2022). **On the Dynamic Parameter Identification of Collaborative Manipulators: Application to a KUKA iiwa.** To be presented at ICARCV 2022. To appear.

Submitted papers

- Ardiani, F.; Mujica, M.; Benoussaad, M.; Janot, A.; and Fourquet, J-Y. (2022). **MESSII: a parameter identification dataset for KUKA IIWA collaborative manipulator.** *The International Journal of Robotics Research.* Under review.

Appendix A

Least-squares estimator

This short theoretical background is inspired on the work of [Lambert, 2018] and related econometrics courses by the author. The notation of this appendix is independent of the rest of the manuscript.

A.1 Derivation

The Least-Squares (LS) estimator models a process by fitting the parameters of a linear equation according to the minimization between the squared error sum of the actual values of the dependant variable minus the fitted values. The real model is:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{A.1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the vector of n dependant variables; $\boldsymbol{\beta} \in \mathbb{R}^m$ is the vector of m parameters to be estimated; $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ the matrix regrouping the effect of the independent variables; and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is the vector of residuals.

The idea of the LS is to find the values of $\boldsymbol{\beta}$ which reduce the sum of the squared errors $\boldsymbol{\epsilon}$. Then, the model becomes:

$$\boldsymbol{y} = \boldsymbol{W}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\epsilon}}, \tag{A.2}$$

where the $\hat{}$ symbol refers to an estimated value. The cost function to be optimized is:

$$S = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \hat{\epsilon}_1^2 + \hat{\epsilon}_2^2 + ... + \hat{\epsilon}_n^2 = \hat{\boldsymbol{\epsilon}}^T \hat{\boldsymbol{\epsilon}}. \tag{A.3}$$

After that, by simple replacement, it follows:

$$S = (\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}})^T (\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}). \tag{A.4}$$

Moreover, by knowing that the transpose is a linear operation:

$$S = (\boldsymbol{y}^T - \hat{\boldsymbol{\beta}}^T \boldsymbol{W}^T)(\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}), \tag{A.5}$$

and, after applying the distributive property, we arrive to the final expression of the cost function S:

$$S = \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{W}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^T\boldsymbol{W}^T\boldsymbol{y} + \hat{\boldsymbol{\beta}}^T\boldsymbol{W}^T\boldsymbol{W}\hat{\boldsymbol{\beta}}. \tag{A.6}$$

The final objective is to find $\hat{\boldsymbol{\beta}}$ that reduces S, thus finding the value of $\hat{\boldsymbol{\beta}}$ that makes the derivative of S with respect to $\hat{\boldsymbol{\beta}}$ equals to zero. Taking into account that: all the four terms of the previous equation are scalar quantities; the first term does not depend of $\hat{\boldsymbol{\beta}}$; the second term and the third term are the same but transposed; the derivative of the generic vectors $\boldsymbol{a}^T\boldsymbol{b}$ with respect to $\boldsymbol{a}$ is equals to $\boldsymbol{b}$; and that the derivative of the generic $\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{a}$ with respect to $\boldsymbol{a}$ is equals to $2\boldsymbol{B}\boldsymbol{a}$; then, the derivative of S with respect to $\hat{\boldsymbol{\beta}}$ is:

$$\frac{\delta S}{\delta \hat{\boldsymbol{\beta}}} = -2\boldsymbol{W}^T\boldsymbol{y} + 2\boldsymbol{W}^T\boldsymbol{W}\hat{\boldsymbol{\beta}} = \boldsymbol{0}. \tag{A.7}$$

Finally, rearranging, we get the final solution of the LS estimate:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{y}. \tag{A.8}$$

This equation shows that the LS estimate cannot be computed if there exists a perfect co-linearity between rows or columns of $\boldsymbol{W}$. This would make $\boldsymbol{W}^T\boldsymbol{W}$ singular, thus, non-invertible.

A.2 BLUE estimator

The LS estimate is considered to be the BLUE (Best Linear Unbiased Estimate) if the error terms have a zero conditional mean, are homoscedastic, present no correlation, and the independent features are not correlated. The proofs behind these assumptions are shown in this section.

Unbiased. By replacing $\boldsymbol{y}$ of Equation (A.8) with its definition of Equation (A.1), and rearranging, we obtain:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta} + (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{\epsilon}. \tag{A.9}$$

Applying the expectation operator $\mathbb{E}$ at both sides of this equation, and making the zero conditional mean of error assumption (meaning that the expectation of the residual given any set of independent variables is zero thus $\mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$), we get:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}) = \mathbb{E}(\boldsymbol{\beta}). \tag{A.10}$$

This shows that given the mentioned noise assumption, the expectation of the estimated values equals the real values, thus the estimates are unbiased.

Variance. Using the property that given a non-stochastic matrix $\boldsymbol{A}$ and a stochastic matrix $\boldsymbol{B}$, it follows that $\mathrm{Var}(\boldsymbol{A}\boldsymbol{B}) = \boldsymbol{A}\mathrm{Var}(\boldsymbol{B})\boldsymbol{A}^T$, being $Var(.)$ the variance operator, then, the variance of the LS estimator is:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T Var(\boldsymbol{y})\boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{W})^{-1}. \tag{A.11}$$

We are now going to make the assumption that the error is homoscedastic (meaning that all dependant variables have the same variance σ^2) and uncorrelated. It can be formulated mathematically like:

$$\text{Var}(\boldsymbol{y}) = \sigma^2 \mathbb{I}. \tag{A.12}$$

By substitution on Equation (A.11) and rearrangement, we get:

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2 (\boldsymbol{W}^T \boldsymbol{W})^{-1}, \tag{A.13}$$

which is the expression of the variance of the estimates.

Best. By applying the Gauss-Markov theorem, it will be proven that the LS estimator is indeed the BLUE. Let's start by supposing there is another estimator $\bar{\boldsymbol{\beta}}$ defined by:

$$\bar{\boldsymbol{\beta}} = (\boldsymbol{W}^T \boldsymbol{W})^{-1} \boldsymbol{W}^T \boldsymbol{y} + \boldsymbol{D}\boldsymbol{y}, \tag{A.14}$$

where $\boldsymbol{D}$ is a non-zero matrix. Substituting with Equation (A.1) and taking expectations under the assumption that there is zero conditional mean error, we get that:

$$\mathbb{E}(\bar{\boldsymbol{\beta}}) = \boldsymbol{\beta} + \boldsymbol{D}\boldsymbol{W}\boldsymbol{\beta}. \tag{A.15}$$

This means that any another estimator different from the LS estimator will always be biased under the mentioned assumptions unless $\boldsymbol{DW}$ equals zero.

Taking the variance in the same way than Equation (A.11) and assuming that the error are homoscedastic and uncorrelated, we have:

$$Var(\bar{\boldsymbol{\beta}}) = \sigma^2 [(\boldsymbol{W}^T \boldsymbol{W})^{-1} + (\boldsymbol{W}^T \boldsymbol{W})^{-1} (\boldsymbol{D}\boldsymbol{W})^T + \boldsymbol{D}\boldsymbol{W}(\boldsymbol{W}^T \boldsymbol{W})^{-1} + \boldsymbol{D}\boldsymbol{D}^T]. \tag{A.16}$$

This equation shows that, even if the new estimator is unbiased, thus $\boldsymbol{DW} = \boldsymbol{0}$, its variance will be greater than that of the LS estimator as the product $\boldsymbol{D}\boldsymbol{D}^T$ returns a positive value greater than zero (note that if $\boldsymbol{D} = \boldsymbol{0}$, then the new estimator is, in fact, the LS estimator).

These proofs show that the LS estimator, besides being consistent (if n increases, then the variance of the estimates gets smaller), it is the most efficient linear estimator, meaning that no other method will yield unbiased linear estimates with lower variance than the LS.

Appendix B

Collaborative robot platform

Without loosing the generality of the methods proposed during this work, which can be applied to other type of robots and platforms, this section will describe the KUKA iiwa manipulator's structure and signals, the load used on some of the experiments, the Force-Torque sensor and the computer used to gather the measurements. This is the experimental setup used all along the work in order to validate the methods in a real scenario (a small modification is done for the experiments carried out in Chapter 4, which is explained in Section 4.5). In this Appendix, we also show the connectivity framework between all the mentioned parts and the way the applications were coded. Further explanation, and information about almost all the trajectories used in this work can be seen in Appendix C and its related website of the *MESSII dataset* [1].

B.1 The components

B.1.1 The robot

In this work, the industrial manipulator KUKA LBR iiwa 14 R820 (lightweight robot intelligent industrial work assistant) is considered [kuka, 2019]. Shown in Figure B.1, it is a 7-*dof* lightweight robotic arm intended for collaborative applications. Similarly to other collaborative robots, it is equipped with both, encoders to sense the joint/link positions and torque sensors to measure the link-side torque. The torque sensors are placed in order to guarantee the safety of humans interacting with the entire robot structure, and to allow gravity and stiffness compensation as well as impedance control [Albu-Schäffer et al., 2007a].

The manipulator is rigidly mounted over a base, thus no movement is considered on link 0. The robot is also equipped in its tip with the interface *Media Flange Touch Electrical* also from KUKA [kuka, 2015a]. It is a universal interface that allows the user to connect, configure and command different components placed on the robot's tip.

The collaborative manipulator's structure is depicted in Figure B.2. There are five signals that the KUKA iiwa controller makes available for users [kuka, 2015b, kuka, 2017], which are also available or can be calculated on almost all collaborative robots:

[1] https://messii-dataset.enit.fr

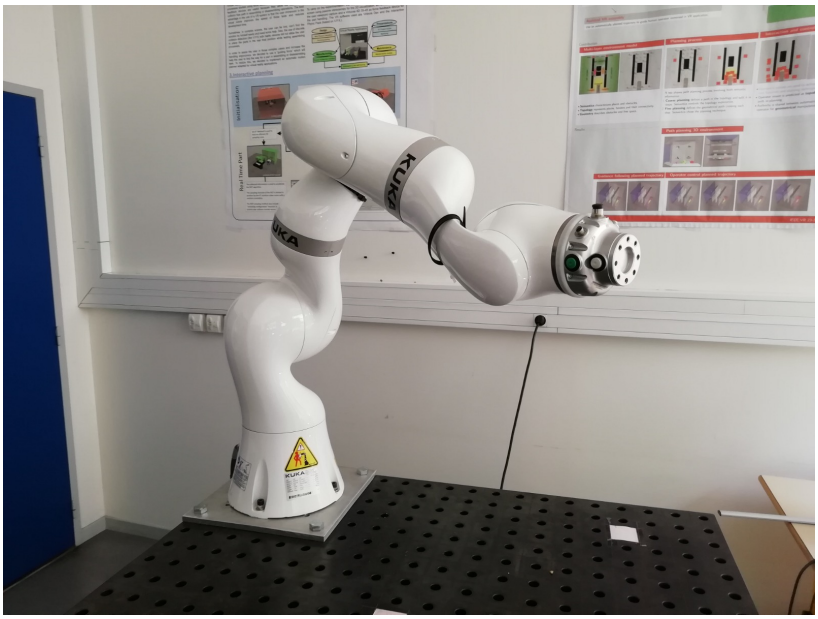


Figure B.1: KUKA LBR iiwa 14 R820.

- *Commanded position*: after processing the desired points/trajectory that the user wants the robot to follow, the controller calculates this signal. It is the position of the successive points that the robot is asked to follow, which means the *reference position*.
- *Commanded torque*: it is the torque related to the *commanded position* by the internal dynamic model that is used by the controller, thus the *reference torque*.
- *Sensed position*: it is the measurement of the link-side position. It is either a real measurement or an estimation made by the manufacturer from sensing the motor position and knowing the gearbox parameters. It is also called the *measured position*.
- *Sensed torque*: it is the measurement of the torque sensor placed after the gearbox typical of these collaborative manipulators, also called the *measured torque*.
- *External torque*: it is a filtered signal product of the difference between the *commanded torque* and the *sensed torque*. It accounts for all phenomena that the model of the controller does not take into consideration: unmodeled effects, friction, uncertainties in the model, external forces, unaccounted loads, noise, etc.. This signal can be monitored for safety, collision and fault detection purposes. It is also called the *residual torque*.

Usually, this kind of controllers are of the model-based type. The controller has already the information about the model of the robot and that is how it gets the relation between the *commanded position* and *commanded torque* (proofs are shown in Section 2.3.6.1 and in Section B.3.4).

Moreover, the load attached to the end-effector can be configured by the user in order to withdraw its effect from the *external torque* and let the controller adapt its behavior to

it. This can be done manually or by a built-in identification procedure that determines the payload by moving links 6 and 7.

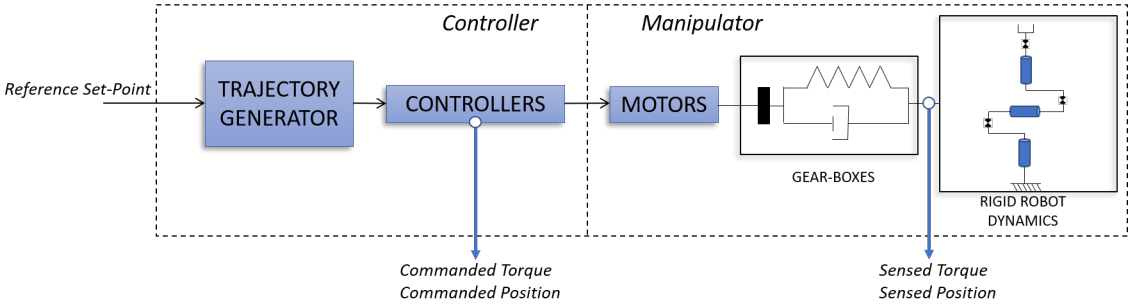


Figure B.2: Industrial collaborative manipulator's structure.

B.1.2 The load

Same as for the manipulator, the methods developed are not limited to one specific load, but they can be used for any load. A rigid object can be modeled as a rigid body with mass M_L, first moment of inertia $\mathbf{MS_L}$ and inertia tensor $\mathbf{I_L}$ defined as follows:

$$\mathbf{MS_L} = [\mathrm{MX}_L \ \mathrm{MY}_L \ \mathrm{MZ}_L]^T, \tag{B.1}$$

$$\mathbf{I_L} = \begin{bmatrix} XX_L & XY_L & XZ_L \\ XY_L & YY_L & YZ_L \\ XZ_L & YZ_L & ZZ_L \end{bmatrix}, \tag{B.2}$$

where the parameters are defined in the same way than for the manipulator's links in Section 1.2, but with the sub-index L referring to the payload.

One of the payloads used in this work is the steel cylinder attached to the tip of the robot shown in Figure B.3. It has 34 mm of height and 140 mm of diameter, and it weights 4.012 kg (4.092 kg when adding screws). Considering its regular shape, we suppose its center of mass is in the geometrical middle point. Thus, from a set of axis in the middle bottom point of the cylinder, we approximately have:

$$\mathbf{MS_L} = [0 \ 0 \ 0.0682]^T kg.m, \tag{B.3}$$

and the inertia tensor can be approximated with a diagonal matrix:

$$\boldsymbol{I_L} = \begin{bmatrix} 0.0054 & 0 & 0 \\ 0 & 0.0054 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} kg.m^2. \tag{B.4}$$

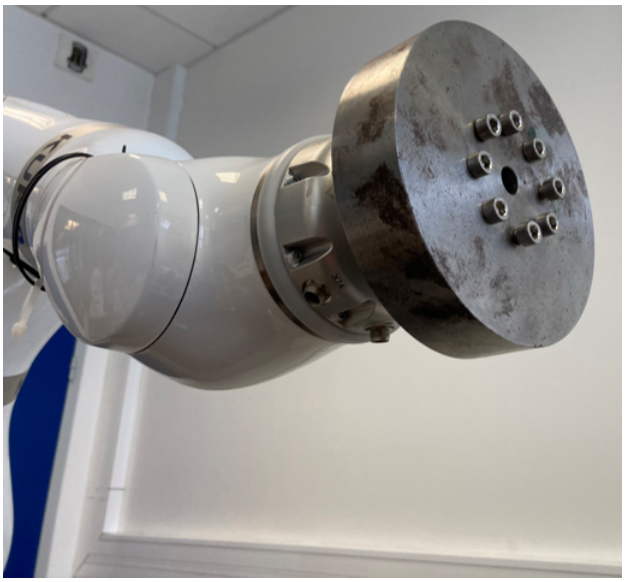


Figure B.3: Steel cylinder fixed to the tip of the robot used as payload.

B.1.3 The Force-Torque sensor

In order to enhance the platform for some set of experiments, a Force-Torque (F/T) sensor was equipped on the tip of the robot. The capacitive 6-axis force torque sensor shown in Figure B.4, model RFT76-HA01 from ROBOTOUS[2], was used. It can provide the measurements of forces in the three directions (with a limit of 300 N and a resolution of 200 mN for each axis) and the three components of torque related to each axis (with a limit of 10 Nm and a resolution of 8 mNm for each axis) at a rate of up to 1000 Hz. Weighting 200 g, it is powered by an etherCAT adapter that uses CAN interface to communicate with the sensor, which was also provided by ROBOTOUS. It was installed on the tip of the robot via a 3D plastic printed coupling.

B.1.4 The computer

The data of both, the robot and the sensor, were logged at a rate of 1000 Hz on an external computer. We used a HP ZBook 15 G2 Mobile Workstation [3] (Intel Core i7-4710MQ 2.5 GHz, 2.50 GHz, 8 Go DDR3L SDRAM) running Ubuntu 18.04.5 LTS.

[2]http://www.robotous.com/
[3]https://www.hp.com/

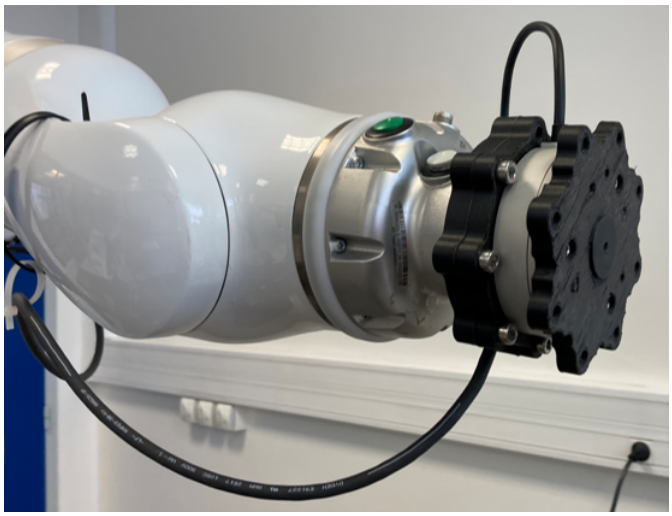


Figure B.4: Force-Torque sensor fixed to the tip of the robot.

B.2 The framework

Having introduced all the components, in this section, we present the way they communicate and interact between each other. The framework can be summarized in the diagram shown in Figure B.5.

As in any other application for this manipulator, the user must code a JAVA program on the KUKA controller, which operates on the SUNRISE operative system (also created by KUKA), with the desired behavior and trajectory to follow. This JAVA application communicates with a C/C++ application running on the computer via a channel performed using the *Fast Robot Interface* (FRI) (a version of the one presented in [Schreiber et al., 2010]), which is an open-source library for remote control of some KUKA robots. It allows deterministic access to information on joint position and axis torque at a rate of 1-100 ms, and also provides the tools to direct path manipulation at a rate of 1-4 ms. Moreover, this C/C++ application is in charge of writing the *log* file with all the obtained measurements coming from the robot and the F/T sensor. In order to do this, this application has reading access to a shared memory space, where another C/C++ application running on the same computer, in charge of the communication with the F/T sensor, writes the respective measurements.

There are several ways of commanding the desired trajectory of each joint of the manipulator, some of which will be detailed because of their utility for the work carried out during this manuscript:

- *Standard motion*: motion is specified by the user by desired points. Then, the controller interpolator generates the required position, velocity and acceleration profiles. Two ways of interpolating these points are:
  - *Point-to-point motion* (PTP): the joint achieves its desired end point along the fastest path. This means that the joints will always try to accelerate in the fastest

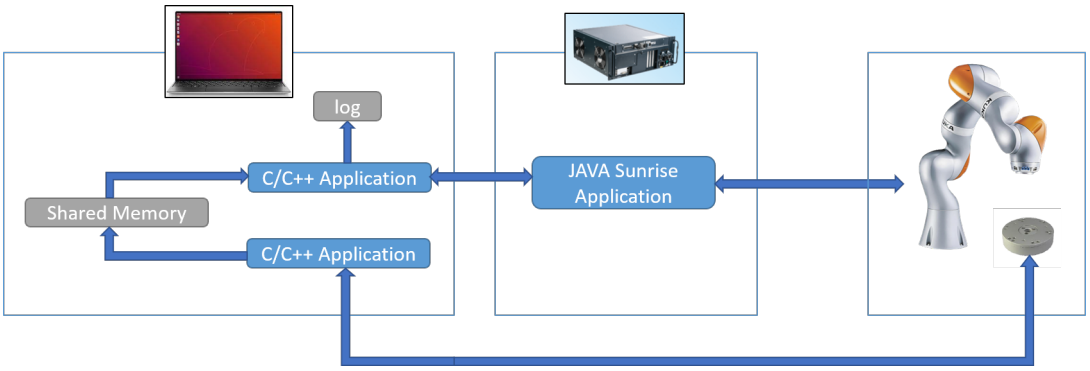


Figure B.5: Communication framework of the experimental setup of the KUKA iiwa manipulator.

possible way to achieve the maximum velocity in the quickest way. The velocity profiles are generally trapezoidal with maximum acceleration and deceleration, stopping each time a desired point is achieved.

– *Spline motion* (SPL): it enables the generation of curved paths. It results in paths that run smoothly through the desired points, without stopping once an intermediate point is achieved.

- *Smart Servo and Direct Servo* (DS): they are methods allowing a quick update of the end-point (>2ms). The end-point can be changed in each iteration. It indirectly allows to have a control on the velocity and acceleration profiles. The controller puts limitations on desired trajectories that are not feasible for the robot.

B.3 Experimental preliminaries

B.3.1 Model selection and reduction

To arrive to the IDIM expression of Equation (1.25), the model of the system has to be derived: model of the kinematic and dynamic phenomena, as well as the posterior model reduction. Figure B.6 shows the parameters describing the kinematics of the manipulator using the MDH (Modified Denavit-Hartenberg) convention, and depicts the robot manipulator with its seven joints A_i for $i = 1, ..., 7$ in zero position, the link frames and their associated parameters.

Using the closed-form relations introduced in Section 1.3.2.1, from the 70 inertial standard parameters (10 parameters for each link without considering friction), 43 base parameters are obtained, which are summarized in Table B.1, and described as follows:

- 17 base parameters due to regrouping are shown in Table B.2.
- 26 base parameters that need no regrouping: XY_i, XZ_i, YZ_i, MX_i for $i = 2, ..., 7$; and MY_7 and ZZ_7.

- All parameters except ZZ_1 from the first link, MZ_2 and M_2 do not have effect on the dynamic model.

Table B.1: Summary of transformation between standard and base parameters for the KUKA LBR iiwa manipulator following the MDH convention of Figure B.6 (R = regrouped, D = direct influence thus no regrouping, - = no influence).

Link \ Parameter	*XX*	*XY*	*XZ*	*YY*	*YZ*	*ZZ*	*MX*	*MY*	*MZ*	*M*
1	-	-	-	-	-	R	-	-	-	-
2	R	D	D	R	D	R	D	R	-	-
3	R	D	D	R	D	R	D	R	R	R
4	R	D	D	R	D	R	D	R	R	R
5	R	D	D	R	D	R	D	R	R	R
6	R	D	D	R	D	R	D	R	R	R
7	R	D	D	R	D	D	D	D	R	R

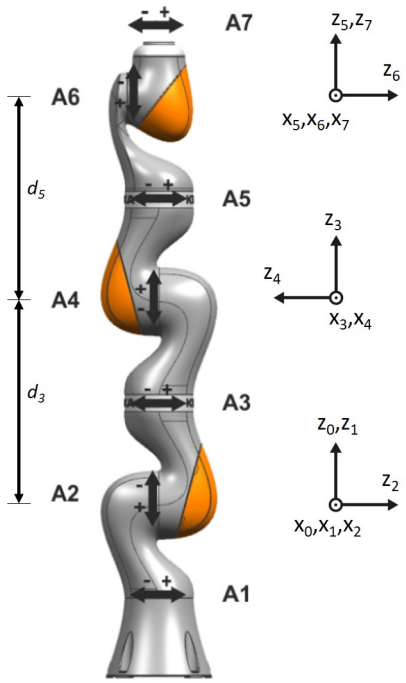
B.3.2 Exciting trajectories choice

As shown in Figure 1.6, the next step in the parameter identification process is to design a trajectory that excites the desired parameters. Some of the trajectories used along this manuscript are explained in the *MESSII data-set* generated by the authors and colleagues which is detailed in Appendix C.

In this section, we show one trajectory to let the reader grasp an idea of how the manipulator moves. A trajectory using PTP technique is used (*Global-PTP_1* from the *MESSII data-set*). To build this trajectory, after considering constraints regarding joint limits and the platform position to avoid collisions, several points are selected in the useful work-space. From these points, the KUKA Sunrise OS generates the trajectory (in this case by interpolating the points using PTP motion). This method of generating trajectories yields similar results than those that make use of a non-linear optimization with linear and non-linear constraints of the condition number of the regressor matrix (or indexes derived from it), which can be time and resource consuming [Jubien et al., 2014b]. It also randomize the process of generating a trajectory, having one that is not specially designed for identification, as it could happen in a real scenario. In Figure B.7, the movement of the end-effector during a portion of the whole trajectory of the robot is shown. This figure was plotted using the Robotics Toolbox of MATLAB [Corke and Khatib, 2011].

B.3.3 Data processing

The measured signals (*sensed position* and *sensed torque*, and consequently, the *external torque*) are noisy by nature. This is the reason why a 4-steps filtering procedure is designed,



Link i	θ_i	d_i	a_i	α_i
1	q_1	0	0	0
2	q_2	0	0	$-\pi/2$
3	q_3	$0.42m$	0	$\pi/2$
4	q_4	0	0	$\pi/2$
5	q_5	$0.40m$	0	$-\pi/2$
6	q_6	0	0	$-\pi/2$
7	q_7	0	0	$\pi/2$

Figure B.6: MDH parameters and frames of the KUKA LBR iiwa 14 R820.

Table B.2: Regrouped base parameters of the KUKA LBR iiwa following the MDH convention stated in Figure B.6.

$\mathrm{ZZR}_1 = \mathrm{ZZ}_1 + \mathrm{YY}_2$
$\mathrm{XXR}_2 = \mathrm{XX}_2 - \mathrm{YY}_2 + \mathrm{YY}_3 + 2d_3\mathrm{MZ}_3 + (M_3+M_4+M_5+M_6+M_7)d_3^2$
$\mathrm{ZZR}_2 = \mathrm{ZZ}_2 + \mathrm{YY}_3 + 2d_3\mathrm{MZ}_3 + (M_3+M_4+M_5+M_6+M_7)d_3^2$
$\mathrm{MYR}_2 = \mathrm{MY}_2 - \mathrm{MZ}_3 - (M_3+M_4+M_5+M_6+M_7)d_3$
$\mathrm{XXR}_3 = \mathrm{XX}_3 - \mathrm{YY}_3 + \mathrm{YY}_4$
$\mathrm{ZZR}_3 = \mathrm{ZZ}_3 + \mathrm{YY}_4$
$\mathrm{MYR}_3 = \mathrm{MY}_3 - \mathrm{MZ}_4$
$\mathrm{XXR}_4 = \mathrm{XX}_4 - \mathrm{YY}_4 + \mathrm{YY}_5 + 2d_5\mathrm{MZ}_5 + (M_5+M_6+M_7)d_5^2$
$\mathrm{ZZR}_4 = \mathrm{ZZ}_4 + \mathrm{YY}_5 + 2d_5\mathrm{MZ}_5 + (M_5+M_6+M_7)d_5^2$
$\mathrm{MYR}_4 = \mathrm{MY}_4 + \mathrm{MZ}_5 + (M_5+M_6+M_7)d_5$
$\mathrm{XXR}_5 = \mathrm{XX}_5 - \mathrm{YY}_5 + \mathrm{YY}_6$
$\mathrm{ZZR}_5 = \mathrm{ZZ}_5 + \mathrm{YY}_6$
$\mathrm{MYR}_5 = \mathrm{MY}_5 + \mathrm{MZ}_6$
$\mathrm{XXR}_6 = \mathrm{XX}_6 - \mathrm{YY}_6 + \mathrm{YY}_7$
$\mathrm{ZZR}_6 = \mathrm{ZZ}_6 + \mathrm{YY}_7$
$\mathrm{MYR}_6 = \mathrm{MY}_6 - \mathrm{MZ}_7$
$\mathrm{XXR}_7 = \mathrm{XX}_7 - \mathrm{YY}_7$

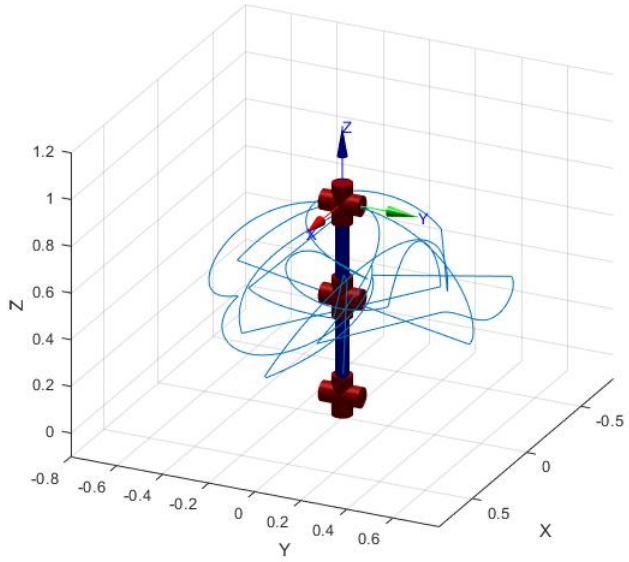


Figure B.7: Movement (in m) of end-effector during a portion of the trajectory *Global-PTP_1*.

inspired by the work done in [Gautier et al., 2012].

After sampling at 1000 Hz, the pertinent signals are filtered with a non-causal zero-phase 2^{nd} order digital Butterworth filter with cutoff frequency of 3.5 Hz, both, in the forward and reverse directions to avoid time-lag. Second, velocities and accelerations are calculated off-line using the central 1^{st} and 2^{nd} order derivatives from the filtered position to avoid time-lag and reduce noise effect. Third, a down-sampling of order 20 is applied to obtain an overall signal of 50 Hz and reduce the computational cost of the identification without loosing information. Finally, border effects are deleted, as well as all those points where the joints in study have a velocity lower than $0.05\frac{rad}{s}$. This is done to avoid problems due to the usually adopted non-smooth around 0 velocity friction models shown in Equation (1.13) and Equation (1.14).

In the case no filtering is needed, just the decimation and differentiation stages are applied.

B.3.4 Analysis of signals

In this subsection we show the commanded and sensed signals and the respective difference between each other of: the position in Figure B.8, the speed in Figure B.9, the acceleration in Figure B.10 and the torque in Figure B.11 from the non-filtered *Global-PTP_1* trajectory.

These figures are important in order to analyze the behavior of the controller, and verify if the decoupling of the non-linear model is well-done by the KUKA controller. The error

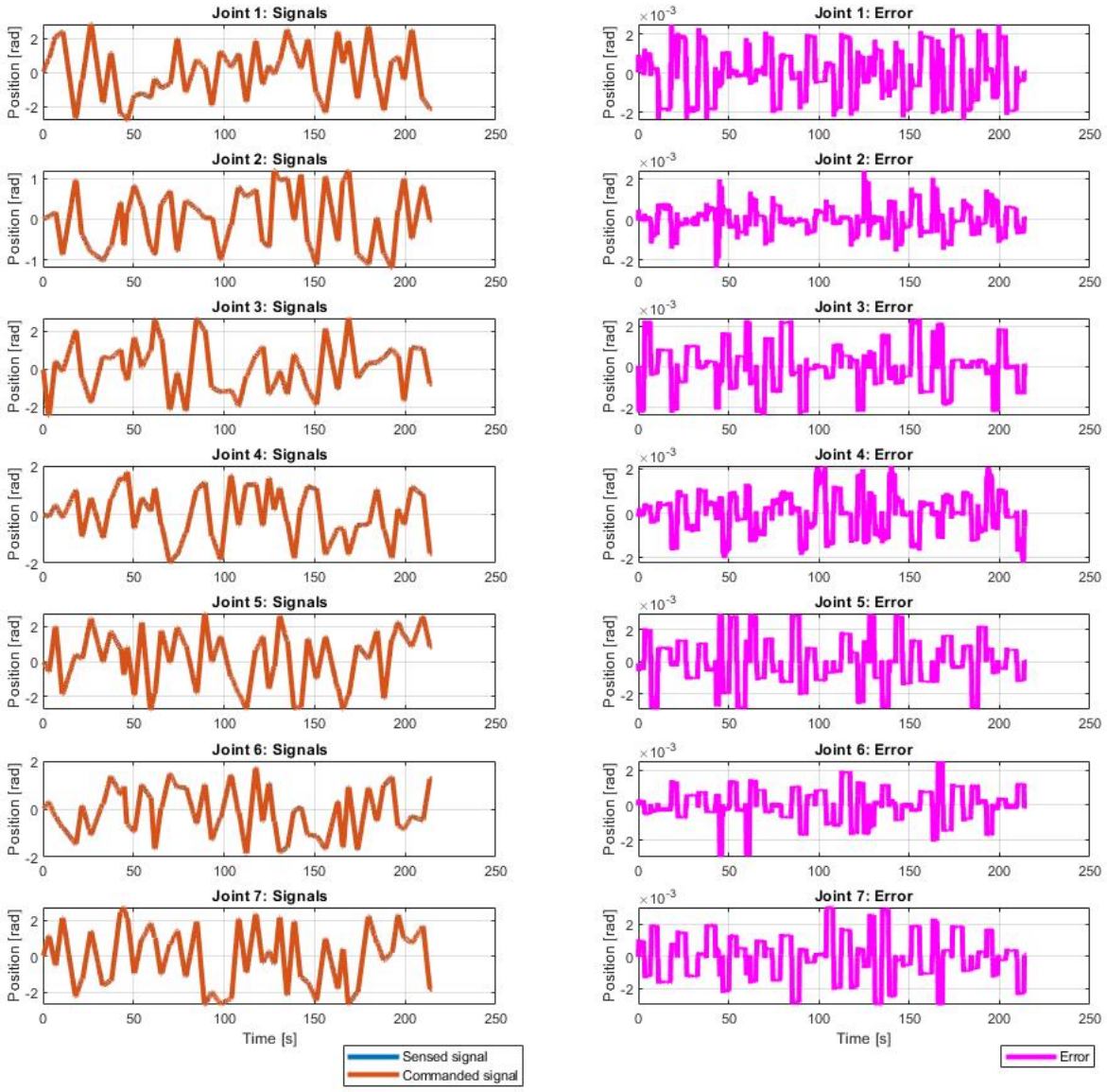


Figure B.8: Position signals and their difference from the non-filtered trajectory *Global-PTP_1*.

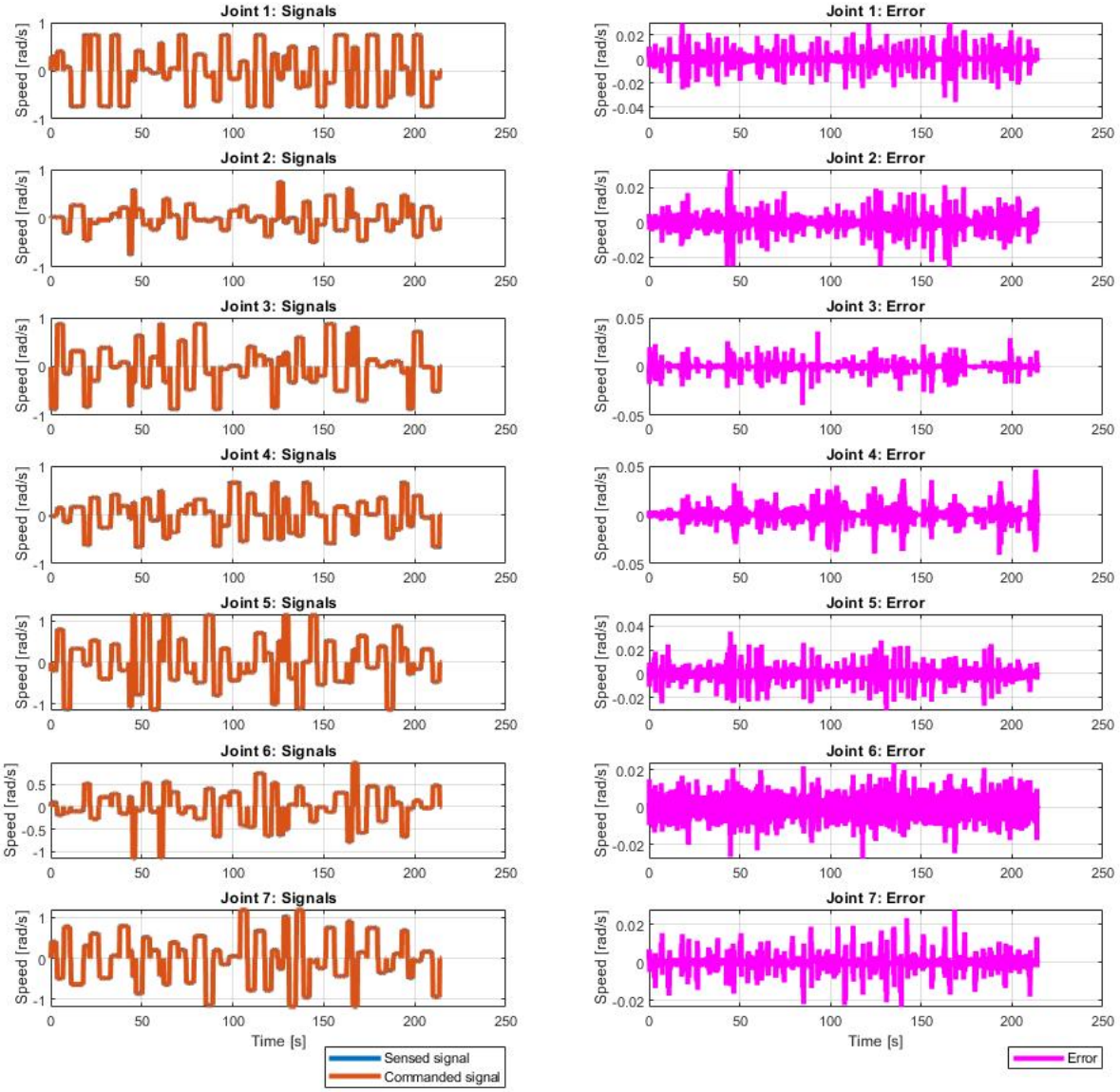


Figure B.9: Velocity signals and their difference from the non-filtered trajectory *Global-PTP_1*.

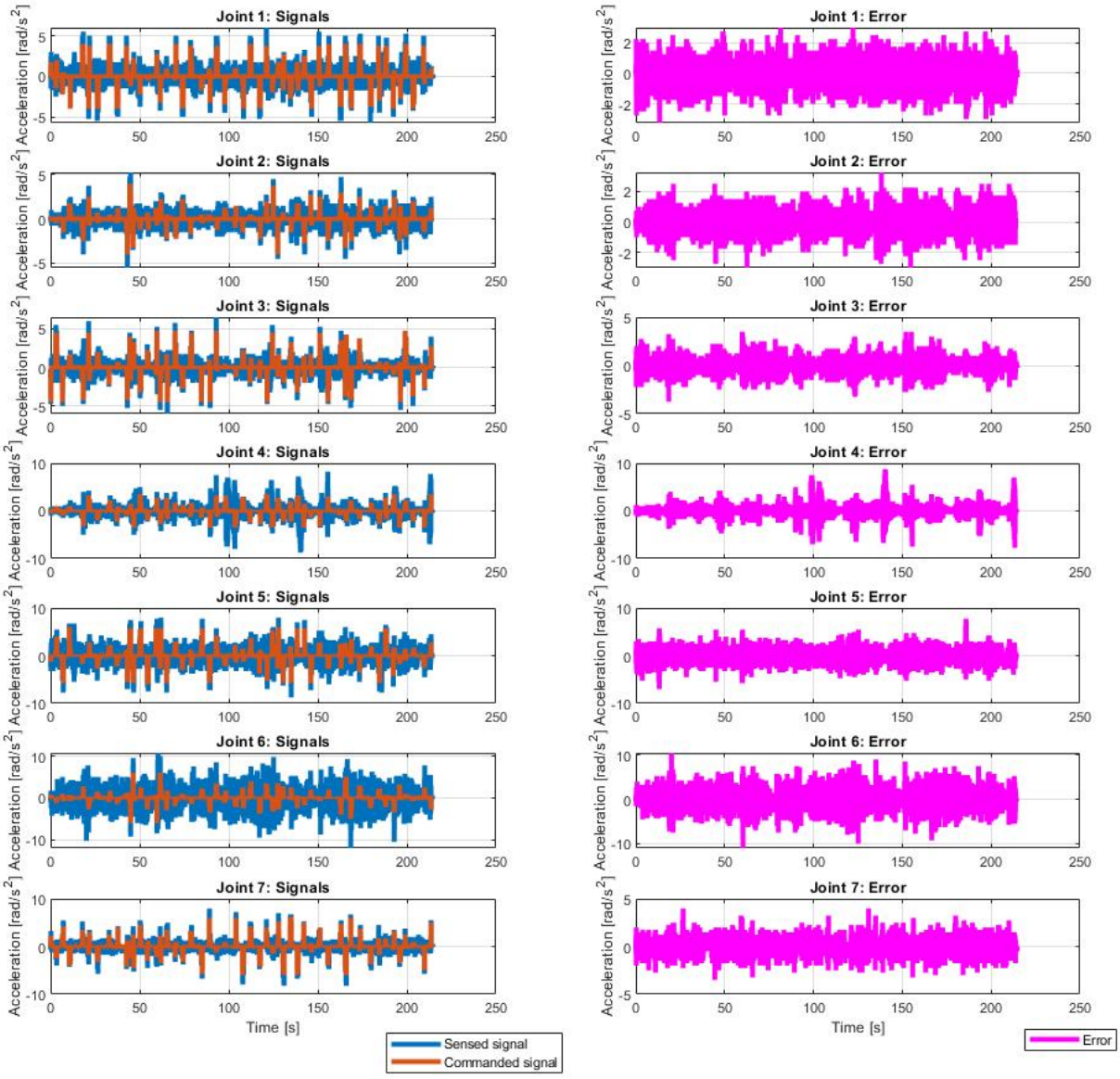


Figure B.10: Acceleration signals and their difference from the non-filtered trajectory *Global-PTP_1*.

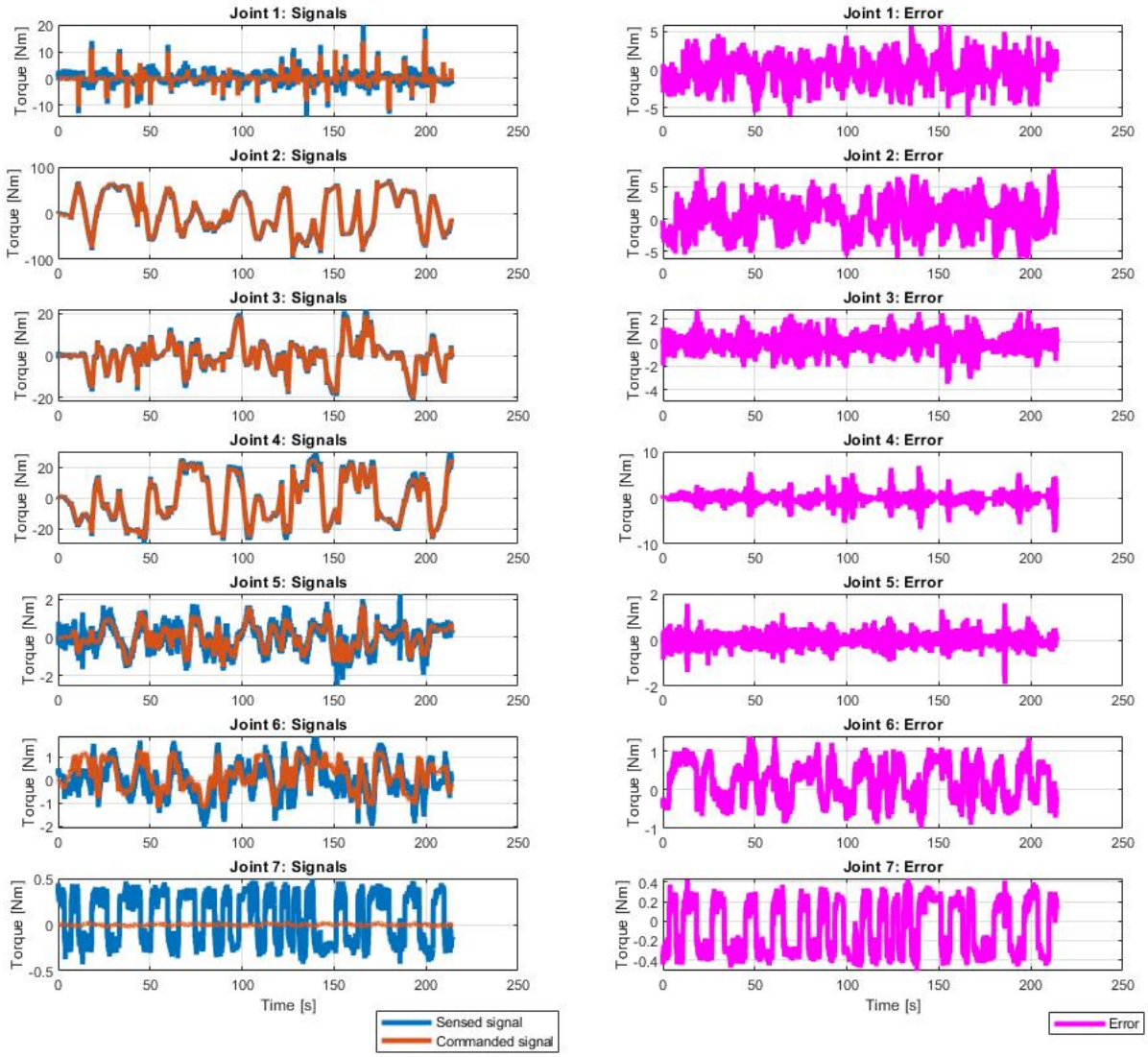


Figure B.11: Torque signals and their difference from the non-filtered trajectory *Global-PTP_1*.

between the sensed position and commanded position is lower than 1%, showing good performance of the KUKA controller. Moreover, the position error is bounded and has a profile similar to the one of the velocity. This means that, for model-based controllers, the inertial effects are well decoupled and compensated with the internal model. As explained in [Khalil and Dombre, 2002], an adequate linearization and decoupling of the equations of the model, provides a uniform dynamic behavior for every joint whatever the configuration of the robot. Moreover, it can be seen that noise has significant effects in the torque and acceleration signals, what explains the need of the filter stage for some of the identification methods.

Appendix C

MESSII dataset

In this Appendix, we present the *MESSII Dataset* (**M**anipulator **E**xperimental **S**y**S**tem **I**dentification and **I**nteraction Dataset). This novel dataset aims at providing numerous sequences of movements of the KUKA iiwa manipulator, in order to allow users to evaluate methods for identification and estimation of the robot's dynamic parameters and signals, respectively. Different movements of the manipulator are provided, including trajectories moving one or multiple joints at the same time, trajectories that are specially designed for parameter identification, and trajectories including a payload attached at the end-effector of the manipulator. The information obtained from the propioceptive sensors of position and torque is presented. Furthermore, as physical Human-Robot Interaction (pHRI) is the main application of this robot, sequences with interaction with a person are provided to estimate the forces applied, where a force-torque sensor acts as the ground truth. The dataset can also be used in ROS to evaluate real-time methods as information is presented in rosbags files. Possible applications are given, highlighting the advantage of the dataset to deal with state-of-the-art challenges without the need of the real robot nor doing new and complex experiments. The dataset, as well as more information and tools, are publicly available in https://messii-dataset.enit.fr.

C.1 Introduction

Physical human-robot collaboration is a growing area in robotics, where the robot and the human do not only share their work-space, but they also share objectives and tasks [De Santis et al., 2008, Ajoudani et al., 2018b]. The fact that a human can interact in an unpredictable way with the robot, brings up new challenges in terms of control and design. These involve not only the analysis of the task's performance and the manipulator's integrity, but also of the human safety [De Luca et al., 2006, Haddadin et al., 2009]. For this, besides a new generation of well-performing robotic manipulators and rich proprioception sensing [Albu-Schäffer et al., 2007a], a suitable control has to be designed, which is usually of the model-based type [Siciliano et al., 2010]. The more the model approaches reality, the simpler will be to design the controller to interact with the human in a compliant and human-friendly way.

Dynamic equations of cobots have been widely studied in literature [Khalil and Dombre, 2002] which relate signals as the torque, position, velocity and acceleration of joints between each other and also with dynamic parameters as the inertia, mass and center of mass coordinates of the links. The signals can be measured by choosing a correct set of sensors, whereas

there are three main methods to obtain the numerical value of the dynamic parameters: physical experiments carried out on the manipulator's individual parts, CAD techniques and dynamic parameter identification methods. The fact that for carrying physical experiments, each link needs to be isolated from the others, thus the manipulator needs to be disassembled, makes this option non-viable most of the times. Moreover, most available collaborative robots in the market are usually not intended for research, and manufacturers do not provide crucial information like its CAD model and related parameters. Because of these reasons, in general, the only feasible option is dynamic parameter identification.

In order to achieve this purpose, there is a large amount of parameter identification methods available in literature (see the survey in [Leboutet et al., 2021]). Moreover, several works on dynamic parameter identification of industrial collaborative manipulators can be found in bibliography as the ones on the: KUKA iiwa [Stürz et al., 2017], KUKA LWR4+ [Jubien et al., 2014a], Franka Panda Emika [Gaz et al., 2019], ABB IRB14000 (YuMi) [Taghbalout et al., 2019], UR3 and UR5 [Raviola et al., 2021].

The process of parameter identification consists of several steps comprising the modeling, the generation and execution of enough exciting trajectories, the data processing, the parameter estimation itself and the results evaluation (for more information about system identification refer to [Ljung, 1998]). This task can be time consuming, being usually a try and error exercise. In addition to this, the manipulator may be unavailable, e.g. it is being used to perform other tasks or it is located in a place where it cannot move freely. One solution could be to design a simulation in order to test methods and algorithms. However, simulated data will always be a simplification of reality, where unmodeled effects both in the deterministic (dynamic effect) and stochastic (noise) parts are present.

In other disciplines, the solution to these issues has been found in shared datasets. They are a positive product of the advent of big-data and open science. They are a way to give the research community and society in general the tools to both: verify publications based on the dataset either because it may have experiments that are difficult, tedious, or impossible to replicate, and also give the possibility to fellow multidisciplinary researchers to continue the research work and deepen the knowledge on one topic from different points of views. This is why there are some datasets that have become quite famous as [Geiger et al., 2013] and [Sturm et al., 2012].

The aim of this work is to present a complete and consistent data-set of the collaborative 7-*dof* KUKA iiwa 14 R820 manipulator which allows the scientific community to try state-of-the-art and new algorithms throughout the steps of the identification process in different scenarios without the need of a physical robot. Classical and collaborative scenarios that allow the identification of the manipulator, load and interaction parameters are considered.

The Appendix is structured as follows: first, a state-of-the-art review of other dataset using robotic manipulators is done. Second, we present the robotic platform used, including the robot, the payload, the sensor and the computer (refer to Appendix B). Then, the new MESSII dataset is presented: its structure, the available data and how it was obtained. After that, some applications and challenges are shown. Finally, conclusions are made.

C.2 Related work

In the area of robotic manipulators, there are several datasets dealing with the problem of grasping and manipulation of objects (for an extensive survey see [Huang et al., 2016]). Most of them are datasets containing information of activities humans carried out in order to make the robots learn from them. [De la Torre et al., 2009] collects multimodal data of human behavior in a cooking task using RGB cameras, accelerometers placed on the human, microphones, and motion capture system. [Mandery et al., 2015] presents a whole-body human motion database consisting of captured motion data from the human and objects being manipulated using a marker-based motion capture system. [Maurice et al., 2019] records persons carrying out industrial activities for an ergonomic study using optical motion capture, inertial motion capture (IMUS placed on the body), hand contact and finger flexion sensors, and video cameras. [Pirsiavash and Ramanan, 2012] present a database with first-person camera views of daily living activities. [Roggen et al., 2010] places a large amount of sensors in an environment in order to carry out machine recognition on human activities.

These are just some of the huge amount of datasets available which are mainly focused on the study of the human movement in order to develop image-recognition and machine-learning algorithms, which could be subsequently used to teach a robot how to behave in different situations. On the other side, there are some datasets that focus on the behavior of robots. [Gao et al., 2014] is a surgical activity dataset for human motion modeling, captured using the *Da Vinci* surgical system and eight surgeons performing different tasks. Although it is a tele-operated system, it presents kinematic data from the robot (Cartesian positions, orientations, velocities and gripper angle) and stereo video data captured from the endoscopic camera. [Dasari et al., 2019] introduces a huge dataset of videos of different manipulators carrying out different trajectories. [Levine et al., 2018] introduces a dataset of pushing and grasping motions for self-supervised learning objectives using the KUKA IIWA manipulator. To the authors knowledge, it represents the only dataset where, besides position measurements and camera's images, measurements of torque from propioceptive sensors are also available.

Most of these publications deal with the problem of behavior learning for robots based on video information. However, as far as the the authors are concerned, there is no dataset focused on dealing with measurements of propioceptive sensors of collaborative manipulators and interaction forces from external sensors, in order to carry out dynamic parameter identification and to estimate human contact. These are tasks that are needed in almost every implementation of a collaborative manipulator, thus its utilisation becomes essential for robotics.

C.3 Robotic platform

Please, refer to Appendix B. The elements used to build the dataset can be seen in Figure C.1, the manipulator's structure in Figure B.2 and its communication framework in Figure B.5.

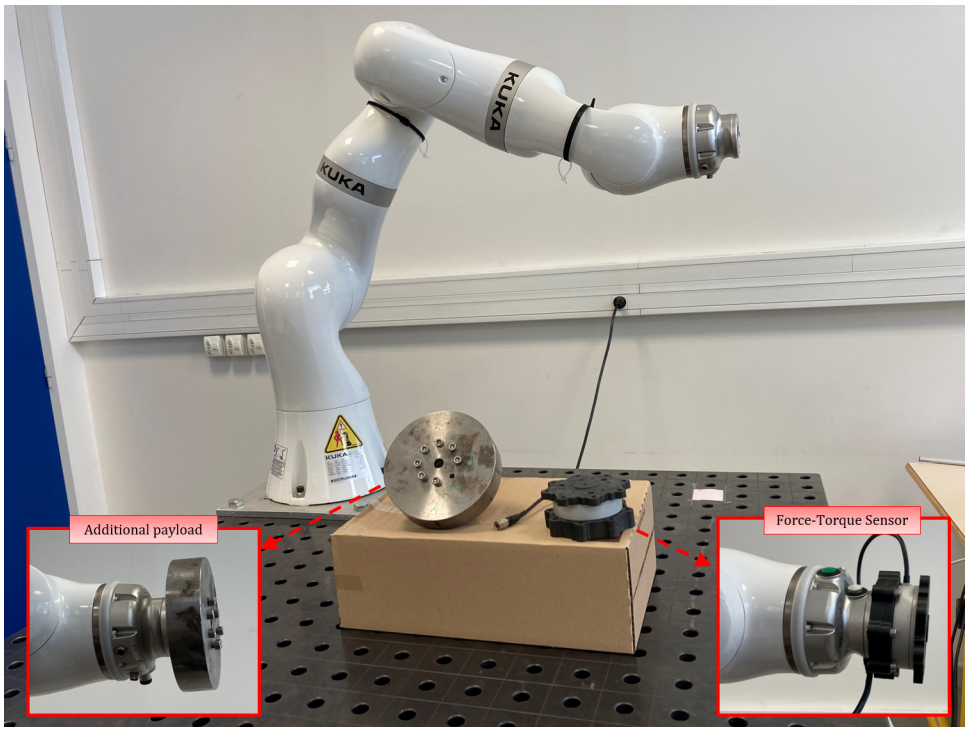


Figure C.1: IIWA robotic manipulator, payload and force-torque sensor used to build the dataset.

C.4 The dataset

C.4.1 Structure

The dataset is structured as presented in Figure C.2. The three main folders (*Standalone*, *ROS* and *RAW*) contain the same sub-folder tree. Hence, each of these three folders contains the four sub-folders corresponding to the possible application, either for the identification of the robot, the sensor, the load or for pHRI. The *Standalone* and *ROS* folders present the data gathered and processed (filtered). The only difference is that *Standalone* presents data in several *.log* files, whilst *ROS* presents the information in *rosbags*. The *RAW* folder presents the non-processed data.

The sub-folders description for each of the four possible applications is shown in Figure C.3. For the case of robot identification, we provide global and sequential (only one joint moves at a time) trajectories. More details about their applications are provided in Section C.5.1n. Similarly, for load identification, we provide global and sequential trajectories. As for the sensor identification, we provide sequential trajectories. Whether the trajectories are global or sequential, we provide several examples with variations of the type of trajectory or the posture of the robot while performing the movement. Finally, for the case of pHRI, we provide two types of interactions with the environment (a person). First, the interaction appears exclusively at the end-effector level (i.e. forces captured by the F/T sensor) and the

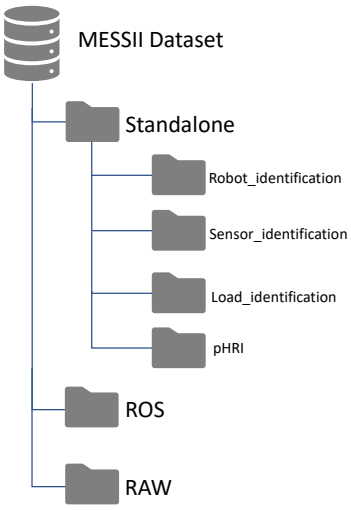


Figure C.2: Structure of the *MESSII Dataset*.

second case contains interactions at different points of the kinematic chain of the robot. For both of these cases, we provide sequences where the interactions happen while the robot is moving (under Impedance Control) and when the robot is in a static condition.

C.4.2 Data collection

C.4.2.1 Trajectories

The dataset is composed of two types of trajectories. The so-called global trajectories where all the joints move at the same time, and the individual or sequential trajectories where just one joint moves. Design of an exciting enough trajectories is an area of identification by itself and has attracted a lot of attention [Pukelsheim, 2006]. It can be summarized in a non-linear optimization problem of the condition number of the regression matrix of the dynamic model, with multiple linear and non-linear constraints. This stage is already provided in the dataset, and in this section a brief explanation of each of the trajectories will be given.

On the one hand, there are three types of global trajectories present in the dataset: *PTP*, *SPL* and *DS*. The first two are obtained from the PTP (point-to-point) and SPL (spline) motions that the KUKA controller provides. 50 points are randomly selected inside the manipulator's reachable workspace, and then the corresponding interpolation is done by the controller. This way of designing trajectories has proven to be enough for identification in

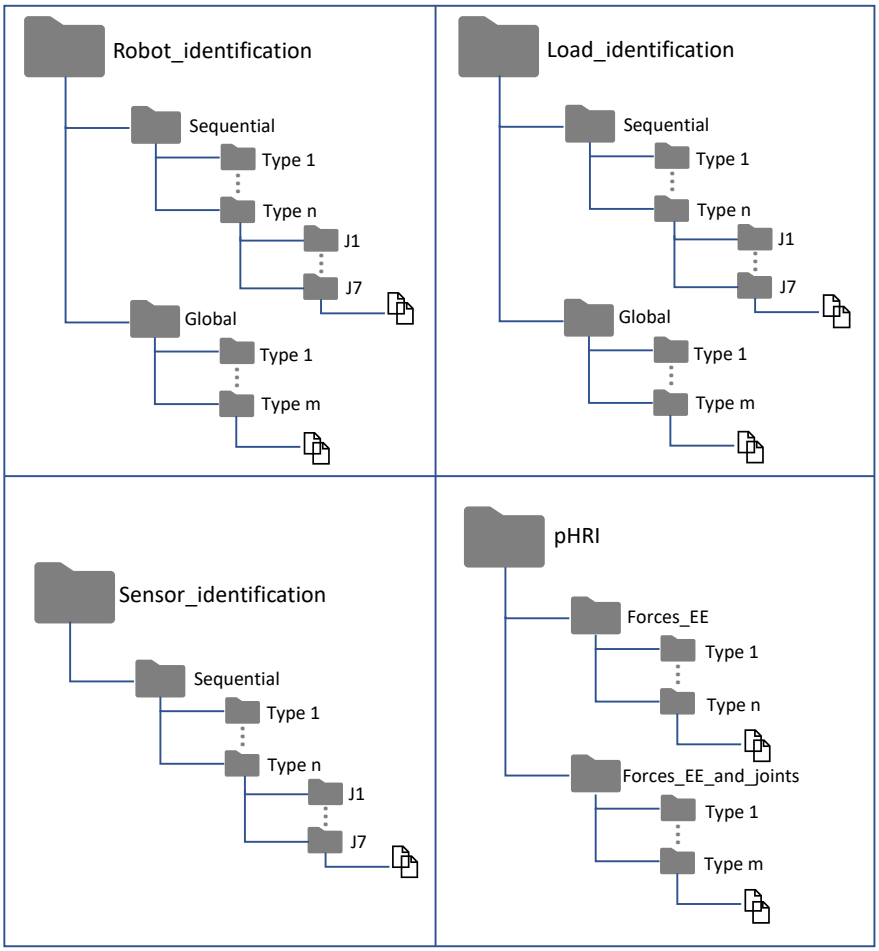


Figure C.3: Sub-folders description of the *MESSII dataset*.

[Jubien et al., 2014a]. The third way is by means of the DS (*DirectServo*) option provided by KUKA. Here, the well-known parameterization based on finite Fourier series presented in [Swevers et al., 1997b] was used. A 5^{th} order sum of harmonics was chosen, and constraints regarding position and velocity limits, zero initial joint position, velocity and acceleration were considered in the optimization problem.

On the other hand, three type of sequential trajectories were carried out (all of them moving one joint while the others are locked). First, a trajectory pretended for friction identification was designed using PTP motion, in which the joint in study moves from one point to another at constant different speeds. In this way, the inertial effects are reduced. Contrary, another trajectory was designed, also by means of a PTP motion, for inertial parameters identification in which the intervals where the joint is accelerating or decelerating are more significant than those of constant velocity. Finally, a trajectory from a finite Fourier series, as the one explained in the previous paragraph, is designed using the DirectServo motion.

C.4.2.2 Synchronization

The computer has two parallel processes, as shown in Figure B.5. The main one is in charge of saving the information coming from the manipulator's controller at a rate of 1000 Hz with its respective timestamp. The second process saves the information of the force-torque sensor on a shared memory also at 1000 Hz, which the main process then reads and saves on the corresponding file. This method will ensure that all the data is time-stamped and well-synchronized by having a maximum shift between the measurements of the manipulator and those of the external sensor of 1ms in the worst-case scenario.

C.4.3 Data processing

C.4.3.1 Filtering

Please, refer to Section B.3.3.

C.4.3.2 Rosbags

To allow the use of the dataset with ROS, the same information of the *Standalone* version is provided by several rosbags. The main advantage of the rosbag version of the dataset is that it can be used with other predefined nodes as well as in real-time applications. For instance, real-time identification of the robot's parameters or force observers can be implemented and tested.

The information presented in the standalone version is replicated in the rosbags. As mentioned, the structure is the same, but instead of having several files for each experiment,

they are all gathered in a single rosbag. Each rosbag contains the following topics :

- **FT_sensor :** Containing the three forces and three torques as a wrench vector of the type *geometry_msgs/WrenchStamped*.
- **jointCommanded:** Containing the commanded states (position, velocity and torque) for the seven joints as a *sensor_msgs/JointState* type of message.
- **jointMeasured:** Containing the measured states (position, velocity and torque) for the seven joints as a *sensor_msgs/JointState* type of message.
- **jointExternal:** Containing the external states (position, velocity and torque) for the seven joints as a *sensor_msgs/JointState* type of message.

Details on how to run the rosbags, listen to the available topics and create new rosgabs are included as tools in the dataset web.

C.4.4 Calibration

As the recorded measurements from the KUKA iiwa manipulator are from propioceptive sensors and in the joint-space, no external calibration needs to be done. However, the robot comes with a load determination tool, which will have a direct effect on the calculation of the *commanded signals* and *external torque*. In this work, the load in the controller was set to zero for the trajectories without load, with the sensor and with human interaction. For the trajectories where the load is placed, the estimated values were already described (see Section B.1.2). This means that, in some experiments, the *external torque* will have just the contribution of friction and uncertainties of the model that is used by KUKA to control the robot, while in others, it will also include the effect of the sensor's weight, paylod's uncertainties and the possible human-robot interaction.

Furthermore, the sensor is well-aligned to the axis of the Media Flange thanks to the "location pins" and the way the coupling was designed (see CAD model present in the dataset web). As for the sensor calibration, each trajectory done in the pHRI part begins with a few seconds of the robot in static condition and with no interactions. This, combined with the identification of the after-sensor dynamic, should allow to recover the forces coming exclusively from the interactions.

Finally, camera recordings are also included to visually depict the interactions with the person in the respective pHRI tests.

C.5 Applications and challenges

C.5.1 Parameter identification

The parameter identification procedure can be summarized in the seven steps shown in Figure 1.6. It is an iterative procedure which will ultimately depend on the application of the obtained model and its expected performance. These tasks can be time consuming, thus, being able to test the whole procedure before executing it on a specific manipulator can be of advantage to users. This is the reason why this dataset facilitates steps 3, 4 and 5 of the procedure, allowing users to test their own overall solution on real data without the need to own and code a manipulator.

Each of the steps can be solved in several ways, presenting interesting issues and challenges, and some of them will be briefly mentioned. First, in collaborative robotics there are still some questions regarding which is the best way to model the joint flexibility, payloads and friction [De Luca, 2000, Van Geffen, 2009, Raviola et al., 2021]. This dataset allows the user to test and compare different models. For instance, the explicit dynamics, the energy or the power methods can be used to derive the model, and analytical closed-form rules or numerical tools as the QR and SVD decomposition can be used to obtain the reduced model.

Furthermore, although we propose filtered data, ready to be used on identification methods, we also provide the raw information, to let users apply their own filtering stages. Digital signal processing textbooks address this issue extensively [Anderson and Moore, 2012], and even though this stage will usually strongly depend on the application, being able to test the whole data processing step on real data can be useful.

In addition, the proposed dataset allows the user to test different parameter identification methods (step 6 in Figure 1.6). The selected method can range from a simple en-bloc method of a system linear with respect to the parameters, such as the Least-Squares solution, to neural networks and complex recursive solutions that could be applied online. This last topic is of much interest, as manipulators working on unpredictably changing environments require an online evaluation in order to estimate interactions and detect faults and collisions. The identification can be done in a global form, meaning that all parameters are estimated at the same time, or sequentially, where parameters are excited at different moments. Furthermore, the identification methods can include physically feasibility constraints to obtain parameters which are congruent with reality, which is a topic that brings a lot of attention lately [Sousa and Cortesao, 2014, Janot and Wensing, 2021]. Moreover, the provided signals also allow the identification of the model that is included in the controller, and by means of black-box identification methods, the structure of the controller. This enlarges widely the application and utility of the dataset, as many aspects of system identification can be addressed.

Finally, different techniques of model validation can be tested and new ones can be proposed depending on the desired application, e.g. tools can be used as analysis of residuals, analysis of estimates and analysis of model fit [Ljung, 1998].

Application example. In order to show the utility of the dataset and emphasize one of the mentioned challenges, we have carried out a simple Least-Squares identification of the parameters of joint 4 using the *Sequential-Inertia-J4-Filtered* trajectory, and knowing already the parameters from joints 5, 6 and 7. The essential estimated parameters and their respective standard deviation are shown in Table C.1, the position, velocity and acceleration of joint 4 are shown in Figure C.4 and the validation of the estimated model is done in Figure C.5.

The identified essential parameters are five: ZZR_4 and MYR_4 are regrouped inertial parameters, and f_{c_4}, f_{v_4} and τ_{off_4} are three friction parameters corresponding to a simple model including Coulomb friction, viscous friction and a torque offset due to the asymmetrical Coulomb friction and other offsets introduced by measurement equipment, respectively. It can be seen that the torque reconstruction is excellent, being the percent error less than 0.5%. However, when analyzing the numerical value of the two parameters corresponding to the Coulomb friction, it can be noticed that it is theoretically impossible to have $\tau_{\text{off}_4} > f_{c_4}$, as it will lead to having the force of friction in the same direction as the velocity for a specific range. There are many alternatives to try to solve this issue. Two of them are: either the model does not explain the reality in a good way, for example, not being able to explain the Stribeck effect, thus another model has to be chosen, or constraints can be included in the optimization process of identification to ensure physical feasibility.

Table C.1: Identified essential inertial parameters and their respective relative standard deviation of joint 4.

Param.	Value	$\%\sigma$
ZZR_4	0.8600	0.06
MYR_4	2.3283	0.05
f_{c_4}	0.0388	7.29
f_{v_4}	0.1154	2.63
τ_{off_4}	0.4584	1.66

Although the identification methods to be used with this dataset can be generalized to different robots, the identified parameters obtained using the information provided can be compared with previous works in [Stürz et al., 2017, Hennersperger et al., 2016, Xu et al., 2020].

C.5.2 Human interaction

In the context of pHRI, the safety of the person is the first and most important layer that has to be addressed. In order to accomplish that, the physical contact has to be recognized to react accordingly (either to detect collisions and react [De Luca et al., 2006], or to collaborate based on the forces applied [Mujica et al., 2023]). Recent manipulators, such as the KUKA iiwa, provide joint torque sensors that can be used for those purposes. Based on these sensors, torques on the joints can be used to detect and recognize collisions. Furthermore, in quasi-

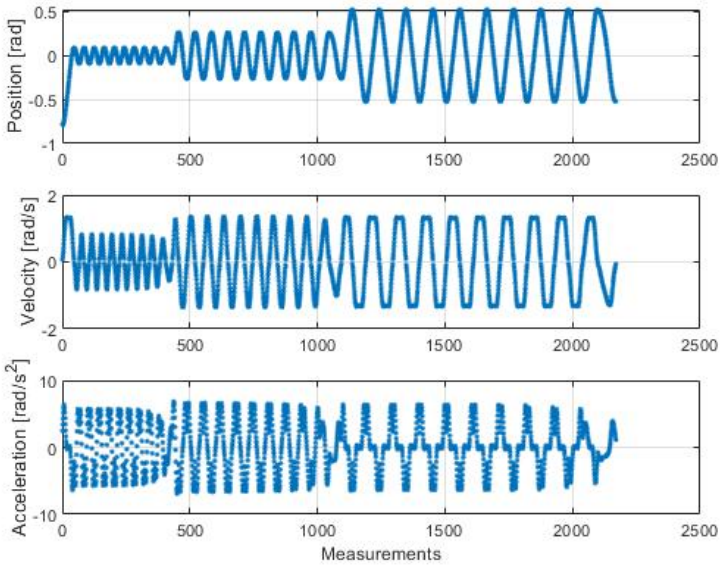


Figure C.4: Position, speed and acceleration signals of joint 4 used for identification.

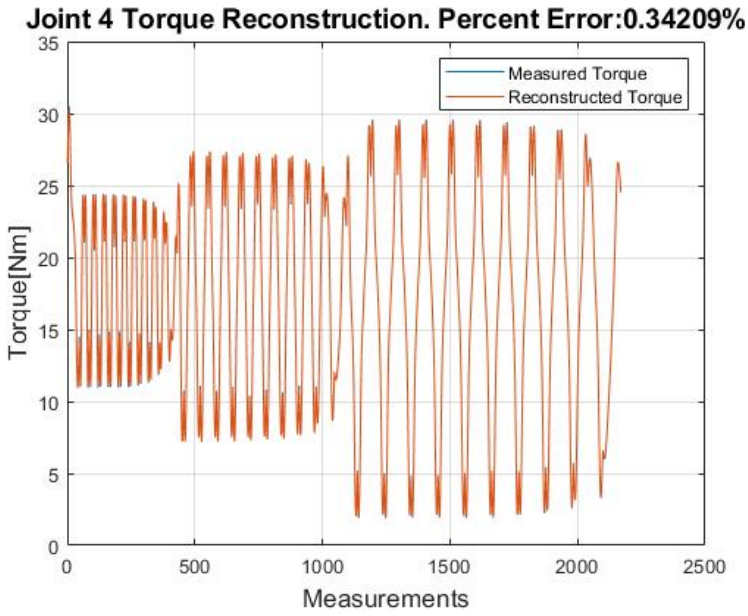


Figure C.5: Comparison between actual and reconstructed torque of the trajectory in Figure C.4 with the identified essential parameters of Table C.1.

static condition, forces at different levels of the kinematic chain can be reconstructed through the virtual work principle. Therefore, the need of a Force/Torque sensor at the tip of the robot can be avoided, reducing the costs associated. This can be done by considering:

$$F_e = J^{-T}\tau_e, \tag{C.1}$$

where F_e is the external wrench (forces and torques) at the end effector level, τ_e are the *external joint torques*, and J is the Jacobian matrix of the robot.

Applying this method in the *Sequence_static_1* of the dataset, the forces applied at the end effector can be reconstructed as seen in Figure C.6, using the external joint torque provided by the robot. However, in the figure, it can be noticed that the reconstructed force and the one of the F/T sensor (considered as the ground truth) present important differences. These differences can be linked to Equation (C.1) where a quasi-static condition is considered and this may not be the case. Also, the presence of non-modeled dynamics in the model used by KUKA for the external torques can produce important differences (e.g., in the first 5 seconds, when no external forces or torques appear). Furthermore, in the proximity of a singularity, the inverse of the Jacobian is ill-defined, producing errors in this estimation. In

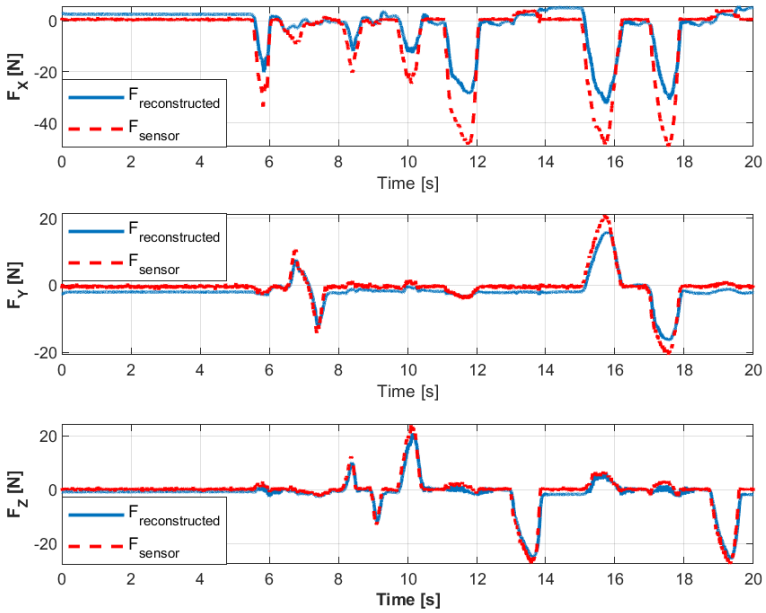


Figure C.6: Comparison between reconstructed forces and the real ones obtained with the F/T sensor.

recent years, improved methods to estimate forces at different levels of the kinematic chain have been considered [De Luca et al., 2006]. They presented advantages, like being able to estimate forces even without the need of the joint torques, but also some complexities like requiring a good model of the robot (e.g. non-modeled friction would easily degrade the results). For these reasons, this remains an open challenge. Different methods consisting of

observers, filters or frequency analysis, can be used to identify, reconstruct and classify the interaction forces applied to the robot. In summary, this part of the dataset provides several sequences that allow testing and evaluating these methods on a real robot.

C.6 Conclusion and future work

This work presented the main features of the novel MESSII dataset, for the collaborative robot KUKA iiwa. The dataset contains an important number of sequences with different movements of the manipulator, with measurements of the propioceptive sensors and a F/T sensor attached to the end effector. This information allows the community to test and evaluate methods for identification of the robot, the sensor, and the load included, without the need to have the real robot or design and code the trajectories. Furthermore, as the dataset can be used in ROS to replay the sequences in real-time, online identification methods can be assessed. As the robot is used, in general, for human-robot interaction, the dataset also provides sequences where a person interacts with the robot. The goal is to allow the use of novel observers, filters and estimation to reconstruct the forces that appear at different levels of the kinematic chain, and compare them with the measurements of the Force/Torque sensor (ground truth). Beyond the structure and elements of the dataset, this work presents examples of applications along with the main challenges that might motivate other researchers to use the MESSII dataset. In future works, this dataset will be enhanced with data from other robots as well as the use of different visual sensors to provide 3D information of the scene as well.

Appendix D

PC-IDIM-IV on TX40 manipulator

This Appendix validates the PC-IDIM-IV method developed in Section 2.3 on the TX40 industrial robot and compares its results with other state-of-the-art methods.

D.1 Introduction

Robot identification consists of identifying the dynamic parameters, mass, center of mass, rotational inertia, and friction that influence the relationship between applied forces and resultant accelerations. It is a classical problem, with results spanning recent decades, see the survey [Leboutet et al., 2021]. Recent years have witnessed a renew of interest in this problem due in part to a rapid increase in robotic hardware platforms capable of accurate model-based control [Ayusawa et al., 2014, Jovic et al., 2016, Villagrossi et al., 2018], and force-controlled actuators, [Semini et al., 2015, Wensing et al., 2017b].

Conventional approaches exploit the linearity of the inverse dynamic model (IDM) to the dynamic parameters allowing identification to be formulated as a linear least-squares problem. The most popular identification method is referred to as the Inverse Dynamic Identification Model with Least-Squares estimation (IDIM-LS method) [Gautier et al., 2012]. However, since we identify robots with closed-loop procedures, it induces a correlation between errors that may lead to inconsistent IDIM-LS estimates, even in the case of proper data filtering [Young, 2011, Janot et al., 2013b]. Instrumental variable (IV) techniques allow the users to get consistent estimates. In [Janot et al., 2013b], the authors have developed an IV approach called IDIM-IV that combines the direct and inverse dynamic models like the direct and inverse dynamics identification models (DIDIM) approach [Gautier et al., 2012, Janot et al., 2013b]. However, like DIDIM, IDIM-IV does not guarantee that the direct dynamic model (DDM) will be well-posed during its iterations because of modeling errors. The use of physical constraints could be an opportunity to address this deficiency for IDIM-IV.

The set of all possible inertial parameters ensuring a positive definite mass matrix is known to be convex. A necessary and sufficient condition for physical plausibility was recently described using 4×4 linear matrix inequalities (LMIs) [Wensing et al., 2017a] posed over the 10 inertial parameters of each rigid body, improving the approach adopted in [Sousa and Cortesao, 2014]. The general approach of combining IDIM-LS and physical constraints is called the Physically-Consistent IDIM-LS method (PC-IDIM-LS) and this method was applied in [Stürz et al., 2017, Sousa and Cortesao, 2019, Gaz et al., 2019]. However, these

works have not treated the statistical properties of PC-IDIM-LS.

In [Janot and Wensing, 2021], the authors provide a complete statistical analysis of PC-IDIM-LS and propose to insert the physical constraints into the DIDIM method. In this way, we get estimates that are statistically and physically consistent. This new method, called PC-DIDIM, has been validated on a 6-degrees-of-freedom (*dos*) robot. Further, results showed that statistical consistency is the critical problem of robot identification while inserting the physical constraints. So, it would be interesting to include these constraints in IDIM-IV. In this manuscript, we propose a new constrained IV approach, called PC-IDIM-IV, by employing IV criteria from Econometrics, the regrouping formula from Robotics, and standard statistical hypotheses made on the signals. PC-IDIM-IV consists of two nested iterative algorithms: an outer one that is the standard IDIM-IV approach and an inner one that accounts for the physical constraints solved by a Gauss-Newton algorithm. Experimental results obtained with the TX40 manipulator show the feasibility of PC-IDIM-IV.

Having described the background of robotic modeling and identification in Chapter 1 and the PC-IDIM-IV method and algorithm in Section 2.3, details on the experimental setup of the TX40 robot, as well as experimental results, are shown in this Appendix.

D.2 Robotic manipulator, exciting trajectories and data acquisition

The TX40 robot (see Figure D.1) has a serial structure with six rotational joints and is characterized by a coupling between the joints 5 and 6 [Janot et al., 2013b]. 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. 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. The bandwidth of the first three position closed-loops is 10Hz.

Figure D.1: TX40 manipulator.

The trajectories $\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r$ provide a conditioning number of 200 for $\mathbf{W}$ to avoid numerical

issues [Gautier and Khalil, 1992]. To evaluate the identification methods involved in this study, all the data is stored with a sampling rate of $f_m = 5kHz$. 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 methods. All cross-validation data is 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 [Janot et al., 2013b] for technical details).

We compare PC-IDIM-IV with PC-IDIM-LS, which can be considered as the reference method, IDIM-IV and PC-DIDIM. The reason why we do not consider DIDIM method is given later. For the comparison, we investigate three scenarios: one with actual data, a second one with downgraded data, and a last one with modeling errors. For the three scenarios and for all methods, data is filtered according to the process described in [Gautier et al., 2012].

Finally, all the simulations are executed on a laptop equipped with an Intel Core i5-10400H processor, 16.0 GB of RAM (DDR4 SDRAM technology), and a capacity of 2 TB. MATLAB version 2021-B is used.

D.3 Experimental results

D.3.1 First scenario: actual data

With actual data, PC-IDIM-IV, IDIM-IV and PC-DIDIM converge in 5 iterations. For any method, the DDM simulation for an 8s trajectory and the calculation of the estimates need 5s. Besides, the Gauss-Newton algorithm needs 10 or 15 iterations to converge and it takes 2s or 4s. So, PC-IDIM-IV and PC-DIDIM converge in less than 1 minute which is acceptable.

The results obtained with the direct comparisons are given in Table D.1, while the results of the cross-test validations are given in Table D.2. We recall that the relative errors are calculated with $rel_{err} = 100 \cdot ||\boldsymbol{y} - \boldsymbol{W}\hat{\boldsymbol{\beta}}_X||/||\boldsymbol{y}||$, where the subscript X stands for PC-IDIM-LS, PC-IDIM-IV, IDIM-V and PC-DIDIM, respectively.

The information gathered in Table D.1 and in Table D.2 show that the results do not vary significantly, since the relative errors rel_{err} are close to each other for all the methods. Furthermore, the cross-test validation for the second joint with the first trajectory illustrated in Figure D.2 shows that the torque reconstructed with the PC-IDIM-IV estimates fits the actual one. We obtain the same plots with the other methods. It comes that the use of an IV approach and the physical constraints do not significantly improve the results when the actual data and appropriate data filtering are employed. These results are consistent with those previously published in [Gautier et al., 2012, Janot et al., 2013b, Janot and Wensing, 2021]. In this case, we have $\boldsymbol{W}_{st} \approx \boldsymbol{Z}_{st} \approx \boldsymbol{W}_{nf_{st}}$ yielding PC-IDIM-LS, PC-IDIM-IV and PC-IDIM-DIDIM estimates that are close to each other. Besides, since $\mathbf{R}_t$ (see Equation (2.11)) has a physical meaning because all the regroupings are based on the Hyugens's formula, so has $\hat{\boldsymbol{\beta}}_{IDIM-IV}$.

Table D.1: Relative errors obtained with direct comparisons for the PC-IDIM-LS, the PC-IDIM-IV, IDIMIV and PC-DIDIM methods - Actual data

Joint j	PC-IDIM-LS	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	5.2%	5.4%	5.4%	5.5%
2	5.0%	5.3%	5.3%	5.3%
3	5.0%	5.1%	5.1%	5.1%
4	5.6%	5.7%	5.7%	5.8%
5	7.2%	7.2%	7.2%	7.2%
6	7.1%	7.5%	7.5%	7.5%

Table D.2: Relative errors obtained with cross-validation for the PC-IDIM-LS, the PC-IDIM-IV, IDIMIV and PC-DIDIM estimates - Actual data

Traj.	PC-IDIM-LS	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	6.1%	6.4%	6.4%	6.4%
2	6.3%	6.8%	6.8%	6.8%
3	6.7%	7.0%	7.0%	7.1%

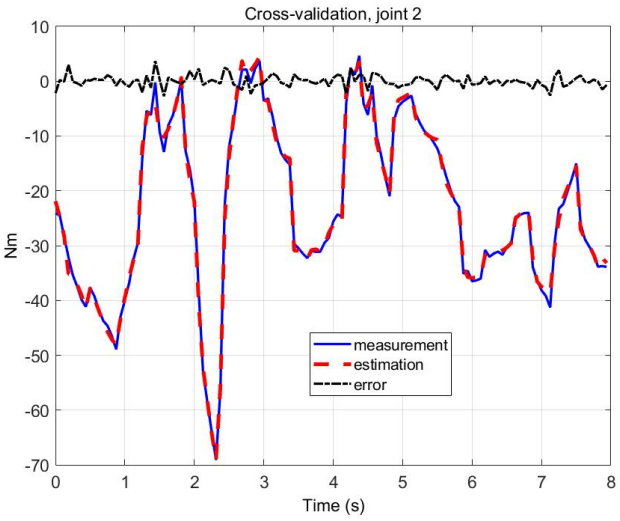


Figure D.2: Cross-validation, second joint and first trajectory, actual data.

Table D.3: Relative errors obtained with direct comparisons for the PC-IDIM-LS, the PC-IDIM-IV, IDIM-IV and PC-DIDIM methods - Downgraded data

Joint j	PC-IDIM-LS	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	25.3%	6.1%	6.1%	6.1%
2	24.7%	5.9%	5.9%	6.0%
3	25.2%	6.0%	6.0%	6.1%
4	25.8%	5.9%	5.9%	5.9%
5	27.5%	8.0%	8.0%	8.0%
6	28.0%	8.1%	8.1%	8.2%

Table D.4: Relative errors obtained with cross-validation, the PC-IDIM-LS, the PC-IDIM-IV, IDIM-IV and PC-DIDIM estimates - Downgraded data

Traj.	PC-IDIM-LS	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	26.4%	7.1%	7.1%	7.2%
2	26.5%	7.6%	7.6%	7.6%
3	27.8%	7.9%	7.9%	8.0%

D.3.2 Second scenario: downgraded data

Now, we downgrade position data from 2e-4 degrees per count to 2e-2 degrees per count. Such a situation is likely to occur when robots are operating in hostile or perturbed environments [Swevers et al., 1997b]. With downgraded data, PC-IDIM-IV and IDIM-IV converge in 7 iterations while PC-DIDIM converges in 6 iterations. One iteration of IDIM-IV needs 5 seconds to compute the estimates, and the Gauss-Newton algorithm needs 10 or 15 iterations to converge between 2 and 4 seconds. In this configuration, PC-IDIM-IV converges in 1 minute which is still acceptable.

From the relative errors obtained with direct and cross-validations gathered in Table D.3 and in Table D.4, we see that the PC-IDIM-LS estimates are no longer consistent whereas PC-IDIM-IV, IDIM-IV, and PC-DIDIM estimates remain consistent. Indeed, the relative errors are still below 10%, whereas they are higher than 20% for PC-IDIM-LS. Besides, the torque reconstruction illustrated in Figure D.3 with PC-IDIM-IV estimates, the second joint and the first trajectory shows that the fitting is excellent despite a noisier signal.

This result is consistent with those exposed in [Gautier et al., 2012, Janot et al., 2013b, Janot and Wensing, 2021]. When the noise is too high, the LS estimator provides inconsistent estimates because of the correlation between the errors. Adding the physical does not remove the persisting bias. On the contrary, IDIM-IV, PC-IDIM-IV, and PC-DIDIM are immune to this correlation because the fundamental relation $\underset{r\to\infty}{\text{plim}}\left(\boldsymbol{Z}_{st}^T\boldsymbol{\rho}\right) = \mathbf{0}$ holds.

Table D.5: Relative errors obtained for the PC-IDIM-IV, IDIMIV and PC-DIDIM methods - Modeling errors

Joint j	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	12.3%	13.3%	12.9%
2	11.2%	15.7%	13.5%
3	11.4%	13.1%	11.9%
4	11.9%	13.2%	12.7%
5	13.2%	15.6%	14.4%
6	14.1%	16.8%	15.0%

Table D.6: Relative errors obtained with cross-validation, the PC-IDIM-IV, IDIMIV and PC-DIDIM estimates - Modeling errors

Traj.	PC-IDIM-IV	IDIM-IV	PC-DIDIM
1	15.2%	18.1%	15.7%
2	14.8%	17.7%	16.0%
3	14.7%	18.2%	15.1%

D.3.3 Third scenario: modeling errors

Finally, we introduce modeling errors to evaluate the robustness of PC-IDIM-IV. Modeling errors are likely to occur in the control structure because industrials do not share it unless with formal agreements. So, the users have to identify it. In this study, we downgrade the gains of the simulated control so that the bandwidth of the first three joints is 5Hz.

With modeling errors, PC-IDIM-IV, IDIM-IV, and PC-DIDIM converge in 5 iterations. Besides, for an 8 seconds trajectory, one iteration of IDIM-IV needs 5 seconds to compute the estimates, and the Gauss-Newton algorithm needs 10 or 15 iterations to converge between 2 and 4 seconds. So, PC-IDIM-IV converges in less than 1 minute.

The relative errors gathered in Table D.5 and in Table D.6 show that the different methods perform well despite the modeling error. Only the DIDIM method fails to converge; this explains why we have not presented its estimates. However, IDIM-IV gives the higher relative errors. This result is interesting because it shows that both PC-IDIM-IV and PC-DIDIM methods are robust against modeling errors, provided they are not too significant to make the problem unfeasible. Since modeling errors are always possible, inserting the physical constraints improves the IDIM-IV method and should be considered in the future.

D.4 Conclusion

In this Appendix, the constrained instrumental variable method for industrial robots, PC-IDIM-IV, has been validated on the 6-*dof* industrial robot TX40 and compared with state-of-the-art methods. We have shown that this method is not time-consuming though it uses two

recursive algorithms: the outer one that is the usual IV method and the inner one to solve the physical constraints. Besides, compared to the standard IDIM-IV approach, PC-IDIM-IV is robust against modeling errors, provided the problem remains feasible.

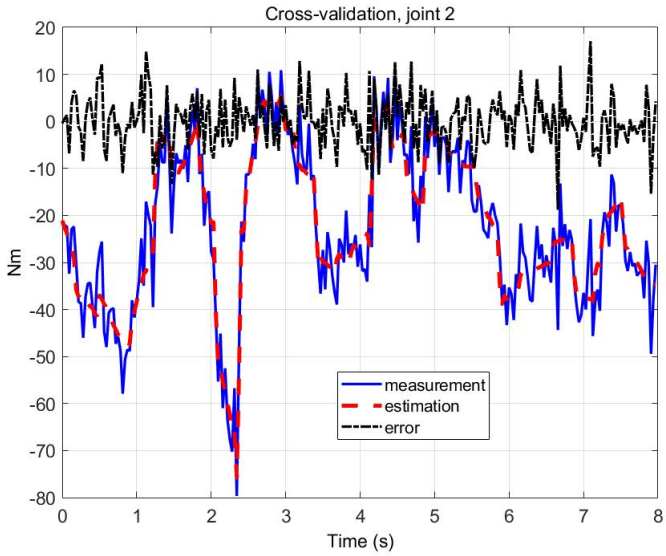


Figure D.3: Cross-validation, second joint and first trajectory, downgraded data.

Appendix E

Model and identification of an unbalanced two-wheeled differential drive mobile robot

*This Appendix is an extension of the work "**Comparison of least-squares and instrumental variables for parameters estimation on differential drive mobile robots**" in* [Ardiani et al., 2021]. *The nomenclature used in this Appendix is independent of the rest of the document.*

E.1 Introduction

Historically, control in mobile robotics has been based just on the kinematic model of the system [Siegwart et al., 2011]. This is due to the accuracy of the dynamic model depending on parameters that may change during execution or are difficult to measure. Indeed, this model is much more complex than the kinematic model and, normally, motors used in mobile robots have already an efficient low-level velocity control loop incorporated whose role is to decouple the kinematics from the dynamics [Morin and Samson, 2008]. However, as applications get more complex, as in mobile manipulators and when it involves high-speeds and heavy loads to be carried, the consideration of the dynamic model becomes essential [Martins et al., 2008, Martins and Brandão, 2018]. These models can be derived using basic laws and constitutive relationships, i.e. First-Principles Models, or from measured data, i.e. Empirical Models. The combination of these methods lies in the so-called grey-box model, where a parameter estimation (fitting of empirical data) of first-principles models is done [Tangirala, 2018].

For mobile robots, specially differential drive mobile robots (DDMR), the state-of-the-art on parameter estimation methods of its dynamic model is the well-known Least-Squares (LS) technique [Innocenti et al., 2004, Filipescu et al., 2007, Yoon and Gur, 2018, Alves et al., 2018]. LS has been proved to be the optimal estimator under the defined assumptions (see Appendix A). For instance, the error needs to be normally distributed with zero mean and constant variance (homoscedasticity), and there should not exist auto correlation between errors [Allen, 2004, Janot et al., 2014]. It may be difficult to meet these assumptions and they may not be applicable in mobile robots, due to several factors such as sliding, external forces and not-perfect rolling. Moreover, LS requires a well-tuned data filtering, and, even

with it, an optimal and unbiased estimate is not ensured.

On the other hand, the Instrumental Variable (IV) method is one of the methods that deals with a noisy observation matrix and is robust against the correlation issue [Brunot et al., 2015]. Hence, it imposes less restrictive assumptions on the error, since it uses the simulation of the direct dynamic model (DDM) to generate the instrumental matrix [Young, 1970]. This technique has been successfully applied in robotics and mechanical systems [Janot et al., 2009a, Brunot and Janot, 2018], but its potential use in estimation of dynamic parameters of mobile robots has not been addressed yet.

Moreover, dynamic modeling of DDMR has already been approached [Boyden and Velinsky, 1994, Zhang et al., 1998, Ivanjko et al., 2010, Dhaouadi and Hatab, 2013, Alves et al., 2018]. Most of these works consider a balanced mobile robot, where the center of mass is equally distanced to both driving wheels. However, the importance of considering a model in an unbalanced situation grows, as applications with a high and not centered load for transporting or mobile manipulators may lead to. This phenomena has been mentioned in [Innocenti et al., 2004, Albagul and Wahyudi, 2004], but there is no formal and complete definition of its dynamic model.

The aim of this work is twofold: first, to show the potential use of the IV method on parameter estimation of the dynamic model of mobile robots, and, secondly, to fill the mentioned gap in unbalanced mobile robot dynamic modeling towards a more complete and useful model. To do so, the derivation of the mentioned inverse dynamic model (IDM) and DDM is carried out by means of the Newton-Euler formulation. Effects as sliding, slipping, unbalanced load, castor wheels and tools attached to the mobile robot as a robotic arm, are considered with the objective of stating a complete model. Many of these effects are grouped in an uncertainty term, which is depreciated and considered as external perturbations to the system.

Therefore, this Appendix is organized as follows: Section E.2 describes the derivation of the kinematic model and direct and inverse dynamic models. Section E.3 briefly reviews the theory behind the two identification methods above mentioned and the trajectory selection. Section E.4 presents the simulation environment analysis and results, and Section E.5 some conclusion remarks.

E.2 Unbalanced differential drive mobile robot model

Figure E.1 shows a diagram of the robot, and Table E.1 summarizes the meaning of parameters and variables of its model. In this section, the kinematic and dynamic models are derived. The first is based on the work of [Zhang et al., 1998], but with more importance given to the non-centered center of mass. Then, the DDM is obtained, and afterwards simplified in order to show the IDM in the appropriate form for parameter estimation.

Table E.1: Symbols.

Symbol	Meaning	Value
B	Mid point between wheels.	-
G	Center of mass.	-
C	Point where resultant force of castor wheels is applied.	-
E	Point where resultant force of tools (e.g. robotic arm) is applied.	-
R	Wheel's radius.	100 mm
L	Distance between wheels.	430 mm
$[X', Y']$	Local coordinate frame fixed on B.	-
$[X'_G, Y'_G]$	Local coordinate frame fixed on G.	-
F_{ex}, F_{ey}, τ_e	Resultant tool forces in X' and Y' directions and moment exerted on E.	-
F_{cx}, F_{cy}, τ_c	Resultant tire forces in X' and Y' directions and moment exerted on C by castor wheels.	-
F_{wlx}, F_{wly}, F_{wrx}, F_{wry}	Resultant tire forces in X' and Y' directions of left and right wheels, respectively.	-
$[c_x, c_y]$; $[e_x, e_y]$	Positions of points of interest required for modelling (see Figure E.1).	-
$[b_x; b_y]$	Position of point G (see Figure E.1).	[19,19] mm
m	Robot's total mass (including wheels).	63.14 kg
I_z	Inertia around the center of mass.	2.97 kg m^2
I_e	Inertia of the wheel, motor's rotor and gearbox around its axis.	7853.98 kg mm^2
Wheel's thickness		50 mm
Rubber's density (wheels' material)		1000 $\frac{kg}{m^3}$
Coefficient of static friction		0.5
Coefficient of dynamic friction		0.5
Contact stiffness		25000 $\frac{N}{m}$
Contact damping		2500 $\frac{Ns}{m}$

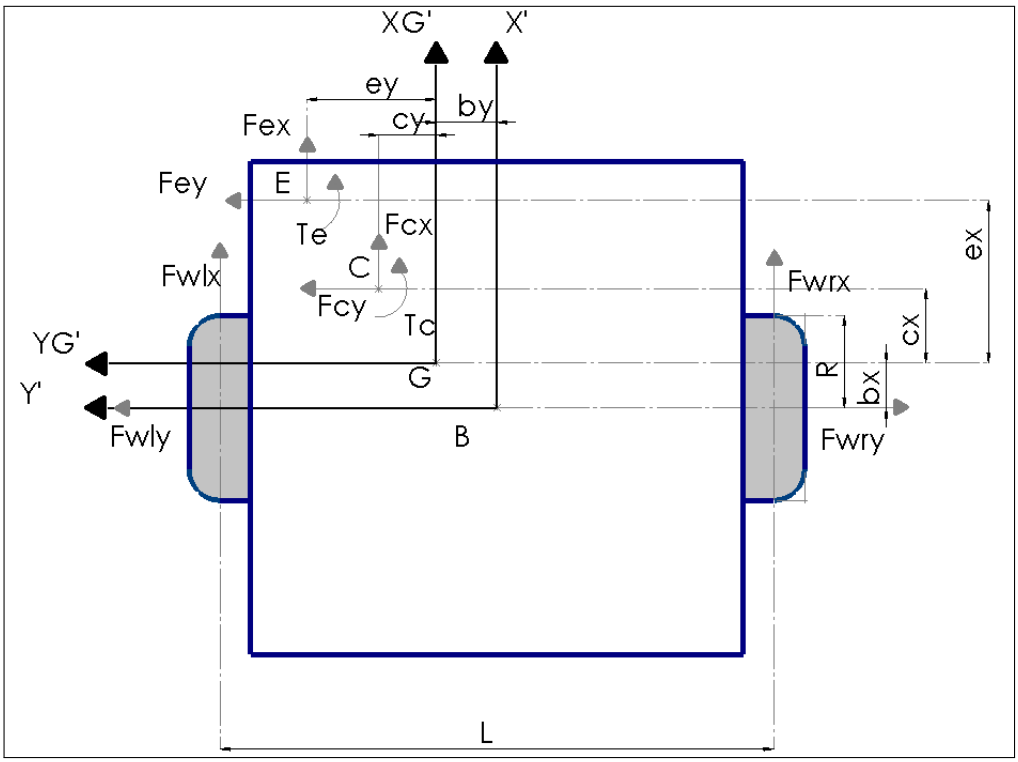


Figure E.1: Differential drive mobile robot.

E.2.1 Kinematic equations of motion

The velocity equations of the geometrical center (point B) are:

$$\begin{cases} v = \dfrac{1}{2}(v_r + v_l) \\ \omega = \dfrac{1}{L}(v_r - v_l) \\ u = u_s \end{cases}, \tag{E.1}$$

where v and u are the translational velocities in X' and Y' directions respectively; ω is the rotational speed of the robot; v_r and v_l are the longitudinal speed of the left and right wheel center, respectively; and u_s is the wheel center's lateral slip speed (lateral speed of both wheels are the same as they are rigidly connected).

Therefore, posture of point B in the world coordinate can be described by:

$$\begin{cases} \dot{x}_B = v\cos\theta - u\sin\theta \\ \dot{y}_B = v\sin\theta + u\cos\theta \\ \dot{\theta} = \omega \end{cases}. \tag{E.2}$$

Moreover, the longitudinal speeds can be described as:

$$v_l = R\dot{\phi}_l + v_{sl};\; v_r = R\dot{\phi}_r + v_{sr}, \tag{E.3}$$

where $\dot{\phi}_r$ and $\dot{\phi}_l$ are the rotational speed of the right and left wheel, respectively; and v_{sr} and v_{sl} are the longitudinal slip speeds of each wheel.

Taking into account this knowledge, the speeds of any point of the robot can be calculated. Specially, the equations regarding the center of mass (point G) are:

$$\begin{cases} v_G = v - b_y\,\omega = \dfrac{1}{2}(v_r + v_l) - b_y\,\omega \\ u_G = u + b_x\,\omega = u_s + b_x\,\omega \\ \omega = \dot{\theta} = \dfrac{1}{L}(v_r - v_l) \end{cases}. \tag{E.4}$$

The respective velocities in the global coordinate frame can be obtained in the same way as in Equation (E.2) but taking into account the velocities from Equation (E.4).

Replacing Equation (E.4) with the expressions from Equation (E.1) and Equation (E.3), it can be written as:

$$v_G = \frac{1}{2}\left[R(\dot{\phi}_r + \dot{\phi}_l) + (v_{sr} + v_{sl})\right] v - \frac{b_y}{L}\left[R(\dot{\phi}_r - \dot{\phi}_l) + (v_{sr} - v_{sl})\right], \tag{E.5}$$

$$\omega = \frac{1}{L}\left[R(\dot{\phi}_r - \dot{\phi}_l) + (v_{sr} - v_{sl})\right], \tag{E.6}$$

$$u_G = \frac{b_x}{L}\left[R(\dot{\phi}_r - \dot{\phi}_l) + (v_{sr} - v_{sl})\right] + u_s. \tag{E.7}$$

Moreover, as v_G and u_G are defined in the local rotating coordinate frame centered at point G defined by the unit vectors $\hat{\boldsymbol{i}}$, $\hat{\boldsymbol{j}}$ and $\hat{\boldsymbol{k}}$, and knowing that $\dfrac{d\hat{\boldsymbol{i}}}{dt} = \omega\hat{\boldsymbol{j}}$, $\dfrac{d\hat{\boldsymbol{j}}}{dt} = -\omega\hat{\boldsymbol{i}}$ and $\dfrac{d\hat{\boldsymbol{k}}}{dt} = 0$, the accelerations expressed in the vectorial way $\dot{\boldsymbol{v}}_{\boldsymbol{G}}$ and $\dot{\boldsymbol{u}}_{\boldsymbol{G}}$ are:

$$\begin{matrix} \dot{\boldsymbol{v}}_{\boldsymbol{G}} = (\dot{v}_G - u_G\omega)\hat{\boldsymbol{i}} \\ \dot{\boldsymbol{u}}_{\boldsymbol{G}} = (\dot{u}_G + v_G\omega)\hat{\boldsymbol{j}} \end{matrix}. \tag{E.8}$$

E.2.2 Dynamic model

Applying the Newton-Euler formulation to the diagram in Figure E.1, similar to the process done in [Boyden and Velinsky, 1994], the forces and moment equations with respect to point

G are:

$$\begin{gathered}\sum F_{xg'} = m\dot{v}_G = F_{wlx'} + F_{wrx'} + F_{ex'} + F_{cx'} \\ \sum F_{yg'} = m\dot{u}_G = F_{wlx'} + F_{wrx'} + F_{ex'} + F_{cx'} \\ \sum M_z = I_z\dot{\omega} = (\frac{L}{2} + b_y)F_{wrx'} - (\frac{L}{2} - b_y)F_{wlx'} - b_x(F_{wry'} + F_{wly'}) - \\ -e_yF_{ex} + e_xF_{ey} - c_yF_{cx} + c_xF_{cy} + \tau_e + \tau_c\end{gathered}, \tag{E.9}$$

where m is the total mass of the robot and I_z is the total inertia of the robot around the axis that goes through the point G and is perpendicular to the plane.

The second term of the first two equations of the system in Equation (E.9) can be expressed in terms of the accelerations and velocities of point B by using the relations from Equation (E.4) and Equation (E.8) (as an extension of the work presented in [Zhang et al., 1998]):

$$\begin{cases} m\,\dot{v}_G = m(\dot{v} - b_x\omega^2 - \omega u - b_y\dot{\omega}) \\ m\,\dot{u}_G = m(\dot{u} - b_y\omega^2 + \omega v + b_x\dot{\omega}) \end{cases}. \tag{E.10}$$

On the other hand, the dynamic equations of the wheel-rotor combinations are:

$$I_e\ddot{\phi}_i + B_e\dot{\phi}_i = \tau_i - F_{wix'}R; \quad i = r, l, \tag{E.11}$$

where i takes the value r for the right wheel and l for the left wheel; I_e is the inertia of the rotor, gearbox and wheel around the wheel axis; B_e is its viscous friction coefficient; and τ_i are the torques applied in each motor.

Inserting Equation (E.8) and Equation (E.11) in Equation (E.9), using Equation (E.3) and Equation (E.4) to express $\dot{\phi}_r + \dot{\phi}_l$ and $\dot{\phi}_r - \dot{\phi}_l$ and their respective derivatives as a function of v, ω, v_{rs} and v_{ls}, and rearranging the expressions, the direct dynamic model can be formulated in the following matricial form:

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} m + \dfrac{2I_e}{R^2} & -mb_y \\ \dfrac{2I_eb_y}{R^2} & I_z + b_x^2m + \dfrac{L^2I_e}{2R^2} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} -\dfrac{2B_e}{R^2} & 0 & 0 & mb_x \\ -\dfrac{2B_eb_y}{R^2} & -\dfrac{L^2B_e}{2R^2} & -mb_x & mb_xb_y \end{bmatrix} \begin{bmatrix} v \\ \omega \\ v\omega \\ \omega^2 \end{bmatrix} + \right.$$
$$\left. + \begin{bmatrix} \dfrac{1}{R} & \dfrac{1}{R} \\ \dfrac{1}{R}(\dfrac{L}{2} + b_y) & -\dfrac{1}{R}(\dfrac{L}{2} - b_y) \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} + \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} \right\}, \tag{E.12}$$

with

$$\begin{cases} \delta_1 = m\omega u_s + \dfrac{B_e}{R^2}(v_{sr} + v_{sl}) + \dfrac{I_e}{R^2}(\dot{v}_{sr} + \dot{v}_{sl}) + F_{ex} + F_{cx} \\ \delta_2 = \dfrac{LI_e}{2R^2}(\dot{v}_{sr} - \dot{v}_{sl}) + \dfrac{b_y I_e}{R^2}(\dot{v}_{sr} + \dot{v}_{sl}) + \dfrac{B_e L}{2R^2}(v_{sr} - v_{sl}) + \\ \quad + \dfrac{B_e b_y}{R^2}(v_{sr} + v_{sl}) + -b_x m\dot{u}_s - e_y F_{ex} + e_x F_{ey} - c_y F_{cx} + c_x F_{cy} + \tau_e + \tau_c \end{cases}.$$

If $b_y = 0$, the model becomes the same as in [Zhang et al., 1998, Boyden and Velinsky, 1994]. Also, notice that the equations shown refer to the translational and rotational speed of the geometrical center. The main reason of studying this point is because its coordinates will not change during motion, while the center of mass coordinates are a priori unknown, and may change dynamically during the robot operation. Furthermore, δ_1 and δ_2 are the uncertainties of the model. They gather terms which will be considered as unknown and/or as external perturbations to the system.

If it is assumed that there is an ideal wheel-ground contact (no-slip), no external disturbance forces nor moments, a negligible resistance force of the castor wheels and only the friction due to the motors rotation, then, the uncertainty term disappears, making the model much simpler to study. Therefore, the IDM can be expressed in the following form, suitable for parameter identification:

$$\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} = \begin{bmatrix} \dot{v} & \dot{\omega} & v & \omega & -\omega^2 & v\omega & 0 & 0 \\ 0 & 0 & v & -\omega & -\omega^2 & -v\omega & \dot{v} & -\dot{\omega} \end{bmatrix} \boldsymbol{\beta}, \tag{E.13}$$

where $\boldsymbol{\beta}$ is the [8x1] vector of a set of base parameters defined by:

$$\begin{cases} \beta_1 = \dfrac{2I_e L + LR^2 m - 2R^2 b_y m}{2LR}; \\ \beta_2 = \dfrac{2R^2 b_y^2 m - R^2 b_y m L + 2R^2 I_z + 2R^2 b_x^2 m + I_e L^2}{2LR}; \\ \beta_3 = \dfrac{B_e}{R}; \beta_4 = \dfrac{B_e L}{2R}; \beta_5 = \dfrac{Rb_x m}{2}; \beta_6 = \dfrac{Rb_x m}{L}; \\ \beta_7 = \dfrac{2I_e L + LR^2 m + 2R^2 b_y m}{2LR}; \\ \beta_8 = \dfrac{2R^2 b_y^2 m + R^2 b_y m L + 2R^2 I_z + 2R^2 b_x^2 m + I_e L^2}{2LR} \end{cases}. \tag{E.14}$$

The knowledge of this set of base parameters can determine the dynamic model uniquely, even though they may not have a direct physical meaning as they are [Gautier and Khalil, 1990].

E.3 Parameter identification

E.3.1 Methods

In a general form, the problem can be re-expressed as follows:

$$\boldsymbol{\tau} = \mathrm{IDM}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})\boldsymbol{\Theta}, \tag{E.15}$$

where $\boldsymbol{\xi}$ and $\dot{\boldsymbol{\xi}}$ are the vectors of velocities and accelerations respectively; $\mathrm{IDM}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$ is a matrix corresponding to the IDM and $\boldsymbol{\Theta}$ is the vector of base parameters.

After data acquisition, sampling and filtering if needed, Equation (E.15) can be rewritten as an over-determined linear system in the form of:

$$\boldsymbol{Y}(\boldsymbol{\tau}) = X(\hat{\boldsymbol{\xi}}, \hat{\dot{\boldsymbol{\xi}}})\boldsymbol{\Theta} + \boldsymbol{\rho}, \tag{E.16}$$

where $\boldsymbol{Y}(\boldsymbol{\tau})$ is the measurements vector built from actual torques $\boldsymbol{\tau}$; $\hat{\boldsymbol{\xi}}$ and $\hat{\dot{\boldsymbol{\xi}}}$ are the observation vectors (measured values) of the generalized velocities and accelerations respectively; $X(\hat{\boldsymbol{\xi}}, \hat{\dot{\boldsymbol{\xi}}})$ is called the observation matrix; and $\boldsymbol{\rho}$ is a vector of error terms due to external perturbations and simplifications done to the model, between other reasons [Janot et al., 2013b, Gautier and Briot, 2014].

Parameter identification consists in finding the set of parameters $\boldsymbol{\Theta}$ that best fits Equation (E.16) by reducing the error $\boldsymbol{\rho}$, which enables the simulation, prediction and control of the dynamic system. Two of the possible estimation methods are detailed hereafter.

E.3.1.1 LS method

The LS method is about estimating parameters by minimizing the squared discrepancies between observed data and their expected values.

If the regression model is assumed to be noise-free, ordinary least-squares can be used to deliver the estimates of the parameters as:

$$\hat{\boldsymbol{\Theta}}_{\boldsymbol{LS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}. \tag{E.17}$$

In order to consider the model as noise-free, taylor-made filters, decimation and elimination of border effects must be carried out accordingly [Briot and Gautier, 2015].

E.3.1.2 IV method

In the case that a good filtering stage cannot be ensured, an alternative is the IV method. It is robust to data filtering and leads to statistically optimal estimations. Generally, the

instrument set used in robotics is built from data calculated from simulation of the DDM. The simulation is based on previous estimates of the parameters and assumes the same reference trajectories and control structure for both actual and simulated robots [Janot et al., 2013b]. It takes the form of:

$$\hat{\boldsymbol{\Theta}}_{\boldsymbol{IV}} = (\boldsymbol{Z}^T\boldsymbol{X})^{-1}\boldsymbol{Z}^T\boldsymbol{Y}, \tag{E.18}$$

where $\boldsymbol{Z}$ is the instrumental variable matrix built in the same way as the observation matrix $\boldsymbol{X}$ but with simulated data.

E.3.2 Trajectory selection

Although the study of the tools used for selecting a trajectory that enables an adequate identification of parameters is not the objective of this work, it plays an important role in the identification process, and therefore, should be briefly mentioned.

In order to be able to select the trajectory that best excites the parameters, a cost function that describes the performance and allows comparison between them must be used. Most of bibliography uses one based on the condition number of the observation matrix [Khalil and Dombre, 2002].

In this work, in contrast to sequential identification, a global identification is carried out: one trajectory that excites all pertinent parameters is designed. First of all, a parametrization of the trajectory is needed. A finite sum of harmonic sine and cosine functions, i.e., a finite Fourier series, was chosen for each wheel [Swevers et al., 1997b]. The velocity and acceleration are then written as:

$$\begin{cases} \xi_i(t) = \sum_{l=1}^{N_i} a_{l,i}\,\cos\left(\omega_f l t\right) + b_{l,i}\,\sin\left(\omega_f l t\right) \\ \dot{\xi}_i(t) = \sum_{l=1}^{N_i} -a_{l,i}\,\omega_f l\,\sin\left(\omega_f l t\right) + b_{l,i}\,\omega_f l\,\cos\left(\omega_f l t\right) \end{cases}, \tag{E.19}$$

where ω_f is the fundamental pulsation of the Fourier series and ξ_i and $\dot{\xi}_i$ are the velocities (v and ω) and accelerations ($\dot{v}$ and $\dot{\omega}$) of the mobile robot.

The problem of selecting the trajectory becomes one of optimization with linear and non-linear constraints. The objective is to find parameters $a_{l,i}$ and $b_{l,i}$ of both input trajectories that lead to a global trajectory that minimizes the condition number of the regressor matrix. SQP (Sequential Quadratic Programming) algorithm was used to solve it. Constraints regarding positions (boundary limits of the robot workspace), maximum velocities, maximum accelerations, pose (in order to avoid the robot to start making circles and have a softer trajectory), and initial position were imposed. The trajectory obtained is, theoretically, a local minimum of the cost function, thus, to different initial values, different results may arise.

E.4 Simulation and results

For simulation, the model provided by the *Simscape Multibody Contact Forces Library* [Miller, 2020] called *Two Wheel Robot* was used. This model represents a quite real situation as it simulates effects that are difficult to model, such as the contact and friction between wheels and the floor.

Parameters used to run the simulation are shown in the third column of Table E.1. Information is based on technical data of the Pioneer LX robot (*Omron Adept Mobile Robots*). The wheels' mass and the robot's total inertia are estimated values from the available parameters, whereas the friction coefficients and contact values are set to a theoretical value.

The value of the parameters to be estimated are shown in Table E.3. β_3 and β_4 depend on the friction coefficient, which is not explicitly defined in the simulation, thus it is not possible to know which is the real value of these parameters.

With regard to the trajectory selection, a five terms Fourier series was used, with a fundamental pulsation of $\omega_f = 0.1$ during $t = 30s$ (see Equation (E.19)) as explained in Section E.3.2. Values are shown in Table E.2. The torque inputs needed to follow this trajectory are shown in Figure E.2 where the path obtained by the mobile robot is shown in Figure E.3. In order to make a cross-validation, two new trajectories were designed: a sum of two sin waves (see Figure E.4) and a sum of steps (see Figure E.5).

Table E.2: Fourier-based trajectory parameters.

Parameter	j	
	1	2
$a_{1,j}$	3.1231	-0.0729
$a_{2,j}$	0.3317	-1.5558
$a_{3,j}$	-0.1874	1.0222
$a_{4,j}$	-1.7562	1.2263
$a_{5,j}$	-1.5105	-0.6198
$b_{1,j}$	2.0287	1.2768
$b_{2,j}$	-2.0549	1.6834
$b_{3,j}$	-0.5311	2.2727
$b_{4,j}$	-1.7335	-1.6215
$b_{5,j}$	2.0589	-0.9885

Moreover, to make the simulation realistic, two types of noise have been added:

- white noise with a SNR (signal-to-noise ratio) of 5.
- steps as an external perturbation that represent phenomena as loosing contact of the wheel with the ground (sudden increase and decrease of wheels' rotational speed) or slipping. Their amplitude is around 25% of the peak amplitude of the measured value.

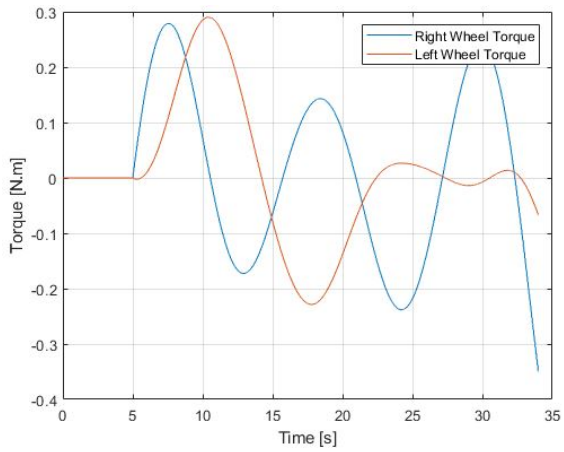


Figure E.2: Inputs of Fourier-based trajectory.

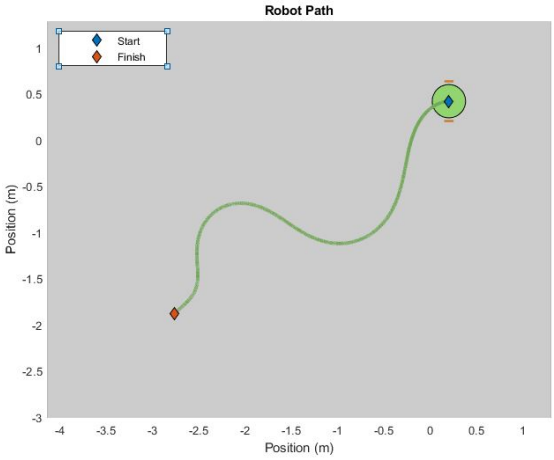


Figure E.3: Position followed by the robot when applying inputs of Figure E.2.

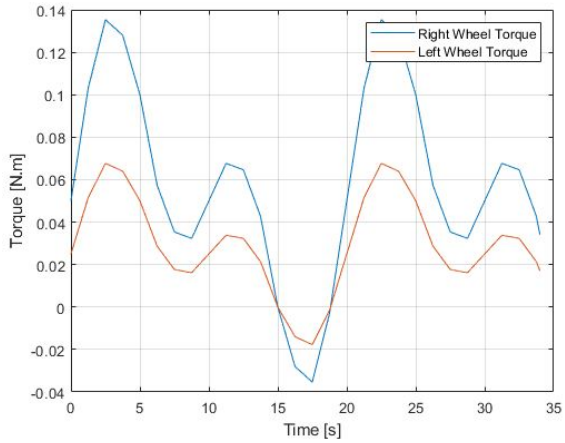


Figure E.4: Inputs obtained as a combination of *sins*.

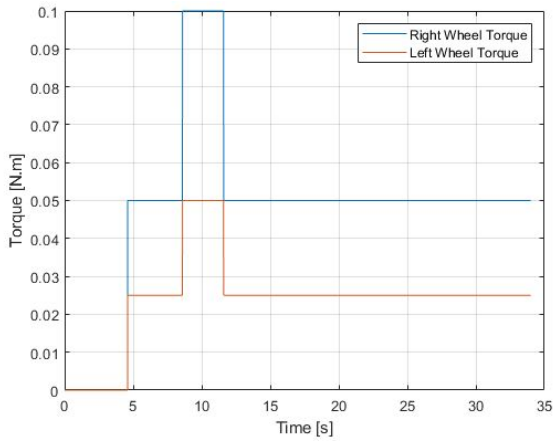


Figure E.5: Inputs obtained as a combination of *steps*.

A proper low-pass filter was designed to eliminate the white noise for LS Estimation, whereas no filter is needed for the IV method [Gautier et al., 2012].

Table E.3 and Table E.4 show performance indicators of both methods when just white noise is considered in the simulation. Both have good results, although LS gives slightly better results. Maximum relative error between real and estimated parameters is less than 10% for both methods, being less than 1.5% for LS and less than 6% for IV. Moreover, both methods success to predict the two testing trajectories with an accuracy of more than 90% (considering the fitness value equals to the one's complement of the NRMSE (Normalized Root-Mean-Square Error)). For prediction of cross-validation trajectories, LS performs better, with a fitness value of more than 99% for both trajectories, which means an almost perfect prediction.

Furthermore, Table E.4 also shows a comparison between the classical model (without considering an unbalanced situation) and the model presented in this work. Having a fitness value less than 65% is an indication that the model fails to predict the dynamic of the system. This is, partially, a way of validating the new model's equations.

Table E.3: Estimates' values.

Parameter	Real	LS		IV	
		Value	Relative error [%]	Value	Relative error [%]
β_1	2956.55	2959.41	0.1	3011.40	1.84
β_2	658596.2	658397.22	0.03	661786.31	0.5
β_3	-	0.98	-	1.40	-
β_4	-	167.43	-	5490.98	-
β_5	60000	59903.77	0.16	58416.37	1.77
β_6	279.07	276.14	1.05	296.47	5.57
β_7	3514.69	3508.01	0.19	3499.09	0.44
β_8	778596.2	776337.11	0.29	777250.63	0.18

Table E.4: Cross-validation: fitness value (one's complement of NRMSE) between real and estimated trajectories [in %].

Model	Classical		New			
Method	LS		LS		IV	
Trajectory	v	ω	v	ω	v	ω
sins	93.12	30.68	99.36	99.29	99.34	93.39
steps	94.31	58.02	99.53	99.54	99.38	97.68

On the other hand, Table E.5 shows the fitness value results for both methods and both cross-validation trajectories when the external perturbation noise is included in the simulation and no change in the taylor-made filter is made. Note that, in this case, IV method performs much better than LS Estimation. IV has a fitness value higher than 75% for both trajectories, whereas LS decreases to 32%. Performance is diminished because the steps are of a considerable amplitude, distorting significantly the signal, in a pessimistic scenario.

Table E.5: Cross-validation with external perturbation: fitness value (complement of NRMSE) between real and estimated trajectories [in %].

Method	LS		IV	
Trajectory	v	ω	v	ω
sins	51.75	32.21	82.7	75.34
steps	52.29	61.29	94.25	85.29

E.5 Conclusion

In this work, a dynamic parameter identification for DDMR was performed on simulation. First of all, a new mobile robot dynamic model with uncertainties and misaligned center of mass was derived, verified and compared with the classical model via parameter identification. It has been shown that the new model will succeed to express the behavior of a mobile robot when the center of mass is not aligned with neither of its axis, whereas the classical model will fail to do so.

Moreover, if a proper filter can be designed, it has been shown that although the IV method has good results, LS estimate is still better. However, in mobile robots, the filter design may be difficult to carry out, as there are many possible perturbations to the measurements that make impossible to guarantee a normal distribution of noise. Therefore, IV method presents an advantage and is a promising approach for this type of systems. Experimental validation is planned on future works.

Appendix F

Identification of multiple KUKA iiwa manipulators

The study done in this Appendix is motivated by the following questions:

- *How different are the identified parameters from two identical robots (same manufacturer and same version)?*
- *How acceptable is to consider the model identified with one of the manipulators to predict, simulate and/or control the other one?*
- *How do the controllers of the same version of manipulators behave against the same input? Do they have the same manipulator's model incorporated? Are they coded in the same way?*

We know that manufacturers want to make their products the more similar to each other as possible in order to reduce costs, to use the same computing configuration and algorithms, and to reduce risks. However, we usually do not know how much similar they are. Even if we consider that kinematic parameters, as the length of the links, and dynamic parameters, as the mass and inertia of the links, are exactly the same in each manipulator (the processes to generate each link are well-known and controlled), what happens with, for example, the friction parameters? There are a lot of factors that may influence this phenomenon that range from a light difference in alignment of parts (sensors, motors, links, wires, etc.) to how tighten are screws. The assembly process (alignment and placement of parts), the behavior against temperature changes from different components (e.g. motors) and the surface treatment of parts are just some of the many factors that could make one manipulator's behavior differ, substantially or not, from others'. These issues have already attracted the attention of some researchers as in [Corke and Armstrong-Helouvry, 1994]. There, the authors dealt with the problem of searching a consensus between the several models available in literature of the PUMA 560 robot.

In this Appendix, we will try to answer some of these questions. For this, we have identified the parameters of three KUKA iiwa manipulators (see Figure F.1) carrying out exactly the same trajectory with exactly the same estimation process and experimental setup. As seen in the figure, the base is the only thing changing from one robot to another. However, all of the manipulators are considered as fixed to the ground. For the experimental tests, we used

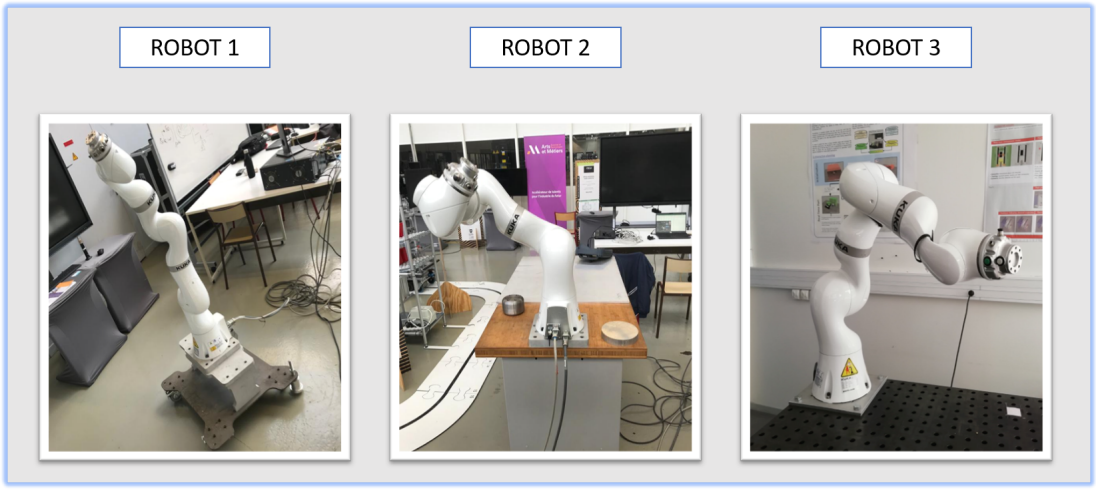


Figure F.1: Three KUKA iiwa manipulators.

the 3-parameters friction model from Equation (1.14), the filtering explained in Appendix B, and the offline IDIM-LS method explained in Section 1.4.2.

The Appendix is structured as follows:

- First, we analyze the *commanded position* and the *sensed torque* from the seven joints of the three manipulators to study its repeatability (see Figure B.2 for a description of the signals). Studying the *commanded position* will give a hint on the difference of the model that is integrated in the controller and/or the built-in trajectory generator. Moreover, the analysis of the *sensed torque* will give a priori information on how different the models are.
- Second, we apply the IDIM-LS method on the three robots carrying out the same four trajectories, and we show the mean value and standard deviation of some of the identified essential parameters to study their variability.
- Finally, we show the performance of using the set of parameters obtained with one of the manipulators to predict the torque of the others.

F.1 Signal analysis

Figure F.2 shows the *commanded position* and Figure F.3 the *sensed torque* for the seven joints of the three manipulators. It can be seen from these signals that, even tough the *commanded position* seems to be the same for all the manipulators, the measured torque may differ substantially, specially on joints 1, 5, 6, and 7. These joints are the ones which have less effect of gravity, thus noise and friction have more relative influence. These two figures depict the motivation of this study: even though the controllers seem to have the same model of the manipulator incorporated in them and the same trajectory generator (because of having

the same commanded signals), the real values of the parameters may differ from one robot to another (because of the difference in measured signals).

F.2 Estimated parameters

Moreover, Figure F.4 shows some of the estimated parameters with their respective standard deviation of the three manipulators over 4 different trajectories in a box-whisker plot. An "x" in this plot means that the identified parameter was not considered as an essential parameter, thus its variance was too big to plot. The friction parameters of joints 1, 2 and 4, and the main inertial parameters from joints 2 and 4 are shown.

As it is known that the identification process is trajectory dependant, it is expected that the result will vary from one trajectory to another even for the same robot. Trajectory 4 seems to be the one that excites the worst the parameters. Although the inertial parameters are all of the same order, there are some slight differences. Moreover, the friction parameters seem to vary more, specially for the models of joint 1 and 2, where they are substantially different. This leads to think that, even if we have a complete model of the robot given by the manufacturer or previous works carried out on the same version of manipulator, friction should be identified in each manipulator separately to complete the model. Next, we will validate this hypothesis by showing the percent error on the torque reconstruction using the different identified models.

F.3 Torque reconstruction

Table F.1 shows the percent error obtained by comparing the real *sensed torque* of three trajectories over the three manipulators and the reconstructed torque by using the different estimated models. The table can be read in the following way: the first row of the table is the percent error due to comparing the sensed torque of *robot 1* during *trajectory 1* with the reconstructed torque obtained with the identification of *robot 1* during *trajectory 1*; the second row, compares the sensed torque of *robot 2* during *trajectory 1* with the reconstructed torque obtained with the identification of *robot 1* during *trajectory 1*; and so on. The cells that are colored in green refer to those joints whose percent error is lower than than 10% and in red to those which is higher than 50%.

It can be seen that for joints where gravity effect is much higher than the friction effect, the percent error is low no matter which of the 3 models is used (joints 2, 3 and 4). However, it can be seen that for the other 4 joints, the models obtained with *robot 1* and *robot 3* fail to predict the torque of *robot 2* (and vice versa: the model obtained of *robot 2* seems to fail to predict the torque of *robot 1* and *robot 3*).

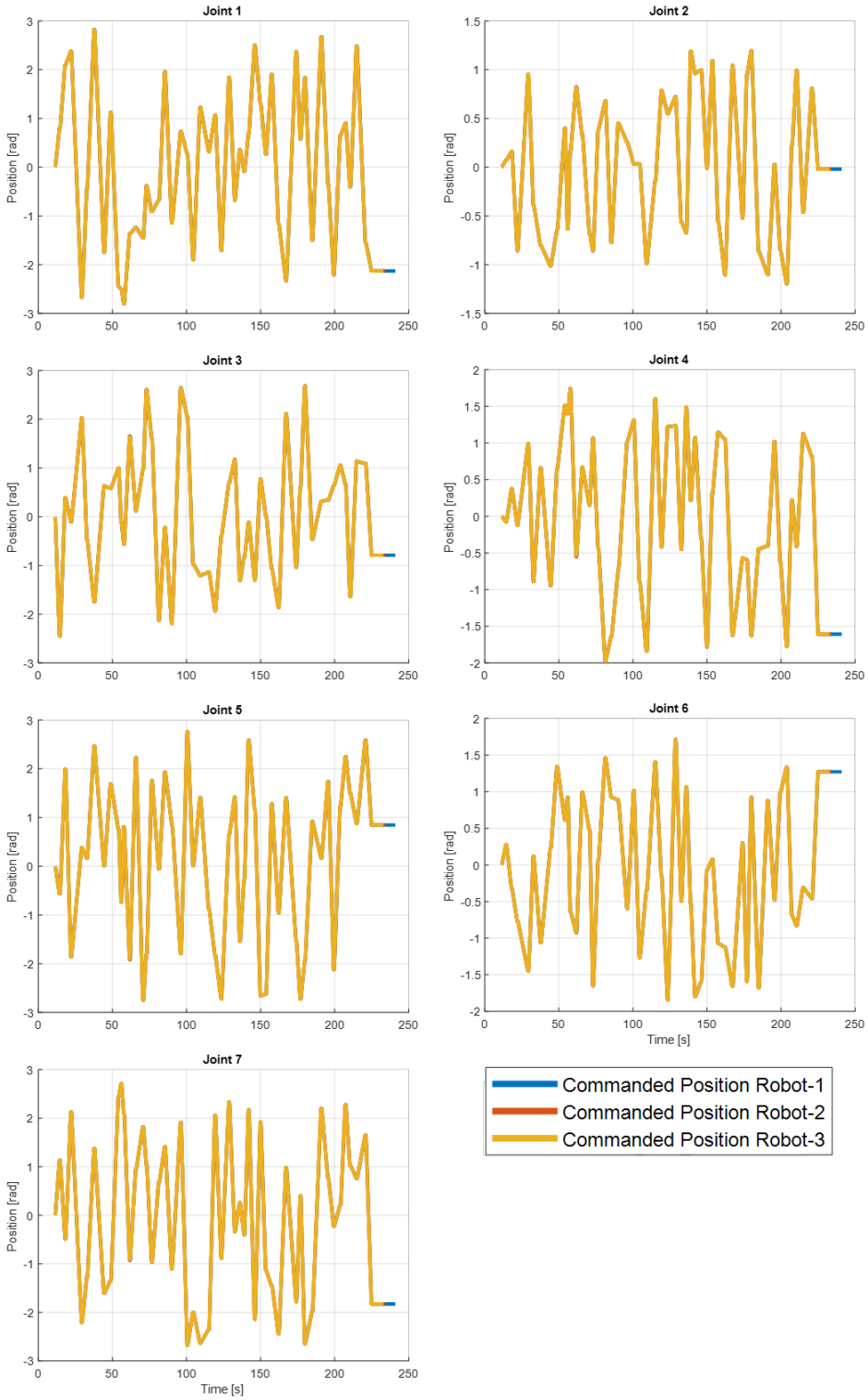


Figure F.2: *Commanded position* for the three KUKA iiwa manipulators.

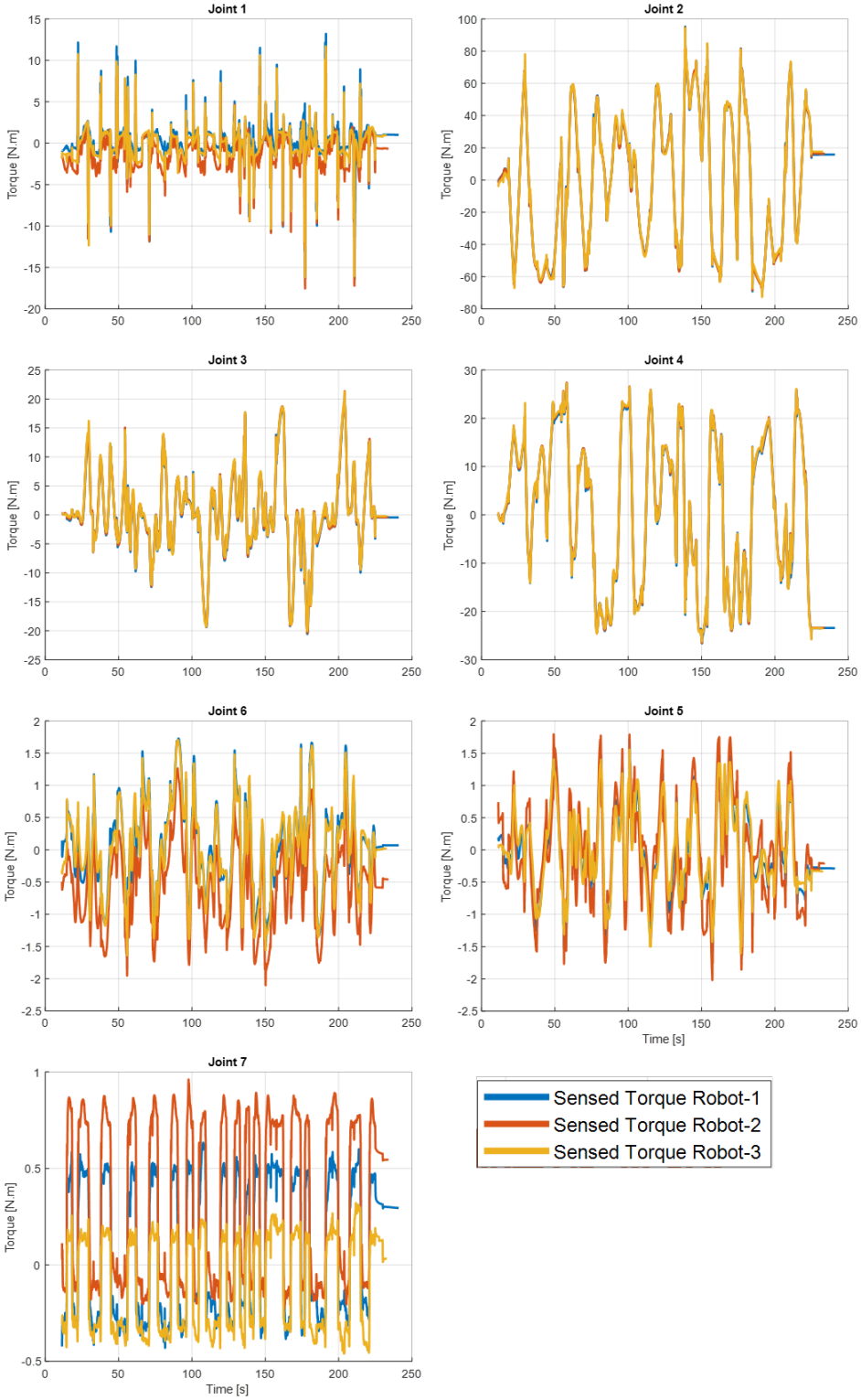


Figure F.3: *Sensed torque* for the three KUKA iiwa manipulators.

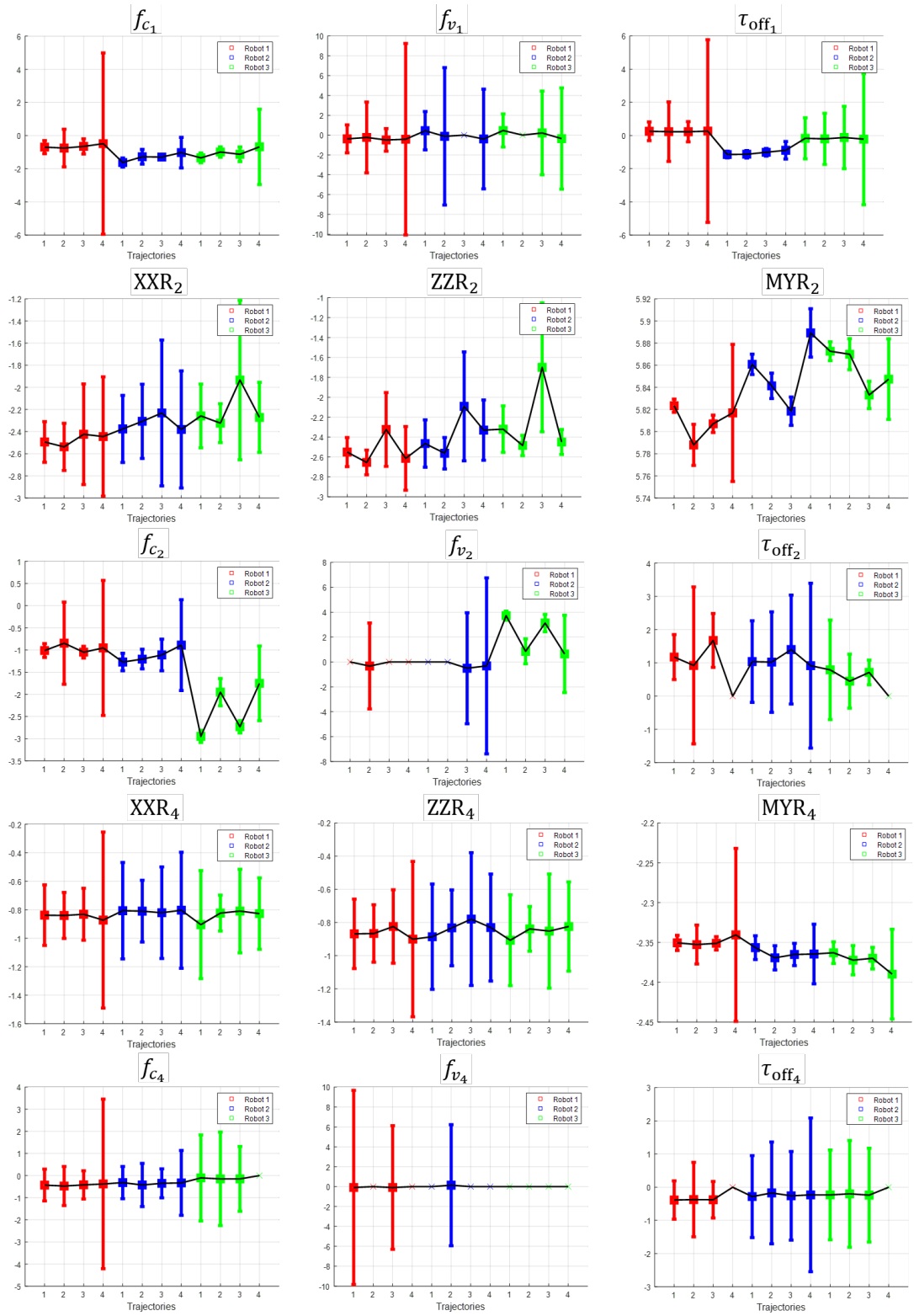


Figure F.4: Some of the identified parameters and its standard deviation for the three KUKA iiwa manipulators over four trajectories.

		Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7
Parameters from robot 1								
Trajectory 1	Robot 1	18.85	0.59	1.63	0.79	11.26	11.23	16.75
	Robot 2	67.58	1.33	3.33	1.83	44.74	75.45	46.65
	Robot 3	33.71	3.34	6.10	3.00	21.71	26.69	87.30
Trajectory 2	Robot 1	12.05	1.44	3.05	1.83	10.56	12.60	14.95
	Robot 2	49.43	1.38	3.33	1.71	39.53	77.47	45.10
	Robot 3	11.25	2.12	5.16	2.52	18.09	20.29	97.84
Trajectory 3	Robot 1	22.11	0.69	2.17	0.63	10.20	11.46	18.99
	Robot 2	71.39	1.44	4.23	1.57	44.70	71.68	48.22
	Robot 3	32.72	4.61	7.47	2.31	22.27	22.35	80.33
Parameters from robot 2								
Trajectory 1	Robot 1	90.64	1.11	2.85	1.85	52.81	90.79	65.74
	Robot 2	22.95	0.94	2.35	0.83	11.59	10.80	13.14
	Robot 3	55.81	3.17	4.37	1.87	72.81	98.55	178.02
Trajectory 2	Robot 1	27.93	1.59	4.03	2.51	47.83	80.23	68.10
	Robot 2	18.45	0.98	1.82	0.78	10.89	14.60	12.23
	Robot 3	19.25	1.85	3.68	1.52	65.46	81.44	202.30
Trajectory 3	Robot 1	90.72	1.15	3.95	1.58	51.54	79.40	70.04
	Robot 2	31.04	1.10	2.36	0.77	14.69	13.59	15.08
	Robot 3	67.01	4.47	4.78	1.49	72.22	78.59	169.14
Parameters from robot 3								
Trajectory 1	Robot 1	37.72	3.25	6.14	2.97	28.56	13.41	59.54
	Robot 2	48.14	3.12	4.52	1.76	65.39	75.44	84.93
	Robot 3	16.74	1.23	2.64	1.03	12.70	22.84	14.62
Trajectory 2	Robot 1	15.46	2.08	5.60	3.09	24.71	12.93	62.78
	Robot 2	36.79	1.64	3.42	1.61	57.86	75.67	83.13
	Robot 3	5.34	1.46	2.42	0.99	12.68	19.64	21.81
Trajectory 3	Robot 1	34.73	4.30	7.23	2.32	28.30	13.59	60.88
	Robot 2	56.78	4.23	4.60	1.50	65.62	68.26	85.66
	Robot 3	20.37	1.84	2.91	0.76	12.18	20.68	17.99

Table F.1: Percent error [%] in torque reconstruction for the three KUKA iiwa manipulators.

F.4 Conclusion

After this short study over 3 KUKA iiwa manipulators, we can conclude that the identified model from one version of the manipulator is not enough to ensure a good performance and prediction of the behavior of the other manipulators. Even though the inertial parameters seem not to change significantly, friction identification should be carried out in each manipulator separately to complete the model. Moreover, it seems that the controller is the same for all of the manipulators: the trajectory generator and the model that is inside the controller are not modified in each of the versions. More studies in the topic are planned in the future. For example, by adding a payload on the tip of the manipulators, the friction effects are going to be relatively smaller, and it would allow to study the inertial effects in a better way. It would also allow to analyze how the controller reacts against an unknown payload and to compare the built-in routine of payload identification.

Bibliography

[Afrough and Hanieh, 2019] Afrough, M. and Hanieh, A. A. (2019). Identification of dynamic parameters and friction coefficients. *Journal of Intelligent & Robotic Systems*, 94(1):3–13. 12

[Ajoudani et al., 2018a] Ajoudani, A., Fang, C., Tsagarakis, N., and Bicchi, A. (2018a). Reduced-complexity representation of the human arm active endpoint stiffness for supervisory control of remote manipulation. *The International Journal of Robotics Research*, 37(1):155–167. 101

[Ajoudani et al., 2018b] Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2018b). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975. 1, 145

[Albagul and Wahyudi, 2004] Albagul, A. and Wahyudi (2004). Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, 1(3):16. 168

[Albu-Schaffer et al., 2008] Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., Wolf, S., and Hirzinger, G. (2008). Soft robotics. *IEEE Robotics & Automation Magazine*, 15(3):20–30. 14

[Albu-Schäffer et al., 2007a] Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., and Hirzinger, G. (2007a). The dlr lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: an international journal*. 14, 27, 131, 145

[Albu-Schaffer and Hirzinger, 2001] Albu-Schaffer, A. and Hirzinger, G. (2001). Parameter identification and passivity based joint control for a 7 dof torque controlled light weight robot. In *Proceedings ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 3, pages 2852–2858. 42

[Albu-Schäffer et al., 2007b] Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007b). A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The international journal of robotics research*, 26(1):23–39. 14

[Aldrich, 1993] Aldrich, J. (1993). Reiersøl, geary and the idea of instrumental variables. *Economic and Social Review*, 24(3):247–273. 43

[Allen, 2004] Allen, M. P. (2004). *Understanding regression analysis*. Springer Science & Business Media. 167

[Alves et al., 2018] Alves, T. G., Lages, W. F., and Henriques, R. V. (2018). Parametric identification and controller design for a differential-drive mobile robot. *IFAC-PapersOnLine*, 51(15):437–442. 167, 168

[Anderson and Moore, 2012] Anderson, B. D. and Moore, J. B. (2012). *Optimal filtering*. Courier Corporation. 28, 153

[Ardiani et al., 2021] Ardiani, F., Benoussaad, M., and Janot, A. (2021). Comparison of least-squares and instrumental variables for parameters estimation on differential drive mobile robots. *IFAC-PapersOnLine*, 54(7):310–315. 43, 51, 125, 167

[Ardiani et al., 2022] Ardiani, F., Benoussaad, M., and Janot, A. (2022). Improving recursive dynamic parameter estimation of manipulators by knowing robot's model integrated in the controller. *IFAC-PapersOnLine*, 55(20):223–228. 80, 125

[Armstrong, 1988] Armstrong, B. (1988). Friction: Experimental determination, modeling and compensation. In *IEEE Proceedings. International Conference on Robotics and Automation*, pages 1422–1427. 12

[Armstrong, 1989] Armstrong, B. (1989). On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics. *The International journal of robotics research*, 8(6):28–48. 24, 25

[Armstrong et al., 1986] Armstrong, B., Khatib, O., and Burdick, J. (1986). The explicit dynamic model and inertial parameters of the puma 560 arm. In *IEEE Proceedings. International conference on robotics and automation*, volume 3, pages 510–518. 16

[Armstrong-Hélouvry et al., 1994] Armstrong-Hélouvry, B., Dupont, P., and De Wit, C. C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138. 11

[Artemiadis et al., 2010] Artemiadis, P. K., Katsiaris, P. T., Liarokapis, M. V., and Kyriakopoulos, K. J. (2010). Human arm impedance: Characterization and modeling in 3d space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3103–3108. 101

[Atkeson et al., 1985] Atkeson, C. G., An, C. H., and Hollerbach, J. M. (1985). Rigid body load identification for manipulators. In *24th IEEE Conference on Decision and Control*, pages 996–1002. 101

[Atkeson et al., 1986] Atkeson, C. G., An, C. H., and Hollerbach, J. M. (1986). Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119. 100

[Ayusawa and Nakamura, 2010] Ayusawa, K. and Nakamura, Y. (2010). Identification of standard inertial parameters for large-dof robots considering physical consistency. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6194–6201. 31

[Ayusawa et al., 2014] Ayusawa, K., Venture, G., and Nakamura, Y. (2014). Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems. *The International Journal of Robotics Research*, 33(3):446–468. 159

[Bahloul et al., 2018] Bahloul, A., Tliba, S., and Chitour, Y. (2018). Dynamic parameters identification of an industrial robot with and without payload. *Ifac-Papersonline*, 51(15):443–448. 41, 100

[Balestrino et al., 1984] Balestrino, A., De Maria, G., and Sciavicco, L. (1984). Robust control of robotic manipulators. *IFAC Proceedings Volumes*, 17(2):2435–2440. 98

[Barfoot, 2017] Barfoot, T. (2017). *State estimation for robotics*. Cambridge University Press. 98

[Bargsten et al., 2013] Bargsten, V., Zometa, P., and Findeisen, R. (2013). Modeling, parameter identification and model-based control of a lightweight robotic manipulator. In *IEEE International conference on control applications (CCA)*, pages 134–139. 42

[Benson and Vanderbei, 2003] Benson, H. Y. and Vanderbei, R. J. (2003). Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. *Mathematical Programming*, 95(2):279–302. 38

[Besset et al., 2016] Besset, P., Olabi, A., and Gibaru, O. (2016). Advanced calibration applied to a collaborative robot. In *IEEE International Power Electronics and Motion Control Conference (PEMC)*, pages 662–667. 42

[Bischoff et al., 2010] Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G., et al. (2010). The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *ISR 2010 (41st international symposium on robotics) and ROBOTIK 2010 (6th German conference on robotics)*, pages 1–8. VDE. 14

[Bittencourt et al., 2010] Bittencourt, A. C., Wernholt, E., Sander-Tavallaey, S., and Brogårdh, T. (2010). An extended friction model to capture load and temperature effects in robot joints. In *IEEE/RSJ international conference on intelligent robots and systems*, pages 6161–6167. 13

[Boeren et al., 2017] Boeren, F., Bruijnen, D., and Oomen, T. (2017). Enhancing feedforward controller tuning via instrumental variables: with application to nanopositioning. *International Journal of Control*, 90(4):746–764. 36, 75, 76, 77

[Boggs and Tolle, 1995] Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta numerica*, 4:1–51. 26

[Bona and Indri, 2005] Bona, B. and Indri, M. (2005). Friction compensation in robotics: an overview. In *IEEE Proceedings of the 44th Conference on Decision and Control*, pages 4360–4367. 11

[Bonnet et al., 2016] Bonnet, V., Fraisse, P., Crosnier, A., Gautier, M., González, A., and Venture, G. (2016). Optimal exciting dance for identifying inertial parameters of an anthropomorphic structure. *IEEE Transactions on Robotics*, 32(4):823–836. 26

[Boyden and Velinsky, 1994] Boyden, F. D. and Velinsky, S. A. (1994). Dynamic modeling of wheeled mobile robots for high load applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3071–3078. 168, 171, 173

[Briot and Gautier, 2015] Briot, S. and Gautier, M. (2015). Global identification of joint drive gains and dynamic parameters of parallel robots. *Multibody System Dynamics*, 33(1):3–26. 40, 174

[Bruls et al., 1999] Bruls, J., Chou, C. T., Haverkamp, B. R., and Verhaegen, M. (1999). Linear and non-linear system identification using separable least-squares. *European Journal of Control*, 5(1):116–128. 63

[Brunot and Janot, 2018] Brunot, M. and Janot, A. (2018). A new recursive instrumental variables approach for robot identification. *IFAC-PapersOnLine*, 51(15):132–137. 73, 75, 76, 77, 79, 168

[Brunot et al., 2020] Brunot, M., Janot, A., Carrillo, F., Cheong, J., and Noël, J.-P. (2020). Output error methods for robot identification. *Journal of Dynamic Systems, Measurement, and Control*, 142(3):031002. 41

[Brunot et al., 2017] Brunot, M., Janot, A., Carrillo, F., and Garnier, H. (2017). Comparison between the idim-iv method and the didim method for industrial robots identification. In *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 571–576. 41

[Brunot et al., 2015] Brunot, M., Janot, A., Carrillo, F., Garnier, H., Vandanjon, P.-O., and Gautier, M. (2015). Physical parameter identification of a one-degree-of-freedom electromechanical system operating in closed loop. *IFAC-PapersOnLine*, 48(28):823–828. 168

[Caccavale and Chiacchio, 1994] Caccavale, F. and Chiacchio, P. (1994). Identification of dynamic parameters and feedforward control for a conventional industrial manipulator. *Control Engineering Practice*, 2(6):1039–1050. 40

[Calafiore and Indri, 1998] Calafiore, G. and Indri, M. (1998). Experiment design for robot dynamic calibration. In *Proceedings. IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 4, pages 3303–3309. 26

[Calafiore et al., 2001] Calafiore, G., Indri, M., and Bona, B. (2001). Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation. *Journal of robotic systems*, 18(2):55–68. 24, 26

[Canudas et al., 1987] Canudas, C., Astrom, K., and Braun, K. (1987). Adaptive friction compensation in dc-motor drives. *IEEE Journal on Robotics and Automation*, 3(6):681–685. 12

[Canudas de Wit et al., 1991] Canudas de Wit, C., Noel, P., Aubin, A., and Brogliato, B. (1991). Adaptive friction compensation in robot manipulators: Low velocities. *The International journal of robotics research*, 10(3):189–199. 12

[Carlson et al., 2015] Carlson, F. B., Robertsson, A., and Johansson, R. (2015). Modeling and identification of position and temperature dependent friction phenomena without temperature sensing. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3045–3051. 13

[Chawda and Niemeyer, 2017a] Chawda, V. and Niemeyer, G. (2017a). Toward controlling a kuka lbr iiwa for interactive tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1808–1814. 42

[Chawda and Niemeyer, 2017b] Chawda, V. and Niemeyer, G. (2017b). Toward torque control of a kuka lbr iiwa for physical human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6387–6392. 42

[Chedmail et al., 1986] Chedmail, P., Gautier, M., and Khalil, W. (1986). Automatic dynamic modelling of robots including parameters of links and actuators. *IFAC Proceedings Volumes*, 19(14):83–87. 14

[Corke and Haviland, 2021] Corke, P. and Haviland, J. (2021). Not your grandmother's toolbox–the robotics toolbox reinvented for python. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11357–11363. 28

[Corke, 1996] Corke, P. I. (1996). A robotics toolbox for matlab. *IEEE Robotics & Automation Magazine*, 3(1):24–32. 28

[Corke and Armstrong-Helouvry, 1994] Corke, P. I. and Armstrong-Helouvry, B. (1994). A search for consensus among model parameters reported for the puma 560 robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1608–1613. 181

[Corke and Khatib, 2011] Corke, P. I. and Khatib, O. (2011). *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer. 28, 137

[Craig, 2005] Craig, J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall. 6, 10

[Craig et al., 1987] Craig, J. J., Hsu, P., and Sastry, S. S. (1987). Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2):16–28. 1, 72

[Curtelin et al., 1991] Curtelin, J.-J., Giordano, M., and Chevallier, E. (1991). Sequential identification of the dynamic parameters of a robot. In *Fifth IEEE International Conference on Advanced Robotics' Robots in Unstructured Environments*, pages 1513–1516. 74

[Dahl, 1968] Dahl, P. R. (1968). A solid friction model. Technical report, Aerospace Corp El Segundo Ca. 13

[Dasari et al., 2019] Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. (2019). Robonet: Large-scale multi-robot learning. *arXiv preprint: 1910.11215*. 147

[Davidson et al., 1993] Davidson, R., MacKinnon, J. G., et al. (1993). *Estimation and inference in econometrics*, volume 63. Oxford New York. 23, 45, 46

[De la Torre et al., 2009] De la Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., and Beltran, P. (2009). Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 147

[De Luca, 2000] De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. In *Proceedings IEEE/ICRA. Millennium Conference. International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 233–240. 15, 153

[De Luca et al., 2006] De Luca, A., Albu-Schaffer, A., Haddadin, S., and Hirzinger, G. (2006). Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630. 14, 101, 145, 154, 156

[De Luca and Flacco, 2012] De Luca, A. and Flacco, F. (2012). Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 288–295. 101

[De Luca et al., 2005] De Luca, A., Siciliano, B., and Zollo, L. (2005). Pd control with on-line gravity compensation for robots with elastic joints: Theory and experiments. *automatica*, 41(10):1809–1819. 14

[De Santis et al., 2008] De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. (2008). An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3):253–270. 145

[De Souza et al., 2021] De Souza, D. A., Batista, J. G., Vasconcelos, F. J., Dos Reis, L. L., Machado, G. F., Costa, J. R., Junior, J. N., Silva, J. L., Rios, C. S., and Júnior, A. B. (2021). Identification by recursive least squares with kalman filter (rls-kf) applied to a robotic manipulator. *IEEE Access*, 9:63779–63789. 73

[De Wit and Aubin, 1990] De Wit, C. C. and Aubin, A. (1990). Parameters identification of robots manipulators via sequential hybrid estimation algorithms. *IFAC Proceedings Volumes*, 23(8):163–168. xi, 74, 89, 93, 94

[De Wit et al., 1995] De Wit, C. C., Olsson, H., Astrom, K. J., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on automatic control*, 40(3):419–425. 13

[Denavit and Hartenberg, 1955] Denavit, J. and Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. 7

[Deremetz et al., 2021] Deremetz, M., Grunwald, G., Cavenago, F., Roa Garzon, M. A., De Stefano, M., Mishra, H., Reiner, M., Govindaraj, S., But, A., Sanz Nieto, I., et al.

(2021). Concept of operations and preliminary design of a modular multi-arm robot using standard interconnects for on-orbit large assembly. In *Proceedings of the International Astronautical Congress, IAC*. 124

[Dhaouadi and Hatab, 2013] Dhaouadi, R. and Hatab, A. A. (2013). Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics & Automation*, 2(2):1–7. 168

[Díaz-Rodríguez et al., 2010] 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. 30

[Ding, 2013] Ding, F. (2013). Coupled-least-squares identification for multivariable systems. *IET Control Theory & Applications*, 7(1):68–79. 89, 90, 91, 93

[Dong et al., 2018] Dong, Y., Ren, T., Chen, K., and Wu, D. (2018). An efficient robot payload identification method for industrial application. *Industrial Robot: An International Journal*, 45(4):505–515. 100

[Driels and Pathre, 1990] Driels, M. R. and Pathre, U. S. (1990). Significance of observation strategy on the design of robot calibration experiments. *Journal of robotic systems*, 7(2):197–223. 24

[Dumas et al., 2012] Dumas, C., Caro, S., Cherif, M., Garnier, S., and Furet, B. (2012). Joint stiffness identification of industrial serial robots. *Robotica*, 30(4):649–659. 42

[Dumora et al., 2012] Dumora, J., Geffard, F., Bidard, C., Brouillet, T., and Fraisse, P. (2012). Experimental study on haptic communication of a human in a shared human-robot collaborative task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5137–5144. 101

[Dumora et al., 2013] Dumora, J., Geffard, F., Bidard, C., and Fraisse, P. (2013). Towards a robotic partner for collaborative manipulation. In *HRI-Workshop on Collaborative Manipulation*, pages 1–6. 101

[Dutkiewicz et al., 1993] Dutkiewicz, P., Kozlowski, K., and Wroblewski, W. (1993). Experimental identification of robot and load dynamic parameters. In *Proceedings of IEEE International Conference on Control and Applications*, pages 767–776. 101

[Erden and Billard, 2014] Erden, M. S. and Billard, A. (2014). End-point impedance measurements at human hand during interactive manual welding with robot. In *International conference on robotics and automation (ICRA)*, pages 126–133. IEEE. 101

[Erden and Tomiyama, 2010] Erden, M. S. and Tomiyama, T. (2010). Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Transactions on Robotics*, 26(2):370–382. 101

[Farsoni et al., 2018] Farsoni, S., Landi, C. T., Ferraguti, F., Secchi, C., and Bonfè, M. (2018). Real-time identification of robot payload using a multirate quaternion-based kalman filter

and recursive total least-squares. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2103–2109. 98

[Filipescu et al., 2007] Filipescu, A., Stancu, A., and Filipescu, S. (2007). Sliding-mode adaptive control of pioneer 3-dx wheeled mobile robot. *The Annals of "Dunarea de Jos "University of Galati. Fascicle III, Electrotechnics, Electronics, Automatic Control, Informatics*, 30:71–76. 167

[Forssell and Ljung, 1999] Forssell, U. and Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35(7):1215–1241. 32

[Fortescue et al., 1981] Fortescue, T., Kershenbaum, L. S., and Ydstie, B. E. (1981). Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831–835. 104

[Fraccaroli et al., 2015] Fraccaroli, F., Peruffo, A., and Zorzi, M. (2015). A new recursive least squares method with multiple forgetting schemes. In *2015 54th IEEE conference on decision and control (CDC)*, pages 3367–3372. 104

[Gan et al., 2014] Gan, Y., Dai, X., and Dong, D. (2014). Robot calibration for cooperative process under typical installation. *Journal of Applied Mathematics*. 124

[Gao et al., 2017] Gao, L., Yuan, J., Han, Z., Wang, S., and Wang, N. (2017). A friction model with velocity, temperature and load torque effects for collaborative industrial robot joints. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3027–3032. 13

[Gao et al., 2014] Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Tao, L., Zappella, L., Béjar, B., Yuh, D. D., et al. (2014). Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3. 147

[Garnier et al., 2008] Garnier, H., Wang, L., and Young, P. C. (2008). Direct identification of continuous-time models from sampled data: Issues, basic solutions and relevance. In *Identification of continuous-time models from sampled data*, pages 1–29. Springer. 44

[Gautier, 1986] Gautier, M. (1986). Identification of robots dynamics. *IFAC Proceedings Volumes*, 19(14):125–130. 40

[Gautier, 1991] Gautier, M. (1991). Numerical calculation of the base inertial parameters of robots. *Journal of robotic systems*, 8(4):485–506. 20

[Gautier, 1992] Gautier, M. (1992). Optimal motion planning for robot's inertial parameters identification. In *Proceedings of the 31st Conference on Decision and Control*, pages 70–73. IEEE. 24

[Gautier, 1997] Gautier, M. (1997). Dynamic identification of robots with power model. In *Proceedings of IEEE international conference on robotics and automation*, volume 3, pages 1922–1927. 17, 40

[Gautier and Briot, 2012] Gautier, M. and Briot, S. (2012). Global identification of robot drive gains parameters using a known payload and weighted total least square techniques. *IFAC Proceedings Volumes*, 45(16):1389–1394. 40

[Gautier and Briot, 2013] Gautier, M. and Briot, S. (2013). Dynamic parameter identification of a 6 dof industrial robot using power model. In *IEEE International Conference on Robotics and Automation*, pages 2914–2920. 40

[Gautier and Briot, 2014] Gautier, M. and Briot, S. (2014). Global identification of joint drive gains and dynamic parameters of robots. *Journal of Dynamic Systems, Measurement, and Control*, 136(5):051025. 174

[Gautier et al., 2013a] Gautier, M., Briot, S., and Venture, G. (2013a). Identification of consistent standard dynamic parameters of industrial robots. In *ASME international conference on advanced intelligent mechatronics*, pages 1429–1435. IEEE. 22, 30, 48

[Gautier et al., 2012] Gautier, M., Janot, A., and Vandanjon, P.-O. (2012). A new closed-loop output error method for parameter identification of robot dynamics. *IEEE Transactions on Control Systems Technology*, 21(2):428–444. 41, 49, 76, 79, 80, 139, 159, 161, 163, 178

[Gautier et al., 2013b] Gautier, M., Jubien, A., Janot, A., and Robet, P.-P. (2013b). Dynamic identification of flexible joint manipulators with an efficient closed loop output error method based on motor torque output data. In *IEEE International Conference on Robotics and Automation*, pages 2949–2955. 41

[Gautier and Khalil, 1988] Gautier, M. and Khalil, W. (1988). On the identification of the inertial parameters of robots. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 3, pages 2264–2269. 17

[Gautier and Khalil, 1990] Gautier, M. and Khalil, W. (1990). Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on robotics and Automation*, 6(3):368–373. 18, 46, 173

[Gautier and Khalil, 1992] Gautier, M. and Khalil, W. (1992). Exciting trajectories for the identification of base inertial parameters of robots. *The International journal of robotics research*, 11(4):362–375. 24, 161

[Gautier and Poignet, 2001] Gautier, M. and Poignet, P. (2001). Extended kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12):1361–1372. 37, 40, 73, 80, 85

[Gautier and Presse, 1991] Gautier, M. and Presse, C. (1991). Sequential identification of base parameters of robots. In *Fifth IEEE International Conference on Advanced Robotics' Robots in Unstructured Environments*, pages 1105–1110. 74

[Gaz et al., 2019] Gaz, C., Cognetti, M., Oliva, A., Giordano, P. R., and De Luca, A. (2019). Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *IEEE Robotics and Automation Letters*, 4(4):4147–4154. 30, 42, 79, 146, 159

[Gaz and De Luca, 2017] Gaz, C. and De Luca, A. (2017). Payload estimation based on identified coefficients of robot dynamics—with an application to collision detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3033–3040. 100

[Gaz et al., 2014] Gaz, C., Flacco, F., and De Luca, A. (2014). Identifying the dynamic model used by the kuka lwr: A reverse engineering approach. In *IEEE International conference on robotics and automation (ICRA)*, pages 1386–1392. 42, 81

[Geiger et al., 2013] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237. 146

[Gilson et al., 2011] Gilson, M., Garnier, H., Young, P. C., and Van den Hof, P. M. (2011). Optimal instrumental variable method for closed-loop identification. *IET control theory & applications*, 5(10):1147–1154. 45, 78

[Golub and Pereyra, 2003] Golub, G. and Pereyra, V. (2003). Separable nonlinear least squares: the variable projection method and its applications. *Inverse problems*, 19(2):R1. 63

[Golub and Pereyra, 1973] Golub, G. H. and Pereyra, V. (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432. 42, 63

[Golub and Van Loan, 1980] Golub, G. H. and Van Loan, C. F. (1980). An analysis of the total least squares problem. *SIAM journal on numerical analysis*, 17(6):883–893. 40

[Golub and Van Loan, 2013] Golub, G. H. and Van Loan, C. F. (2013). *Matrix computations*. JHU press. 20

[Gomi and Kawato, 1997] Gomi, H. and Kawato, M. (1997). Human arm stiffness and equilibrium-point trajectory during multi-joint movement. *Biological cybernetics*, 76(3):163–171. 101

[Grant and Boyd, 2014] Grant, M. and Boyd, S. (2014). Cvx: Matlab software for disciplined convex programming, version 2.1. 48

[Grebenstein et al., 2011] Grebenstein, M., Albu-Schäffer, A., Bahls, T., Chalon, M., Eiberger, O., Friedl, W., Gruber, R., Haddadin, S., Hagn, U., Haslinger, R., et al. (2011). The dlr hand arm system. In *IEEE International Conference on Robotics and Automation*, pages 3175–3182. 14

[Gustavsson et al., 1977] Gustavsson, I., Ljung, L., and Söderström, T. (1977). Identification of processes in closed loop—identifiability and accuracy aspects. *Automatica*, 13(1):59–75. 32

[Haddadin et al., 2008] Haddadin, S., Albu-Schaffer, A., De Luca, A., and Hirzinger, G. (2008). Collision detection and reaction: A contribution to safe physical human-robot

interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3356–3363. 101

[Haddadin et al., 2009] Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. (2009). Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research*, 28(11-12):1507–1527. 14, 101, 145

[Haddadin et al., 2017] Haddadin, S., De Luca, A., and Albu-Schäffer, A. (2017). Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312. 101

[Hägglund, 1985] Hägglund, T. (1985). Recursive estimation of slowly time-varying parameters. *IFAC Proceedings Volumes*, 18(5):1137–1142. 104

[Hamad et al., 2019] Hamad, M., Mansfeld, N., Abdolshah, S., and Haddadin, S. (2019). The role of robot payload in the safety map framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 195–200. 100

[Hamon et al., 2011] Hamon, P., Gautier, M., and Garrec, P. (2011). New dry friction model with load-and velocity-dependence and dynamic identification of multi-dof robots. In *IEEE International Conference on Robotics and Automation*, pages 1077–1084. 13

[Hamon et al., 2010] Hamon, P., Gautier, M., Garrec, P., and Janot, A. (2010). Dynamic identification of robot with a load-dependent joint friction model. In *IEEE Conference on Robotics, Automation and Mechatronics*, pages 129–135. 12

[Han et al., 2020] Han, Y., Wu, J., Liu, C., and Xiong, Z. (2020). An iterative approach for accurate dynamic model identification of industrial robots. *Transactions on Robotics*, 36(5):1577–1594. 40

[Hao et al., 2021] Hao, L., Pagani, R., Beschi, M., and Legnani, G. (2021). Dynamic and friction parameters of an industrial robot: Identification, comparison and repetitiveness analysis. *Robotics*, 10(1):49. 64

[Hashemi and Werner, 2009] Hashemi, S. M. and Werner, H. (2009). Parameter identification of a robot arm using separable least squares technique. In *IEEE European Control Conference (ECC)*, pages 2199–2204. 63

[Hayashibara et al., 1999] Hayashibara, Y., Takubo, T., Sonoda, Y., Arai, H., and Tanie, K. (1999). Assist system for carrying a long object with a human-analysis of a human cooperative behavior in the vertical direction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289)*, volume 2, pages 695–700. 101

[Hennersperger et al., 2016] Hennersperger, C., Fuerst, B., Virga, S., Zettinig, O., Frisch, B., Neff, T., and Navab, N. (2016). Towards mri-based autonomous robotic us acquisitions: a first feasibility study. *Transactions on medical imaging*, 36(2):538–548. 42, 54, 55, 57, 58, 154

[Hirzinger et al., 2001] Hirzinger, G., Albu-Schaffer, A., Hahnle, M., Schaefer, I., and Sporer, N. (2001). On a new generation of torque controlled light-weight robots. In *Proceedings ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3356–3363. 14

[Hollerbach, 1980] Hollerbach, J. M. (1980). A recursive lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(11):730–736. 10

[Hsia, 1986] Hsia, T. (1986). Adaptive control of robot manipulators-a review. In *Proceedings. IEEE International Conference on Robotics and Automation*, volume 3, pages 183–189. 27, 72

[Hu et al., 2020] Hu, J., Li, C., Chen, Z., and Yao, B. (2020). Precision motion control of a 6-dofs industrial robot with accurate payload estimation. *ASME Transactions on Mechatronics*, 25(4):1821–1829. 99

[Huang and Ding, 2017] Huang, W. and Ding, F. (2017). Coupled least squares identification algorithms for multivariate output-error systems. *Algorithms*, 10(1):12. 91

[Huang et al., 2016] Huang, Y., Bianchi, M., Liarokapis, M., and Sun, Y. (2016). Recent data sets on object manipulation: A survey. *Big data*, 4(4):197–216. 147

[Indri et al., 2013] Indri, M., Lazzero, I., Antoniazza, A., and Bottero, A. M. (2013). Friction modeling and identification for industrial manipulators. In *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8. 12

[Indri and Trapani, 2020] Indri, M. and Trapani, S. (2020). Framework for static and dynamic friction identification for industrial manipulators. *ASME Transactions on Mechatronics*, 25(3):1589–1599. 12, 13, 41, 62, 64, 68

[Indri et al., 2016] Indri, M., Trapani, S., and Lazzero, I. (2016). Development of a general friction identification framework for industrial manipulators. In *IECON 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6859–6866. 62, 64, 65

[Innocenti et al., 2004] Innocenti, B., Ridao, P., Gascons, N., El-Fakdi, A., Lopez, B., and Salvi, J. (2004). Dynamical model parameters identification of a wheeled mobile robot. *IFAC Proceedings Volumes*, 37(8):263–268. 167, 168

[Ivanjko et al., 2010] Ivanjko, E., Petrinic, T., and Petrovic, I. (2010). Modelling of mobile robot dynamics. In *7th EUROSIM Congress on Modelling and Simulation*, volume 2. 168

[Janot, 2021] Janot, A. (2021). A separable instrumental variable method for robot identification. In *IEEE American Control Conference (ACC)*, pages 4339–4344. 69

[Janot et al., 2011] Janot, A., Gautier, M., Jubien, A., and Vandanjon, P.-O. (2011). Experimental joint stiffness identification depending on measurements availability. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 5112–5117. 42

[Janot et al., 2014] Janot, A., Gautier, M., Jubien, A., and Vandanjon, P. O. (2014). Comparison between the cloe method and the didim method for robots identification. *IEEE Transactions on Control Systems Technology*, 22(5):1935–1941. 41, 167

[Janot et al., 2009a] Janot, A., Vandanjon, P.-O., and Gautier, M. (2009a). Identification of robots dynamics with the instrumental variable method. In *IEEE International Conference on Robotics and Automation*, pages 1762–1767. 41, 168

[Janot et al., 2009b] Janot, A., Vandanjon, P.-O., and Gautier, M. (2009b). Using robust regressions and residual analysis to verify the reliability of ls estimation: Application in robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1962–1967. 40

[Janot et al., 2009c] Janot, A., Vandanjon, P.-O., and Gautier, M. (2009c). Using the instrumental variable method for robots identification. *IFAC Proceedings Volumes*, 42(10):480–485. 41

[Janot et al., 2012] Janot, A., Vandanjon, P.-O., and Gautier, M. (2012). Identification of physical parameters and instrumental variables validation with two-stage least squares estimator. *IEEE Transactions on Control Systems Technology*, 21(4):1386–1393. 44, 45

[Janot et al., 2013a] Janot, A., Vandanjon, P.-O., and Gautier, M. (2013a). A durbin-wu-hausman test for industrial robots identification. In *IEEE International Conference on Robotics and Automation*, pages 2956–2961. 36

[Janot et al., 2013b] Janot, A., Vandanjon, P.-O., and Gautier, M. (2013b). A generic instrumental variable approach for industrial robot identification. *IEEE Transactions on Control Systems Technology*, 22(1):132–145. 23, 41, 45, 46, 75, 78, 79, 159, 160, 161, 163, 174, 175

[Janot and Wensing, 2021] Janot, A. and Wensing, P. M. (2021). Sequential semidefinite optimization for physically and statistically consistent robot identification. *Control Engineering Practice*, 107:104699. 30, 38, 48, 80, 153, 160, 161, 163

[Jin and Gans, 2015] Jin, J. and Gans, N. (2015). Parameter identification for industrial robots with a fast and robust trajectory design approach. *Robotics and Computer-Integrated Manufacturing*, 31:21–29. 25

[Jovic et al., 2016] Jovic, J., Escande, A., Ayusawa, K., Yoshida, E., Kheddar, A., and Venture, G. (2016). Humanoid and human inertia parameter identification using hierarchical optimization. *IEEE Transactions on Robotics*, 32(3):726–735. 159

[Jubien et al., 2014a] Jubien, A., Gautier, M., and Janot, A. (2014a). Dynamic identification of the kuka lightweight robot: Comparison between actual and confidential kuka's parameters. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 483–488. 26, 42, 80, 146, 151

[Jubien et al., 2014b] Jubien, A., Gautier, M., and Janot, A. (2014b). Dynamic identification of the kuka lwr robot using motor torques and joint torque sensors data. *IFAC Proceedings Volumes*, 47(3):8391–8396. 42, 137

[Jung et al., 2018] Jung, D., Cheong, J., Park, D. I., and Park, C. (2018). Backward sequential approach for dynamic parameter identification of robot manipulators. *International Journal of Advanced Robotic Systems*, 15(1):1729881418758578. 74

[Kane and Levinson, 1983] Kane, T. R. and Levinson, D. A. (1983). The use of kane's dynamical equations in robotics. *The International Journal of Robotics Research*, 2(3):3–21. 10

[Karniadakis et al., 2021] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440. 123

[Kawasaki and Nishimura, 1988] Kawasaki, H. and Nishimura, K. (1988). Terminal-link parameter estimation of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):485–490. 100

[Kermani et al., 2007] Kermani, M. R., Patel, R. V., and Moallem, M. (2007). Friction identification and compensation in robotic manipulators. *IEEE Transactions on Instrumentation and Measurement*, 56(6):2346–2353. 62

[Khalil and Bennis, 1994] Khalil, W. and Bennis, F. (1994). Comments on" direct calculation of minimum set of inertial parameters of serial robots". *IEEE transactions on robotics and automation*, 10(1):78–79. 18

[Khalil et al., 2000] Khalil, W., Besnard, S., and Lemoine, P. (2000). Comparison study of the geometric parameters calibration methods. *International Journal of Robotics and Automation*, 15(2):pp–56. 18

[Khalil and Creusot, 1997] Khalil, W. and Creusot, D. (1997). Symoro+: a system for the symbolic modelling of robots. *Robotica*, 15(2):153–161. 78

[Khalil and Dombre, 2002] Khalil, W. and Dombre, E. (2002). *Modeling identification and control of robots*. CRC Press. 1, 6, 8, 10, 11, 15, 16, 18, 22, 24, 27, 74, 76, 79, 144, 145, 175

[Khalil et al., 2007] Khalil, W., Gautier, M., and Lemoine, P. (2007). Identification of the payload inertial parameters of industrial manipulators. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 4943–4948. 100

[Khalil and Kleinfinger, 1986] Khalil, W. and Kleinfinger, J. (1986). A new geometric notation for open and closed-loop robots. In *Proceedings. IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179. 7

[Kim et al., 2020] Kim, W., Balatti, P., Lamon, E., and Ajoudani, A. (2020). Moca-man: A mobile and reconfigurable collaborative robot assistant for conjoined human-robot actions. In *International IEEE Conference on Robotics and Automation (ICRA)*, pages 10191–10197. 124

[Kovincic et al., 2019] Kovincic, N., Müller, A., Gattringer, H., Weyrer, M., Schlotzhauer, A., and Brandstötter, M. (2019). Dynamic parameter identification of the universal robots ur5. In *Proceedings of the Austrian Robotics Workshop*. 42

[Kozlowski and Dutkiewicz, 1996] Kozlowski, K. R. and Dutkiewicz, P. (1996). Experimental identification of robot and load dynamics. *IFAC Proceedings Volumes*, 29(1):397–402. 100

[Kubus et al., 2007] Kubus, D., Kroger, T., and Wahl, F. M. (2007). On-line rigid object recognition and pose estimation based on inertial parameters. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1402–1408. 73, 98

[Kubus et al., 2008a] Kubus, D., Kroger, T., and Wahl, F. M. (2008a). Improving force control performance by computational elimination of non-contact forces/torques. In *IEEE International Conference on Robotics and Automation*, pages 2617–2622. 73, 98

[Kubus et al., 2008b] Kubus, D., Kroger, T., and Wahl, F. M. (2008b). On-line estimation of inertial parameters using a recursive total least-squares approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3845–3852. 73, 98, 101

[kuka, 2015a] kuka (2015a). *KUKA Robots Media Flange for Product Family LBR iiwa*. Pub Option Medien-Flansch, KUKA Augsburg, Germany. 131

[kuka, 2015b] kuka (2015b). *KUKA Sunrise.Connectivity FRI 1.7*. Pub Sunrise.Connectivity FRI, KUKA Augsburg, Germany. 131

[kuka, 2017] kuka (2017). *KUKA Sunrise OS 1.13*. Pub Sunrise OS, KUKA Augsburg, Germany. 80, 131

[kuka, 2019] kuka (2019). *KUKA Robots LBR iiwa 7 R800, LBR iiwa 14 R820*. Pub Spez LBR, KUKA Augsburg, Germany. 131

[Kurdas et al., 2022] Kurdas, A., Hamad, M., Vorndamme, J., Mansfeld, N., Abdolshah, S., and Haddadin, S. (2022). Online payload identification for tactile robots using the momentum observer. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5953–5959. 98

[Lambert, 2018] Lambert, B. (2018). *A student's guide to Bayesian statistics*. Sage. 127

[Landau et al., 2011] Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. (2011). *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media. 72

[Lawitzky et al., 2010] Lawitzky, M., Mörtl, A., and Hirche, S. (2010). Load sharing in human-robot cooperative manipulation. In *19th IEEEInternational Symposium in Robot and Human Interactive Communication*, pages 185–191. 101

[Leboutet et al., 2021] Leboutet, Q., Roux, J., Janot, A., Guadarrama-Olvera, J. R., and Cheng, G. (2021). Inertial parameter identification in robotics: A survey. *Applied Sciences*, 11(9):4303. 16, 146, 159

[Levine et al., 2018] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436. 147

[Lightcap and Banks, 2009] Lightcap, C. A. and Banks, S. A. (2009). An extended kalman filter for real-time estimation and control of a rigid-link flexible-joint manipulator. *IEEE Transactions on Control Systems Technology*, 18(1):91–103. 73

[Liu et al., 2016] Liu, Z., Huang, P., and Lu, Z. (2016). Recursive differential evolution algorithm for inertia parameter identification of space manipulator. *International Journal of Advanced Robotic Systems*, 13(3):104. 72

[Ljung, 1998] Ljung, L. (1998). *System Identification: Theory for the User*. Pearson Education. 16, 22, 23, 27, 33, 44, 48, 52, 72, 78, 146, 153

[Ljung and Söderström, 1983] Ljung, L. and Söderström, T. (1983). *Theory and practice of recursive identification*. MIT press. 72

[Ljung et al., 2004] Ljung, L., Zhang, Q., Lindskog, P., and Juditski, A. (2004). Estimation of grey box and black box models for non-linear circuit data. *IFAC Proceedings Volumes*, 37(13):399–404. 123

[Luh et al., 1980] Luh, J. Y., Walker, M. W., and Paul, R. P. (1980). On-line computational scheme for mechanical manipulators. 10

[Mandery et al., 2015] Mandery, C., Terlemez, Ö., Do, M., Vahrenkamp, N., and Asfour, T. (2015). The kit whole-body human motion database. In *IEEE International Conference on Advanced Robotics (ICAR)*, pages 329–336. 147

[Markovsky and Van Huffel, 2007] Markovsky, I. and Van Huffel, S. (2007). Overview of total least-squares methods. *Signal processing*, 87(10):2283–2302. 40

[Martins and Brandão, 2018] Martins, F. N. and Brandão, A. S. (2018). Motion control and velocity-based dynamic compensation for mobile robots. *Applications of mobile robots*. 167

[Martins et al., 2008] Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., and Bastos-Filho, T. F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, 16(11):1354–1363. 167

[Mata et al., 2005] Mata, V., Benimeli, F., Farhat, N., and Valera, A. (2005). Dynamic parameter identification in industrial robots considering physical feasibility. *Advanced Robotics*, 19(1):101–119. 30, 41

[Maurice et al., 2019] Maurice, P., Malaisé, A., Amiot, C., Paris, N., Richard, G.-J., Rochel, O., and Ivaldi, S. (2019). Human movement and ergonomics: An industry-oriented dataset for collaborative robotics. *The International Journal of Robotics Research*, 38(14):1529–1537. 147

[Mavrakis and Stolkin, 2020] Mavrakis, N. and Stolkin, R. (2020). Estimation and exploitation of objects' inertial parameters in robotic grasping and manipulation: A survey. *Robotics and Autonomous Systems*, 124:103374. 99

[Mayeda et al., 1984] Mayeda, H., Osuka, K., and Kangawa, A. (1984). A new identification method for serial manipulator arms. *IFAC Proceedings Volumes*, 17(2):2429–2434. 33, 94

[Middletone and Goodwin, 1986] Middletone, R. and Goodwin, G. C. (1986). Adaptive computed torque control for rigid link manipulators. In *25th IEEE Conference on Decision and Control*, pages 68–73. 72

[Miller, 2020] Miller, S. (2020). Simscape multibody contact forces library. *MATLAB Central File Exchange*. 176

[Mooring et al., 1991] Mooring, B. W., Roth, Z. S., and Driels, M. R. (1991). *Fundamentals of manipulator calibration*. Wiley-interscience. 18

[Morin and Samson, 2008] Morin, P. and Samson, C. (2008). Motion control of wheeled mobile robots. *Springer handbook of robotics*, 1:799–826. 167

[Mujica et al., 2023] Mujica, M., Crespo, M., Benoussaad, M., Junco, S., and Fourquet, J.-Y. (2023). Robust variable admittance control for human–robot co-manipulation of objects with unknown load. *Robotics and Computer-Integrated Manufacturing*, 79:102408. 42, 101, 112, 115, 154

[Neuman and Khosla, 1986] Neuman, C. P. and Khosla, P. K. (1986). Identification of robot dynamics: An application of recursive estimation. In *Adaptive and Learning Systems*, pages 175–194. Springer. 72

[Nevmerzhitskiy et al., 2019] Nevmerzhitskiy, M. N., Notkin, B. S., Vara, A. V., and Zmeu, K. V. (2019). Friction model of industrial robot joint with temperature correction by example of kuka kr10. *Journal of Robotics*, 2019. 13

[Ni et al., 2019] Ni, H., Zhang, C., Hu, T., Wang, T., Chen, Q., and Chen, C. (2019). A dynamic parameter identification method of industrial robots considering joint elasticity. *International Journal of Advanced Robotic Systems*. 42

[Nicodemus et al., 2022] Nicodemus, J., Kneifl, J., Fehr, J., and Unger, B. (2022). Physics-informed neural networks-based model predictive control for multi-link manipulators. *IFAC-PapersOnLine*, 55(20):331–336. 123

[Olsen and Petersen, 2001] Olsen, M. M. and Petersen, H. G. (2001). A new method for estimating parameters of a dynamic robot model. *IEEE transactions on robotics and automation*, 17(1):95–100. 40

[Olsen et al., 2002] Olsen, M. M., Swevers, J., and Verdonck, W. (2002). Maximum likelihood identification of a dynamic robot model: Implementation issues. *The international Journal of robotics research*, 21(2):89–96. 40

[Olsson et al., 1998] Olsson, H., Åström, K. J., De Wit, C. C., Gäfvert, M., and Lischinsky, P. (1998). Friction models and friction compensation. *Eur. J. Control*, 4(3):176–195. 12

[Oppenheim et al., 1997] Oppenheim, A. V., Willsky, A. S., Nawab, S. H., Hernández, G. M., et al. (1997). *Signals & systems*. Pearson Educación. 28

[Orin et al., 1979] Orin, D. E., McGhee, R., Vukobratović, M., and Hartoch, G. (1979). Kinematic and kinetic analysis of open-chain linkages utilizing newton-euler methods. *Mathematical Biosciences*, 43(1-2):107–130. 10

[Östring, 2002] Östring, M. (2002). Identification, diagnosis, and control of a flexible robot arm. 1, 34

[Östring and Gunnarsson, 2004] Östring, M. and Gunnarsson, S. (2004). Recursive identification of physical parameters in a flexible robot arm. *Asian Journal of Control*, 6(3):407–414. 73

[Park, 2006] Park, K.-J. (2006). Fourier-based optimal excitation trajectories for the dynamic identification of robots. *Robotica*, 24(5):625–633. 24, 26

[Parkum et al., 1990] Parkum, J., Poulsen, N. K., and Holst, J. (1990). Selective forgetting in adaptive procedures. *IFAC Proceedings Volumes*, 23(8):137–142. 104

[Pennestrì et al., 2016] Pennestrì, E., Rossi, V., Salvini, P., and Valentini, P. P. (2016). Review and comparison of dry friction force models. *Nonlinear dynamics*, 83(4):1785–1801. 11

[Pham and Gautier, 1991] Pham, C. and Gautier, M. (1991). Essential parameters of robots. In *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 2769–2774. 18, 22, 23

[Pham et al., 2001] Pham, M. T., Gautier, M., and Poignet, P. (2001). Identification of joint stiffness with bandpass filtering. In *Proceedings ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 3, pages 2867–2872. 14

[Pieper, 1969] Pieper, D. L. (1969). *The kinematics of manipulators under computer control*. Stanford University. 9

[Pirsiavash and Ramanan, 2012] Pirsiavash, H. and Ramanan, D. (2012). Detecting activities of daily living in first-person camera views. In *IEEE conference on computer vision and pattern recognition*, pages 2847–2854. 147

[Presse and Gautier, 1993] Presse, C. and Gautier, M. (1993). New criteria of exciting trajectories for robot identification. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 907–912. 24

[Pukelsheim, 2006] Pukelsheim, F. (2006). *Optimal design of experiments*. SIAM. 23, 149

[Qin et al., 2010] Qin, Z., Baron, L., and Birglen, L. (2010). A new approach to the dynamic parameter identification of robotic manipulators. *Robotica*, 28(4):539–547. 74

[Rackl et al., 2012] Rackl, W., Lampariello, R., and Hirzinger, G. (2012). Robot excitation trajectories for dynamic parameter estimation using optimized b-splines. In *IEEE International conference on robotics and automation*, pages 2042–2047. 26

[Raissi et al., 2019] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707. 123

[Rao Sripada and Grant Fisher, 1987] Rao Sripada, N. and Grant Fisher, D. (1987). Improved least squares identification. *International Journal of Control*, 46(6):1889–1913. 104

[Raviola et al., 2021] Raviola, A., Guida, R., De Martin, A., Pastorelli, S., Mauro, S., and Sorli, M. (2021). Effects of temperature and mounting configuration on the dynamic parameters identification of industrial robots. *Robotics*, 10(3):83. 13, 42, 146, 153

[Reiersøl, 1941] Reiersøl, O. (1941). Confluence analysis by means of lag moments and other methods of confluence analysis. *Econometrica: Journal of the Econometric Society*, pages 1–24. 43

[Reiersøl, 1945] Reiersøl, O. (1945). *Confluence analysis by means of instrumental sets of variables*. PhD thesis, Almqvist & Wiksell. 43

[Roggen et al., 2010] Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., et al. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In *Seventh IEEE international conference on networked sensing systems (INSS)*, pages 233–240. 147

[Ruderman, 2012] Ruderman, M. (2012). Modeling of elastic robot joints with nonlinear damping and hysteresis. *Robot Syst Control Program*, pages 293–312. 15

[Sælid et al., 1985] Sælid, S., Egeland, O., and Foss, B. A. (1985). A solution to the blow-up problem in adaptive controllers. 104

[Saelid and Foss, 1983] Saelid, S. and Foss, B. (1983). Adaptive controllers with a vector variable forgetting factor. In *The 22nd IEEE Conference on Decision and Control*, pages 1488–1494. 104

[Salgado et al., 1988] Salgado, M. E., Goodwin, G. C., and Middleton, R. H. (1988). Modified least squares algorithm incorporating exponential resetting and forgetting. *International Journal of Control*, 47(2):477–491. 104

[Santolaria and GinéS, 2013] Santolaria, J. and GinéS, M. (2013). Uncertainty estimation in robot kinematic calibration. *Robotics and Computer-Integrated Manufacturing*, 29(2):370–384. 18

[Schreiber et al., 2010] Schreiber, G., Stemmer, A., and Bischoff, R. (2010). The fast research interface for the kuka lightweight robot. In *IEEE workshop on innovative robot control architectures for demanding (Research) applications how to modify and enhance commercial controllers (ICRA)*, pages 15–21. 105, 135

[Sciavicco et al., 1994] Sciavicco, L., Siciliano, B., and Villani, L. (1994). On dynamic modelling of gear-driven rigid robot manipulators. *IFAC Proceedings Volumes*, 27(14):543–549. 14

[Sellami and Respall, 2020] Sellami, S. and Respall, V. M. (2020). Geometric and stiffness modeling and design of calibration experiments for the 7 dof serial manipulator kuka iiwa 14 r820. *arXiv preprint arXiv:2006.06314*. 42

[Semini et al., 2015] Semini, C., Barasuol, V., Boaventura, T., Frigerio, M., Focchi, M., Caldwell, D. G., and Buchli, J. (2015). Towards versatile legged robots through active impedance control. *The International Journal of Robotics Research*, 34(7):1003–1020. 159

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423. 27

[Sheth and Uicker Jr, 1971] Sheth, P. N. and Uicker Jr, J. (1971). A generalized symbolic notation for mechanisms. 7

[Siciliano et al., 2008] Siciliano, B., Khatib, O., and Kröger, T. (2008). *Springer handbook of robotics*, volume 200. Springer. 1

[Siciliano et al., 2010] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: modelling, planning and control*. Springer Science & Business Media. 6, 18, 26, 76, 79, 145

[Sidhom et al., 2010] Sidhom, L., Pham, M. T., Thévenoux, F., and Gautier, M. (2010). Identification of a robot manipulator based on an adaptive higher order sliding modes differentiator. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1093–1098. 72

[Siegwart et al., 2011] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press. 167

[Simoni et al., 2015] Simoni, L., Beschi, M., Legnani, G., and Visioli, A. (2015). Friction modeling with temperature effects for industrial robot manipulators. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3524–3529. 13

[Simoni et al., 2019] Simoni, L., Beschi, M., Legnani, G., and Visioli, A. (2019). Modelling the temperature in joint friction of industrial manipulators. *Robotica*, 37(5):906–927. 13

[Slotine and Li, 1987] Slotine, J.-J. E. and Li, W. (1987). On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3):49–59. 98

[Söderström and Stoica, 1989] Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice Hall. 16

[Sousa and Cortesao, 2014] Sousa, C. D. and Cortesao, R. (2014). Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach. *The International Journal of Robotics Research*, 33(6):931–944. 31, 41, 80, 153, 159

[Sousa and Cortesao, 2019] Sousa, C. D. and Cortesao, R. (2019). Inertia tensor properties in robot dynamics identification: A linear matrix inequality approach. *ASME Transactions on Mechatronics*, 24(1):406–411. 30, 80, 159

[Sousa, 2015] Sousa, C. J. S. D. (2015). *Dynamic model identification of robot manipulators: Solving the physical feasibility problem*. PhD thesis. 30

[Spong, 1987] Spong, M. W. (1987). Modeling and control of elastic joint robots. 14, 15

[Sturm et al., 2012] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. 146

[Stürz et al., 2017] Stürz, Y. R., Affolter, L. M., and Smith, R. S. (2017). Parameter identification of the kuka lbr iiwa robot including constraints on physical feasibility. *IFAC-PapersOnLine*, 50(1):6863–6868. 30, 42, 54, 55, 57, 58, 59, 61, 146, 154, 159

[Swevers et al., 2000] Swevers, J., Al-Bender, F., Ganseman, C. G., and Projogo, T. (2000). An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on automatic control*, 45(4):675–686. 13

[Swevers et al., 1997a] Swevers, J., Ganseman, C., De Schutter, J., and Van Brussel, H. (1997a). Generation of periodic trajectories for optimal robot excitation. 25

[Swevers et al., 1997b] Swevers, J., Ganseman, C., Tukel, D. B., De Schutter, J., and Van Brussel, H. (1997b). Optimal robot excitation and identification. *IEEE transactions on robotics and automation*, 13(5):730–740. 25, 40, 80, 151, 163, 175

[Swevers et al., 2007] Swevers, J., Verdonck, W., and De Schutter, J. (2007). Dynamic model identification for industrial robots. *IEEE Control systems magazine*, 27(5):58–71. 23, 40

[Swevers et al., 2002] Swevers, J., Verdonck, W., Naumer, B., Pieters, S., and Biber, E. (2002). An experimental robot load identification method for industrial application. *The International Journal of Robotics Research*, 21(8):701–712. 100

[Tadese et al., 2021] Tadese, M. A., Yumbla, F., Yi, J.-S., Lee, W., Park, J., and Moon, H. (2021). Passivity guaranteed dynamic friction model with temperature and load correction: Modeling and compensation for collaborative industrial robot. *IEEE Access*, 9:71210–71221. 13

[Taghbalout et al., 2019] Taghbalout, M., Antoine, J. F., and Abba, G. (2019). Experimental dynamic identification of a yumi collaborative robot. *IFAC-PapersOnLine*, 52(13):1168–1173. 42, 146

[Takagi et al., 2017] Takagi, A., Ganesh, G., Yoshioka, T., Kawato, M., and Burdet, E. (2017). Physically interacting individuals estimate the partner's goal to enhance their movements. *Nature Human Behaviour*, 1(3):1–6. 102

[Tangirala, 2018] Tangirala, A. K. (2018). *Principles of system identification: theory and practice*. Crc Press. 16, 22, 32, 167

[Tomei, 1991] Tomei, P. (1991). A simple pd controller for robots with elastic joints. *Transactions on automatic control*, 36(10):1208–1213. 15

[Uicker, 1965] Uicker, J. J. (1965). *On the dynamic analysis of spatial linkages using 4 x 4 matrices*. Northwestern University. 10

[Urrea and Pascal, 2017] Urrea, C. and Pascal, J. (2017). Parameter identification methods for real redundant manipulators. *Journal of applied research and technology*, 15(4):320–331. 73

[Urrea and Pascal, 2018] Urrea, C. and Pascal, J. (2018). Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot. *Computers & electrical engineering*, 67:791–806. 73

[Vahidi et al., 2005] Vahidi, A., Stefanopoulou, A., and Peng, H. (2005). Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, 43(1):31–55. 99, 103, 104

[Van den Hof, 1997] Van den Hof, P. (1997). Closed-loop issues in system identification. *IFAC Proceedings Volumes*, 30(11):1547–1560. 32

[Van Geffen, 2009] Van Geffen, V. (2009). A study of friction models and friction compensation. 11, 153

[Vandanjon et al., 1995] Vandanjon, P.-O., Gautier, M., and Desbats, P. (1995). Identification of robots inertial parameters by means of spectrum analysis. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 3033–3038. 24, 25, 33, 94

[Vandanjon et al., 2007] Vandanjon, P.-O., Janot, A., Gautier, M., and Khatounian, F. (2007). Comparison of two identification techniques: Theory and application. In *ICINCO-RA (1)*, pages 341–350. 41

[Vantilt et al., 2015] Vantilt, J., Aertbeliën, E., De Groote, F., and De Schutter, J. (2015). Optimal excitation and identification of the dynamic model of robotic systems with compliant actuators. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2117–2124. 13, 24, 26

[Villagrossi et al., 2014] Villagrossi, E., Legnani, G., Pedrocchi, N., Vicentini, F., Tosatti, L. M., Abbà, F., and Bottero, A. (2014). Robot dynamic model identification through excitation trajectories minimizing the correlation influence among essential parameters. In *11th IEEE International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 2, pages 475–482. 24, 26

[Villagrossi et al., 2013] Villagrossi, E., Pedrocchi, N., Vicentini, F., and Tosatti, L. M. (2013). Optimal robot dynamics local identification using genetic-based path planning in workspace subregions. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 932–937. 26

[Villagrossi et al., 2018] Villagrossi, E., Simoni, L., Beschi, M., Pedrocchi, N., Marini, A., Tosatti, L. M., and Visioli, A. (2018). A virtual force sensor for interaction tasks with conventional industrial robots. *Mechatronics*, 50:78–86. 159

[Vuong and Ang Jr, 2009] Vuong, N. D. and Ang Jr, M. H. (2009). Dynamic model identification for industrial robots. *Acta Polytechnica Hungarica*, 6(5):51–68. 41

[Wensing et al., 2017a] Wensing, P. M., Kim, S., and Slotine, J.-J. E. (2017a). Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *Robotics and Automation Letters*, 3(1):60–67. 30, 159

[Wensing et al., 2017b] Wensing, P. M., Wang, A., Seok, S., Otten, D., Lang, J., and Kim, S. (2017b). Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on robotics*, 33(3):509–522. 159

[Wernholt and Gunnarsson, 2006] Wernholt, E. and Gunnarsson, S. (2006). Nonlinear identification of a physically parameterized robot model 1. *IFAC Proceedings Volumes*, 39(1):143–148. 63

[Wooldridge, 2015] Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*. Cengage learning. 45

[Wright et al., 1999] Wright, S., Nocedal, J., et al. (1999). Numerical optimization. *Springer Science*, 35(67-68):7. 47

[Wu et al., 2010] 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. 16

[Wu et al., 2012] Wu, W., Zhu, S., Wang, X., and Liu, H. (2012). Closed-loop dynamic parameter identification of robot manipulators using modified fourier series. *International Journal of Advanced Robotic Systems*, 9(1):29. 26

[Xu et al., 2020] Xu, T., Fan, J., Chen, Y., Ng, X., Ang, M. H., Fang, Q., Zhu, Y., and Zhao, J. (2020). Dynamic identification of the kuka lbr iiwa robot with retrieval of physical parameters using global optimization. *IEEE Access*, 8:108018–108031. 42, 54, 58, 59, 61, 154

[Yoon and Gur, 2018] Yoon, Y. and Gur, B. (2018). Parameter estimation and control of nonholonomic mobile robots: A model-based approach. In *6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–6. 167

[Yoshida et al., 1992] Yoshida, K., Ikeda, N., and Mayeda, H. (1992). Experimental study of the identification methods for an industrial robot manipulator. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, volume 1, pages 263–270. 74

[Yoshida and Khalil, 2000] Yoshida, K. and Khalil, W. (2000). Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters. *The International Journal of Robotics Research*, 19(5):498–510. 30

[Yoshida et al., 1996] Yoshida, K., Osuka, K., Mayeda, H., and Ono, T. (1996). When is the set of base-parameter values physically impossible? *Journal of the Robotics Society of Japan*, 14(1):122–130. 30

[Yoshikawa et al., 2001] Yoshikawa, T., Ohta, A., and Kanaoka, K. (2001). State estimation and parameter identification of flexible manipulators based on visual sensor and virtual joint model. In *Proceedings IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 3, pages 2840–2845. 124

[Yoshitani and Hasegawa, 1998] Yoshitani, N. and Hasegawa, A. (1998). Model-based control of strip temperature for the heating furnace in continuous annealing. *IEEE transactions on control systems technology*, 6(2):146–156. 104

[Young, 1985] Young, P. (1985). The instrumental variable method: a practical approach to identification and system parameter estimation. *IFAC Proceedings Volumes*, 18(5):1–15. 41, 44

[Young, 1970] Young, P. C. (1970). An instrumental variable method for real-time identification of a noisy process. *Automatica*, 6(2):271–287. 168

[Young, 2011] Young, P. C. (2011). *Recursive estimation and time-series analysis: An introduction for the student and practitioner*. Springer Science & Business Media. 34, 37, 44, 45, 72, 73, 75, 77, 78, 79, 99, 103, 159

[Zhang et al., 1998] Zhang, Y., Hong, D., Chung, J. H., and Velinsky, S. A. (1998). Dynamic model based robust tracking control of a differentially steered wheeled mobile robot. In *Proceedings of the IEEE American control conference. ACC (Cat. No. 98CH36207)*, volume 2, pages 850–855. 168, 172, 173

Résumé — La robotique a joué un rôle important dans les changements industriels de ces dernières décennies, permettant de réduire les coûts, d'améliorer la qualité, d'augmenter la productivité et de réduire les dangers pour les employés. Bien que les robots aient d'abord été conçus pour travailler dans des scénarios industriels à l'intérieur de cages, on a eu tendance ces dernières années à supprimer ces barrières et à permettre aux robots non seulement de partager leur espace de travail avec les humains, mais aussi leurs tâches et leurs objectifs. Cela a élargi leur champ d'application, avec pour résultat une apparition de plus en plus fréquente des robots dans notre vie quotidienne et la nécessité de mieux maîtriser le robot. Qu'il s'agisse de concevoir une loi de commande, de simuler le système pour prédire les états futurs ou d'effectuer la détection de défauts ou de collisions, un modèle précis et fiable devient nécessaire. La meilleure façon d'y parvenir est d'utiliser des techniques de modélisation de type « gray-box » : elles font appel à la connaissance des lois physiques qui régissent le système et aux données expérimentales. La complexité croissante des systèmes et de leur environnement, ainsi que la nécessité de réadapter le modèle pendant le fonctionnement du système, remettent ces techniques sur le devant de la scène.

Dans le cadre de cette thèse, l'accent est mis sur le développement et l'amélioration des méthodes d'estimation des paramètres à appliquer dans l'identification du modèle dynamique des systèmes robotiques. Le cas de la robotique collaborative est étudié en validant toutes les méthodes avec le KUKA LBR iiwa 14 R820. Cependant, nos méthodes sont générales et applicables à une grande variété de robots, comme en témoignent les résultats obtenus avec le robot mobile Pioneer LX et le manipulateur industriel Stäubli TX40.

Les contributions sont présentées selon deux axes principaux. D'une part, on traite des méthodes d'estimation en-bloc hors ligne, qui sont généralement utilisées pour avoir une première idée du modèle. Considérant que les paramètres ont une signification physique et que les mesures sont intrinsèquement bruitées, le premier algorithme proposé est robuste au bruit de mesure et assure la consistance physique des estimations. La deuxième méthode proposée s'attaque à deux autres obstacles : le fait que les fabricants de robots commerciaux souvent ne divulguent pas des informations et des mesures importantes aux utilisateurs pour des raisons de confidentialité et de sécurité, et le phénomène important mais peu maîtrisé de la friction. La méthode identifie le modèle que le fabricant a inclus dans son contrôleur dans un processus de rétro-ingénierie, et l'utilise pour estimer les paramètres de friction.

D'autre part, dans les systèmes qui changent au cours du temps ou qui interagissent avec un environnement inconnu et dynamique, l'estimation du modèle et de ses paramètres doit se faire en ligne. À cette fin, les variantes récursives doivent être abordées. Des aspects tels que l'initialisation de l'algorithme, la stabilité des estimations et le temps de calcul prennent de l'importance. Dans ce contexte, nous développons d'abord une nouvelle méthode qui produit des estimations cohérentes avec des mesures bruitées, qui est robuste par rapport aux valeurs initiales et qui ne nécessite pas de simulation externe du système. Deuxièmement, bien qu'il soit connu que l'identification globale est plus précise que l'identification séquentielle, cette dernière est parfois la seule option. Nous développons ainsi de nouvelles méthodes qui propagent la distribution statistique des estimations dans les différentes étapes d'identification, ce qui permet de ne pas perdre d'information et d'obtenir des estimations plus précises. Enfin, certaines de ces méthodes sont testées dans plusieurs scénarios en ligne qui incluent une interaction humaine et différentes charges utiles.

Abstract — Robotics has played an important role on industrial changes over the last decades, allowing to reduce costs, improve quality, increase productivity and reduce dangers for employees. Even though robots were first designed to work in industrial scenarios inside cages, there has been a tendency over the last years to delete these barriers and allow the robots not only to share their workspace with humans, but also their tasks and objectives. This has widened their application area whose result is an increasingly frequent appearance of robots in our daily lives and a necessity to master the robot in a better way. Whether it is needed to design a control law, simulate the system to predict future states, or perform fault or collision detection, an accurate and reliable model is mandatory. The best way to do this is by gray-box modeling techniques: they make use of the knowledge of the physical laws that govern the system and of experimental data. The increasingly complexity of the systems and its surroundings, and the need to redefine the model while the system is running, bring these techniques back into the spotlight.

In the context of this thesis, the focus is made on the development and enhancement of parameter estimation methods to be applied in the identification of the dynamic model of robotic systems. The case of collaborative robotics is studied by validating all the methods with the KUKA LBR iiwa 14 R820. However, our methods are general and applicable to a wide variety of robots, as results using the Pioneer LX 2-wheels differential drive mobile robot and the Stäubli TX40 industrial manipulator are also shown.

The contributions are presented in two main axes. On the one hand, offline en-bloc estimation methods are treated, which are usually used to have a first idea of the model. Considering that the parameters have a physical meaning and that measurements are inherently noisy, the first proposed algorithm is robust against noise and ensures physical consistency of the estimates. The second proposed method addresses two other hindrances: the facts that manufacturers of commercial robots often hide important information and measurements to the users due to copyright and safety reasons; and the important but not fully understood friction phenomenon. The method identifies the model that the manufacturer has included in their controller in a reverse-engineering process, and uses it to estimate friction parameters.

On the other hand, in systems that change during time or interact with an unknown dynamically changing environment, the estimation of the model and its parameters must be updated in an online basis. For this purpose, recursive variants must be analyzed. Aspects as algorithm initialization, stability of estimates and computing time gain importance. In this context, we first develop a new method that yields consistent estimates with noisy measurements, its robust against initial values and does not require an external simulation of the system. Second, although it is known that global identification is more accurate than sequential identification, sometimes the latter is the only possibility. We develop new methods which propagate the statistical distribution of estimates in the different identification steps, leading to no loss of information and more precise estimates. Finally, some of these methods are tested in several online scenarios which include human interaction and different payloads.

Keywords: Parameter Estimation; Grey-Box Modeling; Online Identification; Robot Manipulator; Collaborative Robotics; Instrumental Variable; Sequential Identification.