



Université  
de Toulouse

# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :**

Institut National Polytechnique de Toulouse (INP Toulouse)

**Discipline ou spécialité :**

Systèmes Embarqués

---

**Présentée et soutenue par :**

M. MATHIEU BRUNOT

le jeudi 30 novembre 2017

**Titre :**

Identification of rigid industrial robots A system identification perspective

---

**Ecole doctorale :**

Systèmes (Systèmes)

**Unité de recherche :**

Laboratoire de Génie de Productions de l'ENIT (E.N.I.T-L.G.P.)

**Directeur(s) de Thèse :**

M. FRANCISCO CARRILLO

M. ALEXANDRE JANOT

**Rapporteurs :**

M. JOONO CHEONG, KOREA UNIVERSITY

M. YANNICK Aoustin, UNIVERSITE DE NANTES

**Membre(s) du jury :**

M. PHILIPPE BIDAUD, ONERA - CENTRE DE PALAISEAU, Président

M. ALEXANDRE JANOT, ONERA, Membre

M. FRANCISCO CARRILLO, ECOLE NATIONALE D'INGENIEUR DE TARBES, Membre

M. HUGUES GARNIER, UNIVERSITÉ LORRAINE, Membre

M. JEAN-PHILIPPE NOEL, UNIVERSITE DE LIEGE, Membre



# Acknowledgements

Finally, it is time to write my acknowledgements and it is maybe the most difficult part to do. I may forget someone in these lines and I would like to apologise in advance for that.

First of all, I wish to thank you, Alexandre and Francisco, for your support, involvement and availability. Obviously, this work is also the result of your knowledge and experiences. I would also like to thank Pr. Yannick Aoustin and Pr. Joono Cheong for accepting to review this thesis manuscript. Their interesting and kind comments were appreciated. Many thanks also go to Pr. Bidaud who accepted to preside the defence committee as well as to the remaining members of this committee, Pr. Garnier and Dr. Noël, for the interesting questions session.

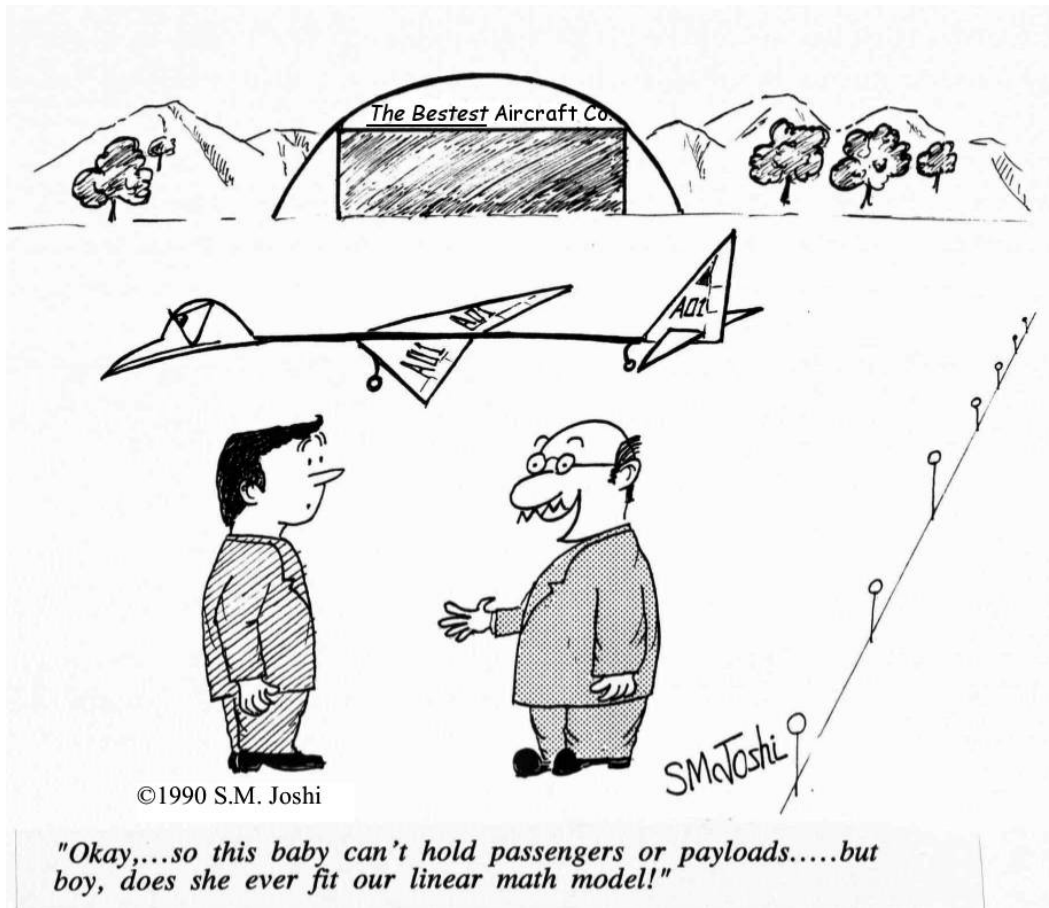
There was a great atmosphere amongst PhD students that has been crucial for my motivation. Manu, it was my pleasure to "share breaks and cycle". Guillaume, after Champo, N6K and Onera, I hope our paths will cross again. Adèle, good luck for the Lot and meanwhile keep ruling the office. Pauline, Gustav and Alexandre, good luck with Adèle! A special thank to our Italian connection of Post-doc: Matteo and Mario. Hélène, Thomas, Jorrit, Mehdi, Vincent, thanks for the joyful coffee breaks and after-works. I forget those who are already doctors: Alvaro, Igor, Martin, Victor, Jérémy, Adrien, Patrick, Nicolas... Good luck to the new candidates!

I wish also to thank the whole staff of the former "Département de Commande des Systèmes et de Dynamique du Vol". That was a pleasure to work amongst you. I am therefore grateful to have had the opportunity to stay. Not the least, I wish to thank Jean and all the Eole team for the trust they placed in me. My thanks are also directed to Onera for the financial support, as well as Region Midi-Pyrénées. I also have a thought for the DIDS team in Tarbes. I unfortunately ran out of time to spend more time with them. A special thank to Johann and his colleagues in the UAV team of DLR, Braunschweig. Those three months were a breath of fresh air and an instructive break.

Last but not least, I wish to thank my family for their unconditional support, at all times. I feel extremely fortunate to have had this opportunity to study for the last nine years. Finally, I thank Amandine without whom nothing would be possible.

*Mathieu*





"All models are wrong but some are useful" [Box 1979]

From this cartoon and this quotation, we will keep in mind throughout this work that the question is not to have the model with the best fit but the one which meets the expectations of the end user.



# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Table of Contents</b>	<b>v</b>
<b>Notations</b>	<b>ix</b>
<b>Acronyms</b>	<b>xi</b>
<b>List of Publications</b>	<b>xiii</b>
<b>I Introduction</b>	<b>1</b>
<b>1 Background and scope</b>	<b>3</b>
1.1 Research motivation . . . . .	4
1.2 State of the art . . . . .	6
1.3 Goal . . . . .	9
1.4 Thesis outline . . . . .	9
<b>II Preliminaries</b>	<b>11</b>
<b>2 Robots modelling and identification</b>	<b>13</b>
2.1 Modelling of robot arms . . . . .	14
2.1.1 Geometric model of serial robots . . . . .	14
2.1.2 Inverse dynamic model . . . . .	15
2.1.3 Robot dynamic parameters . . . . .	17
2.1.4 Direct dynamic model . . . . .	18
2.2 Robot systems architecture . . . . .	19
2.2.1 Robot control laws . . . . .	19
2.2.2 Robot actuators . . . . .	21
2.2.3 Position sensors . . . . .	21
2.2.4 Inverse dynamic identification model . . . . .	23
2.3 Robot arms identification methods . . . . .	23
2.3.1 Filtering methodology . . . . .	24
2.3.2 The least-squares method . . . . .	24
2.3.3 The instrumental variable method . . . . .	26
2.3.4 The DIDIM method . . . . .	28
2.3.5 Exciting trajectories . . . . .	30
2.4 Concluding remarks . . . . .	31

<b>3</b>	<b>System identification for continuous time systems</b>	<b>33</b>
3.1	Closed-loop identification challenges . . . . .	34
3.2	The refined instrumental variable method . . . . .	34
3.2.1	The hybrid Box-Jenkins model . . . . .	34
3.2.2	The hybrid RIVC algorithm . . . . .	36
3.2.3	Statistical elements of the method . . . . .	38
3.3	Prediction error methods . . . . .	39
3.3.1	Prediction error principle . . . . .	39
3.3.2	DT prediction error method . . . . .	40
3.3.3	Hybrid output error method . . . . .	42
3.3.4	Computational aspects . . . . .	43
3.4	Concluding remarks . . . . .	44
<b>III</b>	<b>Contributions to robot system identification</b>	<b>45</b>
<b>4</b>	<b>Evaluation of the robustness of identification methods based on the auxiliary model simulation</b>	<b>47</b>
4.1	Choice of the identification signal . . . . .	48
4.1.1	Output error methods for robot identification . . . . .	48
4.1.2	Sensitivity analysis . . . . .	49
4.1.3	Contribution of the linearity . . . . .	55
4.1.4	Experimental validation . . . . .	55
4.2	Robustness to modeling errors of the DIDIM and IDIM-IV methods	61
4.2.1	Similarity between the methods . . . . .	61
4.2.2	Disparity between the methods . . . . .	63
4.2.3	Experimental validations . . . . .	65
4.3	Miscellaneous remarks . . . . .	66
4.3.1	Closed-loop aspects of the robot identifications methods . . .	66
4.3.2	Considerations on the robot identification with OEM . . . . .	66
4.3.3	Remarks on the linearity . . . . .	68
4.4	Conclusions . . . . .	70
<b>5</b>	<b>Robot system identification without the knowledge of the controller and the bandwidths</b>	<b>71</b>
5.1	Joint derivatives estimation . . . . .	72
5.1.1	Derivatives estimation from noisy signals . . . . .	72
5.1.2	Integrative random walk smoother . . . . .	73
5.1.3	Application to robot identification . . . . .	76
5.1.4	Numerical differentiation tests . . . . .	80
5.2	Controller identification . . . . .	81
5.2.1	Closed-loop system identification with an unknown controller	81
5.2.2	Controller parametric identification . . . . .	83
5.2.3	Controller non-parametric identification . . . . .	83



5.3	Experimental validation . . . . .	85
5.3.1	The IDIM-LS method with the IRWSM technique . . . . .	85
5.3.2	The IDIM-IV method with the IRWSM technique . . . . .	87
5.3.3	The IDIM-IV method with an unknown controller . . . . .	90
5.4	Conclusions . . . . .	94
<b>6</b>	<b>Noise modelling for robot system identification</b>	<b>97</b>
6.1	Noise modelling and identification . . . . .	98
6.1.1	Closed-loop relations . . . . .	98
6.1.2	Filters models . . . . .	99
6.1.3	Auto-regressive filters identification . . . . .	103
6.2	Refined identification methods . . . . .	105
6.2.1	The separable PEM method . . . . .	105
6.2.2	The IDIM-PIV method . . . . .	106
6.2.3	Comments on the extensions . . . . .	107
6.3	Experimental validations . . . . .	108
6.3.1	Study of the black-box noise model . . . . .	108
6.3.2	Study of the grey-box noise model . . . . .	112
6.3.3	Study with noisy data . . . . .	113
6.4	Conclusions . . . . .	116
<b>IV</b>	<b>Conclusion</b>	<b>121</b>
<b>7</b>	<b>Discussions and Future Work</b>	<b>123</b>
7.1	Contributions summary . . . . .	124
7.1.1	Analysis of the OEM for robot identification . . . . .	124
7.1.2	Robustness analysis of the IDIM-IV and DIDIM methods . . . . .	124
7.1.3	Identification with an unknown controller . . . . .	124
7.1.4	Automated estimation of the joint derivatives . . . . .	125
7.1.5	Noise filter identification . . . . .	125
7.2	A practitioner guide . . . . .	126
7.3	Further developments . . . . .	126
7.3.1	Actuator modelling . . . . .	126
7.3.2	Friction modelling . . . . .	126
7.3.3	Broader perspectives . . . . .	128
<b>V</b>	<b>Appendices</b>	<b>129</b>
<b>A</b>	<b>Industrial robot modelling</b>	<b>131</b>
A.1	Computation of the energies . . . . .	131
A.2	Newton-Euler formulation . . . . .	135
A.3	Parameter reduction . . . . .	136

<b>B</b>	<b>Stäubli TX40</b>	<b>143</b>
B.1	Model of the TX40 . . . . .	143
B.2	Reference values of the dynamic parameters . . . . .	143
B.3	Control law . . . . .	145
B.4	Illustration of experimental signals . . . . .	146
<b>C</b>	<b>Mathematical background</b>	<b>149</b>
C.1	Random Variables . . . . .	149
C.2	Properties of Estimators . . . . .	150
C.3	Random Process . . . . .	151
<b>VI</b>	<b>Résumé de thèse en français</b>	<b>153</b>
<b>D</b>	<b>Identification de robots industriels rigides – Apport des méthodes de l’identification de systèmes</b>	<b>155</b>
D.1	Introduction . . . . .	157
D.1.1	Contexte . . . . .	157
D.1.2	État de l’art . . . . .	158
D.1.3	Problématique . . . . .	159
D.2	Evaluation de la robustesse des méthodes d’identification basées sur la simulation du modèle auxiliaire . . . . .	160
D.2.1	Choix du signal d’identification . . . . .	160
D.2.2	Comparaison des méthodes IDIM-IV et DIDIM . . . . .	163
D.2.3	Remarques et conclusion . . . . .	165
D.3	Identification de systèmes robotiques sans connaissance du contrôleur et des bande-passantes . . . . .	165
D.3.1	Estimation des dérivées articulaires . . . . .	166
D.3.2	Identification du contrôleur . . . . .	170
D.3.3	Conclusions . . . . .	173
D.4	Modélisation du bruit pour l’identification de systèmes robotiques . . . . .	174
D.4.1	Modélisation et identification du bruit . . . . .	174
D.4.2	Extension des méthodes d’identification . . . . .	177
D.4.3	Conclusions . . . . .	181
D.5	Conclusions et perspectives . . . . .	181
D.5.1	Résumé des contributions . . . . .	181
D.5.2	Un guide pour l’utilisateur . . . . .	183
D.5.3	Développements futurs . . . . .	184
	<b>References</b>	<b>187</b>
	<b>Index</b>	<b>201</b>

# Notations

## Symbols

$b$	Length of the dynamic parameters vector
$\mathbf{G}_\tau$	Matrix of joint drive gains, $dim(\mathbf{G}_\tau) = (n \times n)$
$\mathbf{H}$	Matrix of sensor noise filter, $dim(\mathbf{H}) = (n \times n)$
$j$	Subscript referring to a robot's link
$l$	Length of the noise parameters vector
$N$	Number of sampling points considered for the identification
$n_m$	Number of recorded sampling points
$n_d$	Factor of decimation
$n$	Number of robot's links
$p$	Differential operator, $p = d/dt$
$\mathbf{q}$	Vector of joint positions, $dim(\mathbf{q}) = (n \times 1)$
$\mathbf{q}_{nf}$	Vector of noise-free joint positions
$\mathbf{q}_m$	Vector of measured joint positions
$\mathbf{q}_r$	Vector of reference joint positions
$\mathbf{q}_s$	Vector of simulated joint positions
$\tilde{\mathbf{q}}$	Vector of measurement noise, $\mathbf{q}_m = \mathbf{q} + \tilde{\mathbf{q}}$
$\mathbf{n}_q$	Vector of position noise, $\mathbf{q}_m = \mathbf{q}_{nf} + \mathbf{n}_q$
$r$	Number of equations considered with the decimation, $r = N \cdot n$
$r'$	Number of equations considered without the decimation, $r' = n_m \cdot n$
$T_s$	Sampling time
$\mathbf{X}$	En-bloc observation matrix, $dim(\mathbf{X}) = (r \times b)$
$\mathbf{Z}$	En-bloc instrumental matrix, $dim(\mathbf{Z}) = (r \times b)$
$\mathbf{v}$	Vector of forces/torques noise, $\boldsymbol{\tau} = \boldsymbol{\tau}_{idm} + \mathbf{v}$
$\boldsymbol{\tau}$	Vector of measured forces/torques, $dim(\boldsymbol{\tau}) = (n \times 1)$
$\boldsymbol{\tau}_{idm}$	Vector of noise-free forces/torques, $dim(\boldsymbol{\tau}_{idm}) = (n \times 1)$
$\boldsymbol{\nu}_\tau$	Vector of computed command, $dim(\boldsymbol{\nu}_\tau) = (n \times 1)$
$\boldsymbol{\theta}$	Vector of the dynamic parameters, $dim(\boldsymbol{\theta}) = (b \times 1)$
$\boldsymbol{\eta}$	Vector of the noise parameters, $dim(\boldsymbol{\eta}) = (l \times 1)$
$\boldsymbol{\phi}$	Observation matrix, $dim(\boldsymbol{\phi}) = (n \times b)$
$\boldsymbol{\zeta}$	Instrumental matrix, $dim(\boldsymbol{\zeta}) = (n \times b)$
$\boldsymbol{\Sigma}$	Covariance matrix of the estimated parameters, $dim(\boldsymbol{\Sigma}) = (b \times b)$
$\Delta_{\mathbf{q}_s}$	Output sensitivity matrix, $dim(\Delta_{\mathbf{q}_s}) = (n \times b)$
$\Delta_{\boldsymbol{\tau}_s}$	Input sensitivity matrix, $dim(\Delta_{\boldsymbol{\tau}_s}) = (n \times b)$
$\omega_s$	Sampling frequency
$\omega_{nyq}$	Nyquist frequency, $\omega_{nyq} = \omega_s/2$
$\omega_{dyn}$	System's bandwidth of the closed-loop position

## Operators

$\mathbf{I}_n$	Identity matrix of size $n$
rank	Rank of the considered matrix
$s$	Laplace transform variable
$\mathbf{A}$	Matrix notation
$\mathbf{A}^{-1}$	Matrix inverse (if it exists)
$\mathbf{A}^T$	Matrix transpose
$z^{-1}$	The delay operator: $z^{-1}x(t) = x(t - 1)$
$\mathbf{x}$	Vector notation
$\mathbf{E}[\mathbf{x}]$	Mathematical expectation of the random vector $\mathbf{x}$
$\bar{\mathbf{E}}[f(t)]$	$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbf{E}[f(t_i)]$ , Total expectation operator

# Acronyms

<i>AIC</i>	Akaike Information Criterion
<i>AR</i>	Auto-Regressive
<i>CAD</i>	Computer-Aided Design
<i>CLIE</i>	Closed-Loop Input Error
<i>CLOE</i>	Closed-Loop Output Error
<i>CRC</i>	CoRrelated Controller
<i>CT</i>	Continuous-Time
<i>DT</i>	Discrete-Time
<i>DDM</i>	Direct Dynamic Model
<i>DIDIM</i>	Direct and Inverse Dynamic Identification Method
<i>DOF</i>	Degree Of Freedom
<i>FIR</i>	Finite Impulse Response
<i>FIS</i>	Fix Interval Smoother
<i>IDIM</i>	Inverse Dynamic Identification Model
<i>IDM</i>	Inverse Dynamic Model
<i>IRWSM</i>	Integrated Random Walk SMoother
<i>IV</i>	Instrumental Variable
<i>LS</i>	Least-Squares
<i>LTI</i>	Linear Time Invariant
<i>MDH</i>	Modified Denavit-Hartenberg
<i>ML</i>	Maximum Likelihood
<i>NVR</i>	Noise Variance Ratio
<i>OEM</i>	Output Error Method
<i>PEM</i>	Prediction Error Method
<i>PID</i>	Proportional-Integral-Derivative
<i>PLR</i>	Pseudo-Linear Regression
<i>PRIV</i>	Pseudo-Refined Instrumental Variable
<i>RIVC</i>	Refined Instrumental Variable for Continuous-time systems
<i>SISO</i>	Single-Input/Single-Output
<i>SNR</i>	Signal to Noise Ratio
<i>TF</i>	Transfer Function



# List of Publications

## Journals

- [1] Mathieu Brunot, Alexandre Janot, Peter C. Young, and Francisco Carrillo. An instrumental variable method for robot identification based on time variable parameter estimation. *Kybernetika*, In Press.
- [2] Mathieu Brunot, Alexandre Janot, Peter C. Young, and Francisco Carrillo. An improved instrumental variable method for industrial robot model identification. *Control Engineering Practice*, Submitted.

## International conferences with review committee

- [1] Mathieu Brunot, Alexandre Janot, and Francisco Carrillo. Continuous-time nonlinear systems identification with output error method based on derivative-free optimisation. 2017. IFAC World Congress.
- [2] Mathieu Brunot, Alexandre Janot, and Francisco Carrillo. State space estimation method for the identification of an industrial robot arm. 2017. IFAC World Congress.
- [3] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, Hugues Garnier, Pierre-Olivier Vandanjon, and Maxime Gautier. Physical parameter identification of a one-degree-of-freedom electromechanical system operating in closed loop. In *17th IFAC Symposium on System Identification (SYSID)*, volume 48, pages 823–828, 2015.
- [4] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, and Hugues Garnier. Comparison between the idim-iv method and the didim method for industrial robots identification. 2017. IEEE/ASME International Conference on Advanced Intelligent Mechatronics.
- [5] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, and Hugues Garnier. A pragmatic and systematic statistical analysis for identification of industrial robots. 2017. IEEE/ASME International Conference on Advanced Intelligent Mechatronics.
- [6] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, and Maxime Gautier. A separable prediction error method for robot identification. In *7th IFAC Symposium on Mechatronic Systems, Loughborough University, Leicestershire, UK, 5–8 September 2016*, volume 49, pages 487 – 492, 2016.

- [7] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, and Maxime Gautier. State space estimation method for robot identification. In *7th IFAC Symposium on Mechatronic Systems, Loughborough University, Leicestershire, UK, 5–8 September 2016*, volume 49, pages 228 – 233, 2016.



## Part I

# Introduction



# Background and scope

---

This chapter is the introduction of this thesis work where the main problems considered are introduced. They concern industrial rigid robots and their parametric identification. Having high quality models is of the highest importance in order to design effective control laws. Furthermore, for stability and safety reasons, robots must be identified in closed-loop which induces a correlation in the experimental data. In order to tackle those challenges, the robotic community has developed specific identification techniques based on rules of thumb and the skills of the practitioner. That motivates the development of generic methods able to deal with a system in an automatic way without *a priori* knowledge. A state of the art of the different fields of research involved is proposed and the major contributions of this thesis are introduced.

This chapter is organized as follows. Firstly, a general introduction to the considered domain is proposed in section 1.1. Secondly, a state of the art of the models and methodologies involved is provided in section 1.2. The goal of this thesis is stated in section 1.3. Finally, the manuscript outline is detailed in section 1.4.

---

<b>1.1</b>	<b>Research motivation</b> . . . . .	<b>4</b>
<b>1.2</b>	<b>State of the art</b> . . . . .	<b>6</b>
<b>1.3</b>	<b>Goal</b> . . . . .	<b>9</b>
<b>1.4</b>	<b>Thesis outline</b> . . . . .	<b>9</b>

---

## 1.1 Research motivation

Industrial robots are mechanical arms made of several links connected by joints, also called axes. Each joint is driven by an electric motor coupled with a position sensor. A computer controls the movements of the arm in order to place the end-effector in the workspace. Figure 1.1 depicts such a robot. Industrial robots are essential components of modern and automated manufacturing plants, because they allow to save costs, increase production rates, eliminate dangerous and laborious tasks for human workers, etc. This is why, during last decades, the number of industrial robots has increased radically; see Figure 1.2.

It is expected that industrial robots perform repetitive tasks with high precision and rapidity. To achieve such tasks, different characteristics must be fulfilled depending on the application. The standard ISO [ISO 1999] specifies the useful characteristics for the user that the robot's manufacturer must provide. For example, there is the position *repeatability* defining the precision with which the robot returns to a commanded position. This is thoroughly considered in section 6.1.2. According to [Siciliano *et al.* 2010], for a robot arm with a maximum reach of 1.5 m, the repeatability varies between 0.02 and 0.2 mm. In order to meet such requirements, the control laws need reliable and complete models of those robots. This precise modelling comes from a rigorous identification based on experimental data. The issue is thus for the practitioner to provide a model as precise and complete as possible.

Many difficulties are considered. Firstly, the robots operate in closed-loop, i.e. with a feedback, for stability and precision purposes. It is often difficult, even impossible for safety reasons, to plan an experiment without the feedback. Therefore, the identification methods must be able to deal with the issues induced by the closed-loop. In brief, the closed-loop creates a correlation between the output noise and the input that leads to statistical difficulties for the identification. Furthermore, if the controller is adequately tuned, the system is insensitive to any perturbation and especially to parameters' variations. Unfortunately, identification processes rely on the sensitivity of the system to parameters' variations. Secondly, according to the technology of the considered robot, many dynamics can be encountered: linear or non-linear friction, rigid or flexible. In the case of industrial robots, rigid dynamics are favoured in order to insure the precision of the task, even though for cost saving flexible structures can be envisaged; see e.g. [Wernholt 2007]. Thirdly, all the signals required for the identification are not necessarily measured. The robots are usually controlled in position and only this information is sensed. Hence, the velocity and acceleration signals must be retrieved before the identification. Finally, due to the previous difficulties, specific processes have been developed by robotic experts for identification purpose. Those processes rely on the robotics engineer's skills and several rules of thumb. In the following part, we introduce the main elements on which the identification processes are based.

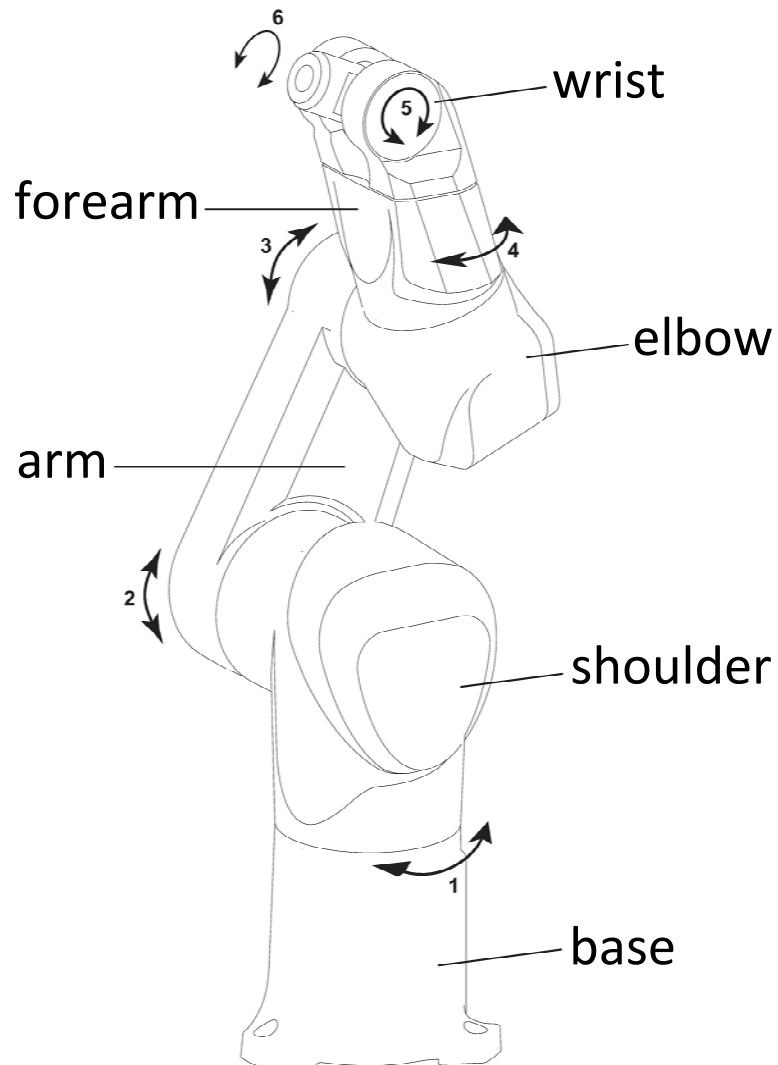


Figure 1.1: Blueprint of the Stäubli TX40, source: [Stäubli Favergues 2015]

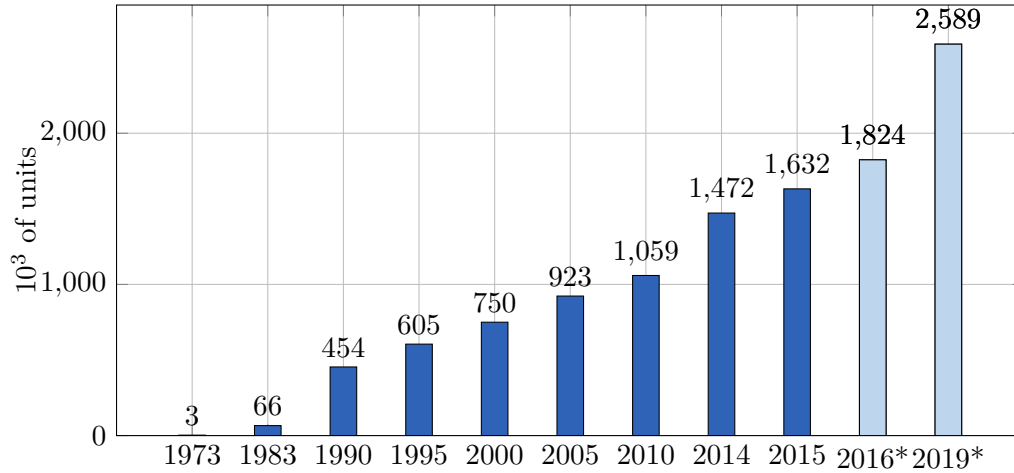


Figure 1.2: Worldwide estimated operational stock of industrial robots, source: IFR World Robotics 2016

\*: forecast

## 1.2 State of the art

The robot arms models come from the laws of classical mechanics: Newton-Euler or Lagrange formulations. In order to derive the equations of motion, a specific notation must be defined to characterise the problem. In [Denavit & Hartenberg 1955], the authors have introduced such a notation for industrial robots that has been improved by Khalil and Kleinfinger in [Khalil & Kleinfinger 1986]. This notation defines the geometric relations between the components of the robot by introducing the minimal amount of parameters. That allows expressing the dynamic models in a systematic and standardized way. If those works consider the inertial forces/torques, they do not deal with the friction phenomena. Other works have studied friction and different models have proposed; see e.g. [Canudas de Wit *et al.* 1989, Bona & Indri 2005]. With the combination of the standardized notations for inertia forces and the friction modelling, a comprehensive model of robot arms' dynamics is available.

From those modelling techniques, two model reduction methods have been proposed in [Gautier & Khalil 1990] and [Gautier 1991]. The first one is based on the algebraic relations whereas the second one uses numerical evaluation of the model. Firstly, this reduction gives the set of base parameters that can be structurally identified. Secondly, it keeps unspoiled the torque linearity with respect to the parameters. Therefore, the base parameters can be estimated with the Least-Squares (LS) technique, provided that the issue of the noise correlation is correctly handled. For this purpose, a specific procedure of experimental data prefiltering has been designed. The first step of this procedure, described in [Gautier 1997], consists in filtering the measured position to estimate the velocity and the acceleration. For the second step, a decimate filter is applied to the torque in order to reject high fre-

quency perturbations. That process makes the LS feasible from a statistical point of view. This whole process (prefiltering and LS estimation) is referred to as the IDIM-LS method. The drawbacks of this method are that the estimation result is sensitive to the tuning of the filters and this tuning is based on *a priori* knowledge of the system's bandwidth.

Other robot identification approaches have been developed over the years, without really improving the IDIM-LS method coupled with an appropriate setting of the filters. One can cite: the Total Least-Squares (TLS) method [Gautier *et al.* 1994, Xi 1995], a method based on Linear Matrix Inequality (LMI) tools [Calafiore & Indri 2000], the Extended Kalman Filter (EKF) [Kostic *et al.* 2004] or an approach based on the Maximum Likelihood (ML) [Olsen & Petersen 2001, Olsen *et al.* 2002].

In [Janot *et al.* 2014c], the authors have introduced an Instrumental Variable (IV) approach, coming from the econometrics and for the robot identification. The idea is to simulate the closed-loop model to obtain signals non-correlated with the measurement noise. This method, named IDIM-IV, uses the same prefiltering process as the IDIM-LS method although the final estimates are less sensitive to the setting of the filters. In addition to the system's bandwidths, the controller model for the simulation is also required. To simplify the prefiltering process, in [Gautier *et al.* 2013a], the authors have developed the Direct and Inverse Dynamic Identification Method (DIDIM) that is also based on the simulation of the closed-loop model. This method has the advantage of not requiring the first step of the prefiltering process because it based only on the torque measurement. However, the second step (the decimate filter) is still necessary as well as *a priori* knowledge (controller and bandwidths).

The robot identification methods introduced above share the common property of not identifying the noise model. In the system identification community, Ljung has introduced the Prediction Error Method (PEM) that is able to identify mainly Linear Time Invariant (LTI) models in open- or closed-loop; see [Ljung 1976]. The considered models are discrete-time and include the noise model. Unfortunately, this method cannot be applied on robots since their models are nonlinear and continuous-time. Almost simultaneously, Young and Jakeman have introduced the Refined IV for Continuous time systems (RIVC) in [Young & Jakeman 1980]. With a different approach, this method is able to identify LTI models in open- or closed-loop, but in a continuous time framework. Due to the nonlinearities, the RIVC method can neither be applied to industrial robots in a straightforward manner. For both methods, identifying the noise allows an automatic filtering of the data and provides optimal estimated parameters. It should be noticed that a specific kind of PEM, called Output Error Method (OEM), is able to deal with the identification of continuous-time and nonlinear systems; see e.g. [Richalet *et al.* 1971] for the generic framework or [Landau *et al.* 1999] for closed-loop systems applications. Nonetheless, OEM do not take into account the noise model.

Figure 1.3 depicts the usual robot identification process. The general frame is common to all systems. The specificities of each application lie in the content of

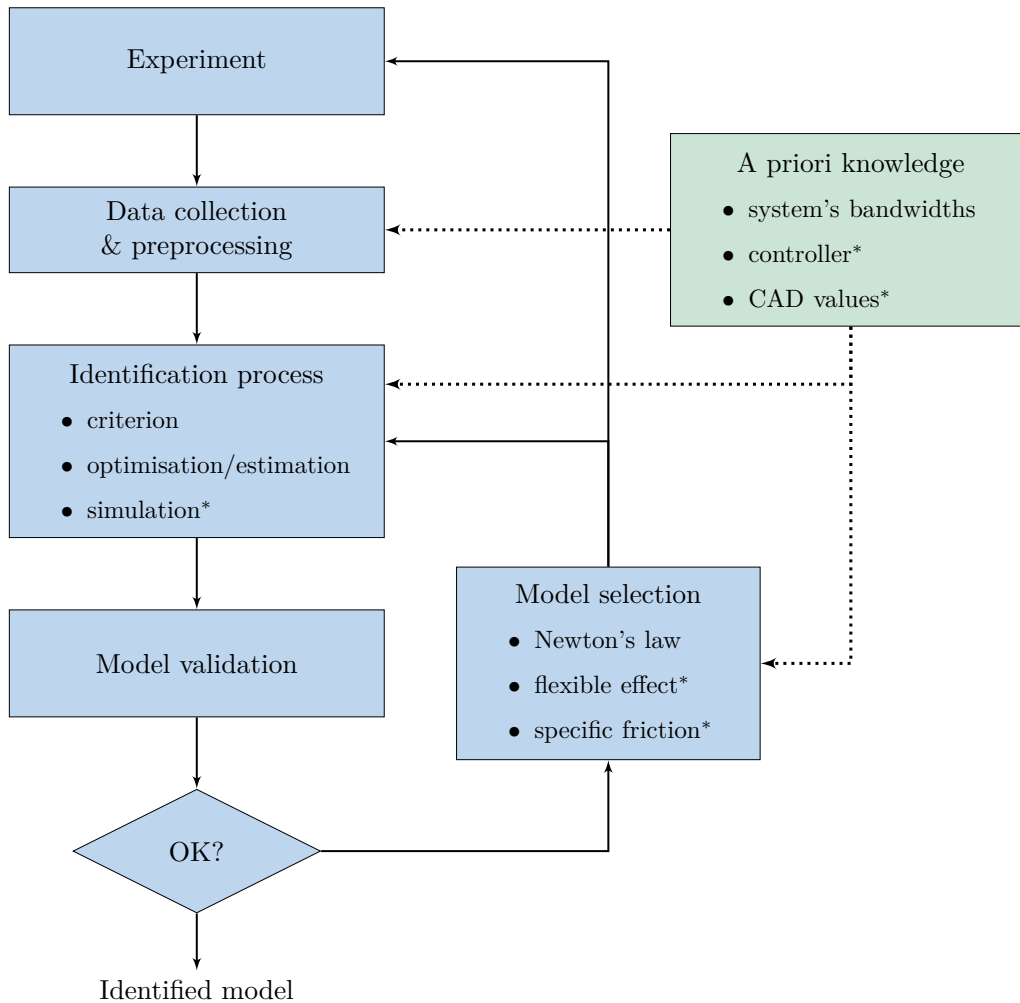


Figure 1.3: Block diagram of the robot identification process

\*: not necessary for certain methods

each block. At the beginning, there is an experiment. The data must be collected and eventually preprocessed. An identification method is run and the results are analysed. If the result is satisfactory the identification is complete otherwise the practitioner has many possibilities. First of all, the model can be modified to include other dynamics. The identification method can be modified: optimisation algorithm, initial estimate, criterion, etc. In some cases, another experiment may be required to better excite the system's dynamics or to investigate specific phenomena. The block diagram also illustrates the need of *a priori* knowledge for robot identification. For the preprocessing, the system's bandwidths are required to tune the filters. For the identification, initial parameters are required for the IDIM-IV and DIDIM methods. Concerning the model selection, an expertise is required to know if flexible dynamics must be modelled or if the friction model is adequate, for instance.



## 1.3 Goal

It appears that the robotic community has at its disposal methods able of identifying industrial robots operating in closed-loop. The challenges of the closed-loop are handled differently depending on the considered method. In parallel, the system identification community has developed methods to identify closed-loop systems in an automatic and accurate way. That is to say that they identify the noise model without *a priori* knowledge on the system. Nonetheless, those methods deal with LTI systems, which make them impractical for industrial robots. On their side, the robotic engineers do not have dealt with the noise modelling thoroughly. Their methods are based on an "home-made" prefiltering relying on a physical knowledge of the system. If they wonder about the level of the noise and its location in the frequency range, they do not try to model it.

That is in this context that this work take place. This thesis consists in developing identification methods for industrial robot systems in an automatic and accurate way, while minimising the required *a priori* knowledge, in order to bridge the gap between the robotic and system identification communities.

## 1.4 Thesis outline

This manuscript is divided into four parts. Part I is devoted to the introduction. Materials on the existing theories required to understand the contributions of this work are provided in Part II. Part III contains the main contributions of this thesis with regard to robot system identification. Conclusions are provided in Part IV before some appendices in Part V.

### Part II: Preliminaries

The preliminaries are divided in two chapters. Chapter 2 aims at describing the robot system modelling as well as the usual methods for the estimation of the dynamic parameters. The elements of modelling are mainly based on the reference materials [Khalil & Dombre 2004] and [Siciliano *et al.* 2010]. In accordance with the preview given in the state of the art, the identification methods introduced are the IDIM-LS, IDIM-IV and DIDIM ones.

Chapter 3 introduced other methods coming from the system identification community. If those methods cannot be applied straightforwardly to robot applications, they have interesting properties that may improve the robot identification process. The chapter mainly focuses on the RIVC and PEM methods introduced in the state of the art. The core material of this chapter can be found in [Söderström & Stoica 1988], [Ljung 1999] and [Young 2011].

### Part III: Contributions to robot system identification

In chapter 4, the approach consists in evaluating the sensitivity and the robustness of the robot identification methods based on the auxiliary model simulation. Firstly,

the best observation signal must be selected for the identification purpose: the input torque or the output position. To this end, a sensitivity analysis with respect to the dynamic parameters is performed. In addition, the benefit of the torque's linearity with respect to the parameters is highlighted. This element has resulted in a publication [Brunot *et al.* 2015]. Afterwards, we show a better robustness of the IDIM-IV method compared with the DIDIM one in order to establish their limits. This comparison is undertaken by considering an error-free model to illustrate a nominal case. Then, an error is introduced in auxiliary model to put the methods in default. This analysis has been presented in [Brunot *et al.* 2017c].

The following step is to revisit those methods in chapter 5 to reduce the *a priori* knowledge required to their execution. The idea is to deal with a robot system whose bandwidths and controller are unknown. In a first part, we look for an automatic method to estimate the joint velocities and accelerations to avoid the prefiltering process. To do so, a method is selected from literature and adapted to industrial robots. This work has resulted in several publications [Brunot *et al.* 2016b], [Brunot *et al.* 2017b] and a submission to [Brunot *et al.* In Press 2017]. The second part focuses on the identification of the controller. It is indeed necessary to the auxiliary model simulation, which is itself necessary to the dynamic model identification. The controller identification is tackled in parametric and in a non-parametric way. The principles of the controller identification has been submitted to [Brunot *et al.* Submitted 2017]. The third part is dedicated to experimental validations of the suggested methods.

The last step consists in continuing the effort of *a priori* knowledge reduction by taking into account the noise model instead of using the decimate filter. The aim is to provide estimated parameters as accurate as possible without relying on the practitioner's skills. A first part is devoted to the noise modelling through the closed-loop system. From the closed-loop relations, the model of the noise affecting the input torque is derived. The second part uses this noise model to replace the usual decimate filter used prior to the identification. A first method, based on the DIDIM approach, is called separable PEM in connection with the method developed within the system identification community. A second method, referred to as IDIM-PIV, is directly inspired from the RIVC method. Those methods have been published in [Brunot *et al.* 2016a] and submitted to [Brunot *et al.* Submitted 2017], following a preliminary work presented in [Brunot *et al.* 2017d]. The third part deals with the experimental validation of the two suggested methods.

Part II

Preliminaries



# Robots modelling and identification

Robot modelling has been extensively studied during the last decades. To standardize the coordinate frames, several conventions have been proposed. One of the most popular is the Denavit-Hartenberg convention developed in [Denavit & Hartenberg 1955] and modified in [Khalil & Kleinfinger 1986]. In this thesis, we focus on the modelling of rigid robots with single open chain structure thanks to the Modified Denavit-Hartenberg (MDH) convention. Based on this geometric convention, the Newton's laws give the equations of motion. From these equations and taking into account the robot systems architecture, techniques have been developed to identify the dynamic models. The material of this chapter mainly come from the book [Khalil & Dombre 2004] and summarizes the main results in robots modelling and identification.

An introduction to the MDH convention and the different dynamic models considered for robot identification is proposed in section 2.1. In section 2.2, the whole structure of a robot arm system is described taking into account the controller, the sensors and the actuators. Section 2.3 outlines the common methods for robot identification. Conclusions are provided in section 2.4.

<b>2.1</b>	<b>Modelling of robot arms</b>	<b>14</b>
2.1.1	Geometric model of serial robots	14
2.1.2	Inverse dynamic model	15
2.1.3	Robot dynamic parameters	17
2.1.4	Direct dynamic model	18
<b>2.2</b>	<b>Robot systems architecture</b>	<b>19</b>
2.2.1	Robot control laws	19
2.2.2	Robot actuators	21
2.2.3	Position sensors	21
2.2.4	Inverse dynamic identification model	23
<b>2.3</b>	<b>Robot arms identification methods</b>	<b>23</b>
2.3.1	Filtering methodology	24
2.3.2	The least-squares method	24
2.3.3	The instrumental variable method	26
2.3.4	The DIDIM method	28
2.3.5	Exciting trajectories	30
<b>2.4</b>	<b>Concluding remarks</b>	<b>31</b>

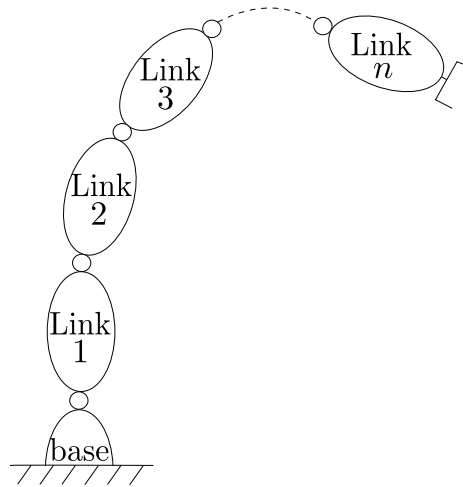


Figure 2.1: Robot with serial structure

## 2.1 Modelling of robot arms

### 2.1.1 Geometric model of serial robots

A robot with a single open chain (or serial) structure, also called robot arm, is composed of  $n$  moving links as shown in Figure 2.1. From this figure, we can give some definitions.

- *Link*: A link is an element of the robot. In this thesis, the links are assumed to be rigid and the subscript  $j$  designates a robot link between 0 and  $n$ .
- *Joint*: A joint connects two successive links and defines the number of degrees of freedom between them. This number lies between 1 and 6 (3 translations and 3 rotations). In robotics, it is usually equal to 1. In this case, the joint is either *prismatic* (translation) or *revolute* (rotation).
- *Base*: The base is the link 0 of the arm which is fixed.
- *End-effector*: The end-effector is any device at the other end of the arm. This is the interface between the robot and its environment. It can be seen as the payload of the robot. Its subscript is noticed  $n$ .

With the MDH convention, a frame  $R_j$  is attached to each link  $j$ . The axes are defined with three unit vectors such that:

- $\mathbf{z}_j$  is along the axis of joint  $j$ ;
- $\mathbf{x}_j$  is perpendicular to  $\mathbf{z}_j$  and  $\mathbf{z}_{j-1}$ . If  $\mathbf{z}_j$  and  $\mathbf{z}_{j-1}$  are parallel,  $\mathbf{x}_j$  is not uniquely defined. The origin  $O_j$  is defined by the intersection of  $\mathbf{x}_j$  and  $\mathbf{z}_j$ ;
- $\mathbf{y}_j$  is defined by the right hand screw rule.

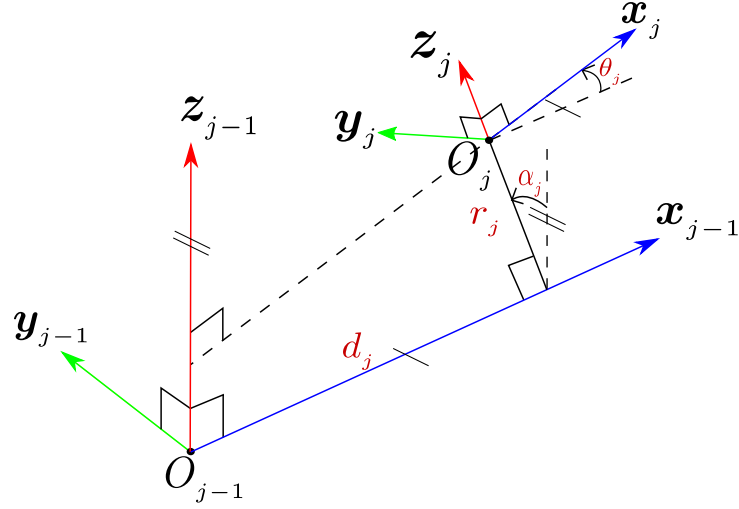


Figure 2.2: Modified Denavit-Hartenberg notations

Figure 2.2 depicts the geometric relations between the axes. To go from  $R_{j-1}$  to  $R_j$ , the following transformations are required:

- a rotation through an angle  $\alpha_j$  about  $\mathbf{x}_{j-1}$ ;
- a translation  $d_j$  along  $\mathbf{x}_{j-1}$ ;
- a rotation through an angle  $\theta_j$  about  $\mathbf{z}_j$ ;
- a translation  $r_j$  along  $\mathbf{z}_j$ .

The set of parameters  $(\alpha_j, d_j, \theta_j, r_j)$  has the minimal size to model the relation between two consecutive links. It should be noticed that  $\theta_j$  is used in this section by respect for the literature. In the remainder of this thesis,  $\theta$  will refer to the parameters to be estimated. We also define the position  $q_{nf_j}$  as the relative orientation between links  $j - 1$  and  $j$ , such as:

$$q_{nf_j} = \begin{cases} r_j, & \text{if joint } j \text{ is prismatic} \\ \theta_j, & \text{if joint } j \text{ is revolute} \end{cases} . \quad (2.1)$$

The subscript  $nf$  designates noise-free signals which are perfectly known. As we are going to see later in section 2.2.3, the measurement noises introduced by the positions sensors lead to noisy positions signals.

### 2.1.2 Inverse dynamic model

The Inverse Dynamic Model (IDM) is the expression of the joint torques and forces in terms of joint positions, velocities and accelerations. This model can be derived from the Lagrange or the Newton-Euler formulations. In this section, we succinctly

develop the Lagrange formulation to illustrate the robots dynamics. The Newton-Euler formulation is developed in appendix A.2.

For a robot with  $n$  moving links, the Lagrange equations can be written in the form:

$$\boldsymbol{\tau}_{idm} = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}_{nf}} \right) - \frac{\partial L}{\partial \mathbf{q}_{nf}} + \boldsymbol{\tau}_f, \quad (2.2)$$

with

- $\boldsymbol{\tau}_{idm}$  is the  $(n \times 1)$  vector of joint torques or forces, depending on the joint technology (revolute or prismatic);
- $\mathbf{q}_{nf}$ ,  $\dot{\mathbf{q}}_{nf}$ ,  $\ddot{\mathbf{q}}_{nf}$  are respectively the  $(n \times 1)$  noise-free vectors of joint positions, velocities and accelerations;
- $\boldsymbol{\tau}_f$  is the  $(n \times 1)$  vector of friction;
- $L = E - U$ , where  $L$ ,  $E$  and  $U$  are respectively the Lagrangian, the kinetic energy and the potential energy of the system.

The expressions of the energies are developed in appendix A.1. The development of the equations leads to :

$$\boldsymbol{\tau}_{idm} = \mathbf{M}(\mathbf{q}_{nf}) \ddot{\mathbf{q}}_{nf} + \mathbf{N}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}), \quad (2.3)$$

where  $\mathbf{M}$  is the  $(n \times n)$  inertia matrix of the robot (see appendix A.1) and  $\mathbf{N}$  is the  $(n \times 1)$  vector of centrifugal, Coriolis, gravitational, and friction torques. As it has been said, the dynamics is modelled in rigid framework. The reader interested in a flexible representation could refer to [Cheong *et al.* 2004] for instance.  $\mathbf{N}$  can be broken down such as:

$$\mathbf{N}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) = \mathbf{C}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) \dot{\mathbf{q}}_{nf} + \mathbf{Q}(\mathbf{q}_{nf}) + \boldsymbol{\tau}_f \quad (2.4)$$

where:

- $\mathbf{C}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) \dot{\mathbf{q}}_{nf} = \frac{d\mathbf{M}}{dt} \dot{\mathbf{q}}_{nf} - \frac{\partial E}{\partial \mathbf{q}_{nf}}$  is the  $(n \times 1)$  vector of centrifugal and Coriolis torques;
- $\mathbf{Q}(\mathbf{q}_{nf}) = \frac{\partial U}{\partial \mathbf{q}_{nf}}$  is the  $(n \times 1)$  vector of gravity torques.

The coefficients of the matrix  $\mathbf{M}$ , as well as those of the vectors  $\mathbf{C}$  and  $\mathbf{Q}$ , are functions of the geometric and inertial parameters of the considered robot. This is emphasized in appendix A.1. The IDM is thus a system of  $n$  second order differential equations, coupled and nonlinear with respect to the states (positions and velocities).



### 2.1.3 Robot dynamic parameters

According to [Khalil & Dombre 2004] and the references given therein like [Gautier & Khalil 1990], a joint  $j$  of an industrial robot has 14 standard parameters :

$$\boldsymbol{\chi}_j = [XX_j \quad XY_j \quad XZ_j \quad YY_j \quad YZ_j \quad ZZ_j \quad MX_j \quad MY_j \quad MZ_j \quad M_j \quad Ia_j \quad F_{v_j} \quad F_{c_j} \quad \tau_{off_j}]^T \quad (2.5)$$

where

- $XX_j, XY_j, XZ_j, YY_j, YZ_j$  and  $ZZ_j$  are the six components of the inertia tensor  ${}^j\mathbf{J}_{O_j}$  defined at the origin of frame  $O_j$ , in the axes of the frame  $R_j$ , such as:

$$\begin{aligned} {}^j\mathbf{J}_{O_j} &= \begin{bmatrix} XX_j & XY_j & XZ_j \\ XY_j & YY_j & YZ_j \\ XZ_j & YZ_j & ZZ_j \end{bmatrix} \\ &= \begin{bmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2) dm \end{bmatrix} \end{aligned} \quad (2.6)$$

- $MX_j, MY_j, MZ_j$  are the three components of the first moments  ${}^j\mathbf{M}\mathbf{S}_j = M_j \cdot {}^j\mathbf{S}_j = [MX_j \quad MY_j \quad MZ_j]^T$ , where  ${}^j\mathbf{S}_j = [X_j \quad Y_j \quad Z_j]^T$  is vector from the origin  $O_j$  to the center of gravity  $G_j$ , expressed in the axes of the frame  $R_j$ ;
- $M_j$  is the mass of link  $j$ ;
- $Ia_j$  is the total inertia moment for rotor and gears of the actuator;
- $F_{v_j}$  and  $F_{c_j}$  are respectively the viscous and Coulomb friction coefficients;
- $\tau_{off_j}$  is an offset parameter containing the asymmetry of the Coulomb friction with respect to the sign of the velocity and the current amplifier offset which supplies the motor.

The friction is an important phenomenon which can represent 10 to 20 % of the nominal actuator torque, even 30% in some cases [Lischinsky *et al.* 1999]. In accordance with [Canudas de Wit *et al.* 1989, Daemi & Heimann 1997, Bona & Indri 2005], for a given link  $j$ , a simple and common friction model used in robotics is:

$$\tau_{f_j} = F_{c_j} \text{sign}(\dot{q}_{nf_j}) + F_{v_j} \dot{q}_{nf_j} + \tau_{off_j}, \quad (2.7)$$

with

$$\text{sign}(\dot{q}_{nf_j}) = \begin{cases} 1, & \text{if } \dot{q}_{nf_j} > 0 \\ 0, & \text{if } \dot{q}_{nf_j} = 0 \\ -1, & \text{if } \dot{q}_{nf_j} < 0 \end{cases}$$

This model is usually satisfactory for the range of velocity used for robot trajectories. However, in some specific cases like low velocity, the friction model may be more involved [Hamon *et al.* 2011, Janot *et al.* 2017]. For simulation purpose, the *sign* function can be replaced by the *arctangent* function which has the advantage to be continuous. In fact, the function  $\frac{2}{\pi} \text{atan}(\gamma \dot{q}_j)$  tends to  $\text{sign}(\dot{q}_j)$  when  $\gamma$  tends to infinity. The value of  $\gamma$  depends on the practical case and especially on the velocity range of the trajectory. The user must keep in mind that, with a too large value, the numerical integration of the differential equations would become stiff. A stiff model can be described as a system with very fast components compare with others; see e.g. [Murray-Smith 1995, Hairer & Wanner 1996] for further information on stiff systems simulation.

All the standard parameters are not identifiable. As shown in [Gautier 1991] for instance, some dynamic parameters have indeed no influence on the IDM while some others are regrouped together thanks to linear relations. The set of identifiable dynamic parameters has to be determined to obtain a  $(b \times 1)$  vector:  $\boldsymbol{\theta}$ . This vector is called as the set of base parameters. They are in fact the minimum number of dynamic parameters from which the IDM can be calculated. Appendix A.3 details the two methods available to determine the base parameters. The IDM can be written as a linear function of those base parameters:

$$\boldsymbol{\tau}_{idm} = \boldsymbol{\phi}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}, \ddot{\mathbf{q}}_{nf}) \boldsymbol{\theta}, \quad (2.8)$$

where  $\boldsymbol{\phi}$  is the  $(n \times b)$  matrix of basis functions (also called observation matrix) and  $\boldsymbol{\theta}$  is the  $(b \times 1)$  vector of parameters. Each element of  $\boldsymbol{\phi}$  is a basis function of the body dynamics, which is also called *regressor* or *independant variable*. Those basis functions can be nonlinear relations of the positions, velocities and accelerations.

#### 2.1.4 Direct dynamic model

The Direct Dynamic Model (DDM) provides the joint accelerations in terms of the joint positions, velocities and torques as well as the parameters. It is described by:

$$\ddot{\mathbf{q}}_{nf} = \mathbf{M}^{-1}(\mathbf{q}_{nf}) \left( \boldsymbol{\tau}_{idm} - \mathbf{N}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) \right). \quad (2.9)$$

The DDM can also be written as a state-space form given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{q}_{nf} \\ -\mathbf{M}^{-1}(\mathbf{q}_{nf}) \mathbf{N}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}(\mathbf{q}_{nf})^{-1} \end{bmatrix} \mathbf{u}, \quad (2.10)$$

with

- $\mathbf{x} = [\mathbf{q}_{nf}^T \quad \dot{\mathbf{q}}_{nf}^T]^T$  is the  $(2n \times 1)$  state vector;
- $\mathbf{u} = \boldsymbol{\tau}_{idm}$  is the  $(n \times 1)$  input vector.

According to (2.10), the DDM is a nonlinear relation of the states and the dynamic parameters included in  $\mathbf{M}$  and  $\mathbf{N}$ .

The prime purpose of the DDM is to simulate the robot. With known input torques and base parameters, the dynamic equations are solved for the joint accelerations and the current state of the robot. We detail here the method developed in [Walker & Orin 1982] that is used in practice. This method has the advantage to use the IDM. Therefore, an explicit calculation of the DDM is not required. With the subscript  $s$  designating the simulated signals, the method is divided in three steps:

- Calculation of  $\mathbf{N}(\mathbf{q}_s, \dot{\mathbf{q}}_s)$ .  
The simulated acceleration is temporally set to zero,  $\ddot{\mathbf{q}}_s = \mathbf{0}_{n \times 1}$ . such that  $\mathbf{N}(\mathbf{q}_s, \dot{\mathbf{q}}_s) = \boldsymbol{\tau}_s = \boldsymbol{\phi}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \mathbf{0}_{n \times 1}) \boldsymbol{\theta}$ , according to (2.8) and (2.9).
- Calculation of  $\mathbf{M}(\mathbf{q}_{nf})$ .  
In order to set  $\mathbf{N}(\mathbf{q}_s, \dot{\mathbf{q}}_s)$  to zero, we fix:
  - the simulated velocity:  $\dot{\mathbf{q}}_s = \mathbf{0}_{n \times 1}$ ;
  - the gravity:  $g = 0$ ;
  - the Coulomb friction coefficients and the offsets, for each  $j = 1, \dots, n$ :  
 $F_{c_j} = 0$  and  $\tau_{off_j} = 0$ .

The position vector  $\mathbf{q}_s$  cannot be set to zero since it appears in the inertia matrix. We define  $\boldsymbol{\alpha}^j$  that is a  $(n \times 1)$  vector with the  $j^{\text{th}}$  element equal to 1 and 0 everywhere else. Each column  $j$  of the matrix  $\mathbf{M}$  is calculated separately by setting  $\ddot{\mathbf{q}}_s = \boldsymbol{\alpha}^j$  and evaluating  $\mathbf{M}(:, j) = \mathbf{M}(\mathbf{q}_{nf}) \boldsymbol{\alpha}^j = \boldsymbol{\phi}(\mathbf{q}_s, \mathbf{0}_{n \times 1}, \boldsymbol{\alpha}^j) \boldsymbol{\theta}$ .

- Solution of the linear equation (2.9) by taking  $\boldsymbol{\tau}_{idm}$  equal to the input of the simulated system.

Section 3.3.3 provide more details on the simulation process and especially on integration solver.

## 2.2 Robot systems architecture

In this thesis, we study the identification of the robots dynamical models. However, since they operate in closed-loop, the robots can be viewed as a global system regrouping: a controller, actuators, sensors and the robot arm. The robot arm models have been described in section 2.1, the remaining elements of the robotic systems are described here. Figure 2.3 provides an illustration of the architecture and defines the signals involved.

### 2.2.1 Robot control laws

Robots operate in closed-loop due to their double integrator behaviour. This is the reason why are controlled in position with two nested loops: the inner-loop for the current control and the outer-loop for the position control. The control

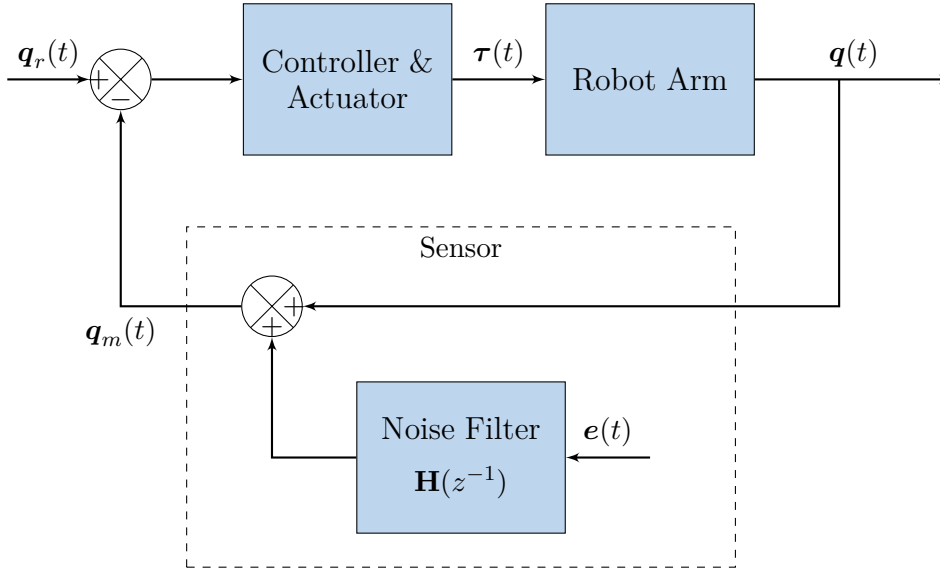


Figure 2.3: Block diagram of the robots architecture

laws in position mainly used are proportional-derivative (PD) and proportional-integral-derivative (PID), but also computed torque (flatness control) and passive control. When identifying the base parameters, the PD control is preferred to the others because it is easy to tune and an excellent tracking is not necessary [Gautier *et al.* 2013a]. Practically, the integral term is weak since the proportional action is usually sufficient. Furthermore, when the position error is too small, it may cause oscillation due to the Coulomb frictions. According to [Samson 1983, Arimoto 1989], this type of control is acceptable if there are high-gear transmission ratios, if the velocity range of the robot is low and if there are high position gains.

When fast motion and high accuracy are required, the performance of the control laws must be improved. The computed torque and passivity-based control have been developed for that purpose. The computed torque is based on the flatness control theory [Fliess *et al.* 1995] and the passive control can be approached with [Landau & Horowitz 1988] and [Berghuis & Nijmeijer 1993]. Those methodologies require the on-line computation of the IDM as well as the identified base parameters. That explains the need of an accurate estimation of those parameters. Those advanced control schemes also rely on the measurement of the joint positions and velocities. The position sensors are reliable, as it is explained in section 2.2.3, but the tachometers (velocity sensors) furnish noisy signals. To overcome this drawback, nonlinear velocity observers have been developed by [Nicosia & Tomei 1990, Canudas de Wit *et al.* 1992] for instance.

In this thesis, unless otherwise indicated, the controller is assumed to be linear and given by the  $(n \times n)$  matrix such as

$$\boldsymbol{\nu}_\tau(t) = \mathbf{C}(p)(\mathbf{q}_r(t) - \mathbf{q}_m(t)), \quad (2.11)$$

where  $p = d/dt$  is the differential operator,  $\boldsymbol{\nu}_\tau$  is the  $(n \times 1)$  vector of control signals,  $\mathbf{q}_r$  is the  $(n \times 1)$  vector of reference trajectories and  $\mathbf{q}_m$  is the  $(n \times 1)$  vector of measured positions. In the remainder of this thesis,  $\mathbf{C}$  designates the controller and not the Coriolis matrix. For convenience, the controller is modelled as a Continuous-Time (CT) system, although in practice it is implemented in Discrete Time (DT) on the micro-controllers that are used to perform the controls. Since each axis of the robot arm is controlled separately, the matrix  $\mathbf{C}$  can be assumed diagonal. Hence, for each link  $j$ , there is

$$\nu_{\tau_j}(t) = C_j(p)(q_{r_j}(t) - q_{m_j}(t)). \quad (2.12)$$

As explained in [Arimoto 1989] for instance, the controller at each joint can be designed independently because the nonlinear cross-coupling effects are smaller than the dynamics of each individual axis. That is due to the mechanical design of the robot arms.

### 2.2.2 Robot actuators

Rigid industrial robots are steered by current driven electrical actuators, also called drive trains. Those actuators encompass a current-controlled voltage source amplifier, a motor (permanent magnet DC or brushless) and a gear train. There is one drive train per link. For further information about the design of drive trains, see for instance [Pasch & Seering 1984]. The voltage source amplifiers are current-controlled with a proportional-integral (PI) law. The current-loop usually has a bandwidth greater than 500Hz. Then, within the frequency range of body dynamics (usually less than 10Hz), its transfer function is modelled as a static gain  $k_{aj}$  for link  $j$  [Gautier & Briot 2012b]. Considering link  $j$ , the motor has a torque constant  $k_{tj}$  and the gear ratio is  $N_j$ . At last, the joint torques are connected with the control signals by the following relation

$$\boldsymbol{\tau}(t) = \mathbf{G}_\tau \boldsymbol{\nu}_\tau(t), \quad (2.13)$$

where  $\mathbf{G}_\tau$  is the  $(n \times n)$  diagonal matrix of joint drive gains and  $\boldsymbol{\nu}_\tau$  is the  $(n \times 1)$  vector of the currents serving as a references for the current amplifiers. The matrix  $\mathbf{G}_\tau$  is diagonal, since there is one actuator per link, and the diagonal components are given by

$$G_{\tau_j} = k_{aj}k_{tj}N_j. \quad (2.14)$$

Those actuators parameters have *a priori* values given by the manufacturers which can be checked with special tests; see e.g. [Gautier & Briot 2012b]. In this thesis, we consider that those static gains are already available.

### 2.2.3 Position sensors

Many technologies exist to sense robot position. For industrial robot arms, there are two main possibilities: a *resolver* or an *encoder* [Warnecke *et al.* Y Nof 1999].

A resolver is a rotary transformer which provides an analog output signal. It is typically composed of a rotor carrying the primary winding and a stator carrying two secondary windings. When an AC voltage is applied to the primary winding, it induces current in the stator windings. The amplitudes of the voltages in the stator windings vary as the sine and cosine of the angular position. By comparing the two signals, the angular position is retrieved. In practice, for a better accuracy, many pairs of stator windings can be considered.

An encoder can measure linear or angular displacements. It can be either incremental or absolute. An incremental encoder records just changes in position whereas an absolute encoder records an absolute position. In other words, it does not need a reference run after switching on. The technology can be either magnetic, conductive or optical. According to [Warnecke *et al.* Y Nof 1999], only optical encoders are considered in industrial robotics (incremental or absolute). In optical encoders, light from LEDs passes through a code disk fixed to the robot link  $j$ . The code disk is composed of parallel tracks of binary patterns made of opaque and transparent segments. On the other side, fixed on link  $j - 1$ , photovoltaic diodes read the optical pattern which results of the disc's position.

During this thesis, we mainly study the Stäubli TX40 robot arm that uses encoders for position control [Stäubli Favergues 2015]. Therefore, we will focus on encoder technology for the noise analysis. Introducing the measurement noise, we define the  $(n \times 1)$  vector of measured joint positions as

$$\mathbf{q}_m(t) = \mathbf{q}(t) + \tilde{\mathbf{q}}(t) = \mathbf{q}(t) + \mathbf{H}(z^{-1})\mathbf{e}(t), \quad (2.15)$$

where  $\tilde{\mathbf{q}}$  is the measurement noise,  $\mathbf{H}$  is the  $(n \times n)$  output noise matrix with  $z^{-1}$  the delay operator,  $\mathbf{q}$  is the  $(n \times 1)$  vector of joint positions and  $\mathbf{e}$  is a  $(n \times 1)$  vector of white noises, with zero means and  $(n \times n)$  covariance matrix  $\mathbf{\Lambda}$ . For the same reasons as [Gilson *et al.* 2008], in this thesis, the noise filters are considered as DT systems. Those reasons are:

- The discrete modelling is more practical than purely stochastic CT noise models;
- The main function of the noise modelling is to improve the statistical efficiency of the estimated parameters, which can be adequately achieved with such filters.

Since there is one independent sensor per link,  $\mathbf{H}$  is diagonal and composed of filters  $H_j(z^{-1})$  with  $j$  from 1 to  $n$ . Furthermore, the noises contained in the  $(n \times 1)$  vector  $\mathbf{e}$  are uncorrelated and the covariance matrix  $\mathbf{\Lambda}$  is also diagonal, with a covariance noted  $\lambda_j$  for the link  $j$ . It comes out for each link  $j$

$$q_{m_j} = q_j + \tilde{q}_j = q_j + H_j(z^{-1})e_j. \quad (2.16)$$

According to [Bélanger *et al.* 1998], a shaft encoder has a white, zero mean and uniformly distributed noise with a variance equal to  $\frac{1}{3}\Delta_e^2$ , where  $\Delta_e$  is the

encoder resolution. In [Swevers *et al.* 2007], the authors have pointed out that, in a factory environment, the position sensors can be influenced by other machines like welding apparatus and other electromagnetic disturbances. For this reason, we consider a more general case where the noise is not necessary white especially at high frequency. The limit between low and high frequencies can be defined by the cut-off frequency of the closed-loop position:  $\omega_{dyn}$ . A covariance proportional to  $\Delta_e^2$  seems reasonable in the operating range of system, i.e. below  $\omega_{dyn}$ . With respect to [Marcassus *et al.* 2007], the angular resolution of the Stäubli TX40 robot is  $2 \cdot 10^{-4}$  degree per count. That order of magnitude shows that the spectral density of the noise is really low below  $\omega_{dyn}$ . Concerning the high frequencies, beyond  $\omega_{dyn}$ , we assume that the spectral density can vary without any specific assumption.

#### 2.2.4 Inverse dynamic identification model

Because of perturbations coming from measurement noise and modelling errors, the actual torque  $\boldsymbol{\tau}$  differs from  $\boldsymbol{\tau}_{idm}$  by an error  $\boldsymbol{v}$ . This usual definition of the Inverse Dynamic Identification Model (IDIM) is given by

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_{idm}(t) + \boldsymbol{v}(t) = \boldsymbol{\phi} \left( \boldsymbol{q}_{nf}(t), \dot{\boldsymbol{q}}_{nf}(t), \ddot{\boldsymbol{q}}_{nf}(t) \right) \boldsymbol{\theta} + \boldsymbol{v}(t). \quad (2.17)$$

Sections 2.3.3 and 6.1.2 provide further information on the model of this error.

### 2.3 Robot arms identification methods

As shown by (2.17), the IDIM is linear with respect to the parameters. Therefore, methods relying on this linearity have been considered in the first place for robot identification. The aim of this section is to present the usual methods for robot identification:

- the Least-Squares (LS) method, based on the IDIM, referred to as IDIM-LS;
- the Instrumental Variable (IV) method, based on the IDIM, referred to as IDIM-IV;
- the method based on the Direct and Inverse Dynamic Identification Model (DIDIM).

The IDIM-LS method has a long history in robot identification and is still considered as the reference method [Khalil & Dombre 2004]. The DIDIM and IDIM-IV methods have been introduced recently in [Gautier *et al.* 2013a] and [Janot *et al.* 2014b] respectively. As explained in section 1.2, there exist many other identification methods; see e.g. [Urrea & Pascal 2016] for a recent comparison of some of them. In this thesis, we focus on the three methods listed above. Before presenting the methods, we detail the filtering process required to deal with robots data in order to obtain all the signals included in the mathematical relations.

### 2.3.1 Filtering methodology

In most applications, the available information is the  $(n \times 1)$  measurement vector of the joint positions,  $\mathbf{q}_m$  defined by (2.15). The joint velocities and accelerations have to be retrieved from this information in order to build the observation matrix  $\phi$  as described in [Gautier 1997].  $\mathbf{q}_m$  is firstly filtered to obtain  $\hat{\mathbf{q}}$ . From this filtered position, the derivatives can be calculated with finite differences while avoiding noise amplification. The filter type and the cut-off frequency,  $\omega_{f_q}$ , are selected such as  $(\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}}) \approx (\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}, \ddot{\mathbf{q}}_{nf})$  in the range  $[0, \omega_{f_q}]$ . The filter, which is usually a Butterworth one, is applied in both forward and reverse directions to avoid lag introduction. The signals are indeed used thereafter to construct the nonlinear basis functions and those nonlinearities do not tolerate any phase shift. The rule of thumb for the cut-off frequency is  $\omega_{f_q} \geq 5 \cdot \omega_{dyn}$ . The combination of the Butterworth filter and the central differentiation is referred to as the *bandpass* filtering process.

In practice, the torque is perturbed by high-frequency ripples: unmodelled friction and flexibility effects, which are rejected by the controller. Those ripples are removed prior to the identification with a low-pass filtering of each basis function and the torque, at the cut-off frequency  $\omega_{F_p} \geq 2 \cdot \omega_{dyn}$ . Since there is no more useful information beyond the cut-off frequency, the data are also re-sampled by keeping one sample over  $n_d$ . This combination of parallel filtering and re-sampling is referred to as the *decimate* process. After data acquisition and parallel filtering, we obtain:

$$\boldsymbol{\tau}_{F_p}(t) = F_p(z^{-1})\boldsymbol{\tau}(t) = \phi_{F_p}(\hat{\mathbf{q}}(t), \hat{\dot{\mathbf{q}}}(t), \hat{\ddot{\mathbf{q}}}(t)) \boldsymbol{\theta} + \mathbf{v}_{F_p}(t), \quad (2.18)$$

with  $F_p$  the *parallel*<sup>1</sup> filter applied to each element of the observation matrix and the error vector such as:

$$\begin{aligned} \phi_{F_p}(\hat{\mathbf{q}}(t), \hat{\dot{\mathbf{q}}}(t), \hat{\ddot{\mathbf{q}}}(t)) &= F_p(z^{-1})\phi(\hat{\mathbf{q}}(t), \hat{\dot{\mathbf{q}}}(t), \hat{\ddot{\mathbf{q}}}(t)) \\ \mathbf{v}_{F_p}(t) &= F_p(z^{-1})\mathbf{v}(t). \end{aligned}$$

Those rules are thoroughly studied and developed in [Pham *et al.* 2001, Pham 2002].

### 2.3.2 The least-squares method

If  $n_m$  measurements are recorded during the experiment, after the re-sampling we have  $N = n_m/n_d$  available sets of data. From (2.18), there is an overdetermined linear system which can be solved thanks to Ordinary LS (OLS):

$$\hat{\boldsymbol{\theta}}_{OLS}(N) = \left[ \frac{1}{N} \sum_{i=1}^N \phi_{F_p}^T(t_{(i)}) \phi_{F_p}(t_{(i)}) \right]^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \phi_{F_p}^T(t_{(i)}) \boldsymbol{\tau}_{F_p}(t_{(i)}) \right], \quad (2.19)$$

<sup>1</sup>The term parallel is used here with respect to the literature, but it does not mean it is applied in an on-line manner.



with  $t_{(i)} = t_{i \cdot n_d} = t_0 + i \cdot n_d / f_m$ , where  $t_0$  and  $f_m$  are respectively the initial time and the recording frequency. In the following of this thesis, that will be simplified by noting just  $t_i$  instead of  $t_{(i)}$ . The reader has to be aware of this feature if a decimate filter is involved. Without modelling errors, the LS estimator is consistent under the two conditions:

- $\bar{E} \left[ \phi_{F_p}^T(t) \phi_{F_p}(t) \right]$  is full column rank;
- $\bar{E} \left[ \phi_{F_p}^T(t) \mathbf{v}_{F_p}(t) \right] = 0$ .

The notation  $\bar{E}[f(t)] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbf{E}[f(t_i)]$ , with  $\mathbf{E}$  the mathematical expectation, comes from the Prediction Error Framework (PEM), see [Ljung 1999]. For closed-loop systems, the assumption that the observation matrix is not correlated with the error is not valid due to the feedback [Van den Hof 1998]. In practice, thanks to the appropriate filtering, the IDIM-LS predictor is still consistent provided that  $\omega_{f_q}$  and  $\omega_{F_p}$  are tuned accordingly to  $\omega_{dyn}$ .

In practice, the LS estimation can be computed thanks to a regrouped matrix formulation, which we may also call *en-bloc* formulation. The IDIM is re-written

$$\mathbf{y}(\boldsymbol{\tau}) = \mathbf{X} \left( \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}} \right) \boldsymbol{\theta} + \boldsymbol{\varepsilon} \quad (2.20)$$

where

- $\mathbf{y}(\boldsymbol{\tau})$  is the  $(r \times 1)$  measurements vector built from the filtered torques  $\boldsymbol{\tau}_{F_p}$ ;
- $\mathbf{X} \left( \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}} \right)$  is the  $(r \times b)$  regrouped observation matrix;
- $\boldsymbol{\varepsilon}$  is the  $(r \times 1)$  vector of errors terms;
- $r = n \cdot N$  is the number of rows in (2.20).

In  $\mathbf{y}$  and  $\mathbf{X}$ , the equations of each joint  $j$  are regrouped together. Thus,  $\mathbf{y}$  and  $\mathbf{X}$  are partitioned so that

$$\mathbf{y}(\boldsymbol{\tau}) = \begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^n \end{bmatrix}, \quad \mathbf{X} \left( \hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}} \right) = \begin{bmatrix} \mathbf{X}^1 \\ \vdots \\ \mathbf{X}^n \end{bmatrix}, \quad (2.21)$$

with

- $\mathbf{y}^j = \begin{bmatrix} \tau_{F_p}^j(t_1) \\ \vdots \\ \tau_{F_p}^j(t_N) \end{bmatrix}$ ;
- $\mathbf{X}^j = \begin{bmatrix} \phi_{F_p}^j \left( \hat{\mathbf{q}}(t_1), \hat{\dot{\mathbf{q}}}(t_1), \hat{\ddot{\mathbf{q}}}(t_1) \right) \\ \vdots \\ \phi_{F_p}^j \left( \hat{\mathbf{q}}(t_N), \hat{\dot{\mathbf{q}}}(t_N), \hat{\ddot{\mathbf{q}}}(t_N) \right) \end{bmatrix}$ ;

- $\phi_{F_p}^j(\hat{\mathbf{q}}(t_k), \hat{\mathbf{q}}(t_k), \hat{\mathbf{q}}(t_k))$  is the  $j^{\text{th}}$  row of the  $(n \times b)$  filtered observation matrix at time  $t_k$  ( $k$  between 1 and  $N$ ).

With the *en-bloc* matrix formulation, the Ordinary LS (OLS) estimates are computed with (2.22). The solution exists if  $(\mathbf{X}^T \mathbf{X})$  is invertible. That is to say that  $\mathbf{X}$  is full column rank.

$$\hat{\boldsymbol{\theta}}_{OLS}(N) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}(\boldsymbol{\tau}) \quad (2.22)$$

Alternatively, the parameters can be estimated with the Weighted LS (WLS) solution:

$$\hat{\boldsymbol{\theta}}_{WLS}(N) = (\mathbf{X}^T \hat{\boldsymbol{\Omega}}_{\tau}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \hat{\boldsymbol{\Omega}}_{\tau}^{-1} \mathbf{y}(\boldsymbol{\tau}), \quad (2.23)$$

with  $\boldsymbol{\Omega}_{\tau}$  defined such as:

$$\boldsymbol{\Omega}_{\tau} = \text{diag}(\sigma_1^2 \mathbf{I}_N, \dots, \sigma_j^2 \mathbf{I}_N, \dots, \sigma_n^2 \mathbf{I}_N) \quad (2.24)$$

where  $\mathbf{I}_N$  is the  $(N \times N)$  identity matrix and  $\sigma_j^2$  is the noise variance of link  $j$ . This matrix is constructed from the covariance matrix of the  $\mathbf{v}_{F_p}$  defined by:

$$\boldsymbol{\Lambda}_{\tau} = \text{diag}(\sigma_1^2, \dots, \sigma_j^2, \dots, \sigma_n^2). \quad (2.25)$$

In other words, the noise  $\mathbf{v}_{F_p}$  is assumed to have zero mean, to be serially uncorrelated and to be homoskedastic; i.e. a white noise. In practice, the covariance is estimated with

$$\hat{\boldsymbol{\Lambda}}_{\tau} = \frac{1}{N} \sum_{i=1}^N \left( \boldsymbol{\tau}_{F_p}(t_i) - \boldsymbol{\phi}_{F_p}^T(t_i) \hat{\boldsymbol{\theta}}_{OLS} \right) \left( \boldsymbol{\tau}_{F_p}(t_i) - \boldsymbol{\phi}_{F_p}^T(t_i) \hat{\boldsymbol{\theta}}_{OLS} \right)^T. \quad (2.26)$$

Finally, the covariance matrix of the LS estimates is

$$\boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}_{LS}) = (\mathbf{X}^T \hat{\boldsymbol{\Omega}}_{\tau}^{-1} \mathbf{X})^{-1}. \quad (2.27)$$

### 2.3.3 The instrumental variable method

Another well-known technique for linear estimation is the IV method which is suitable for system identification in closed-loop. We give here some elements of the extended IV theory in a general framework, based on [Söderström & Stoica 1983], before explaining how it is employed for robot identification.

### The Extended IV theory

The extended-IV estimator is given by

$$\hat{\boldsymbol{\theta}}_{IV}(N) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\| \left[ \frac{1}{N} \sum_{i=1}^N \boldsymbol{\zeta}^T(t_i) \mathbf{L}(z^{-1}) \boldsymbol{\phi}(t_i) \right] \boldsymbol{\theta} - \left[ \frac{1}{N} \sum_{i=1}^N \boldsymbol{\zeta}^T(t_i) \mathbf{L}(z^{-1}) \boldsymbol{\tau}(t_i) \right] \right\|_{\mathbf{W}}^2 \quad (2.28)$$

where  $\boldsymbol{\zeta}$  is the  $(n \times b)$  instrument matrix,  $\mathbf{L}$  is a  $(n \times n)$  matrix of prefilters and  $\mathbf{W}$  is a  $(n \times n)$  positive-definite weighting matrix. For the rest of this thesis, the weighting matrix will be the identity. If there is no modelling errors, the extended-IV is consistent under the two conditions:

- $\bar{E} \left[ \boldsymbol{\zeta}^T(t) \boldsymbol{\phi}_{\mathbf{L}}(t) \right]$  is full column rank;
- $\bar{E} \left[ \boldsymbol{\zeta}^T(t) \mathbf{v}_{\mathbf{L}}(t) \right] = 0$ .

The first condition means that the instrumental matrix must be well correlated with the observation one. This condition can be called the *instrument relevance* [Wooldridge 2008]. The second condition expresses the fact that the instrumental matrix must be uncorrelated with the error, which is known as the *instrument exogeneity*. Assuming no modelling error, the vector  $\mathbf{v}$  defined in (2.17) contains only measurement noises, such as

$$\mathbf{v}(t) = \mathbf{H}_{\tau}(z^{-1}) \mathbf{e}(t), \quad (2.29)$$

where  $\mathbf{e}$  is a  $(n \times 1)$  vector of white noises, with zero means and  $(n \times n)$  covariance matrix  $\boldsymbol{\Lambda}$ .  $\mathbf{H}_{\tau}$  is the  $(n \times n)$  matrix filter modelling the input noise, assumed to be asymptotically stable and invertible. In [Söderström & Stoica 1988], Chapter 8, the authors showed that the optimal variance is reached with

$$\mathbf{L}(z^{-1}) = \boldsymbol{\Lambda}^{-1} \mathbf{H}_{\tau}^{-1}(z^{-1}) \quad \boldsymbol{\zeta}(t) = \mathbf{L}(z^{-1}) \boldsymbol{\phi}_{nf}(t). \quad (2.30)$$

The optimal covariance matrix (i.e. the lower bound) is given by

$$\boldsymbol{\Sigma}^{opt} = \left\{ \bar{E} \left[ \left[ \mathbf{H}_{\tau}^{-1}(z^{-1}) \boldsymbol{\phi}_{nf}(t) \right]^T \boldsymbol{\Lambda}^{-1} \left[ \mathbf{H}_{\tau}^{-1}(z^{-1}) \boldsymbol{\phi}_{nf}(t) \right] \right] \right\}^{-1}. \quad (2.31)$$

The main question with this methodology is the choice of the instruments to estimate  $\boldsymbol{\phi}_{nf}$ . That topic was widely studied in automatic control, see e.g. [Söderström & Stoica 1983] and the references given therein. We are going to see now how the problem is tackled in robot identification.

### The IDIM-IV method

Based on [Young 2011], in [Janot *et al.* 2014c], the authors have shown that the simulation of the DDM provides a very convenient way to obtain the instruments for robot identification. This simulation model contains the whole closed-loop and

is referred to as the *auxiliary model*. From the simulation of this auxiliary model, noise-free simulated signals are retrieved and used to construct the instrument matrix. The signals are noise-free since the only input is the reference trajectory which is perfectly known. The process is iterative because the simulation is based on the parameters previously identified. By noting the simulated signals with a subscript  $s$ , the instrumental matrix is  $\zeta(t) = F_p(z^{-1})\phi(\mathbf{q}_s(t), \dot{\mathbf{q}}_s(t), \ddot{\mathbf{q}}_s(t))$ . The instrumental matrix can be viewed as an estimation of the noise-free part of the observation matrix. This IDIM-IV method also includes the decimate filter, i.e.  $\mathbf{L}(z^{-1}) \leftarrow F_p(z^{-1})\mathbf{I}_n$ , with  $\mathbf{I}_n$  the  $(n \times n)$  identity matrix. The fact is that the IDIM-IV method does not take into account the noise model to provide optimal estimates.

The  $(r \times b)$  instrumental matrix is constructed such as

$$\mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it}) = \begin{bmatrix} \mathbf{Z}_{IV}^{it,1} \\ \vdots \\ \mathbf{Z}_{IV}^{it,n} \end{bmatrix}, \quad (2.32)$$

where  $\mathbf{Z}_{IV}^{it,j} = \begin{bmatrix} \zeta^j(t_1, \hat{\boldsymbol{\theta}}_{IV}^{it}) \\ \vdots \\ \zeta^j(t_N, \hat{\boldsymbol{\theta}}_{IV}^{it}) \end{bmatrix}$ . At iteration  $it$ , the new estimated parameters are computed with

$$\hat{\boldsymbol{\theta}}_{IV}^{it+1} = \left[ \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{X} \right]^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{y}(\tau), \quad (2.33)$$

with  $\mathbf{X}$  given by (2.21). Alternatively, a weighted IV can be performed similarly to the WLS method. As explained in [Janot *et al.* 2014c] and described in Figure 2.4, the process is iterated until its convergence. The convergence criterion is based on the relative variation of the estimated parameters and the one of the estimation error:  $\boldsymbol{\varepsilon}^{it} = \mathbf{y}(\tau) - \mathbf{X}\hat{\boldsymbol{\theta}}_{IV}^{it}$ . Concerning the initialisation, the inertia parameters are usually initialised with Computer-Aided Design (CAD) values, whereas the others parameters are set to zero. After convergence (superscript  $cv$ ), the covariance matrix of the IV estimates is given by:

$$\boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}_{IV}^{cv}) = \left( \mathbf{Z}^T(\hat{\boldsymbol{\theta}}_{IV}^{cv}) \hat{\boldsymbol{\Omega}}_{\tau}^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{cv}) \right)^{-1}. \quad (2.34)$$

### 2.3.4 The DIDIM method

Recently [Gautier *et al.* 2013a] has introduced a method based on the simulation of the DDM. This method, called DIDIM for Direct and Inverse Identification Model, minimizes the squared difference between the actual torques and the simulated ones. In other words, there is no observation matrix, the predicted signal comes only from the simulation. The considered signal is the input torque and, at time  $t_i$ , the error

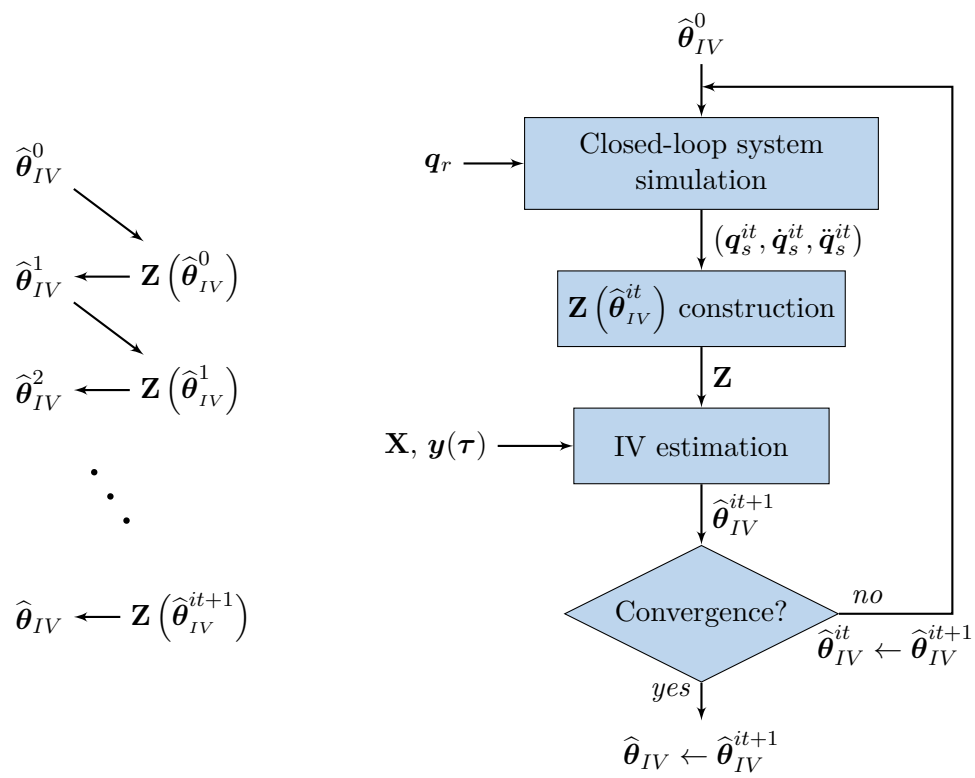


Figure 2.4: Diagram of the IDIM-IV method

is given by:

$$\begin{aligned}\varepsilon_{didim}(t_i, \boldsymbol{\theta}) &= \boldsymbol{\tau}_{F_p}(t_i) - \boldsymbol{\tau}_s(t_i, \boldsymbol{\theta}) \\ &= \boldsymbol{\tau}_{F_p}(t_i) - \boldsymbol{\phi}_{F_p}(\mathbf{q}_s(t_i), \dot{\mathbf{q}}_s(t_i), \ddot{\mathbf{q}}_s(t_i)) \boldsymbol{\theta},\end{aligned}\quad (2.35)$$

where  $\boldsymbol{\tau}_s(t_i, \boldsymbol{\theta})$  is the  $(n \times 1)$  simulated torques vector. The vectors  $\mathbf{q}_s$ ,  $\dot{\mathbf{q}}_s$  and  $\ddot{\mathbf{q}}_s$  contain respectively the angular positions, velocities and accelerations of robot joints, coming from the simulation of the auxiliary model. Hence, the knowledge of the controller is required for the simulation. Since the input of the simulation,  $\mathbf{q}_r$ , is perfectly known (i.e. noise free),  $\boldsymbol{\tau}_s$  is not correlated with the measurement noise  $\tilde{\mathbf{q}}$ . That insures the consistency of the estimation, assuming no modelling error.

Furthermore, if the dependence of  $\boldsymbol{\phi}$  in  $\boldsymbol{\theta}$  is neglected, the optimisation is greatly enhanced. In fact, the gradient of the simulated torque with respect to the estimated parameters is just  $\boldsymbol{\phi}(\mathbf{q}_s(t_i), \dot{\mathbf{q}}_s(t_i), \ddot{\mathbf{q}}_s(t_i))$ . In the field of system identification, this technique is called Pseudo-Linear Regression (PLR), see Eq. (7.112) in [Ljung 1999]. According to the same reference, PLR is derived from [Solo 1979]. Since the DIDIM relies on this assumption, the parameters are iteratively estimated thanks to the linear Least-Squares:

$$\hat{\boldsymbol{\theta}}_{DIDIM}^{it+1} = \left[ \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it})^T \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it}) \right]^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it})^T \mathbf{y}(\tau), \quad (2.36)$$

with  $\mathbf{Z}$  and  $\mathbf{y}$  constructed accordingly to (2.21) and (2.32). Like the IDIM-IV method, the DIDIM one is iterative. In practice, they can share the same initialisation and the same convergence criterion. A complete description of the method is available in [Janot 2007]. In addition, [Gautier *et al.* 2011, Robet *et al.* 2012] provide some applications examples. After convergence, the covariance matrix of the DIDIM estimates is given by:

$$\boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}_{DIDIM}^{cv}) = \left( \mathbf{Z}^T(\hat{\boldsymbol{\theta}}_{DIDIM}^{cv}) \hat{\boldsymbol{\Omega}}_{\tau}^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{cv}) \right)^{-1}. \quad (2.37)$$

### 2.3.5 Exciting trajectories

To finish this section about robots identification, we shortly address the issue of exciting trajectories. The idea is to insure that  $\mathbf{X}$ , or  $\mathbf{Z}$ , is full column rank and has a good condition number. Section 4.1.2 provide further information on the conditioning number and its influence on the estimation solution. First of all, as explained in [Walter & Pronzato 1994] for instance, (2.22) should not be applied as it stands. The singular values decomposition is a better alternative from a computation point of view, especially if the conditioning is not perfect. However, an effective algorithm is not enough to insure a good estimation. The experimental data must contain enough information. With the vocabulary of the system identification community, the robot trajectory must be *persistently exciting*. [Söderström & Stoica 1988, Ljung 1999] give a mathematical definition of a persistent excitation and [Gautier & Khalil 1992] presents a method dedicated to the optimal trajectory

design for robot identification.

The principle of the method develop by [Gautier & Khalil 1992] is to find a sequence of optimal points with respect to a given criterion, the conditioning number of  $\mathbf{X}$  for example. Thereafter, a continuous and smooth trajectory is calculated by interpolating a function at the optimal points. This interpolating function can be a time polynomial. Alternatively, special test motions can be designed to excite some parameters specifically and/or sequentially; see e.g. [Vandanjon *et al.* 1995, Janot *et al.* 2007].

To illustrate physically a counterexample of an exciting trajectories, we can think about a basic trajectory where the arm  $j$  moves always in the same direction, i.e. its velocity has a constant sign. In this case, the regressor of the Coulomb friction is constantly  $+1$ , or  $-1$  depending on the direction, and the regressor of the offset is always 1. Therefore, both regressors are linearly dependent and consequently  $\mathbf{X}$  is column rank deficient.

## 2.4 Concluding remarks

In this chapter, the notations for robot modelling have been introduced. The geometry of rigid serial robots is parametrised with the MDH notation. From the Newton's law, the DDM expresses the joint accelerations as a function of the input torques and the dynamic parameters. Nonetheless, the input torques can be expressed as a linear function of those dynamic parameters with the IDM. The IDIM differs from the IDM in that it takes into account the noise inherent in experimental data. Based on this linear IDIM, three identification methods have been introduced. The reference technique is the IDIM-LS method that relies on a careful prefiltering of the measured data. In addition, the recently introduced IDIM-IV and DIDIM techniques are able to deal with closed-loop data and proved their relevance for robot identification.





# System identification for continuous time systems

---

In this chapter, some techniques from the system identification community are presented. If a lot of research has been done on Discrete Time (DT) system identification, see e.g. chapters 6-7 of [Eykhoff 1974] or [Ljung 1999] and the references given therein, the identification of Continuous Time (CT) systems has also been developed over the years [Young 1966, Young 1981, Unbehauen & Rao 1998, Garnier *et al.* 2008]. The major part of the identification research (DT or CT) has focused on Linear Time Invariant (LTI) systems. As explained in the previous chapter, the industrial robot are not linear with respect to the states. Thus, if all the techniques developed by the identification community cannot be applied directly to industrial robots, they still represent a valuable background that we summarize in this chapter.

An introduction to the issues raised by the closed-loop configuration is proposed in section 3.1. The following section introduces the Refined Instrumental Variable method dedicated to CT- and DT-LTI systems. Section 3.3 outlines the Prediction Error Method for hybrid systems (CT and DT). Concluding remarks are provided in section 3.4.

---

<b>3.1</b>	<b>Closed-loop identification challenges</b>	<b>34</b>
<b>3.2</b>	<b>The refined instrumental variable method</b>	<b>34</b>
3.2.1	The hybrid Box-Jenkins model	34
3.2.2	The hybrid RIVC algorithm	36
3.2.3	Statistical elements of the method	38
<b>3.3</b>	<b>Prediction error methods</b>	<b>39</b>
3.3.1	Prediction error principle	39
3.3.2	DT prediction error method	40
3.3.3	Hybrid output error method	42
3.3.4	Computational aspects	43
<b>3.4</b>	<b>Concluding remarks</b>	<b>44</b>

---

### 3.1 Closed-loop identification challenges

The identification experiments can be performed in closed-loop for miscellaneous reasons. Firstly, the system may be unstable in open-loop. Therefore, it is impossible to operate it safely without a control law. Secondly, it can be impossible to do without the regulator because the system has been bought "off the self" with an integrated controller. One can also think about the case where, although the system is stable in open-loop, the practitioner has to process closed-loop data because he has no influence on the experimental work. At last, in a control design purpose, it can be more relevant to identify the system directly in closed-loop [De Callafon 1998].

An issue with closed-loop identification is that the usual methods for open-loop data fail if they are applied directly without precaution. The reason of this failure is the correlation between the input and the unmeasurable output noise as illustrated in Figure 2.3, see e.g. [Van den Hof 1998]. That issue of the non zero correlation between the noise and the input has already been stressed with the IDIM-LS method in section 2.3.2. The other main issue with closed-loop data is that identifiability may be lost. In this sense, it is impossible to estimate a unique set of parameters from measured input-output data. This is due to lack of sensitivity of the system output with respect to the parameters due to the controller action, as shown for instance in [Janot *et al.* 2014a] for robots.

It is common in the literature to divide closed-loop identification methods in three different types [Gustavsson *et al.* 1977, Söderström & Stoica 1988, Forsell & Ljung 1999]: *direct*, *indirect* and *joint input-output* methods. Each type corresponds to different assumptions concerning the nature of the feedback. According to [Forsell 1999], in the direct approach, no assumption is made on the data generation and the method is applied directly on the measured data. With the indirect approach, the practitioner is perfectly aware of the feedback used during the experiment. From the closed-loop system identified, the open-loop parameters are retrieved using the knowledge of the controller. The joint input-output approach considers the input and the output signals jointly as a global output of an augmented system, driven by the reference signal and the unmeasurable noise. From the identification of this augmented system, the open-loop and the controller models are retrieved.

### 3.2 The refined instrumental variable method

In this section, we give a brief overview of the Refined Instrumental Variable method for Continuous-time (RIVC) systems introduced in [Young & Jakeman 1979]. This method has been developed for LTI systems and is extensively presented in chapters 8-9 of [Young 2011] and in [Young 2015] for instance.

#### 3.2.1 The hybrid Box-Jenkins model

To illustrate the RIVC method, we need to introduce the hybrid Box-Jenkins (BJ) model for SISO systems. This BJ model is called hybrid because it consists of a

CT state model coupled with DT observations. The CT state model is driven by the differential equation

$$\frac{d^{n_a} x(t)}{dt^{n_a}} + a_1 \frac{d^{n_a-1} x(t)}{dt^{n_a-1}} + \dots + a_{n_a} x(t) = b_0 \frac{d^{n_b} u(t)}{dt^{n_b}} + \dots + b_{n_b} u(t), \quad (3.1)$$

where  $x$  is the state and  $u$  is the input signal. In the transfer function form, the CT can be written:

$$x(t) = \frac{B(p)}{A(p)} u(t), \quad (3.2)$$

with  $B(p) = b_0 p^{n_b} + b_1 p^{n_b-1} + \dots + b_{n_b}$ ,  $A(p) = p^{n_a} + a_1 p^{n_a-1} + \dots + a_{n_a}$  and  $p = \frac{d}{dt}$ , the differential operator. The polynomial coefficients are regrouped in the  $(n_a + n_b + 1 \times 1)$  vector:

$$\boldsymbol{\theta} = [a_1 \quad \dots \quad a_{n_a} \quad b_0 \quad \dots \quad b_{n_b}]^T. \quad (3.3)$$

The signals are assumed to be uniformly sampled, with a sampling interval  $T_s$ . The measured output  $y(t_i)$  is expressed as

$$y(t_i) = x(t_i) + \xi(t_i), \quad (3.4)$$

where  $\xi(t_i)$  is an additive measurement noise coloured such as

$$\xi(t_i) = \frac{D_n(z^{-1})}{C_n(z^{-1})} e(t_i), \quad (3.5)$$

with  $e(t_i)$  a Gaussian white noise, i.e.  $e(t_i) \sim \mathcal{N}(0, \sigma^2)$ . Such a noise filter defines an AutoRegressive Moving Average (ARMA) model. It thus assumes that the noise has a rational spectral density.

Based on the normality of the Gaussian noise  $e(t_i)$ , it can be defined the prediction error:

$$\begin{aligned} \varepsilon(t_i) &= \frac{C_n(z^{-1})}{D_n(z^{-1})} \left( y(t_i) - \frac{B(p)}{A(p)} u(t_i) \right) \\ &= \frac{C_n(z^{-1})}{D_n(z^{-1})} \frac{1}{A(p)} (A(p)y(t_i) - B(p)u(t_i)). \end{aligned} \quad (3.6)$$

By defining, the hybrid filter

$$f = \frac{C_n(z^{-1})}{D_n(z^{-1})} \frac{1}{A(p)} = f_d(z^{-1}, \boldsymbol{\eta}) f_c(p, \boldsymbol{\theta}), \quad (3.7)$$

with the  $(l \times 1)$  vector  $\boldsymbol{\eta}$  regrouping the coefficients of  $C_n$  and  $D_n$ . Since the polynomial operators commute:

$$\varepsilon(t_i) = A(p)y_f(t_i) - B(p)u_f(t_i). \quad (3.8)$$

Finally, this can be written as:

$$\begin{aligned} p^{n_a} y_f(t_i) &= -a_1 p^{n_a-1} y_f(t_i) + \dots - a_{n_a} y_f(t_i) + B(p) u_f(t_i) + \varepsilon(t_i) \\ &= \boldsymbol{\phi}_f(t_i) \boldsymbol{\theta} + \varepsilon(t_i), \end{aligned} \quad (3.9)$$

with the  $(1 \times n_a + n_b + 1)$  matrix

$$\boldsymbol{\phi}_f(t_i) = \left[ -p^{n_a-1} y_f(t_i) \quad \dots \quad -y_f(t_i) \quad p^{n_b} u_f(t_i) \quad \dots \quad u_f(t_i) \right].$$

Equation (3.9) is a problem of linear estimation that can be handled with an instrumental variable technique. Nonetheless, it assumes that the hybrid filter is known, which includes  $A(p)$ . Therefore, in practice, the hybrid BJ model is identified with an iterative approach. That iterative algorithm is depicted by Figure 3.1. The coefficients of  $C_n$  and  $D_n$ , regrouped in  $\boldsymbol{\eta}$ , are estimated separately as highlighted with the algorithm in the following section. The filter defined by (3.7) is an hybrid one. In practice, the input signal is firstly filtered by the continuous time implementation of  $1/A(p)$  to obtain the derivatives. In a second phase, those derivatives are sampled and filtered by the inverse noise filter  $C_n(z^{-1})/D_n(z^{-1})$ .

Concerning the derivatives, they are obtained when the signals are filtered by  $f_c$ . For example the derivative of  $y$  at the order  $m < n_a$  is obtained with  $\frac{p^m}{A(p)} y(t)$ . Two elements are noteworthy. Firstly, if  $1/A(p)$  is unstable, it must be stabilized by projection of its eigenvalues. If the system operates in closed-loop, in [Young 2011], the author suggests to use the two steps variants developed in [Young et al. 2009] and summarized in section 5.2.1. Secondly, the derivatives are obtained by filtering the signals with respect to open-loop bandwidth. With the bandpass process, described in section 2.3.1, the measured position is filtered taking into account the closed-loop bandwidth.

With regard to the instruments, they are generated with an auxiliary model similarly to the IDIM-IV method. That also plays into the hands of an iterative process. For a system identified in closed-loop, the auxiliary model must encompass the whole closed-loop system, whereas the transfer  $B(p)/A(p)$  is sufficient in the open-loop case. That point is highlighted in the following section.

### 3.2.2 The hybrid RIVC algorithm

As explained in the previous section, the RIVC identification process is iterative. The sequential steps are summarized here for the open-loop case.

Step 1. **Initialisation:** find an initial estimate of the parameters  $\hat{\boldsymbol{\theta}}_0$  from a sub-optimal identification technique or from *a priori* knowledge on the system.

Step 2. **Recursive-iterative IV estimation:** for  $it = 1$  to convergence

- (1) Obtain the estimate of the noise-free output signal,  $\hat{x}$ , thanks to the auxiliary model based on the parameters estimated at the previous iteration

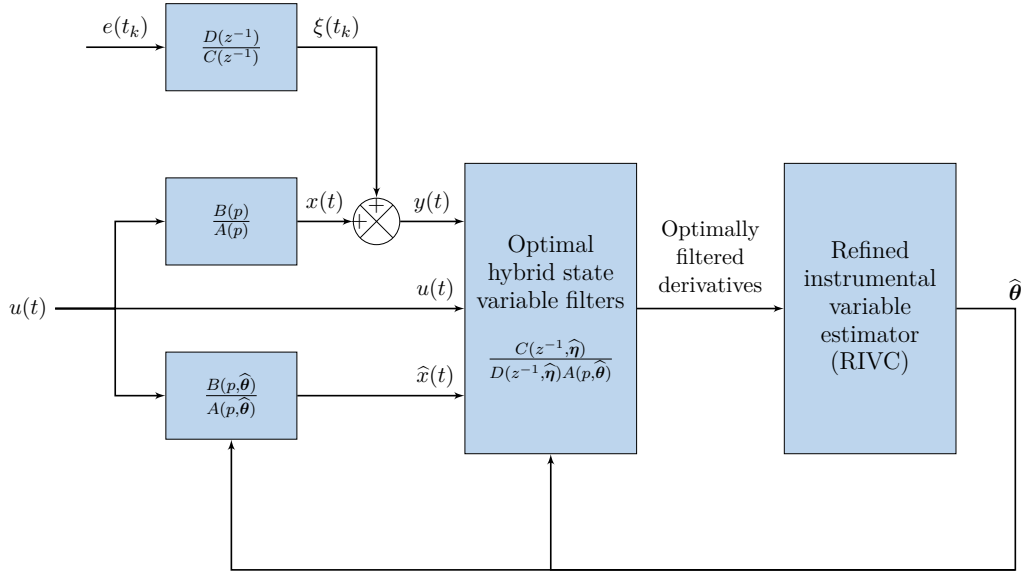


Figure 3.1: Block diagram of the RIVC algorithm for the hybrid BJ model, source [Young 2011]

$\hat{\theta}^{it-1}$ :

$$\hat{x}(t) = \frac{B(p, \hat{\theta}^{it-1})}{A(p, \hat{\theta}^{it-1})} u(t).$$

If the estimated model is unstable, the unstable eigenvalues of the denominator polynomial are reflected into the stable region.

- (2) From the sampled signals and the estimated noise  $\hat{\xi}(t_i) = y(t_i) - \hat{x}(t_i)$ , obtain the estimate of the noise model parameters  $\hat{\eta}^{it}$ . [Young 2006] provides further information about the identification of ARMA noise models.
- (3) Prefilter the input,  $u(t)$ , the output,  $y(t)$  and the estimated output,  $\hat{x}(t)$  by the continuous time filter:

$$f_c(p, \hat{\theta}^{it-1}) = \frac{1}{A(p, \hat{\theta}^{it-1})}.$$

- (4) After signals sampling, prefilter the input and output derivatives by the discrete filter:

$$f_d(z^{-1}, \hat{\eta}^{it}) = \frac{C_n(z^{-1}, \hat{\eta}^{it})}{D_n(z^{-1}, \hat{\eta}^{it})}.$$

- (5) Compute the IV estimate

$$\hat{\theta}^{it} = \left[ \frac{1}{N} \sum_{i=1}^N \zeta_{f_{it}}^T(t_i) \phi_{f_{it}}(t_i) \right]^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \zeta_{f_{it}}^T(t_i) (p^{n_a} y_{f_{it}}(t_i)) \right] \quad (3.10)$$

with

$$\begin{aligned}\phi_{f_{it}}(t_i) &= \begin{bmatrix} -p^{n_a-1}y_{f_{it}}(t_i) & \cdots & -y_{f_{it}}(t_i) & p^{n_b}u_{f_{it}}(t_i) & \cdots & u_{f_{it}}(t_i) \end{bmatrix}, \\ \zeta_{f_{it}}(t_i) &= \begin{bmatrix} -p^{n_a-1}\hat{x}_{f_{it}}(t_i) & \cdots & -\hat{x}_{f_{it}}(t_i) & p^{n_b}u_{f_{it}}(t_i) & \cdots & u_{f_{it}}(t_i) \end{bmatrix}, \\ f_{it} &= f_d(z^{-1}, \hat{\boldsymbol{\eta}}^{it})f_c(s, \hat{\boldsymbol{\theta}}^{it-1}).\end{aligned}$$

Step 3. Computation of the estimated parametric error **covariances matrices**:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \hat{\sigma}^2 \left[ \sum_{i=1}^N \zeta(t_i)^T \zeta(t_i) \right]^{-1}, \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\eta}} = \hat{\sigma}^2 \left[ \sum_{i=1}^N \phi_n(t_i)^T \phi_n(t_i) \right]^{-1}, \quad (3.11)$$

where  $\phi_n$  is the  $(1 \times n_c + n_d + 1)$  observation matrix of the noise model.

The algorithm presented here is the open-loop version of the RIVC method. In the closed-loop case, the signal  $u$  and by construction  $\hat{x}$  are correlated with the measurement noise. Consequently, the instrument exogeneity is not validated anymore. The solution is then to estimate the input,  $u$ , and the output,  $x$ , thanks to the closed-loop relations like with the IDIM-IV method. That changes part (1) of the step 2 in the previous algorithm. The reader can refer to [Gilson *et al.* 2008] for a comprehensive study about the RIVC method for closed-loop systems.

### 3.2.3 Statistical elements of the method

With the algorithm previously described, it appears that the RIVC method relies on the decomposition into two separate identification problems. The theoretical justification comes from the theorem of [Pierce 1972] that is based on the following assumptions:

- the noises of the sequence  $\{e(t_i)\}_{i=1}^N$  are independent and identically distributed with zero mean and variance  $\sigma^2$ ;
- the parameters values are admissible, i.e. the model is stable and identifiable;
- the input sequence  $\{u(t_i)\}_{i=1}^N$  is persistently existing.

As outlined in [Young 2011], the result of [Pierce 1972] validates the RIVC algorithm.

The RIVC method has been developed rather separately from the automatic control literature. However, that can be seen as an application of the extended IV theory summarized in section 2.3.3. Instead of  $\boldsymbol{\tau}$ , the observation variable is the output  $y$  and the prefilter matrix  $\mathbf{L}$  is replaced by  $f$ . In the opposite of (2.30), the noise covariance matrix  $\boldsymbol{\Lambda}$  does not appear in the filter because it is a SISO case and this weighting can be removed from the equations.

### 3.3 Prediction error methods

The prediction error is a wide-spread family of identification methods. Several books deal with those methods: Chapter 7 in [Söderström & Stoica 1988] and also Chapter 7 in [Ljung 1999]. We give here a brief overview of the techniques, staying quite close to the material presented in [Åström 1980, Ljung 1999].

#### 3.3.1 Prediction error principle

The purpose of the Prediction Error Methods (PEM) is to find the best parametric model with respect to a specific criterion. The criterion is a function of the error between the noisy measured output and the predicted model output. Sometimes, the model structure comes from the knowledge of the concerned scientific community, like in robot identification with the Newton's law, and the problem becomes the determination or the estimation of the model's parameters. The generic criterion of PEM is given by:

$$V_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N l\left(\mathbf{L}(z^{-1})\boldsymbol{\varepsilon}(t_i, \boldsymbol{\theta})\right) \quad (3.12)$$

where  $l(\cdot)$  is a scalar function like the euclidean norm,  $\mathbf{L}$  is a  $(n \times n)$  matrix of stable linear filters and  $\boldsymbol{\varepsilon}$  is the  $(n \times 1)$  vector of prediction error, which is function of the measured data. At time  $t_i$ , the prediction vector is given by

$$\boldsymbol{\varepsilon}(t_i|t_m, \boldsymbol{\theta}) = \mathbf{y}(t_i) - \hat{\mathbf{y}}(t_i|t_m, \boldsymbol{\theta}), \quad i \geq m \quad (3.13)$$

where  $\mathbf{y}$  and  $\hat{\mathbf{y}}$  are the  $(n \times 1)$  vectors of the measured and predicted outputs respectively. A *prediction error method* solves the minimization problem:

$$\hat{\boldsymbol{\theta}}(N) = \arg \min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta}). \quad (3.14)$$

It thus appears that the PEM family is large depending on the choice of  $l(\cdot)$ , the choice of the prefilters  $\mathbf{L}$ , the choice of the model structure, the choice of the considered output signals and even the choice of the minimization algorithm. To evaluate the difference between the two outputs many scalar functions may be used, as explained in [Walter & Pronzato 1994]. The general principle of the PEM is illustrated in Figure 3.2, where  $\mathbf{u}$  and  $\mathbf{y}$  are respectively the inputs and the outputs of a the system.

The prediction error formulation is usually presented with the Maximum Likelihood (ML) that requires normality assumption for the observations. It has been shown that the ML estimator is *asymptotically efficient*. That is to say that there is no other unbiased estimator giving a smaller covariance. However, as pointed out by [Åström 1980], in many practical cases the normality assumption is not validated and it is not necessary to postulate (3.12).

To illustrate the wide spectrum of methods encompassed by the definition, we

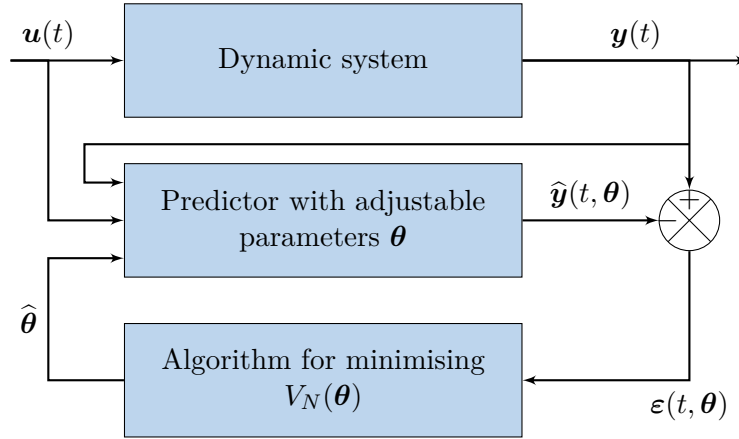


Figure 3.2: Block diagram of the prediction error method

can reconsider the IDIM-LS method. The identification variable  $\mathbf{y}$  are replaced by  $\boldsymbol{\tau}$ . The prediction is given by the IDM:  $\boldsymbol{\tau}_{idm}(t) = \boldsymbol{\phi}(t)\boldsymbol{\theta}$ . The prefilter is the decimate one, such as  $\mathbf{L}(z^{-1}) \leftarrow F_p(z^{-1})\mathbf{I}_n$ . The scalar function is the euclidean norm and the minimization algorithm is the OLS. Finally, the IDIM-LS method belongs to the PEM family.

Regarding the two closed-loop challenges mentioned in section 3.1, the prediction error methods treat them differently depending on the model assumptions. For the noise correlation, if the feedback is unknown, the method falls into the direct approaches and the system can be identified only if the noise model can be described by the true noise properties [Forssell 1999]. If the regulator is known, the PEM can be considered as indirect and thus an exact modelling of the noise is not required. In this case, the prediction is indeed generated with the reference signal that is perfectly known and uncorrelated with the measurement noise; see e.g. [Van Donkelaar & Van den Hof 2000]. For the sensitivity issue, this is usually addressed during the experiment design.

### 3.3.2 DT prediction error method

In the system identification community, the notion of PEM usually refers to the method developed in [Åström & Bohlin 1966] for DT LTI systems that is named *one-step-ahead* PEM. This method has been extended to a *k-step-ahead* prediction in [Åström 1970] and has been extensively analysed by [Ljung 1976, Åström 1980] for instance. We detail here this method considering a SISO DT Box-Jenkins model given by:

$$\begin{aligned} y(t_i) &= \frac{B_d(z^{-1})}{A_d(z^{-1})}u(t_i) + \frac{D_n(z^{-1})}{C_n(z^{-1})}e(t_i) \\ &= G_d(z^{-1})u(t_i) + H(z^{-1})e(t_i), \end{aligned} \quad (3.15)$$



where  $B_d$  and  $A_d$  are respectively the numerator and denominator of the DT transfer function  $G_d$ . Given a data set  $\mathcal{D}_N = \{y(t_i), u(t_i)\}_{i=1}^N$ , the idea of the one step ahead PEM is to predict the output, such as:

$$\begin{aligned}\hat{y}(t_i|t_{i-1}) &= G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i) + \hat{\xi}(t_i|t_{i-1}) \\ &= G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i) + \left(1 - H^{-1}(z^{-1}, \hat{\boldsymbol{\eta}})\right) \xi(t_i) \\ &= G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i) + \left(1 - H^{-1}(z^{-1}, \hat{\boldsymbol{\eta}})\right) \left(y(t_i) - G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i)\right) \\ &= \left(1 - H^{-1}(z^{-1}, \hat{\boldsymbol{\eta}})\right) y(t_i) + H^{-1}(z^{-1}, \hat{\boldsymbol{\eta}})G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i).\end{aligned}\tag{3.16}$$

The details of the one-step-ahead prediction of  $\hat{\xi}$  are available in Chapter 3 of [Ljung 1999]. In the final relation of (3.16), the first term on the right-hand-side is only function of  $y(t_m)$ , with  $m < i$ . The predicted output is thus independent of the current one. That expression requires that the filters  $H^{-1}G_d$  and  $H^{-1}$  are stable. From the output prediction (3.16), the error can be written:

$$\begin{aligned}\varepsilon_{pem}(t_i, \hat{\boldsymbol{\rho}}) &= y(t_i) - \hat{y}(t_i|t_{i-1}) \\ &= H^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}) \left(y(t_i) - G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i)\right),\end{aligned}\tag{3.17}$$

with  $\boldsymbol{\rho} = [\boldsymbol{\theta}^T \quad \boldsymbol{\eta}^T]^T$ .

The one-step-ahead PEM takes into account the past measured output. Nevertheless, the estimated output could be simulated only with the past measured input, such as:

$$\hat{y}(t_i|t_{i-1}) = G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i) + H(z^{-1}, \hat{\boldsymbol{\eta}})e^*(t_i),\tag{3.18}$$

where  $e^*$  is a random number that would be generated by the computer. However, with such a mathematical framework, the simulation is not appropriate to identify the noise model due to the randomness of  $e^*$ . There is one exception when  $H = 1$ . In this case, the additive noise is white and can be neglected in the simulation. For this case, we talk about an *output error* model and the prediction error is defined by:

$$\varepsilon_{oem}(t_i, \hat{\boldsymbol{\theta}}) = y(t_i) - G_d(z^{-1}, \hat{\boldsymbol{\theta}})u(t_i).\tag{3.19}$$

As outlined with Exercise 61 in [Schoukens *et al.* 2012], the error converge to zero more rapidly with the PEM than with the simulation. That is due to the initial states effects which vanishes faster.

Regarding the closed-loop aspects, this one-step-ahead method perfectly suits for a direct identification approach. As outlined in [Forssell & Ljung 1999], the identification of the system and noise models are linked. An error in one part causes fallacies in the other.

### 3.3.3 Hybrid output error method

In the previous section, the output error model has been introduced. A well-known specific case of the PEM, the Output Error Method (OEM) was named after it. This approach is widely used to identify CT linear or nonlinear systems. It has proven its suitability in Automatic Control [Richalet *et al.* 1971, Landau *et al.* 2001, Carrillo *et al.* 2009], in robotics [Gautier *et al.* 2011, Gautier *et al.* 2013a] and in aeronautics [Klein 1989, Klein & Morelli 2006] for instance. Even though the dynamic model is continuous time, the identification is based on sampled data. Hence, similarly to the hybrid RIVC method, it is appropriate to use the term *hybrid OEM*.

To simulate the CT system and obtain a simulated output, the differential equations must be solved. Many numerical solvers exist in the literature like the well-known Runge-Kutta method, for further examples see [Hairer *et al.* 1993]. In this thesis, they are referred to as *integration solvers* to avoid confusion with the *optimisation solvers* introduced in the following section. In practice, the integration solver needs the same input as the real system and a set of values for the parameters to identify. The choice of the integration solver is decisive. For each model, the practitioner must find the integration solver which suits to the system properties. For instance, if the system presents two dynamics whose the characteristic times greatly differ, a stiff solver should be employed. We can also think about the simulation of the Coulomb friction which can represent a difficulty for many solvers. If the integration solver is not appropriate, it may lead to a biased identification.

The initial values is a crucial point for OEM. With a bad initialisation the optimisation solver may lead to local minimum (if it is a local optimizer) or even diverge. The integration solver may also diverge if the parameters are not suitable. Depending on the application, different techniques may be used to initialize correctly the method. If the problem is linear with respect to the parameters and if all the states are available, a LS estimation can be employed. As shown in [Gautier *et al.* 2013a], in the field of robotics the Computer-Aided Design (CAD) values of the inertia are enough accurate to initialize. In aircraft identification, initial values can be available from wind tunnel test or computational fluid dynamics.

Inspired from [Jategaonkar 2006], Figure 3.3 illustrates the OEM principle where  $\mathbf{y}$  is the  $(N \times 1)$  vector of the measured output,  $\mathbf{y}_s$  is the  $(N \times 1)$  vector of the simulated output,  $\hat{\boldsymbol{\theta}}$  the  $(b \times 1)$  vector of estimated parameters, and  $\frac{\partial \mathbf{y}_s}{\partial \boldsymbol{\theta}}$  is the output sensitivity, which is a  $(N \times b)$  Jacobian matrix.  $N$  is the number of sampling points considered and the  $b$  is the number of unknown parameters. As it can be seen, the only stochastic signal is the measurement noise. The input signal is indeed assumed to be noise free. In addition, the integration solver is deterministic. It is consequently impossible to take into account process noise in the simulation with such a method.

About the noise correlation due to the closed-loop, the OEM simulate the whole closed-loop system with the reference signal as input. Consequently, the estimated signals do not contain noise correlated with the measurement one.

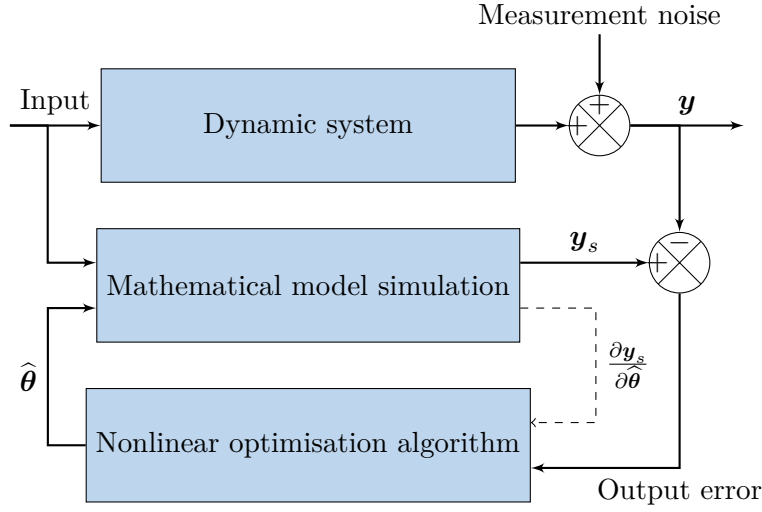


Figure 3.3: Block diagram of the output error method

### 3.3.4 Computational aspects

With the PEM, the unknown system parameters are tuned iteratively, so that the predicted model output fits the measured system output, with

$$\hat{\boldsymbol{\theta}}^{it+1} = \hat{\boldsymbol{\theta}}^{it} + \Delta \hat{\boldsymbol{\theta}}^{it} \quad (3.20)$$

where  $\Delta \hat{\boldsymbol{\theta}}^{it}$  is the innovation vector. This innovation is calculated differently depending on the applied nonlinear optimisation algorithm. The criterion minimisation is usually solved thanks to nonlinear optimisation algorithms based on a first- or second-order Taylor series expansion like the gradient, the Gauss-Newton and the Levenberg-Marquardt methods. With these methods, the parameter innovations are respectively given by

$$\Delta \hat{\boldsymbol{\theta}}^{it} = -\mu \frac{\partial V_N}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{it}} = -\mu V'_N(\hat{\boldsymbol{\theta}}^{it}), \quad (3.21)$$

$$\Delta \hat{\boldsymbol{\theta}}^{it} = -[V''_N(\hat{\boldsymbol{\theta}}^{it})]^{-1} V'_N(\hat{\boldsymbol{\theta}}^{it}), \quad (3.22)$$

$$\Delta \hat{\boldsymbol{\theta}}^{it} = -[V''_N(\hat{\boldsymbol{\theta}}^{it}) + \mu \cdot \text{diag}(V''_N(\hat{\boldsymbol{\theta}}^{it}))]^{-1} V'_N(\hat{\boldsymbol{\theta}}^{it}), \quad (3.23)$$

where  $V'_N(\hat{\boldsymbol{\theta}}^{it})$  and  $V''_N(\hat{\boldsymbol{\theta}}^{it})$  are the gradient vector and the hessian matrix of the criterion  $V_N$ ,  $\mu$  is a parameter which permits the tuning of the iterative algorithms. Those innovations require the computation of the criterion derivatives with respect to the parameters. In some cases those derivatives can be exactly known. For example, in [Carrillo *et al.* 2009], the authors have developed an exact formulation of the first derivative for CT-LTI systems. In [Landau *et al.* 2001], the authors have done the same for nonlinear systems. Those derivatives of the criterion are function

of the derivatives of the system's outputs with respect to the parameters, which are called *sensitivities*.

However, those sensitivities can be difficult to calculate exactly for complex nonlinear systems, like the robots. In such cases, they are usually approximated with finite differences. Furthermore, they can even be undefined due to the selected scalar function (e.g. absolute value at zero) or the model's properties (e.g. sign function at zero). The user should then select a solver that is not based on the calculation of the derivatives. Those solvers are referred to as derivative-free optimisation solvers; see e.g. the Nelder-Mead method [Nelder & Mead 1965]. The user can also consider stochastic and/or global optimisation solvers. It appears that many solutions are available for the optimisation, depending on the considered identification problem. For example, several optimisation solutions available for an identification purpose are summarized in [Walter & Pronzato 1994].

### 3.4 Concluding remarks

The *Refined Instrumental Variable for Continuous-time* systems has been presented in this chapter. Although this approach has its roots in the areas of time series analysis and environmental modelling, it has been used successfully in many applications; see [Young 2011] and the references given therein. In parallel, the system identification community has established the *Prediction Error Methods*. This framework encompasses a lot of methods including the *Output Error Method* that is able to identify CT models with white measurement noise. All those methods are able to identify systems operating in closed-loop by considering different approaches: direct, indirect or joint input-output.

This chapter concludes the preliminary part of this thesis. The contributions are now presented starting with the study of the identification methods based on the model simulation.

## **Part III**

# **Contributions to robot system identification**



# Evaluation of the robustness of identification methods based on the auxiliary model simulation

---

As shown in chapter 2, the usual robot identification methods use the input torque because of its linearity with respect to the parameters. However, it has been seen in chapter 3 that the OEM are able to identify models nonlinear with respect to the parameters. The resultant question is: would it be possible to consider another identification signal like the output position to improve the robustness of the estimation? Furthermore, since those methods rely on the simulation of the closed-loop system, an underlying issue is the robustness of those simulations. In other words, the question is to what extent those methods can reject modelling errors or a too large noise.

In section 4.1, we investigate the choice of the identification signal for OEM applied to robot identification. Then, the robustness of the simulation used for the DIDIM and IDIM-IV methods is investigated in section 4.2. Then, some miscellaneous remarks are made in section 4.3 before concluding in section 4.4.

---

<b>4.1</b>	<b>Choice of the identification signal . . . . .</b>	<b>48</b>
4.1.1	Output error methods for robot identification . . . . .	48
4.1.2	Sensitivity analysis . . . . .	49
4.1.3	Contribution of the linearity . . . . .	55
4.1.4	Experimental validation . . . . .	55
<b>4.2</b>	<b>Robustness to modeling errors of the DIDIM and IDIM-IV methods . . . . .</b>	<b>61</b>
4.2.1	Similarity between the methods . . . . .	61
4.2.2	Disparity between the methods . . . . .	63
4.2.3	Experimental validations . . . . .	65
<b>4.3</b>	<b>Miscellaneous remarks . . . . .</b>	<b>66</b>
4.3.1	Closed-loop aspects of the robot identifications methods . . . . .	66
4.3.2	Considerations on the robot identification with OEM . . . . .	66
4.3.3	Remarks on the linearity . . . . .	68
<b>4.4</b>	<b>Conclusions . . . . .</b>	<b>70</b>

---

## 4.1 Choice of the identification signal

### 4.1.1 Output error methods for robot identification

By applying directly the OEM to a robot model, it seems natural to take the joint position vector as the identification signal. The output error vector is defined by:

$$\boldsymbol{\varepsilon}_{CLOE}(t, \boldsymbol{\theta}) = \mathbf{q}_m(t) - \mathbf{q}_s(t, \boldsymbol{\theta}), \quad (4.1)$$

where  $\mathbf{q}_s$  is the  $(n \times 1)$  vector of simulated joint positions. Since the robots are unstable in open-loop, they are identified in closed-loop and the dedicated identification method is called the Closed-Loop Output Error (CLOE) method. As explained in section 3.3.3, the simulated output  $\mathbf{q}_s$  is generated with the noise-free reference signal  $\mathbf{q}_r$ . Therefore, there is not bias induced by a noise correlation and the estimation is consistent, assuming that there is no modelling error and that the optimisation solver has converged to the global minimum.

As explained in section 3.3.4, the OEM problem is usually solved thanks to nonlinear optimisation algorithms such as the gradient or Newton methods. In this part, we focus on the Gauss-Newton method (GN method), which is based on a second order Taylor series expansion of  $\mathbf{q}_s$ , at current estimates  $\boldsymbol{\theta}_{CLOE}$ ; see (3.22). After data sampling, the following over-determined system is obtained at iteration  $it$ :

$$\Delta \mathbf{y}(\mathbf{q}) = \boldsymbol{\Psi}_{CLOE}^{it} \Delta \boldsymbol{\theta}_{CLOE}^{it} + \mathbf{e}_{CLOE} \quad (4.2)$$

where

- $\Delta \mathbf{y}(\mathbf{q})$  is the  $(r' \times 1)$  vector built from the sampling of  $\boldsymbol{\varepsilon}_{CLOE}(t, \boldsymbol{\theta})$ , similarly to (2.20);

- $\boldsymbol{\Psi}_{CLOE}^{it}$  is the  $(r' \times b)$  matrix built from the  $n$  matrices  $\boldsymbol{\Psi}_{CLOE}^{it,j} = \begin{bmatrix} \Delta \mathbf{q}_s^j(t_1) \\ \vdots \\ \Delta \mathbf{q}_s^j(t_{n_m}) \end{bmatrix}$ , where  $\Delta \mathbf{q}_s^j(\cdot)$  is the  $j^{\text{th}}$  row of the  $(n \times b)$  jacobian matrix  $\Delta \mathbf{q}_s = \left. \frac{\partial \mathbf{q}_s}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{CLOE}^{it}}$ ;

- $\mathbf{e}_{CLOE}$  is the  $(r' \times 1)$  vector built from the sampling of  $\boldsymbol{\varepsilon}_{CLOE}$  and the residuals of the Taylor series expansion

- $r' = n \cdot n_m$  is the number of equation considered, without any decimation.

$\Delta \hat{\boldsymbol{\theta}}_{CLOE}^{it}$  is the  $(b \times 1)$  vector of estimated parameters increments and is the LS solution of (4.2). Each element of the jacobian  $\Delta \mathbf{q}_s$  is an output sensitivity function which defines the variation of the output position with respect to the parameters. Usually, those sensitivities functions are not exactly known and approximated with finite differences.

The construction of the *en-bloc* formulation is really similar to the one of the IDIM-LS method (2.20). The difference is the use of  $r'$  instead of  $r$ . This is due



to the fact that we are in an output error framework. Hence, the additive noise is assumed white and does not need any prefiltering such as the decimate filter. It begins to emerge here a limitation of OEM for robot identification. In addition, as we shall see later the decimation process presents also an advantage for the conditioning number. The number of sampling points to treat is indeed reduced for the optimisation solver.

As we have seen in section 2.3, it is common to use the input torque/force for robot identification. Therefore, a variant of the CLOE method based on the input signal can be considered. This technique called the Closed-Loop Input Error (CLIE) method relies on the input error vector

$$\boldsymbol{\varepsilon}_{CLIE}(t, \boldsymbol{\theta}) = \boldsymbol{\tau}(t) - \boldsymbol{\tau}_s(t, \boldsymbol{\theta}). \quad (4.3)$$

If the problem is solved with the GN algorithm,  $\Delta\boldsymbol{\theta}_{CLIE}^{it}$  is the LS solution of

$$\Delta\mathbf{y}(\boldsymbol{\tau}) = \boldsymbol{\Psi}_{CLIE}^{it} \Delta\boldsymbol{\theta}_{CLIE}^{it} + \mathbf{e}_{CLIE} \quad (4.4)$$

where

- $\Delta\mathbf{y}(\boldsymbol{\tau})$  is the  $(r' \times 1)$  vector built from the sampling of  $\boldsymbol{\varepsilon}_{CLIE}(t, \boldsymbol{\theta})$ , similarly to (2.20);
- $\boldsymbol{\Psi}_{CLIE}^{it}$  is the  $(r' \times b)$  matrix built from the  $n$  matrices  $\boldsymbol{\Psi}_{CLIE}^{it,j} = \begin{bmatrix} \Delta_{\boldsymbol{\tau}_s}^j(t_1) \\ \vdots \\ \Delta_{\boldsymbol{\tau}_s}^j(t_{n_m}) \end{bmatrix}$ , where  $\Delta_{\boldsymbol{\tau}_s}^j(\cdot)$  is the  $j^{\text{th}}$  row of the  $(n \times b)$  jacobian matrix  $\Delta_{\boldsymbol{\tau}_s} = \left. \frac{\partial \boldsymbol{\tau}_s}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{CLIE}^{it}}$ ;
- $\mathbf{e}_{CLIE}$  is the  $(r' \times 1)$  vector built from the sampling of  $\boldsymbol{\varepsilon}_{CLIE}$  and the residuals of the Taylor series expansion.

The general principle of the CLOE and CLIE methods is illustrated by Figure 4.1. Both methods are iterative and can share the initialisation and the convergence criterion of the IDIM-IV and DIDIM methods. In the following section, we investigate their properties.

### 4.1.2 Sensitivity analysis

#### Sensitivities relation

For the CLIE method, at iteration  $it$ , the input sensitivity is defined such as

$$\Delta_{\boldsymbol{\tau}_s}(t) = \left. \frac{\partial \boldsymbol{\tau}_s(t)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{CLIE}^{it}}. \quad (4.5)$$

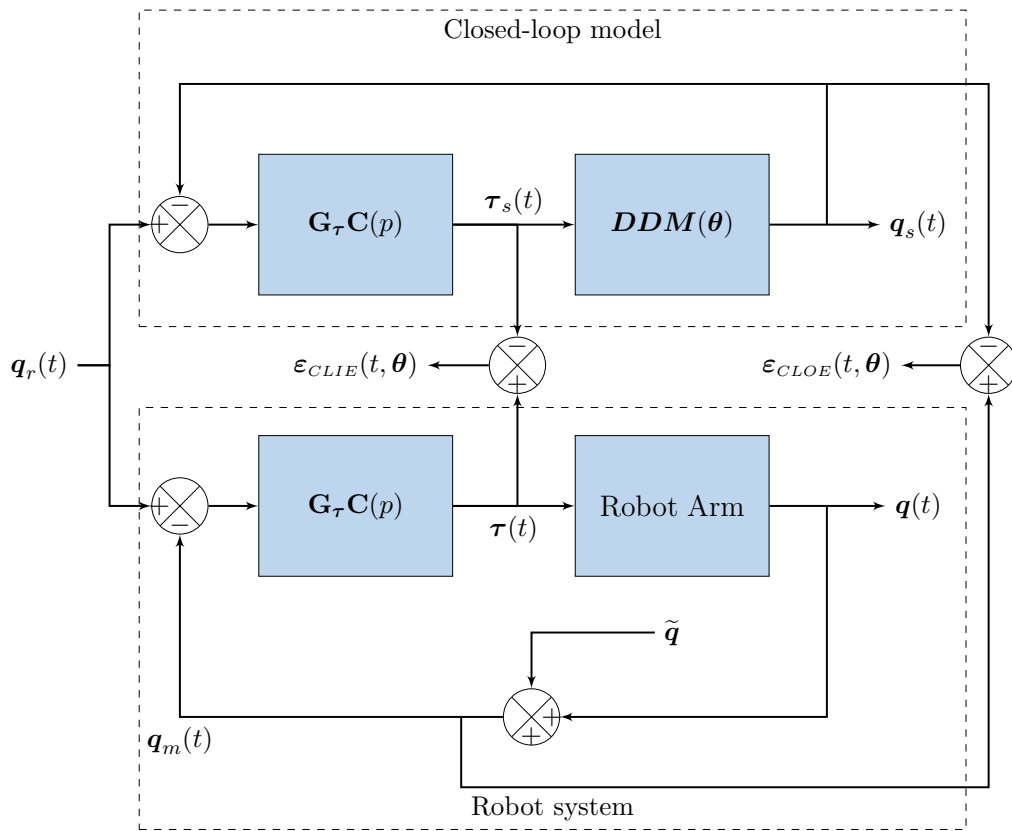


Figure 4.1: Block diagram of the CLOE and CLIE methods

By using the controller definition (2.11), it comes:

$$\Delta_{\tau_s}(t) = \left. \frac{\partial \mathbf{G}_\tau \mathbf{C}(p) \left( \mathbf{q}_r(t) - \mathbf{q}_s(t, \hat{\boldsymbol{\theta}}_{CLIE}^{it}) \right)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{CLIE}^{it}}, \quad (4.6)$$

and assuming that the controller is known, or at least not identified at the same time:

$$\begin{aligned} \Delta_{\tau_s}(t) &= -\mathbf{G}_\tau \mathbf{C}(p) \left. \frac{\partial \left( \mathbf{q}_s(t, \hat{\boldsymbol{\theta}}_{CLIE}^{it}) \right)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{CLIE}^{it}} \\ &= -\mathbf{G}_\tau \mathbf{C}(p) \Delta_{q_s}(t). \end{aligned} \quad (4.7)$$

Eq. (4.7) is the key relation to compare the CLOE and CLIE methods. Loosely speaking, the CLIE method is a frequency weighting of the CLOE method by the controller. In practice, the input sensitivity functions can be obtained from the filtering of the output ones, if they are available.

Because the robots are controlled in position, the following relation is expected:

$$\mathbf{q}_m(t) \approx \mathbf{q}_r(t). \quad (4.8)$$

Assuming that the optimisation solver is adequately initialized, it implies:

$$\mathbf{q}_s(t) \approx \mathbf{q}_r(t), \quad (4.9)$$

at each iteration *it* of the algorithm. Because of (4.8) and (4.9), the controller may be assumed to operate in low frequencies range and, then, to be a  $(n \times n)$  constant matrix  $\mathbf{C}_0$ . If the controller is a PD control law then this result is straightforward. If the controller contains an integral part, this result can be still considered as valid. In fact, as explained in section 2.2.1, to avoid oscillations due to the Coulomb friction, the integral action is deactivated when the position error is too small. Thus, (4.7) may be re-written

$$\Delta_{\tau_s}(t) = -\mathbf{G}_\tau \mathbf{C}_0 \Delta_{q_s}(t). \quad (4.10)$$

### Equivalence of the CLOE and CLIE estimations

Considering relation (4.10), the CLIE method can be seen as a weighted CLOE method (WCLOE). That can be seen with the LS solution of (4.4):

$$\begin{aligned}\Delta\widehat{\boldsymbol{\theta}}_{CLIE}^{it} &= \left[ \left( \boldsymbol{\Psi}_{CLIE}^{it} \right)^T \boldsymbol{\Psi}_{CLIE}^{it} \right]^{-1} \left( \boldsymbol{\Psi}_{CLIE}^{it} \right)^T \Delta\mathbf{y}(\boldsymbol{\tau}) \\ &= \left[ \left( \boldsymbol{\Psi}_{CLOE}^{it} \right)^T \bar{\mathbf{C}}_0^T \bar{\mathbf{C}}_0 \boldsymbol{\Psi}_{CLOE}^{it} \right]^{-1} \left( \boldsymbol{\Psi}_{CLOE}^{it} \right)^T \bar{\mathbf{C}}_0^T \bar{\mathbf{C}}_0 \Delta\mathbf{y}(\mathbf{q}) \\ &= \left[ \left( \boldsymbol{\Psi}_{CLOE}^{it} \right)^T \mathbf{W} \boldsymbol{\Psi}_{CLOE}^{it} \right]^{-1} \left( \boldsymbol{\Psi}_{CLOE}^{it} \right)^T \mathbf{W} \Delta\mathbf{y}(\mathbf{q}),\end{aligned}\quad (4.11)$$

where

- $\bar{\mathbf{C}}_0$  is  $(r' \times r')$  matrix built from the constant controller matrix  $\mathbf{G}_\tau \mathbf{C}_0$ ;
- $\Delta\mathbf{y}(\boldsymbol{\tau})$  is equal to  $-\bar{\mathbf{C}}_0 \Delta\mathbf{y}(\mathbf{q})$  because it is built from

$$\begin{aligned}\boldsymbol{\tau}(t) - \boldsymbol{\tau}_s(t) &= \mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_r(t) - \mathbf{q}_m(t)) - \mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_r(t) - \mathbf{q}_s(t)) \\ &= -\mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_m(t) - \mathbf{q}_s(t)) \\ &\approx -\mathbf{G}_\tau \mathbf{C}_0 (\mathbf{q}_m(t) - \mathbf{q}_s(t)).\end{aligned}$$

Relation (4.11) is clearly a weighted least-squares solution. If  $n_m$  is sufficiently large, according to the theory of statistics [Davidson & MacKinnon 1993], one has

$$\mathbb{E}[\widehat{\boldsymbol{\theta}}_{CLIE}^{it}] \rightarrow \mathbb{E}[\widehat{\boldsymbol{\theta}}_{CLOE}^{it}].$$

It comes out that the CLIE and CLOE methods asymptotically provide the same estimates. The resulting question is: is there a dominant estimator?

Because we are in an output error framework, the output position can be written:

$$\mathbf{q}_m(t) = \mathbf{q}_{nf}(t) + \mathbf{n}_q(t), \quad (4.12)$$

where  $\mathbf{n}_q$  is a  $(n \times 1)$  vector of independent white noises with a  $(n \times n)$  covariance matrix  $\boldsymbol{\Lambda}_q$ . With the closed-loop, the input is given by:

$$\begin{aligned}\boldsymbol{\tau}(t) &= \mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_r(t) - \mathbf{q}_m(t)) \\ &= \mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_r(t) - \mathbf{q}_{nf}(t)) - \mathbf{G}_\tau \mathbf{C}(p) \mathbf{n}_q(t).\end{aligned}\quad (4.13)$$

The reference trajectory is noise-free. Consequently, the noise seen by the input torque is  $\mathbf{v}(t) = -\mathbf{G}_\tau \mathbf{C}(p) \mathbf{n}_q(t)$ ; see (2.17). With no further assumption, it appears that if the output error assumption is made on the output position, it is not validated for the input torque. Nonetheless, we assume that the closed-loop system operates in its bandwidth and consequently  $\mathbf{C}(p) \approx \mathbf{C}_0$ . The noises relation is given by:

$$\mathbf{v}(t) = -\mathbf{G}_\tau \mathbf{C}_0 \mathbf{n}_q(t). \quad (4.14)$$

Consequently,  $\mathbf{E}[\mathbf{v}] = -\mathbf{G}_\tau \mathbf{C}_0 \mathbf{E}[\mathbf{n}_q] = 0$  and, using covariance matrix definition (C.8), the covariances are linked such as:

$$\begin{aligned} \mathbf{\Lambda}_v &= \mathbf{E} \left[ (\mathbf{v} - \mathbf{E}[\mathbf{v}]) (\mathbf{v} - \mathbf{E}[\mathbf{v}])^T \right] \\ &= \mathbf{E} \left[ \mathbf{G}_\tau \mathbf{C}_0 \mathbf{n}_q \mathbf{n}_q^T \mathbf{C}_0^T \mathbf{G}_\tau^T \right] \\ &= \mathbf{G}_\tau \mathbf{C}_0 \mathbf{\Lambda}_{n_q} \mathbf{C}_0^T \mathbf{G}_\tau^T. \end{aligned} \quad (4.15)$$

In practice, the matrices  $\mathbf{G}_\tau$  and  $\mathbf{C}_0$  can be assumed diagonal, which is justified by the technology of the actuators, the controller and the sensors presented in section 2.2. Attention is drawn to the fact that the covariance  $\mathbf{\Lambda}_v$  is the one of  $\mathbf{v}(t)$  whereas  $\mathbf{\Lambda}_\tau$  is the one of  $\mathbf{v}_{F_p}(t)$ ; see (2.25). Assuming no modelling error and that the optimisation solver has converged to true parameters values, the estimated parameters covariances are then given by:

$$\begin{aligned} \mathbf{\Sigma}(\hat{\boldsymbol{\theta}}_{CLIE}^{it}) &= \left[ (\boldsymbol{\Psi}_{CLIE}^{it})^T \boldsymbol{\Omega}_v^{-1} \boldsymbol{\Psi}_{CLIE}^{it} \right]^{-1} \\ &= \left[ (\boldsymbol{\Psi}_{CLOE}^{it})^T \bar{\mathbf{C}}_0^T (\bar{\mathbf{C}}_0 \boldsymbol{\Omega}_q \bar{\mathbf{C}}_0^T)^{-1} \bar{\mathbf{C}}_0 \boldsymbol{\Psi}_{CLOE}^{it} \right]^{-1} \\ &= \left[ (\boldsymbol{\Psi}_{CLOE}^{it})^T \boldsymbol{\Omega}_q^{-1} \boldsymbol{\Psi}_{CLOE}^{it} \right]^{-1} \\ &= \mathbf{\Sigma}(\hat{\boldsymbol{\theta}}_{CLOE}^{it}), \end{aligned} \quad (4.16)$$

where  $\boldsymbol{\Omega}_v$  and  $\boldsymbol{\Omega}_q$  are the  $(r' \times r')$  covariances matrices built respectively built from  $\mathbf{\Lambda}_v$  and  $\mathbf{\Lambda}_q$  like in (2.24). It comes out that the CLIE and the CLOE methods provide the same estimates with the same variances.

To summarize, the users have the choice between  $\mathbf{q}_m$  and  $\boldsymbol{\tau}$ , but the CLIE and the CLOE methods asymptotically provide the same results. The question is: what is the best choice to identify continuous-time systems operating in closed loop? The following subsection presents the main difference between the CLOE and CLIE methods.

### Robustness of the CLIE method to errors

With the good tracking assumption (4.9),  $\mathbf{q}_s$  has a little dependence to parameters' variations and the output sensitivity matrix  $\boldsymbol{\Delta}_{\mathbf{q}_s}$  contains little information. This implies that the singular values of  $\boldsymbol{\Delta}_{\mathbf{q}_s}$  are small whereas the its conditioning number denoted as  $cond_2^{CLOE}$  may be very good i.e.  $cond_2^{CLOE} \approx 1$ ; see e.g. [Lawson & Hanson 1974]. To show that, from (4.2), the OLS solution can be written

$$\boldsymbol{\Delta} \hat{\boldsymbol{\theta}}_{CLOE}^{it} = \left[ (\boldsymbol{\Psi}_{CLOE}^{it})^T \boldsymbol{\Psi}_{CLOE}^{it} \right]^{-1} (\boldsymbol{\Psi}_{CLOE}^{it})^T \boldsymbol{\Delta} \mathbf{y}(\mathbf{q}) \quad (4.17)$$

Then, the relative variation of the solution  $\Delta\hat{\theta}_{CLOE}^{it}$  denoted as  $d\Delta\hat{\theta}_{CLOE}^{it}$  is expressed by two upper bounds given by

$$\frac{\|d\Delta\hat{\theta}_{CLOE}^{it}\|_2}{\|\Delta\hat{\theta}_{CLOE}^{it}\|_2} \leq \text{cond}_2^{CLOE} \frac{\|d\Delta\mathbf{y}(\mathbf{q})\|_2}{\|\Delta\mathbf{y}(\mathbf{q})\|_2}, \quad (4.18)$$

$$\frac{\|d\Delta\hat{\theta}_{CLOE}^{it}\|_2}{\|\Delta\hat{\theta}_{CLOE}^{it} + d\Delta\hat{\theta}_{CLOE}^{it}\|_2} \leq \text{cond}_2^{CLOE} \frac{\|d\Psi_{CLOE}^{it}\|_2}{\|\Psi_{CLOE}^{it}\|_2}, \quad (4.19)$$

where  $d\Delta\hat{\theta}_{CLOE}^{it}$ ,  $d\Delta\mathbf{y}(\mathbf{q})$  and  $d\Psi_{CLOE}^{it}$  are small variations of  $\Delta\hat{\theta}_{CLOE}^{it}$ ,  $\Delta\mathbf{y}(\mathbf{q})$  and  $\Psi_{CLOE}^{it}$  respectively.  $\|\cdot\|_2$  is the 2-norm of a vector or a matrix.

Let  $\mu_{min}^{CLOE}$  and  $\mu_{max}^{CLOE}$  be the smallest and the greatest singular values of  $\Psi_{CLOE}^{it}$  respectively. With,  $\text{cond}_2^{CLOE} = \mu_{max}^{CLOE} / \mu_{min}^{CLOE}$  and  $\|\Psi_{CLOE}^{it}\|_2 = \mu_{max}^{CLOE}$ , one obtains

$$\frac{\|d\Delta\hat{\theta}_{CLOE}^{it}\|_2}{\|\Delta\hat{\theta}_{CLOE}^{it}\|_2} \leq \frac{\mu_{max}^{CLOE}}{\mu_{min}^{CLOE}} \frac{\|d\Delta\mathbf{y}(\mathbf{q})\|_2}{\|\Delta\mathbf{y}(\mathbf{q})\|_2}, \quad (4.20)$$

$$\frac{\|d\Delta\hat{\theta}_{CLOE}^{it}\|_2}{\|\Delta\hat{\theta}_{CLOE}^{it} + d\Delta\hat{\theta}_{CLOE}^{it}\|_2} \leq \frac{\|d\Psi_{CLOE}^{it}\|_2}{\mu_{min}^{CLOE}}. \quad (4.21)$$

Assuming that  $\mu_{min}^{CLOE}$  and  $\mu_{max}^{CLOE}$  are very small with  $\mu_{min}^{CLOE} \approx \mu_{max}^{CLOE}$ , one has  $\text{cond}_2^{CLOE} \approx 1$  and  $1/\mu_{min}^{CLOE}$  very large.

With (4.20), it appears the interest of having a conditioning number as small as possible to minimize the relative norm error on the solution vector. That property was pointed out in [Pressé & Gautier 1993] where a new criterion of exciting trajectory was developed based on this relation. Equation (4.21) pinpoints the role of  $\mu_{min}^{CLOE}$  as we shall see below. In practice, two cases must be considered for the CLOE method.

- The first case is when (4.8) and (4.9) are fulfilled. That is to say that the controller has been designed to provide an excellent tracking, while rejecting perturbations and model mismatches. In this case, there is  $\varepsilon_{CLOE}(t, \theta) \approx 0$ , for each  $t$ , or equivalently  $\Delta\mathbf{y}(\mathbf{q}) \approx 0$ . Then, from (4.17), the innovation is approximately zero. It comes out that the CLOE method may be totally insensitive to modelling errors and/or to measurement noise, if the control law is effective enough. In practice, the optimisation solver would not move from the initialisation point.
- The second case to be considered is when (4.8) and (4.9) are not totally fulfilled. In other words, the controller presents relatively poor tracking performances. Regarding the upper bound (4.21),  $\Delta\hat{\theta}_{CLOE}^{it}$  may not be robust against a small variation  $d\Psi_{CLOE}^{it}$ . In fact,  $\|d\Psi_{CLOE}^{it}\|_2$  is amplified by

$1/\mu_{min}^{CLOE}$ . It can be thought that  $d\Psi_{CLOE}^{it}$  is mainly due to modelling errors and a poor initialisation, since the simulated output is not affected by the measurement noise. This leads to unpredictable results because those errors can be interpreted as an information by the optimisation solver. It comes out that the CLOE method may be sensitive to small modelling errors, if the control law is not effective enough.

As a result, the objective is to obtain a conditioning number as small as possible while having the smallest singular value as large as possible. Contrary to the CLOE method, the CLIE one will not suffer from small singular values because it is the CLOE method weighted by the control. As regards the industrial robots, the gains of the controller are usually high and this implies  $\mathbf{C}_0 \gg \mathbf{I}_n$ . Then, it follows that  $\mu_{min}^{CLIE}$  and  $\mu_{max}^{CLIE}$  the smallest and the greatest singular values of  $\Psi_{CLIE}^{it}$  are far greater than  $\mu_{min}^{CLOE}$  and  $\mu_{max}^{CLOE}$ . It comes out that the CLIE method is more robust against a small modelling error and is more sensitive to parameters' variations than the CLOE method.

#### 4.1.3 Contribution of the linearity

A careful reader might notice that the CLIE method is really similar to the DIDIM method introduced in section 2.3.4. The error considered by the DIDIM method (2.35) is the CLIE error (4.3) filtered by the parallel filter. The introduction of this filter implies that it is technically not an output error model anymore. However, the DIDIM method is not a PEM because the noise model is not identified in the process. The second specificity of the DIDIM method is the PLR assumption. With such an assumption the GN algorithm becomes equivalent to the linear LS, as shown in section 4.3.3.3 of [Walter & Pronzato 1994]. The input sensitivity is written:

$$\begin{aligned} \Delta\tau_s(t) &= \left. \frac{\partial\tau_s(t)}{\partial\theta} \right|_{\theta=\hat{\theta}_{DIDIM}^{it}} \\ &= \phi\left(\mathbf{q}_s(t, \hat{\theta}_{DIDIM}^{it}), \dot{\mathbf{q}}_s(t, \hat{\theta}_{DIDIM}^{it}), \ddot{\mathbf{q}}_s(t, \hat{\theta}_{DIDIM}^{it})\right). \end{aligned} \quad (4.22)$$

Thanks to this relation, the input sensitivity can be calculated with only one simulation of the closed-loop system. In the opposite, with finite differences,  $b + 1$  simulations are needed to evaluate the sensitivity; considering a forward or a backward first order and one-sided difference scheme. The gain in computing time is therefore not negligible.

#### 4.1.4 Experimental validation

In this part, we illustrate the performances of the CLIE, CLOE and DIDIM methods with the Stäubli TX40 presented in Appendix B. In a first time, the CLIE and CLOE methods are studied in a strict output error framework. In a second time, the three methods are compared in more general framework by taking into account the decimate filter.

The DDM is simulated thanks to a Simulink<sup>©</sup> model integrated with the *ode3* integration solver: Bogacki-Shampine. The gains of the simulated controller are not updated to keep the bandwidth constant as it could be done with the DIDIM method, see e.g. [Gautier *et al.* 2013a]. For the CLIE and CLOE methods, the optimisation is done with the Levenberg-Marquardt (LM) algorithm implemented in the *lsqnonlin* function of the MatLab<sup>©</sup> Optimization Toolbox. The three methods are initialized with acceptable CAD values: all base parameters equal to 0 except  $ia_j = 1$  for  $j \neq 5$  and  $ia_5 = 2$  because of the coupling effect. The essential estimated parameters and relative standard deviation are summarized in Table 4.4.

### Strict output error framework

First of all, the CLIE estimated parameters are close to those provided in Appendix B and lie in the  $3\sigma$  confidence intervals of the LS estimates. The CLOE estimates are not so far except for few parameters like  $mx_4$  or  $fc_6$ . This can be explained by the lack of sensitivity of this method. As explained in section 4.1.2, the CLIE method is in fact a CLOE method weighted by the controller. Due to large controller gains, the sensitivity with respect to the parameters is more important at the input than at the output. That is confirmed by the singular values of the jacobian matrices in Table 4.2. The interest of considering the input torque for the identification is also visible in Table 4.3. The CLOE method indeed provides larger simulation errors on the torques than the CLIE method, whereas both have equivalent errors on the positions. The relative standard deviations of the estimated parameters may seem promising, however there is an issue with the computation of the variances. The additive noises are indeed assumed to be serially uncorrelated which is not verified with the residuals autocorrelations; see Figures 4.2 and 4.3 for the CLOE and CLIE methods respectively. Consequently, the CLOE and CLIE methods cannot be applied in a strict output error framework to industrial robots, although the CLIE method provides consistent estimates. In fact, the CLIE estimator appears to be consistent but inefficient; see Appendix C.2.

### OEM with the decimate filter

In this part, we consider the CLIE and CLOE methods with the decimate filter as well as the DIDIM method. According to Appendix B, the cut-off frequency is set at 20 Hz. The estimated parameters provided in Table 4.4 are close to the previous estimations. There is still a lack of sensitivity for the CLOE method. Concerning the relative standard deviations of the CLIE, CLOE and DIDIM methods, they proved to be equivalent. Since the CLIE and DIDIM results are really close, only the correlation of the DIDIM residuals is presented. Thus, Figure 4.4 and Figure 4.5 depict the autocorrelations of the identification residuals of the CLOE and DIDIM methods respectively. For both methods, the effect of the parallel filter is clear since the estimated autocorrelations coefficients are included in the confidence intervals indicated by the blue lines. These estimated autocorrelations prove that the CLOE



Table 4.1: CLIE and CLOE estimates - No decimate filter

	$\hat{\theta}_{CLIE}^T$	$\hat{\theta}_{CLOE}^T$
$zz_{1_r}$	1.25 (0.32 %)	1.22 (0.24 %)
$fv_1$	7.96 (0.28 %)	8.18 (0.12 %)
$fc_1$	7.21 (0.87 %)	6.34 (0.37 %)
$xx_{2_r}$	-0.47 (0.97 %)	-0.42 (0.55 %)
$xz_{2_r}$	-0.16 (1.85 %)	-0.17 (0.72 %)
$zz_{2_r}$	1.09 (0.29 %)	1.08 (0.18 %)
$mx_{2_r}$	2.24 (0.94 %)	2.43 (0.73 %)
$fv_2$	5.47 (0.35 %)	5.68 (0.22 %)
$fc_2$	8.35 (0.53 %)	7.40 (0.28 %)
$xx_{3_r}$	0.13 (2.07 %)	0.18 (1.31 %)
$zz_{3_r}$	0.12 (1.81 %)	0.11 (1.59 %)
$my_{3_r}$	-0.59 (0.44 %)	-0.53 (0.60 %)
$ia_3$	0.09 (1.94 %)	0.09 (1.62 %)
$fv_3$	1.94 (0.31 %)	1.93 (0.46 %)
$fc_3$	6.47 (0.31 %)	5.80 (0.31 %)
$mx_4$	-0.03 (3.40 %)	<b>0.01</b> (12.4 %)
$ia_4$	0.03 (1.08 %)	0.03 (1.38 %)
$fv_4$	1.11 (0.19 %)	1.15 (0.28 %)
$fc_4$	2.44 (0.31 %)	2.17 (0.54 %)
$my_{5_r}$	-0.04 (1.81 %)	-0.06 (1.31 %)
$ia_5$	0.04 (1.35 %)	0.05 (1.29 %)
$fv_5$	1.82 (0.24 %)	1.92 (0.34 %)
$fc_5$	3.01 (0.39 %)	2.26 (0.48 %)
$ia_6$	0.01 (2.06 %)	0.01 (1.78 %)
$fv_6$	0.66 (0.20 %)	0.67 (0.23 %)
$fc_6$	0.18 (5.92 %)	<b>0.04</b> (24.4 %)
$fv_{m6}$	0.60 (0.22 %)	0.57 (0.46 %)
$fc_{m6}$	1.94 (0.46 %)	1.95 (0.78 %)

Table 4.2: CLIE and CLOE optimisation parameters - No decimate filter

	CLIE	CLOE
$\mu_{max}$	$2.26 \cdot 10^4$	9.85
$\mu_{min}$	19.3	$2.6 \cdot 10^{-3}$
Conditioning number	1172	3845

Table 4.3: Direct comparison - Relative errors - CLIE and CLOE

	$\ \mathbf{q}_{m_j} - \mathbf{q}_{s_j}\  / \ \mathbf{q}_{m_j}\ $	$\ \boldsymbol{\tau}_j - \boldsymbol{\tau}_{s_j}\  / \ \boldsymbol{\tau}_j\ $
CLIE	0.017 %	6.97 %
CLOE	0.017 %	10.7 %

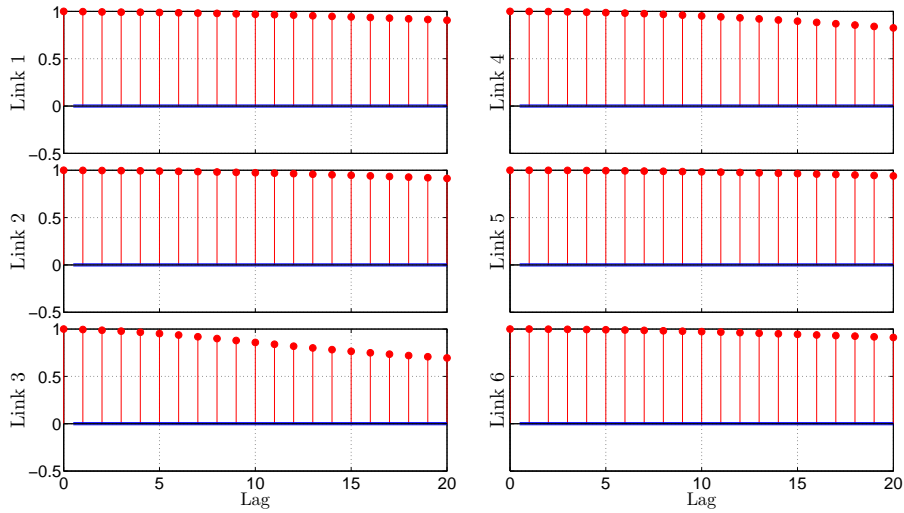


Figure 4.2: CLOE residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines) - No decimate filter

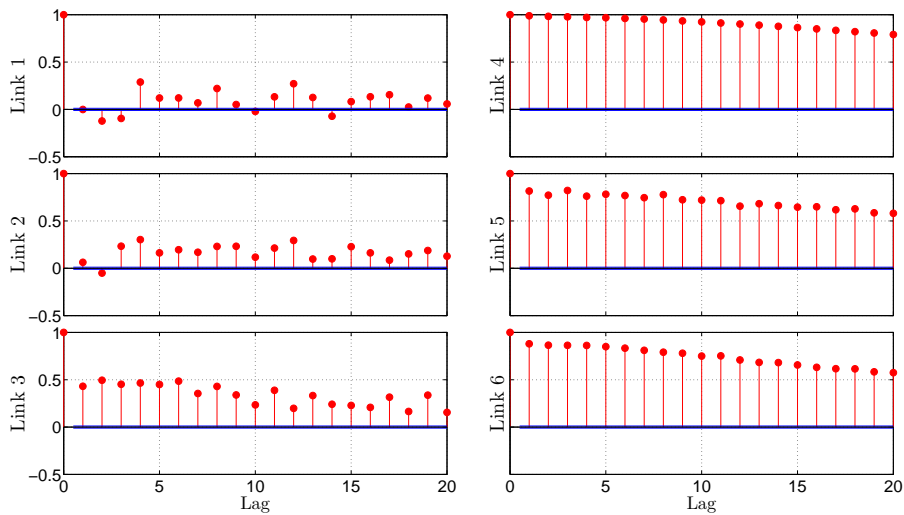


Figure 4.3: CLIE residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines) - No decimate filter

Table 4.4: CLIE, CLOE and DIDIM estimates - Decimate filter

	$\hat{\theta}_{CLIE}^T$	$\hat{\theta}_{CLOE}^T$	$\hat{\theta}_{DIDIM}^T$
$zz_{1r}$	1.25 (1.52 %)	1.22 (2.69 %)	1.25 (1.23 %)
$fv_1$	7.96 (0.91 %)	8.18 (1.39 %)	7.99 (0.75 %)
$fc_1$	7.21 (2.84 %)	6.34 (4.22 %)	7.11 (2.36 %)
$xx_{2r}$	-0.47 (3.72 %)	-0.42 (5.95 %)	-0.48 (3.06 %)
$xz_{2r}$	-0.16 (6.82 %)	-0.17 (7.92 %)	-0.15 (6.01 %)
$zz_{2r}$	1.09 (1.37 %)	1.08 (2.05 %)	1.09 (1.12 %)
$mx_{2r}$	2.24 (4.05 %)	2.43 (8.57 %)	2.22 (3.32 %)
$fv_2$	5.47 (1.55 %)	5.68 (2.66 %)	5.45 (1.24 %)
$fc_2$	8.35 (2.36 %)	7.40 (3.38 %)	8.35 (1.85 %)
$xx_{3r}$	0.13 (11.1 %)	0.18 (14.5 %)	0.13 (8.97 %)
$zz_{3r}$	0.12 (10.3 %)	0.11 (17.8 %)	0.11 (8.47 %)
$my_{3r}$	-0.59 (2.70 %)	-0.53 (6.38 %)	-0.59 (2.11 %)
$ia_3$	0.09 (11.1 %)	0.09 (17.7 %)	0.09 (8.61 %)
$fv_3$	1.94 (2.14 %)	1.93 (4.51 %)	1.94 (1.58 %)
$fc_3$	6.47 (2.19 %)	5.79 (3.22 %)	6.47 (1.59 %)
$mx_4$	-0.03 (26.9 %)	<b>0.01</b> (124 %)	-0.03 (18.8 %)
$ia_4$	0.03 (9.33 %)	0.03 (12.8 %)	0.03 (7.78 %)
$fv_4$	1.11 (1.81 %)	1.15 (2.52 %)	1.11 (1.48 %)
$fc_4$	2.44 (2.97 %)	2.17 (5.00 %)	2.42 (2.43 %)
$my_{5r}$	-0.04 (14.5 %)	-0.06 (13.7 %)	-0.03 (11.0 %)
$ia_5$	0.04 (12.1 %)	0.05 (15.4 %)	0.04 (9.78 %)
$fv_5$	1.82 (2.33 %)	1.92 (4.20 %)	1.82 (1.84 %)
$fc_5$	3.01 (3.72 %)	2.26 (6.00 %)	2.99 (2.92 %)
$ia_6$	0.01 (20.2 %)	0.01 (20.9 %)	0.01 (18.7 %)
$fv_6$	0.66 (2.07 %)	0.67 (2.85 %)	0.65 (1.59 %)
$fc_6$	0.18 (19.5 %)	<b>0.042</b> (315 %)	0.25 (11.6 %)
$fv_{m6}$	0.60 (2.19 %)	0.57 (5.66 %)	0.60 (1.67 %)
$fc_{m6}$	1.94 (4.50 %)	1.96 (9.84 %)	1.92 (3.44 %)

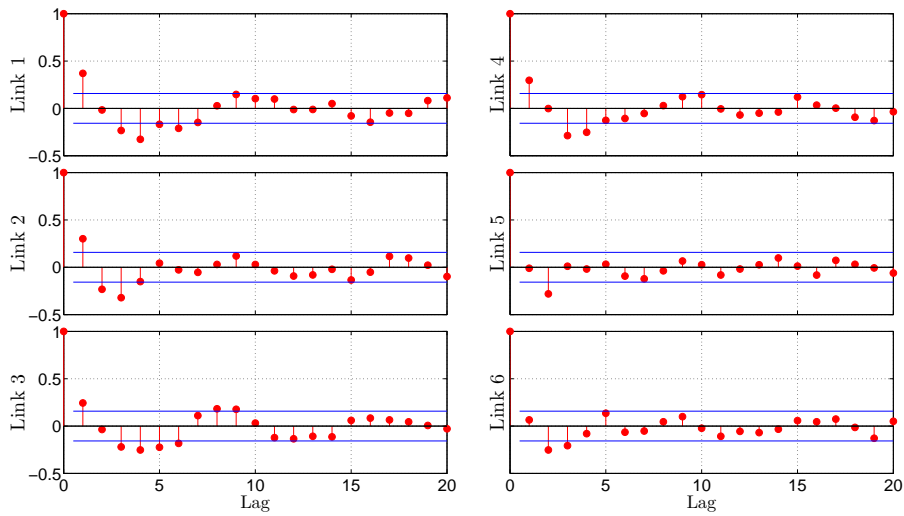
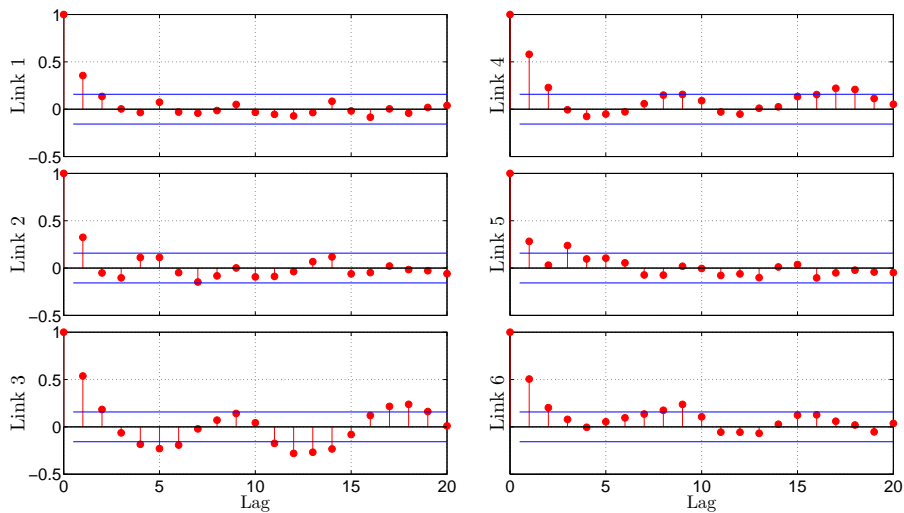
and DIDIM residuals can be considered as almost serially uncorrelated, and so are the CLIE ones. The differences between the intervals sizes compared with Figures 4.2 and 4.3 are due to the number of samples considered for each methods. If the methods without decimate filter take into account  $n_m = 34500$  sampling points for the estimation, the others consider  $N = n_m/n_d = 276$  sampling points due to the downsampling.

The DIDIM method has the advantage of calling only one time the simulator at each optimisation step thanks to the PLR assumption. In the opposite, the CLIE method must call the simulator  $b + 1$  times the simulator to estimate the jacobian matrix with finite differences. This explains the lower computing time of the DIDIM method in Table 4.5.

The experimental results show the advantages of considering the input (i.e. torque) signal for the identification. If the PLR assumption is admissible, the

Table 4.5: CLIE, CLOE and DIDIM optimisation parameters

	CLIE	CLOE	DIDIM
$\mu_{max}$	1589	1.34	1571
$\mu_{min}$	1.3	$1.74 \cdot 10^{-4}$	1.3
Conditioning number	1178	7622	1165
Computing time	25s	23s	1s

Figure 4.4: CLOE residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines) - Decimate filterFigure 4.5: DIDIM residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines) - Decimate filter

DIDIM method should be preferred to save computation time.

## 4.2 Robustness to modeling errors of the DIDIM and IDIM-IV methods

In the previous section, the interest of considering the input torque for the identification was demonstrated. Furthermore, it appeared that the DIDIM method is the most effective way of applying OEM to robot systems. This method, presented in section 2.3.4, is based on the simulation of the DDM and relies on the input torque like the IDIM-IV method presented in section 2.3.3. In this section, we investigate the relations between those methods.

### 4.2.1 Similarity between the methods

At first glance, by looking at the estimation formula (2.33) and (2.36), the IDIM-IV and the DIDIM methods are very similar. They indeed rely on the simulation of the DDM and their estimations are iterative. In practice, they use the same convergence criterion and the same initialisation, which is based on CAD values.

Assuming that there is no modelling error and that the IDM is well specified, we obtain

$$\mathbf{Z}(\widehat{\boldsymbol{\theta}}_{DIDIM}^{it}) = \mathbf{Z}(\widehat{\boldsymbol{\theta}}_{IV}^{it}) = \mathbf{X}_{nf}, \quad (4.23)$$

where  $\mathbf{X}_{nf}$  is the noise free component of the observation matrix  $\mathbf{X}$  defined by (2.21). Equivalently, we have at each sampling time:

$$\zeta_{F_p}(t, \widehat{\boldsymbol{\theta}}_{IV}^{it}) = \zeta_{F_p}(t, \widehat{\boldsymbol{\theta}}_{DIDIM}^{it}) = \phi_{F_p}^{nf}(t),$$

where  $\phi_{F_p}^{nf}$  is the noise free component of  $\phi_{F_p}$ . It is indeed assumed that the filtered observation matrix can be divided such as  $\phi_{F_p} = \phi_{F_p}^{nf} + \tilde{\phi}_{F_p}$ . The noise free and noisy parts are uncorrelated, i.e.  $\mathbb{E} \left[ \left( \phi_{F_p}^{nf} \right)^T \tilde{\phi}_{F_p} \right] = 0$ , with  $\mathbb{E} \left[ \tilde{\phi}_{F_p} \right] = 0$ . This assumption deserves some explanations that are provided in [Janot 2017] and recalled here. The separation between a noise-free and a noisy component comes from the fact that the noise is mostly contained in the joint accelerations. As it is highlighted in section 6.1.2, due to the ISO standard, the output joint positions and their measurements can be considered as noise-free

$$\mathbf{q}_m \approx \mathbf{q}_{nf}. \quad (4.24)$$

With  $\mathbf{n}_q$ , the position noise defined in (4.12), and  $T_s$ , the sampling time, the noises introduced with finite differences to joint velocities and accelerations are respectively  $\mathbf{n}_q/T_s$  and  $\mathbf{n}_q/T_s^2$ . In practice, the sampling frequency is greater than 100 Hz. Therefore, the following relation can be expected:

$$\mathbf{n}_q/T_s^2 \gg \mathbf{n}_q/T_s \gg \mathbf{n}_q.$$

This relation explains that the noises contained in the regressors linear with respect to the joint accelerations are dominant over those contained in the basis functions linear with respect to the joint positions and velocities. For the basis functions that are nonlinear combinations of joint positions and velocities, this requires further investigations. With respect to (2.3) and (2.4), it appears that the matrix  $\mathbf{M}$  and the vector  $\mathbf{Q}$  contain sine and cosine basis functions of the position. Regarding (4.24), we obtain

$$\cos(\mathbf{q}_m) \approx \cos(\mathbf{q}_{nf}) \quad \text{and} \quad \sin(\mathbf{q}_m) \approx \sin(\mathbf{q}_{nf}). \quad (4.25)$$

Consequently, one has  $\mathbf{M}(\mathbf{q}_m) \approx \mathbf{M}(\mathbf{q}_{nf})$  and  $\mathbf{Q}(\mathbf{q}_m) \approx \mathbf{Q}(\mathbf{q}_{nf})$ . The last nonlinearities to consider are included in the vector of centrifugal and Coriolis torques. According to Appendix A, only the joint velocities can be squared. For each joint  $j$ , we then have

$$\dot{q}_{m_j}^2 = \left(\dot{q}_{nf_j} + n_{q_j}/T_s\right)^2 = \dot{q}_{nf_j}^2 + n_{q_j}/T_s \left(n_{q_j}/T_s + 2\dot{q}_{nf_j}\right). \quad (4.26)$$

Because the trajectories are sufficiently exciting, it can be stated  $\dot{q}_{nf_j} \gg n_{q_j}/T_s$ , which yields

$$\dot{q}_{m_j}^2 \approx \dot{q}_{nf_j}^2 + 2\dot{q}_{nf_j}n_{q_j}/T_s \approx \dot{q}_{nf_j} \left(\dot{q}_{nf_j} + 2n_{q_j}/T_s\right) \approx \dot{q}_{nf_j}^2. \quad (4.27)$$

That leads to the relation  $\mathbf{C}(\mathbf{q}_m, \dot{\mathbf{q}}_m) \approx \mathbf{C}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf})$ <sup>1</sup>. This reasoning proves that the division of the observation matrix in two parts is admissible.

Coming back to our comparison and assuming there are enough sampling points, i.e.  $N$  large enough, it comes out

$$\mathbb{E} \left[ \left( \boldsymbol{\phi}_{F_p}^{nf} \right)^T \tilde{\boldsymbol{\phi}}_{F_p} \right] \approx \frac{1}{N} \sum_{i=1}^N \left( \boldsymbol{\phi}_{F_p}^{nf}(t_i) \right)^T \tilde{\boldsymbol{\phi}}_{F_p}(t_i) \quad (4.28)$$

and

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{IV}^{it+1} &= \left[ \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{X} \right]^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{y}(\tau) \\ &= \mathbf{P}_{\phi_{F_p}}^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \boldsymbol{\zeta}_{F_p}^T(t_i, \hat{\boldsymbol{\theta}}_{IV}^{it}) \boldsymbol{\tau}_{F_p}(t_i) \right] \\ &\approx \mathbf{P}_{\phi_{F_p}^{nf}}^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \left( \boldsymbol{\phi}_{F_p}^{nf}(t_i) \right)^T \boldsymbol{\tau}_{F_p}(t_i) \right] \\ &= \left[ \mathbf{X}_{nf}^T \mathbf{X}_{nf} \right]^{-1} \mathbf{X}_{nf}^T \mathbf{y}(\tau) \\ &= \hat{\boldsymbol{\theta}}_{DIDIM}^{it+1}, \end{aligned} \quad (4.29)$$

<sup>1</sup>Exceptionally, the matrix  $\mathbf{C}$  designates the centrifugal and Coriolis effects instead of the controller

with

$$\begin{aligned}
\mathbf{P}_{\phi_{F_p}} &= \frac{1}{N} \sum_{i=1}^N \zeta_{F_p}^T(t_i, \hat{\boldsymbol{\theta}}_{IV}^{it}) \phi_{F_p}(t_i) \\
&= \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \left[ \phi_{F_p}^{nf}(t_i) + \tilde{\phi}_{F_p}(t_i) \right] \\
&\approx \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \phi_{F_p}^{nf}(t_i) = \mathbf{P}_{\phi_{F_p}^{nf}}.
\end{aligned}$$

Therefore, if there is no modelling error, the DIDIM and IDIM-IV methods are perfectly equivalent and unbiased. The unbiased property of the DIDIM estimator was originally shown in [Gautier *et al.* 2013a]. It is worth noting that the DIDIM estimation given by (2.36) is the IV solution raised in [Söderström & Stoica 1983] equation (3.43b) page 38. If we follow the authors' point of view, the DIDIM approach can be considered as a sort of *bootstrap* IV variant.

If there is a modelling error in the dynamic model of the robot arm, like a friction mismatch, this error is present in the observation matrix and by construction in the instrumental matrix. In this case, the DIDIM and IDIM-IV methods are also equivalent and biased.

#### 4.2.2 Disparity between the methods

The difference between those methods lies in the use of the observation matrix,  $\mathbf{X}$ , for the IDIM-IV method. That may be seen as a drawback since it requires the measurement of the position signals,  $\mathbf{q}_m$ , and the careful bandpass filtering, detailed in section 2.3.1, in order to obtain the joint velocities and accelerations. As stressed in [Gautier *et al.* 2013a], the DIDIM method has the advantage of requiring only the measured torques.

However, the use of  $\mathbf{X}$  can make the IDIM-IV method less sensitive to modelling errors, in the simulation model, than the DIDIM method. We do not consider modelling errors in the dynamic model but in the rest of the closed-loop system; e.g. drive gains, bandwidth of the controller, etc. In fact, if we made an error in  $\mathbf{Z}$ , that error would also be present in  $\mathbf{X}$  due to the construction of the instruments and, consequently, both the IDIM-IV and DIDIM estimates would be biased. In other words, we consider here that the simulator is biased but not  $\mathbf{X}$ . The simulator error can be located in the controller used for the simulation for instance. If there is such an error, the auxiliary model is biased and, therefore, also the simulated signals. Hence the assumption (4.23) does not hold and the simulation errors can be introduced such as

$$\begin{aligned}
\zeta_{F_p}(t, \hat{\boldsymbol{\theta}}_{IV}^{it}) &= \phi_{F_p}^{nf}(t) + \Delta \phi_{IV}^{it}(t), \\
\zeta_{F_p}(t, \hat{\boldsymbol{\theta}}_{DIDIM}^{it}) &= \phi_{F_p}^{nf}(t) + \Delta \phi_{DIDIM}^{it}(t).
\end{aligned} \tag{4.30}$$

Whatever the method used, one assumption can be made concerning the  $(n \times n)$  simulation error matrix  $\Delta\phi$ : it is uncorrelated with the measurement noise, i.e.

$$\mathbb{E} \left[ \left( \Delta\phi^{it} \right)^T \tilde{\phi}_{F_p} \right] = 0 \quad \text{and} \quad \mathbb{E} \left[ \left( \Delta\phi^{it} \right)^T \mathbf{v}_{F_p} \right] = 0, \quad (4.31)$$

with  $\mathbf{v}_{F_p}$  the noisy component of  $\boldsymbol{\tau}_{F_p}$ ; see (2.18). This assumption makes sense because the simulator is noise-free, as explained in section 2.3.3. From those definitions, two cases can be considered according to  $\Delta\phi$ .

The first case is when expectation of the error is null:

$$\mathbb{E} \left[ \Delta\phi_{IV}^{it} \right] = \mathbb{E} \left[ \Delta\phi_{DIDIM}^{it} \right] = 0.$$

Hence, it appears for IDIM-IV solution:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{IV}^{it+1} &= \mathbf{P}_{\phi_{F_p}}^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) + \Delta\phi_{IV}^{it}(t_i) \right)^T \boldsymbol{\tau}_{F_p}(t_i) \right] \\ &\approx \mathbf{P}_{\phi_{F_p}^{nf}}^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \boldsymbol{\tau}_{F_p}(t_i) \right] \\ &= \left[ \mathbf{X}_{nf}^T \mathbf{X}_{nf} \right]^{-1} \mathbf{X}_{nf}^T \mathbf{y}(\tau), \end{aligned} \quad (4.32)$$

with

$$\begin{aligned} \mathbf{P}_{\phi_{F_p}} &= \frac{1}{N} \sum_{i=1}^N \left[ \phi_{F_p}^{nf}(t_i) + \Delta\phi_{IV}^{it}(t_i) \right]^T \left[ \phi_{F_p}^{nf}(t_i) + \tilde{\phi}_{F_p}(t_i) \right] \\ &\approx \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \phi_{F_p}^{nf}(t_i) = \mathbf{P}_{\phi_{F_p}^{nf}}. \end{aligned}$$

Concerning the DIDIM method, there is a persistent term

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{DIDIM}^{it+1} &\approx \left( \mathbf{P}_{\phi_{F_p}^{nf}} + \Delta\mathbf{P} \right)^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \boldsymbol{\tau}_{F_p}(t_i) \right] \\ &= \left[ \mathbf{X}_{nf}^T \mathbf{X}_{nf} + N \cdot \Delta\mathbf{P} \right]^{-1} \mathbf{X}_{nf}^T \mathbf{y}(\tau), \end{aligned} \quad (4.33)$$

with

$$\Delta\mathbf{P} = \frac{1}{N} \sum_{i=1}^N \left( \Delta\phi_{DIDIM}^{it}(t_i) \right)^T \Delta\phi_{DIDIM}^{it}(t_i).$$

Equation (4.32) shows that the IDIM-IV method is robust against a modelling error located in the simulator with a null expected value. In the opposite, (4.33) shows that the DIDIM method is sensitive to such an error and its estimates are biased.

The second case is when the simulation error has a non zero expectation, i.e.  $\mathbb{E} \left[ \Delta\phi_{IV}^{it} \right] \neq 0$  and  $\mathbb{E} \left[ \Delta\phi_{DIDIM}^{it} \right] \neq 0$ . In this case, both methods are biased and so are the estimated parameters.



### 4.2.3 Experimental validations

In a first time, we illustrate the behaviour of the methods in the nominal case. That is to say that the auxiliary model is error-free. This auxiliary model is composed of the robot dynamic model, the actual controller, the gains of the actuators and the gear ratios. The DDM is simulated thanks to a Simulink<sup>®</sup> model integrated with the *ode3* solver. The methods are initialized with acceptable CAD values; see Table 4.4. The essential estimated parameters and relative standard deviation are summarized in Table 4.6. The IDIM-IV and the DIDIM methods respectively converged in 3 and 7 iterations. As expected, both methods estimate equivalent parameters with comparable standard deviations. That is confirmed by the satisfactory relative errors provided in Table 4.7. Furthermore, the estimated values are consistent with previous work on this robot arm; see Table 4.4. Those observations are consistent with the work of [Gautier *et al.* 2013a] and [Janot *et al.* 2014c]: the IDIM-IV and the DIDIM methods are appropriate to identify industrial robots.

In a second time, we highlight the difference between the IDIM-IV and DIDIM methods. Thus, an error is introduced in the auxiliary model: the gear ratios are taken equal to 80% of their actual values. Table 4.6 summarizes the estimated parameters and the relative standard deviations. The IDIM-IV and the DIDIM methods respectively converged in 3 and 7 iterations. At a first glance, looking at Table 4.7, the difference is not so significant. With a closer scrutiny, it appears that the inertia parameters are poorly estimated with the DIDIM method, like  $zz_{1r}$ ,  $zz_{2r}$  or  $ia_3$  for instance. Compared with the parameters previously estimated, there is a gap of almost 20 %. On the other hand, the IDIM-IV estimates are still consistent. However, the relative standard deviations seem a little optimistic, see the same parameters as previously. This can be explained by the calculation of the estimated parameters covariance matrix, which relies on the assumption that the instrumental matrix is an estimation of the noise free observation matrix; see e.g. [Young 2011] equation (7.31) page 211. In fact, the exact covariance matrix for the IV estimate is obtained as:

$$\Sigma(\hat{\theta}_{IV}) = \frac{1}{N} \mathbf{P}_{\phi_{F_p}}^{-1} \left[ \frac{1}{N} \sum_{i=1}^N \zeta_{F_p}^T(t_i, \hat{\theta}_{IV}) \mathbf{\Lambda}_{\tau}^{-1}(t_i) \zeta_{F_p}(t_i, \hat{\theta}_{IV}) \right] \mathbf{P}_{\phi_{F_p}}^{-1}, \quad (4.34)$$

with

$$\mathbf{P}_{\phi_{F_p}} = \frac{1}{N} \sum_{i=1}^N \zeta_{F_p}^T(t_i, \hat{\theta}_{IV}) \phi_{F_p}(t_i)$$

and  $\mathbf{\Lambda}_{\tau}$  is the error covariance matrix defined in (2.25). Usually, assumption (4.23)

holds and it comes out:

$$\begin{aligned}\Sigma(\hat{\theta}_{IV}) &= \frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \Lambda_{\tau}^{-1}(t_i) \phi_{F_p}^{nf}(t_i) \right]^{-1} \\ &= \frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N \zeta_{F_p}^T(t_i, \hat{\theta}_{IV}) \Lambda_{\tau}^{-1}(t_i) \zeta_{F_p}(t_i, \hat{\theta}_{IV}) \right]^{-1}.\end{aligned}\quad (4.35)$$

Equation (4.35) significantly simplifies the computation of the covariance. In practice, this approximate formula is used for the IDIM-IV method; see (2.34). Nonetheless, by using the approximate relation and although  $\mathbf{E}[\Delta\phi_{IV}^{it}] = 0$ , we introduced a persistent error:

$$\begin{aligned}& \frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N \left( \zeta_{F_p}(t_i, \hat{\theta}_{IV}) + \Delta\phi_{IV}(t_i) \right)^T \Lambda_{\tau}^{-1}(t_i) \left( \zeta_{F_p}(t_i, \hat{\theta}_{IV}) + \Delta\phi_{IV}(t_i) \right) \right]^{-1} \\ & \neq \frac{1}{N} \left[ \frac{1}{N} \sum_{i=1}^N \left( \phi_{F_p}^{nf}(t_i) \right)^T \Lambda_{\tau}^{-1}(t_i) \phi_{F_p}^{nf}(t_i) \right]^{-1} = \Sigma(\hat{\theta}_{IV})\end{aligned}\quad (4.36)$$

Therefore, the approximate covariance should not be used if there is a risk of modelling error in the simulator. This practical example illustrates the impact of a biased simulator on the estimation when the simulation error has a zero expectation. In this case, we have illustrated a better robustness of the IDIM-IV method compared with the DIDIM one.

## 4.3 Miscellaneous remarks

### 4.3.1 Closed-loop aspects of the robot identifications methods

In the previous sections, we considered different methods for robot identification: the CLOE, CLIE, DIDIM and IDIM-IV methods. Like the usual IDIM-LS method described in section 2.3.2, those methods do not fall within the definitions of the closed-loop approaches given in section 3.1. The open-loop parameters are indeed identified directly from the measured data. There is no identification of the whole closed-loop model (indirect approach) or of an augmented system (joint input-output approach). Nevertheless, they do not belong to direct approach neither since the controller is taken into account for the simulation of the DDM, in the case of the OEM and the IDIM-IV method, and to tune the filters, for all the methods including the IDIM-LS one. This may explain why the robot identification methods and especially the IDIM-LS one are not well perceived by the automatic control community.

### 4.3.2 Considerations on the robot identification with OEM

In section 4.1, we considered three pseudo-OEM for robot identification: the CLOE, CLIE and DIDIM methods. They are qualified as pseudo-OEM because they need

Table 4.6: Estimated parameters and relative standard deviations - Robustness validation

Param.	Nominal Identification		Biased Simulator	
	IDIM-IV	DIDIM	IDIM-IV	DIDIM
$zz_{1_r}$	1.25 (1.27 %)	1.25 (1.53 %)	1.25 (0.65 %)	<b>0.96</b> (1.22 %)
$fc_1$	7.23 (2.09 %)	7.27 (2.81 %)	7.16 (1.75 %)	7.49 (2.81 %)
$fv_1$	7.95 (0.67 %)	7.95 (0.90 %)	7.98 (0.43 %)	6.13 (0.95 %)
$xx_{2_r}$	-0.47 (2.96 %)	-0.49 (3.61 %)	-0.48 (1.52 %)	-0.24 (4.46 %)
$xz_{2_r}$	-0.16 (5.02 %)	-0.15 (7.33 %)	-0.15 (2.95 %)	-0.10 (7.36 %)
$zz_{2_r}$	1.09 (1.02 %)	1.08 (1.39 %)	1.08 (0.63 %)	<b>0.85</b> (1.29 %)
$mx_{2_r}$	2.25 (2.80 %)	2.21 (4.08 %)	2.27 (1.66 %)	1.37 (4.61 %)
$fc_2$	8.32 (1.61 %)	8.46 (2.33 %)	8.36 (1.32 %)	8.47 (2.31 %)
$fv_2$	5.46 (1.08 %)	5.40 (1.57 %)	5.44 (0.70 %)	4.19 (1.59 %)
$xx_{3_r}$	0.13 (9.13 %)	0.13 (11.1 %)	0.14 (5.19 %)	0.07 (14.4 %)
$zz_{3_r}$	0.12 (8.62 %)	0.11 (11.1 %)	0.13 (3.78 %)	0.17 (4.16 %)
$my_{3_r}$	-0.59 (2.37 %)	-0.59 (2.68 %)	-0.56 (1.42 %)	-0.47 (2.30 %)
$ia_3$	0.09 (9.27 %)	0.10 (10.6 %)	0.09 (4.98 %)	<b>0.03</b> (22.5 %)
$fc_3$	6.53 (2.05 %)	6.57 (2.14 %)	6.57 (1.67 %)	6.54 (2.11 %)
$fv_3$	1.93 (2.04 %)	1.92 (2.15 %)	1.92 (1.36 %)	1.56 (2.11 %)
$mx_4$	-0.03 (28.3 %)	-0.03 (21.4 %)	-0.04 (14.9 %)	-0.03 (19.5 %)
$ia_4$	0.03 (14.0 %)	0.03 (9.38 %)	0.03 (9.61 %)	0.02 (9.12 %)
$fc_4$	2.50 (5.73 %)	2.45 (3.08 %)	2.49 (4.75 %)	2.59 (2.47 %)
$fv_4$	1.10 (3.56 %)	1.11 (1.88 %)	1.10 (2.40 %)	0.88 (1.65 %)
$my_{5_r}$	-0.04 (16.1 %)	-0.04 (12.8 %)	-0.03 (10.5 %)	-0.02 (19.4 %)
$ia_5$	0.04 (13.9 %)	0.04 (12.2 %)	0.04 (8.26 %)	0.03 (11.5 %)
$fc_5$	3.05 (4.32 %)	3.01 (3.78 %)	3.02 (3.59 %)	2.88 (3.17 %)
$fv_5$	1.80 (2.77 %)	1.82 (2.38 %)	1.82 (1.78 %)	1.47 (1.88 %)
$ia_6$	0.01 (12.9 %)	0.01 (15.1 %)	0.01 (8.31 %)	0.01 (14.6 %)
$fc_6$	0.25 (35.9 %)	0.26 (32.5 %)	0.26 (32.3 %)	0.29 (29.4 %)
$fv_6$	0.65 (1.90 %)	0.64 (2.19 %)	0.64 (1.54 %)	0.51 (1.65 %)
$fc_{m6}$	1.95 (4.86 %)	1.95 (4.49 %)	1.92 (4.84 %)	1.91 (3.62 %)
$fv_{m6}$	0.60 (2.26 %)	0.59 (2.24 %)	0.60 (2.09 %)	0.47 (1.72 %)

Table 4.7: Direct comparison - Relative errors - Robustness validation

Case	IDIM-IV	DIDIM
Nominal	5.59 %	5.57 %
Biased simulator	5.66 %	9.71 %

Table 4.8: Sensitivity properties of the aircraft example

	CLIE	CLOE
$\mu_{max}$	2.27	$1.30 \cdot 10^3$
$\mu_{min}$	$7.83 \cdot 10^{-8}$	$1.22 \cdot 10^{-2}$
Conditioning number	$2.91 \cdot 10^7$	$1.07 \cdot 10^5$

the parallel filter in order to provide estimates with uncorrelated residuals. We have highlighted the interest of considering the input signal rather than the output one for identification. That works has significantly expanded the study initiated in [Janot *et al.* 2014a] where the CLOE method was compared to the DIDIM one. This better sensitivity property can be extended to the IDIM-LS and IDIM-IV methods. In their case, the sensitivity to consider is the one of the actual system instead of the simulated one.

Two elements are critical for the sensitivity analysis, the assumption of a good tracking (4.8) and the underlying assumption of a good actuation of the system, i.e. one link, one sensor, one actuator. Without those two elements, the results may not be validated. This is a limitation to the results which cannot be extended to any system. In fact, one can think about a system with a poor tracking behaviour, the static control gain would be low, even lower than the unity. In the case of an under actuated system, we may have less information by considering the inputs.

To illustrate such a case, an example of an aircraft's longitudinal control law is borrowed from [Cook 2007], example 11.6, see Figure 4.6. This is an example based on the McDonnell Douglas F-4C Phantom flying at Mach 1.1 that requires a stability augmentation system. The appropriate gain found to meet the requirements has a low value. We consider the parameters modelling the short period mode, i.e.  $\boldsymbol{\theta} = [z_w \ z_q \ m_w \ m_q \ z_{\delta_m} \ m_{\delta_m}]^T$ , see Eq. (6.14) of the previous reference for further information on the model. The closed-loop system is excited with a doublet which can be seen as a square approximation of a sine wave. Its frequency is the one of the mode (1.3 Hz) and its amplitude is 5 degrees. To illustrate the CLOE method, a joint output  $\mathbf{y} = [w \ q]^T$  is considered and the input  $u = \delta_m$  is considered for the CLIE method. Table 4.8 gives the information about the input and output sensitivities. It appears with this example that the CLOE method may have larger singular values and then be more robust. The conditioning numbers of this example are relatively large because of a modelling choice. The system's states would have been better balanced with the angle of attack instead of the normal speed  $w$ . Another solution could consist in using a weighted criterion like the ML one. This short example illustrates the difficulty to extend the robotic results to other systems.

### 4.3.3 Remarks on the linearity

Chapter 2 has introduced the robot arm dynamic models and the related identification techniques. The linearity of the torques with respect to the base parameters

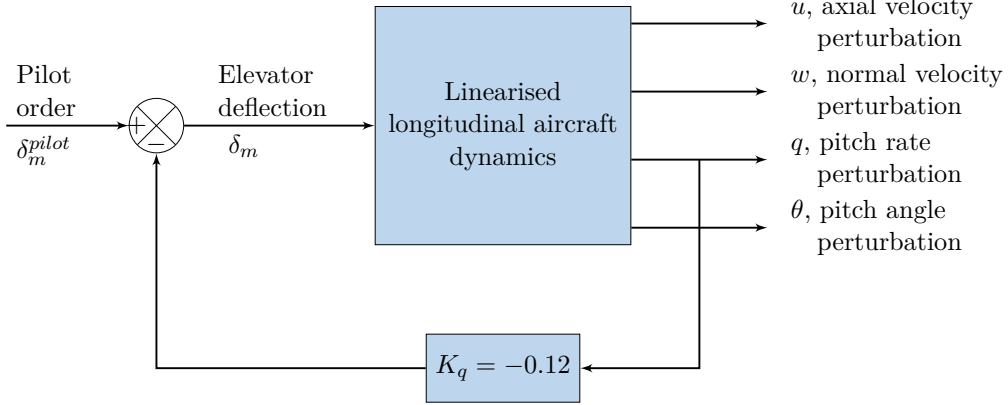


Figure 4.6: Block diagram of an aircraft's longitudinal feedback

has been outlined. Section 4.1 has shown the utility of the linearity for the computing time by transforming the nonlinear optimisation problem in a linear one. One can wonder if the output could be linear with respect to some parameters to improve the computation. The work undertaken in [Brunot *et al.* 2015] with a 1 DOF system shows that, if another parametrization is possible, the physical interpretation of the parameters is lost.

Regarding the PLR assumption, it is worth coming back to the LS estimation. The quadratic criterion is defined such as:

$$\begin{aligned} j_{OLS}(\boldsymbol{\theta}) &= (\mathbf{y}(\boldsymbol{\tau}) - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y}(\boldsymbol{\tau}) - \mathbf{X}\boldsymbol{\theta}) \\ &= \mathbf{y}^T(\boldsymbol{\tau})\mathbf{y}(\boldsymbol{\tau}) - 2\mathbf{y}^T(\boldsymbol{\tau})\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}. \end{aligned} \quad (4.37)$$

It should be notice that  $\mathbf{y}^T(\boldsymbol{\tau})\mathbf{X}\boldsymbol{\theta}$  is equal to  $\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y}$  because they give the same scalar value. To find the minimum of this cost function, its derivative with respect to  $\boldsymbol{\theta}$  must computed:

$$\begin{aligned} \frac{\partial j_{OLS}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left( \mathbf{y}^T(\boldsymbol{\tau})\mathbf{y}(\boldsymbol{\tau}) - 2\mathbf{y}^T(\boldsymbol{\tau})\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \right) \\ &= \frac{\partial}{\partial \boldsymbol{\theta}} \left( -2\mathbf{y}^T(\boldsymbol{\tau})\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \right). \end{aligned} \quad (4.38)$$

This first simplification comes from the fact that the measurement is independent of the parameters. If we assume also that  $\mathbf{X}$  is independent of the parameters, it comes out:

$$\frac{\partial j_{OLS}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y}(\boldsymbol{\tau}) + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}. \quad (4.39)$$

At the optimum, the derivative is zero and the Ordinary LS solution is given by:

$$\hat{\boldsymbol{\theta}}_{OLS} = [\mathbf{X}^T\mathbf{X}]^{-1} \mathbf{X}^T\mathbf{y}(\boldsymbol{\tau}), \quad (4.40)$$

which is (2.22) given in section 2.3.2. We do not discuss the fact that this optimum

is a minimum and not a maximum. It thus appears that the IDIM-LS method implicitly relies on the PLR assumption. Otherwise the derivative of  $\mathbf{X}$  with respect to  $\boldsymbol{\theta}$  would not be zero and the estimate would be different.

## 4.4 Conclusions

This chapter has been dedicated to evaluate the robustness of the methods introduced in Chapter 2 and, more specifically, the robustness of the methods relying on the simulation of the auxiliary model. In a first time, it has been shown that it is more appropriate to consider the input torque for robot identification, even in an OEM framework. The CLIE method is indeed more robust to modelling errors thanks to its better sensitivity. Even though, in the ideal case where there is no modelling error, it produces as accurate estimates as the CLOE method. If the PLR assumption is met, the DIDIM method should be employed to save computation time. The experimental results have highlighted that, if it is possible to obtain consistent estimates in a strict OEM framework, the decimate filter is required to provide a consistent estimate of the covariances.

In second time, we have shown that the IDIM-IV method can be more robust than the DIDIM method, although they are really similar. The IDIM-IV method can indeed tolerate some error in the auxiliary model as long as the expected value of the resulting error in the IDM is null. If the error is located directly in the IDM, both methods are identically biased.

All the identification methods need *a priori* knowledge concerning the system. For instance, the IDIM-IV requires the system's bandwidth to tune the bandpass filter. That knowledge is also needed by the DIDIM and the IDIM-IV methods in order to set the cut-off frequency of the decimate filter. Furthermore, the actual controller is needed to simulate the auxiliary model. In the next chapter, efforts are ongoing to reduce the dependency to *a priori* knowledge.

# Robot system identification without the knowledge of the controller and the bandwidths

---

In the previous chapter, the robustness of the IDIM-IV and DIDIM methods has been investigated. It appeared that those methods rely on a careful prefiltering methodology which requires the knowledge of the bandwidths. Furthermore, they need the knowledge of the controller for the simulation of their auxiliary model. That may be an issue if the robot system has a proprietary controller. It emerges here an usual dilemma of system identification: to identify a system correctly, we need a good knowledge of this system. In this chapter, the robot identification methodologies are revisited in order to avoid *a priori* knowledge as much as possible. This work is a new way of dealing with robot identification that has been presented in [Brunot *et al.* 2017b] as well as in [Brunot *et al.* In Press 2017].

The chapter is organised as follows. The derivatives estimation is studied in section 5.1. Then, the issue of an unknown controller is stated in section 5.2. The selected approaches are then applied on the TX40 example in section 5.3 before concluding in section 5.4 on future perspectives for the developed methodologies.

---

<b>5.1</b>	<b>Joint derivatives estimation</b>	<b>72</b>
5.1.1	Derivatives estimation from noisy signals	72
5.1.2	Integrative random walk smoother	73
5.1.3	Application to robot identification	76
5.1.4	Numerical differentiation tests	80
<b>5.2</b>	<b>Controller identification</b>	<b>81</b>
5.2.1	Closed-loop system identification with an unknown controller	81
5.2.2	Controller parametric identification	83
5.2.3	Controller non-parametric identification	83
<b>5.3</b>	<b>Experimental validation</b>	<b>85</b>
5.3.1	The IDIM-LS method with the IRWSM technique	85
5.3.2	The IDIM-IV method with the IRWSM technique	87
5.3.3	The IDIM-IV method with an unknown controller	90
<b>5.4</b>	<b>Conclusions</b>	<b>94</b>

---

## 5.1 Joint derivatives estimation

### 5.1.1 Derivatives estimation from noisy signals

Many researches are ongoing concerning the numerical differentiation issue, see e.g. [Diop *et al.* 1994, Vasiljevic & Khalil 2006, Dridi 2011, Sidhom 2011] and the references given therein. Some of them are summarized here to have an overview of the possibilities. Nonetheless, the goal is here to suggest a practical and straightforward technique. The idea is to find a solution that requires neither *a priori* knowledge of the system nor a complex computational process.

The issue of estimating unknown system states is not recent for the automatic control community. It has been indeed the subject of a long-established literature about *observers*. The goal is to provide the signals derivatives to the controller in order to reduce the number of required sensors. The problem has been introduced by Luenberger [Luenberger 1971]. In the case of a noisy signal, a stochastic observer like the Kalman one may be more appropriate; see e.g. [Singer 1970]. There is also a large and active research on nonlinear observers; see e.g. [Dabroom & Khalil 1997, Besançon 2007]. For example, the Extended Kalman Filter (EKF) can be considered to estimate the states of a nonlinear stochastic system, like in [Ljung 1979]. Whether the observer is deterministic or stochastic, it relies on the model of the considered system. In other words, the observer needs the system's model and its parameters to provide unbiased estimations of the derivatives. Therefore, an observer does not appear as a solution to estimate the joint derivatives in order to estimate the dynamic parameters. Although a solution could consist in estimating the states and the parameters at the same time in a similar manner to [Gautier & Poignet 2001], a less complex solution is sought. In fact, we do not want such a solution in order to avoid coupling effects.

Another approach consists in relying on the signals only and not on the model. It is then more appropriate to talk about *differentiator*. The usual bandpass filter belongs to this approach; see section 2.3.1. Another method is based on the polynomial interpolation of the signal over a sliding window; see e.g. section 11.2.2 of [Klein & Morelli 2006]. The signal derivative is approximated by the derivative of the polynomial.

It exists also the algebraic differentiators which have been recently revisited; see e.g. [Fliess & Sira-Ramírez 2003]. The idea is to estimate the signal derivatives thanks to successive integrations of the measured signal. It is based on a truncated Taylor expansion of the signal. Thus, the practitioner has to provide the order of the truncation. Since the excitation trajectories used for robot identification are not really smooth, the required order of the truncation can be large.

Another family of techniques is based on the sliding mode theory introduced during the 1960s [Filippov 1960]. Among the developed techniques, we can mention the Super Twisting algorithm introduced in [Levant 1998]. Without going through



the theory, the estimation of the signal  $y$  and its first derivative is given by:

$$\hat{\dot{y}} = -\lambda_1 \cdot |\hat{y} - y|^{1/2} \cdot \text{sign}(\hat{y} - y) + z \quad (5.1)$$

$$\dot{z} = -\lambda_0 \cdot \text{sign}(\hat{y} - y). \quad (5.2)$$

Further details can be found in [Shtessel *et al.* 2014] for instance. In practice, the user has to tune the two parameters  $(\lambda_0, \lambda_1)$ . The major drawback of this technique is the phenomenon of *chattering*. It consists in high frequency oscillations due to the *sign* function. Some solutions exist like the use of a sigmoid function instead of the *sign* one. However, that introduces more parameters for the practitioner.

Finally, the solution we have selected is called the Integrative Random Walk Smoother (IRWSM). It has the practical advantage of not requiring *a priori* information. That solution is developed in the following section.

### 5.1.2 Integrative random walk smoother

This section is devoted to the introduction of the IRWSM approach in a general framework. The links with our robotic applications are made in section 5.1.3. Those parts take the work presented in [Brunot *et al.* In Press 2017].

#### The State Space Model: IRW

The IRWSM method is based on the well-known Kalman filter technique, in a discrete time framework. This technique is summarized in [Norton 1975, Young 2000]. It allows an off-line estimation of the states without using the dynamic model, unlike High Gain observers for instance. Eq. (5.3) defines the state vector, with  $x$  the position state and  $\nabla x$  the rate of change. Relation (5.4) is the state equation and relation (5.5) is the observation equation.

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ \nabla x(k) \end{bmatrix} \quad (5.3)$$

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{D}\boldsymbol{\kappa}(k-1) \quad (5.4)$$

$$y(k) = \mathbf{h}(k)\mathbf{x}(k) + e(k) \quad (5.5)$$

With,

$$\mathbf{A} = \begin{bmatrix} p_1 & p_2 \\ 0 & p_3 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} p_4 & 0 \\ 0 & p_5 \end{bmatrix} \quad (5.6)$$

$\mathbf{h}$  is the  $(1 \times 2)$  row observation vector.  $\boldsymbol{\kappa}$  is the state noise, assumed to be white and zero mean, with covariance matrix  $\mathbf{Q}_{\boldsymbol{\kappa}}$  (diagonal). The measurement noise  $e$  is also zero mean and white. Its covariance is written  $\sigma_e^2$ . This model, developed in [Young 2011], is named Generalized Random Walk (GRW). Many variants exist depending on the choice of the hyper-parameters  $\boldsymbol{\gamma} = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ \mathbf{Q}_{\boldsymbol{\kappa}_{11}} \ \mathbf{Q}_{\boldsymbol{\kappa}_{22}}]$ . For this study, the *Integrated Random Walk* (IRW:  $p_1 = p_2 = p_3 = p_5 = 1$ ,  $p_4 = 0$  and  $\mathbf{h} = [1 \ 0]$ ) is considered. In that case, since  $p_4 = 0$ , the term  $\mathbf{Q}_{\boldsymbol{\kappa}_{11}}$  has

no influence and is equal to  $\mathbf{Q}_{\kappa 22}$  in order to preserve the definite-positive property of the covariance matrix. Finally, the only remaining hyper-parameter is  $\mathbf{Q}_{\kappa 22}$ . As it will be seen later, its value may be estimated thanks to a ML optimization.

### The Kalman and FIS Equations

From the model previously described, a specific Kalman filter is implemented. First of all, it is associated with a backward Fixed Interval Smoother (FIS) to take advantage of the off-line process. Secondly, the filter and smoother equations are modified to avoid the knowledge of the observation noise variance,  $\sigma_e^2$ . In a classical Kalman Filter, this information is indeed required, like the covariance of the state noise,  $\mathbf{Q}_{\kappa}$ . Instead, all the equations are written as functions of the Noise Variance Ratio (NVR), which is defined by  $\mathbf{Q}_{nvr} = \mathbf{Q}_{\kappa}/\sigma_e^2$ . The algorithm described in [Young 2011] is summarized below.

*Prediction step:*

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{A}\hat{\mathbf{x}}(k-1) \quad (5.7)$$

$$\mathbf{P}(k|k-1) = \mathbf{A}\mathbf{P}(k-1)\mathbf{A}^T + \mathbf{D}\mathbf{Q}_{nvr}\mathbf{D}^T \quad (5.8)$$

*Correction step:*

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{g}(k)[y(k) - \mathbf{h}(k)\hat{\mathbf{x}}(k|k-1)] \quad (5.9)$$

$$\mathbf{g}(k) = \mathbf{P}(k|k-1)\mathbf{h}(k)[\mathbf{I} + \mathbf{h}(k)\mathbf{P}(k|k-1)\mathbf{h}^T(k)]^{-1} \quad (5.10)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{g}(k)\mathbf{h}(k)\mathbf{P}(k|k-1) \quad (5.11)$$

$$\mathbf{P}^*(k|k) = \hat{\sigma}_e^2\mathbf{P}(k|k) \quad (5.12)$$

*Smoothing step:*

$$\hat{\mathbf{x}}(k|N) = \mathbf{A}^{-1} [\hat{\mathbf{x}}(k+1|N) + \mathbf{D}\mathbf{Q}_{\kappa}\mathbf{D}^T\boldsymbol{\lambda}(k)] \quad (5.13)$$

$$\boldsymbol{\lambda}(k-1) = \left[ \mathbf{I} - \mathbf{P}^*(k|k) \frac{\mathbf{h}^T(k)\mathbf{h}(k)}{\hat{\sigma}_e^2} \right]^T \quad (5.14)$$

$$\left( \mathbf{A}^T\boldsymbol{\lambda}(k) - \frac{\mathbf{h}^T(k)}{\hat{\sigma}_e^2} [y(k) - \mathbf{h}(k)\mathbf{A}\hat{\mathbf{x}}(k-1|k-1)] \right)$$

with  $\boldsymbol{\lambda}(N) = 0$

$$\mathbf{P}^*(k|N) = \mathbf{P}^*(k|k) + \mathbf{P}^*(k)\mathbf{A}^T\mathbf{P}^*(k+1|k)^{-1} \quad (5.15)$$

$$[\mathbf{P}^*(k+1|N) - \mathbf{P}^*(k+1|k)]\mathbf{P}^*(k+1|k)^{-1}\mathbf{A}\mathbf{P}^*(k|k)$$

The combination of the Kalman filtering and the FIS based on the IRW model is referred to as the IRWSM. The observation noise covariance,  $\sigma_e^2$ , is estimated at the end of the filtering process in order to obtain the state covariance matrix,  $\mathbf{P}^*$ , for the smoothing process. By defining  $n_x$  the size of the state vector ( $n_x = 2$  for

the IRW), the estimation is given by:

$$\begin{aligned}\hat{\sigma}_e^2 &= \frac{1}{N - n_x} \sum_{k=n_x+1}^N \frac{(y(k) - \mathbf{h}(k)\hat{\mathbf{x}}(k|k-1))^2}{1 + \mathbf{h}(k)\mathbf{P}(k|k-1)\mathbf{h}^T(k)}, \\ &= \frac{1}{N - n_x} \sum_{k=n_x+1}^N \frac{\varepsilon^2(k)}{\nu(k)}.\end{aligned}\quad (5.16)$$

In the time domain, the first order derivative of the signal is then approximated as follows  $\frac{dx}{dt}(t_k) \approx \frac{\widehat{\nabla x}(k)}{t_{k+1}-t_k}$ , with  $\widehat{\nabla x}(k)$  the second term of the estimated state vector  $\hat{\mathbf{x}}(k|N)$ . Similarly,  $\mathbf{x}$  could be augmented with  $\nabla^2 x$  in order to estimate the second order derivative.

## Hyper-Parameters Optimisation

If the IRWSM is able to estimate the states, it remains the issue of the hyper-parameters and more specifically of the NVR. As it has been said, the user does not have to provide the observation noise covariance to the IRWSM method contrary to a classical Kalman filter. The hyper-parameters can indeed be estimated by maximizing the likelihood of the prediction error,  $\varepsilon(k)$ , defined in (5.16). It assumes that  $\varepsilon(k)$  is a zero mean, normally distributed, white noise sequence. Its variance is given by:

$$\begin{aligned}\mathbb{E} [\varepsilon^2(k)] &= \mathbb{E} [(\mathbf{h}(k)\mathbf{x}(k) - \mathbf{h}(k)\hat{\mathbf{x}}(k|k-1) + e(k))^2] \\ &= \mathbb{E} [(\mathbf{h}(k)\tilde{\mathbf{x}}(k) + e(k))^2] \\ &= \mathbf{h}(k)\mathbb{E} [\tilde{\mathbf{x}}(k)\tilde{\mathbf{x}}^T(k)]\mathbf{h}^T(k) + \mathbb{E} [e(k)] \\ &= \mathbf{h}(k)\mathbf{P}^*(k|k-1)\mathbf{h}^T(k) + \sigma_e^2 = \sigma_e^2\nu(k)\end{aligned}\quad (5.17)$$

where the parametric estimation error  $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)$  is independent of the noise  $e(k)$ ,  $\mathbf{P}^*(k|k-1) = \hat{\sigma}_e^2\mathbf{P}(k|k-1)$  and  $\nu(k)$  is defined in (5.16). In practice, we use the *concentrated* case of the log-likelihood because the first  $n_x$  observations are regarded as fixed and the NVR relation is considered; see e.g. [Durbin & Koopman 2012] for further details on these assumptions. The log-likelihood of the series  $\mathbf{Y}(N) = [y(n_x+1), \dots, y(N)]^T$  is defined such as:

$$\log L = \log p(\mathbf{Y}(N)) = \log \prod_{k=n_x+1}^N p(y(k)|\mathbf{Y}(k-1)), \quad (5.18)$$

where  $p$  is the probability density function defined in appendix (C.1). Assuming  $p(y(k)|\mathbf{Y}(k-1)) \sim \mathcal{N}(\mathbf{h}(k)\hat{\mathbf{x}}(k|k-1), \sigma_e^2\nu(k))$ , it comes out:

$$\begin{aligned} \log L &= -\frac{N-n_x}{2}\log(2\pi) - \frac{1}{2} \sum_{k=n_x+1}^N \left( \log(\sigma_e^2\nu(k)) + \frac{\varepsilon^2(k)}{\sigma_e^2\nu(k)} \right) \\ &= -\frac{N-n_x}{2}\log(2\pi) - \frac{N-n_x}{2}\log(\sigma_e^2) - \frac{1}{2} \sum_{k=n_x+1}^N \left( \log(\nu(k)) + \frac{\varepsilon^2(k)}{\sigma_e^2\nu(k)} \right) \end{aligned} \quad (5.19)$$

By maximising (5.18) with respect to  $\sigma_e^2$ , we obtain (5.16) and the log-likelihood becomes:

$$\log L = -\frac{N-n_x}{2}(\log(2\pi) + 1) - \frac{N-n_x}{2}\log(\hat{\sigma}_e^2) - \frac{1}{2} \sum_{k=n_x+1}^N \log(\nu(k)). \quad (5.20)$$

Finally, the optimisation problem is

$$\arg \min_{\gamma} L(\gamma) = (N-n_x)\log(\hat{\sigma}_e^2) + \sum_{k=n_x+1}^N \log(\nu(k)). \quad (5.21)$$

For further information on the hyper-parameters estimation with the likelihood optimisation, see e.g. [Harvey & Peters 1990].

### Practical implementation

From a practical point of view, the IRWSM algorithm is implemented in the function `irwsm` of the CAPTAIN Toolbox, developed by a team of Lancaster University; see [Taylor *et al.* 2007] and <http://captaintoolbox.co.uk>. Fortunately, the toolbox provides also a function called `irwsmopt` which estimates the hyper-parameters solving the optimisation problem (5.21). This toolbox thus allows the user to process the data from a system without *a priori* knowledge about it. Obviously, it does not prevent him to be vigilant on the results.

### 5.1.3 Application to robot identification

The idea for robot identification is to replace the usual bandpass filter by the IRWSM technique. The generic state space model defined by (5.4) and (5.5) is thus reconsidered from a robotic perspective.

#### Observation equation

Regarding the observation equation, the output  $y$  is replaced by the measured position of link  $j$ :  $q_{m_j}$ . By considering the link  $j$ , we have the following relation for the measured position:

$$q_{m_j}(t) = q_j(t) + \tilde{q}_j(t), \quad (5.22)$$

where  $q_j$  is the joint position and  $\tilde{q}_j$  is the sensor noise. According to section 2.2.3, that noise is white with a covariance  $\frac{1}{3}\Delta_e^2$ , where  $\Delta_e$  is the encoder resolution. Regarding the Stäubli<sup>®</sup> TX40, the encoders' resolution is  $2 \cdot 10^{-4}$  degrees per count. As it has been said, this system is representative of industrial robots. If the measurement noise is also assumed to be normally distributed, the observation relation (5.5) is valid and the noise standard deviation  $\sigma_e \approx \Delta_q$  can be estimated from the robot performance data sheet. Furthermore, the position  $q_j$  is equal to the state  $x$  defined by (5.3).

### State equation

Concerning the state equation (5.4), two cases can be thought in practice to retrieve the joint acceleration:

- Applying the algorithm twice with  $\mathbf{A}_{irwsm1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ ;
- Applying the algorithm once with  $\mathbf{A}_{irwsm2} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ .

The relevance of the random walk model is firstly investigated with  $\mathbf{A}_{irwsm1}$ . From (5.4), it comes out

$$\begin{bmatrix} q_j(k) \\ \nabla q_j(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_j(k-1) \\ \nabla q_j(k-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \kappa_{1,j}(k-1) \end{bmatrix}. \quad (5.23)$$

The second state  $\nabla q_j$  represents the velocity. This velocity information is not driven by dynamic equations but a random walk. In other words, the noise  $\eta_{1,j}$  encompasses the unmodelled dynamics; i.e. this is a modelling error. As explained in [Young 2011], this kind of models works for systems with slowly varying states. For robotic applications, the controller requires high frequency sampling in order to have large feedback gains and, as a result, a more robust control; see e.g. [Khalil & Dombre 2004]. For example, the robot considered in section 5.3 has a sampling rate of 5000 Hz. Nonetheless, the mechanical bandwidth is usually located below 10 Hz. Consequently, with an IRW model expressed at the robot sampling frequency, the assumption of a slowly varying state is met.

With the augmented model  $\mathbf{A}_{irwsm2}$ , the velocity is not driven by a random walk but the acceleration. As with the velocity, the assumption of a slowly varying state seems appropriate. From (5.4), it comes out

$$\begin{bmatrix} q_j(k) \\ \nabla q_j(k) \\ \nabla^2 q_j(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_j(k-1) \\ \nabla q_j(k-1) \\ \nabla^2 q_j(k-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \kappa_{2,j}(k-1) \end{bmatrix}. \quad (5.24)$$

### Numerical approximation

As explained previously, the random walks can be seen as modelling errors; i.e. the difference between the model and the real joint dynamics. From experimental signals, the modelling error information can be looked for in the tracking error:

$$\varepsilon_{tr_j}(t) = q_{r_j}(t) - q_{m_j}(t), \quad (5.25)$$

where  $q_{r_j}$  is the reference trajectory of link  $j$ . In fact, the tracking error includes the modelling errors as well as the dynamics of the control law. Assuming that the controller has been correctly designed to provide zero tracking error and that the system is in a steady state, the tracking error  $\varepsilon_{tr_j}$  can then be considered as the image of the modelling error. If the tracking error variation  $\Delta\varepsilon_{tr_j}(k) = \varepsilon_{tr_j}(k) - \varepsilon_{tr_j}(k-1)$  is assumed to be a white and normally distributed noise, we have  $\kappa_{1,j}(k) \approx \Delta\varepsilon_{tr_j}(k)$ . If the 2<sup>nd</sup> variation  $\Delta^2\varepsilon_{tr_j}$  is assumed to be a white and normally distributed noise, we have  $\kappa_{2,j}(k) \approx \Delta^2\varepsilon_{tr_j}(k)$ . Finally, the NVR of link  $j$  can be approximated by

$$\widehat{\text{NVR}}_j^{irwsm1} = \left( \frac{\text{std}(\Delta\varepsilon_{tr_j})}{\Delta_{e_j}} \right)^2 \quad (5.26)$$

$$\widehat{\text{NVR}}_j^{irwsm2} = \left( \frac{\text{std}(\Delta^2\varepsilon_{tr_j})}{\Delta_{e_j}} \right)^2 \quad (5.27)$$

where  $\Delta_{e_j}$  is the encoder resolution of link  $j$  and  $\text{std}(\cdot)$  is the standard deviation. Table 5.1 provides the estimation of the NVR considering both models. The coefficients  $\widehat{\text{NVR}}_j^{irwsm1}$  and  $\widehat{\text{NVR}}_j^{irwsm2}$  were estimated with the *irwsmopt* function. The data are the experimental ones that are used in section 5.3. The encoders' resolutions come from the data-sheet [Stäubli Favergues 2015]. Regarding the IRWSM 1, if there is not a perfect matching between the estimated values, the orders of magnitude are the same; except for link 6. However, for the model IRWSM 2, the estimated values are absolutely not comparable. This finding suggests that  $\Delta^2\varepsilon_{tr_j}$  cannot be seen as a white normally distributed noise. Figure 5.1 gives the histograms of  $\Delta\varepsilon_{tr_1}$  and  $\Delta^2\varepsilon_{tr_1}$  as well as normal distribution fits. For sake of clarity, only link 1 is presented. The second histogram confirms that the distribution of  $\Delta^2\varepsilon_{tr_1}$  is not normal.

This practical reasoning does not intend to replace the *irwsmopt* routine but to illustrate the relevance of the IRWSM technique for our application. The goal is still to provide an automatic solution to the practitioner. To summarize, the IRW model seems a valid solution for the estimation of the joint derivatives, assuming the modelling error committed is white and normally distributed. In this regard, the IRWSM 1 approach appears to be more appropriate.

Table 5.1: Estimated NVR

Link	1	2	3	4	5	6
$\Delta_{e_j}$ ( $10^{-3}$ deg)	0.057	0.057	0.122	0.114	0.122	0.172
$\text{NVR}_j^{irwsm1}$	0.1476	0.1567	0.5846	0.2940	1.1101	0.6645
$\widehat{\text{NVR}}_j^{irwsm1}$	0.1951	0.2222	0.1521	0.2162	0.7491	1.6106
$\text{NVR}_j^{irwsm2}$	2.5368	2.2643	3.6919	2.5581	4.9713	3.8572
$\widehat{\text{NVR}}_j^{irwsm2}$	0.0015	0.0019	0.0037	0.0104	0.0045	0.0165

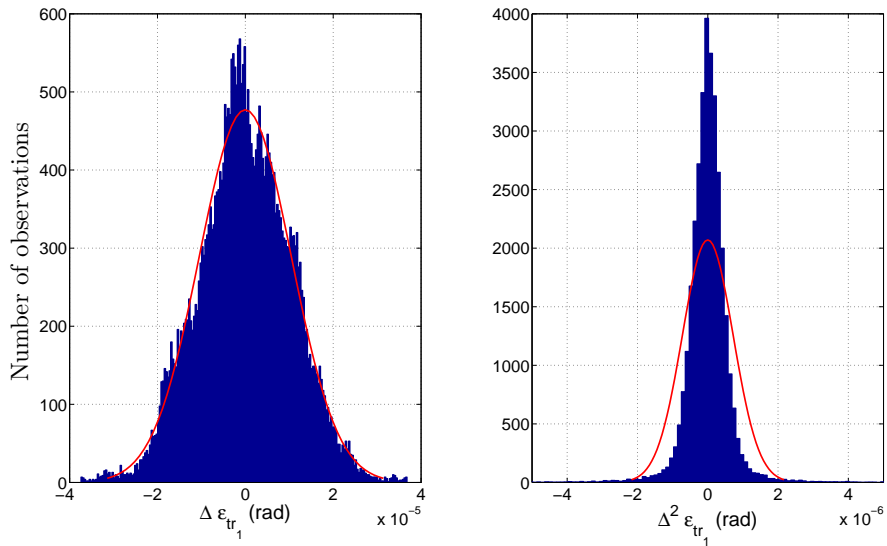


Figure 5.1: Histograms with normal distribution fits - Link 1

### 5.1.4 Numerical differentiation tests

In [Brunot *et al.* 2016b], the use of the IRWSM to differentiate the position signal of the rigid 1 DOF Electro-Mechanical Positioning System (EMPS) is studied. That is a robotic system simpler than the TX40. We recall here briefly its model to evaluate the differentiation with the IRWSM technique. The EMPS is modelled by

$$\tau(t) = M\ddot{q}(t) + F_v\dot{q}(t) + F_c\text{sign}(\dot{q}(t)), \quad (5.28)$$

where  $M$  is the inertia of the arm;  $F_v$  and  $F_c$  are respectively the viscous and Coulomb friction coefficients;  $q$ ,  $\dot{q}$  and  $\ddot{q}$  are respectively the position, velocity and acceleration;  $\tau$  is the motor force. The system is driven by a nested PD controller such as

$$\tau(t) = g_f k_v (k_p (q_r(t) - q(t)) - \dot{q}(t)), \quad (5.29)$$

where  $q_r$  is the reference trajectory and  $g_f$  is the electronic gain of the actuator. In practice, the controller gains have been chosen  $k_p = 160.2$  1/s and  $k_v = 242.5$  V.s/m. The actuator gain has previously been identified to 35.2 N/V.

The study is done thanks to simulated data. The values of parameters are chosen close to those which have previously been estimated for other works:  $M = 96.0$  kg,  $F_v = 205.0$  N/(m/s) and  $F_c = 20.0$  N. A measurement noise is added to the simulated output. To be realistic, the Signal to Noise Ratio (SNR) is taken equal to 100 dB; see section 6.1.2 for further justification. Monte Carlo Simulations (MCS) are run to evaluate the effect of this noise over the estimates.

From the noisy position signal generated by the simulator, three methods are compared. The first one is the standard bandpass filter introduced in section 2.3.1 and will be named by "Bandpass". The second method is the IRWSM implemented in the CAPTAIN Toolbox with the model  $\mathbf{A}_{irwsm1}$  and will be referred as "IRWSM 1". The last one is a variant of the IRWSM with  $\mathbf{A}_{irwsm2}$ , which allows estimating directly the second derivative without calling the algorithm twice. This approach will be named "IRWSM 2". Concerning the bandpass method, from previous works, it is known that the highest mode frequency is about 20 Hz. It should be noticed that this relation requires knowing the controller parameters and structure. The Butterworth filter is designed with a cut-off frequency equal to 40 Hz (i.e. twice the highest natural mode). To get the acceleration, its order is fixed at  $2+2=4$ .

Table 5.2 summarizes the results by providing the mean of the 100 relative errors for each estimate. The relative error of the signal  $s(t)$  is given by  $100 \cdot \|\mathbf{s}_{nf} - \hat{\mathbf{s}}\| / \|\mathbf{s}_{nf}\|$ , where  $\mathbf{s}_{nf}(t)$  is the noise free component of  $s(t)$  generated by the simulator. IRWSM 1 and 2 give very good results for the velocity and the acceleration, since the relative errors are very small and less than those of the bandpass approach. The methods seem to be equivalently effective in order to estimate the position.

This simple example tends to show the IRWSM technique is appropriate to differentiate signals from a robotic system. To go further, in section 5.3, we consider the TX40 and investigate the identification with the IDIM-LS and IDIM-IV



Table 5.2: Mean relative errors of the estimated signals for 100 Monte Carlo Simulations

Signal	Bandpass	Method	
		IRWSM 1	IRWSM 2
$q$	$6.38 \cdot 10^{-5}\%$	$5.78 \cdot 10^{-5}\%$	$5.10 \cdot 10^{-5}\%$
$\dot{q}$	$5.49 \cdot 10^{-1}\%$	$1.20 \cdot 10^{-3}\%$	$3.20 \cdot 10^{-3}\%$
$\ddot{q}$	$1.15 \cdot 10^0\%$	$1.34 \cdot 10^{-1}\%$	$2.64 \cdot 10^{-1}\%$

methods.

## 5.2 Controller identification

### 5.2.1 Closed-loop system identification with an unknown controller

In this section, the issue of the robot identification with an unknown controller is addressed. The idea is to study, for instance, a robot bought "off the shelf" without an open source controller in order to design our own controller. One solution may consist in using the IDIM-LS method, which does not require any knowledge on the controller. However, as shown in [Brunot *et al.* 2015], it is more interesting to consider the IDIM-IV method if the system bandwidth is not certain. That is likely the case for the first identification process. The drawback is that the IDIM-IV method requires the controller knowledge for the simulation of the DDM. That is why the identification of the controller is considered in this section.

The identification community has already considered the problem and developed specific methods for LTI systems. Few two steps approaches have been developed for this purpose. Their main features are summarized below.

- The two steps method of [Van Den Hof *et al.* 1992] consists in identifying the transfer function between  $\mathbf{q}_r$  and  $\boldsymbol{\tau}$  thanks to a PEM. Thereafter, the Transfer Function (TF) between  $\hat{\boldsymbol{\tau}}$  and  $\mathbf{q}_m$  is estimated. This method is referred to as PEM<sup>2</sup>.
- In [Forsell & Ljung 2000], the authors suggested to use a non-parametric method (Wiener filter) for the first step; the second is identical to PEM<sup>2</sup>. Hence, the measured  $\boldsymbol{\tau}$  is projected on the reference signal. Thus it is called *projection* method (PROJ). This solution has the advantage of not requiring any modelling of the controller.
- In [Young *et al.* 2009], the author proposed two variants of the two steps approach for the continuous time case, based on the Refined Instrumental Variable (RIV); see section 3.2. On the one hand, the first variant is perfectly similar to the PEM<sup>2</sup>: the RIV is employed instead of the PEM. On the other hand, the second variant consists in identifying the TF between  $\mathbf{q}_r$  and  $\boldsymbol{\tau}$  as well as the one between  $\mathbf{q}_r$  and  $\mathbf{q}_m$ . For the second step, the TF between  $\hat{\boldsymbol{\tau}}$  and  $\hat{\mathbf{q}}$  is estimated. Those methods are referred to as RIV<sup>2</sup>.

Table 5.3: Two-stages method for direct dynamic model

Method	1 <sup>st</sup> Step	2 <sup>nd</sup> Step
PEM <sup>2</sup>	Parametric Ident. - PEM	Parametric Ident. - PEM
PROJ	Non-Parametric Ident. - Wiener Filter	Parametric Ident. - PEM
RIV <sup>2</sup>	Parametric Ident. - RIV	Parametric Ident. - RIV
VCM	Design of 4 filters	Parametric Ident. - PEM

- The contribution of [Agüero *et al.* 2011] must also be noticed. Their idea is to introduce a *virtual controller*. This Virtual Controller Method (VCM) includes four auxiliary TF which allow a decorrelation of the input signal and the noise, without making any assumption on the controller.

Originally, the PEM<sup>2</sup> and PROJ methods were designed for DT models identification. However, they can be extended to CT models. Table 5.3 summarizes those methods.

When subjected to closer scrutiny, the previous methods appear to be unequal. In fact, if PROJ and VCM assume no knowledge on the controller at all, PEM<sup>2</sup> and RIV<sup>2</sup> need at least the controller structure. Nonetheless, VCM requires the design of four auxiliary TF: model and coefficients, which could be time consuming. Hence, one should wonder if it is worthy to spend too much effort on the controller identification, if the final goal is the robot model. From that perspective, PROJ seems more attractive. Nevertheless, all those methods assume that the system is linear with respect to the states and the parameters. Robot systems are known to be linear in the parameters but not in the states.

From this short literature review, the problem of the controller identification appears to be mainly driven by two elements. The first element is the initial knowledge on the controller. The question is: does the practitioner know the structure of the controller? The second element is the final goal of the identification process. In other words, the question is: the aim is the knowledge of the robot model or the knowledge of the whole closed-loop system? Depending on the answers to both questions, different approaches can be considered to deal with the controller identification. Table 5.4 summarizes the approaches available to the practitioner depending on its requirements. From the methods summarized in 5.3, the general idea of a two step procedure is kept:

- Controller identification with a parametric or non-parametric method;
- Identification of the dynamic model with the IDIM-IV or DIDIM method.

In practice, one can imagine to include the controller identification in the optimisation process of the dynamic model identification, as it is suggested in [Gilson *et al.* 2011] for instance. Nevertheless, we prefer to separate the two identification problems to avoid a correlation between the models and the impact of potential modelling errors. We will now detailed the approaches for the first step.

Table 5.4: Approaches for the identification of the controller

Identification goal	Whole closed-loop system	Robot model only
Initial knowledge		
Controller structure known	Parametric	Parametric/Non-parametric
Controller structure unknown	Parametric	Non-parametric

### 5.2.2 Controller parametric identification

To identify the controller, different strategies can be developed depending on the *a priori* knowledge. If the controller structure is known to the practitioner, a parametric identification method is a straightforward solution. If the structure is unknown, the control law (model and parameters) can be identified with many classical techniques such as: the Prediction Error Method [Ljung 1999] or the Simplified Refined IV [Young 2011] for instance. It is assumed that the following signals are available: the controller inputs:  $\mathbf{q}_r$ ,  $\hat{\mathbf{q}}$ ,  $\hat{\dot{\mathbf{q}}}$  and  $\hat{\ddot{\mathbf{q}}}$ , as well as its output:  $\boldsymbol{\nu}_\tau$ . That is thus a Multiple Input Single Output (MISO) problem of system identification, which can be treated with the method that best suits the practitioner; see e.g. [Pascu *et al.* 2016].

As explained in [Khalil & Dombre 2004], a predictive/feed-forward action can be employed to reduce the tracking error. Therefore, it could be relevant to take also into account  $\dot{\mathbf{q}}_r$  and  $\ddot{\mathbf{q}}_r$  as inputs, for the control law identification.

From a system identification perspective, the identification of the controller does not seem too complex theoretically. The noise contained in  $\boldsymbol{\nu}_\tau$  indeed comes only from the inputs, because  $\boldsymbol{\nu}_\tau$  is a computed output and not a measured one. There is no instrumentation noise. Nonetheless, that does not change anything regarding the establishment of the controller's model and its potential complexity.

### 5.2.3 Controller non-parametric identification

With another perspective, the practitioner can identify only the impulse response of the control law with a non-parametric technique, which avoids any thought on the controller structure. This technique can be the correlation analysis for example. The idea is to identify one Finite Impulse Response (FIR) for each link in order to simulate the controller. The FIR filters are identified such as  $\nu_{\tau_j}(t) = IR_j(z^{-1})(q_{r_j}(t) - q_{m_j}(t))$ , for each link  $j$ . That is to say that they are MA filters of sizes  $m_j$ . The identified control law is referred to as CoRrelated Controller (CRC).

If the practitioner selects a size  $m_j$  different from the real one or if the controller is in fact an Infinite Impulse Response (IIR) filter, the identified controller  $IR_j(z^{-1})$  will be biased. One can also think about a nonlinear control law like a computed torque one; see section 2.2.1. The identified controller, used in the auxiliary model, may be biased as long as the simulated signals are enough close to the measured ones, in order to insure the correlation between the instrument and observation matrices. This idea of an approximate controller for the auxiliary model is valid for

the IDIM-IV method but not for the DIDIM one. In fact, as it has been explained in section 4.2.2, the IDIM-IV method can provide efficient estimates while having an error in the auxiliary model.

If the correlation analysis is thoroughly developed in Chapter 3 of [Söderström & Stoica 1988] for instance, we summarize here the main features of this method. The form of the DT model used is:

$$\nu_{\tau_j}(t_i) = \sum_{k=0}^{\infty} h^j(k) \varepsilon_{tr_j}(t_i - k \cdot \Delta t) + \varepsilon_d(t_i) \quad (5.30)$$

where  $\varepsilon_{tr_j}(t) = q_{r_j}(t) - q_{m_j}(t)$ ,  $\{h^j(k)\}$  is a weighting sequence and  $\varepsilon_d$  is a disturbance not correlated with the input  $\varepsilon_{tr_j}$ . Hence, it comes out:

$$\begin{aligned} R_{\nu_{\tau} \varepsilon_{tr}}^j(\tau_d) &= \mathbb{E}[\nu_{\tau_j}(t + \tau_d) \varepsilon_{tr_j}(t)] \\ &= \sum_{k=0}^{\infty} h^j(k) \mathbb{E}[\varepsilon_{tr_j}(t + \tau_d - k \cdot T_s) \varepsilon_{tr_j}(t)] \\ &= \sum_{k=0}^{\infty} h^j(k) R_{\varepsilon_{tr} \varepsilon_{tr}}^j(\tau_d - k \cdot T_s), \end{aligned} \quad (5.31)$$

with  $\tau_d$  the discrete time lag. With the estimated correlations and a truncated filter response at order  $m_j$ , we have the following relation:

$$\widehat{R}_{\nu_{\tau} \varepsilon_{tr}}^j(\tau_d) = \sum_{k=0}^{m_j-1} h^j(k) \widehat{R}_{\varepsilon_{tr} \varepsilon_{tr}}^j(\tau_d - k \cdot T_s). \quad (5.32)$$

Writing the previous equation for  $\tau_d = 0, \dots, (m_j - 1) \cdot T_s$ , we obtain a linear system of  $m_j$  equations with  $m_j$  unknowns  $IR_j = \{h^j(k)\}_{k=0}^{m_j-1}$ . In practice, the algorithm can be implemented as follows:

- (1) Identify an AR filter based on the input signal  $\varepsilon_{tr_j}$ , see section 6.1.3;
- (2) Filter the input and the output by the inverse of the identified AR filter to have  $\varepsilon_{tr_j}^f$  and  $\nu_{\tau_j}^f$ ;
- (3) Estimate the correlation  $R_{\nu_{\tau} \varepsilon_{tr}}^j$  and  $R_{\varepsilon_{tr} \varepsilon_{tr}}^j$  from the filtered signals:  $\varepsilon_{tr_j}^f$  and  $\nu_{\tau_j}^f$ ;
- (4) Solve the linear system defined by (5.32).

This algorithm includes in addition an AR filter to prewhiten the data in accordance with the corresponding Matlab<sup>®</sup> function named `cra` which come from the technique developed in section 11.2 of [Box & Jenkins 1976].

## 5.3 Experimental validation

### 5.3.1 The IDIM-LS method with the IRWSM technique

#### Robot identification with good a priori knowledge

The two IRWSM approaches defined in section 5.1.3 are compared with the usual differentiation technique by considering the IDIM-LS method. For reminder, the usual technique is composed of the BandPass (BP) filter and the Finite Differences (FD). The identification is performed with experimental data of the TX40. The filters cut-off frequencies are 50 Hz and 20 Hz for the Butterworth (or BP) and the decimate filters respectively, assuming  $\omega_{dyn} = 10$  Hz, in accordance with appendix B. Contrary to the BP filter, the decimate filter is applied with the three methods. Table 5.5 summarizes the results of the identification. The three methods almost estimate the same parameters. Those estimated values are satisfactory since they are similar to those found in previous studies on this robot. Their relative errors are equivalent and can be considered as satisfactory, see Table 5.6.

This first identification proves that methods, based on the IRWSM approach, are able to provide as good estimates as the Bandpass one. However, in this case, we assumed good *a priori* knowledge; i.e. the system bandwidth was well-known and the filters were adequately tuned. That can be an issue especially with the first identification process for a system totally unknown, like a robot bought off-the-self.

To test the robustness of our proposed methodology, we consider the same data set but assuming  $\omega_{dyn} = 20$  Hz due to a lack of knowledge on the system. Following the rules of thumb provided in section 2.3.1, the relation  $\omega_q = 10 \cdot \omega_{dyn}$  is considered, which leads to a 200 Hz cut-off frequency for the Butterworth filter. Concerning the decimate filter, we are slightly less restrictive with  $\omega_f = 5 \cdot \omega_{dyn} = 100$  Hz.

#### Robot identification with poor a priori knowledge

Two ways are considered for the evaluation of the NVR: the `irwsmopt` function and a manual tuning. For the manual tuning, the NVR is equal to  $10^{-5}$ , which has proved to be an appropriate choice; see [Brunot *et al.* 2016b]. This value of NVR is fixed for both "manual" IRWSM methods. The results of the identification with poor *a priori* knowledge are summarized in Table 5.7.

This example illustrates the difficulty of the classical IDIM-LS method when the system bandwidth is not well-known and when non-optimal relations are used to tune the filters. It also appears some difficulties for the IRWSM methods when the NVR is estimated with the `irwsmopt` routine. That shows the importance of the decimate filter in the identification process. It allows to compensate some high-frequency noise components that would not have been removed by the `irwsm` function. In fact, the `irwsmopt` algorithm may tend to catch all the dynamics of the noisy signal. In fact, it gives too much importance to the covariance of the state noise compared to the one of the measurement noise to model the perturbation. A careful visual inspection of the signals, prior to the identification, by the user is

Table 5.5: Estimated parameters and relative standard deviations - Good *a priori* knowledge case - IDIM-LS

Param.	BP + LS	IRWSM 1	IRWSM 2
$zz_{1_r}$	1.24 (1.39 %)	1.24 (1.39 %)	1.24 (1.40 %)
$fc_1$	7.34 (2.66 %)	7.29 (2.70 %)	7.28 (2.72 %)
$fv_1$	7.93 (0.88 %)	7.95 (0.89 %)	7.96 (0.89 %)
$xx_{2_r}$	-0.48 (3.28 %)	-0.47 (3.31 %)	-0.48 (3.30 %)
$xz_{2_r}$	-0.16 (5.40 %)	-0.16 (5.38 %)	-0.16 (5.38 %)
$zz_{2_r}$	1.09 (1.29 %)	1.09 (1.29 %)	1.09 (1.29 %)
$mx_{2_r}$	2.21 (3.01 %)	2.21 (3.02 %)	2.22 (3.01 %)
$fc_2$	8.24 (2.26 %)	8.20 (2.28 %)	8.20 (2.27 %)
$fv_2$	5.50 (1.45 %)	5.52 (1.45 %)	5.52 (1.45 %)
$xx_{3_r}$	0.13 (10.6 %)	0.13 (10.6 %)	0.13 (10.6 %)
$zz_{3_r}$	0.11 (9.57 %)	0.11 (9.56 %)	0.11 (9.56 %)
$my_{3_r}$	-0.60 (2.52 %)	-0.60 (2.52 %)	-0.60 (2.53 %)
$ia_3$	0.09 (9.39 %)	0.09 (9.39 %)	0.09 (9.56 %)
$fc_3$	6.48 (2.06 %)	6.47 (2.05 %)	6.48 (2.07 %)
$fv_3$	1.93 (2.05 %)	1.93 (2.04 %)	1.93 (2.06 %)
$mx_4$	-0.02 (35.1 %)	-0.02 (35.3 %)	-0.02 (35.0 %)
$ia_4$	0.03 (9.10 %)	0.03 (9.30 %)	0.03 (9.41 %)
$fc_4$	2.57 (2.46 %)	2.55 (2.53 %)	2.55 (2.54 %)
$fv_4$	1.09 (1.62 %)	1.09 (1.65 %)	1.09 (1.66 %)
$my_{5_r}$	-0.03 (15.2 %)	-0.03 (15.4 %)	-0.03 (15.5 %)
$ia_5$	0.04 (10.6 %)	0.04 (10.8 %)	0.04 (10.9 %)
$fc_5$	3.07 (3.59 %)	3.06 (3.60 %)	3.06 (3.60 %)
$fv_5$	1.79 (2.33 %)	1.80 (2.33 %)	1.80 (2.33 %)
$ia_6$	0.01 (13.3 %)	0.01 (13.6 %)	0.01 (13.6 %)
$fc_6$	0.30 (32.9 %)	0.29 (34.0 %)	0.29 (33.9 %)
$fv_6$	0.65 (1.85 %)	0.65 (1.85 %)	0.65 (1.85 %)
$fc_{m6}$	1.90 (4.40 %)	1.90 (4.41 %)	1.90 (4.41 %)
$fv_{m6}$	0.61 (1.97 %)	0.61 (1.98 %)	0.61 (1.98 %)

Table 5.6: Direct comparison - Relative errors - IDIM-LS

	Bandpass	IRWSM 1	IRWSM 2
Good knowledge	5.1 %	5.1 %	5.1 %
Poor knowledge (irwsmopt)	9.2 %	9.0 %	8.9 %
Poor knowledge (manual)	–	7.5 %	8.6 %

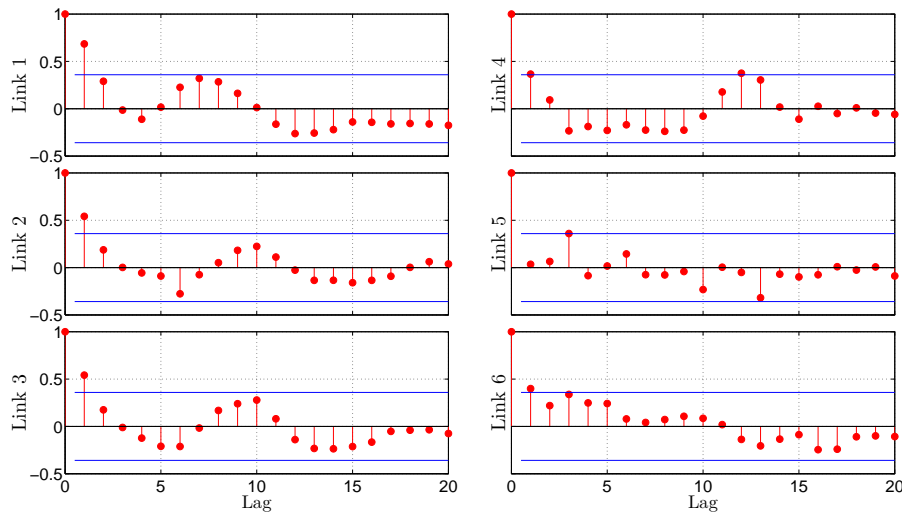


Figure 5.2: Residuals autocorrelation - Good *a priori* knowledge case - IDIM-LS with IRWSM 1

therefore required.

In the manual tuning case, IRWSM 1 provides better results than the others since it has a lower relative error, see Table 5.6. That is also visible with some estimated parameters like  $zz_{1r}$  or  $zz_{3r}$ . The relative standard deviations provided in Table 5.7 may seem promising. However, they are not valid since the residuals are not serially uncorrelated as it should be the case in theory. Figures 5.2 and 5.3 provide the residuals autocorrelation estimations respectively for the good and the poor (manual) *a priori* knowledge cases considering the IRWSM 1 method. Because the residuals are completely similar for the IRWSM 2 method, they are not shown here for the sake of clarity. The IRWSM 1 residuals clearly appear to be serially correlated in the poor knowledge case. In practice, to obtain valid estimated parameters and standard deviations, this identification process could be used as a first step to retrieve the closed-loop dynamics. Subsequently, a second step would be performed to identify the system with filters appropriately tuned.

This more practical case shows that the IRWSM approach is robust against poor *a priori* knowledge. However, it requires a careful use with a potential manual selection of the NVR. This estimation can be used as a first step for the design of pre-filters for the usual Bandpass method. In practice, the IRWSM 1 solution should be preferred instead of the IRWSM 2.

### 5.3.2 The IDIM-IV method with the IRWSM technique

To be complete, the IRWSM technique is also combined with the IDIM-IV method. For the sake of clarity, the good *a priori* knowledge case is not presented. The results are indeed totally similar to those obtained with the bandpass filter in section

Table 5.7: Estimated parameters and relative standard deviations - Poor *a priori* knowledge case - IDIM-LS

Param.	Bandpass	irwsmopt		manual	
		IRWSM 1	IRWSM 2	IRWSM 1	IRWSM 2
$zz_{1,r}$	1.21 (0.73 %)	1.20 (0.72 %)	1.21 (0.72 %)	1.24 (0.55 %)	1.21 (0.70 %)
$fc_1$	7.15 (1.61 %)	7.10 (1.60 %)	7.09 (1.60 %)	7.12 (1.14 %)	7.10 (1.53 %)
$fv_1$	7.99 (0.52 %)	8.01 (0.51 %)	8.02 (0.51 %)	8.01 (0.36 %)	8.01 (0.49 %)
$xx_{2,r}$	-0.47 (1.87 %)	-0.47 (1.85 %)	-0.47 (1.84 %)	-0.48 (1.34 %)	-0.47 (1.77 %)
$xz_{2,r}$	-0.16 (3.26 %)	-0.16 (3.20 %)	-0.16 (3.22 %)	-0.16 (2.18 %)	-0.16 (3.06 %)
$zz_{2,r}$	1.02 (0.85 %)	1.02 (0.85 %)	1.02 (0.85 %)	1.09 (0.51 %)	1.03 (0.79 %)
$mx_{2,r}$	2.29 (1.96 %)	2.29 (1.93 %)	2.29 (1.93 %)	2.21 (1.20 %)	2.28 (1.81 %)
$fc_2$	8.29 (1.55 %)	8.26 (1.54 %)	8.26 (1.54 %)	8.11 (0.91 %)	8.23 (1.43 %)
$fv_2$	5.49 (1.00 %)	5.50 (0.99 %)	5.50 (0.99 %)	5.56 (0.57 %)	5.52 (0.91 %)
$xx_{3,r}$	0.12 (5.72 %)	0.12 (5.68 %)	0.12 (5.70 %)	0.13 (4.23 %)	0.12 (5.55 %)
$zz_{3,r}$	0.06 (8.64 %)	0.06 (8.31 %)	0.06 (8.14 %)	0.12 (3.63 %)	0.07 (7.50 %)
$my_{3,r}$	-0.62 (1.14 %)	-0.62 (1.13 %)	-0.63 (1.12 %)	-0.60 (0.97 %)	-0.62 (1.10 %)
$ia_3$	0.12 (3.44 %)	0.12 (3.43 %)	0.12 (3.48 %)	0.09 (3.61 %)	0.11 (3.50 %)
$fc_3$	6.33 (0.90 %)	6.33 (0.89 %)	6.34 (0.89 %)	6.35 (0.77 %)	6.33 (0.87 %)
$fv_3$	1.95 (0.87 %)	1.96 (0.86 %)	1.96 (0.86 %)	1.96 (0.74 %)	1.96 (0.84 %)
$mx_4$	-0.01 (39.9 %)	-0.01 (36.3 %)	-0.01 (34.6 %)	-0.02 (13.3 %)	-0.01 (47.8 %)
$ia_4$	0.03 (4.04 %)	0.03 (3.98 %)	0.03 (4.01 %)	0.03 (3.58 %)	0.03 (4.24 %)
$fc_4$	2.50 (1.12 %)	2.50 (1.12 %)	2.50 (1.12 %)	2.50 (1.05 %)	2.50 (1.11 %)
$fv_4$	1.11 (0.71 %)	1.11 (0.70 %)	1.11 (0.71 %)	1.11 (0.66 %)	1.11 (0.70 %)
$my_{5,r}$	-0.03 (6.13 %)	-0.03 (6.09 %)	-0.03 (6.11 %)	-0.03 (5.78 %)	-0.03 (5.94 %)
$ia_5$	0.04 (4.07 %)	0.04 (3.99 %)	0.04 (4.02 %)	0.05 (3.82 %)	0.04 (4.02 %)
$fc_5$	3.07 (1.50 %)	3.07 (1.48 %)	3.07 (1.48 %)	3.04 (1.36 %)	3.06 (1.40 %)
$fv_5$	1.79 (0.98 %)	1.79 (0.97 %)	1.79 (0.97 %)	1.80 (0.88 %)	1.79 (0.91 %)
$ia_6$	0.01 (5.25 %)	0.01 (5.07 %)	0.01 (5.07 %)	0.01 (4.90 %)	0.01 (5.94 %)
$fc_6$	0.30 (13.7 %)	0.30 (13.5 %)	0.30 (13.5 %)	0.30 (12.2 %)	0.29 (13.2 %)
$fv_6$	0.65 (0.78 %)	0.66 (0.77 %)	0.65 (0.77 %)	0.66 (0.67 %)	0.66 (0.71 %)
$fc_{m6}$	1.88 (1.83 %)	1.88 (1.81 %)	1.88 (1.81 %)	1.86 (1.65 %)	1.88 (1.70 %)
$fv_{m6}$	0.61 (0.83 %)	0.61 (0.82 %)	0.61 (0.82 %)	0.61 (0.73 %)	0.61 (0.76 %)



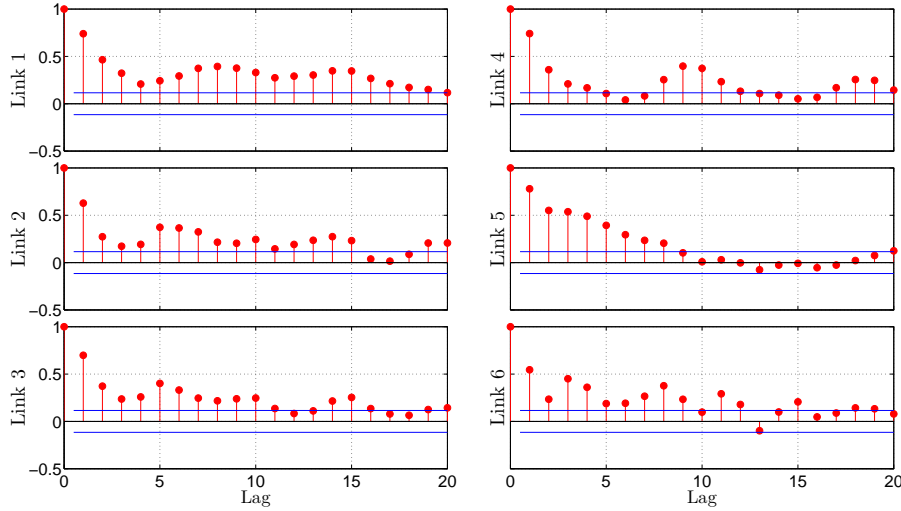


Figure 5.3: Residuals autocorrelation - Poor *a priori* knowledge case - IDIM-LS with IRWSM 1

4.2.3 for the nominal case. From a methodological point of view, four methods are compared. The first one is the usual IDIM-LS approach, with the BandPass (BP) filter and the Finite Differences (FD). The second method is the usual IDIM-IV approach described in section 2.3.3, with the instruments generated by the auxiliary model and the observation matrix built with the BP filter and the FD. The third method is also an IDIM-IV approach, but with the observation matrix built with the `irwsm` function implemented in the CAPTAIN Toolbox. This method is referred to as IDIM-IV (IRWSM 1) because it is based on the IRWSM 1 variant suggested in section 5.1.3. The last method is a variant of the previous one where the GRW model contains three states as illustrated with IRWSM 2 variant in section 5.1.3. The resulting method is called IDIM-IV (IRWSM 2) and allows estimating directly the second derivative without calling the algorithm twice.

### Robot identification with poor *a priori* knowledge

In order to challenge the robustness of the IRWSM approach, we consider the same experimental data but with a lack of knowledge on the system. The closed-loop bandwidth is assumed to be  $\omega_{dyn} = 30$  Hz. Following the rules provided in section 2.3.1, we take  $\omega_q = 6 \cdot \omega_{dyn}$  which leads to a 180 Hz cut-off frequency for the BP filter. With respect to the decimate filter, we apply the following relation  $\omega_f = 5 \cdot \omega_{dyn} = 150$  Hz.

Table 5.8 summarizes the identification results with poor *a priori* knowledge. First of all, the usual IDIM-LS method, with the bandpass filter, has difficulties. That is especially visible with some inertia like  $zz_{1,r}$  or  $zz_{3,r}$ . It illustrates the difficulty of the classical IDIM-LS method when the system bandwidth is not well-known

and when non-optimal relations are used to set the filters. The relative errors provided in Table 5.10 do not highlight this difficulty because the observation matrices of the IV methods are significantly noisy. The relative errors are indeed evaluated with  $\|\mathbf{y}(\boldsymbol{\tau}) - \mathbf{X}\hat{\boldsymbol{\theta}}\| / \|\mathbf{y}(\boldsymbol{\tau})\|$ . Nonetheless, the noise located in  $\mathbf{X}$  is compensated by the instrumental matrices for the parameters estimation.

Concerning the IDIM-IV methods, they provide similar results that are close to those given in Appendix B.1. Nonetheless, the IRWSM approach for the derivatives' estimation brings two advantages. Firstly, the user does not have to wonder at all about the setting of the BP filter. Secondly, as shown in Table 5.9, the IRWSM requires less computation to reach the convergence.

One fact is worth noting about the relative standard deviations in Table 5.8. If they seem promising, they are not valid due to the wrong setting of the decimate filter. Here also, the residuals are not serially uncorrelated as it should be the case in theory. In fact, the following chapter is devoted to the development of a methodology able to provide serially uncorrelated residuals without *a priori* knowledge.

### 5.3.2.1 Robot identification without a priori knowledge

The last experimental case to consider is when there is no knowledge about the system's bandwidths. In a brutal way, the user can undertake the identification without the BP and the decimate filters. As visible in Table 5.9, the IDIM-IV method based on the FD did not converge in the limit of 20 iterations. The IDIM-LS results are presented in Table 5.11 but they are not acceptable,  $zz_{3,} = -0.04$  for instance. That table also gives the estimated parameters for the IDIM-IV with IRWSM 1 and 2. The results are completely similar to the values found in the previous sections. Here also, the relative standard deviations are not admissible due to the serial autocorrelation.

Concerning the relative errors in Table 5.10, the one of the IDIM-IV (IRWSM 1) method appears to be noticeably lower. Nonetheless, such an estimation is not acceptable as a final result. It can be used as a first step to adequately tune the decimate filter. With regard to the choice between IRWSM 1 or 2, this last example suggests that the IRWSM 2 approach provides a more noisy observation matrix. In other words, the estimated derivatives contains more disturbances. Therefore, the IRWSM 1 approach should be the first user's choice.

That last example, which is a limit case, illustrates the interest of the IRWSM approach for an unknown system.

### 5.3.3 The IDIM-IV method with an unknown controller

To illustrate the case where the controller is unknown, we identify a parametric model of the controller for the IDIM-IV technique. This is done with the Simplified Refined IV (SRIV) method available in the CAPTAIN Toolbox. The identified controller model is composed of six PID and is indicated by  $\hat{C}$ ; see appendix B.3

Table 5.8: Estimated parameters and relative standard deviations - Poor *a priori* knowledge case - IDIM-IV

Param.	IDIM-LS (BP+FD)	IDIM-IV (BP+FD)	IDIM-IV (IRWSM 1)	IDIM-IV (IRWSM 2)
$zz_{1_r}$	<b>1.18</b> (0.65 %)	1.26 (0.78 %)	1.25 (0.48 %)	1.25 (0.57 %)
$fc_1$	7.12 (1.11 %)	7.19 (1.32 %)	7.14 (0.82 %)	7.13 (0.31 %)
$fv_1$	8.00 (0.36 %)	7.96 (0.42 %)	7.98 (0.26 %)	7.99 (0.31 %)
$xx_{2_r}$	-0.46 (1.44 %)	-0.47 (1.83 %)	-0.47 (1.12 %)	-0.47 (1.35 %)
$xz_{2_r}$	-0.16 (2.09 %)	-0.16 (3.12 %)	-0.16 (1.92 %)	-0.16 (2.29 %)
$zz_{2_r}$	0.99 (0.55 %)	1.09 (0.63 %)	1.10 (0.39 %)	1.09 (0.47 %)
$mx_{2_r}$	2.38 (1.10 %)	2.25 (1.74 %)	2.25 (1.08 %)	2.25 (1.29 %)
$fc_2$	8.36 (0.86 %)	8.29 (1.01 %)	8.26 (0.63 %)	8.26 (0.75 %)
$fv_2$	5.49 (0.58 %)	5.48 (0.67 %)	5.50 (0.41 %)	5.50 (0.49 %)
$xx_{3_r}$	0.14 (4.37 %)	0.13 (5.73 %)	0.13 (3.49 %)	0.13 (4.13 %)
$zz_{3_r}$	<b>-0.02</b> (17.3 %)	0.12 (4.99 %)	0.12 (3.15 %)	0.12 (3.64 %)
$my_{3_r}$	-0.66 (1.07 %)	-0.59 (1.48 %)	-0.59 (0.91 %)	-0.58 (1.09 %)
$ia_3$	0.14 (2.67 %)	0.09 (5.96 %)	0.09 (3.64 %)	0.09 (4.34 %)
$fc_3$	6.24 (1.15 %)	6.52 (1.29 %)	6.50 (0.80 %)	6.52 (0.95 %)
$fv_3$	1.96 (1.09 %)	1.93 (1.28 %)	1.94 (0.79 %)	1.93 (0.94 %)
$mx_4$	-0.001 (127 %)	-0.03 (18.4 %)	-0.03 (11.0 %)	-0.03 (13.2 %)
$ia_4$	0.03 (6.81 %)	0.03 (8.75 %)	0.03 (5.33 %)	0.03 (6.53 %)
$fc_4$	2.43 (3.20 %)	2.50 (3.63 %)	2.48 (2.24 %)	2.50 (2.67 %)
$fv_4$	1.13 (1.90 %)	1.10 (2.25 %)	1.10 (1.38 %)	1.10 (1.66 %)
$my_{5_r}$	-0.06 (4.94 %)	-0.03 (11.0 %)	-0.04 (6.15 %)	-0.03 (7.97 %)
$ia_5$	0.04 (5.44 %)	0.04 (8.26 %)	0.04 (4.99 %)	0.04 (5.91 %)
$fc_5$	3.07 (2.35 %)	3.08 (2.71 %)	3.07 (1.67 %)	3.08 (2.00 %)
$fv_5$	1.78 (1.54 %)	1.80 (1.74 %)	1.80 (1.07 %)	1.79 (1.29 %)
$ia_6$	0.01 (10.2 %)	0.01 (18.1 %)	0.01 (11.0 %)	0.01 (14.2 %)
$fc_6$	0.26 (34.6 %)	0.28 (37.4 %)	0.28 (22.3 %)	0.27 (27.8 %)
$fv_6$	0.65 (2.08 %)	0.65 (2.46 %)	0.65 (1.51 %)	0.65 (1.82 %)
$fc_{m6}$	1.96 (3.18 %)	1.93 (3.68 %)	1.91 (2.28 %)	1.93 (2.71 %)
$fv_{m6}$	0.59 (1.78 %)	0.60 (2.03 %)	0.60 (1.25 %)	0.60 (1.50 %)

Table 5.9: Number of iterations for the IDIM-IV methods

	IDIM-IV (BP+FD)	IDIM-IV (IRWSM 1)	IDIM-IV (IRWSM 2)
Poor knowledge	7	4	3
No knowledge	–	4	3

Table 5.10: Direct comparison - Relative errors - IDIM-IV

	IDIM-LS (BP+FD)	IDIM-IV (BP+FD)	IDIM-IV (IRWSM 1)	IDIM-IV (IRWSM 2)
Poor knowledge	9.91 %	10.9 %	11.3 %	11.3 %
No knowledge	42.2 %*	– *	37.3 %	82.2 %

\*: no BP in this case

Table 5.11: Estimated parameters and relative standard deviations - Without *a priori* knowledge case - IDIM-IV

Param.	IDIM-LS	IDIM-IV	IDIM-IV
	(FD)	(IRWSM 1)	(IRWSM 2)
$zz_{1_r}$	0.07 (4.46 %)	1.25 (0.76 %)	1.25 (2.73 %)
$fc_1$	6.47 (1.39 %)	7.14 (1.31 %)	7.14 (4.72 %)
$fv_1$	8.14 (0.40 %)	7.99 (0.41 %)	7.98 (1.49 %)
$xx_{2_r}$	-0.07 (4.33 %)	-0.47 (1.79 %)	-0.47 (6.45 %)
$xz_{2_r}$	-0.02 (6.74 %)	-0.16 (3.06 %)	-0.16 (11.1 %)
$zz_{2_r}$	0.05 (3.12 %)	1.09 (0.62 %)	1.09 (2.23 %)
$mx_{2_r}$	4.02 (0.68 %)	2.25 (1.73 %)	2.25 (6.23 %)
$fc_2$	9.88 (0.83 %)	8.26 (1.00 %)	8.27 (3.62 %)
$fv_2$	4.34 (0.80 %)	5.50 (0.66 %)	5.50 (2.38 %)
$xx_{3_r}$	-0.03 (9.43 %)	0.13 (5.47 %)	0.13 (20.0 %)
$zz_{3_r}$	-0.04 (3.83 %)	0.12 (4.68 %)	0.12 (17.2 %)
$my_{3_r}$	-0.20 (1.96 %)	-0.59 (1.44 %)	-0.59 (5.16 %)
$ia_3$	0.06 (2.53 %)	0.09 (5.73 %)	0.09 (20.5 %)
$fc_3$	6.20 (1.33 %)	6.53 (1.28 %)	6.52 (4.62 %)
$fv_3$	2.10 (1.17 %)	1.93 (1.27 %)	1.93 (4.56 %)
$mx_4$	0.05 (2.98 %)	-0.03 (17.1 %)	-0.03 (60.6 %)
$ia_4$	0.01 (10.2 %)	0.03 (8.34 %)	0.03 (29.3 %)
$fc_4$	2.60 (3.45 %)	2.50 (3.59 %)	2.50 (12.9 %)
$fv_4$	1.13 (2.18 %)	1.10 (2.22 %)	1.10 (8.01 %)
$my_{5_r}$	-0.02 (8.15 %)	-0.03 (10.2 %)	-0.03 (36.4 %)
$ia_5$	0.01 (7.19 %)	0.04 (6.45 %)	0.04 (24.9 %)
$fc_5$	3.09 (2.70 %)	3.08 (2.66 %)	3.08 (9.58 %)
$fv_5$	1.77 (1.77 %)	1.79 (1.72 %)	1.80 (6.19 %)
$ia_6$	0.001 (21.3 %)	0.01 (14.5 %)	0.01 (56.8 %)
$fc_6$	0.25 (41.9 %)	0.27 (36.5 %)	0.28 (132 %)
$fv_6$	0.66 (2.38 %)	0.65 (2.42 %)	0.65 (8.71 %)
$fc_{m6}$	1.89 (3.81 %)	1.93 (3.60 %)	1.93 (13.0 %)
$fv_{m6}$	0.61 (1.99 %)	0.60 (2.00 %)	0.60 (7.21 %)

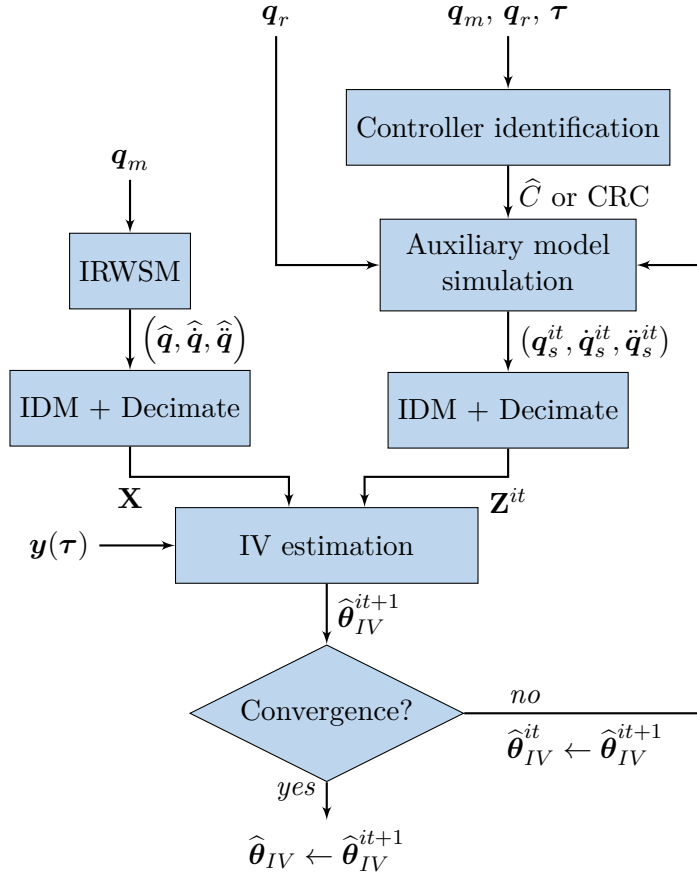


Figure 5.4: Block diagram of the IDIM-IV method

for further information of the controller's structure. For link  $j$ , the input and the output of the corresponding PID are respectively  $(q_{r_j} - q_{m_j})$  and  $\nu_{r_j}$ . In this case, the observation matrix is constructed with the IRWSM 1 technique and the decimate filter is tuned with a cut-off frequency of 20 Hz.

We also consider a non-parametric identification of the controller thanks to the `cra` function of the System Identification Toolbox; see [Ljung 1988] and [Ljung 1999]. That function follows the principles described in section 5.2.3. One Finite Impulse Response (FIR) is identified for each link in order to simulate the controller. The size of this filter is noticed  $m_j$ . The `cra` function allows to prewhiten the input,  $q_{r_j}(t) - q_{m_j}(t)$ , with an AR model of order  $na_j$ , for link  $j$ . Table 5.12 provides the filter sizes used to identify the six impulse responses. The CRC is used for the simulation of the DDM to run the IDIM-IV method. The identification process is summarized in Figure 5.4 using (2.33) for the IV estimation.

With regard to the convergence, the IDIM-IV (CRC) and IDIM-IV ( $\hat{C}$ ) methods took 4 and 3 iterations respectively. As depicted in Table 5.13, the CRC is an adequate solution since the IDIM-IV estimates match those of the reference methodology: the IDIM-LS method. That is also the case for the IDIM-IV ( $\hat{C}$ ) method.

Table 5.12: *cra* function parameters

Link	1	2	3	4	5	6
$m_j$	10	20*	3	13	10	20*
$na_j$	10*	10*	10*	10*	10*	20

\* default value of the *cra* function

The six PID identified are indeed really close to the actual ones. The effectiveness of the methods is confirmed by Table 5.14 providing the relative errors. The fact that the CRC works with relatively simple impulse responses can be explained by the negligible effect of the integral action. As explained in section 2.2.1, the controller can be considered as a PD control law because the integral term is usually weak. Few trials were undertaken to run the DIDIM method with the CRC without success.

## 5.4 Conclusions

In this chapter, the robot identification methodologies introduced in chapter 2 have been reformulated to avoid the use of the bandpass filter and the knowledge of the controller for the purpose of simulation. Differentiation techniques have been considered in section 5.1 and the IRWSM technique has been selected. Details have been made on the manner in which it can automatically find the position signal derivatives based on a ML optimisation. The other major contribution of this chapter is the implementation of the IDIM-IV method when the controller is unknown. A parametric and a non-parametric identification techniques have been successfully applied to obtain an estimated controller for the auxiliary model, which generates the instruments.

In section 5.3, those two contributions are assessed with the identification of the TX40 based on experimental data. Notably, it has been illustrated that the IRWSM performs as well as the bandpass filtering process for the IDIM-LS method, if the decimate filter is properly tuned. Otherwise, a default NVR value has proven to be an appropriate solution to compensate the lack of high-frequency ripples rejection. For the IDIM-IV method, the use of the IRWSM is straightforward and at least as effective as the usual preprocess.

Concerning the unknown controller case, the example has illustrated the suitability of the controller non-parametric identification for the IDIM-IV method but not for the DIDIM one. This is due to the better robustness highlighted in section 4.2. Nonetheless, with the parametric identification, both the IDIM-IV and DIDIM methods have proven to be effective. A clear perspective of this work is to consider more complex controller cases, like a computed torque one, to evaluate the performance of the approach on the considered robot.

It appeared that a well-tuned Butterworth filter or the *irwsm* routine are necessary to obtain consistent estimated parameters with the IDIM-LS method. The decimate filter does not have an effect on the consistency but the estimation of the

Table 5.13: Estimated parameters and relative standard deviations - Controller unknown

Param.	IDIM-LS	IDIM-IV (CRC)	IDIM-IV ( $\hat{C}$ )
$zz_{1_r}$	1.24 (1.20%)	1.25 (1.26%)	1.25 (1.27%)
$fv_1$	7.95 (0.66%)	7.94 (0.69%)	7.98 (0.67%)
$fc_1$	7.29 (1.99%)	7.27 (2.14%)	7.16 (2.09%)
$xx_{2_r}$	-0.47 (2.68%)	-0.47 (2.94%)	-0.47 (2.96%)
$xz_{2_r}$	-0.16 (4.10%)	-0.16 (4.96%)	-0.16 (5.02%)
$zz_{2_r}$	1.09 (0.99%)	1.10 (1.00%)	1.09 (1.02%)
$mx_{2_r}$	2.21 (2.17%)	2.25 (2.79%)	2.25 (2.80%)
$fv_2$	5.51 (1.05%)	5.45 (1.08%)	5.46 (1.08%)
$fc_2$	8.24 (1.60%)	8.34 (1.62%)	8.32 (1.61%)
$xx_{3_r}$	0.14 (8.50%)	0.13 (9.09%)	0.13 (9.13%)
$zz_{3_r}$	0.11 (8.30%)	0.12 (8.48%)	0.12 (8.61%)
$my_{3_r}$	-0.60 (2.32%)	-0.59 (2.36%)	-0.59 (2.37%)
$ia_3$	0.09 (8.47%)	0.09 (9.23%)	0.09 (9.27%)
$fv_3$	1.93 (2.02%)	1.92 (2.05%)	1.93 (2.04%)
$fc_3$	6.47 (2.03%)	6.55 (2.06%)	6.53 (2.05%)
$mx_4$	-0.02 (29.5%)	-0.03 (28.1%)	-0.03 (28.3%)
$ia_4$	0.03 (14.8%)	0.03 (14.0%)	0.03 (14.0%)
$fv_4$	1.09 (3.59%)	1.10 (3.58%)	1.10 (3.56%)
$fc_4$	2.55 (5.60%)	2.50 (5.76%)	2.50 (5.73%)
$my_{5_r}$	-0.03 (17.1%)	-0.04 (16.5%)	-0.04 (16.4%)
$ia_5$	0.04 (13.1%)	0.04 (13.8%)	0.04 (13.9%)
$fv_5$	1.80 (2.80%)	1.80 (2.77%)	1.80 (2.77%)
$fc_5$	3.07 (4.31%)	3.05 (4.35%)	3.05 (4.32%)
$ia_6$	0.01 (17.2%)	0.01 (22.8%)	0.01 (22.7%)
$fv_6$	0.65 (3.76%)	0.65 (3.88%)	0.65 (3.90%)
$fc_6$	0.27 (11.4%)	0.24 (19.9%)	0.25 (16.0%)
$fv_{m6}$	0.60 (3.22%)	0.60 (3.27%)	0.60 (3.26%)
$fs_{m6}$	1.94 (5.88%)	1.95 (5.85%)	1.95 (5.86%)

Table 5.14: Direct comparison relative errors - Unknown controller

IDIM-LS	IDIM-IV (CRC)	IDIM-IV ( $\hat{C}$ )
5.1 %	5.6 %	5.3 %

covariance. As far as the decimate process is concerned, an automated selection of the cut-off frequency could be of interest. The current method requires to be familiar with the system, which hinders its applicability quite a bit. In the next chapter, the noise circulation through the system is analysed in order to provide an alternative to the usual decimate process.



# Noise modelling for robot system identification

---

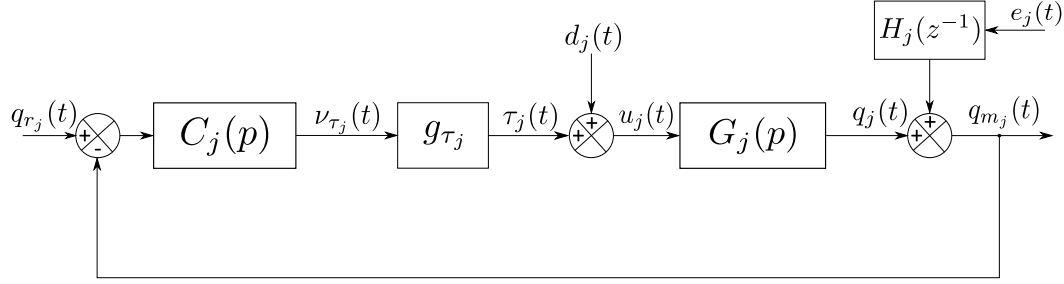
In the previous chapter, the bandpass filter and the knowledge of the controller have been bypassed to reduce the knowledge required prior to the identification. If the results were satisfactory from this perspective, they have underlined the role of the decimate filter to whiten the noise and provide efficient estimates of the covariances. In order to tune adequately this filter, a solution could consist in estimating the dynamical parameters with incorrect covariances in a first time to retrieve the dynamics, before estimating correct covariances in a second time with a correctly tuned filter. In this chapter, an automatic and one-step solution is rather investigated. The idea is to take into account the noise model similarly to the RIVC method, introduced in section 3.2, to provide estimates with low covariances and white residuals. This work has been presented in [Brunot *et al.* 2016a] and submitted in [Brunot *et al.* Submitted 2017].

The chapter is organised as follows. The noise modelling is investigated in section 6.1. From the selected noise model, the robot identification methods are revised to provide estimates automatically in section 6.2. The methods are then applied on the TX40 example in section 6.3 before concluding in section 6.4 on future perspectives for those extensions.

---

<b>6.1</b>	<b>Noise modelling and identification</b>	<b>98</b>
6.1.1	Closed-loop relations	98
6.1.2	Filters models	99
6.1.3	Auto-regressive filters identification	103
<b>6.2</b>	<b>Refined identification methods</b>	<b>105</b>
6.2.1	The separable PEM method	105
6.2.2	The IDIM-PIV method	106
6.2.3	Comments on the extensions	107
<b>6.3</b>	<b>Experimental validations</b>	<b>108</b>
6.3.1	Study of the black-box noise model	108
6.3.2	Study of the grey-box noise model	112
6.3.3	Study with noisy data	113
<b>6.4</b>	<b>Conclusions</b>	<b>116</b>

---

Figure 6.1: Robot model of link  $j$ 

## 6.1 Noise modelling and identification

In this section, we come back to the noise modelling. Until now, in the different identification cases, the additive noise was assumed to be white or at least processed by the decimate filter properly tuned. Since the goal is here to identify the noise model in order to provide more accurate estimates, the noise circulation through the closed-loop system is investigated.

### 6.1.1 Closed-loop relations

We firstly expand the relations of the closed-loop system by considering a joint  $j$ . The linear part of the DDM,  $G_j(p)$ , is defined by

$$G_j(p) = \frac{1}{p(J_j p + F_{v_j})}, \quad (6.1)$$

with  $F_{v_j}$  the viscous friction coefficient and  $J_j$  given by

$$J_j = \max_{\mathbf{q}_{nf}}(\mathbf{M}_{jj}(\mathbf{q}_{nf})). \quad (6.2)$$

$J_j$  is the maximum value, with respect to  $\mathbf{q}_{nf}$ , of the inertia moment. That gives the smallest stability margin of the position closed-loop while  $\mathbf{q}_{nf}$  varies. The joint  $j$  can be modelled as shown in Figure 6.1, where  $d_j$  is the nonlinear disturbance regrouping the Coulomb friction, the centrifugal, Coriolis, gravitational torques and the coupling effects, such as

$$d_j(t) = -N_j(\mathbf{q}_{nf}(t), \dot{\mathbf{q}}_{nf}(t)) + F_{v_j} \dot{q}_{nf_j}(t) - \sum_{k=1, k \neq j}^n M_{jk}(\mathbf{q}_{nf}(t)) \ddot{q}_{nf_k}(t) \quad (6.3)$$

with  $N_j$ , the  $j^{\text{th}}$  element of  $\mathbf{N}$  defined by (2.3). The whole system is modelled as a hybrid one: the controller and the dynamic model are continuous time whereas the noise filter is discrete time.

From Figure 6.1, it comes out the hybrid closed-loop relation

$$\begin{aligned} q_{m_j}(t) &= T_j(p)q_{r_j}(t) + S_j(p)H_j(z^{-1})e_j(t) + S_j(p)G_j(p)d_j(t) \\ T_j(p) &= \frac{g_{\tau_j}C_j(p)G_j(p)}{1 + g_{\tau_j}C_j(p)G_j(p)}, \quad S_j(p) = \frac{1}{1 + g_{\tau_j}C_j(p)G_j(p)} \end{aligned} \quad (6.4)$$

where  $S_j$  is the sensitivity function and  $T_j$  is the complementary sensitivity function; see e.g. [Aström & Murray 2010]. Since for each  $p$ :  $T_j(p) + S_j(p) = 1$ , both functions cannot be made small simultaneously. As explained in [Janot *et al.* 2014a] and reminded in section 2.2.1, in order to have a good tracking at low frequencies, the controller is tuned to insure  $T_j(p) \approx 1$  and consequently  $S_j(p) \approx 0$ . Since  $T_j$  is a low-pass filter, at high frequencies, we have the opposite configuration. Noting that the relevant information comes from the reference signal, the closed-loop transfer function  $T_j$  provides an ideal frequency range for the filtering process. In this range, we have  $T_j(p) \approx 1$ ,  $S_j(p) \approx 0$  and the component from the noise becomes negligible. Therefore, it is consistent to filter the measured position close to the closed-loop dynamics to retrieve a noise-free position signal.

The closed-loop transfer function of the position,  $q_j$ , is given by

$$q_j(t) = T_j(p)q_{r_j}(t) - T_j(p)H_j(z^{-1})e_j(t) + S_j(p)G_j(p)d_j(t). \quad (6.5)$$

The torque closed-loop transfer function is defined by

$$\begin{aligned} \tau_j(t) &= g_{\tau_j}C_j(p)S_j(p) \left[ q_{r_j}(t) - H_j(z^{-1})e_j(t) \right] - T_j(p)d_j(t) \\ &= F_{\tau_j}(p) \left[ q_{r_j}(t) - H_j(z^{-1})e_j(t) \right] - T_j(p)d_j(t). \end{aligned} \quad (6.6)$$

The reference signal and the noise have the same transfer function:  $F_{\tau_j}(p) = g_{\tau_j}C_j(p)S_j(p)$ .

### 6.1.2 Filters models

The IDIM and the measured position respectively defined in (2.17) and (4.12) are recalled here

$$\begin{aligned} \boldsymbol{\tau}(t) &= \boldsymbol{\tau}_{idm}(t) + \boldsymbol{v}(t) \\ \boldsymbol{q}_m(t) &= \boldsymbol{q}_{nf}(t) + \boldsymbol{n}_q(t), \end{aligned} \quad (6.7)$$

where  $\boldsymbol{\tau}_{idm}$  and  $\boldsymbol{q}_{nf}$  are the noise-free components,  $\boldsymbol{v}$  and  $\boldsymbol{n}_q$  are respectively the torque and position noises we want to model. Furthermore, based on the noise-free assumption of the reference trajectory, it comes out the key relation between the input and output noises established in section 4.1.1:

$$\boldsymbol{v}(t) = -\mathbf{G}_\tau \mathbf{C}(p) \boldsymbol{n}_q(t). \quad (6.8)$$

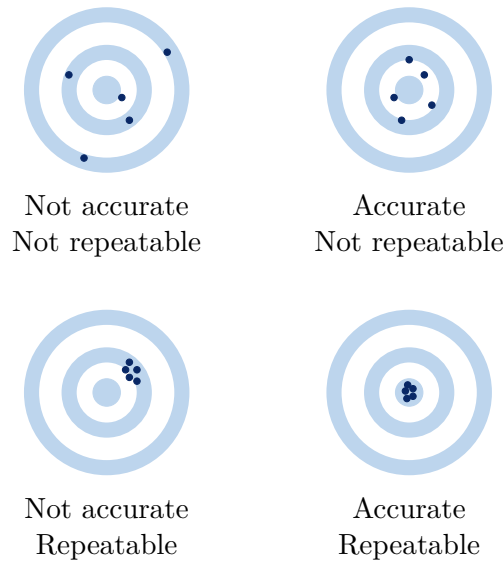


Figure 6.2: Accuracy and repeatability

### Theoretical approach

From the closed-loop relations derived in the previous part, theoretical models of the noises seen by the position and torque of joint  $j$  are available:

$$\begin{aligned} v_j(t) &= -g_{\tau_j} C_j(p) S_j(p) H_j(z^{-1}) e_j(t) \\ n_{q_j}(t) &= S_j(p) H_j(z^{-1}) e_j(t). \end{aligned} \quad (6.9)$$

It should be noticed that those models assume that the nonlinear disturbance is noise-free.

### Empirical approach

Another way of dealing with the noise modelling is to consider the ISO standard [ISO 1998] that must be respected by an industrial robot. Such a standard contains two key parameters to evaluate robot's performances: *accuracy* and *repeatability*. As explained in [Siciliano *et al.* 2010] page 87, in practice, the mechanical dimensions of the robot differ from the MDH notations because of mechanical tolerances. Due to those discrepancies, there is a difference between the reached position and the one computed with direct kinematics. This deviation is called *accuracy*. According to the same reference, the *accuracy* is function of the end-effector position and is usually below one millimeter. The *repeatability* is the ability of the robot to consistently return to a specified position. It depends not only on the mechanical parts but also on the controller and the sensors. With respect to [Siciliano *et al.* 2010], the *repeatability* is typically smaller than the *accuracy*. Figure 6.2 illustrates those principles of accuracy and repeatability.

From those definitions, we define the following relation:

$$\mathbf{q}(t) = \mathbf{q}_r(t) - \boldsymbol{\varepsilon}_{tr}^q(t) = \mathbf{q}_r(t) - (\mathbf{a}_{cc}(t) + \mathbf{r}_{ep}(t)) \quad (6.10)$$

where  $\boldsymbol{\varepsilon}_{tr}^q$  is the closed-loop tracking error divided in a deterministic accuracy vector  $\mathbf{a}_{cc}$  and a stochastic repeatability vector  $\mathbf{r}_{ep}$ . Although the accuracy and repeatability are defined in static framework, we consider them here within a dynamic model. The transient effects of the tracking law can be seen as a part of the accuracy vector  $\mathbf{a}_{cc}$ . At first approximation, the repeatability may be modelled by a white noise with a covariance matrix  $\mathbf{R}_r^2$ . Since  $\mathbf{q}_r$  is noise-free, the noise seen by the measurement position of link  $j$  is given by:

$$n_{q_j}(t) = -r_{ep_j}(t) + \tilde{q}_j(t), \quad (6.11)$$

with the sensor noise  $\tilde{q}_j(t) = H_j(z^{-1})e_j(t)$ . The goal of the control law is to make the tracking error  $\boldsymbol{\varepsilon}_{tr_j}^q$  negligible to set the tool at the right position in the workspace. Therefore, from a design perspective, the noise introduced by the sensor should have a mean and a covariance lower than those of the tracking error. Continuing this line of reasoning, it may even be written

$$n_{q_j}(t) \approx -r_{ep_j}(t). \quad (6.12)$$

According to [Stäubli Favergues 2015], the repeatability is  $\pm 0.02$  mm with a maximum reach of 450 mm between joint 1 and 5. Using the Pythagoras's theorem, it comes out the angular repeatability of joint 1:  $R_{r_1} = 2.5 \cdot 10^{-3}$  degree, assuming that the joints are perfectly rigid and that all the error comes from link 1. That is consistent with the sensor's angular resolution of  $2 \cdot 10^{-4}$  degree per count provided in section 2.2.3. On the other hand, the order of magnitude of the reference trajectory is  $10^2$  degree. Therefore, the output Signal to Noise Ratio (SNR) is approximately

$$SNR_{dB}^q = 10 \cdot \log_{10} \left( \left( \frac{A_{signal}}{A_{noise}} \right)^2 \right) = 92 \text{ dB}, \quad (6.13)$$

where  $A_{signal} = 10^2$  and  $A_{noise} = 2.5 \cdot 10^{-3}$  are the amplitudes of the signal and the noise respectively. In the light of this low SNR, the ultimate assumption is to neglect the noise on the joint positions and to consider them as noise-free. The consequence of such an assumption is that the circulation of the noise, on which is based the noise model (6.9), is neglected.

By considering (6.8) and (6.12), it comes out  $\mathbf{v}(t) = \mathbf{G}_\tau \mathbf{C}(p) \mathbf{r}_{ep}(t)$ . The deterministic component of the torque is then  $\boldsymbol{\tau}_{det}(t) = \mathbf{G}_\tau \mathbf{C}(p) \mathbf{a}_{cc}(t)$ . Around an operating point, there is  $\mathbf{C}(p) \approx \mathbf{C}_0$ . Therefore, the input SNR is approximately

$$SNR_{dB}^\tau = 10 \cdot \log_{10} \left( \left( \frac{a_{cc}}{r_{ep}} \right)^2 \right). \quad (6.14)$$

In the case of the TX40, the accuracy is not given by the manufacturer. According to [Siciliano *et al.* 2010], we know that the repeatability should be lower than the accuracy. If we assume that the ratio is  $a_{cc} = 10 \cdot r_{ep}$ , there is  $SNR_{dB}^{\tau} = 20$  dB. For example, with the accuracy and repeatability of the PUMA 560 provided in [Conrad *et al.* 2000], we have  $SNR_{dB}^{\tau} = 32$  dB. Those simple numerical applications show that there is much more noise in the input torque signals than in the output ones.

The orders of magnitude of the SNR found, for the input and output signals, are similar to what is observed in practice; see e.g. [Calanca *et al.* 2011]<sup>1</sup>. If it does not theoretically validate this empirical noise model, that illustrates its credibility.

### Amplified noise

The reader has to keep in mind that we study here the noise proper to the physical system and more specifically the noise seen by the input torques. As explained in section 4.2.1, the practitioner may also introduce another noisy component into the IDIM with the amplification of the measurement noise through the numerical differentiation of the joint positions. In fact, from the development of the section 4.2.1, we would have:

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_{idm}(t) + \mathbf{v}(t) + \mathbf{M}(t)\mathbf{n}_q(t)/T_s^2 \quad (6.15)$$

where  $\mathbf{M}$  is the inertia matrix. With a high sampling frequency, this amplified noise may become predominant. That shows why a specific attention must be paid to this issue. As illustrated in section 5.3.1, even with the IRWSM method, the practitioner must be careful.

### Retained noise model

As it has been seen, the question of the noise model is complex and our practical reasonings do not intend to perfectly model the circulation of this noise through the system. For our identification purpose, we will consider two cases:

$$v_j(t) = H_{\tau_j}(z^{-1})e_j(t), \quad (6.16)$$

$$v_j(t) = -g_{\tau_j}C_j(p)n_{q_j}(t) = -g_{\tau_j}C_j(p)H_j^c(z^{-1})e_j(t). \quad (6.17)$$

The idea of the first model (6.16) is to evaluate the noise filter with a black-box point of view. For reminder, the noise of the sensor is  $\tilde{q}_j(t) = H_j(z^{-1})e_j(t)$ . Assuming that the noise filter  $H_j$  just "shapes" the white noise but does not amplify it, it comes out  $var(\tilde{q}_j) \approx var(e_j)$ . With such an assumption, it is expected that the residual  $\hat{e}_j$  has a variance approximately equal to the one of the encoder; i.e.  $var(\hat{e}_j) \approx \Delta_e^2$ .

The second model (6.17) takes into account the controller relation and can thus be considered as a grey-box approach. Based on the theoretical development, we would have  $H_j^c = S_j H_j$ ; i.e. the sensor noise filter weighted by the closed-loop

<sup>1</sup>It should be noticed that the SNR definition is slightly different in this reference.

sensitivity function. In the opposite, based on the empirical approach,  $H_j^c$  would be a filter shaping the noisy part of the tracking error; i.e.  $n_{q_j}(t) = H_j^c(z^{-1})e_j(t) = -r_{ep_j}(t)$ . The idea of this second noise model, with two variants, is to see the controller influence and, especially, if that knowledge of the controller allows a better estimation. To distinguish between the two variants, the variance of the estimated position error  $\hat{n}_{q_j}$  can be examined.

- With the empirical model, the estimated position error  $\hat{n}_{q_j}$  should have a standard deviation nearly equal to the the repeatability, following (6.12).
- With the theoretical noise model,  $n_{q_j}(t) = S_j(p)\tilde{q}_j(t)$  should have a variance slightly lower than the one of the measurement noise. The transfer  $S_j$  is assumed to be a perfect high-pass filter such as

$$S_j(\omega) = \begin{cases} 0, & \text{if } \omega \in [-\omega_{dyn}, \omega_{dyn}] \\ 1, & \text{otherwise} \end{cases}. \quad (6.18)$$

It comes out:

$$\begin{aligned} \sigma_{n_{q_j}}^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} P_{\tilde{q}_j\tilde{q}_j}(\omega) |S_j(\omega)|^2 d\omega \\ &= \frac{1}{2\pi} \left( \int_{-\omega_{nyq}}^{-\omega_{dyn}} P_{\tilde{q}_j\tilde{q}_j} d\omega + \int_{\omega_{dyn}}^{\omega_{nyq}} P_{\tilde{q}_j\tilde{q}_j} d\omega \right) \\ &= \frac{1}{\pi} P_{\tilde{q}_j\tilde{q}_j} (\omega_{nyq} - \omega_{dyn}) = \sigma_{q_j}^2 - \omega_{dyn} P_{\tilde{q}_j\tilde{q}_j} / \pi \\ &= \sigma_{q_j}^2 \left( 1 - \frac{\omega_{dyn}}{\omega_{nyq}} \right), \end{aligned}$$

where  $P_{\tilde{q}_j\tilde{q}_j}$  is the power spectral density of  $\tilde{q}_j$  assumed to be white with a covariance  $\sigma_{q_j}^2 = \frac{1}{\pi} P_{\tilde{q}_j\tilde{q}_j} \cdot \omega_{nyq}$ , using the material from Appendix C.3. The previous estimation of the variance  $\sigma_{n_{q_j}}^2$  makes rough assumptions. However, it illustrates the order of magnitude which should be expected.

In any case, the noise is assumed to be modelled with a rational spectral density and to be independent between the links.

### 6.1.3 Auto-regressive filters identification

The noise model may be identified thanks to the instrumental variable approach for AutoRegressive - Moving - Average (ARMA) models described in [Young 2006]. This method is implemented in the `ivarma` function of the CAPTAIN Toolbox. For a given link, the function requires the sizes of the numerator and the denominator of the ARMA model. One solution consists in identifying just the AutoRegressive (AR) model which gives the best Akaike Information Criterion (AIC) without taking into account the Moving - Average (MA) filter [Akaike 1974]. That approach is implemented in the `aic` function of the CAPTAIN. Another solution consists in

estimating the sizes of MA and AR filters by a visual inspection of the autocorrelation and partial autocorrelation functions of the signals, as explained in Chapter 6 of [Box & Jenkins 1976].

Because our goal is to find an automatic solution which does not require the practitioner's skills, we focus on the aic technique. The user has to specify the maximum expected order of the AR filter and the function performs a line search to find the order corresponding to the best AIC. The AR filter is defined such as:

$$\begin{aligned}\xi(t_i) &= \frac{1}{C_l(z^{-1})}e(t_i) \\ &= -c_1\xi(t_{i-1}) - \dots - c_l\xi(t_{i-l}) + e(t_i),\end{aligned}\quad (6.19)$$

where  $l$  is the number of estimated parameters and  $e$  is the generating zero mean, normally distributed, white noise sequence with variance  $\lambda^2$ . With the autocovariance function:

$$R_k = \frac{1}{N} \sum_{i=1}^N \xi(t_{i-k})\xi(t_i), \quad k \geq 0 \quad (6.20)$$

it comes the Yule-Walker equation

$$R_m = - \sum_{k=1}^l c_k R_{m-k} + \lambda^2 \delta_{m,0} \quad (6.21)$$

with  $m = 0, \dots, l$  and  $\delta_{m,0}$  the Kronecker delta function. From those  $l+1$  equations, the linear system of  $l$  equations (6.22) provides the estimation of the coefficients with OLS. The last equation (6.23) provides an estimation of the covariance.

$$\begin{bmatrix} R_0 & R_1 \cdots & R_{l-1} \\ \vdots & \vdots & R_{l-2} \\ R_{l-1} & R_{l-2} \cdots & R_0 \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_l \end{bmatrix} = \begin{bmatrix} c - R_1 \\ \vdots \\ -R_l \end{bmatrix} \quad (6.22)$$

$$R_0 = - \sum_{k=1}^l c_k R_{-k} + \lambda^2 \quad (6.23)$$

In practice, the *en-bloc* estimation should be avoided due to potential conditioning issues. Several recursive and robust methods have been developed like the Burg method or the Levinson-Durbin algorithm; see e.g. [Burg 1975] and the section 10.1 of [Ljung 1999]. The AIC is defined by:

$$AIC = 2l - 2\log(L), \quad (6.24)$$

where  $L$  is the likelihood function. According to [Shibata 1976] for the AR process considered, it comes out:

$$AIC = 2l + N\log(\hat{\lambda}^2), \quad (6.25)$$

where  $\hat{\lambda}^2$  is the estimated covariance of the generating noise estimated with (6.23).



## 6.2 Refined identification methods

### 6.2.1 The separable PEM method

If the PEM is an attractive methodology, it means identifying the dynamic model and the noise model at the same time, as it is visible with (3.17). This raises two problems for a robotic application. Firstly, we have no prior information about the order of the filter. Secondly, the identification of both models at the same time may lead to a complex nonlinear optimisation problem, whereas it has been simplified with the PLR assumption. Therefore, inspired by the RIVC method, we propose a separable approach by considering the following error:

$$\varepsilon_{seppem}(t_k, \hat{\boldsymbol{\rho}}) = \mathbf{H}_{\tau}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}) \varepsilon_{didim}(t_k, \hat{\boldsymbol{\theta}}), \quad (6.26)$$

where  $\boldsymbol{\rho}$  is the  $(b + l \times 1)$  vector of unknown parameters, regrouped such as  $\boldsymbol{\rho}^T = [\boldsymbol{\theta}^T \quad \boldsymbol{\eta}^T]$ , and  $\boldsymbol{\eta}$  is the  $(l \times 1)$  vector regrouping all the noise parameters. The proposed methodology, a SEParable PEM (SEP-PEM), is composed of three sequential steps.

1. Identification of the **dynamic model**. The physical parameters are estimated with the DIDIM method without using the decimate filter.
2. Identification of the **noise model**. For each link  $j$ , obtain an estimate of the noise parameters,  $\hat{\boldsymbol{\eta}}_j$ , and an estimate of the noise covariance,  $\hat{\lambda}_j$ . The input noise filter  $H_{\tau_j}$  can be identified based on

$$\varepsilon_{didim_j}(t) = H_{\tau_j}(z^{-1}, \boldsymbol{\eta}_j) e_j(t), \quad (6.27)$$

for the black-box model or based on

$$-C_j^{-1}(p) \varepsilon_{didim_j}(t) = H_j^c(z^{-1}, \boldsymbol{\eta}_j) e_j(t), \quad (6.28)$$

for the grey-box model.

3. **Estimated covariance**. The estimated parametric error covariance matrix of the physical parameters is computed with the following relation

$$\boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}) = \left\{ \bar{E} \left[ \sum_{j=1}^n \left[ H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j) \boldsymbol{\phi}_j^s(t, \hat{\boldsymbol{\theta}}) \right]^T \hat{\lambda}_j^{-1} \left[ H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j) \boldsymbol{\phi}_j^s(t, \hat{\boldsymbol{\theta}}) \right] \right] \right\}^{-1} \quad (6.29)$$

where  $\boldsymbol{\phi}_j^s$  is  $j^{\text{th}}$  row of the simulated observation matrix  $\boldsymbol{\phi}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \ddot{\mathbf{q}}_s)^2$ .

The separation of the identification of the physical parameters and the one of the noise model implies that both models are statistically independent. To go through

<sup>2</sup>The simulated observation matrix is noted  $\boldsymbol{\phi}^s$  and not  $\boldsymbol{\zeta}$  because there is no filter involved.

the second step, we must be sure that the first one has converged and that there is no significant modelling error. However, as shown in section 4.1.4, the CLIE and DIDIM methods can converge to consistent parameters even without the decimate or the noise filter. Therefore, the first step of the method is admissible. This would not be the case with a direct approach since the noise and the dynamic models would be linked, as explained in section 3.1.

It should be noticed that in the algorithm above that we assume the matrix  $\mathbf{H}_\tau$  is diagonal. Following the development of the section 6.1.1, this assumption seems reasonable. Furthermore, it simplifies radically the process.

### 6.2.2 The IDIM-PIV method

Still inspired by the RIVC method, we reconsider the IDIM-IV method by taking into account the noise filter instead of the decimate one. The introduced technique is called Prefiltered IV (PIV). The term "refined" is not used because the RIVC theory has been developed for LTI systems. We do not develop here a theory for nonlinear systems, like the robots, but we modify the algorithm so that it can be applied on the IDIM. With the same notations as the SEP-PEM method, the IDIM-PIV algorithm is implemented as follows:

1. **Initialisation.** For the initial physical parameters,  $\hat{\boldsymbol{\theta}}^0$ , we use the CAD values for the inertia. The other physical parameters are set to zero. For each link, the initial noise filter is set as follows:  $H_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^0) = 1$  with  $\hat{\lambda}_j^0 = 1$ . If the controller is unknown, it must be identified with an approach described in section 5.2 in order to simulate the auxiliary model generating the instruments. Once the controller available, step 2 can be performed. The observation matrix should be constructed with the IRWSM technique in order to have a systematic process.
2. **Iteration:** repeat until convergence the following steps ( $k$  stands for the  $k^{th}$  iteration)
  - (a) Simulate the auxiliary model (i.e. the DDM) to retrieve the noise-free signals for the instruments by using  $\hat{\boldsymbol{\theta}}^{k-1}$ .
  - (b) Determine the IV estimate of the physical parameters using

$$\hat{\boldsymbol{\theta}}^k = \left[ \sum_{i=1}^{n_m} \sum_{j=1}^n \zeta_j^T(t_i, \hat{\boldsymbol{\rho}}^{k-1}) \phi_{L_j}(t_i, \hat{\boldsymbol{\rho}}^{k-1}) \right]^{-1} \left[ \sum_{i=1}^{n_m} \sum_{j=1}^n \zeta_j^T(t_i, \hat{\boldsymbol{\rho}}^{k-1}) \tau_{L_j}(t_i, \hat{\boldsymbol{\rho}}^{k-1}) \right] \quad (6.30)$$

with

$$\begin{aligned}\phi_{L_j}(t, \hat{\boldsymbol{\rho}}^{k-1}) &= L_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^{k-1}) \phi_j(\hat{\mathbf{q}}(t), \hat{\dot{\mathbf{q}}}(t), \hat{\ddot{\mathbf{q}}}(t)), \\ \zeta_j(t, \hat{\boldsymbol{\rho}}^{k-1}) &= L_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^{k-1}) \phi_j(\mathbf{q}_s(t, \hat{\boldsymbol{\theta}}^{k-1}), \dot{\mathbf{q}}_s(t, \hat{\boldsymbol{\theta}}^{k-1}), \ddot{\mathbf{q}}_s(t, \hat{\boldsymbol{\theta}}^{k-1})), \\ \tau_{L_j}(t, \hat{\boldsymbol{\rho}}^{k-1}) &= L_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^{k-1}) \tau_j(t), \\ L_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^{k-1}) &= (\hat{\lambda}_j^{k-1})^{-1} H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j^{k-1}).\end{aligned}$$

- (c) For each link  $j$ , obtain an estimate of the noise parameters,  $\hat{\boldsymbol{\eta}}_j^k$ , and an estimate of the noise covariance,  $\hat{\lambda}_j^k$ . The input noise filter  $H_{\tau_j}$  can be identified based either on (6.27) for the black-box noise model or on (6.28) for the grey-box noise model.

3. **Estimated covariance.** After convergence, based on (2.31), the estimated parametric error covariance matrix of the physical parameters is computed with the following relation

$$\Sigma(\hat{\boldsymbol{\theta}}) = \left\{ \bar{E} \left[ \sum_{j=1}^n \left[ H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j) \phi_j^{nf}(t, \hat{\boldsymbol{\theta}}) \right]^T \hat{\lambda}_j^{-1} \left[ H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j) \phi_j^{nf}(t, \hat{\boldsymbol{\theta}}) \right] \right] \right\}^{-1}. \quad (6.31)$$

In practice, the noise-free observation matrix  $\phi^{nf}$  is approximated by the simulated one  $\phi(\mathbf{q}_s, \dot{\mathbf{q}}_s, \ddot{\mathbf{q}}_s)$ .

If the method is really similar to the SEP-PEM method introduced in section 6.2.1, there are few discrepancies. Firstly, the IDIM-PIV method iteratively identifies the dynamic model then the noise filters. An error in the noise filters could thus contaminate the dynamic model and vice versa. The practitioner should be cautious. Secondly, since it is an IV method, it requires the observation matrix  $\phi$  and consequently the measurement of the joint positions. However, in the end, they share the same third step to estimate the covariance matrix.

### 6.2.3 Comments on the extensions

This section is devoted to three comments concerning the two extended methods proposed in previous sections. We first want to underline the philosophical difference between the decimate filter and the noise filter identification. In both cases, the goal is to provide white residuals to have consistent and minimal estimates of the variances. Both methods accept that the measurement noise cannot be white over all the frequency range. With the decimate filter, the noise is assumed to be white only over the two times the system's bandwidth. That explains the rule given in section 2.3.1. Consequently, the practitioner focuses only on this range and removes anything else. With the identification of the noise filter, the idea is to

whiten the residuals over the whole frequency range of the system. That may lead to two issues of which we must be aware. Firstly, the noise filters are most likely lowpass filter. Thus, their inverses amplify high frequency components like unmodelled flexibilities, electromagnetic disturbances in the motor or numerical artefacts coming from the integration solver. Secondly, the process is more time consuming than the decimate filter. As highlighted in section 4.1.4, with the decimate filter the optimisation solver has to consider  $N = 276$  sampling points against  $n_m = 34500$  with the filter identification.

The second element we want to discuss in this section is the optimality. The PEM and RIVC methods have been indeed developed in order to provide optimal estimates; i.e. the Cramér-Rao lower bound is reached. With the two extensions introduced here, we do not strictly speak about optimality. In fact, the PEM and RIVC theories were developed for LTI systems. Consequently, the optimality has not been demonstrated for nonlinear systems like robots. Our goal is not here to develop a theory of optimality in a nonlinear framework but only to refine/extend the usual methods.

Finally, the *aic* function described in section 6.1.3 is reconsidered. If this method is able to identify AR filters, it can be viewed as an auto-correlation information. The function indeed finds the order of the AR filter which gives the best Akaike's information criterion. That information indicates the range on which the signal is serially correlated. Therefore, the size of the AR filter can be used to tune the decimate filter such as  $n_d = size(\widehat{AR})$ . This solution makes the decimate process automated while avoiding the potential issues of the filtering by the inverses of the noise models described previously.

## 6.3 Experimental validations

### 6.3.1 Study of the black-box noise model

For the beginning, the black-blox noise model is tested with our SEParable PEM (SEP-PEM) method as well as the standard DIDIM method for comparison. For the DIDIM and SEP-PEM methods, the controller used for the simulation is the actual controller of the robot, also used in Chapter 4. It has been shown in the previous chapter that the identification of the controller is possible. Therefore, we consider it as known to separate the problem.

The DIDIM and SEP-PEM methods converged in 7 steps. Two elements are noteworthy concerning the values of the estimated parameters in Table 6.1. Firstly, the two iterative methods estimate dynamical parameters close to those estimated by the IDIM-LS method when the bandwidths are perfectly known; see Appendix B. The discrepancies are not critical since they fell in  $3\sigma$  confidence intervals. Secondly, it can be remarked that the SEP-PEM estimates converged to the same values than those of the DIDIM method. Those satisfactory results concerning the estimated parameters are confirmed by the low and equivalent relative estimation errors provided in Table 6.2.

Regarding the relative standard deviations of the DIDIM and SEP-PEM methods, they proved to be equivalent to the IDIM-LS results; see e.g. Table 5.13. Figure 6.3 depicts the autocorrelations of the identification residuals. For the SEP-PEM method, the effect of the identified noise filter is clear since the estimated autocorrelations coefficients are included in the confidence intervals indicated by the blue lines. If the estimated coefficients are larger for the DIDIM method, they can still be considered as negligible because they are also included in their confidence intervals. The differences between the intervals sizes are due to the number of samples considered for each method. These estimated autocorrelations prove that the DIDIM and SEP-PEM residuals can be considered as serially uncorrelated and consequently as white. With those results, it appears that the decimate filter that has the same effect as the inverse of the noise filter,  $\mathbf{H}_\tau^{-1}$ .

With regard to the input noise filters identified, the average order found is 141. The function thus needs in average 141 successive samples to whiten the noise. In other words, the signals are serially correlated over a range of 141 samples. With the Nyquist frequency  $\omega_{nyq}$ , the usual decimate filter keeps one sample over  $n_d = \omega_{nyq}/\omega_{Fp} = \omega_{nyq}/2/\omega_{dyn} = 125$  to insure a white noise. Those close numbers show that the decimate and the refined filters behave similarly. Therefore, that strengthens the idea suggested in section 6.2.3 of setting the decimate filter's cut-off frequency thanks to the aic function.

Beyond those encouraging results from an estimation perspective, we investigate the estimated noise

$$\hat{\varepsilon}_j(t) = H_{\tau_j}^{-1}(z^{-1}, \hat{\boldsymbol{\eta}}_j) \varepsilon_{didim_j}(t, \hat{\boldsymbol{\theta}}). \quad (6.32)$$

Table 6.3 gives the estimated covariances of  $\varepsilon_{didim_j}$  and  $\hat{\varepsilon}_j$  along with the squared encoders' resolutions. The DIDIM residuals are considered with and without the decimate filter, given by  $\varepsilon_{didim_j}$  and  $\varepsilon_{didim_j}^{ND}$  respectively. The decimation process indeed modifies the noise characteristics. To illustrate this property, we consider the relation  $y(k) = F_p(z^{-1})x(k) = x(k \cdot n_d)$  where  $F_p$  is the decimate filter with a ratio  $n_d$  and  $x$  is the input noise assumed to be white. With an ideal decimate filter, the transfer is given by

$$F_p(\omega) = \begin{cases} 1, & \text{if } \omega \in [-\omega_{Fp}, \omega_{Fp}] \\ 0, & \text{otherwise} \end{cases}. \quad (6.33)$$

In accordance with Appendix C.3,  $x$  is a white noise with a constant power spectral density  $P_{xx}(\omega) = P_e$  and a covariance  $\sigma_x^2 = \frac{\omega_{nyq}}{\pi} P_e$ . Therefore, the output noise

Table 6.1: Estimated parameters and relative standard deviations - Black-box noise model

Param.	DIDIM	SEP-PEM
$zz_{1_r}$	1.25 (1.23 %)	1.25 (1.09 %)
$fv_1$	7.98 (0.77 %)	7.98 (0.66 %)
$fc_1$	7.16 (2.42 %)	7.16 (2.31 %)
$xx_{2_r}$	-0.48 (3.06 %)	-0.48 (3.59 %)
$xz_{2_r}$	-0.15 (6.01 %)	-0.15 (7.73 %)
$zz_{2_r}$	1.08 (1.12 %)	1.08 (1.33 %)
$mx_{2_r}$	2.22 (3.32 %)	2.22 (1.34 %)
$fv_2$	5.44 (1.26 %)	5.44 (1.21 %)
$fc_2$	8.39 (1.89 %)	8.39 (1.52 %)
$xx_{3_r}$	0.13 (8.98 %)	0.13 (12.7 %)
$zz_{3_r}$	0.11 (8.48 %)	0.11 (9.40 %)
$my_{3_r}$	-0.59 (2.11 %)	-0.59 (3.32 %)
$ia_3$	0.09 (8.59 %)	0.09 (8.19 %)
$fv_3$	1.93 (1.61 %)	1.93 (1.90 %)
$fc_3$	6.50 (1.63 %)	6.50 (1.66 %)
$mx_4$	-0.03 (18.7 %)	-0.03 (22.0 %)
$ia_4$	0.03 (7.78 %)	0.03 (7.27 %)
$fv_4$	1.11 (1.51 %)	1.11 (1.41 %)
$fc_4$	2.43 (2.49 %)	2.43 (3.04 %)
$my_{5_r}$	-0.04 (10.9 %)	-0.04 (13.1 %)
$ia_5$	0.04 (9.76 %)	0.04 (10.8 %)
$fv_5$	1.82 (1.87 %)	1.82 (1.62 %)
$fc_5$	3.00 (2.99 %)	3.00 (3.21 %)
$ia_6$	0.01 (15.6 %)	0.01 (21.5 %)
$fv_6$	0.65 (1.62 %)	0.65 (1.53 %)
$fc_6$	0.26 (17.8 %)	0.26 (16.5 %)
$fv_{m6}$	0.60 (1.69 %)	0.60 (2.01 %)
$fc_{m6}$	1.92 (3.50 %)	1.92 (3.68 %)

Table 6.2: Relative estimation errors - Black-box noise model

DIDIM	SEP-PEM
5.56 %	5.56 %

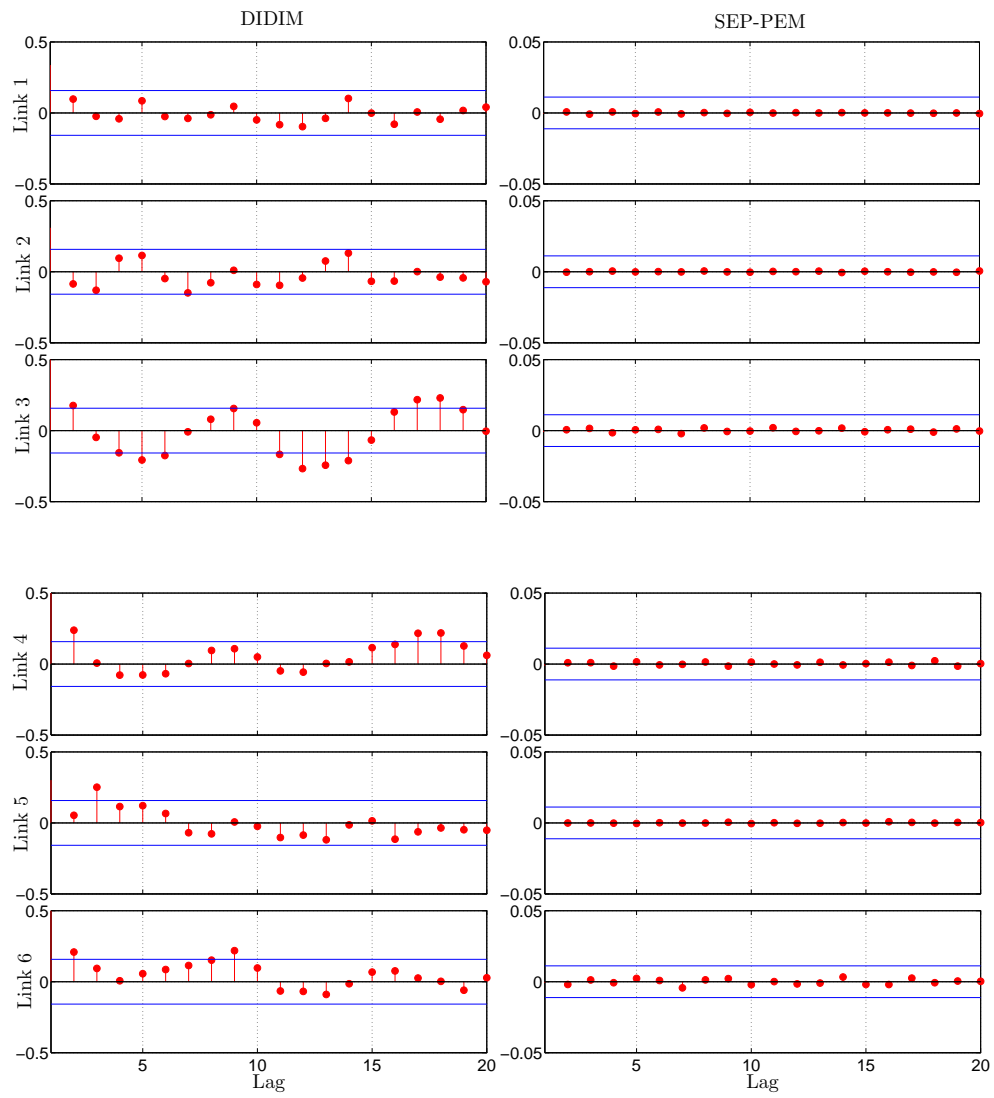


Figure 6.3: DIDIM (left) and SEP-PEM (right) residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines)

Table 6.3: Estimated noises covariances - Black-box noise model

Link	1	2	3	4	5	6
$\Delta_{e_j}^2$ ( $10^{-11}$ rad <sup>2</sup> )	0.0990	0.0990	0.4534	0.3959	0.4534	0.9012
$var(\hat{e}_j)$	34.07	21.14	3.931	0.3864	1.412	0.4946
$var(\varepsilon_{didim_j}^{ND})$	34.35	21.01	4.198	0.4400	1.425	0.5057
$var(\varepsilon_{didim_j})$	1.537	1.793	0.9120	0.2199	0.6207	0.2523

covariance is given by

$$\begin{aligned}
\sigma_y^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} P_{xx}(\omega) |F_p(\omega)|^2 d\omega \\
&= \frac{1}{2\pi} \int_{-\omega_{F_p}}^{\omega_{F_p}} P_e d\omega = \omega_{F_p} \frac{P_e}{\pi} \\
&= (\omega_{nyq}/n_d) \frac{P_e}{\pi} = \frac{\sigma_x^2}{n_d}.
\end{aligned} \tag{6.34}$$

In the present case, the input  $\varepsilon_{didim_j}^{ND}$  is not white and the filter is not ideal. However, (6.34) explains the difference of magnitude between the variances of  $\varepsilon_{didim_j}$  and  $\varepsilon_{didim_j}^{ND}$ . As it is visible, there is not a marked reduction of the orders of magnitude between the residuals: from  $\varepsilon_{didim_j}^{ND}$  to  $\hat{e}_j$ . In addition, the variances of the residuals  $\hat{e}_j$  are really far from the squared encoders resolutions. Consequently, they cannot be considered as the sensors noises. They encompass other unmodelled elements.

To conclude this first experimental validation, the SEP-PEM approach with the black-box noise model is able to consistently identify the dynamic model and the noise filters. Nonetheless, the exogenous noise estimated is not comparable to a sensor noise. The grey-box model might provide results with a better physical interpretation.

### 6.3.2 Study of the grey-box noise model

Since the SEP-PEM method has shown its relevance, the grey-box noise model is tested with the introduced IDIM-PIV one. The usual IDIM-IV method is also presented for comparison. For reminder, the particularity of the grey-box noise model is to filter the residuals by the inverse of the controller in order to identify the filter  $\mathbf{H}^c(z^{-1})$  defined in (6.17).

Table 6.4 summarizes the estimated values and their relative standard deviations. The IDIM-IV and IDIM-PIV methods respectively converged in 3 and 5 iterations. As it can be seen, the IDIM-PIV identified parameters are slightly different compared with those of the IDIM-IV method. However, those discrepancies are not critical since the reference parameters, estimated with the IDIM-LS method and known bandwidths, are still within the  $3\sigma$  confidence intervals of the IDIM-PIV estimates; see [Sargan 1958] for further details on the comparison between the IV and the LS methods. Those satisfactory results are confirmed by the similar relative errors provided in Table 6.5. Regarding the relative standard deviations, the IDIM-



PIV method performs as well as the IDIM-IV one providing that the decimate filter is properly tuned. The effect of the prefilter on the residuals is illustrated by the autocorrelation coefficients in Figure 6.4.

The critical point is the variances of the estimated position noises,  $\hat{\mathbf{n}}_q$ . Following the discussion of section 6.1.2, the order of magnitude of the standard deviation of  $\hat{\mathbf{n}}_q$  should be  $10^{-3}$  and  $10^{-4}$  degree (or  $1.7 \cdot 10^{-5}$  and  $1.7 \cdot 10^{-6}$  radian) respectively for the empirical variant (repeatability) and the theoretical variant (measurement noise). Table 6.6 provides the estimated values with a mean value of  $1.47 \cdot 10^{-2} \text{ rad}^2$  for  $\text{var}(\hat{n}_{q_j})$ , or  $\text{std}(\hat{n}_{q_j}) = 0.121 \text{ rad}$ . This result does not fit our models and suggests that the controller knowledge does not bring enough information for the noise modelling. Looking at  $\text{var}(\hat{n}_{q_j})$ , there is still a clear reduction compared with the variance of  $\varepsilon_{didim_j}^{ND}$  in Table 6.3 thanks to the filtering by the inverse of the controller.

Further investigations are hence necessary to find a physical interpretation of the observed noise. An idea could be to take into account more complete models of the actuators. That is strengthened by [Wernholt 2007] and the reference given therein where it is stressed that the torque ripples are "caused by distortion of the stator flux linkage distribution, variable magnetic reluctance at the stator slots, and secondary phenomena". Such an electronic phenomenon has not been taken into account in the current grey-box model.

### 6.3.3 Study with noisy data

In order to test the limit of the IDIM-PIV method, another identification is conducted using rather noisy data. The black-box noise model is considered for simplicity's sake. For this purpose, the resolution of the encoders is downgraded in an equivalent manner to [Marcassus *et al.* 2007]. The resolution is chosen equal to 3600 points per revolution. Three identification methods are considered: the IDIM-LS, IDIM-IV and IDIM-PIV methods. The observation matrix is constructed with the IRWSM for the IDIM-PIV method in order to have an automatic process.

Table 6.7 summarizes the estimated values and their relative standard deviations. The estimated parameters of the IDIM-LS method are not satisfactory due to large mismatches like the inertia  $zz_{1r}$  and  $zz_{3r}$ . This is explained by the noise correlation from the closed-loop that biases the estimation. The IDIM-IV and IDIM-PIV methods respectively converged in 3 and 5 iterations. The estimated values of the IDIM-IV and IDIM-PIV methods are similar to those found in previous works. The results of the IDIM-PIV method corroborate the relevance of the IRWSM to construct the observation matrix. Nonetheless, due to the noise introduced, the relative errors are larger in this case; see Table 6.8.

Concerning the relative standard deviations of the IDIM-PIV method, the values give a good indication on the order of magnitude of the optimal solution. They are slightly lower than those of the IDIM-IV method whereas the filters of the latter are correctly tuned. Furthermore, the estimated covariances of the IDIM-PIV method seem more reliable by looking at Figure 6.5 which depicts the autocorrelations of

Table 6.4: Estimated parameters and relative standard deviations - Grey-box noise model

Param.	IDIM-IV	IDIM-PIV
$zz_{1_r}$	1.25 (1.27%)	1.24 (1.17%)
$fv_1$	7.95 (0.67%)	8.01 (0.76%)
$fc_1$	7.23 (2.09%)	7.25 (2.13%)
$xx_{2_r}$	-0.47 (2.96%)	-0.46 (3.10%)
$xz_{2_r}$	-0.16 (5.02%)	-0.18 (6.02%)
$zz_{2_r}$	1.09 (1.02%)	1.11 (1.13%)
$mx_{2_r}$	2.25 (2.80%)	2.44 (1.95%)
$fv_2$	5.46 (1.08%)	5.51 (1.28%)
$fc_2$	8.32 (1.61%)	8.33 (1.87%)
$xx_{3_r}$	0.13 (9.13%)	0.14 (12.5%)
$zz_{3_r}$	0.12 (8.62%)	0.11 (9.94%)
$my_{3_r}$	-0.59 (2.37%)	-0.58 (3.17%)
$ia_3$	0.09 (9.27%)	0.09 (8.56%)
$fv_3$	1.93 (2.04%)	1.92 (2.29%)
$fc_3$	6.53 (2.05%)	6.54 (2.42%)
$mx_4$	-0.03 (28.3%)	-0.03 (29.6%)
$ia_4$	0.03 (14.0%)	0.03 (13.0%)
$fv_4$	1.10 (3.56%)	1.13 (2.96%)
$fc_4$	2.50 (5.73%)	2.52 (5.43%)
$my_{5_r}$	-0.04 (16.4%)	-0.04 (15.7%)
$ia_5$	0.04 (13.9%)	0.04 (11.1%)
$fv_5$	1.80 (2.77%)	1.81 (2.99%)
$fc_5$	3.05 (4.32%)	3.06 (4.05%)
$ia_6$	0.01 (20.9%)	0.01 (19.8%)
$fv_6$	0.65 (3.90%)	0.65 (3.79%)
$fc_6$	0.25 (15.9%)	0.26 (14.1%)
$fv_{m6}$	0.60 (3.26%)	0.59 (4.20%)
$fc_{m6}$	1.95 (5.86%)	1.95 (6.16%)

Table 6.5: Relative estimation errors - Grey-box model

IDIM-IV	IDIM-PIV
5.56 %	6.02 %

Table 6.6: Estimated noises covariances - Grey-box noise model

Link	1	2	3	4	5	6
$var(\hat{n}_{q_j})$	0.0035	0.0038	0.0064	0.0117	0.0398	0.0227
$var(\hat{e}_j) (10^{-4})$	0.1847	0.1086	0.4521	0.1185	0.0814	0.0682

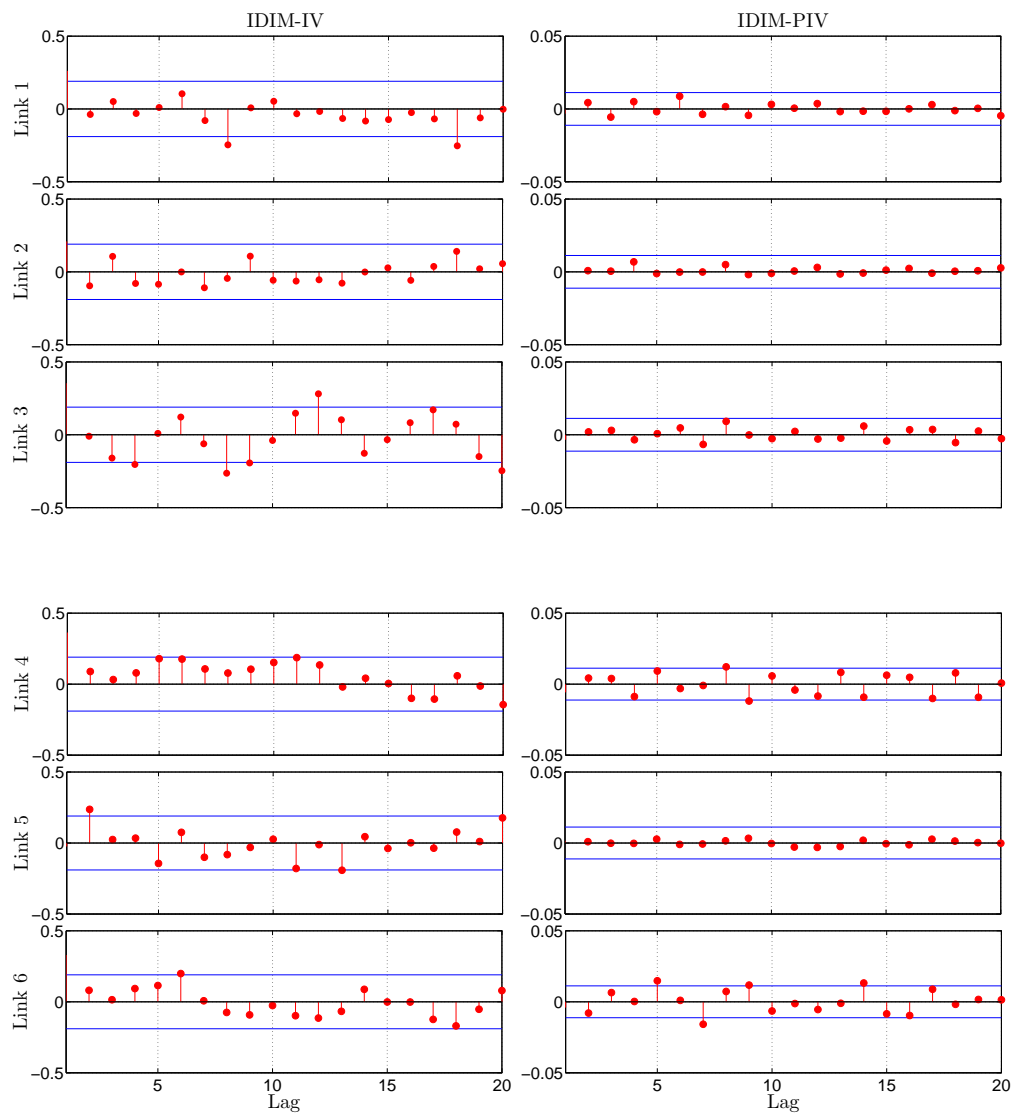


Figure 6.4: IDIM-IV (left) and IDIM-PIV (right) residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines)

the identification residuals. For the IDIM-PIV method, the effect of the refined input filter is clear since the estimated autocorrelations coefficients are included in the confidence intervals indicated by the blue lines. The estimated coefficients are larger for the IDIM-IV method and they cannot all be considered as negligible because some are not included in the confidence intervals. Therefore, the serial decorrelation is not insured.

A visual inspection of the residuals time series brings a limitation of those results. In fact, as illustrated with Figure 6.6, the residuals are not perfectly homoskedastic: the variance changes over time. Only Link 1 is presented for the sake of clarity. A correlation can be found with the acceleration signal. The idea would be that during acceleration phases the input noise is amplified. Here again, it appears the issue of the noise amplification during the estimation of the joint derivatives, as reminded in section 6.1.2.

## 6.4 Conclusions

In this chapter, the question of the noise circulation through the closed-loop system has been investigated. The underlying goal is to find an automated filtering of the experimental data and have consistent estimates with covariances as low as possible. Two noise models have been drawn in section 6.1. From those models, the DIDIM and IDIM-IV methods have been revised to achieve our goal. The methods along with the two noise models have been assessed with the identification of the TX40 based on experimental data in section 6.3. In practice, the methods have provided as accurate estimates as the usual ones. In the large noise case, the IDIM-PIV method has even provided estimates with lower covariances. Regarding the noise modelling, the black-box model should be preferred for now since the suggested grey-box model showed some limitations.

As far as a physical interpretation of the noise is concerned, further investigations should be conducted. The noise introduced by the position measurement does not seem indeed sufficient to explain everything. A finer model of the actuators and especially the motors could be of some interest for this purpose.

Table 6.7: Estimated parameters and relative standard deviations - Black-box noise model - Noisy data

Param.	IDIM-LS	IDIM-IV	IDIM-PIV
$zz_{1_r}$	<b>1.12</b> (2.43%)	1.24 (2.64%)	1.25 (1.20%)
$fv_1$	8.06 (1.23%)	7.61 (1.45%)	7.61 (1.12%)
$fc_1$	6.94 (3.95%)	8.34 (3.74%)	8.33 (2.89%)
$xx_{2_r}$	-0.43 (5.45%)	-0.46 (6.30%)	-0.46 (4.34%)
$xz_{2_r}$	-0.15 (7.87%)	-0.16 (10.6%)	-0.16 (8.71%)
$zz_{2_r}$	0.97 (1.94%)	1.13 (2.05%)	1.13 (1.70%)
$mx_{2_r}$	2.44 (3.72%)	2.25 (5.84%)	2.16 (1.66%)
$fv_2$	5.59 (1.93%)	5.11 (2.38%)	5.19 (2.48%)
$fc_2$	7.94 (3.09%)	9.23 (3.00%)	8.99 (3.22%)
$xx_{3_r}$	0.16 (13.8%)	0.14 (18.2%)	0.12 (16.2%)
$zz_{3_r}$	<b>-0.01</b> (265%)	0.11 (18.2%)	0.14 (10.9%)
$my_{3_r}$	-0.61 (4.20%)	-0.61 (4.77%)	-0.56 (4.50%)
$ia_3$	0.16 (8.68%)	0.09 (18.5%)	0.09 (11.2%)
$fv_3$	2.06 (3.59%)	1.77 (4.62%)	1.71 (4.12%)
$fc_3$	5.88 (4.21%)	7.23 (3.83%)	7.60 (3.07%)
$mx_4$	0.00 (272%)	-0.03 (50.2%)	-0.07 (16.5%)
$ia_4$	0.02 (39.6%)	0.03 (28.8%)	0.03 (18.0%)
$fv_4$	1.20 (6.21%)	0.97 (8.37%)	0.85 (4.26%)
$fc_4$	2.13 (12.6%)	3.13 (9.49%)	3.55 (3.68%)
$my_{5_r}$	-0.05 (19.6%)	-0.04 (30.4%)	-0.03 (24.0%)
$ia_5$	0.05 (16.7%)	0.05 (24.5%)	0.05 (16.0%)
$fv_5$	1.97 (4.80%)	1.55 (6.68%)	1.60 (4.13%)
$fc_5$	2.43 (10.1%)	3.82 (7.16%)	3.71 (4.66%)
$ia_6$	0.01 (40.7%)	0.01 (64.5%)	0.01 (20.5%)
$fv_6$	0.68 (6.79%)	0.59 (8.84%)	0.59 (3.96%)
$fc_6$	0.14 (146%)	0.19 (68.1%)	0.26 (59.3%)
$fv_{m6}$	0.62 (5.91%)	0.53 (7.57%)	0.56 (3.99%)
$fc_{m6}$	1.91 (11.0%)	2.35 (9.98%)	2.61 (4.97%)

Table 6.8: Relative estimation errors - Black-box noise model - Noisy data

IDIM-IV	IDIM-PIV
11.7 %	10.6 %

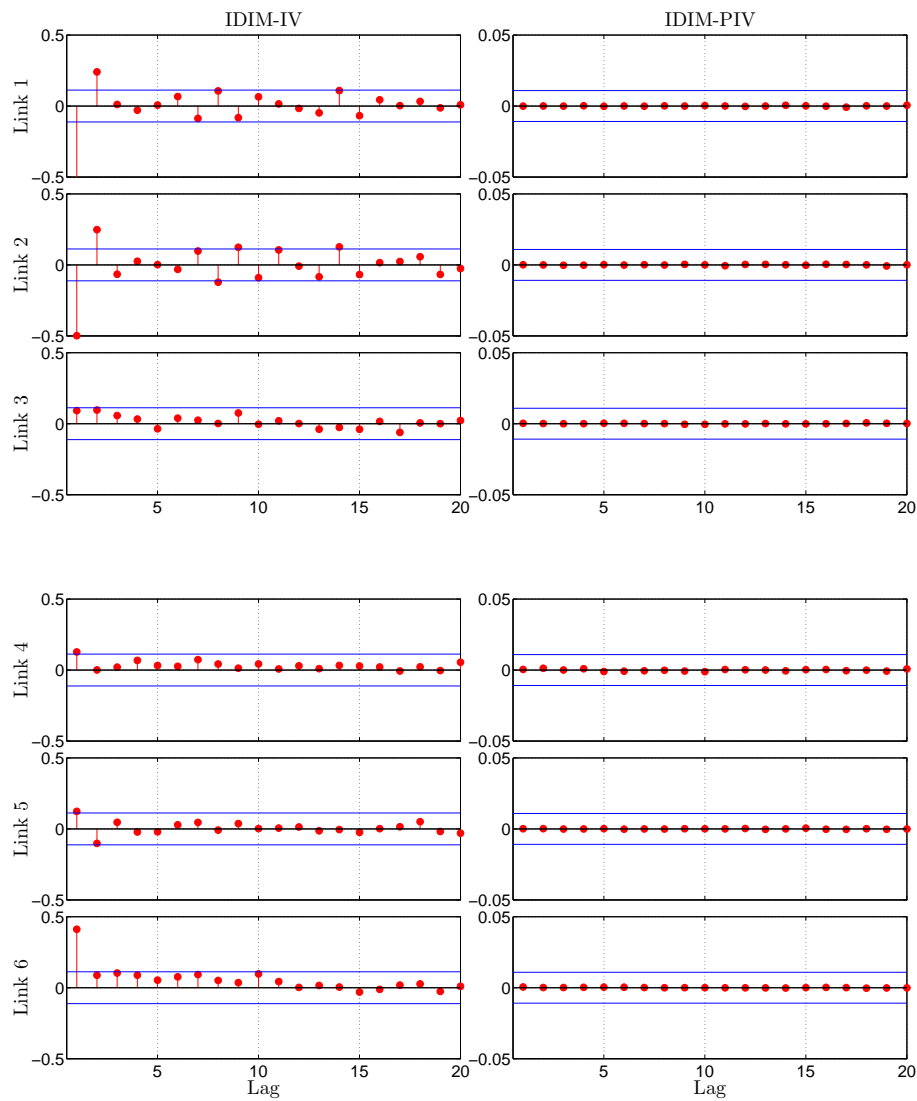


Figure 6.5: IDIM-IV (left) and IDIM-PIV (right) residuals autocorrelations (red dots) and  $2\sigma$  confidence intervals (blue lines) - Noisy data

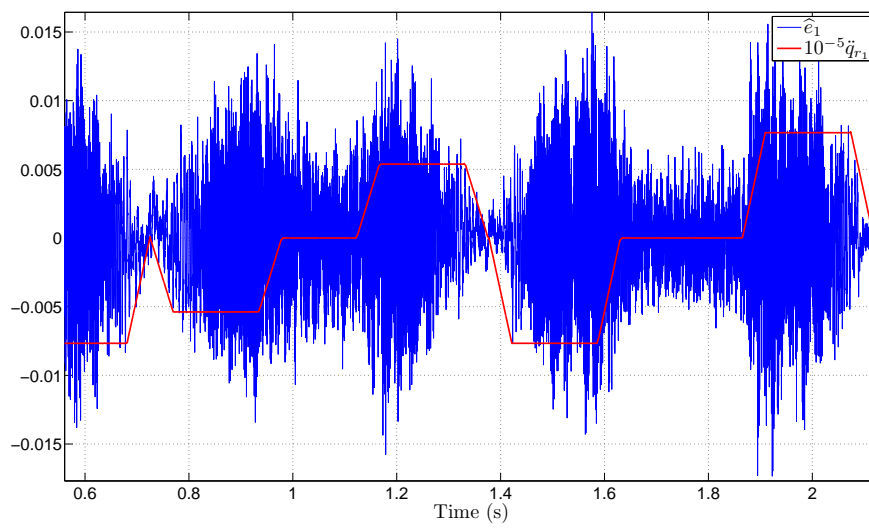


Figure 6.6: IDIM-PIV residuals,  $\hat{e}_1$ , and scaled reference acceleration,  $\ddot{q}_{r1}$  - Link 1





**Part IV**

**Conclusion**



# Discussions and Future Work

---

In this final chapter, an overview of the contributions presented in Part III is provided. The goal is to give a clear picture of how the problem of robotic systems identification has been tackled. The different methods that were presented have obviously their limitations. Those are recalled in this chapter along with hints for future works.

---

<b>7.1</b>	<b>Contributions summary . . . . .</b>	<b>124</b>
7.1.1	Analysis of the OEM for robot identification . . . . .	124
7.1.2	Robustness analysis of the IDIM-IV and DIDIM methods . . . . .	124
7.1.3	Identification with an unknown controller . . . . .	124
7.1.4	Automated estimation of the joint derivatives . . . . .	125
7.1.5	Noise filter identification . . . . .	125
<b>7.2</b>	<b>A practitioner guide . . . . .</b>	<b>126</b>
<b>7.3</b>	<b>Further developments . . . . .</b>	<b>126</b>
7.3.1	Actuator modelling . . . . .	126
7.3.2	Friction modelling . . . . .	126
7.3.3	Broader perspectives . . . . .	128

---

## 7.1 Contributions summary

### 7.1.1 Analysis of the OEM for robot identification

The preliminary comparison of the DIDIM and CLOE methods introduced in [Janot *et al.* 2014a] has been enlarged in Chapter 4 to consider the CLIE method. It has been highlighted the relation between the methods CLIE and DIDIM. The interest of considering those methods instead of the CLOE one has been shown. The input torque appears indeed to have a better sensitivity to the parameters. The difficulty of extending this result to non-robotic systems has been discussed in the same chapter. Furthermore, the role of the decimate filter for robot identification has been emphasized. It thus appeared the dual nature of the DIDIM method which is neither an output error method nor a prediction error one, because the measurement noise is neither white nor modelled.

### 7.1.2 Robustness analysis of the IDIM-IV and DIDIM methods

One of the major contributions of this thesis is the comparison of the IDIM-IV and DIDIM methods. Both methods proved to be effective solutions for robot identification. It is known that both methods are iterative and can share the same convergence criterion and initial values. By contrast, the DIDIM method does not require the same careful bandpass filtering as the IDIM-IV one to build the observation matrix [Gautier *et al.* 2013a]. The novelty lies in the demonstration of the better robustness of the IDIM-IV method to a modelling error located in the auxiliary model. That has been experimentally validated with the TX40 robot in Chapter 4. Furthermore, it has been also highlighted that a special attention must be paid to the evaluation of the estimates' covariance if any errors exist. This contribution was presented in [Brunot *et al.* 2017c].

### 7.1.3 Identification with an unknown controller

In this thesis work, considering our necessity to process data coming from closed-loop experiments, it has been decided to consider the case where the control law is not available. If some works have already dealt with this problem, they considered LTI systems only. To overcome the nonlinear feature of robotic models, the perspective has been changed in order to treat the identification of the controller separately from the rest of the system. Two scenarios have been envisaged depending on the practitioner's goal. Firstly, if the objective is to know the actual control law or if there is enough time available, the controller can be parametrically identified. That could be a complex task especially if nonlinearities are included in the law. Secondly, the practitioner can choose a non-parametric identification to spare time and focus on the dynamic model, at the cost of a less complete knowledge on the system. Those two approaches are quite novel and have been successfully tested with the TX40 robot in Chapter 5, by considering the IDIM-IV and DIDIM methods. This work has been submitted in [Brunot *et al.* Submitted 2017].

Perspectives include considering unknown models of the actuators as well as more complex control laws. An assessment of the control laws identifiability could also be of interest from an industrial perspective. For a robot manufacturer, the dual goal would be indeed to discover the competitor's control law and keep its own secret.

#### 7.1.4 Automated estimation of the joint derivatives

As mentioned previously, the usual identification methods, IDIM-LS and IDIM-IV, need a numerical differentiation of the measured position signal that comes with a bandpass filter to avoid noise amplification. In order to reduce the information necessary to deal with the setting of the filter, we wanted to find another automatic method for the differentiation of the measured signal. We have tested a solution, called IRWSM, based on a random walk model, a Kalman filter and a FIS that was originally developed for time varying parameters estimation. It has been shown that the random walk model does not exactly represent the dynamics of the robotic system. Nonetheless, it has been observed that the use of the IRWSM improves the performances, especially when the system's bandwidths are unavailable to the practitioner. However, some limitations to this contribution can be mentioned:

- A modified version of the model taking into account the reference trajectory as an input could be of some interest in a similar manner to [Kostic *et al.* 2004];
- The automated IRWSM approach allows a consistent estimation of the dynamic parameters but their standard deviations. Another procedure is required to automatically set the cut-off frequency of the decimate filter.

#### 7.1.5 Noise filter identification

To deal with the setting of the decimate filter, the usual approach of the system identification community has been investigated. The idea is to filter the data by the inverse of the estimated noise model. That technique has the advantage of insuring optimal estimates for LTI systems. If the optimality is not our target with the nonlinear robot models, a refinement of estimates is of some interest along with the automation of the process. To achieve such a goal, a discussion has been initiated on the noise seen by the identification algorithm coming from the whole closed-loop system. The identification of the noise model has been thought to be automatic without *a priori* knowledge on the system. The IDIM-IV and DIDIM methods have been reconsidered to integrate the noise filter identification as well as the resulting prefiltering. Furthermore, it appeared an alternative way to automatically estimate the order of the decimate filter thanks to the Akaike information criterion.

The experimental results have shown that the approach is suitable to identify an industrial robot systematically and can even improve the precision, based on a black-box noise model. Nonetheless, it appeared that our understanding of the noise circulation through the closed-loop system is still limited. A perspective would be to develop a more comprehensive grey-box noise model.

## 7.2 A practitioner guide

We reconsider the identification process depicted in Figure 1.3. Based on the contributions, Figure 7.1 describes our vision of the process including the developed methodologies. The *a priori* information block is replaced by three preliminary questions put to the user. Those three questions are the following:

- *Does the practitioner know the system's bandwidth?* If the answer is positive, the usual methods based on the bandpass and decimate filters can be used. Otherwise, the joint derivatives should be estimated with the IRWSM approach. Concerning the noise whitening, the practitioner can either identify a black-box noise model or tune the decimate filter thanks to the AIC. A third solution, which we would recommend, consists in coupling a roughly tuned decimate filter and the identification of a filter modelling the remaining noise. The cost of this solution is the lost of the physical interpretation of the identified noise model.
- *Does the practitioner know the system's controller?* If the answer is negative, the identification of the controller must be performed. If the objective is to use the DIDIM method or if there is an interest in the controller knowledge, a parametric identification must be conducted. Otherwise, a non-parametric identification coupled with the IDIM-IV method can achieve the goal.
- *Is there an interest in a statistical analysis?* If the answer is positive, the noise identification should be performed with a black-box or even a grey-box model. Otherwise, the usual decimate filtering can provide reliable estimated parameters in potentially less computation time.

## 7.3 Further developments

### 7.3.1 Actuator modelling

Regarding the modelling, the characterisation of the actuators would be an interesting development. Along this thesis, they have indeed been considered as known subsystems and more specifically as static gains. Their knowledge could improve the understanding of the noise model. In this regard, further efforts are needed to better understand the noise faced by the identification methods. A more comprehensive statistical analysis could be of interest.

### 7.3.2 Friction modelling

As shown in Chapter 2, the dynamic models are formulated from the Newton's laws or Lagrange's equations. The practitioner thus needs a complete knowledge of the physical phenomena. One major difficulty lies in the model of the friction which is usually nonlinear at low velocities and may depend on many exogenous parameters

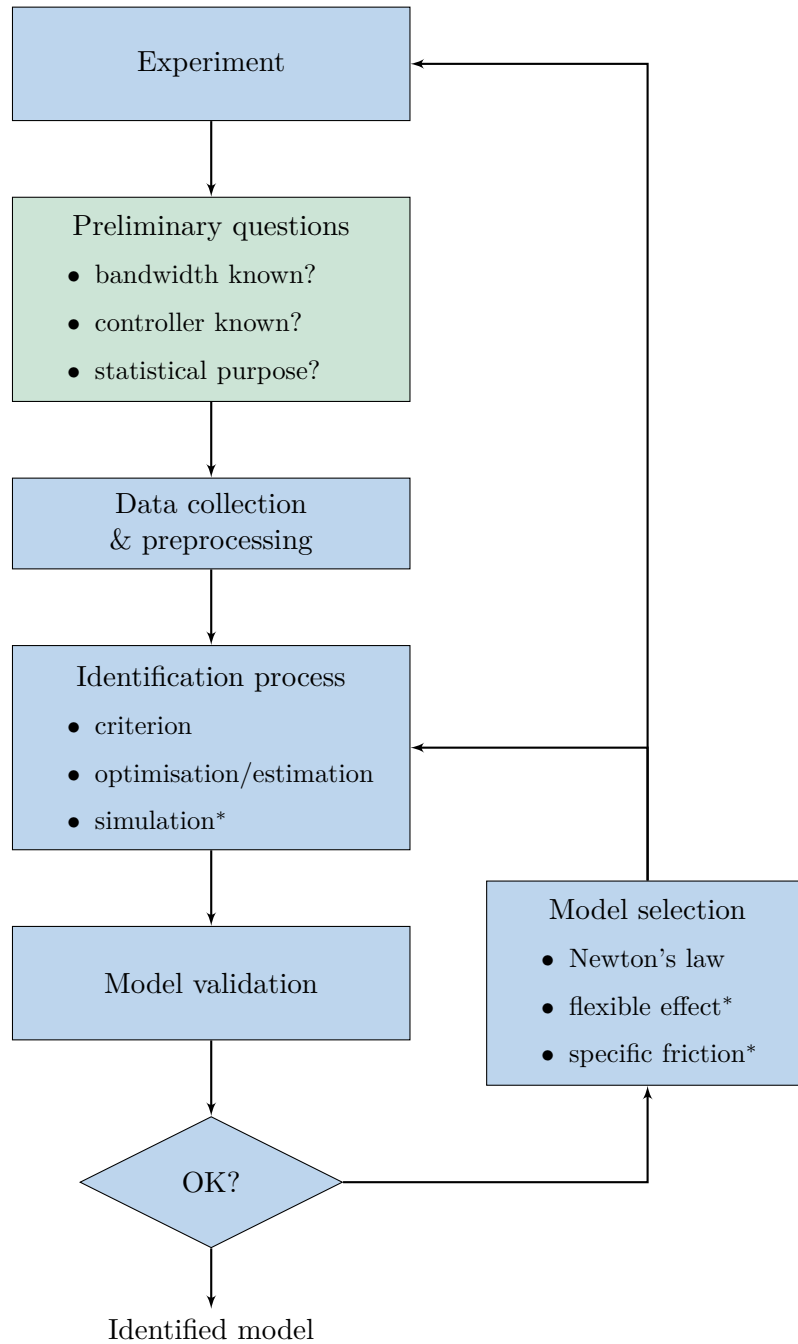


Figure 7.1: Block diagram of the revisited robot identification process

\*: not necessary for certain methods

like the temperature; see e.g. [Bittencourt & Axelsson 2014]. Therefore, the friction remains a major challenge for robot identification.

Recently, in [Janot *et al.* 2017], a new way of dealing with this challenge has been introduced. The idea is to separate the problem in two: a "grey-box" part regrouping the inertial forces and a "black-box" part to capture the nonlinear effects of the friction. The grey-box part is identified with an usual method described in section 2.3 like the IDIM-IV one. The black-box part is non-parametrically identified with the State-Dependent Parameter (SDP) method. In a second time, from the non-parametric model provided by the SDP method, a parametric model of the friction can be identified. This method is a statistical procedure able to identify the presence of nonlinearities in dynamic system models by giving their graphical shape. The procedure is based on experimental sampled data, with a minimum of assumptions about the nature of the nonlinearities. The SDP estimation algorithm is an extension of the stochastic approach to Time-Varying Parameter (TVP) estimation; see e.g. [Young 1999] and the prior references therein. The SDP method represents thus a valuable tool which should be further investigated in order to deal with more complex nonlinear friction phenomena. In addition, it could be of some interest to consider techniques like the one developed in [Noël & Schoukens 2017] where the model is divided in a linear part and a nonlinear one. The nonlinearity is then seen as an additional input applied to the underlying linear system.

### 7.3.3 Broader perspectives

As far as the robot identification is concerned, the amount of knowledge required for the process has been reduced in Chapter 5 and Chapter 6. Further developments should focus on considering the element in the process that has not been studied: the design of excitation signals. In fact, to a certain extent, the design of input signals for identification can require *a priori* knowledge.

Regarding the identification methods themselves, they could be adapted to work in a recursive framework. The goal would be ultimately to perform online identification for monitoring or adaptive control. Furthermore, the application of those methods to other closed-loop or nonlinear systems could be explored. This has been started with the study of the output error methods for nonlinear systems with undefined sensitivity functions [Brunot *et al.* 2017a].



**Part V**

**Appendices**



# Industrial robot modelling

---

This appendix summarizes the main results for robot modelling in order to have a comprehensive overview. All the elements presented here can be found in [Khalil & Dombre 2004] or [Siciliano *et al.* 2010] for instance.

## A.1 Computation of the energies

This section is devoted to the development of the kinetic and potential energies required to compute the Lagrangian of an industrial robot. As explained in section 2.1, those energies are linear with respect to the standard dynamic parameters. Let us consider a link  $j$  where

- $G_j$  and  $O_j$  are respectively the center of mass and the origin of link  $j$ ;
- $L_j$  is the  $(3 \times 1)$  vector from the  $O_{j-1}$  to  $O_j$  ;
- $S_j$  is the  $(3 \times 1)$  vector from the origin point to the center of gravity, defined in section 2.1.3.

### Kinetic energy

For a robot with  $n$  links, the kinematic energy is given by:

$$E = \sum_{j=1}^n E_j, \quad (\text{A.1})$$

with  $E_j$  the kinematic energy of link  $j$ . By introducing the  $(3 \times 1)$  vectors  $\omega_j$  and  $V_{G_j}$ , which are respectively the angular velocity and the linear velocity (at the center of gravity  $G_j$ ) of link  $j$ , defined in the inertial frame  $R_0$  (fixed base), it comes out:

$$E_j = \frac{1}{2} \left( \omega_j^T \mathbf{J}_{G_j} \omega_j + M_j V_{G_j}^T V_{G_j} \right) \quad (\text{A.2})$$

where  $\mathbf{J}_{G_j}$  is the inertia tensor at the center of gravity  $G_j$  and  $M_j$  is the mass of the link. By composition of the velocities, we have:

$$V_{G_j} = V_{O_j} + \omega_j \wedge S_j, \quad (\text{A.3})$$

with the cross-product  $\wedge$ . With respect to the Huygens-Steiner theorem, the inertia tensor at the origin is

$$\mathbf{J}_{O_j} = \mathbf{J}_{G_j} - M_j \widehat{S} \widehat{S}^T, \quad (\text{A.4})$$

where  $\widehat{\mathbf{S}}_j$  is the  $(3 \times 3)$  skew-symmetric matrix defined by the components of the vector  $\mathbf{S}_j$  in order to calculate the vector product  $\mathbf{S}_j \wedge \boldsymbol{\omega}_j$ . Hence (A.2) can be developed:

$$\begin{aligned} E_j &= \frac{1}{2} [\boldsymbol{\omega}_j^T \mathbf{J}_{O_j} \boldsymbol{\omega}_j - M_j \boldsymbol{\omega}_j^T \widehat{\mathbf{S}}_j \boldsymbol{\omega}_j + M_j (\mathbf{V}_{O_j} + \boldsymbol{\omega}_j \wedge \mathbf{S}_j)^T (\mathbf{V}_{O_j} + \boldsymbol{\omega}_j \wedge \mathbf{S}_j)] \\ &= \frac{1}{2} [\boldsymbol{\omega}_j^T \mathbf{J}_{O_j} \boldsymbol{\omega}_j + M_j \mathbf{V}_{O_j}^T \mathbf{V}_{O_j} + 2M_j \mathbf{V}_{O_j}^T (\boldsymbol{\omega}_j \wedge \mathbf{S}_j) \\ &\quad - M_j \boldsymbol{\omega}_j^T \widehat{\mathbf{S}}_j \boldsymbol{\omega}_j + M_j (\boldsymbol{\omega}_j \wedge \mathbf{S}_j)^T (\boldsymbol{\omega}_j \wedge \mathbf{S}_j)]. \end{aligned} \quad (\text{A.5})$$

Using the scalar triple product, it comes out:  $\mathbf{V}_{O_j}^T (\boldsymbol{\omega}_j \wedge \mathbf{S}_j) = \mathbf{S}_j^T (\mathbf{V}_{O_j} \wedge \boldsymbol{\omega}_j)$ .

Furthermore, it appears  $(\boldsymbol{\omega}_j \wedge \mathbf{S}_j)^T (\boldsymbol{\omega}_j \wedge \mathbf{S}_j) = (\widehat{\mathbf{S}}_j \boldsymbol{\omega}_j)^T (\widehat{\mathbf{S}}_j \boldsymbol{\omega}_j) = \boldsymbol{\omega}_j^T \widehat{\mathbf{S}}_j \boldsymbol{\omega}_j$ . Therefore, the kinematic energy of link  $j$  can be written as a function of the inertia and velocities defined in  $O_j$  such as:

$$E_j = \frac{1}{2} [\boldsymbol{\omega}_j^T \mathbf{J}_{O_j} \boldsymbol{\omega}_j + M_j \mathbf{V}_{O_j}^T \mathbf{V}_{O_j} + 2M_j \mathbf{S}_j^T (\mathbf{V}_{O_j} \wedge \boldsymbol{\omega}_j)] \quad (\text{A.6})$$

If (A.2) is linear in the elements of  $\mathbf{J}_{G_j}$ , it is not linear with respect to the distances of the vector  $\mathbf{S}_j$ . On the contrary, (A.6) has the advantage to be linear in the mass  $M_j$  and the elements of  $\mathbf{J}_{O_j}$  and  $\mathbf{M}\mathbf{S}_j$ ; i.e. the standard inertial parameters.

Until this point, the velocities were expressed with respect to the inertial frame,  $R_0$ . To be consistent with the standard parameters introduced in section 2.1.3, the kinematic energy should be expressed as a function of the velocities and the parameters in the frame  $R_j$ . We introduce the rotation matrix  $\mathbf{R}_0^j(\mathbf{q}_{nf})$ , from  $R_0$  to  $R_j$ , such as:

$$\begin{aligned} \boldsymbol{\omega}_j &= \mathbf{R}_0^j(\mathbf{q}_{nf})^j \boldsymbol{\omega}_j, & \mathbf{V}_{O_j} &= \mathbf{R}_0^j(\mathbf{q}_{nf})^j \mathbf{V}_{O_j}, \\ {}^j \mathbf{J}_{O_j} &= (\mathbf{R}_0^j(\mathbf{q}_{nf}))^T \mathbf{J}_{O_j} \mathbf{R}_0^j(\mathbf{q}_{nf}), & \mathbf{M}\mathbf{S}_j &= \mathbf{R}_0^j(\mathbf{q}_{nf})^j \mathbf{M}\mathbf{S}_j, \end{aligned}$$

with the superscript  $j$ , before the vectors, which indicates that it is expressed in frame  $R_j$ . Consequently, it appears a dependence on the relative positions,  $\mathbf{q}_{nf}$ , due to the rotations. To make the notation less cluttered, the dependence on the relative positions will not be expressed thereafter. Since it is a rotation matrix, we have  $(\mathbf{R}_0^j)^{-1} = (\mathbf{R}_0^j)^T = \mathbf{R}_j^0$ . Eq. (A.6) can be rewritten:

$$E_j = \frac{1}{2} [(\boldsymbol{\omega}_j)^T (\mathbf{R}_0^j)^T {}^j \mathbf{J}_{O_j} \mathbf{R}_0^j \boldsymbol{\omega}_j + M_j (\mathbf{V}_{O_j})^T \mathbf{V}_{O_j} + 2({}^j \mathbf{M}\mathbf{S}_j)^T \mathbf{R}_j^0 (\mathbf{V}_{O_j} \wedge \boldsymbol{\omega}_j)].$$

## Relations between velocities

In the previous part, the energy was expressed as a function of the translational and rotational velocity vectors. We need to express those velocities as functions of the joint derivatives. For that purpose, we define the prismatic boolean  $p_{r_j}$  which is equal to 0 or 1 if the joint is revolute or prismatic respectively. In practice, the

velocities of each link are recursively calculated such as:

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_{j-1} + p_{r_j} \dot{q}_{nf_j} \mathbf{z}_j, \quad (\text{A.7})$$

$$\mathbf{V}_{O_j} = \mathbf{V}_{O_{j-1}} + \boldsymbol{\omega}_{j-1} \wedge \mathbf{L}_j + p_{r_j} \dot{q}_{nf_j} \mathbf{z}_j, \quad (\text{A.8})$$

with  $\mathbf{z}_j$  the unit vector defined in section 2.1.1. Considering that the base is fixed, we have  $\mathbf{V}_{O_0} = \mathbf{0}_3$  and  $\boldsymbol{\omega}_0 = \mathbf{0}_3$ . It should be noticed that the vector  $\mathbf{L}_j$  contains  $q_{nf_j}$  if the link is prismatic. From those recursive relations, it comes the linear expressions:

$$\boldsymbol{\omega}_j = \mathbf{T}_\omega^j \dot{\mathbf{q}}_{nf}, \quad (\text{A.9})$$

$$\mathbf{V}_{O_j} = \mathbf{T}_V^j \dot{\mathbf{q}}_{nf}, \quad (\text{A.10})$$

with  $\mathbf{T}_\omega^j$  and  $\mathbf{T}_V^j$  two  $(3 \times n)$  jacobian matrices defined by:

$$\mathbf{T}_\omega^j = \begin{bmatrix} \mathbf{t}_{\omega 1}^j & \dots & \mathbf{t}_{\omega j}^j & \mathbf{0}_3 & \dots & \mathbf{0}_3 \end{bmatrix}, \quad (\text{A.11})$$

$$\mathbf{T}_V^j = \begin{bmatrix} \mathbf{t}_{V 1}^j & \dots & \mathbf{t}_{V j}^j & \mathbf{0}_3 & \dots & \mathbf{0}_3 \end{bmatrix}. \quad (\text{A.12})$$

As explained in [Siciliano *et al.* 2010], the  $(6 \times n)$  matrix  $\mathbf{T}^n$  is the manipulator *geometric Jacobian*

$$\mathbf{T}^n = \begin{bmatrix} \mathbf{T}_\omega^n \\ \mathbf{T}_V^n \end{bmatrix}. \quad (\text{A.13})$$

According to the recursive relations (A.7) and (A.8), the columns can be computed such as

$$\mathbf{t}_{\omega k}^j = p_{r_k}^- \mathbf{z}_k \quad (\text{A.14})$$

$$\mathbf{t}_{V k}^j = p_{r_k} \mathbf{z}_k + p_{r_k}^- \mathbf{z}_k \wedge \mathbf{L}_j, \quad (\text{A.15})$$

with  $1 \leq k \leq j$ .

With the scalar triple product, it comes out:

$$\begin{aligned} ({}^j \mathbf{M} \mathbf{S}_j)^T \mathbf{R}_j^0 (\mathbf{V}_{O_j} \wedge \boldsymbol{\omega}_j) &= (\boldsymbol{\omega}_j)^T \left( (\mathbf{R}_0^j)^T ({}^j \mathbf{M} \mathbf{S}_j) \wedge \mathbf{V}_{O_j} \right) \\ &= (\mathbf{T}_\omega^j \dot{\mathbf{q}}_{nf})^T \left( \mathbf{R}_0^j ({}^j \mathbf{M} \mathbf{S}_j) \wedge \mathbf{T}_V^j \dot{\mathbf{q}}_{nf} \right) \\ &= \dot{\mathbf{q}}_{nf}^T (\mathbf{T}_\omega^j)^T \widehat{\mathbf{M} \mathbf{S}_j} \mathbf{T}_V^j \dot{\mathbf{q}}_{nf}, \end{aligned}$$

where  $\widehat{\mathbf{M} \mathbf{S}_j} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}$  and  $\mathbf{R}_0^j ({}^j \mathbf{M} \mathbf{S}_j) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ .

At last, we have shown that the kinetic energy can also be written:

$$E = \frac{1}{2} \dot{\mathbf{q}}_{nf}^T \mathbf{M}(\mathbf{q}_{nf}) \dot{\mathbf{q}}_{nf}, \quad (\text{A.16})$$

with  $\mathbf{M}$  the  $(n \times n)$  inertia matrix. The diagonal element  $M_{jj}$  are the coefficients of  $\dot{q}_{nfj}^2/2$  and, for  $i \neq j$ ,  $M_{ij} = M_{ji}$  is equal to the coefficient of  $\dot{q}_{nfi}\dot{q}_{nfj}$ .

### Potential energy

The potential energy of the robot arm is defined as

$$U = \sum_{j=1}^n U_j = \sum_{j=1}^n -M_j \mathbf{g}^T (\mathbf{L}_{0,j} + \mathbf{S}_j) = \sum_{j=1}^n -\mathbf{g}^T (M_j \mathbf{L}_{0,j} + \mathbf{M} \mathbf{S}_j) \quad (\text{A.17})$$

with  $\mathbf{L}_{0,j}$  the  $(3 \times 1)$  vectors from the origin of the base  $O_0$  to  $O_j$  and  $\mathbf{g}$  the  $(3 \times 1)$  vector of gravity. The potential energy is thus linear with respect to  $M_j$  and the elements of  $\mathbf{M} \mathbf{S}_j$ . Here also the vectors must be projected in the same frame. It is visible in (A.17) that the potential energy is not function of the velocities:  $\frac{\partial U}{\partial \dot{\mathbf{q}}_{nf}} = 0$ .

### Lagrangian

As a reminder, the Lagrangian is defined such as  $L = E - U$  and the input torque is calculated with (2.2). Therefore, it comes out:

$$\begin{aligned} \tau_{idm} &= \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}_{nf}} \right) - \frac{\partial L}{\partial \mathbf{q}_{nf}} + \boldsymbol{\tau}_f \quad (\text{A.18}) \\ &= \frac{d}{dt} \left( \frac{\partial E}{\partial \dot{\mathbf{q}}_{nf}} \right) - \frac{\partial E}{\partial \mathbf{q}_{nf}} - \frac{\partial U}{\partial \mathbf{q}_{nf}} + \boldsymbol{\tau}_f \\ &= \frac{d}{dt} \left( \mathbf{M}(\mathbf{q}_{nf}) \dot{\mathbf{q}}_{nf} \right) - \frac{\partial E}{\partial \mathbf{q}_{nf}} - \frac{\partial U}{\partial \mathbf{q}_{nf}} + \boldsymbol{\tau}_f \\ &= \dot{\mathbf{M}}(\mathbf{q}_{nf}) \dot{\mathbf{q}}_{nf} + \mathbf{M}(\mathbf{q}_{nf}) \ddot{\mathbf{q}}_{nf} - \frac{\partial E}{\partial \mathbf{q}_{nf}} - \frac{\partial U}{\partial \mathbf{q}_{nf}} + \boldsymbol{\tau}_f \\ &= \mathbf{M}(\mathbf{q}_{nf}) \ddot{\mathbf{q}}_{nf} + \left( \dot{\mathbf{M}}(\mathbf{q}_{nf}) \dot{\mathbf{q}}_{nf} - \frac{\partial E}{\partial \mathbf{q}_{nf}} \right) + \boldsymbol{\tau}_f - \frac{\partial U}{\partial \mathbf{q}_{nf}}. \end{aligned}$$

This proves the IDM relation (2.3) with  $\mathbf{C}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) \dot{\mathbf{q}}_{nf} = \dot{\mathbf{M}}(\mathbf{q}_{nf}) \dot{\mathbf{q}}_{nf} - \frac{\partial E}{\partial \mathbf{q}_{nf}}$ . Noting that for any  $i$  and  $j$ :

$$\dot{M}_{ij}(\mathbf{q}_{nf}) = \sum_{k=1}^n \frac{\partial q_{nfk}}{\partial t} \frac{\partial M_{ij}(\mathbf{q}_{nf})}{\partial q_k} = \sum_{k=1}^n \frac{\partial M_{ij}(\mathbf{q}_{nf})}{\partial q_k} \dot{q}_{nfk}, \quad (\text{A.19})$$

and by using (A.16), it comes out:

$$\mathbf{C}_{ij}(\mathbf{q}_{nf}, \dot{\mathbf{q}}_{nf}) = \sum_{k=1}^n c_{ijk} \dot{q}_{nfk}, \quad (\text{A.20})$$

with

$$c_{ijk} = \frac{\partial M_{ij}(\mathbf{q}_{nf})}{\partial q_k} - \frac{1}{2} \frac{\partial M_{jk}(\mathbf{q}_{nf})}{\partial q_i}. \quad (\text{A.21})$$

## A.2 Newton-Euler formulation

As suggested by [Khalil & Dombre 2004], the Newton-Euler equations can be written such as

$$\begin{aligned}\mathbf{F}_j &= M_j \dot{\mathbf{V}}_{O_j} + \dot{\boldsymbol{\omega}}_j \wedge \mathbf{M}S_j + \boldsymbol{\omega}_j \wedge (\boldsymbol{\omega}_j \wedge \mathbf{M}S_j) \\ \mathbf{M}_j &= \mathbf{J}_{O_j} \dot{\boldsymbol{\omega}}_j + \mathbf{M}S_j \wedge \dot{\mathbf{V}}_{O_j} + \boldsymbol{\omega}_j \wedge (\mathbf{J}_{O_j} \boldsymbol{\omega}_j)\end{aligned}\quad (\text{A.22})$$

where

- $\mathbf{F}_j$  is the  $(3 \times 1)$  vector of total forces applied to the link  $j$ ;
- $\mathbf{M}_j$  is the  $(3 \times 1)$  vector of total torques applied to the link  $j$  about  $O_j$ .

Those equations are obtained with two recursions.

### Forward recursion

The forward recursion calculates the vectors  $\boldsymbol{\omega}_j$ ,  $\dot{\boldsymbol{\omega}}_j$  and  $\dot{\mathbf{V}}_{O_j}$  with  $j$  from 1 to  $n$ . The rotational speed vector  $\boldsymbol{\omega}_j$  is obtained with relation (A.7). By differentiating (A.7) and (A.8), we obtain

$$\begin{aligned}\dot{\boldsymbol{\omega}}_j &= \dot{\boldsymbol{\omega}}_{j-1} + \bar{p}_{r_j} \left( \ddot{q}_{nf_j} \mathbf{z}_j + \boldsymbol{\omega}_{j-1} \wedge \dot{q}_{nf_j} \mathbf{z}_j \right), \\ \dot{\mathbf{V}}_{O_j} &= \dot{\mathbf{V}}_{O_{j-1}} + \dot{\boldsymbol{\omega}}_{j-1} \wedge \mathbf{L}_j + \boldsymbol{\omega}_{j-1} \wedge (\boldsymbol{\omega}_{j-1} \wedge \mathbf{L}_j) + p_{r_j} \left( \ddot{q}_{nf_j} \mathbf{z}_j + 2\boldsymbol{\omega}_{j-1} \wedge \dot{q}_{nf_j} \mathbf{z}_j \right).\end{aligned}\quad (\text{A.23})$$

If the base is fixed, the recursion is initialized with  $\boldsymbol{\omega}_0 = \dot{\boldsymbol{\omega}}_0 = \dot{\mathbf{V}}_{O_0} = \mathbf{0}_3$ .

### Backward recursion

The backward recursion calculates the total forces and moments on each link:

$$\begin{aligned}\mathbf{F}_j &= \mathbf{f}_j - \mathbf{f}_{j+1} + M_j \mathbf{g} - \mathbf{f}_{ej}, \\ \mathbf{M}_j &= \mathbf{m}_j - \mathbf{m}_{j+1} - \mathbf{L}_{j+1} \wedge \mathbf{f}_{j+1} + \mathbf{S}_j \wedge M_j \mathbf{g} - \mathbf{m}_{ej},\end{aligned}\quad (\text{A.24})$$

where

- $\mathbf{f}_j$  is the  $(3 \times 1)$  vector of total forces applied to link  $j$  from the previous link and the actuator  $j$ ;
- $\mathbf{f}_{ej}$  is the  $(3 \times 1)$  vector of total forces applied to link  $j$  from the environment;
- $\mathbf{m}_j$  is the  $(3 \times 1)$  vector of total moments applied to link  $j$  from the previous link and the actuator  $j$ , about  $O_j$ ;
- $\mathbf{m}_{ej}$  is the  $(3 \times 1)$  vector of total forces applied to link  $j$  from the environment, about  $O_j$ .

The recursion is initialized with  $\mathbf{f}_{n+1} = \mathbf{m}_{n+1} = \mathbf{0}_3$ . To avoid an explicit formulation of the gravity  $\mathbf{g}$  in the equation, we can set  $\dot{\mathbf{V}}_{O_0} = -\mathbf{g}$ . Therefore, it comes out

$$\begin{aligned}\mathbf{f}_j &= \mathbf{F}_j + \mathbf{f}_{j+1} + \mathbf{f}_{ej}, \\ \mathbf{m}_j &= \mathbf{M}_j + \mathbf{m}_{j+1} - \mathbf{L}_{j+1} \wedge \mathbf{f}_{j+1} + \mathbf{m}_{ej}.\end{aligned}\quad (\text{A.25})$$

The vectors  $\mathbf{F}_j$  and  $\mathbf{M}_j$  are calculated with the relations (A.22). The input torque or force, for link  $j$ , is then calculated with

$$\tau_{idm_j} = \left( p_{r_j} \mathbf{f}_j + p_{\bar{r}_j} \mathbf{m}_j \right)^T \mathbf{z}_j + \tau_{f_j} + I a_j \ddot{q}_{nf_j}. \quad (\text{A.26})$$

### A.3 Parameter reduction

In this sections, the two methods to retrieve the base parameters are summarized. The first one is a formal method whereas the second one is a numerical approach using the QR factorisation. As stressed in 2.1.3, the base parameters are the only identifiable parameters with IDM.

#### Formal method

The total energy of the link  $j$  is defined as the sum of the kinematic and potential energies. This is therefore linear with respect to the standard parameters as the Lagrangian:

$$H^j = E^j + U^j = \mathbf{h}_j {}^j \boldsymbol{\chi}_{I_j}, \quad (\text{A.27})$$

with  $\mathbf{h}_j$ , the  $(1 \times 10)$  row vector of the total energy of the link  $j$  and  ${}^j \boldsymbol{\chi}_{I_j}$  the  $(10 \times 1)$  vector of the standard inertial parameters of the link  $j$ ; see section 2.1.3. The elements of  $\mathbf{h}_j$  can be obtained with the recursion:

$$\mathbf{h}_j = \mathbf{h}_{j-1} {}^{j-1} \boldsymbol{\lambda}_j + \dot{q}_{nf_j} \boldsymbol{\mu}_j, \quad (\text{A.28})$$

where

- ${}^{j-1} \boldsymbol{\lambda}_j$  is a  $(10 \times 10)$  matrix function of the geometric parameters of frame  $R_j$ , see Table A.1, which gives  ${}^{j-1} \boldsymbol{\xi}_{I_j} = {}^{j-1} \boldsymbol{\lambda}_j {}^j \boldsymbol{\chi}_{I_j}$ ;
- $\boldsymbol{\mu}_j$  is a  $(1 \times 10)$  row vector written as:

$$\begin{aligned}\boldsymbol{\mu}_j &= p_{\bar{r}_j} \left[ 0 \quad 0 \quad \omega_{1,j} \quad 0 \quad \omega_{2,j} \quad \left( \omega_{3,j} - \frac{1}{2} \dot{q}_{nf_j} \right) \quad V_{2,j} \quad -V_{1,j} \quad 0 \quad 0 \right] \\ &\quad + p_{r_j} \left[ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\omega_{2,j} \quad \omega_{1,j} \quad 0 \quad \left( V_{3,j} - \frac{1}{2} \dot{q}_{nf_j} \right) \right],\end{aligned}\quad (\text{A.29})$$

where  ${}^j \boldsymbol{\omega}_j = [\omega_{1,j} \quad \omega_{2,j} \quad \omega_{3,j}]^T$  and  ${}^j \mathbf{V}_j = [V_{1,j} \quad V_{2,j} \quad V_{3,j}]^T$ .



In order to take into account the total inertia moment for rotor and gears of the actuator, the eleventh parameter  $Ia_j$  is introduced with  $h_{11,j} = \frac{1}{2}\dot{q}_{nf_j}$ . Two operations allow obtaining the base parameters:

- **elimination** of an inertial parameter if it has no effect on the dynamic (i.e.  $h_j$  constant);
- **grouping** of the parameter  $\chi_{I_j}$  with the parameters  $\chi_{I_j}^p$ ,  $p = 1 \dots r$ , if the corresponding energy function can be written  $h_j = \sum_{p=1}^r k_{j,p} h_{j,p} + \text{constant}$ , with constant coefficients  $k_{j,p}$ .

From the recursive relation (A.28), the following algorithm can be implemented. We define  $r_1$  the first revolute joint from the base and  $r_2$  the subsequent revolute joint whose axis is not parallel to the axis of joint  $r_1$ . For  $j = n \dots 1$ :

- 1) If  $p_{r_j} = 0$  (revolute joint), the parameters  $yy_j$ ,  $mz_j$  and  $M_j$  can be regrouped such as:

$$\begin{aligned}
 xx_{jr} &= xx_j - yy_j \\
 xx_{(j-1)r} &= xx_{j-1} + yy_j + 2r_j mz_j + r_j^2 M_j \\
 xy_{(j-1)r} &= xy_{j-1} + d_j S_{\alpha_j} mz_j + d_j r_j S_{\alpha_j} M_j \\
 xz_{(j-1)r} &= xz_{j-1} - d_j C_{\alpha_j} mz_j - d_j r_j C_{\alpha_j} M_j \\
 yy_{(j-1)r} &= yy_{j-1} + CC_{\alpha_j} yy_j + 2r_j CC_{\alpha_j} mz_j + (d_j^2 + r_j^2 CC_{\alpha_j}) M_j \\
 yz_{(j-1)r} &= yz_{j-1} + CS_{\alpha_j} yy_j + 2r_j CS_{\alpha_j} mz_j + r_j^2 CS_{\alpha_j} M_j \\
 zz_{(j-1)r} &= zz_{j-1} + SS_{\alpha_j} yy_j + 2r_j SS_{\alpha_j} mz_j + (d_j^2 + r_j^2 SS_{\alpha_j}) M_j \\
 mx_{(j-1)r} &= mx_{j-1} + d_j M_j \\
 my_{(j-1)r} &= my_{j-1} - S_{\alpha_j} mz_j - r_j S_{\alpha_j} M_j \\
 mz_{(j-1)r} &= mz_{j-1} + C_{\alpha_j} mz_j + r_j C_{\alpha_j} M_j \\
 M_{(j-1)r} &= M_{j-1} + M_j.
 \end{aligned}$$

Table A.1: Expression of  $j^{-1}\mathbf{A}_j$ 

$CC_\theta$	$-2CS_\theta$	0	$SS_\theta$	0	0	0	0	0	$2r$	$r^2$
$CS_\theta C_\alpha$	$(CC_\theta - SS_\theta)C_\alpha$	$-C_\theta S_\alpha$	$-CS_\theta C_\alpha$	$S_\theta S_\alpha$	0	$-dS_\theta C_\alpha + rC_\theta S_\alpha$	$-dC_\theta C_\alpha - rS_\theta S_\alpha$	$dS_\alpha$	$drS_\alpha$	
$CS_\theta S_\alpha$	$(C_\theta - SS_\theta)S_\alpha$	$C_\theta C_\alpha$	$-CS_\theta S_\alpha$	$-S_\theta C_\alpha$	0	$-dS_\theta S_\alpha - rC_\theta C_\alpha$	$-dC_\theta S_\alpha + rS_\theta C_\alpha$	$-dC_\alpha$	$drC_\alpha$	
$SS_\theta CC_\alpha$	$2CS_\theta CC_\alpha$	$-2S_\theta CS_\alpha$	$CC_\theta CC_\alpha$	$-2C_\theta CS_\alpha$	$SS_\alpha$	$2(dC_\theta S_\alpha + rS_\theta CS_\alpha)$	$2(-dS_\theta + rC_\theta CS_\alpha)$	$2rCC_\alpha$	$d^2 + r^2 CC_\alpha$	
$SS_\theta CS_\alpha$	$2CS_\theta CS_\alpha$	$S_\theta (CC_\alpha - SS_\alpha)$	$CC_\theta CS_\alpha$	$C_\theta (CC_\alpha - SS_\alpha)$	$-CS_\alpha$	$rS_\theta (SS_\alpha - CC_\alpha)$	$rC_\theta (SS_\alpha - CC_\alpha)$	$2rCS_\alpha$	$r^2 CS_\alpha$	
$SS_\theta SS_\alpha$	$2CS_\theta SS_\alpha$	$2S_\theta CS_\alpha$	$CC_\theta SS_\alpha$	$2C_\theta CS_\alpha$	$CC_\alpha$	$2(dC_\theta - rS_\theta CS_\alpha)$	$2(-dS_\theta - rC_\theta CS_\alpha)$	$2rSS_\alpha$	$d^2 + r^2 SS_\alpha$	
0	0	0	0	0	0	$C_\theta$	$-S_\theta$	0	$d$	
0	0	0	0	0	0	$S_\theta C_\alpha$	$C_\theta C_\alpha$	$-S_\alpha$	$-rS_\alpha$	
0	0	0	0	0	0	$S_\theta S_\alpha$	$C_\theta S_\alpha$	$C_\alpha$	$rC_\alpha$	
0	0	0	0	0	0	0	0	0	1	

With  $CS_* = \cos(*)\sin(*)$ ,  $CC_* = \cos^2(*)$  and  $SS_* = \sin^2(*)$ . The dependence in  $j$  is omitted.

- 2) If  $p_{r_j} = 1$  (prismatic joint), the parameters of  $\mathbf{J}_j$  can be regrouped such as:

$$\begin{aligned}
xx_{(j-1)r} &= xx_{j-1} + CC_{\theta_j}xx_j - 2CS_{\theta_j}xy_j + SS_{\theta_j}yy_j \\
xy_{(j-1)r} &= xy_{j-1} + CS_{\theta_j}C_{\alpha_j}xx_j + (CC_{\theta_j} - SS_{\theta_j})C_{\alpha_j}xy_j - C_{\theta_j}S_{\alpha_j}xz_j \\
&\quad - CS_{\theta_j}C_{\alpha_j}yy_j + S_{\theta_j}S_{\alpha_j}yz_j \\
xz_{(j-1)r} &= xz_{j-1} + CS_{\theta_j}S_{\alpha_j}xx_j + (CC_{\theta_j} - SS_{\theta_j})S_{\alpha_j}xy_j + C_{\theta_j}C_{\alpha_j}xz_j \\
&\quad - CS_{\theta_j}S_{\alpha_j}yy_j - S_{\theta_j}C_{\alpha_j}yz_j \\
yy_{(j-1)r} &= yy_{j-1} + SS_{\theta_j}CC_{\alpha_j}xx_j + 2CS_{\theta_j}CC_{\alpha_j}xy_j - 2S_{\theta_j}CS_{\alpha_j}xz_j \\
&\quad + CC_{\theta_j}CC_{\alpha_j}yy_j - 2C_{\theta_j}CS_{\alpha_j}yz_j + SS_{\alpha_j}zz_j \\
yz_{(j-1)r} &= yz_{j-1} + SS_{\theta_j}CS_{\alpha_j}xx_j + 2CS_{\theta_j}CS_{\alpha_j}xy_j + S_{\theta_j}(CC_{\alpha_j} - SS_{\alpha_j})xz_j \\
&\quad + CC_{\theta_j}CS_{\alpha_j}yy_j + C_{\theta_j}(CC_{\alpha_j} - SS_{\alpha_j})yz_j - CS_{\alpha_j}zz_j \\
zz_{(j-1)r} &= zz_{j-1} + SS_{\theta_j}SS_{\alpha_j}xx_j + 2CS_{\theta_j}SS_{\alpha_j}xy_j + 2S_{\theta_j}CS_{\alpha_j}xz_j \\
&\quad + CC_{\theta_j}SS_{\alpha_j}yy_j + 2C_{\theta_j}CS_{\alpha_j}yz_j - CC_{\alpha_j}zz_j.
\end{aligned}$$

- 3) If  $p_{r_j} = 1$  (prismatic joint) and  $\mathbf{z}_j$  is parallel to  $\mathbf{z}_{r_1}$  ( $r_1 < j < r_2$ ),  $mz_j$  is eliminated and  $mx_j$  and  $my_j$  are regrouped such as:

$$\begin{aligned}
mx_{(j-1)r} &= mx_{j-1} + C_{\theta_j}mx_j - S_{\theta_j}my_j \\
my_{(j-1)r} &= my_{j-1} + S_{\theta_j}C_{\alpha_j}mx_j + C_{\theta_j}C_{\alpha_j}my_j \\
mz_{(j-1)r} &= mz_{j-1} + S_{\theta_j}S_{\alpha_j}mx_j + C_{\theta_j}S_{\alpha_j}my_j \\
zz_{jr} &= zz_j + 2d_jC_{\theta_j}mx_j - 2d_jS_{\theta_j}my_j.
\end{aligned}$$

- 4) If  $p_{r_j} = 1$  (prismatic joint) and  $\mathbf{z}_j$  is not parallel to  $\mathbf{z}_{r_1}$  ( $r_1 < j < r_2$ ),  $mz_j$  is eliminated and  $mx_j$  and  $my_j$  are regrouped such as:

$$\begin{aligned}
mx_{(j-1)r} &= mx_{j-1} - ({}^jz_{r_1x}/{}^jz_{r_1z})mz_j \\
my_{(j-1)r} &= my_{j-1} - ({}^jz_{r_1y}/{}^jz_{r_1z})mz_j,
\end{aligned}$$

if  ${}^jz_{r_1z} \neq 0$  else

$$mx_{(j-1)r} = mx_{j-1} - ({}^jz_{r_1x}/{}^jz_{r_1y})my_j,$$

with  ${}^jz_{r_1} = [{}^jz_{r_1x} \quad {}^jz_{r_1y} \quad {}^jz_{r_1z}]^T$  is the unit vector along the axis referred to frame  $R_j$ .

- 5) If  $p_{r_j} = 0$  (revolute joint) and  $r_1 \leq j \leq r_2$ , the parameters  $xx_j$ ,  $xy_j$ ,  $xz_j$  and  $yz_j$  are eliminated.
- 5) If  $p_{r_j} = 0$  (revolute joint),  $r_1 \leq j \leq r_2$ ,  $\mathbf{z}_j$  is along  $\mathbf{z}_{r_1}$  and  $\mathbf{z}_{r_1}$  is parallel to  $\mathbf{z}_i$  and  $\mathbf{g}$ , for  $i < j$ , the parameters  $mx_j$  and  $my_j$  are eliminated.
- 6) If  $p_{r_j} = 1$  (prismatic joint) and  $j < r_1$ , the parameters  $mx_j$ ,  $my_j$  and  $mz_j$  are eliminated.

7) The rotor inertial parameter  $ia_j$  can be grouped in the following case:

- If  $j = r_1$ ;
- If  $j = r_2$  and is orthogonal to  $r_1$ ;
- If the axis of the first prismatic joint  $p_1$  is orthogonal to the gravity and  $p_1 = 1$  or its axis is aligned with the preceding revolute joints.

### Numerical method

The numerical method is based on the QR decomposition of  $\mathbf{X}_{st}$  obtained from the IDM containing all the standard parameters. The vectors of joint positions, velocities and accelerations can be randomly generated or can come from experimental data. The QR decomposition of  $\mathbf{X}_{st}$  gives:

$$\mathbf{Q}^T \mathbf{X}_{st} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(r-c) \times c} \end{bmatrix} \quad (\text{A.30})$$

where

- $\mathbf{Q}$  is an orthogonal matrix of size  $(r \times r)$ ;
- $\mathbf{R}$  is an upper triangular matrix of size  $(c \times c)$ .

The rank of  $\mathbf{X}_{st}$  is equal of the number of base parameters; i.e.  $rank(\mathbf{X}_{st}) = b$ . This rank is evaluated with the number of zero elements on the diagonal  $\mathbf{R}_{ii}$ . In practice, due to the numerical aspect, a tolerance threshold  $tol$  is defined such as  $|\mathbf{R}_{ii}| \leq tol$ . The elements, such as  $|\mathbf{R}_{ii}| > tol$ , provide the base inertial parameters. They can be regrouped in a vector  $\boldsymbol{\chi}_1$  and their corresponding columns in  $\mathbf{X}_{st}$  are regrouped in matrix  $\mathbf{X}_1$ . The other parameters and columns are respectively regrouped in  $\boldsymbol{\chi}_2$  and  $\mathbf{X}_2$  such as

$$\mathbf{X}_{st} \boldsymbol{\chi} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\chi}_1 \\ \boldsymbol{\chi}_2 \end{bmatrix}. \quad (\text{A.31})$$

Due to the linear dependences, it can be written  $\mathbf{X}_2 = \mathbf{X}_1 \mathbf{B}$  then

$$\mathbf{X}_{st} \boldsymbol{\chi} = \mathbf{X}_1 (\boldsymbol{\chi}_1 + \mathbf{B} \boldsymbol{\chi}_2) = \mathbf{X}_1 \boldsymbol{\theta}, \quad (\text{A.32})$$

with  $\boldsymbol{\theta} = \boldsymbol{\chi}_1 + \mathbf{B} \boldsymbol{\chi}_2$ . This matrix  $\mathbf{B}$  shows the linear relations between the columns of  $\mathbf{X}_{st}$ . To evaluate  $\mathbf{B}$ , we calculate the QR decomposition of  $\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix}$ :

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0}_{(r-b) \times b} & \mathbf{0}_{(r-b) \times (c-b)} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 \mathbf{R}_1 & \mathbf{Q}_1 \mathbf{R}_2 \end{bmatrix} \quad (\text{A.33})$$

where  $\mathbf{R}_1$  is a  $(r \times r)$  regular upper triangular matrix and  $\mathbf{R}_2$  is a  $(b \times (c - b))$

matrix. We get

$$\begin{aligned}\mathbf{X}_1 &= \mathbf{Q}_1 \mathbf{R}_1 & (\text{A.34}) \\ \mathbf{X}_2 &= \mathbf{Q}_1 \mathbf{R}_2 = \mathbf{X}_1 \mathbf{R}_1^{-1} \mathbf{R}_2\end{aligned}$$

and finally

$$\boldsymbol{\theta} = \boldsymbol{\chi}_1 + \mathbf{R}_1^{-1} \mathbf{R}_2 \boldsymbol{\chi}_2. \quad (\text{A.35})$$



# Stäubli TX40

---

This appendix summarizes the key features of the robot Stäubli TX40 that is a 6 DOF, rigid, serial arm.

## B.1 Model of the TX40

This section is dedicated to the model of the Stäubli TX40 robot; see Figure 1.1. Its nominal load is 1.7 kg for a weight of 27 kg. The first three links are driven by Stäubli reducers whereas link 4 is composed of a belt drive followed by a reducer. The two last links are kinematically coupled. The recorded signals are the currents and the positions of the motors as well as the robot's commands.

The kinematics of the robot is defined with MDH described in section 2.1.1. The geometric parameters are given by Table B.1. All the joints are revolute. This robot has the special feature of having a kinematic coupling between links 5 and 6. According to [Khalil *et al.* 2007] and [Gautier & Briot 2012a], this coupling introduced three other standard parameters: an inertia  $i_{am_6}$ , a viscous friction  $f_{vm_6}$  and a Coulomb friction  $f_{cm_6}$ , such as:

$$\begin{aligned}\tau_5^c &= i_{am_6}\ddot{q}_6 + f_{vm_6}\dot{q}_6 + f_{cm_6}\text{sign}(\dot{q}_5 + \dot{q}_6) \\ \tau_6^c &= i_{am_6}\ddot{q}_5 + f_{vm_6}\dot{q}_5 + f_{cm_6}\text{sign}(\dot{q}_5 + \dot{q}_6),\end{aligned}\tag{B.1}$$

where  $\tau_5^c$  and  $\tau_6^c$  are the additional torques due to the coupling applied on link 5 and 6 respectively.

The robot has 87 standard dynamic parameters including the  $14 \cdot 6$  usual standard parameters and the 3 coupling parameters. The IDM is computed with the software Symoro+ based on the Newton-Euler formulation described in section A.2. After reduction, there are 61 base parameters: 11 are eliminated and 15 are regrouped. Table B.2 summarizes the base inertial parameters and Table B.3 gives the regrouping formula.

## B.2 Reference values of the dynamic parameters

From the 61 robot's base parameters, only 28 are well identified with good relative standard deviations. These 28 parameters define a set of essential parameters which are enough to describe the dynamics. This set was validated with a F-statistic, as shown in [Janot *et al.* 2014b]. Along the thesis, we present only those parameters. Since the robot has been widely studied, the bandwidths are well-known. According

Table B.1: Geometric parameters of the Stäubli TX40

Link	$\alpha_j$	$d_j$	$\theta_j^*$	$r_j$
1	0	0	$q_1$	0
2	$-\pi/2$	0	$q_2 - \pi/2$	0
3	0	$d_3 (= 0.225 \text{ m})$	$q_3 + \pi/2$	$r_3 (= 0.035 \text{ m})$
4	$\pi/2$	0	$q_4$	$r_4 (= 0.225 \text{ m})$
5	$-\pi/2$	0	$q_5$	0
6	$\pi/2$	0	$q_6 + \pi$	0

\*:  $\theta_j$  designated here an angle and not the vector of base parameters

Table B.2: Base inertial parameters of the Stäubli TX40

Link	$xx_j$	$xy_j$	$xz_j$	$yy_j$	$yz_j$	$zz_j$	$mx_j$	$my_j$	$mz_j$	$m_j$	$ia_j$
1						$zz_{1r}$					
2	$xx_{2r}$	$xy_2$	$xz_{2r}$		$yz_2$	$zz_{2r}$	$mx_{2r}$				
3	$xx_{3r}$	$xy_3$	$xz_3$		$yz_3$	$zz_{3r}$	$mx_3$	$my_{3r}$			$ia_3$
4	$xx_{4r}$	$xy_4$	$xz_4$		$yz_4$	$zz_{4r}$	$mx_4$	$my_{4r}$			$ia_4$
5	$xx_{5r}$	$xy_5$	$xz_5$		$yz_5$	$zz_{5r}$	$mx_5$	$my_{5r}$			$ia_5$
6	$xx_{6r}$	$xy_6$	$xz_6$		$yz_6$	$zz_6$	$mx_6$	$my_6$			$ia_6$

Table B.3: Regrouped parameters of the Stäubli TX40

$zz_{1r}$	=	$zz_1 + ia_1 + yy_2 + yy_3 + 2 \cdot r_3 \cdot mz_3 + (r_3^2 + d_3^2)(m_3 + m_4 + m_5 + m_6)$
$xx_{2r}$	=	$xx_2 - yy_2 - d_3^2(m_3 + m_4 + m_5 + m_6)$
$xz_{2r}$	=	$xz_2 - d_3 \cdot mz_3 - r_3 \cdot d_3 \cdot (m_3 + m_4 + m_5 + m_6)$
$zz_{2r}$	=	$zz_2 + ia_2 + d_3^2 \cdot (m_3 + m_4 + m_5 + m_6)$
$mx_{2r}$	=	$mx_2 + d_3 \cdot (m_3 + m_4 + m_5 + m_6)$
$xx_{3r}$	=	$xx_3 - yy_3 + yy_4 + 2 \cdot d_3 \cdot mz_4 + r_4^2 \cdot (m_4 + m_5 + m_6)$
$zz_{3r}$	=	$zz_3 + yy_4 + 2 \cdot d_3 \cdot mz_4 + r_4^2 \cdot (m_4 + m_5 + m_6)$
$my_{3r}$	=	$my_3 - mz_4 - r_4 \cdot (m_4 + m_5 + m_6)$
$xx_{4r}$	=	$xx_4 - yy_4 + yy_5$
$zz_{4r}$	=	$zz_4 + yy_5$
$my_{4r}$	=	$my_4 + mz_5$
$xx_{5r}$	=	$xx_5 - yy_5 + yy_6$
$zz_{5r}$	=	$zz_5 + yy_6$
$my_{5r}$	=	$my_5 - mz_6$
$xx_{6r}$	=	$xx_6 - yy_6$



Table B.4: TX40 - Estimated parameters and relative standard deviations - IDIM-LS

Param.	IDIM-LS	Param.	IDIM-LS
$zz_{1r}$	1.24 (1.45 %)	$fv_3$	1.93 (2.05 %)
$fc_1$	7.34 (2.66 %)	$mx_4$	-0.02 (35.1 %)
$fv_1$	8.00 (0.91 %)	$ia_4$	0.03 (9.10 %)
$xx_{2r}$	-0.48 (3.28 %)	$fc_4$	2.57 (2.46 %)
$xz_{2r}$	-0.16 (5.40 %)	$fv_4$	1.09 (1.62 %)
$zz_{2r}$	1.09 (1.29 %)	$my_{5r}$	-0.03 (15.2 %)
$mx_{2r}$	2.21 (3.01 %)	$ia_5$	0.04 (10.6 %)
$fc_2$	8.24 (2.26 %)	$fc_5$	3.07 (3.59 %)
$fv_2$	5.50 (1.45 %)	$fv_5$	1.79 (2.33 %)
$xx_{3r}$	0.13 (10.6 %)	$ia_6$	0.01 (13.3 %)
$zz_{3r}$	0.11 (9.57 %)	$fc_6$	0.30 (32.9 %)
$my_{3r}$	-0.60 (2.52 %)	$fv_6$	0.65 (1.85 %)
$ia_3$	0.09 (9.39 %)	$fc_{m6}$	1.90 (4.40 %)
$fc_3$	6.48 (2.06 %)	$fv_{m6}$	0.61 (1.97 %)

to [Gautier *et al.* 2013a], the maximum bandwidth is  $\omega_{dyn} = 10$  Hz for joint 6. In practice, the joint positions and control signals are stored with a measurement frequency  $f_m = 5$  kHz. For the IDIM-LS method, the filters cut-off frequencies are tuned according to the rules provided in section 2.3.1:  $\omega_{fq} = 5\omega_{dyn} = 50$  Hz and  $\omega_{Fp} = 2\omega_{dyn} = 20$  Hz respectively for the Butterworth and the decimate filters.

In practice, the Coulomb friction is modelled with the arctangent function instead of the sign one in order to have a continuous model. This way of doing is common in mechanical engineering [Gautier *et al.* 2013b]. In fact, the function  $\frac{2}{\pi}atan(\gamma\dot{q}_j(t))$  tends to  $sign(\dot{q}_j(t))$  when  $\gamma$  tends to infinity. For our purpose,  $\gamma = 150$  appeared to be a satisfactory choice. With a too large value, the numerical integration of the differential equations would be stiff.

Concerning the control law, the robot was bought "off the shelf" with a propriety controller. The exact knowledge of this controller is thus unavailable. Consequently, a dedicated controller was designed to perform tests in the laboratory. This controller is composed of one PID per axis. We refer to it as the actual controller.

Table B.4 gives the estimated parameters that can be seen as reference values for the other methods.

### B.3 Control law

The original control law of the robot designed by Stäubli is obviously not available. For our research purpose, a dedicated controller has been designed. That controller is composed of one nested PD per link<sup>1</sup> in accordance with the material provided in [Khalil & Dombre 2004] and [Siciliano *et al.* 2010]. Figure B.1 depicts the general

<sup>1</sup>For access to the detailed controller, please contact the author

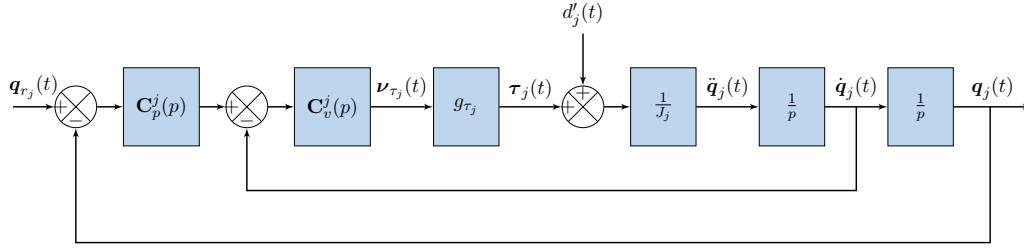


Figure B.1: Block diagram of general independent joint control

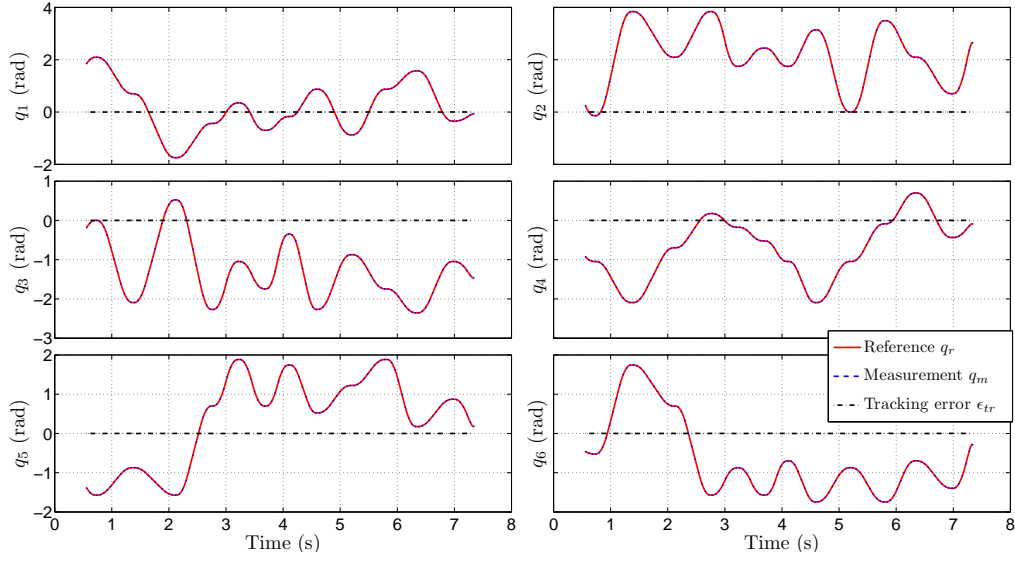


Figure B.2: TX40 - Excitation trajectories example

scheme of the control law for each link, where  $C_p^j(p)$  and  $C_v^j(p)$  are respectively the position and velocity controllers. The inertia  $J_j$  was defined in (6.2) and the disturbance  $d'_j(t) = d_j(t) - F_{v_j} \dot{q}_{nf_j}(t)$  with  $d_j$  defined in (6.3).

## B.4 Illustration of experimental signals

The reference trajectories are trapezoidal velocities (also called smoothed bang-bang accelerations). Since  $\text{cond}(\phi_{F_p}) = 200$ , the reference trajectories excite well the base parameters. Figure B.2 depicts the joint trajectories as well as the corresponding tracking error. Figure B.3 provides the identified torques by the IDIM-LS method with the parameters estimated in the previous section.

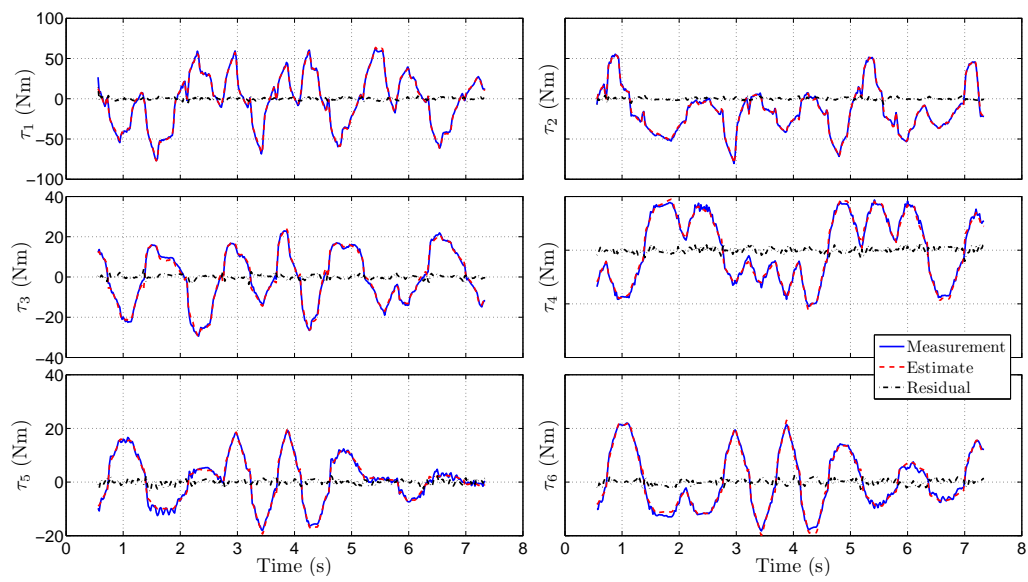


Figure B.3: TX40 - IDIM-LS identification example



# Mathematical background

---

This appendix gives some mathematical elements useful for the reading of this thesis. The first three sections are devoted to the fields of probability and statistics whereas the fourth one concerns few optimisation schemes.

## C.1 Random Variables

In this section, several definitions of probabilistic concepts used in the thesis are recalled. Let us start with a random variable  $X$  that can take values randomly a probability  $\Pr(X \leq x) = \Pr(x)$ , with  $x$  a selected value. That probability  $\Pr(x)$  gives a scalar values on the interval  $[0, 1]$  expressing the probability that  $X$  takes a value less than or equal to  $x$ . In the following development, the random variable is assumed to be continuous although there exists a discrete counterpart.

The probability density function of the random variable  $X$  is defined as

$$p(x) = \frac{d\Pr(x)}{dx}. \quad (\text{C.1})$$

The expected value, or mean, is defined as

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} xp(x)dx. \quad (\text{C.2})$$

If an the same experiment is conducted  $N$  times, the expected value can be approximated by

$$\mathbb{E}(X) \approx \frac{1}{N} \sum_{i=1}^N x_i, \quad (\text{C.3})$$

where the  $x_i$  are the sample values of the random variable  $X$ .

Considering a random variable  $X$  with expected value  $\mathbb{E}(X) = m_X$ , its variance is defined by

$$\begin{aligned} \sigma_X^2 &= \mathbb{E}[(X - m_X)^2] = \int_{-\infty}^{\infty} (x - m_X)^2 p(x) dx \\ &= \mathbb{E}(X^2) - m_X^2 = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2. \end{aligned} \quad (\text{C.4})$$

Two random variables,  $X$  and  $Y$ , are uncorrelated if

$$\mathbb{E}(XY) = \mathbb{E}(X)\mathbb{E}(Y) \quad (\text{C.5})$$

and they are orthogonal if

$$E(XY) = 0 \quad (\text{C.6})$$

A random vector is a collection of  $n$  random variables  $X_1, X_2, \dots, X_n$  such as  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$ . Similarly to the scalar case, the mean value and covariance matrix of  $\mathbf{X}$  are defined by

$$E(\mathbf{X}) = \mathbf{m}_X \quad (\text{C.7})$$

$$\text{cov}(\mathbf{X}) = E[(\mathbf{X} - \mathbf{m}_X)(\mathbf{X} - \mathbf{m}_X)^T] = \Sigma_X \quad (\text{C.8})$$

## C.2 Properties of Estimators

With a model defined by a vector of parameters  $\boldsymbol{\theta}$ , we obtain an estimate  $\hat{\boldsymbol{\theta}}$  based on  $N$  observations  $y_1, \dots, y_N$ . The algorithm processing the observations is called the *estimator*. An estimator is said to be *unbiased* if  $B(\hat{\boldsymbol{\theta}}) = E[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta} = 0$ , for all  $\boldsymbol{\theta}$ .

An estimate can be said to be *asymptotically unbiased* if  $\hat{\boldsymbol{\theta}}(N)$  based on  $N$  samples is unbiased for  $N \rightarrow \infty$ .

The *mean square error* (MSE) of  $\hat{\boldsymbol{\theta}}$  is given by

$$MSE(\hat{\boldsymbol{\theta}}) = E[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2]. \quad (\text{C.9})$$

The *variance* of  $\hat{\boldsymbol{\theta}}$  is defined by

$$\text{var}(\hat{\boldsymbol{\theta}}) = E[(\hat{\boldsymbol{\theta}} - E(\hat{\boldsymbol{\theta}}))^2]. \quad (\text{C.10})$$

The *standard deviation* is the square root of the variance. The MSE, bias and variance are linked by  $MSE(\hat{\boldsymbol{\theta}}) = \text{var}(\hat{\boldsymbol{\theta}}) + B(\hat{\boldsymbol{\theta}})^2$ .

An estimator is said to be *consistent* if

$$\text{p} \lim_{k \rightarrow \infty} \hat{\boldsymbol{\theta}}(k) = \boldsymbol{\theta}, \quad (\text{C.11})$$

where p lim is the *probability in the limit* defined such as  $\lim_{k \rightarrow \infty} \Pr(|\hat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta}| \geq \varepsilon) = 0$  for any  $\varepsilon > 0$ .

The *weak law of large numbers* states that that the sample average converges in probability towards the expected value; i.e.  $\text{p} \lim_{k \rightarrow \infty} \bar{\boldsymbol{\theta}}(k) = \boldsymbol{\mu}$  with  $\bar{\boldsymbol{\theta}}(k) = \frac{1}{k} \sum_{i=1}^k \boldsymbol{\theta}_i$  and  $\boldsymbol{\mu} = E[\boldsymbol{\theta}_1] = \dots = E[\boldsymbol{\theta}_k]$ .

Two features are interesting for an estimator: to be unbiased and to have a minimal MSE. These dual objectives cannot be necessary satisfied at the same time. In some cases, it can be defined an unbiased *efficient* estimator which has the lowest variance among the unbiased estimators and satisfies the Cramér-Rao bound. That bound is the lower bound on the variance of any unbiased estimator. All the material provided in this section can be found in [Kay 1993] for instance.

### C.3 Random Process

A random process, also called stochastic process, is a set random variables  $X(t)$  indexed by a parameter  $t$  from a set  $T$ . In our study, the index parameter is the time. If the set  $T$  is continuous, respectively discrete, we talk about CT random process, respectively DT. A random process  $X(t)$  is said to be *strictly stationary* if for each  $(t_1, t_2, \dots, t_N)$  and any  $\tau$ :

$$\Pr[X(t_1 + \tau), \dots, X(t_N + \tau)] = \Pr[X(t_1), \dots, X(t_N)]. \quad (\text{C.12})$$

A random process is said to be *wide-sense stationary* if  $m_x$  and  $\sigma_x^2$  are not function of the time. Using the definition of the expected value and the covariance, it can be shown that the strict stationary condition implies the wide-sense stationary condition.

A stationary random process  $X(t)$  has an *autocorrelation* function defined such as

$$R_{XX}(\tau) = \mathbb{E}[X(t)X(t + \tau)] \quad (\text{C.13})$$

where  $\tau$  is a time shift. Then, we can define the *Power Spectral Density* (PSD) function as the Fourier transform of this autocorrelation

$$P_{XX}(\omega) = \int_{-\infty}^{\infty} R_{XX}(\tau)e^{-j\omega\tau} d\tau, \quad (\text{C.14})$$

where  $j^2 = -1$  and  $\omega$  is a frequency in rad/s over the range  $(-\infty, \infty)$ . Reciprocally

$$R_{XX}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_{XX}(\omega)e^{j\omega\tau} d\omega. \quad (\text{C.15})$$

A white noise is wide-sense stationary process with

$$\begin{aligned} m_X(t) &= 0 & \forall t \\ \text{cov}[X(t_1), X(t_2)] &= \begin{cases} \sigma_X^2 & t_1 = t_2 \\ 0 & t_1 \neq t_2 \end{cases} \end{aligned}$$

With a constant PSD,  $P_e$ , a white noise has a covariance given by

$$\begin{aligned} \sigma_X^2 &= R_{XX}(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_e d\omega & (\text{C.16}) \\ &= \frac{1}{2\pi} P_e \int_{-\omega_{nyq}}^{\omega_{nyq}} d\omega = \frac{1}{2\pi} P_e \cdot 2 \cdot \omega_{nyq} = \frac{P_e}{T_s} \end{aligned}$$

where  $\omega_{nyq} = \omega_s/2 = \frac{\pi}{T_s}$  is the Nyquist frequency. Finally, for a SISO model with the input  $X$  and the output  $Y$  defined such as

$$Y(t) = \int_0^{\infty} h(\tau)X(t - \tau)d\tau, \quad (\text{C.17})$$

the spectral relation is given by

$$P_{YY}(\omega) = |H(\omega)|^2 P_{XX}(\omega). \quad (\text{C.18})$$

The interested reader can refer to [Bendat & Piersol 2010] for further information on this topic.



## Part VI

# Résumé de thèse en français



# Identification de robots industriels rigides – Apport des méthodes de l’identification de systèmes

---

L’identification paramétrique de robots industriels est problème complexe mais néanmoins nécessaire pour la synthèse de lois de commandes. Cela permet à terme de mettre au point des lois commandes rapides et précises satisfaisant les besoins des divers utilisateurs industriels. Dans cette thèse, nous explorons et enrichissons les différentes méthodes d’identification dédiées aux robots industriels.

Ce chapitre reprend de manière fidèle l’organisation du manuscrit écrit en anglais à l’exception des chapitres de prérequis contenus dans la partie II. La problématique de la thèse est introduite dans la section D.1, accompagnée d’un état de l’art sommaire. L’étude de la robustesse des méthodes d’identification est présentée dans la section D.2. La section D.3 est dédiée à l’identification de robots sans la connaissance du contrôleur ni des bandes-passantes. La modélisation du bruit est considérée dans la section D.4. Les conclusions de l’étude sont exposées dans la section D.5.

---

<b>D.1 Introduction</b>	<b>157</b>
D.1.1 Contexte	157
D.1.2 État de l’art	158
D.1.3 Problématique	159
<b>D.2 Evaluation de la robustesse des méthodes d’identification basées sur la simulation du modèle auxiliaire</b>	<b>160</b>
D.2.1 Choix du signal d’identification	160
D.2.2 Comparaison des méthodes IDIM-IV et DIDIM	163
D.2.3 Remarques et conclusion	165
<b>D.3 Identification de systèmes robotiques sans connaissance du contrôleur et des bande-passantes</b>	<b>165</b>
D.3.1 Estimation des dérivées articulaires	166
D.3.2 Identification du contrôleur	170
D.3.3 Conclusions	173

<b>D.4 Modélisation du bruit pour l'identification de systèmes robotiques</b> . . . . .	<b>174</b>
D.4.1 Modélisation et identification du bruit . . . . .	174
D.4.2 Extension des méthodes d'identification . . . . .	177
D.4.3 Conclusions . . . . .	181
<b>D.5 Conclusions et perspectives</b> . . . . .	<b>181</b>
D.5.1 Résumé des contributions . . . . .	181
D.5.2 Un guide pour l'utilisateur . . . . .	183
D.5.3 Développements futurs . . . . .	184

---

## D.1 Introduction

### D.1.1 Contexte

Les robots industriels sont des bras mécaniques composés de plusieurs corps reliés par des articulations, également appelés axes. Chaque articulation est actionnée par un moteur électrique couplé à un capteur de position. Un ordinateur contrôle les mouvements du bras afin de placer la charge utile dans l'espace de travail. La figure 1.1 illustre un tel robot. Les robots industriels sont des composants essentiels de l'industrie moderne car ils permettent une réduction des coûts, l'augmentation des cadences, l'élimination des tâches dangereuses pour les travailleurs humains, etc. C'est pourquoi, pendant les dernières décennies, le nombre de robots industriels a fortement augmenté ; voir la figure 1.2.

Les robots industriels doivent effectuer des tâches répétitives rapidement et de manière précise. Afin d'accomplir de telles tâches, différentes caractéristiques doivent être satisfaites suivant l'application. La norme ISO [ISO 1999] spécifie les caractéristiques utiles pour l'utilisateur que doit fournir le constructeur du robot. Par exemple, il y a la *répétabilité* définissant la précision avec laquelle le robot retourne à une position commandée. D'après [Siciliano *et al.* 2010], pour un bras robotique avec un élan maximum de 1.5 m, la répétabilité varie entre 0.02 and 0.2 mm. Afin de satisfaire de tels spécifications, les lois de commande ont besoin de modèles fiables et complets de ces robots. Cette modélisation précise vient d'une identification rigoureuse basée sur des données expérimentales. Ainsi, le problème est pour le praticien de fournir un modèle aussi précis et complet que possible.

Plusieurs difficultés sont envisagées. Tout d'abord, les robots opèrent en boucle fermée, i.e. avec une rétroaction, pour des raisons de stabilité et de précision. Il est souvent difficile, voire impossible pour des raisons de sécurité, de mettre en œuvre une expérience sans cette rétroaction. La méthode d'identification doit donc pouvoir traiter cette problématique de la boucle fermée. En quelques mots, la rétroaction crée une corrélation entre le bruit de sortie et l'entrée qui mène à des difficultés statistiques pour l'identification. De plus, si le contrôleur est réglé correctement, le système sera insensible à n'importe quelle perturbation et spécialement aux variations des paramètres. Or, les méthodes d'identification sont basées sur la sensibilité du système aux variations des paramètres. Ensuite, suivant la technologie du robot considéré, différentes dynamiques peuvent être rencontrées : frottement linéaire ou non-linéaire, modes rigides ou flexibles. Dans le cas de robots à usages industriels, une dynamique rigide est privilégiée dès la conception afin d'assurer la précision de la tâche, bien que des structures flexibles peuvent être envisagées afin de réduire les coûts ; voir par exemple [Wernholt 2007]. Enfin, suite aux difficultés énoncées précédemment, des processus spécifiques ont été développés par les roboticiens pour l'identification. Ces processus sont basés sur les compétences et le savoir-faire du roboticien ainsi que sur quelques règles empiriques. Dans la partie suivante, nous introduisons les éléments principaux de ces méthodes.

### D.1.2 État de l'art

Les modèles dynamiques de bras industriels sont issus des lois de la mécanique générale : formalisme de Newton ou de Lagrange. Afin d'exprimer les équations du mouvement, une notation doit tout d'abord être définie afin de paramétrer le problème. Dans [Denavit & Hartenberg 1955], les auteurs ont introduit une notation pour les robots industriels, qui a été améliorée ensuite dans [Khalil & Kleinfinger 1986]. Cette notation définit les relations géométriques entre les éléments constitutifs du robot, en introduisant un nombre minimal de paramètres. Cela a permis d'exprimer les modèles dynamiques de manière générique et normalisée. Si ces travaux permettent de modéliser fidèlement les forces inertielles, ils n'ont pas abordé spécifiquement la question des forces de frottements. D'autres travaux ont ainsi proposé différents modèles de frottement, comme [Canudas de Wit *et al.* 1989, Bona & Indri 2005] par exemple. En combinant la notation standard pour les forces inertielles et la modélisation des frottements, un modèle complet des bras robotiques est ainsi à notre disposition.

À partir de ces modélisations, deux techniques de réduction de modèle ont été proposées dans [Gautier & Khalil 1990] et [Gautier 1991]. L'une est basée sur l'analyse des relations algébriques alors que la deuxième repose sur une évaluation numérique du modèle. Cette réduction permet d'une part d'extraire l'ensemble des paramètres de base, dont les effets peuvent être identifiés de manière distincte. D'autre part, cette réduction conserve la linéarité du couple par rapport aux paramètres issue de la modélisation de Denavit-Hartenberg. Ainsi les paramètres de base peuvent être estimés à l'aide de la technique des moindres carrés, à condition de traiter la question des bruits de mesure corrélés à cause de la rétroaction. À cette fin, dans [Gautier 1997], l'auteur a mis en place une procédure de pré-filtrage des données expérimentales en deux étapes. La première étape consiste à filtrer la position articulaire mesurée pour en déduire la vitesse et l'accélération via un filtre passe-bande. La deuxième étape consiste à filtrer le couple mesuré afin de rejeter les perturbations haute-fréquences via un filtre decimate, encore appelé filtrage parallèle. Ce processus de pré-filtrage rend les moindres carrés (LS) viables statistiquement. L'ensemble pré-filtrage et estimation compose la méthode IDIM-LS. L'inconvénient de cette méthode est qu'elle est sensible au réglage des filtres qui reposent sur une connaissance *a priori* des bandes passantes du système.

D'autres approches d'identification ont été développées au cours des années, sans réellement améliorer la méthode IDIM-LS si elle est couplée avec des filtres correctement réglés. Il peut être cité par exemple : la méthode des moindres carrés totaux (TLS) [Gautier *et al.* 1994, Xi 1995], la méthode basée sur les outils d'inégalité matricielle linéaire (LMI) [Calafiore & Indri 2000], le filtre de Kalman étendu (EKF) [Kostic *et al.* 2004] ou une approche basée sur le maximum de vraisemblance (ML) [Olsen & Petersen 2001, Olsen *et al.* 2002].

Dans [Janot *et al.* 2014c], les auteurs ont introduit une approche de la variable instrumentale (IV), provenant de l'économétrie et adaptée à l'identification de robots. L'idée est de simuler le modèle en boucle-fermée afin d'obtenir des signaux

non-corrélés avec le bruit de mesure. Cette méthode, nommée IDIM-IV, utilise le même processus de pré-filtrage que la méthode IDIM-LS bien que les estimées finales soient moins sensibles au réglage des filtres. En plus des bandes passantes du système, le modèle du contrôleur est aussi requis pour la simulation. Pour simplifier le processus de pré-filtrage, dans [Gautier *et al.* 2013a], les auteurs ont développé une méthode d'identification basée sur les modèles dynamiques direct et indirect (DIDIM) qui repose aussi sur la simulation du modèle en boucle-fermée. Cette méthode a l'avantage de ne pas avoir besoin de la première étape du processus de pré-filtrage car elle est basée uniquement sur la mesure du couple. Cependant, la deuxième étape (filtre decimate) reste nécessaire tout comme la connaissance *a priori* (contrôleur et bandes passantes).

Les méthodes d'identification introduites ci-avant partagent le fait de ne pas identifier le modèle du bruit. Au sein de la communauté d'identification des systèmes, Ljung a introduit la méthode d'erreur de prédiction (PEM) qui s'applique principalement à des modèles linéaire temps invariant (LTI) en boucle ouverte ou fermée; cf [Ljung 1976]. Les modèles considérés sont à temps discret et incluent le modèle de bruit. Malheureusement, cette méthode ne peut pas être appliquée aux robots car leurs modèles sont non-linéaires et à temps continu. Quasi simultanément, Young and Jakeman ont mis au point la variable instrumentale raffinée pour les systèmes à temps continu (RIVC) dans [Young & Jakeman 1980]. Avec une approche différente, cette méthode s'applique à l'identification de modèles LTI en boucle ouverte ou fermée, mais dans un cadre temps continu. À cause des non-linéarités, la méthode RIVC ne peut pas être appliqué non plus aux robots industriels d'une manière directe. Pour les deux méthodes, identifier le bruit permet un filtrage automatique des données et fournit des paramètres estimés optimaux. Il doit être noté qu'un cas particulier de la PEM, appelé méthode d'erreur de sortie (OEM), peut traiter l'identification de systèmes non-linéaires et à temps continu; voir par exemple [Richalet *et al.* 1971] pour un contexte général ou [Landau *et al.* 1999] pour les systèmes en boucle-fermée. Néanmoins, la méthode à erreur de sortie ne prend pas en compte le modèle de bruit.

### D.1.3 Problématique

Il apparaît ainsi que la communauté robotique dispose de méthodes capables d'identifier des robots industriels opérant en boucle fermée. Cette problématique de boucle fermée a été traitée de différentes manières suivant la méthode d'identification considérée. En parallèle, la communauté d'automatique a mis au point des méthodes permettant d'identifier des systèmes fonctionnant en boucle fermée, de manière précise et systématique. C'est-à-dire qu'elles prennent en compte le modèle de bruit sans connaissance *a priori*. Cependant, ces méthodes ne s'appliquent que sur des systèmes LTI, ce que ne sont pas les robots industriels. De leur côté, les roboticiens n'ont pas traité la question de la modélisation du bruit de manière approfondie. Leurs méthodes reposent en effet sur un pré-filtrage empirique basée sur une connaissance physique du système *a priori*. Si le niveau de bruit et sa localisation

fréquentielle sont une préoccupation majeure, les roboticiens n'ont pas essayé de le modéliser.

C'est dans ce contexte que cette thèse a été réalisée. Elle consiste à mettre en place des méthodes d'identification de systèmes robotiques industriels de manière précise et automatique, tout en minimisant la connaissance *a priori* nécessaire, afin de faire un lien entre la communauté d'identification de systèmes et celle de robotique.

## D.2 Evaluation de la robustesse des méthodes d'identification basées sur la simulation du modèle auxiliaire

### D.2.1 Choix du signal d'identification

#### Méthodes à erreur de sortie

En considérant la méthode d'erreur de sortie pour un robot, il semble naturel de prendre la position articulaire comme signal pour l'identification. Ainsi, le vecteur d'erreur de sortie est défini par :

$$\boldsymbol{\varepsilon}_{CLOE}(t, \boldsymbol{\theta}) = \mathbf{q}_m(t) - \mathbf{q}_s(t, \boldsymbol{\theta}), \quad (\text{D.1})$$

où  $\mathbf{q}_m$  et  $\mathbf{q}_s$  sont les vecteurs ( $n \times 1$ ) de position articulaires mesurées et simulées respectivement. Puisque les robots sont instables en boucle-ouverte, ils sont identifiés en boucle-fermée. La méthode est ainsi nommée *Closed-Loop Output Error* (CLOE) pour erreur de sortie en boucle-fermée.

En identification de robot, il est cependant plus usuel de considérer le signal d'entrée (couple ou force) pour l'identification car il a l'avantage d'être linéaire vis-à-vis des paramètres. Une variante de la méthode CLOE peut donc être envisagée avec le signal d'entrée. Cette technique est ainsi nommée *Closed-Loop Input Error* (CLIE) pour erreur d'entrée en boucle-fermée. Le vecteur d'erreur d'entrée est défini par :

$$\boldsymbol{\varepsilon}_{CLIE}(t, \boldsymbol{\theta}) = \boldsymbol{\tau}(t) - \boldsymbol{\tau}_s(t, \boldsymbol{\theta}), \quad (\text{D.2})$$

où  $\boldsymbol{\tau}$  et  $\boldsymbol{\tau}_s$  sont les vecteurs ( $n \times 1$ ) de couples/forces mesurées et simulées respectivement.

Les deux méthodes se basent sur une optimisation non-linéaire afin de régler itérativement les paramètres. Cette optimisation est généralement réalisée à l'aide d'un algorithme de Gauss-Newton comme il est indiqué avec (4.2). L'information clé pour l'optimiseur est la sensibilité, qui est la dérivée relative de la sortie considérée par rapport aux paramètres. Généralement ces sensibilités ne sont pas connues exactement mais approximées par des différences finies.

Un élément théorique de la méthode de l'erreur de sortie est à noter. Le bruit



additif sur le signal considéré doit être blanc, sinon les hypothèses statistiques pour l'estimation des variances des paramètres ne sont pas vérifiées. Comme nous le verrons par la suite, cela représente une limitation à l'application d'une telle méthode aux robots industriels. Le principe général des méthodes CLOE et CLIE est illustré par la Figure 4.1.

### Analyse de la sensibilité

Pour la méthode CLIE, à l'itération  $it$ , la sensibilité en entrée est donnée par

$$\Delta_{\tau_s}(t) = \left. \frac{\partial \tau_s(t)}{\partial \theta} \right|_{\theta = \hat{\theta}_{CLIE}^{it}}. \quad (D.3)$$

En utilisant la définition de contrôleur (2.11), il vient :

$$\Delta_{\tau_s}(t) = \left. \frac{\partial \mathbf{G}_\tau \mathbf{C}(p) (\mathbf{q}_r(t) - \mathbf{q}_s(t, \hat{\theta}_{CLIE}^{it}))}{\partial \theta} \right|_{\theta = \hat{\theta}_{CLIE}^{it}}, \quad (D.4)$$

avec  $\mathbf{G}_\tau$ , la matrice ( $n \times n$ ) de gain des actionneurs ;  $\mathbf{C}(p)$ , la matrice ( $n \times n$ ) de transfert du contrôleur ; et  $\mathbf{q}_r$  est le vecteur ( $n \times 1$ ) de position de référence. Et en supposant le contrôleur connu :

$$\Delta_{\tau_s}(t) = -\mathbf{G}_\tau \mathbf{C}(p) \Delta_{\mathbf{q}_s}(t). \quad (D.5)$$

Puisque les robots sont contrôlés en position, la relation suivante est attendue :

$$\mathbf{q}_m(t) \approx \mathbf{q}_r(t). \quad (D.6)$$

En supposant que l'optimiseur est bien initialisé, on obtient :

$$\mathbf{q}_s(t) \approx \mathbf{q}_r(t). \quad (D.7)$$

À partir de (D.6) et (D.7), il peut être supposé que le contrôleur opère dans les basses fréquences et qu'il peut être approximé par une ( $n \times n$ ) matrice  $\mathbf{C}_0$ . Dans le cas d'une loi Proportionnelle-Dérivée, ce résultat est direct. Si une action intégrale est présente, comme expliqué dans la section 2.2.1, le terme intégral est désactivé lorsque l'erreur de position est faible, afin d'éviter des oscillations dues aux frottement de Coulomb. Ainsi, nous avons :

$$\Delta_{\tau_s}(t) = -\mathbf{G}_\tau \mathbf{C}_0 \Delta_{\mathbf{q}_s}(t). \quad (D.8)$$

En considérant (D.8), il peut être montré que la méthode CLIE est une version pondérée de la méthode CLOE ; voir (4.11). Par conséquent, d'après la théorie des statistiques [Davidson & MacKinnon 1993], nous avons

$$\mathbb{E}[\hat{\theta}_{CLIE}^{it}] \rightarrow \mathbb{E}[\hat{\theta}_{CLOE}^{it}].$$

Il apparaît ainsi que les méthodes CLIE et CLOE fournissent asymptotiquement les mêmes paramètres estimés. La question qui en suit est : existe-t-il un estimateur plus efficace que l'autre ?

Dans le cadre de l'erreur de sortie, le bruit additif sur la position est supposé blanc. Le signal peut ainsi être écrit :

$$\mathbf{q}_m(t) = \mathbf{q}_{nf}(t) + \mathbf{n}_q(t), \quad (\text{D.9})$$

où  $\mathbf{n}_q$  est un vecteur ( $n \times 1$ ) de bruits blancs indépendants. En se basant sur l'équation du contrôleur, il peut être montré que le bruit vu par le couple d'entrée est donné par :

$$\mathbf{v}(t) = -\mathbf{G}_\tau \mathbf{C}_0 \mathbf{n}_q(t). \quad (\text{D.10})$$

De cette relation, un lien entre les covariances des bruits d'entrée et de sortie peut être établi. Par la suite, les covariances des paramètres estimés apparaissent égales pour les deux méthodes ; voir le développement (4.16). Les méthodes CLIE et CLOE fournissent donc les mêmes estimations avec les mêmes variances. La question en suspens est donc de savoir quelle méthode est la plus intéressante pour le praticien.

Pour répondre à cette interrogation, deux cas doivent être considérés pour la méthode CLOE.

- Le premier cas est lorsque les relations (D.6) et (D.7) sont satisfaites. la loi de commande assure alors un excellent tracking. Dans ce cas, nous avons  $\varepsilon_{CLOE}(t, \boldsymbol{\theta}) \approx 0$ , pour tout  $t$ , et l'innovation est nulle d'après (4.17). Ainsi, la méthode CLOE peut être totalement insensible aux erreurs de modèle et/ou bruit de mesure si la loi de commande est assez performante. En pratique, l'optimiseur ne varierait pas du point d'initialisation.
- Le deuxième cas est lorsque (D.6) et (D.7) ne sont pas totalement satisfaites, suite à une loi de commande pas assez performante. D'après (4.21), il se peut alors que l'estimation ne soit pas assez robuste face à une erreur de modélisation à cause des faibles valeurs propres de la matrice jacobienne regroupant les sensibilités.

A contrario, la méthode CLIE présente des valeurs propres plus grandes que la méthode CLOE grâce à la pondération du contrôleur (D.8). Par conséquent, la méthode CLIE est plus robuste face à de faibles erreurs de modèle et elle est plus sensible aux variations des paramètres.

### Résultats expérimentaux

Pour illustrer l'intérêt de considérer le signal d'entrée, trois méthodes d'identification sont considérées : les méthodes CLOE, CLIE et DIDIM. Il peut être montré que la méthode DIDIM est une méthode CLIE avec deux particularités. Premièrement, le couple est supposé linéaire par rapport aux paramètres, ce qui simplifie grandement le calcul des sensibilités. Cette hypothèse est appelée *Pseudo-Linear*

*Regression* (PLR), pour régression pseudo-linéaire. Deuxièmement, le filtrage decimate est inclus dans la méthode.

Dans un premier temps, seules les méthodes CLOE et CLIE sont testées dans un cadre d'erreur de sortie strict. Autrement dit, le bruit additif est blanc et aucune opération de filtrage n'est effectuée. Les résultats, récapitulés dans les Tableaux 4.2 et 4.3, montrent une meilleure robustesse de la méthode CLIE. Néanmoins, il apparaît que les résidus de l'estimation sont auto-corrélés. Par conséquent, les variances des paramètres estimés sont biaisées.

Dans un deuxième temps, un filtre decimate est inclus dans le processus d'identification afin d'obtenir des résidus dé-corrélés. Les paramètres estimés par les méthodes CLOE et CLIE sont alors totalement comparables à ceux estimés sans le filtre decimate. Il y a toujours une meilleure robustesse pour la méthode CLIE. La méthode DIDIM, quant à elle, fournit des résultats parfaitement équivalents à ceux de la méthode CLIE, mais avec un temps de calcul beaucoup plus court comme indiqué dans le tableau 4.5. Pour les trois méthodes, le filtre decimate permet bien une dé-corrélation des résidus et une bonne estimation des variances.

Les résultats expérimentaux ont donc montré qu'il est plus intéressant de considérer l'entrée (couple ou force) pour l'identification, mais que le filtre decimate est nécessaire à une bonne estimation des variances. Si l'hypothèse PLR est valide alors la méthode DIDIM doit être préférée afin de gagner en temps de calcul.

## D.2.2 Comparaison des méthodes IDIM-IV et DIDIM

Dans la partie précédente, il a été montré l'intérêt de considérer le couple d'entrée pour l'identification de robot. De plus, la méthode DIDIM est apparue comme la méthode la plus efficace pour appliquer le principe de l'erreur de sortie à un robot industriel. Tout comme la méthode IDIM-IV, la méthode DIDIM repose sur la simulation d'un modèle auxiliaire. Cette partie est dédiée à la comparaison de ces deux méthodes.

### Similarité entre les méthodes

À première vue, les deux méthodes sont très similaires. En effet, à l'itération  $it$ , les paramètres regroupés dans le vecteur  $\boldsymbol{\theta}$  sont estimés via :

$$\hat{\boldsymbol{\theta}}_{IV}^{it+1} = \left[ \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{X} \right]^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{IV}^{it})^T \mathbf{y}(\tau), \quad (\text{D.11})$$

$$\hat{\boldsymbol{\theta}}_{DIDIM}^{it+1} = \left[ \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it})^T \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it}) \right]^{-1} \mathbf{Z}(\hat{\boldsymbol{\theta}}_{DIDIM}^{it})^T \mathbf{y}(\tau), \quad (\text{D.12})$$

où  $\mathbf{X}$  et  $\mathbf{Z}$  sont les matrices ( $r \times b$ ) d'observations et d'instruments respectivement.  $\mathbf{X}$  est construite à partir des positions, vitesses et accélérations articulaires mesurées alors que  $\mathbf{Z}$  est construite à partir des positions, vitesses et accélérations articulaires simulées. Les deux matrices utilisent les équations issues des lois de Newton qui expriment l'entrée (couple ou force) en fonction des paramètres ainsi que les positions, vitesses et accélérations articulaires. Cet ensemble d'équations est

nommé modèle dynamique inverse, *Inverse Dynamic Model* (IDM). Les méthodes sont itératives, basées sur la simulation du modèle auxiliaire pour la construction de  $\mathbf{Z}$  et elles peuvent utiliser la même initialisation.

Le développement de la section 4.2.1 montre que si le IDM ne contient pas d'erreur de modélisation alors la matrice d'instruments est égale à partie non-bruïtée de la matrice d'observation. Il peut être montré que les deux méthodes estiment alors des paramètres égaux et sans biais. Le caractère sans biais de la méthode DIDIM a été montré dans [Gautier *et al.* 2013a].

Si l'IDM contient une erreur de modèle, comme un frottement mal modélisé, cette erreur est présente dans les matrices d'observations et d'instruments par construction. Dans ce cas, les deux méthodes fournissent des résultats équivalents et biaisés.

### Différence entre les méthodes

La différence entre les deux méthodes se situe dans l'utilisation de la matrice d'observations pour la méthode IDIM-IV (D.11). Cela peut être vu comme un inconvénient parce que la mesure des positions articulaires est alors requise. De plus, un filtrage précis des signaux est nécessaire afin de retrouver les dérivées via différences finies. Comme cela a été mis en évidence dans [Gautier *et al.* 2013a], la méthode DIDIM présente l'avantage de n'avoir besoin que de la mesure du couple (ou force).

Cependant, il a été montré dans la section 4.2.2 que la méthode IDIM-IV peut être plus robuste que la méthode DIDIM, dans le cas où une erreur est présente dans le modèle mais pas dans l'IDM. L'erreur peut se trouver par exemple dans le modèle des actionneurs ou encore le contrôleur. Entre d'autres termes, nous considérons une erreur dans le simulateur mais pas dans la matrice  $\mathbf{X}$ . En pratique deux cas doivent être considérés :

- Si l'espérance de l'erreur introduite est nulle alors l'estimation via la méthode IDIM-IV est non-biaisée contrairement à celle via la méthode DIDIM ;
- Si l'espérance de l'erreur introduite n'est pas nulle alors les deux méthodes fournissent une estimation biaisée.

### Résultats expérimentaux

Dans un premier temps, les deux méthodes ont été appliquées dans un cas nominal où le modèle auxiliaire ne contient pas d'erreur. Les résultats, fournis dans le tableau 4.6, sont équivalents et similaires à ceux de précédents travaux comme [Gautier *et al.* 2013a].

Dans un second temps, une erreur a été introduite dans le modèle auxiliaire. Il s'agit d'une erreur sur les rapports de transmission qui sont pris égaux à 80% de leurs valeurs réelles. À la vue des résultats du tableau 4.6, les paramètres estimés par la méthode DIDIM sont biaisés et en particulier les inerties comme  $zz_{1,r}$ ,  $zz_{2,r}$  on encore  $ia_3$ . A contrario, les paramètres estimés par la méthode IDIM-IV sont sans

biais, ce qui démontre la meilleure robustesse de cette méthode face à une erreur dans le simulateur.

Un point mérite une attention particulière : l'estimation des covariances pour la méthode IDIM-IV. En effet, la formule usuelle suppose que la matrice d'instruments est une approximation de la partie non-bruitée de la matrice d'observation. Or, avec une erreur persistante dans le simulateur, ce n'est pas le cas. C'est pourquoi la formule non approchée (4.34) doit être employée en cas de doute sur le simulateur.

### D.2.3 Remarques et conclusion

Dans la partie D.2.1, nous avons considéré trois méthodes d'erreur de sortie pour l'identification de robot : les méthodes CLOE, CLIE et DIDIM. En fait, elles ne sont pas des méthodes à erreur de sortie stricto sensu car elles ont besoin d'un filtrage parallèle afin de fournir des estimées avec des résidus non-corrélés. S'il a été mis en évidence l'intérêt de considérer le signal d'entrée plutôt que celui de sortie, ce résultat peut être étendu aux méthodes IDIM-LS et IDIM-IV. Dans leur cas, la sensibilité à prendre en compte est celle du système réel au lieu du système simulé. Pour en revenir aux méthodes à erreur de sortie dans le cadre robotique, il est plus intéressant d'employer la méthode DIDIM si l'hypothèse PLR est vérifiée. Cela permet en effet de réduire significativement le temps de calcul.

Deux éléments sont fondamentaux dans l'analyse de la sensibilité effectuée. Le premier est l'hypothèse de bon tracking ou, autrement dit, que la loi de commande est performante. Le deuxième est l'hypothèse sous-jacente que le système est bien actionné. C'est-à-dire qu'il y a un capteur et un actionneur par axe. Sans ces deux éléments, les résultats observés ne seraient plus valides. En d'autres termes, pour un système ayant une loi de contrôle avec une faible capacité de tracking et étant sous-actionné, il pourrait être plus intéressant de considérer le signal de sortie pour l'identification. La partie 4.3.2 donne un tel exemple basé sur l'identification avion.

Toutes les méthodes d'identification présentées dans cette section requièrent une certaine connaissance *a priori*. Par exemple, la méthode IDIM-IV a besoin des bandes-passantes du systèmes afin de régler le filtre passe-bande. Cette information est également nécessaire aux méthodes DIDIM et IDIM-IV pour régler le filtre decimate. De plus, la connaissance des actionneurs et du contrôleur est incluse dans le simulateur. Dans la section suivante, nos efforts portent sur la réduction de cette connaissance *a priori* nécessaire.

## D.3 Identification de systèmes robotiques sans connaissance du contrôleur et des bande-passantes

Dans la section précédente nous avons étudié la robustesse des méthodes IDIM-IV et DIDIM. Ces méthodes reposent sur un préfiltrage minutieux qui nécessite une connaissance des bandes-passantes. De plus, elles ont besoin de la connaissance du contrôleur pour la simulation du modèle auxiliaire. Il apparaît ainsi un dilemme

classique de l'identification de systèmes : pour identifier correctement un système, nous avons besoin d'une bonne connaissance *a priori*.

### D.3.1 Estimation des dérivées articulaires

#### Estimation de dérivées à partir de signaux bruités

De nombreuses recherches portent sur la différentiation numérique ; voir par exemple [Diop *et al.* 1994, Vasiljevic & Khalil 2006, Dridi 2011, Sidhom 2011] et les références données par les auteurs. Nous récapitulons ici certaines de ces méthodes afin d'avoir une vue d'ensemble. Néanmoins, l'objectif est pour nous de sélectionner une méthode simple qui ne requière pas de connaissance *a priori* sur le système.

La problématique d'estimation des états d'un système inconnu n'est pas récente pour la communauté d'automatique. Cela a fait l'objet d'une longue littérature sur les *observateurs*. L'objectif est de fournir les dérivées des signaux au contrôleur afin de réduire le nombre de capteurs nécessaires. Le problème a été introduit par Luenberger [Luenberger 1971]. Dans le cas d'un signal bruité, un observateur stochastique, comme celui de Kalman, peut être plus approprié ; voir par exemple [Singer 1970]. Il existe également une recherche abondante sur les observateurs non-linéaires ; voir entre autres [Dabroom & Khalil 1997, Besançon 2007]. Par exemple, le filtre de Kalman étendu peut être considéré pour l'estimation des états d'un système nonlinéaire stochastique, comme dans [Ljung 1979]. Qu'il soit déterministe ou stochastique, un observateur est basé sur le modèle dynamique du système. Ainsi, un observateur n'est pas la solution pour l'estimation des dérivées articulaires dans le but d'identifier le modèle dynamique. Bien qu'une solution pourrait consister à estimer les paramètres et les états en même temps comme dans [Gautier & Poignet 2001], nous cherchons une solution moins complexe, ne couplant pas l'estimation des paramètres et des états.

Une autre approche consiste à ne considérer que les signaux et non les modèles. Il est alors plus approprié de parler de *différentiateur*. La technique usuelle de filtre passe-bande, décrite dans la section 2.3.1, appartient à ce type d'approche. Il existe une autre méthode basée sur l'interpolation polynomiale sur une fenêtre glissante ; voir par exemple [Klein & Morelli 2006]. La dérivée du signal est alors approximée la dérivée du polynôme.

Il existe également les différentiateurs algébriques qui ont été récemment révisités ; voir entre autres [Fliess & Sira-Ramírez 2003]. L'idée est d'estimer les dérivées du signal grâce à des intégrations successives. Cela est basé sur une troncature d'une série de Taylor du signal considéré. Ainsi l'utilisateur doit fournir l'ordre de la troncature. Puisque les trajectoires utilisées pour l'excitation de robots ne sont pas très lisses, l'ordre requis peut être élevé.

Une autre famille de techniques est basée sur la théorie des modes glissants introduite dans les années 1960 [Filippov 1960]. Parmi les techniques développées, nous pouvons citer celle de l'algorithme Super Twisting introduite par [Levant 1998]. L'Eq. (5.1) donne le principe de fonctionnement de la méthode qui repose sur deux

hyper-paramètres que l'utilisateur doit régler.

Finalement, la solution que nous avons retenue est intitulée l'*Integrative Random Walk Smoother* (IRWSM). Elle a l'avantage pratique de ne pas avoir besoin de connaissance *a priori*. Cette solution est développée dans le paragraphe suivant.

### Integrative random walk smoother

La méthode IRWSM est basée sur le filtre de Kalman dans un cadre temps-discret. Cette méthode est récapitulée dans [Norton 1975, Young 2000]. Elle permet une estimation hors-ligne des états sans utiliser le modèle dynamique, à la différence des observateurs grands gains par exemple. L'Eq. (D.13) définit le vecteur d'état, avec  $x$  le signal de position et  $\nabla x$  son taux de variation. la relation (D.14) est l'équation d'état et (D.15) est l'équation d'observation.

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ \nabla x(k) \end{bmatrix} \quad (\text{D.13})$$

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{D}\boldsymbol{\kappa}(k-1) \quad (\text{D.14})$$

$$y(k) = \mathbf{h}(k)\mathbf{x}(k) + e(k) \quad (\text{D.15})$$

Avec,

$$\mathbf{A} = \begin{bmatrix} p_1 & p_2 \\ 0 & p_3 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} p_4 & 0 \\ 0 & p_5 \end{bmatrix} \quad (\text{D.16})$$

$\mathbf{h}$  est le vecteur ( $1 \times 2$ ) d'observation.  $\boldsymbol{\kappa}$  est le bruit d'état, supposé blanc avec une matrice de covariance  $\mathbf{Q}_{\boldsymbol{\kappa}}$  (diagonale). Le bruit de mesure  $e$  est également blanc avec une variance  $\sigma_e^2$ . Ce modèle, développé dans [Young 2011], est intitulé *Generalized Random Walk* (GRW). Plusieurs variantes existent suivant le choix des hyper-paramètres  $\boldsymbol{\gamma} = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ \mathbf{Q}_{\boldsymbol{\kappa}_{11}} \ \mathbf{Q}_{\boldsymbol{\kappa}_{22}}]$ . Pour cette étude, l'*Integrated Random Walk* (IRW :  $p_1 = p_2 = p_3 = p_5 = 1$ ,  $p_4 = 0$  et  $\mathbf{h} = [1 \ 0]$ ) est considéré. Dans ce cas, puisque  $p_4 = 0$ , le terme  $\mathbf{Q}_{\boldsymbol{\kappa}_{11}}$  n'a pas d'influence et est pris égal à  $\mathbf{Q}_{\boldsymbol{\kappa}_{22}}$  afin de s'assurer que la matrice de covariance reste définie positive. Au final, le seul hyper-paramètre restant est  $\mathbf{Q}_{\boldsymbol{\kappa}_{22}}$ . Comme il a été montré dans la section 5.1.2, cet hyper-paramètre peut être estimé via une optimisation du maximum de vraisemblance.

En pratique ce modèle IRW est utilisé par un filtre de Kalman qui est couplé à un processus de lissage, *Fixed Interval Smoother* (FIS), afin de profiter de l'aspect hors-ligne. De plus, le filtre et le lisseur sont modifiés afin d'éviter la connaissance de la variance du bruit d'observation,  $\sigma_e^2$ . Dans un filtre de Kalman traditionnel, cette information est en effet requise, tout comme la covariance du bruit d'état,  $\mathbf{Q}_{\boldsymbol{\kappa}}$ . À la place, toutes les équations sont écrites en fonction du *Noise Variance Ratio* (NVR) qui est défini par  $\mathbf{Q}_{nvr} = \mathbf{Q}_{\boldsymbol{\kappa}}/\sigma_e^2$ . L'algorithme décrit dans [Young 2011] est récapitulé avec les équations (5.7) à (5.15). La méthode IRWSM désigne la combinaison du filtre de Kalman, du FIS et du modèle IRW. Une fois cette méthode appliquée, la dérivée première du signal est approximée par  $\frac{dx}{dt}(t_k) \approx \frac{\widehat{\nabla x}(k)}{t_{k+1}-t_k}$ , avec

$\widehat{\nabla x}(k)$  le second élément du vecteur d'état estimé  $\widehat{\mathbf{x}}(k|N)$ . De manière similaire,  $\mathbf{x}$  pourrait être augmenté avec  $\nabla^2 x$  pour estimer la dérivée seconde.

D'un point de vue pratique, l'algorithme IRWSM est implémenté dans la fonction `irwsm` de la Toolbox CAPTAIN, développée par une équipe de la Lancaster University; voir [Taylor *et al.* 2007] et <http://captaintoolbox.co.uk>. Heureusement, la Toolbox fournit aussi une fonction intitulée `irwsmopt` qui estime les hyperparamètres. Cette Toolbox permet ainsi à l'utilisateur de traiter les données sans connaître les bande-passantes. Cela n'empêche en rien d'être vigilant quant aux résultats.

### Application à l'identification de robots

Pour l'identification de robot, l'idée est de remplacer le filtre passe-bande habituel par l'IRWSM. Dans l'équation d'observation (D.15), la sortie  $y$  est remplacée par la mesure de l'axe  $j$  :  $q_{m_j}$ , tel que :

$$q_{m_j}(t) = q_j(t) + \tilde{q}_j(t), \quad (\text{D.17})$$

où  $q_j$  est la position articulaire et  $\tilde{q}_j$  est le bruit de capteur; voir la Figure 2.3 pour une présentation des signaux du système. D'après la section 2.2.3, ce bruit est blanc avec une variance  $\frac{1}{3}\Delta_e^2$ , où  $\Delta_e$  est la résolution de l'encodeur. Si on suppose aussi que le bruit de mesure est normalement distribué, l'équation d'observation (D.15) est valide et l'écart type du bruit  $\sigma_e \approx \Delta_q$  peut être estimé avec les données fournies par le constructeur. De plus, la position  $q_j$  est égale à l'état  $x$  défini par (D.13).

Concernant l'équation d'état (D.14), deux cas peuvent être envisagés pour retrouver l'accélération articulaire :

- Appliquer l'algorithme deux fois avec  $\mathbf{A}_{irwsm1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ ;
- Appliquer l'algorithme une fois avec  $\mathbf{A}_{irwsm2} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ .

La pertinence du modèle IRW est d'abord étudiée pour  $\mathbf{A}_{irwsm1}$ . De (D.14), il vient :

$$\begin{bmatrix} q_j(k) \\ \nabla q_j(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_j(k-1) \\ \nabla q_j(k-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \kappa_{1,j}(k-1) \end{bmatrix}. \quad (\text{D.18})$$

Le deuxième état  $\nabla q_j$  représente la vitesse. Cette information de vitesse n'est pas caractérisée par les équations dynamiques mais par une marche aléatoire. En d'autres termes, le bruit  $\eta_{1,j}$  regroupe la dynamique non-modélisée; c'est-à-dire que c'est une erreur de modèle. D'après [Young 2011], ce type de modélisation fonctionne pour des systèmes dont les états varient lentement. Pour les systèmes robotiques, le contrôleur nécessite une grande fréquence d'échantillonnage pour assurer une commande robuste à l'aide de grands gains; voir [Khalil & Dombre 2004]. Néanmoins,



la bande-passante mécanique est bien plus basse. Par conséquent, avec un IRW exprimé à la fréquence d'échantillonnage du contrôleur, l'hypothèse d'états lentement variable est validée.

Avec le système augmenté  $\mathbf{A}_{irwsm2}$ , ce n'est pas la vitesse mais l'accélération qui est caractérisée par une marche aléatoire. Comme pour la vitesse, l'hypothèse d'état lentement variable semble appropriée. L'Eq. (D.14) donne :

$$\begin{bmatrix} q_j(k) \\ \nabla q_j(k) \\ \nabla^2 q_j(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_j(k-1) \\ \nabla q_j(k-1) \\ \nabla^2 q_j(k-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \kappa_{2,j}(k-1) \end{bmatrix}. \quad (\text{D.19})$$

### Résultats expérimentaux

Dans la section 5.1.4, quelques tests exécutés sur le robot EMPS ont montré que l'IRWSM est une solution appropriée pour la différentiation de signaux d'un système robotique. De fait, la technique IRWSM a été testée pour l'identification du robot industriel TX40. D'abord, dans la section 5.3.1, la méthode d'identification IDIM-LS est considérée via un premier test comparatif avec des filtres Butterworth et decimate bien réglés. Le Tableau 5.5 montre que les paramètres ainsi obtenus sont tout à fait similaires à ceux de référence. Ce résultat est confirmé par les erreurs relatives fournies dans le Tableau 5.6.

Afin de pousser la méthode IDIM-LS dans ses retranchements, l'identification est ensuite considérée avec des filtres mal réglés afin de prendre en compte un manque de connaissance *a priori* sur le robot. Le Tableau 5.7 donne les résultats obtenus avec deux méthodes de réglages des NVR : la méthode automatique avec la fonction `irwsmopt` et une version manuelle. Pour cette dernière, une valeur de  $10^{-5}$  a prouvé être un choix judicieux dans [Brunot *et al.* 2016b]. Les résultats montrent les difficultés de la méthode IDIM-LS classique (Butterworth et decimate) quand les bandes-passantes sont mal connues. De plus, la technique IRWSM apparaît en difficulté avec la sélection automatique du NVR. Par comparaison avec les résultats précédents, cela souligne le rôle du filtre decimate. Cela permet en effet de compenser les composantes du bruit haute-fréquence qui n'auraient pas été rejetées par la fonction `irwsm`. En effet, la fonction `irwsmopt` peut donner trop d'importance au bruit d'état par rapport à celui de mesure. L'utilisateur doit donc rester prudent avec une telle technique. Dans le cas manuel, les paramètres estimés sont satisfaisant mais pas leurs écarts-types. Comme le montre la Figure 5.3, les résidus sont en effet auto-corrélés, ce qui va à l'encontre des hypothèse de calcul des écarts-types. Néanmoins, la bonne estimation des paramètres dynamiques, dans un premier temps, pourrait permettre d'estimer les bandes-passantes afin de régler correctement les filtres pour une identification complète, dans un second temps.

Afin d'être complet, la technique IRWSM est également étudiée avec la méthode IDIM-IV dans la section 5.3.2. Avec le mauvais réglage des filtres Butterworth et decimate, les méthodes donnent les résultats du Tableau 5.8. Que ce soit avec l'IRWSM ou la méthode de différentiation classique, la méthode IDIM-IV fournit les mêmes

résultats. Ici aussi, les écarts-types estimés ne sont pas recevables à cause de l'autocorrélation résidus résultant du mauvais réglage du filtre decimate. La technique IRWSM présente tout de même deux avantages. Elle permet de ne pas se poser de questions quant aux bandes-passantes et, dans ce cas, elle requière moins de temps de calcul ; voir le Tableau 5.9. Un dernier test critique est réalisé en omettant totalement les filtres. Le Tableau 5.9 montre que la méthode usuelle basée sur la différentiation finie ne converge pas dans la limite de 20 itérations. Au contraire, avec la technique IRWSM, le méthode IDIM-IV converge vers des résultats satisfaisants comme le montrent les paramètres exposés dans le Tableau 5.11. À noter que la mauvaise estimation des écarts-types n'est pas résolue.

### D.3.2 Identification du contrôleur

#### Identification avec un contrôleur inconnu

Dans cette partie, nous traitons de l'identification de robot dans le cas où le contrôleur est inconnu. L'idée est d'étudier, par exemple, un robot acheté "sur étagère" avec un contrôleur sans code source ouvert. Une solution pourrait consister à employer la méthode IDIM-LS qui n'a pas besoin de la connaissance du contrôleur pour la simulation. Cependant, comme il a été montré dans [Brunot *et al.* 2015], il est plus intéressant de considérer la méthode IDIM-IV si les bandes-passantes ne sont pas certaines. C'est vraisemblablement le cas pour le premier processus d'identification. L'inconvénient est que la méthode IDIM-IV nécessite le contrôleur pour simuler le modèle auxiliaire. C'est pourquoi nous considérons ce problème dans cette partie.

La communauté d'identification a d'ores et déjà envisagé ce problème et développé des méthodes dédiées pour les systèmes LTI. Quelques méthodes à deux étapes sont récapitulées ci-après.

- La méthode à deux étapes de [Van Den Hof *et al.* 1992] consiste à identifier le transfert entre  $\mathbf{q}_r$  et  $\boldsymbol{\tau}$  grâce à la méthode PEM. Ensuite, la fonction de transfert (TF) entre  $\hat{\boldsymbol{\tau}}$  et  $\mathbf{q}_m$  est identifiée à son tour. Cette méthode est désignée par l'acronyme PEM<sup>2</sup>.
- Dans [Forssell & Ljung 2000], les auteurs proposent d'utiliser une méthode non-paramétrique pour la première étape ; la deuxième est identique à celle de PEM<sup>2</sup>. La mesure  $\boldsymbol{\tau}$  est projetée sur le signal de référence. Ainsi cette méthode est nommée *projection method* (PROJ). Cette solution a l'avantage de ne pas requérir une modélisation du contrôleur.
- Dans [Young *et al.* 2009], l'auteur propose deux variantes de l'approche à deux étapes pour les modèles à temps continu, à partir de la méthode *Refined Instrumental Variable* (RIV) ; voir la section 3.2. D'un part, la première variante est parfaitement similaire à l'approche PEM<sup>2</sup> puisque la méthode RIV est utilisée à la place de la PEM. D'autre part, la seconde variante consiste à identifier la TF entre  $\mathbf{q}_r$  et  $\boldsymbol{\tau}$  ainsi que celle entre  $\mathbf{q}_r$  et  $\mathbf{q}_m$ . Pour la deuxième

étape, la TF entre  $\hat{\tau}$  et  $\hat{q}$  est identifiée. Ces méthodes sont désignées par l'acronyme RIV<sup>2</sup>.

- L'article [Agüero *et al.* 2011] doit également être cité. L'idée de cette contribution est d'introduire un "contrôleur virtuel". Cette méthode du contrôleur virtuel (VCM) fait intervenir quatre TF auxiliaires qui permettent une décorrélation entre le signal d'entrée et le bruit, sans faire aucune hypothèse sur la loi de commande.

À l'origine, les méthodes PEM<sup>2</sup> et PROJ ont été créées pour l'identification de modèles à temps discret. Néanmoins, elles peuvent être étendues à des modèles temps continu. Le Tableau 5.3 récapitule ces différentes méthodes.

À y regarder de plus près, les méthodes citées précédemment ne sont pas toutes égales. En effet, si PROJ et VCM ne suppose rien quant à la loi de commande, PEM<sup>2</sup> et RIV<sup>2</sup> ont besoin au moins de sa structure. Cependant, VCM requière la synthèse de quatre TF auxiliaires, ce qui peut s'avérer chronophage. Nous pouvons donc s'interroger sur la pertinence de dépenser trop de temps et d'efforts à l'identification du contrôleur si l'objectif final est l'identification du modèle mécanique. De ce point de vue, la méthode PROJ semble plus attrayante. Mais toutes ces méthodes supposent que le système est linéaire vis-à-vis des états et des paramètres. Or ce n'est pas le cas des systèmes robotiques qui sont linéaires vis-à-vis des paramètres mais pas des états.

De cet état de l'art rapide, le problème de l'identification du contrôleur apparaît être gouverné par deux éléments. Le premier élément est la connaissance initiale du contrôleur. La question est : est-ce que l'utilisateur connaît la structure de la loi de commande ? Le second élément est l'objectif final de l'identification. En d'autres termes, la question est : l'objectif est l'identification du modèle du bras ou celle du système bouclé complet ? Suivant la réponse à ces deux questions, différentes approches peuvent être considérées pour traiter de l'identification du contrôleur. Le Tableau 5.4 récapitule les approches disponibles pour l'utilisateur. De ces méthodes, l'idée générale d'une approche à deux étapes est gardée :

- Identification du contrôleur avec une méthode paramétrique ou non ;
- Identification du modèle dynamique avec la méthode IDIM-IV ou DIDIM.

En pratique, nous pouvons envisager d'inclure l'identification du contrôleur dans celle du modèle dynamique via un modèle augmenté, comme cela a été proposé dans [Gilson *et al.* 2011] par exemple. Néanmoins, nous préférons séparer les deux problèmes d'identification afin d'éviter toute corrélation entre les deux modèles et l'impact d'une potentielle erreur de modélisation. Par la suite, les approches pour la première étape (identification du contrôleur) sont détaillées.

### Identification paramétrique du contrôleur

Pour identifier le contrôleur, différentes stratégies peuvent être développées suivant la connaissance *a priori*. Si la structure du contrôleur est connue par l'utilisateur,

une méthode d'identification paramétrique est une solution immédiate. Si la structure est inconnue alors la loi de commande (modèle et paramètres) peut être identifiée avec de nombreuses méthodes classiques comme la PEM [Ljung 1999] ou la *Simplified Refined IV* (SRIV) [Young 2011] par exemple. Il est supposé que les signaux suivants sont accessibles : les entrées du contrôleur  $q_r$ ,  $\hat{q}$ ,  $\hat{q}$  et  $\hat{q}$ , ainsi que sa sortie  $\nu_\tau$ . Il s'agit ainsi d'un problème d'identification multi-entrée mono-sortie (MISO) qui peut être traité par la méthode qui convient le mieux à l'utilisateur ; voir par exemple [Pascu *et al.* 2016].

Comme expliqué dans [Khalil & Dombre 2004], un terme de pré-compensation (feedforward) peut être utilisé afin de réduire l'erreur de tracking. Par conséquent, il peut être pertinent de prendre aussi en compte  $\dot{q}_r$  et  $\ddot{q}_r$  comme entrées, pour l'identification de la loi de commande.

D'un point de vue de l'identification de système, l'identification du contrôleur n'apparaît pas comme un problème trop complexe en théorie. En effet, le bruit contenu dans  $\nu_\tau$  provient seulement des entrées car  $\nu_\tau$  est une sortie calculée et non pas mesurée. Il n'y a ainsi pas de bruit de mesure. Néanmoins, cela ne préjuge en rien de la potentielle complexité à identifier le modèle du contrôleur.

### Identification non-paramétrique du contrôleur

Avec une autre perspective, l'utilisateur peut identifier seulement la réponse impulsionnelle de loi de commande avec une technique non-paramétrique, afin d'éviter toute réflexion sur la structure de la loi. Cette technique peut être celle de l'analyse de corrélation (correlation analysis) décrite dans la section 5.2.3. Elle permet d'identifier un filtre à réponse impulsionnelle finie (FIR) par axe. Les filtres peuvent ensuite être utilisés pour la simulation du modèle auxiliaire. Ils sont identifiés tel que  $\nu_{\tau_j}(t) = IR_j(z^{-1})(q_{r_j}(t) - q_{m_j}(t))$ , pour chaque axe  $j$ . Le contrôleur ainsi identifié est nommé contrôleur corrélé (CRC).

Le contrôleur identifié peut être biaisé s'il s'agit en réalité d'un filtre à réponse impulsionnelle infinie, ou encore si le contrôleur est non-linéaire. Néanmoins, un contrôleur biaisé peut être utilisé pour la simulation tant que les signaux simulés sont assez proches de ceux mesurés afin d'assurer la corrélation entre les matrices d'observation et instrumentale. Ce concept de contrôleur biaisé pour le modèle auxiliaire est valide pour la méthode IDIM-IV mais pas DIDIM. Cela provient de la meilleure robustesse de la méthode IDIM-IV démontrée dans la partie D.2.2.

### Résultats expérimentaux

Pour illustrer le cas où le contrôleur est inconnu, nous avons tout d'abord identifié un modèle paramétrique avec la méthode SRIV disponible dans la Toolbox CAPTAIN. La loi de commande ainsi identifiée est composée de six PID notés  $\hat{C}$  ; voir l'annexe B.3 pour plus d'information sur la loi de commande. Pour un axe  $j$ , l'entrée et la sortie du PID sont respectivement  $(q_{r_j} - q_{m_j})$  et  $\nu_{\tau_j}$ . Dans ce cas, la matrice d'observation est construite avec la technique IRWSM 1 et le filtre decimate est réglé avec une fréquence de coupure à 20 Hz.

Nous considérons également une identification non-paramétrique via la fonction `cra` de la Toolbox System Identification ; voir [Ljung 1988, Ljung 1999]. Cette fonction suit le principe décrit dans la section 5.2.3. Un filtre FIR est identifié par axe pour simuler le contrôleur. La taille du filtre est notée  $m_j$ . La fonction `cra` permet de blanchir l'entrée au préalable avec un filtre AR de taille  $na_j$ , pour l'axe  $j$ . Le Tableau 5.12 fournit les tailles des filtres utilisés pour identifier les six réponses impulsionnelles. Que ce soit pour l'identification paramétrique ou non, la Figure 5.4 décrit le processus d'identification avec la méthode IDIM-IV.

En ce qui concerne la convergence, les méthodes IDIM-IV (CRC) et IDIM-IV ( $\hat{C}$ ) ont eu besoin de 3 et 4 itérations respectivement. Comme le montre le Tableau 5.12, le CRC est une solution adéquate puisque les estimées de la méthode IDIM-IV correspondent à ceux de la méthode IDIM-LS qui peut être considérée comme la référence. C'est également le cas pour la méthode IDIM-IV ( $\hat{C}$ ). L'efficacité des méthodes est confirmée par le Tableau 5.14 qui donne les erreurs relative. Le fait que le CRC fonctionne avec des réponses impulsionnelles simples (ordres faibles) peut être expliqué par l'action intégrale négligeable du contrôleur. Comme expliqué dans la section 2.2.1, les lois de commande pour les robots industriels ont souvent des actions intégrales faibles voire nulles. Quelques essais ont été entrepris pour utiliser la méthode DIDIM avec le CRC mais ils sont restés infructueux.

### D.3.3 Conclusions

Dans cette section, les méthodes d'identification de robots ont été revisitées afin d'éviter la connaissance des bandes-passantes et celle du contrôleur. Des techniques de différentiation numérique ont été considérées et la technique IRWSM a été retenue. Les résultats expérimentaux ont montré que cette technique est aussi performante que le filtrage usuel pour la méthode IDIM-LS dans le cas où les bandes-passantes sont connues. Si elles ne le sont pas, la technique IRWSM avec une valeur par défaut du NVR a prouvé être une solution acceptable. Pour la méthode IDIM-IV, l'utilisation de l'IRWSM est au moins aussi efficace que le processus habituel.

L'autre contribution de cette section est la mise en œuvre de la méthode IDIM-IV lorsque le contrôleur est inconnu. L'idée consiste à identifier le contrôleur avant de l'utiliser pour la simulation du modèle auxiliaire. Deux techniques d'identification, une paramétrique et l'autre non, ont été testées avec succès sur le robot TX40. À noter que l'identification paramétrique du contrôleur peut également s'appliquer à la méthode DIDIM.

Il est apparu qu'un filtre de Butterworth bien réglé ou la technique IRWSM sont nécessaires à l'obtention de paramètres estimés asymptotiquement sans biais. Le filtre `decimate`, lui, n'affecte que l'estimation des écarts-types (ou covariances). Ainsi un réglage automatique de sa fréquence de coupure, ou même une alternative, semble nécessaire afin d'en faciliter l'emploi par un utilisateur non-expert. La prochaine section est donc dédiée à l'étude du bruit circulant dans le système afin de régler correctement ce filtre `decimate`.

## D.4 Modélisation du bruit pour l'identification de systèmes robotiques

Dans la section précédente, le filtre passe-bande et la connaissance du contrôleur ont été contournés afin de réduire la connaissance *a priori* nécessaire à l'identification. Si les résultats ont été satisfaisants, ils ont souligné le rôle du filtre decimate pour blanchir le bruit et apporter une estimation non biaisée des covariances. Afin de régler correctement ce filtre, une solution pourrait consister à d'abord identifier correctement les paramètres, avec des covariances incorrectes, pour retrouver la dynamique du système et identifier dans un second temps les covariances avec un filtre réglé correctement. Dans cette section, nous cherchons plutôt une solution en une étape et automatisée. L'idée est de prendre en compte le modèle du bruit comme la méthode RIVC, introduite dans la partie 3.2, pour fournir des paramètres estimés avec des faibles covariances et des résidus blancs. Ces travaux ont été présentés à [Brunot *et al.* 2016a] et soumis dans [Brunot *et al.* Submitted 2017].

### D.4.1 Modélisation et identification du bruit

Dans cette partie, nous revenons sur la modélisation du bruit. Jusqu'à présent, dans les différentes situations, le bruit a été supposé blanc ou correctement traité par le filtre decimate. Parce que notre objectif est ici d'identifier le modèle du bruit vu par l'algorithme d'identification, la circulation du bruit à travers le système bouclée est étudiée.

#### D.4.1.1 Relations en boucle-fermée

Tout d'abord, nous développons les relations en boucle fermée pour l'axe  $j$ . La partie linéaire du modèle dynamique direct,  $G_j(p)$ , est définie par

$$G_j(p) = \frac{1}{p(J_j p + F_{v_j})}, \quad (\text{D.20})$$

avec  $F_{v_j}$  le coefficient de frottement visqueux et  $J_j$  donné par

$$J_j = \max_{\mathbf{q}_{nf}} (\mathbf{M}_{jj}(\mathbf{q}_{nf})). \quad (\text{D.21})$$

$J_j$  est la valeur maximale du moment d'inertie par rapport aux variations de  $\mathbf{q}_{nf}$ . Cela donne la marge de stabilité de la boucle-fermée la plus petite possible lorsque  $\mathbf{q}_{nf}$  varie. L'axe  $j$  peut être modélisé comme indiqué par la Figure 6.1, où  $d_j$  est la perturbation non-linéaire regroupant le frottement de Coulomb, les frottements centrifuges, de Coriolis, gravitationnelles et les effets de couplages conformément à (6.3). Le système présente une modélisation hybride : le contrôleur et le modèle dynamique sont à temps-continu alors que le filtre de bruit est à temps-discret.

À partir de la Figure 6.1, il vient la relation boucle-fermée hybride

$$\begin{aligned} q_{m_j}(t) &= T_j(p)q_{r_j}(t) + S_j(p)H_j(z^{-1})e_j(t) + S_j(p)G_j(p)d_j(t) \\ T_j(p) &= \frac{g_{\tau_j}C_j(p)G_j(p)}{1 + g_{\tau_j}C_j(p)G_j(p)}, \quad S_j(p) = \frac{1}{1 + g_{\tau_j}C_j(p)G_j(p)} \end{aligned} \quad (\text{D.22})$$

où  $S_j$  est la fonction de sensibilité et  $T_j$  est la fonction de sensibilité complémentaire; voir par exemple [Aström & Murray 2010]. Puisque pour chaque  $p$  :  $T_j(p) + S_j(p) = 1$ , les deux fonctions ne peuvent pas être rendues petites simultanément. Comme il est expliqué dans [Janot *et al.* 2014a] et rappelé dans la section 2.2.1, afin d'avoir un bon tracking à basses fréquences, la loi de commande est élaborée tel que  $T_j(p) \approx 1$  et par conséquent  $S_j(p) \approx 0$ . Puisque  $T_j$  est un filtre passe-bas, à hautes fréquences, nous avons la configuration opposée. Sachant que l'information importante provient du signal de référence, la fonction de transfert en  $T_j$  fournit un intervalle de fréquence idéal pour le filtrage. Dans cet intervalle, nous avons  $T_j(p) \approx 1$ ,  $S_j(p) \approx 0$  et la composante du bruit devient négligeable. Ainsi, il est cohérent de filtrer la position mesurée par une fréquence proche de la dynamique en boucle fermée afin d'obtenir un signal de position presque sans bruit.

#### D.4.1.2 Modèles de filtres

L'IDIM et la position mesurée, respectivement définis par (2.17) et (4.12), sont rappelés ici

$$\begin{aligned} \boldsymbol{\tau}(t) &= \boldsymbol{\tau}_{idm}(t) + \mathbf{v}(t) \\ \mathbf{q}_m(t) &= \mathbf{q}_{n_f}(t) + \mathbf{n}_q(t), \end{aligned} \quad (\text{D.23})$$

où  $\boldsymbol{\tau}_{idm}$  et  $\mathbf{q}_{n_f}$  sont des composantes non-bruitées,  $\mathbf{v}$  et  $\mathbf{n}_q$  sont respectivement les bruits du couple et de la position que nous voulons modéliser. De plus, grâce à l'hypothèse de trajectoire de référence non-bruitée, il vient la relation clé entre les bruits d'entrée et de sortie :

$$\mathbf{v}(t) = -\mathbf{G}_\tau \mathbf{C}(p) \mathbf{n}_q(t). \quad (\text{D.24})$$

**Approche théorique** À partir des relations en boucle-fermée de la partie précédente, nous avons des modèles théoriques des bruits vus par la position et le couple de l'axe  $j$  :

$$\begin{aligned} v_j(t) &= -g_{\tau_j}C_j(p)S_j(p)H_j(z^{-1})e_j(t) \\ n_{q_j}(t) &= S_j(p)H_j(z^{-1})e_j(t). \end{aligned} \quad (\text{D.25})$$

Il doit être noté que ces modèles supposent que la perturbation non-linéaire est non-bruitée.

**Approche empirique** Une autre manière de traiter la modélisation du bruit est de considérer la norme ISO [ISO 1998] qui doit être respectée par un robot

industriel. Une telle norme contient deux éléments clés pour évaluer les performances d'un robot : la précision et la répétabilité. Comme expliqué à la page 87 de [Siciliano *et al.* 2010], en pratique, les dimensions du robot diffèrent des notations de Denavit-Hartenberg à cause des tolérances mécaniques. Cela entraîne une différence entre la position atteinte et la position calculée par la cinématique directe. Cette écart est appelé précision. D'après la même référence, la précision est fonction de la position de la charge utile et elle est généralement inférieure au millimètre. La répétabilité est la capacité du robot à retourner à une position spécifique. Cela ne dépend pas seulement des composants mécaniques mais également du contrôleur et des capteurs. La Figure 6.2 illustre ces principe de précision et répétabilité.

De ces définitions, nous posons la relation suivante :

$$\mathbf{q}(t) = \mathbf{q}_r(t) - \boldsymbol{\varepsilon}_{tr}^q(t) = \mathbf{q}_r(t) - (\mathbf{a}_{cc}(t) + \mathbf{r}_{ep}(t)) \quad (\text{D.26})$$

où  $\boldsymbol{\varepsilon}_{tr}^q$  est l'erreur de tracking en boucle-fermée divisée en une précision déterministe  $\mathbf{a}_{cc}$  et une répétabilité stochastique  $\mathbf{r}_{ep}$ . Bien que la précision et la répétabilité aient été définies dans un cadre statique, nous les considérons ici dans un cadre dynamique. Les effets transitoires de la commande peuvent être vus comme appartenant à la précision  $\mathbf{a}_{cc}$ . Puisque la référence est non-bruitée, le bruit vu par la position mesurée est donnée par :

$$n_{q_j}(t) = -r_{ep_j}(t) + \tilde{q}_j(t), \quad (\text{D.27})$$

avec le bruit du capteur  $\tilde{q}_j(t) = H_j(z^{-1})e_j(t)$ . L'objectif de loi de commande est de rendre l'erreur de tracking  $\boldsymbol{\varepsilon}_{tr}^q$  négligeable pour positionner l'outils à l'emplacement voulu dans l'espace de travail. D'un point de vue la conception, le bruit introduit par le capteur ne doit donc pas excéder l'erreur de tracking. Ainsi, il peut même être écrit :

$$n_{q_j}(t) \approx -r_{ep_j}(t). \quad (\text{D.28})$$

La section 6.1.2 fournit quelques ordres de grandeur basés sur des robots existants afin d'évaluer la pertinence d'une telle modélisation. Si les résultats ne confirment pas théoriquement la modélisation, ils lui donnent néanmoins une certaine crédibilité.

**Bruit amplifié** Le lecteur doit garder à l'esprit que nous étudions ici le bruit du système physique et plus spécifiquement le bruit vu par le couple d'entrée. Cependant, l'utilisateur peut également introduire une autre composante stochastique dans l'IDIM en amplifiant le bruit de mesure via le processus de différentiation des positions articulaires. Il a été illustré dans la section 5.3.1 que même avec l'IRWSM l'utilisateur doit précautionneux.

**Modèle de bruit retenu** Comme il été vu, la question du bruit est complexe et nos raisonnements pratiques n'ont pas la prétention de modéliser parfaitement la circulation du bruit à travers le système non-linéaire. Pour notre objectif d'iden-



tification, nous considérerons deux cas :

$$v_j(t) = H_{\tau_j}(z^{-1})e_j(t), \quad (\text{D.29})$$

$$v_j(t) = -g_{\tau_j}C_j(p)n_{q_j}(t) = -g_{\tau_j}C_j(p)H_j^c(z^{-1})e_j(t). \quad (\text{D.30})$$

L'idée du premier modèle (6.16) est d'évaluer le filtre de bruit avec une perspective black-box (boîte noire). Le second modèle (6.17) prend en compte la relation du contrôleur et peut ainsi être considérée comme une approche grey-box (boîte grise). À partir du développement théorique, on aurait  $H_j^c = S_j H_j$  ; c'est-à-dire que le filtre du bruit est pondéré par la fonction de sensibilité en boucle fermée. A contrario, à partir du raisonnement empirique,  $H_j^c$  serait un filtre façonnant la partie bruitée de l'erreur de tracking :  $n_{q_j}(t) = H_j^c(z^{-1})e_j(t) = -r_{ep_j}(t)$ . L'idée de ce second modèle de bruit, avec deux variantes, est de voir l'influence du contrôleur et particulièrement si cette connaissance du contrôleur permet une meilleure estimation. La section 6.1.2 donne des éléments afin de distinguer entre les deux variantes. Dans tous les cas, le bruit est supposé être modélisé avec une densité spectrale rationnelle et indépendant entre les axes.

### D.4.1.3 Identification des filtres auto-regressifs

Le modèle du bruit peut être identifié grâce à l'approche de la variable instrumentale pour les modèles autorégressifs à moyennes glissantes, *AutoRegressive - Moving - Average* (ARMA), décrits dans [Young 2006]. Cette méthode est implémentée dans la fonction `ivarma` de la Toolbox Captain. Pour un axe donné, la fonction a besoin des tailles du numérateur et du dénominateur du modèle ARMA. Une solution consiste à identifier seulement le modèle autorégressif, *AutoRegressive* (AR), qui donne le meilleur critère d'information d'Akaike, *Akaike Information Criterion* (AIC), sans prendre en compte le filtre à moyenne glissante [Akaike 1974]. Cette approche est implémentée dans la fonction `aic` de la Toolbox Captain. Une autre solution consiste à estimer les tailles des filtres MA et AR par une inspection visuelle des leurs fonctions d'autocorrélation et d'autocorrélation partielle, comme expliqué dans le chapitre 6 de [Box & Jenkins 1976]. Puisque notre objectif est de mettre en place une méthode automatisée qui ne requière pas de compétences particulières de la part de l'utilisateur, nous nous sommes concentrés sur la technique `aic`. Son fonctionnement est détaillé dans la section 6.1.3.

## D.4.2 Extension des méthodes d'identification

### D.4.2.1 La méthode PEM séparable

Si la PEM est une méthode attractive, elle demande d'identifier le modèle dynamique et le bruit de modèle en même temps, comme cela est visible avec (3.17). Cela soulève deux problèmes pour une application robotique. Premièrement, nous n'avons pas d'idée *a priori* de l'ordre du filtre. Deuxièmement, l'identification des deux modèles en même temps peut mener à un problème d'optimisation non-linéaire

complexe, alors qu'il a été simplifié avec l'hypothèse de régression pseudo-linéaire. Inspirés par la méthode RIVC, nous proposons donc une approche séparable en considérant l'erreur de prédiction (6.26) où les paramètres inconnus sont regroupés dans le vecteur  $\boldsymbol{\rho}^T = [\boldsymbol{\theta}^T \quad \boldsymbol{\eta}^T]$ , avec  $\boldsymbol{\eta}$  le vecteur contenant les paramètres de bruit. La méthode proposée, la PEM séparable (SEP-PEM), est composée de trois étapes successives.

1. Identification du **modèle dynamique**. Les paramètres physique sont estimés avec la méthode DIDIM sans le filtre decimate.
2. Identification du **modèle de bruit**. Pour chaque axe  $j$ , obtenir une estimation des paramètres de bruit,  $\hat{\boldsymbol{\eta}}_j$ , et une estimation de la variance du brut,  $\hat{\lambda}_j$ . Le filtre du bruit d'entrée peut être identifié à partir de (6.27) pour le modèle black-box ou de (6.28) pour le modèle grey-box.
3. **Estimation de la covariance**. L'estimée de la matrice de covariance des erreurs paramétriques est calculée avec (6.29) pour les paramètres physiques.

La séparation de l'identification des paramètres physiques et de ceux du bruit implique que les deux modèles sont statistiquement indépendants. Avant de mettre en œuvre la deuxième étape, nous devons être sûrs que la première ait convergé et qu'il n'y ait pas de d'erreurs de modèle significatives. Cependant, comme il a été montré dans la section 4.1.4, les méthodes CLIE et DIDIM peuvent converger vers des paramètres consistant sans filtrage decimate. La première étape de la méthode est donc admissible.

#### D.4.2.2 La méthode IDIM-PIV

Toujours inspirés par la méthode RIVC, nous reconsidérons la méthode IDIM-IV pour prendre en compte le filtre de bruit à la place du filtre decimate. La technique ainsi introduite est nommée variable instrumentale préfiltrée, *Prefiltered IV* (PIV). Le terme "refined" n'est pas employé car la méthode RIVC a été développée pour des systèmes LTI. Nous ne développons pas ici une nouvelle théorie pour les systèmes non-linéaires, tels que les robots, mais nous modifions l'algorithme pour qu'il soit applicable aux robots. Avec les mêmes notations que la méthode SEP-PEM, l'algorithme IDIM-PIV est implémenté ainsi :

1. **Initialisation**. Pour les paramètres physiques initiaux,  $\hat{\boldsymbol{\theta}}^0$ , nous utilisons les valeurs CAO pour les inerties. Les autres paramètres physiques sont mis à zéro. Pour chaque axe, le filtre de bruit initial est réglé tel que :  $H_j(z^{-1}, \hat{\boldsymbol{\eta}}_j^0) = 1$  avec  $\hat{\lambda}_j^0 = 1$ .  
Si le contrôleur est inconnu alors il doit être identifié avec une approche décrite dans la section 5.2 afin de simuler le modèle auxiliaire et générer les instruments. Une fois le contrôleur disponible, l'étape 2 peut être entreprise. La matrice d'observation devrait être construite avec la technique IRWSM afin d'avoir un processus automatisé.

2. **Itération** : répéter jusqu'à convergence les étapes suivantes ( $k$  représente la  $k^{\text{ème}}$  itération)
  - (a) Simuler le modèle auxiliaire pour obtenir les signaux non-bruités pour les instruments en utilisant  $\hat{\theta}^{k-1}$ .
  - (b) Obtenir l'estimation des paramètres physiques avec la relation (6.30).
  - (c) Pour chaque axe  $j$ , obtenir une estimation des paramètres de bruit,  $\hat{\eta}_j^k$ , et une estimation de la covariance du bruit,  $\hat{\lambda}_j^k$ . Le filtre du bruit d'entrée  $H_{\tau_j}$  peut être identifié soit avec (6.27) pour le modèle black-box, soit avec (6.28) pour le modèle grey-box.
  
3. **Estimation de la covariance**. Après convergence, l'estimée de la matrice de covariance des erreurs paramétriques est calculée avec (6.31) pour les paramètres physiques.

Si la méthode est très similaire à la PEM séparable introduite dans la partie précédente, il y a quelques différences. Premièrement, la méthode IDIM-PIV identifie itérativement le modèle dynamique puis les filtres de bruit. Une erreur dans les filtres de bruit pourrait ainsi contaminer le modèle dynamique et inversement. L'utilisateur doit donc être précautionneux. Deuxièmement, puisqu'il s'agit d'une méthode IV, la matrice d'observation est requise et par conséquent la mesure des positions articulaires aussi. Néanmoins, au final, les deux méthodes partagent la même troisième étape pour estimer les covariances.

#### D.4.2.3 Commentaires sur les extensions

Cette partie est dédiée à trois commentaires relatifs aux méthodes étendues proposées ci-avant. Nous voulons d'abord souligner la différence de philosophie entre le filtre decimate et l'identification du filtre de bruit. Dans les deux cas, l'objectif est de fournir des résidus blancs pour avoir des variances asymptotiquement non-biaisée et minimales. Avec le filtre decimate, l'utilisateur se focalise sur la bande-passante du système mécanique, où le bruit est supposé blanc, et rejette tout le reste. Avec l'identification du filtre de bruit, l'idée est de blanchir les résidus sur l'ensemble des fréquences du système. Cela peut mener à deux problèmes pratiques. Premièrement, les filtres de bruit sont vraisemblablement des filtres passe-bas. Leurs inverses amplifient donc les hautes fréquences comme des flexibilités non-modélisées par exemple. Deuxièmement, le processus est plus coûteux en temps de calcul que le filtre decimate. Comme souligné dans la section 4.1.4, avec le filtre decimate l'optimiseur doit considérer  $N = 276$  points d'échantillonnage contre  $n_m = 34500$  pour l'identification des filtres de bruit.

Le deuxième élément à aborder est l'optimalité. Les méthodes PEM et RIVC ont en effet été développées afin d'assurer l'optimalité des paramètres estimés. Mais puisqu'elles ont été développées pour des systèmes LTI, l'optimalité n'est pas assurée pour les systèmes robotiques considérés ici. Notre objectif n'est pas de mettre en

place une théorie pour l'optimalité dans le cadre non-linéaire mais d'affiner/étendre les méthodes usuelles.

Finalement, la fonction `aic` décrite dans la section 6.1.3 est reconsidérée. Si cette méthode est capable d'identifier des filtres AR, elle peut être vue comme une information d'autocorrélation. La fonction trouve en effet l'ordre du filtre qui donne le meilleur critère d'information d'Akaike. Cela indique la plage sur laquelle le signal est auto-corrélé. Par conséquent, la taille du filtre AR peut être utilisée pour régler le filtre `decimate` tel que  $n_d = \text{size}(\widehat{AR})$ . Cette solution rend le processus `decimate` automatisé tout en évitant les risques potentiels du filtrage par l'inverse du modèle de bruit.

#### D.4.2.4 Résultats expérimentaux

##### Study of the black-box noise model

Le modèle black-box a d'abord été étudié avec la méthode SEP-PEM. La comparaison avec la méthode DIDIM montre que les résultats sont satisfaisants : paramètres et variances ; voir Tableau 6.1. La Figure 6.3 montre l'efficacité du filtrage afin d'éliminer l'autocorrélation des résidus. L'ordre moyen des filtres est 141. Cet ordre est relativement élevé mais cela renforce l'idée d'utiliser la fonction `aic` pour régler le filtre `decimate`. Celui-ci garde en effet 1 échantillon sur 125 lorsqu'il est bien réglé. Malgré ces résultats encourageants, l'analyse des variances des bruits générateurs montre que ces derniers ne peuvent pas être expliqués qu'avec les erreurs des capteurs de position ; voir le Tableau 6.3.

La méthode SEP-PEM est donc capable d'identifier le modèle dynamique avec des paramètres et des variances sans biais. Néanmoins, le bruit exogène estimé ne correspond pas aux capteurs. Nous considérons donc maintenant le modèle grey-box avec la méthode IDIM-PIV, afin d'obtenir une interprétation physique. Le Tableau 6.4 récapitule des les résultats qui apparaissent être satisfaisants puisque comparables aux travaux de références. La Figure 6.4 montre que les résidus sont correctement blanchis. Concernant la pertinence du modèle de bruit, l'analyse des variances montre que le modèle n'explique pas les niveaux de bruit observés, quelque soit la variante du modèle grey-box utilisée. Ainsi, d'autres recherches seraient nécessaires afin d'expliquer la dynamique du bruit observé.

Pour finir nos tests expérimentaux, nous considérons la méthode IDIM-PIV avec le modèle black-box, mais dans un cas dégradé où les données sont plus bruitées. Comme il est visible dans le Tableau 6.7, les résultats de la méthode IDIM-PIV sont comparables à ceux de la méthode IDIM-IV, sauf que la première donne des écarts-types plus faibles. La Figure 6.5 démontre que les résidus sont bien décorrélés. Néanmoins, il apparaît un souci avec l'hypothèse d'homoscédasticité avec la Figure 6.6 vraisemblablement dû à l'amplification du bruit lors de la différentiation.

### D.4.3 Conclusions

Dans cette section, la question de la circulation du bruit à travers le système bouclé a été étudiée. L'objectif est de trouver un filtrage automatisé et d'avoir des estimées avec covariances aussi faibles que possible. Deux modèles de bruit ont été établis dans la section 6.1. À partir de ces deux modèles, les méthodes DIDIM et IDIM-IV ont été revisitées et testées avec des données expérimentales. Les résultats ont montré que ces méthodes fournissent des résultats aussi précis que les méthodes usuelles et même avec des variances plus faibles dans le cas où le système est très bruité. En pratique le modèle de bruit black-box devrait être préféré pour l'instant.

Tant qu'une interprétation physique du bruit est nécessaire, d'autres recherches devraient être menées. Le bruit introduit par la mesure de la position ne semble en effet pas suffisant pour tout expliquer. Un modèle plus fin des actionneurs pourrait être intéressant dans cette perspective.

## D.5 Conclusions et perspectives

Cette dernière section donne une vue d'ensemble des contributions présentées dans la Partie III. L'objectif est de fournir une vision claire de la manière dont l'identification de robots a été abordée. Les différentes méthodes considérées ont évidemment leurs limitations qui sont rappelées ici. De plus, des pistes pour des travaux futurs sont esquissées.

### D.5.1 Résumé des contributions

#### Analyse des méthodes à erreur de sortie

La comparaison préliminaire des méthodes DIDIM et CLOE introduite dans [Janot *et al.* 2014a] a été élargie dans le Chapitre 4 afin de prendre en compte la méthode CLIE. Il a été ainsi mis en exergue la relation entre les méthodes CLIE et DIDIM. L'intérêt de considérer ces méthodes à la place de la méthode CLOE a été démontré. Le couple en entrée présente en effet une meilleure sensibilité aux paramètres. La difficulté d'étendre ce résultat à des systèmes non-robotiques a également été abordée dans ce chapitre. De plus, le rôle du filtre decimate pour l'identification de robot a été mis en évidence. Il est ainsi apparu la nature duale de la méthode DIDIM qui n'est ni une méthode à erreur de sortie au sens strict ni une méthode d'erreur de prédiction, car le bruit de mesure n'est pas modélisé.

#### Analyse de la robustesse des méthodes IDIM-IV et DIDIM

Une des contributions majeures de cette thèse est la comparaison des méthodes IDIM-IV et DIDIM. Les deux méthodes ont prouvé leur efficacité pour l'identification de robots. Il est connu que ces deux méthodes sont itératives et partagent le même critère de convergence ainsi que les valeurs initiales. Par contre, la méthode

DIDIM n'a pas besoin du même préfiltrage minutieux que celui utilisé par la méthode IDIM-IV pour construire la matrice d'observation [Gautier *et al.* 2013a]. La nouveauté se situe dans la démonstration de la meilleure robustesse de la méthode IDIM-IV par rapport aux erreurs de modèle localisées dans le modèle auxiliaire. Cela a été validé expérimentalement avec le robot TX40. De plus, il a été mis en évidence la nécessaire attention à porter à l'estimation des covariances, si une erreur existe. Cette contribution a été présentée dans [Brunot *et al.* 2017c].

### Identification avec un contrôleur inconnu

Dans le cadre de cette thèse, dû à la nécessité de traiter des données provenant d'expériences en boucle-fermée, nous avons décidé d'aborder le cas où la loi de commande n'est pas accessible. Si certains travaux ont déjà abordé le problème, ils n'ont considéré que des systèmes LTI. Pour surmonter l'aspect non-linéaire des modèles robotiques, nous avons décidé de traiter l'identification du contrôleur séparément de celle du modèle dynamique. Deux scénarii ont été envisagés suivant l'objectif de l'utilisateur. Premièrement, si l'objectif est de connaître la loi de commande réelle, le contrôleur peut être identifié de manière paramétrique. Cela peut se révéler une tâche complexe, en particulier si des non-linéarités sont présentes. Deuxièmement, l'utilisateur peut choisir une méthode d'identification non-paramétrique afin d'économiser du temps et se concentrer sur l'identification du modèle dynamique, au prix d'une connaissance moins complète du système. Ces deux approches ont été testées avec succès sur le robot TX40 dans le Chapitre 5, en considérant les méthodes IDIM-IV et DIDIM. Ce travail a été soumis dans [Brunot *et al.* Submitted 2017].

Les perspectives de ces travaux peuvent porter sur des lois de commande plus complexes ainsi que des modèles d'actionneurs inconnus. Une évaluation de l'identifiabilité des lois de commande peut également être intéressante pour un acteur industriel. Pour le fabricant, le double objectif serait de découvrir les contrôleurs de ses concurrents tout en gardant les siens secrets.

### Estimation automatisée des dérivées articulaires

Comme il a été mentionné précédemment, les méthodes usuelles (IDIM-LS et IDIM-IV) ont besoin d'une différentiation du signal de la position mesurée, accompagnée d'un filtrage passe-bande pour éviter l'amplification du bruit. Afin de réduire l'information nécessaire à la mise en œuvre du filtrage, nous avons cherché une autre méthode automatisée pour la différentiation du signal mesuré. Nous avons sélectionné et testé la technique IRWSM, basée sur une marche aléatoire, un filtre de Kalman et un lisseur, qui fut développée à l'origine pour l'estimation de paramètres variant dans le temps. Si la marche aléatoire ne modélise pas la dynamique exacte du système, il a été vérifié qu'elle améliore les performances des méthodes d'identification, particulièrement si les bandes-passantes sont inconnues de l'utilisateur. Cependant quelques limitations ont été entrevues :

- Une version modifiée de l'algorithme prenant en compte le signal de référence comme entrée pourrait apporter en précision, d'une manière similaire à [Kostic *et al.* 2004];
- L'approche automatisée IRWSM permet d'estimer correctement les paramètres dynamiques mais pas leurs écarts-types. Une autre procédure est donc requise pour automatiser le réglage de la fréquence de coupure du filtre decimate.

### Identification du filtre de bruit

Pour traiter du réglage du filtre decimate, l'approche usuelle de la communauté de l'identification de systèmes a été étudiée. L'idée est de filtrer les données par l'inverse du filtre colorant le bruit. Cette technique a l'avantage d'assurer l'optimalité des paramètres estimés pour les systèmes LTI. Si l'optimalité n'est pas notre objectif avec les modèles non-linéaires de robots, une amélioration des estimations présente un intérêt non négligeable en même temps que l'automatisation du processus. Pour atteindre un tel objectif, une discussion a été initiée sur le bruit vu par l'algorithme d'identification provenant de l'ensemble du système bouclé. L'identification du modèle de bruit a été pensée pour être automatisée et sans connaissance *a priori* sur le système. Les méthodes IDIM-IV et DIDIM ont été reconsidérées afin d'intégrer l'identification du modèle de bruit ainsi que le filtrage résultant. En plus, il est apparu une manière alternative de régler automatiquement l'ordre du filtre decimate grâce au critère d'information d'Akaike.

Les résultats expérimentaux ont montré qu'une telle approche convient pour identifier un robot industriel d'une manière systématique et qu'elle peut même améliorer la précision des paramètres estimés. Néanmoins, il est également apparu que notre compréhension de la circulation du bruit à travers le système est encore limitée. Une perspective réside dans le développement d'un modèle plus complet du bruit.

#### D.5.2 Un guide pour l'utilisateur

Nous reconsidérons ici le processus d'identification décrit par la Figure 1.3. À partir des contributions, la Figure 7.1 décrit notre vision du processus en incluant les méthodes développées. Le bloc d'information *a priori* est remplacé par trois questions préliminaires pour l'utilisateur. Ces trois questions sont les suivantes :

- *Est-ce que l'utilisateur connaît les bandes-passantes du système ?* Si la réponse est positive, les méthodes usuelles basées sur les filtres passe-bande et decimate peuvent être employées. Sinon, les dérivées articulaires devraient être estimées avec la technique IRWSM. En ce qui concerne le blanchissement du bruit, l'utilisateur peut soit identifier un modèle black-box, soit régler le filtre decimate avec AIC. Une troisième solution, que nous préconiserions, consiste à coupler un filtre decimate mal réglé et une estimation du filtre du bruit restant, au prix d'une perte de l'interprétation physique de ce bruit.

- *Est-ce que l'utilisateur connaît le contrôleur du système ?* Si la réponse est négative, l'identification du contrôleur doit être réalisée. Si l'objectif est d'utiliser la méthode DIDIM ou s'il y a un intérêt dans la connaissance précise du contrôleur, une identification paramétrique doit être conduite. Sinon, une identification non-paramétrique couplée avec la méthode IDIM-IV permettra d'atteindre l'objectif : l'identification du modèle dynamique.
- *Y a-t-il un intérêt dans l'analyse statistique ?* Si la réponse est positive, la modélisation du bruit devrait être réalisée. Sinon, le filtre decimate habituel peut fournir des paramètres estimés avec un temps de calcul potentiellement réduit.

### D.5.3 Développements futurs

#### Modélisation des actionneurs

En ce qui concerne la modélisation, la caractérisation des actionneurs serait un développement intéressant. Tout au long de cette thèse, ils ont en effet considérés comme des sous-systèmes connus et, plus particulièrement, comme des gains statiques. Leur connaissance pourrait améliorer la connaissance du modèle de bruit. À cet égard, des efforts supplémentaires sont nécessaires à la meilleure compréhension du bruit vu par les méthodes d'identification. Ainsi, une analyse statistique plus complète pourrait être intéressante.

#### Modélisation du frottement

Comme il a été montré dans le chapitre 2, les modèles dynamiques sont formulés à partir des lois de Newton ou du principe de Lagrange. L'utilisateur doit donc avoir une connaissance complète des phénomènes physiques en présence. Une difficulté majeure est le modèle de frottement qui est généralement non-linéaire à basses vitesses et qui peut dépendre de plusieurs variables exogènes comme la température ; voir par exemple [Bittencourt & Axelsson 2014]. Par conséquent, le frottement reste un défi majeur pour l'identification de robots.

Récemment, dans [Janot *et al.* 2017], une nouvelle approche pour relever ce défi a été proposée. L'idée est de séparer le problème en deux : une partie "grey-box" regroupant les forces inertielles et une partie "black-box" pour capturer les effets non-linéaires du frottement. La partie grey-box est identifiée avec une méthode usuelle décrite dans la section 2.3 comme la méthode IDIM-IV. La partie black-box est identifiée non-paramétriquement avec la méthode *State-Dependent Parameter* (SDP). Dans un second temps, à partir du modèle non-paramétrique fourni par la méthode SDP, un modèle paramétrique du frottement peut être identifié. Cette méthode est une procédure statistique capable d'identifier la présence de non-linéarités dans le modèle dynamique en donnant leur représentation graphique. Cette procédure est basée sur les données expérimentales, avec un minimum d'hypothèses concernant la nature des non-linéarités. L'algorithme d'estimation SDP est



une extension de l'approche stochastique pour l'estimation de paramètres variant dans le temps, *Time Varying Parameter* (TVP); voir par exemple [Young 1999] et les références précédentes qui y sont indiquées. La méthode SDP représente ainsi un outil précieux qui devrait être étudié d'avantage afin d'aborder des phénomènes de frottement non-linéaires plus complexes. En plus, il pourrait être intéressant de considérer des techniques comme celle développée dans [Noël & Schoukens 2017] où le modèle est divisé en une partie linéaire et une autre non-linéaire. Les non-linéarités sont alors vues comme une entrée additionnelle appliquée au système sous-jacent.

### Perspectives générales

En ce qui concerne l'identification des robots, la quantité de connaissances requises pour le processus a été réduite dans les chapitres 5 et 6. Des développements futurs devraient porter sur l'élément du processus qui n'a pas été étudié : la conception de signaux d'excitation optimaux. En effet, dans une certaine limite, la conception de signaux d'excitation pour l'identification peut nécessiter une certaine connaissance *a priori*.

Vis-à-vis des méthodes d'identification à proprement parler, elles pourraient être adaptées afin de fonctionner dans un cadre récursif. L'objectif final serait de réaliser une identification en ligne pour la surveillance et le contrôle adaptatif. De plus, l'application de ces méthodes à d'autres systèmes en boucle fermée et/ou non-linéaires pourrait être explorée. Cela a été initié avec l'étude des méthodes à erreur de sortie pour les systèmes non-linéaires avec des fonctions de sensibilités non-définies [Brunot *et al.* 2017a].



# References

- [Agüero *et al.* 2011] Juan C. Agüero, Graham C. Goodwin and Paul M.J. Van den Hof. *A virtual closed loop method for closed loop identification*. *Automatica*, vol. 47, no. 8, pages 1626–1637, 2011. (Cited on pages 82 and 171.)
- [Akaike 1974] Hirotugu Akaike. *A new look at the statistical model identification*. *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pages 716–723, Dec 1974. (Cited on pages 103 and 177.)
- [Arimoto 1989] Suguru Arimoto. *Design of robot control systems*. *Advanced Robotics*, vol. 4, no. 1, pages 79–97, 1989. (Cited on pages 20 and 21.)
- [Åström & Bohlin 1966] Karl Johan Åström and Torsten Bohlin. *Numerical identification of linear dynamic systems from normal operating records*. In PH Hammond: *Theory of Self-Adaptive Control Systems*, pages 96–111. Plenum Press, 1966. (Cited on page 40.)
- [Åström & Murray 2010] Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010. (Cited on pages 99 and 175.)
- [Åström 1970] Karl Johan Åström. *Introduction to stochastic control theory*. Academic Press, 1970. (Cited on page 40.)
- [Åström 1980] Karl Johan Åström. *Maximum likelihood and prediction error methods*. *Automatica*, vol. 16, no. 5, pages 551–574, 1980. (Cited on pages 39 and 40.)
- [Bélanger *et al.* 1998] P.R. Bélanger, P. Dobrovolny, A. Helmy and X. Zhang. *Estimation of angular velocity and acceleration from shaft-encoder measurements*. *The International Journal of Robotics Research*, vol. 17, no. 11, pages 1225–1233, 1998. (Cited on page 22.)
- [Bendat & Piersol 2010] Julius S Bendat and Allan G Piersol. *Random data: analysis and measurement procedures*. Wiley series in probability and statistics. Wiley, 4th Edition, 2010. (Cited on page 152.)
- [Berghuis & Nijmeijer 1993] Harry Berghuis and Henk Nijmeijer. *A passivity approach to controller-observer design for robots*. *IEEE Transactions on robotics and automation*, vol. 9, no. 6, pages 740–754, 1993. (Cited on page 20.)
- [Besançon 2007] Gildas Besançon. *Nonlinear observers and applications*, volume 363. Springer, 2007. (Cited on pages 72 and 166.)

- [Bittencourt & Axelsson 2014] Andre Carvalho Bittencourt and Patrik Axelsson. *Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data*. IEEE/ASME Transactions on Mechatronics, vol. 19, no. 5, pages 1694–1706, 2014. (Cited on pages 128 and 184.)
- [Bona & Indri 2005] Basilio Bona and Marina Indri. *Friction compensation in robotics: an overview*. In Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on, pages 4360–4367. IEEE, 2005. (Cited on pages 6, 17 and 158.)
- [Box & Jenkins 1976] George E.P. Box and Gwilym M. Jenkins. Time series analysis: forecasting and control. Holden-Day Inc., revised Edition, 1976. (Cited on pages 84, 104 and 177.)
- [Box 1979] George EP Box. *Robustness in the strategy of scientific model building*. In Robustness in statistics, pages 201–236. Academic Press, 1979. (Cited on page iii.)
- [Brunot *et al.* 2015] Mathieu Brunot, Alexandre Janot, Francisco Carrillo, Hugues Garnier, Pierre-Olivier Vandanjon and Maxime Gautier. *Physical parameter identification of a one-degree-of-freedom electromechanical system operating in closed loop*. In 17th IFAC Symposium on System Identification (SYSID), volume 48, pages 823–828, 2015. (Cited on pages 10, 69, 81 and 170.)
- [Brunot *et al.* 2016a] Mathieu Brunot, Alexandre Janot, Francisco Carrillo and Maxime Gautier. *A Separable Prediction Error Method for Robot Identification*. In 7th IFAC Symposium on Mechatronic Systems, Loughborough University, Leicestershire, UK, 5–8 September 2016, volume 49, pages 487 – 492, 2016. (Cited on pages 10, 97 and 174.)
- [Brunot *et al.* 2016b] Mathieu Brunot, Alexandre Janot, Francisco Carrillo and Maxime Gautier. *State Space Estimation Method for Robot Identification*. In 7th IFAC Symposium on Mechatronic Systems, Loughborough University, Leicestershire, UK, 5–8 September 2016, volume 49, pages 228 – 233, 2016. (Cited on pages 10, 80, 85 and 169.)
- [Brunot *et al.* 2017a] Mathieu Brunot, Alexandre Janot and Francisco Carrillo. *Continuous-Time Nonlinear Systems Identification with Output Error Method Based on Derivative-Free Optimisation*. 2017. IFAC World Congress. (Cited on pages 128 and 185.)
- [Brunot *et al.* 2017b] Mathieu Brunot, Alexandre Janot and Francisco Carrillo. *State Space Estimation Method for the Identification of an Industrial Robot Arm*. 2017. IFAC World Congress. (Cited on pages 10 and 71.)
- [Brunot *et al.* 2017c] Mathieu Brunot, Alexandre Janot, Francisco Carrillo and Hugues Garnier. *Comparison Between the IDIM-IV Method and the DIDIM*

- Method for Industrial Robots Identification*. 2017. IEEE/ASME International Conference on Advanced Intelligent Mechatronics. (Cited on pages 10, 124 and 182.)
- [Brunot *et al.* 2017d] Mathieu Brunot, Alexandre Janot, Francisco Carrillo and Hugues Garnier. *A Pragmatic and Systematic Statistical Analysis for Identification of Industrial Robots*. 2017. IEEE/ASME International Conference on Advanced Intelligent Mechatronics. (Cited on page 10.)
- [Brunot *et al.* In Press 2017] Mathieu Brunot, Alexandre Janot, Peter C. Young and Francisco Carrillo. *An instrumental variable method for robot identification based on time variable parameter estimation*. *Kybernetika*, In Press 2017. (Cited on pages 10, 71 and 73.)
- [Brunot *et al.* Submitted 2017] Mathieu Brunot, Alexandre Janot, Peter C. Young and Francisco Carrillo. *An improved instrumental variable method for industrial robot model identification*. *Control Engineering Practice*, Submitted 2017. (Cited on pages 10, 97, 124, 174 and 182.)
- [Burg 1975] John Parker Burg. *Maximum entropy spectral analysis*. PhD thesis, Stanford University, 1975. (Cited on page 104.)
- [Calafiore & Indri 2000] Giuseppe Calafiore and Marina Indri. *Robust calibration and control of robotic manipulators*. In American Control Conference, 2000. Proceedings of the 2000, volume 3, pages 2003–2007. IEEE, 2000. (Cited on pages 7 and 158.)
- [Calanca *et al.* 2011] A. Calanca, L. M. Capisani, A. Ferrara and L. Magnani. *MIMO Closed Loop Identification of an Industrial Robot*. *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pages 1214–1224, September 2011. (Cited on page 102.)
- [Canudas de Wit *et al.* 1989] Carlos Canudas de Wit, P. Noel, A. Aubin, B. Brogliato and P. Drevet. *Adaptive friction compensation in robot manipulators: low-velocities*. In IEEE International Conference on Robotics and Automation, pages 1352–1357. IEEE, 1989. (Cited on pages 6, 17 and 158.)
- [Canudas de Wit *et al.* 1992] Carlos Canudas de Wit, N. Fixot and Karl Johan Åström. *Trajectory tracking in robot manipulators via nonlinear estimated state feedback*. *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pages 138–144, 1992. (Cited on page 20.)
- [Carrillo *et al.* 2009] Francisco Carrillo, Arnaud Baysse and Abdallah Habbadi. *Output error identification algorithms for continuous-time systems operating in closed-loop*. *IFAC Proceedings Volumes*, vol. 42, no. 10, pages 408–413, 2009. (Cited on pages 42 and 43.)

- [Cheong *et al.* 2004] Joono Cheong, Wan Kyun Chung and Youngil Youm. *Inverse kinematics of multilink flexible robots for high-speed applications*. IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pages 269–282, 2004. (Cited on page 16.)
- [Conrad *et al.* 2000] Kevin L Conrad, Panayiotis S Shiakolas and T.C. Yih. *Robotic calibration issues: Accuracy, repeatability and calibration*. In Proceedings of the 8th Mediterranean Conference on Control and Automation (MED2000), Rio, Patras, Greece, page 6, 2000. (Cited on page 102.)
- [Cook 2007] Michael V Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2nd Edition, 2007. (Cited on page 68.)
- [Dabroom & Khalil 1997] Ahmed Dabroom and Hassan K Khalil. *Numerical differentiation using high-gain observers*. In Decision and Control, 1997., Proceedings of the 36th IEEE Conference on, volume 5, pages 4790–4795. IEEE, 1997. (Cited on pages 72 and 166.)
- [Daemi & Heimann 1997] M Daemi and B Heimann. *Identification and compensation of gear friction for modeling of robots*. In ROMANSY 11, pages 89–96. Springer, 1997. (Cited on page 17.)
- [Davidson & MacKinnon 1993] Russell Davidson and James MacKinnon. *Estimation and inference in econometrics*. Oxford University Press, 1993. (Cited on pages 52 and 161.)
- [De Callafon 1998] Raymond A. De Callafon. *Feedback oriented identification for enhanced and robust control*. PhD thesis, Delft University of Technology, Delft, the Netherlands, 1998. (Cited on page 34.)
- [Denavit & Hartenberg 1955] Jacques Denavit and Richard Hartenberg. *A kinematic notation for lower-pair mechanisms based on matrices*. Trans. of the ASME. Journal of Applied Mechanics, vol. 22, pages 215–221, 1955. (Cited on pages 6, 13 and 158.)
- [Diop *et al.* 1994] SSD Diop, JW Grizzle, PE Moraal and A Stefanopoulou. *Interpolation and numerical differentiation for observer design*. In Proceedings of the American Control Conference, volume 2, pages 1329–1329. AMERICAN AUTOMATIC CONTROL COUNCIL, 1994. (Cited on pages 72 and 166.)
- [Dridi 2011] Mehdi Dridi. *Dérivation numérique: synthèse, application et intégration*. PhD thesis, Ecole Centrale de Lyon, 2011. (Cited on pages 72 and 166.)
- [Durbin & Koopman 2012] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Number 38. Oxford University Press, 2012. (Cited on page 75.)

- [Eykhoff 1974] Pieter Eykhoff. System identification : Parameter and state estimation. Wiley-Interscience London, 1974. (Cited on page 33.)
- [Filippov 1960] AF Filippov. *Application of the theory of differential equations with discontinuous right-hand sides to non-linear problems in automatic control*. In 1st IFAC congress, pages 923–925, 1960. (Cited on pages 72 and 166.)
- [Fliess & Sira-Ramírez 2003] Michel Fliess and Hebertt Sira-Ramírez. *An algebraic framework for linear identification*. ESAIM: Control, Optimisation and Calculus of Variations, vol. 9, pages 151–168, 2003. (Cited on pages 72 and 166.)
- [Fliess *et al.* 1995] Michel Fliess, Jean Lévine, Philippe Martin and Pierre Rouchon. *Flatness and defect of non-linear systems: introductory theory and examples*. International journal of control, vol. 61, no. 6, pages 1327–1361, 1995. (Cited on page 20.)
- [Forssell & Ljung 1999] Urban Forssell and Lennart Ljung. *Closed-loop identification revisited*. Automatica, vol. 35, no. 7, pages 1215–1241, 1999. (Cited on pages 34 and 41.)
- [Forssell & Ljung 2000] Urban Forssell and Lennart Ljung. *A projection method for closed-loop identification*. Automatic Control, IEEE Transactions on, vol. 45, no. 11, pages 2101–2106, 2000. (Cited on pages 81 and 170.)
- [Forssell 1999] Urban Forssell. *Closed-loop identification: Methods, theory, and applications*. PhD thesis, Linköping University Electronic Press, 1999. (Cited on pages 34 and 40.)
- [Garnier *et al.* 2008] Hugues Garnier, Liuping Wang and Peter C. Young. *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, 2008. (Cited on page 33.)
- [Gautier & Briot 2012a] Maxime Gautier and Sébastien Briot. *Global identification of drive gains parameters of robots using a known payload*. In International Conference on Robotics and Automation (ICRA), pages 2812–2817. IEEE, 2012. (Cited on page 143.)
- [Gautier & Briot 2012b] Maxime Gautier and Sébastien Briot. *Global Identification of Robot Drive Gains Parameters Using a Known Payload and Weighted Total Least Square Techniques*. In 16th IFAC Symposium on System Identification (SYSID), volume 45, pages 1389 – 1394, 2012. (Cited on page 21.)
- [Gautier & Khalil 1990] Maxime Gautier and Wisama Khalil. *Direct calculation of minimum set of inertial parameters of serial robots*. IEEE Transactions on robotics and Automation, vol. 6, no. 3, pages 368–373, 1990. (Cited on pages 6, 17 and 158.)

- [Gautier & Khalil 1992] Maxime Gautier and Wisama Khalil. *Exciting trajectories for the identification of base inertial parameters of robots*. The International journal of robotics research, vol. 11, no. 4, pages 362–375, 1992. (Cited on pages 30 and 31.)
- [Gautier & Poignet 2001] Maxime Gautier and Ph Poignet. *Extended Kalman filtering and weighted least squares dynamic identification of robot*. Control Engineering Practice, vol. 9, no. 12, pages 1361–1372, 2001. (Cited on pages 72 and 166.)
- [Gautier *et al.* 1994] M Gautier, PO Vandanjon and C Presse. *Identification of inertial and drive gain parameters of robots*. In Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, volume 4, pages 3764–3769. IEEE, 1994. (Cited on pages 7 and 158.)
- [Gautier *et al.* 2011] Maxime Gautier, Alexandre Janot, Anthony Jubien and Pierre-Olivier Vandanjon. *Joint stiffness identification from only motor force/torque data*. In 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pages 5088–5093. IEEE, 2011. (Cited on pages 30 and 42.)
- [Gautier *et al.* 2013a] Maxime Gautier, Alexandre Janot and Pierre-Olivier Vandanjon. *A new closed-loop output error method for parameter identification of robot dynamics*. IEEE Transactions on Control Systems Technology, vol. 21, no. 2, pages 428–444, 2013. (Cited on pages 7, 20, 23, 28, 42, 56, 63, 65, 124, 145, 159, 164 and 182.)
- [Gautier *et al.* 2013b] Maxime Gautier, Anthony Jubien and Alexandre Janot. *Iterative learning identification and computed torque control of robots*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3419–3424. IEEE, 2013. (Cited on page 145.)
- [Gautier 1991] Maxime Gautier. *Numerical calculation of the base inertial parameters of robots*. Journal of Field Robotics, vol. 8, no. 4, pages 485–506, 1991. (Cited on pages 6, 18 and 158.)
- [Gautier 1997] Maxime Gautier. *Dynamic identification of robots with power model*. In Proceedings of International Conference on Robotics and Automation, volume 3, pages 1922–1927 vol.3, Apr 1997. (Cited on pages 6, 24 and 158.)
- [Gilson *et al.* 2008] Marion Gilson, Hugues Garnier, Peter C Young and Paul M.J. Van den Hof. *Instrumental variable methods for closed-loop continuous-time model identification*. In Identification of continuous-time models from sampled data, pages 133–160. Springer, 2008. (Cited on pages 22 and 38.)
- [Gilson *et al.* 2011] Marion Gilson, Hugues Garnier, Peter C. Young and Paul M.J. Van den Hof. *Optimal instrumental variable method for closed-loop identifi-*



- cation*. IET control theory & applications, vol. 5, no. 10, pages 1147–1154, 2011. (Cited on pages 82 and 171.)
- [Gustavsson *et al.* 1977] Ivar Gustavsson, Lennart Ljung and Torsten Söderström. *Identification of processes in closed loop – identifiability and accuracy aspects*. Automatica, vol. 13, no. 1, pages 59–75, 1977. (Cited on page 34.)
- [Hairer & Wanner 1996] Ernst Hairer and Gerhard Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, 1996. (Cited on page 18.)
- [Hairer *et al.* 1993] Ernst Hairer, Syvert P. Nørsett and Gerhard Wanner. *Solving ordinary differential equations i*, volume 8. Springer-Verlag, Berlin, 2 Edition, 1993. (Cited on page 42.)
- [Hamon *et al.* 2011] Pauline Hamon, Maxime Gautier and Philippe Garrec. *New dry friction model with load-and velocity-dependence and dynamic identification of multi-dof robots*. In IEEE International Conference on Robotics and Automation (ICRA), pages 1077–1084. IEEE, 2011. (Cited on page 18.)
- [Harvey & Peters 1990] Andrew C. Harvey and Simon Peters. *Estimation procedures for structural time series models*. Journal of Forecasting, vol. 9, no. 2, pages 89–108, 1990. (Cited on page 76.)
- [ISO 1998] EN ISO. *9283: 1998*. Manipulating industrial robots-Performance criteria and related test methods, 1998. (Cited on pages 100 and 175.)
- [ISO 1999] EN ISO. *9946: 1999*. Manipulating industrial robots - Presentation of characteristics, 1999. (Cited on pages 4 and 157.)
- [Janot *et al.* 2007] Alexandre Janot, Catherine Bidard, Florian Gosselin, Maxime Gautier, Delphine Keller and Yann Perrot. *Modeling and identification of a 3 DOF haptic interface*. In Robotics and Automation, 2007 IEEE International Conference on, pages 4949–4955. IEEE, 2007. (Cited on page 31.)
- [Janot *et al.* 2014a] Alexandre Janot, Maxime Gautier, Anthony Jubien and Pierre Olivier Vandanjon. *Comparison between the CLOE Method and the DIDIM Method for Robots Identification*. IEEE Transactions on Control Systems Technology, vol. 22, no. 5, pages 1935–1941, 2014. (Cited on pages 34, 68, 99, 124, 175 and 181.)
- [Janot *et al.* 2014b] Alexandre Janot, Pierre-Olivier Vandanjon and Maxime Gautier. *A generic instrumental variable approach for industrial robot identification*. IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pages 132–145, 2014. (Cited on pages 23 and 143.)

- [Janot *et al.* 2014c] Alexandre Janot, Pierre Olivier Vandanjon and Maxime Gautier. *An instrumental variable approach for rigid industrial robots identification*. Control Engineering Practice, vol. 25, pages 85–101, 2014. (Cited on pages 7, 27, 28, 65 and 158.)
- [Janot *et al.* 2017] Alexandre Janot, Peter C Young and Maxime Gautier. *Identification and control of electro-mechanical systems using state-dependent parameter estimation*. International Journal of Control, vol. 90, no. 4, pages 643–660, 2017. (Cited on pages 18, 128 and 184.)
- [Janot 2007] Alexandre Janot. *Contribution à la modélisation et à l'identification des interfaces haptiques*. PhD thesis, Nantes, 2007. (Cited on page 30.)
- [Janot 2017] Alexandre Janot. *On the identification of continuous-time inverse dynamic model of electromechanical systems operating in closed loop with an instrumental variable approach: application to industrial robots*. Technical report, ONERA, 2017. (Cited on page 61.)
- [Jategaonkar 2006] Ravindra Jategaonkar. Flight vehicle system identification: a time domain methodology, volume 216. American Institute of Aeronautics and Astronautics Reston, Va, USA, 2006. (Cited on page 42.)
- [Kay 1993] Steven M. Kay. Fundamentals of statistical signal processing, volume i: Estimation theory. Prentice Hall, 1993. (Cited on page 150.)
- [Khalil & Dombre 2004] Wisama Khalil and Etienne Dombre. Modeling, identification and control of robots. Butterworth-Heinemann, 2004. (Cited on pages 9, 13, 17, 23, 77, 83, 131, 135, 145, 168 and 172.)
- [Khalil & Kleinfinger 1986] Wisama Khalil and J. Kleinfinger. *A new geometric notation for open and closed-loop robots*. In IEEE International Conference on Robotics and Automation., volume 3, pages 1174–1179. IEEE, 1986. (Cited on pages 6, 13 and 158.)
- [Khalil *et al.* 2007] Wisama Khalil, Maxime Gautier and Philippe Lemoine. *Identification of the payload inertial parameters of industrial manipulators*. In IEEE International Conference on Robotics and Automation (ICRA), pages 4943–4948. IEEE, 2007. (Cited on page 143.)
- [Klein & Morelli 2006] Vladislav Klein and Eugene A. Morelli. Aircraft system identification: theory and practice. American Institute of Aeronautics and Astronautics Reston, Va, USA, 2006. (Cited on pages 42, 72 and 166.)
- [Klein 1989] Vladislav Klein. *Estimation of aircraft aerodynamic parameters from flight data*. Progress in Aerospace Sciences, vol. 26, no. 1, pages 1–77, 1989. (Cited on page 42.)

- [Kostic *et al.* 2004] Dragan Kostic, Bram De Jager, Maarten Steinbuch and Ron Hensen. *Modeling and identification for high-performance robot control: An RRR-robotic arm case study*. IEEE Transactions on Control Systems Technology, vol. 12, no. 6, pages 904–919, 2004. (Cited on pages 7, 125, 158 and 183.)
- [Landau & Horowitz 1988] Ioan-Doré Landau and Roberto Horowitz. *Synthesis of adaptive controllers for robot manipulators using a passive feedback systems approach*. In IEEE International Conference on Robotics and Automation, pages 1028–1033. IEEE, 1988. (Cited on page 20.)
- [Landau *et al.* 1999] ID Landau, BDO Anderson and F De Bruyne. *Closed-loop output error identification algorithms for nonlinear plants*. In Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, volume 1, pages 606–611. IEEE, 1999. (Cited on pages 7 and 159.)
- [Landau *et al.* 2001] ID Landau, Brian DO Anderson and Franky De Bruyne. *Recursive identification algorithms for continuous-time nonlinear plants operating in closed loop*. Automatica, vol. 37, no. 3, pages 469–475, 2001. (Cited on pages 42 and 43.)
- [Lawson & Hanson 1974] C.L. Lawson and R.J. Hanson. Solving least squares problems. Englewood Cliffs: Prentice-Hall, 1974. (Cited on page 53.)
- [Levant 1998] Arie Levant. *Robust exact differentiation via sliding mode technique*. automatica, vol. 34, no. 3, pages 379–384, 1998. (Cited on pages 72 and 166.)
- [Lischinsky *et al.* 1999] P Lischinsky, C Canudas-de Wit and G Morel. *Friction compensation for an industrial hydraulic robot*. IEEE Control Systems Magazine, vol. 19, no. 1, pages 25–32, 1999. (Cited on page 17.)
- [Ljung 1976] Lennart Ljung. *On the consistency of prediction error identification methods*. Mathematics in Science and Engineering, vol. 126, pages 121–164, 1976. (Cited on pages 7, 40 and 159.)
- [Ljung 1979] Lennart Ljung. *Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems*. IEEE Transactions on Automatic Control, vol. 24, no. 1, pages 36–50, 1979. (Cited on pages 72 and 166.)
- [Ljung 1988] Lennart Ljung. *System identification toolbox*. The Matlab user’s guide, 1988. (Cited on pages 93 and 173.)
- [Ljung 1999] Lennart Ljung. *System identification: theory for the user*. PTR Prentice Hall, Upper Saddle River, NJ, 1999. (Cited on pages 9, 25, 30, 33, 39, 41, 83, 93, 104, 172 and 173.)
- [Luenberger 1971] David Luenberger. *An introduction to observers*. IEEE Transactions on automatic control, vol. 16, no. 6, pages 596–602, 1971. (Cited on pages 72 and 166.)

- [Marcassus *et al.* 2007] Nicolas Marcassus, Pierre-Olivier Vandanjon, Alexandre Janot and Maxime Gautier. *Minimal resolution needed for an accurate parametric identification-application to an industrial robot arm*. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2455–2460. IEEE, 2007. (Cited on pages 23 and 113.)
- [Murray-Smith 1995] David Murray-Smith. *Continuous system simulation*. Springer Science & Business Media, 1995. (Cited on page 18.)
- [Nelder & Mead 1965] John A Nelder and Roger Mead. *A simplex method for function minimization*. The computer journal, vol. 7, no. 4, pages 308–313, 1965. (Cited on page 44.)
- [Nicosia & Tomei 1990] S. Nicosia and P. Tomei. *Robot control by using only joint position measurements*. IEEE Transactions on Automatic Control, vol. 35, no. 9, pages 1058–1061, 1990. (Cited on page 20.)
- [Noël & Schoukens 2017] J.P. Noël and J. Schoukens. *Grey-box state-space identification of nonlinear mechanical vibrations*. International Journal of Control, pages 1–22, 2017. (Cited on pages 128 and 185.)
- [Norton 1975] J.P. Norton. *Optimal smoothing in the identification of linear time-varying systems*. In Proceedings of the Institution of Electrical Engineers, volume 122, pages 663–668. IET, 1975. (Cited on pages 73 and 167.)
- [Olsen & Petersen 2001] Martin M Olsen and Henrik Gordon Petersen. *A new method for estimating parameters of a dynamic robot model*. IEEE Transactions on Robotics and Automation, vol. 17, no. 1, pages 95–100, 2001. (Cited on pages 7 and 158.)
- [Olsen *et al.* 2002] Martin M Olsen, Jan Swevers and Walter Verdonck. *Maximum likelihood identification of a dynamic robot model: Implementation issues*. The international Journal of robotics research, vol. 21, no. 2, pages 89–96, 2002. (Cited on pages 7 and 158.)
- [Pasch & Seering 1984] K. A. Pasch and W. P. Seering. *On the Drive Systems for High-Performance Machines*. Journal of Mechanisms Transmissions and Automation in Design, vol. 106, 1984. (Cited on page 21.)
- [Pascu *et al.* 2016] Valentin Pascu, Hugues Garnier, Lennart Ljung and Alexandre Janot. *Developments towards formalizing a benchmark for continuous-time model identification*. In 55th IEEE Conference on Decision and Control (CDC), pages 7171–7176. IEEE, 2016. (Cited on pages 83 and 172.)
- [Pham *et al.* 2001] Minh Tu Pham, Maxime Gautier and Philippe Poignet. *Identification of joint stiffness with bandpass filtering*. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, volume 3, pages 2867–2872. IEEE, 2001. (Cited on page 24.)

- [Pham 2002] Minh Tu Pham. *Contribution à la modélisation, l'identification et la commande de systèmes mécaniques à flexibilités localisées: application à des axes de machines-outils rapides*. PhD thesis, Nantes, 2002. (Cited on page 24.)
- [Pierce 1972] David A. Pierce. *Least squares estimation in dynamic-disturbance time series models*. Biometrika, pages 73–78, 1972. (Cited on page 38.)
- [Pressé & Gautier 1993] C Pressé and Maxime Gautier. *New criteria of exciting trajectories for robot identification*. In Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on, pages 907–912. IEEE, 1993. (Cited on page 54.)
- [Richalet *et al.* 1971] Jacques Richalet, André Rault and R Pouliquen. *Identification des processus par la méthode du modèle*. Gordon & Breach, 1971. (Cited on pages 7, 42 and 159.)
- [Robet *et al.* 2012] P. Ph. Robet, Maxime Gautier, Anthony Jubien and Alexandre Janot. *A new output error method for a decoupled identification of electrical and mechanical dynamic parameters of DC motor-driven robots*. IFAC Proceedings Volumes, vol. 45, no. 22, pages 25–30, 2012. (Cited on page 30.)
- [Samson 1983] Claude Samson. *Problèmes en identification et commande de systlèmes dynamiques*. PhD thesis, Université de Rennes I, 1983. Thèse d'Etat. (Cited on page 20.)
- [Sargan 1958] John D. Sargan. *The estimation of economic relationships using instrumental variables*. Econometrica: Journal of the Econometric Society, pages 393–415, 1958. (Cited on page 112.)
- [Schoukens *et al.* 2012] Johan Schoukens, Rik Pintelon and Yves Rolain. *Mastering system identification in 100 exercises*. John Wiley & Sons, 2012. (Cited on page 41.)
- [Shibata 1976] Ritei Shibata. *Selection of the order of an autoregressive model by Akaike's information criterion*. Biometrika, vol. 63, no. 1, pages 117–126, 1976. (Cited on page 104.)
- [Shtessel *et al.* 2014] Yuri Shtessel, Christopher Edwards, Leonid Fridman and Arie Levant. *Sliding mode control and observation*. Springer, 2014. (Cited on page 73.)
- [Siciliano *et al.* 2010] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010. (Cited on pages 4, 9, 100, 102, 131, 133, 145, 157 and 176.)
- [Sidhom 2011] Lilia Sidhom. *Sur les différentiateurs en temps réel: Algorithmes et Applications*. PhD thesis, INSA de Lyon, 2011. (Cited on pages 72 and 166.)

- [Singer 1970] Robert A Singer. *Estimating optimal tracking filter performance for manned maneuvering targets*. IEEE Transactions on Aerospace and Electronic Systems, no. 4, pages 473–483, 1970. (Cited on pages 72 and 166.)
- [Söderström & Stoica 1983] Torsten Söderström and Petre Stoica. Instrumental variable methods for system identification, volume 57. Springer, 1983. (Cited on pages 26, 27 and 63.)
- [Söderström & Stoica 1988] Torsten Söderström and Petre Stoica. System identification. Prentice-Hall, Inc., 1988. (Cited on pages 9, 27, 30, 34, 39 and 84.)
- [Solo 1979] Victor Solo. *Time series recursions and stochastic approximation*. Bulletin of the Australian Mathematical Society, vol. 20, no. 01, pages 159–160, 1979. (Cited on page 30.)
- [Stäubli Favergues 2015] Stäubli Favergues. *Arm - TX series 40 family*. Stäubli, 02 2015. (Cited on pages 5, 22, 78 and 101.)
- [Swevers *et al.* 2007] Jan Swevers, Walter Verdonck and Joris De Schutter. *Dynamic model identification for industrial robots*. IEEE Control Systems, vol. 27, no. 5, pages 58–71, 2007. (Cited on page 23.)
- [Taylor *et al.* 2007] C. James Taylor, Diego J. Pedregal, Peter C. Young and Wlodek Tych. *Environmental time series analysis and forecasting with the Captain toolbox*. Environmental Modelling & Software, vol. 22, no. 6, pages 797–814, 2007. (Cited on pages 76 and 168.)
- [Unbehauen & Rao 1998] H. Unbehauen and G.P. Rao. *A review of identification in continuous-time systems*. Annual reviews in Control, vol. 22, pages 145–171, 1998. (Cited on page 33.)
- [Urrea & Pascal 2016] Claudio Urrea and José Pascal. *Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot*. Computers & Electrical Engineering, 2016. (Cited on page 23.)
- [Van Den Hof *et al.* 1992] Paul M.J. Van Den Hof, Ruud J.P. Schrama and Okko H. Bosgra. *An indirect method for transfer function estimation from closed loop data*. In Proceedings of the 31st IEEE Conference on Decision and Control,, pages 1702–1706. IEEE, 1992. (Cited on pages 81 and 170.)
- [Van den Hof 1998] Paul M.J. Van den Hof. *Closed-loop issues in system identification*. Annual reviews in control, vol. 22, pages 173–186, 1998. (Cited on pages 25 and 34.)
- [Van Donkelaar & Van den Hof 2000] Edwin T. Van Donkelaar and Paul M.J. Van den Hof. *Analysis of closed-loop identification with a tailor-made parameterization*. European Journal of Control, vol. 6, no. 1, pages 54–62, 2000. (Cited on page 40.)

- [Vandanjon *et al.* 1995] Pierre-Olivier Vandanjon, Maxime Gautier and P. Desbats. *Identification of robots inertial parameters by means of spectrum analysis*. In IEEE International Conference on Robotics and Automation (ICRA), volume 3, pages 3033–3038. IEEE, 1995. (Cited on page 31.)
- [Vasiljevic & Khalil 2006] Luma K Vasiljevic and Hassan K Khalil. *Differentiation with high-gain observers the presence of measurement noise*. In Decision and Control, 2006 45th IEEE Conference on, pages 4717–4722. IEEE, 2006. (Cited on pages 72 and 166.)
- [Walker & Orin 1982] M. W. Walker and D. E. Orin. *Efficient Dynamic Computer Simulation of Robotic Mechanisms*. Journal of Dynamic Systems Measurement and Control, vol. 104, 1982. (Cited on page 19.)
- [Walter & Pronzato 1994] Éric Walter and Luc Pronzato. *Identification de modèles paramétriques à partir de données expérimentales*. Masson, 1994. (Cited on pages 30, 39, 44 and 55.)
- [Warnecke *et al.* Y Nof 1999] H.-J. Warnecke, R.D. Schraft, M. Hagele, O. Barth and G. Schmierer. Manipulator design, in handbook of industrial robotics. John Wiley & Sons, Inc., Second Edition, (ed S. Y. Nof) 1999. (Cited on pages 21 and 22.)
- [Wernholt 2007] Erik Wernholt. *Multivariable frequency-domain identification of industrial robots*. PhD thesis, Institutionen för systemteknik, 2007. (Cited on pages 4, 113 and 157.)
- [Wooldridge 2008] JM Wooldridge. *Introductory econometrics: A modern approach*. South-Western, fourth Edition, 2008. (Cited on page 27.)
- [Xi 1995] Fengfeng Xi. *Effect of non-geometric errors on manipulator inertial calibration*. In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on, volume 2, pages 1808–1813. IEEE, 1995. (Cited on pages 7 and 158.)
- [Young & Jakeman 1979] Peter C. Young and Anthony Jakeman. *Refined instrumental variable methods of recursive time-series analysis Part I. Single input, single output systems*. International Journal of Control, vol. 29, no. 1, pages 1–30, 1979. (Cited on page 34.)
- [Young & Jakeman 1980] Peter C. Young and Anthony Jakeman. *Refined instrumental variable methods of recursive time-series analysis Part III. Extensions*. International Journal of Control, vol. 31, no. 4, pages 741–764, 1980. (Cited on pages 7 and 159.)
- [Young *et al.* 2009] Peter C. Young, Hugues Garnier and Marion Gilson. *Simple refined IV methods of closed-loop system identification*. In 15th IFAC Symposium on System Identification (SYSID), 2009. (Cited on pages 36, 81 and 170.)

- 
- [Young 1966] Peter C Young. *Process parameter estimation and self adaptive control*. In Theory of Self-Adaptive Control Systems, pages 118–140. Springer, 1966. (Cited on page 33.)
- [Young 1981] Peter C. Young. *Parameter estimation for continuous-time models – a survey*. Automatica, vol. 17, no. 1, pages 23–39, 1981. (Cited on page 33.)
- [Young 1999] Peter C Young. *Nonstationary time series analysis and forecasting*. Progress in Environmental Science, vol. 1, pages 3–48, 1999. (Cited on pages 128 and 185.)
- [Young 2000] Peter C. Young. *Stochastic, dynamic modelling and signal processing: time variable and state dependent parameter estimation*. Nonlinear and nonstationary signal processing, pages 74–114, 2000. (Cited on pages 73 and 167.)
- [Young 2006] Peter C. Young. *An instrumental variable approach to arma model identification and estimation*. In 14th IFAC Symposium on System Identification (SYSID), pages 410 – 415, Newcastle, Australia, 2006. (Cited on pages 37, 103 and 177.)
- [Young 2011] Peter C. Young. Recursive estimation and time-series analysis: an introduction for the student and practitioner. Springer Berlin Heidelberg, 2nd Edition, 2011. (Cited on pages 9, 27, 34, 36, 37, 38, 44, 65, 73, 74, 77, 83, 167, 168 and 172.)
- [Young 2015] Peter C. Young. *Refined instrumental variable estimation: maximum likelihood optimization of a unified Box–Jenkins model*. Automatica, vol. 52, pages 35–46, 2015. (Cited on page 34.)



# Index

- A**
- Accuracy ..... 98
  - Auxiliary model..... 27, 59
- B**
- Basis function ..... *see* Regressor
- C**
- Controller ..... 20, 49, 79
- D**
- DIDIM ..... 29, 59
  - Drive gains ..... 21
- E**
- Encoder resolution..... 23
- F**
- Filter
    - bandpass ..... 24
    - decimate ..... 24, 46, 53, 66
  - Frame ..... 14
  - Friction model..... 17
- I**
- Independant variable ..... *see* Regressor
  - Inertia
    - matrix..... 16, 132
    - tensor ..... 17, 129
  - Instrument
    - exogeneity ..... 27, 36
    - relevance ..... 27
  - Instrumental variable
    - estimate ..... 28, 59
    - extended ..... 26
- J**
- Joint ..... 14
- L**
- Least-Squares
    - ordinary ..... 26, 67
    - weighted ..... 26
- Link** ..... 14
- Center of gravity ..... 17, 129
  - Origin ..... 17, 129
- M**
- Mass ..... 17
  - Maximum Likelihood..... 37, 73, 102
  - Modified Denavit-Hartenberg .... 14, 98
- N**
- Noise
    - measurement ..... 22, 100
    - position ..... 50, 97
    - torque..... 23, 50, 97, 100
  - Noise variance ratio ..... 72
- P**
- Parallel filter..... *see* Filter decimate
  - Parameters
    - base..... 18
    - standard..... 17, 130
  - Persistent excitation ..... 30, 126
  - Prediction error method
    - one-step-ahead ..... 38
- R**
- Regressor ..... 18, 60
  - Repeatability ..... 4, 98
- S**
- Sensitivity ..... 41, 46, 47, 54
  - Solver
    - integration ..... 40
    - optimisation..... 40
  - Stiff system ..... 18, 40
- T**
- Torque error..... *see* Noise torque

## Identification of rigid industrial robots – A system identification perspective

**Abstract** In modern manufacturing, industrial robots are essential components that allow saving cost, increase quality and productivity for instance. To achieve such goals, high accuracy and speed are simultaneously required. The design of control laws compliant with such requirements demands high-fidelity mathematical models of those robots. For this purpose, dynamic models are built from experimental data. The main objective of this thesis is thus to provide robotic engineers with automatic tools for identifying dynamic models of industrial robot arms. To achieve this aim, a comparative analysis of the existing methods dealing with robot identification is made. That allows discerning the advantages and the limitations of each method. From those observations, contributions are presented on three axes. First, the study focuses on the estimation of the joint velocities and accelerations from the measured position, which is required for the model construction. The usual method is based on a home-made prefiltering process that needs a reliable knowledge of the system's bandwidths, whereas the system is still unknown. To overcome this dilemma, we propose a method able to estimate the joint derivatives automatically, without any setting from the user. The second axis is dedicated to the identification of the controller. For the vast majority of the method its knowledge is indeed required. Unfortunately, for copyright reasons, that is not always available to the user. To deal with this issue, two methods are suggested. Their basic philosophy is to identify the control law in a first step before identifying the dynamic model of the robot in a second one. The first method consists in identifying the control law in a parametric way, whereas the second one relies on a non-parametric identification. Finally, the third axis deals with the home-made setting of the decimate filter. The identification of the noise filter is introduced similarly to methods developed in the system identification community. This allows estimating automatically the dynamic parameters with low covariance and it brings some information about the noise circulation through the closed-loop system. All the proposed methodologies are validated on an industrial robot with 6 degrees of freedom. Perspectives are outlined for future developments on robotic systems identification and other complex problems.

*Keywords: system identification, robot identification, prediction error methods, instrumental variable, output error methods*

## Identification de robots industriels rigides – Apport des méthodes de l'identification de systèmes

**Résumé** L'industrie moderne fait largement appel à des robots industriels afin de réduire les coûts, ou encore améliorer la productivité et la qualité par exemple. Pour ce faire, une haute précision et une grande vitesse sont simultanément nécessaires. La conception de lois de commande conformes à de telles exigences demande une modélisation mathématique précise de ces robots. A cette fin, des modèles dynamiques sont construits à partir de données expérimentales. L'objectif de cette thèse est ainsi de fournir aux ingénieurs roboticiens des outils automatiques pour l'identification de bras robotiques. Dans cette perspective, une analyse comparative des méthodes existantes pour l'identification de robot est réalisée. Les avantages et inconvénients de chaque méthode sont ainsi mis en exergue. À partir de ces observations, les contributions sont articulées selon trois axes. Premièrement, l'étude porte sur l'estimation des vitesses et accélérations des corps du robot à partir de la position mesurée. Ces informations sont en effet nécessaires à la construction du modèle. La méthode usuelle est basée sur prétraitement "sur mesure" qui requière une connaissance fiable des bande-passantes du système, alors que celui-ci est encore inconnu. Pour surmonter ce dilemme, nous proposons une méthode capable d'estimer les dérivées automatiquement sans réglage préalable par l'utilisateur. Le deuxième axe concerne l'identification du contrôleur. Sa connaissance est en effet requise par la grande majorité des méthodes d'identification. Malheureusement, pour des raisons de propriété industrielle, il n'est pas toujours accessible. Pour traiter ce problème, deux méthodes sont introduites. Leur principe de base est d'identifier la loi de commande dans un premier temps avant d'identifier le modèle dynamique du bras robotique dans un second temps. La première méthode consiste à identifier la loi de commande de manière paramétrique, alors que la seconde fait appel à une identification non-paramétrique. Finalement, le troisième axe porte sur le réglage "sur mesure" du filtre decimate. L'identification du filtre de bruit est introduite en s'inspirant des méthodes développées par la communauté d'identification de systèmes. Ceci permet l'estimation automatique des paramètres dynamiques avec de faibles covariances tout en apportant une connaissance concernant la circulation du bruit à travers le système en boucle-fermée. Toutes les méthodes proposées sont validées sur un robot industriel à six degrés de liberté. Des perspectives sont esquissées pour de futurs travaux portant sur l'identification de systèmes robotiques, voire d'autres applications.

*Mots-clés: identification de systèmes, identification de robots, méthodes d'erreur de prédiction, variable instrumentale, méthodes d'erreur de sortie*