Online Interaction-Based Property Estimation of Objects

Matteo Ciocca
Online Interaction-Based Property Estimation of Objects

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Matteo Ciocca

March 18, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology
The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

**Online Interaction-Based Property Estimation of Objects**

by

**Matteo Ciocca**

in partial fulfillment of the requirements for the degree of

**Master of Science Systems and Control**

Dated: March 18, 2016

**Supervisor(s):**

__________________________
dr. G.A.D. Lopes

**Reader(s):**

__________________________
prof.dr. R. Babuška

__________________________
ir. K.A.J. Verbert

__________________________
prof.dr.ir. H. van der Kooij
The goal of this thesis is to develop online model-based parameter estimation techniques for robotics towards extracting geometrical or physical properties of an unknown (or partially unknown) dynamic environment through a tactile interaction with it.

The thesis focuses on a one-dimensional case of a rigid movable object, not yet studied in the literature with parametric model-based techniques. The aim is to estimate through a tactile interaction the mass of the object and the friction between the object and the (unknown) surface where it is placed. The thesis focus on the use of Recursive Least-Squares algorithms. An explicit derivation, in a matrix form, of these algorithm is presented. The persistent excitation problem is taken under analysis since it is one of the key element for a correct parameter estimation.

The thesis brings under study a novel simulated experiment where a robot finger pushes and excites the dynamics of a rigid movable mass in order to estimate the properties listed above. Because of the mobility of the object, signal discontinuities (during the pushing action) are the challenges of the estimation strategy. Possible solutions are implemented and results are discussed.
## Contents

Reflections on Robotics .................................................. xiii
Acknowledgements .......................................................... xvii

1 Introduction ................................................................. 1
1-1 Interactive Tasks in Robotics ....................................... 1
1-2 Motivation ............................................................... 2
1-3 Research Question ..................................................... 4
1-4 Thesis Overview ....................................................... 4
1-5 Outline of the Thesis .................................................. 5

2 Background on Model-Based Parameter Estimation Techniques .... 7
2-1 Environment models .................................................. 7
  2-1-1 Compliant Surface Models .................................... 10
  2-1-2 Rigid Movable Object Models ............................... 11
2-2 Online Model-Based Parameter Estimation Techniques .......... 12
  2-2-1 Parametric System Identification ............................ 13
  2-2-2 Adaptive Control ................................................ 16
  2-2-3 Adaptive Observer ................................................ 18
2-3 Conclusions ............................................................. 20

3 Identification Framework .................................................. 23
3-1 Recursive Least-Squares Framework ................................ 23
  3-1-1 Recursive Least-Squares Algorithms with Forgetting factor: \( RLS_\kappa \) ............................................. 27
  3-1-2 Parameter estimation convergence for \( RLS_\kappa \) ......... 30
3-2 Convergence properties for LTI systems ........................ 33
3-3 Mention of Subspace identification and offline algorithms .... 34
3-4 Conclusions ............................................................. 35

Master of Science Thesis ............................................... Matteo Ciocca
4 Model
4-1 Introduction .................................................. 37
4-2 Object Model ................................................. 39
  4-2-1 Viscous Friction coefficient and Filter Design ............... 39
  4-2-2 From a Viscous friction model to a Static friction model ....... 43
4-3 Test with Parameter-Estimation Algorithms ...................... 45
  4-3-1 Recursive Least-Squares techinques ........................ 48
4-4 The Problem of Persistent Excitation (PE) ....................... 55
4-5 Conclusions .................................................. 60

5 Experiment .................................................... 61
5-1 Experiment Description ........................................ 61
  5-1-1 How the experiment works .................................. 62
5-2 Hybrid Architecture, Trajectory and Controller Selection ....... 64
5-3 Pushing finger Experiment ..................................... 71
5-4 Discontinuities Problem ....................................... 84
5-5 Time-varying parameters ....................................... 92
  5-5-1 A different approach ...................................... 92
5-6 Conclusions .................................................. 103

6 Experimental Setup and Results ................................ 105
6-1 Introduction .................................................. 105
6-2 Relevant informations on the platform and object ............... 106
6-3 Understanding the signals ...................................... 107
6-4 Experiment Results .......................................... 111
6-5 Conclusions .................................................. 120

7 Conclusions .................................................... 121
7-1 Thesis Results ................................................. 121
7-2 Thesis Contributions .......................................... 123
7-3 Perspectives and Future Work .................................. 124

A Additional Simulations and Experiments ......................... 127
A-1 Filter and Controller design ................................... 127
  A-1-1 Filter Design ............................................ 128
  A-1-2 Impedance Controller Design .............................. 129
A-2 Third and Wood object for viscous friction coefficient ......... 131
A-3 Third and Steel object for Coulomb friction .................... 137
A-4 Estimation with only a polynomial trajectory .................. 143
### B Mistakes in the Literature

- **B-1 Adaptive regulation of impact induced forces for three degree of freedom collisions: a backstepping approach [98]**
  - Design procedure: Adaptive Backstepping
  - Summary of the Adaptive Backstepping control law

- **B-2 Adaptive Regulation of Impact Induced Forces for One-Degree-of-Freedom Collisions [45]**

### Bibliography
List of Figures

2-1 Rigid Collision ................................................................. 7
2-2 Compliant Collision .......................................................... 8
2-3 Compliant surface model (in this case Kelvin-Voigt model) ............ 10
2-4 Rigid Movable Object Model .................................................. 11
2-5 Euler-Venn Diagrams on the distinction between Parametric System Identification and Adaptive Control .................................................. 13
2-6 Parametric System Identification Standard Scheme ........................................ 14
2-7 PSI Scheme to Extract Environment properties ................................... 15
2-8 Model Reference Adaptive Control Block Scheme .................................. 16
2-9 Two different Adaptive Control Schemes ........................................ 17
2-10 Variable Impedance Filter scheme [21] ........................................ 18
2-11 \([K_{sc1}, K_{sc2}]\) two stiffness candidates, \(\Delta x_{c,k} = \Delta x_c(k)\) for error between the force sensor and the output of the observer. \(\hat{K}_{s,k} = \hat{K}_s(k)\) final estimation of environment stiffness ........................................ 19

4-1 Rigid Movable Object Model .................................................. 37
4-2 Simulation using the gradient law technique. Estimation for the object model: mass and viscous friction. Experiment-1, wood object. ....................... 47
4-3 Experiment-1, third object. Simulation using the Continuous RLS\textsubscript{Cont} and the discrete version RLS, both with a forgetting factor \(\varkappa = 0.99\). The Figure shows the force and velocity profile and below the evolution of the covariance matrix, \(P\), initialized for both algorithms with: \(P(0) = \alpha I, \alpha = 100\). ........................................ 51
4-4 Experiment-1, third object. Simulation using the Continuous RLS\textsubscript{Cont} and the discrete version RLS, both with a forgetting factor \(\varkappa = 0.99\). The Figure shows the estimation of the \(\hat{\Theta}\) vector, for both the algorithms. ........................................ 52
4-5 Experiment-2, steel object. Simulation using the Continuous RLS\textsubscript{Cont} and the discrete version RLS, both with a forgetting factor \(\varkappa = 0.99\). The Figure shows the force and velocity profile and below the estimation of the \(\hat{\Theta}\) vector, for both the algorithms. ........................................ 53
4-6 Experiment-2, steel object. Simulation using the Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS$, both with a forgetting factor $\alpha = 0.99$. The Figure shows the evolution of the covariance matrix, $P$, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 100$. .......................................................... 54

4-7 Simulation for the PE problem. Force input is a single unit step. Two cases: above the wood object, below the third object. .......................................................... 57

4-8 Simulation for the wood cube using the continuous version of $RLS_{\text{Cont}}$. Force input is a single unit step. The trace of covariance matrix $P(t)$ is unbounded, but the estimates converge to their nominal/real value. The single unit force step was persistent excited enough for the wood cube model. This is the third case in the logic table .......................................................... 58

4-9 Simulation for the third object using the continuous version of $RLS_{\text{Cont}}$. Force input is a single unit step. The trace of covariance matrix $P(t)$ is unbounded and the estimates do not converge to their nominal/real value. The single unit force step was NOT persistent excited enough for the third object. This is the fourth case in the logic table .......................................................... 59

5-1 One-finger-pushing experiment. The reference frame $\Psi_0$ on the left. The positive movement direction to the right. The position of the finger $x$ and the position of the rigid movable object $y$. When the robotic finger and the object pass into the contact phase, a force sensor (spring and damper in parallel) is added to generate a force signal $F$. .......................................................... 64

5-2 The polynomial trajectory. The generation of the trajectory (position, velocity and acceleration) was achieved by choosing $\Delta S_d = 3 [\text{dm}]$ and $t_f = 20 [\text{s}]$. The plots show the simple polynomial trajectory and the same trajectory with the addition of a selected sinusoidal: $S^1_d(t)$ and $S^2_d(t)$ .......................................................... 68

5-3 Experiment-1, steel object. Relevant measurements (above) of the robot finger ($\cdot$)$_r$, the object ($\cdot$)$_o$, and desired trajectory ($\cdot$)$_d$: position, velocities. Filtered signals and control input (below) .......................................................... 73

5-4 Experiment-1, steel object. Estimation of the $\Theta$ vector and the error function $\epsilon$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\text{Cont}}$ algorithms, both with a forgetting factor $\alpha = 0.99$. .......................................................... 74

5-5 Experiment-1, steel object. Evolution of the trace of the covariance matrix (ZOOM-IN: plots on the right), $P$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\text{Cont}}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 100$. .......................................................... 75

5-6 Experiment-2, wood object. Relevant measurements (above) of the robot finger ($\cdot$)$_r$, the object ($\cdot$)$_o$, and desired trajectory ($\cdot$)$_d$: position, velocities. Filtered signals and control input (below) .......................................................... 76

5-7 Experiment-2, wood object. Estimation of the $\Theta$ vector and the error function $\epsilon$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS$ algorithms, both with a forgetting factor $\alpha = 0.99$. .......................................................... 77

5-8 Experiment-2, wood object. Evolution of the trace of the covariance matrix, $P$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\text{Cont}}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 0.01$. .......................................................... 78

5-9 Experiment-1, steel object. Closer look at the estimation of the $\Theta$ vector .......................................................... 79

5-10 Experiment-1, steel object. Estimation of the $\Theta$ vector and the error function $\epsilon$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\text{Cont}}$ algorithms, both with a forgetting factor $\alpha = 0.99$. .......................................................... 87
5-11 Experiment-1, steel object. Evolution of the trace of the covariance matrix, $\mathcal{P}$, for Continuous $\text{RLS}_{\text{Cont}}$ and the discrete version $\text{RLS}_{\text{dis}}$ algorithms, initialized for both algorithms with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. .................................................. 88

5-12 Experiment-2, wood object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for Continuous $\text{RLS}_{\text{Cont}}$ and the discrete version $\text{RLS}$ algorithms, both with a forgetting factor $\zeta = 0.99$. .................................................. 89

5-13 Experiment-2, wood object. Evolution of the trace of the covariance matrix, $\mathcal{P}$, for Continuous $\text{RLS}_{\text{Cont}}$ and the discrete version $\text{RLS}_{\text{dis}}$ algorithms, initialized for both algorithms with: $\mathcal{P}(0) = \alpha I$, $\alpha = 0.01$. .................................................. 90

5-14 Different modeling prospective for a tactile interaction, in the case of a rigid movable object. .................................................. 92

5-15 Experiment-1, steel object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ - discrete version $\text{RLS}$ algorithms, forgetting factor $\zeta = 0.75$. .................................................. 95

5-16 Experiment-1, steel object. Above, the signal filtered: $(F =)u$ and velocity of the robot $(\cdot)_v$. Below, the evolution of the trace of the covariance matrix, $\mathcal{P}$ - discrete version $\text{RLS}$ algorithm, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. .................................................. 96

5-17 Experiment-2, wood object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ - discrete version $\text{RLS}_{\text{dis}}$ algorithm, forgetting factor $\zeta = 0.99$. .................................................. 97

5-18 Experiment-2, wood object. Evolution of the trace of the covariance matrix, $\mathcal{P}$ - discrete version $\text{RLS}_{\text{dis}}$ algorithm, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. .................................................. 98

5-19 Experiment-1, steel object, changed robot mass $m_{\text{robot}} = 1 \ [Kg]$. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ - discrete version $\text{RLS}$ algorithm, forgetting factor $\zeta = 0.75$. .................................................. 99

5-20 Experiment-1, steel object, changed robot mass $m_{\text{robot}} = 1 \ [Kg]$. Evolution of the trace of the covariance matrix, $\mathcal{P}$ - discrete version $\text{RLS}_{\text{dis}}$ algorithm, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. .................................................. 100

6-1 Robot and object (left). Model of the real-time experiment (centre). Object (right). .................................................. 106

6-2 Robot in a stand-still position to understand some possible offset of the signals. .................................................. 108

6-3 Robot following a polynomial trajectory. The figure shows the position in free motion, $(\cdot)^p$, and the position pushing the object $(\cdot)^p$. Below, the force profiles and the force inaccuracy. .................................................. 109

6-4 Free motion. Robot following a polynomial trajectory with the addition of a sinusoidal: $A = 10 \ [mm]$, $w = 4.5 \ [rad/s]$. .................................................. 110

6-5 Experiment-(a). Above, the matrix enrices of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version $\text{RLS}_{\text{dis}}$ algorithms, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 0.01$. .................................................. 112

6-6 Experiment-(a). Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, discrete version $\text{RLS}_{\text{dis}}$ algorithms, with a forgetting factor $\zeta = 0.99$. .................................................. 113

6-7 Experiment-(a). Force and velocity signal: real and filtered. Below the three components of the regressor vector. .................................................. 114

6-8 Experiment-(b). Above, the matrix enrices of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version $\text{RLS}_{\text{dis}}$ algorithms, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 0.01$. .................................................. 115

6-9 Experiment-(b). Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, discrete version $\text{RLS}_{\text{dis}}$ algorithms, with a forgetting factor $\zeta = 0.99$. .................................................. 116
6-10 Experiment-(b). Force and velocity signal: real and filtered. Below the three components of the regressor vector. .................................................. 117
6-11 Experiment (a)-(b). Closer look at the contact point between the robot and the object (left). Distance between the robot end-effector and the table (right). . . . 118
7-1 The two fingers represented by two actuated \((u)\) masses \((M)\) and the object \((m)\) that can slide in two directions. ................................................................. 124
7-2 Compliant surface model (above) and Rigid Movable Object Model (below). . . . 124
A-1 Above: Bode plots, step response and tracking of a sinusoidal \(u = \sin(10t)\), frequency \(f = 10/2\pi = 1.59\) [Hz] Below: Bode plots, poles and zeros map and step response. \(G(\cdot)\) representing the mass-spring-damper transfer function for the \((\cdot)\) Experiment. ................................................................. 130
A-2 Experiment-1, third object. Relevant measurements (above) of the robot finger \((\cdot)_{r}\), the object \((\cdot)_{o}\) and desired trajectory \((\cdot)_{d}\): position, velocities. Filtered signals and control input (below). ................................................................. 132
A-3 Experiment-1, wood object. Relevant measurements (above) of the robot finger \((\cdot)_{r}\), the object \((\cdot)_{o}\) and desired trajectory \((\cdot)_{d}\): position, velocities. Filtered signals and control input (below). ................................................................. 133
A-4 Experiment-1, third object. Estimation of the \(\hat{\Theta}\) vector and the error function \(\epsilon\), for the discrete version \(RLS_{\kappa}\) algorithm, with a forgetting factor \(\kappa = 0.85\). . . . 134
A-5 Experiment-1, third object. Estimation of the \(\hat{\Theta}\) vector and the error function \(\epsilon\), for the discrete version \(RLS_{\kappa}\) algorithm, with a forgetting factor \(\kappa = 0.85\). . . . 135
A-6 Experiment-1, third object (above), wood object (below). Evolution of the trace of the covariance matrix, \(P\), for the discrete version \(RLS_{\kappa}\) algorithm, initialized for both algorithms with: \(P(0) = \alpha I, \alpha = 100\). ................................................................. 136
A-7 Experiment-2, third object. Relevant measurements (above) of the robot finger \((\cdot)_{r}\), the object \((\cdot)_{o}\) and desired trajectory \((\cdot)_{d}\): position, velocities. Filtered signals and control input (below). ................................................................. 138
A-8 Experiment-2, steel object. Relevant measurements (above) of the robot finger \((\cdot)_{r}\), the object \((\cdot)_{o}\) and desired trajectory \((\cdot)_{d}\): position, velocities. Filtered signals and control input (below). ................................................................. 139
A-9 Experiment-2, third object. Estimation of the \(\hat{\Theta}\) vector and the error function \(\epsilon\), for the discrete version \(RLS_{\kappa}\) algorithm, with a forgetting factor \(\kappa = 0.85\). . . . 140
A-10 Experiment-2, steel object. Estimation of the \(\hat{\Theta}\) vector and the error function \(\epsilon\), for the discrete version \(RLS_{\kappa}\) algorithm, with a forgetting factor \(\kappa = 0.85\). . . . 141
A-11 Experiment-2, third object (above), steel object (below). Evolution of the trace of the covariance matrix, \(P\), for the discrete version \(RLS_{\kappa}\) algorithm, initialized for both algorithms with: \(P(0) = \alpha I, \alpha = 100\). ................................................................. 142
A-12 Experiment-(b). Above, the matrix entries of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version \(RLS_{\kappa}\) algorithms, initialized with: \(P(0) = \alpha I, \alpha = 0.01\). ................................................................. 144
A-13 Experiment-(b). Estimation of the \(\hat{\Theta}\) vector and the error function \(\epsilon\), discrete version \(RLS_{\kappa}\) algorithms, with a forgetting factor \(\kappa = 0.99\). . . . . . . . . . . . . . 145
A-14 Experiment-(b). Force and velocity signal: real and filtered. Below the three components of the regressor vector. ................................................................. 146
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>Force input signal and Filter parameters for Experiment-1 and Experiment-2</td>
<td>45</td>
</tr>
<tr>
<td>4-2</td>
<td>Experiment-1: object model parameters and nominal $\Theta$ vector</td>
<td>46</td>
</tr>
<tr>
<td>4-3</td>
<td>Experiment-2: object model parameters and nominal $\Theta$ vector - composed with the Coulomb friction force</td>
<td>46</td>
</tr>
<tr>
<td>4-4</td>
<td>Test Set-up, Parameters for gradient law</td>
<td>46</td>
</tr>
<tr>
<td>4-5</td>
<td>Experiment-1: (third) object model parameters and nominal $\Theta$ vector</td>
<td>50</td>
</tr>
<tr>
<td>4-6</td>
<td>Experiment-2: (steel) object model parameters and nominal $\Theta$ vector - composed with the Coulomb friction force</td>
<td>50</td>
</tr>
<tr>
<td>4-7</td>
<td>Given the boundedness condition on $\Phi$, called $C$. The PE of $\Phi$, called $A$, is a necessary condition for $P$ to be bounded, called $B$. Highlighted in red are the conditions where I would like not to be. Because if $B$ is not satisfied, I do not know if $A$ is correct or not: because in both cases (when $A$ is true or false), the necessary condition is satisfied (true).</td>
<td>56</td>
</tr>
<tr>
<td>5-1</td>
<td>Parameter of the sinusoidal</td>
<td>66</td>
</tr>
<tr>
<td>5-2</td>
<td>Filter and Controller Parameters</td>
<td>72</td>
</tr>
<tr>
<td>5-3</td>
<td>Pushing Finger, Experiment-1: (steel) object model parameters and nominal $\Theta$ vector. $T_p = 2$ [s], $\delta = 10^{-2}$</td>
<td>83</td>
</tr>
<tr>
<td>5-4</td>
<td>Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 10$ [s], $\delta = 10^{-3}$</td>
<td>83</td>
</tr>
<tr>
<td>5-5</td>
<td>Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 15$ [s], $\delta = 10^{-3}$</td>
<td>83</td>
</tr>
<tr>
<td>5-6</td>
<td>Pushing Finger (switching law), Experiment-1: (steel) object model parameters and nominal $\Theta$ vector</td>
<td>91</td>
</tr>
<tr>
<td>5-7</td>
<td>Filter and Controller Parameters</td>
<td>94</td>
</tr>
<tr>
<td>5-8</td>
<td>Pushing Finger ($m_{\text{robot}} = 1$ [Kg]), Experiment-1: (steel) object model parameters and nominal $\Theta$ vector. $T_p = 1$ [s], $\delta_{\theta_1} = 10^{-5}$ and $\delta_{\theta_2} = 10^{-3}$</td>
<td>102</td>
</tr>
<tr>
<td>5-9</td>
<td>Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 15$ [s], $\delta = 10^{-3}$</td>
<td>102</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6-1</td>
<td>Selected trajectory Parameters: polynomial+sinusoidal, and Controller parameters</td>
<td>107</td>
</tr>
<tr>
<td>6-2</td>
<td>Filter and Impedance Controller Parameters</td>
<td>111</td>
</tr>
<tr>
<td>6-3</td>
<td>Selected Empirical Parameters</td>
<td>111</td>
</tr>
<tr>
<td>6-4</td>
<td>Nominal (Empirical) Estimates, $\Theta$</td>
<td>111</td>
</tr>
<tr>
<td>A-1</td>
<td>Filter and Controller Parameters</td>
<td>127</td>
</tr>
<tr>
<td>A-2</td>
<td>Filter and Controller Parameters</td>
<td>128</td>
</tr>
<tr>
<td>A-3</td>
<td>Experiment-1: wood and third object models parameters and nominal $\Theta_{\text{touch}}$ vector. $T_p = 1 , [s]$, $\delta_{\theta_1} = 10^{-5}$ and $\delta_{\theta_2} = 10^{-3}$.</td>
<td>137</td>
</tr>
<tr>
<td>A-4</td>
<td>Experiment-2: steel and third object models parameters and nominal $\Theta_{\text{touch}}$ vector. $T_p = 1 , [s]$, $\delta_{\theta_1} = 10^{-5}$, $\delta_{\theta_2} = 10^{-3}$ and $\delta_{\theta_3} = 10^{-3}$.</td>
<td>143</td>
</tr>
</tbody>
</table>
Reflections on Robotics

English

The advancement of technology in the world of robotics allowed to obtain incredible results that have grown exponentially both in qualitative and quantitative terms in industry and academia in the last thirty years. Think about the comparison between the first industrial robotic arm, Unimate [1, 2], made in 1961, with considerable dimensions that could serve a glass of champagne, and humanoid robots specimens, iCub [3] and Asimo [4], appeared in the early twenty-first century. Of course, these cutting-edge platforms can serve a glass of champagne, but they are also capable of walking and thinking. It is amazing how in such small amount of time, in about fifty years, we can see real robots that perform actions similar to those imagined for a distant future: walking, grasping objects, utter words, ask about situations.

However this fascinating evolution is superseded and overshadowed by another idea on robotics, under different aspect. In general, it is popular to think that robotics is synonymous with unemployment. I think this common opinion is wrong. Given the fact that technological development has brought new jobs to the market\(^\text{1}\) and made it disappear other\(^\text{2}\), not always new jobs can be called jobs. They are instead full exploitation of the human resource in industries: labor work in assembly lines (for example for the assembly of components of a computer). The latter are jobs always existed and continue to exist. It is important to realize that these types of jobs can be automated. The solution is called industrial robotics: robots that can operate in daily repetitive tasks for assembling materials.

Robotics should not be seen as cause of massive unemployment and cruel. Robotics should aim to automate these frustrating jobs, labor, with the hope that the abolition of this work is to bring a different organization of industrial production both to discover new intellectual challenges, re-evaluating jobs in the fields of art, philosophical and literary, as well as social and environmental ones.

Furthermore, there is a vision where one day these machines will help us during our daily lives. A vision in which man and (intelligent) machine can coexist. Due to developments

\(^1\)jobs in the field of computer science: software developer, data scientists; work in the field of renewable energy and so on.

\(^2\)the scribe, the lamplighter, the "phone-receptionist", and so on
in the field of robotics, the realization of this vision is no longer a question of how, but when. In fact, prototypes of robots that can coexist with humans already exist: Pepper, Nao and Romeo [5, 6, 7], iCub and Asimo. These prototypes have mechanical and surprising cognitive functions and are now a reality. Therefore, this exponential evolution in robotics is outstanding as a whole, becoming a source of study in various engineering fields (electronic, mechanical and software) in many industries, research laboratories and universities around the world, parallel to the linguistic-cognitive research.

In conclusion, I did not want emphasize deliberately the philosophical aspects of robotics [8] and on artificial intelligence too, that, according to many experts [9] could become a threat in a not so distant future. I wanted to write this preface in order to invite readers to do not look at automatic machines as enemies of society/humanity, but to hope, like me, that these machines possibly become workmates and/or our helpers in difficult situations for us.

Italian


Comunque questa affascinante evoluzione è soppiantata e oscurata da un’altra idea sulla robotica, sotto tutt’altro aspetto. In generale, é popolare pensare che la robotica sia sinonimo di disoccupazione. Secondo me questa comune opinione é sbagliata. Partendo dal fatto che l’evoluzione tecnologica ha portato nuovi lavori sul mercato[3] e ne ha fatto scomparire altri[4], non sempre nuovi lavori possono definirsi lavori. Essi sono invece completo sfruttamento della risorsa umana in industrie: lavori di manodopera nelle catene di montaggio (per esempio per l’assemblaggio di componenti di un computer). Quest’ultimi sono lavori sempre esistenti e continuano ad esistere. É importante però rendersi conto che questi tipi lavori possono essere automatizzati, e la soluzione si chiama robotica industriale: robot capaci di operare giornalmente in operazioni ripetitive per assemblaggio di materiali.

Quindi, la robotica non dovrebbe essere vista come cause di una massiccia e crudele disoccupazione. La robotica dovrebbe mirare a automatizzare questi lavori frustranti, di manodopera, con la speranza che l’abolizione di tali lavori porti sia a una diversa organizzazione della produzione industriale sia a scoprire nuove sfide intellettuali, rivalutando lavori nei campi artistici, filosofici e letterari, come pure in quelli sociali e ambientali.

[3] lavori nel campo della computer science: sviluppatore software, data scientist; lavori nel campo delle energie rinnovabili e così via
[4] lo scrivano, il lampionaio, il centralinista, e così via

Matteo Ciocca Master of Science Thesis
Inoltre, esiste una visione dove un giorno queste macchine automatiche ci aiuteranno durante la nostra vita quotidiana. Una visione nella quale uomo e macchina (intelligente) possono convivere. Data l’evoluzione nel campo della robotica, la realizzazione di questa visione non è più un problema del *come*, ma del *quando*. Infatti, prototipi di robot che potranno convivere con umani già esistono: Pepper, Nao e Romeo [5, 6, 7], iCub e Asimo. Questi prototipi hanno funzioni meccaniche e cognitive sorprendenti e sono oggi realtà. Quindi, questa evoluzione esponenziale nel campo della robotica è eccezionale nel suo complesso, diventando fonte di studio sotto vari campi ingegneristici (elettronica, meccanica e informatica) in molte industrie, laboratori di ricerca e università in tutto il mondo, parallelamente alle ricerche linguistico-cognitive.

Concludendo, non ho volutamente sottolineare aspetti filosofici sulla robotica [8], tanto meno sulla Intelligenza Artificiale che secondo molti esperti [9] potrebbe diventare una minaccia in un futuro non troppo lontano. Ho voluto scrivere questa prefazione con lo scopo di invitare i lettori a non guardare macchine automatiche come nemici della società/l’umanità, ma sperare, come me, che queste macchine possibilmente diventino compagni di lavoro e/o nostri aiutanti in situazioni per noi difficili.
Acknowledgements

I think that *Family is everything* and for this reason I would like first to dedicate *my life* to my parents, my relatives and family members in Italy. I would like to dedicate to them *my past, my present and my future*. Because without them, there would be no reality where I could be here writing this thesis, no causes that could have brought me to live an extraordinary experience here in Delft. *You are my roots, the reason I exist.*

I would like to remember the motto of my grandfather Osvaldo - president of COOPerativa di Consumatori di San Giovanni Valdarno (1980 - 2007): *we can do this, together*. Although what I will say can be hard to understand and probably silly, I think his motto truly embraced the idea of of scientific research and in general, the idea of progress. Because we must do things, whatever they are, together and absolutely not alone. So, in my own small way, I pushed myself to share my thesis problem with my academic department and friends to come up with solutions. Solutions that could be achieved, or at least thought up, *together*.

I would like to thank my supervisor (dr.) Gabriel Lopes for being patient and supportive throughout the entire thesis process, for having left me the freedom to explore all the paths that I wanted to follow. I truly enjoyed carrying out research alongside him and his fresh personality as a young professor.

I would like to thank the secretary of DCSC. In a sentence, *an excellent team: precise, competent, reliable people.*

I would like to dedicate this thesis to (prof.) Nicola Schiavoni. He is the one that believed in me from the moment I asked him for a recommendation letter to come here. Without knowing me too well, he agreed to recommend me and from that day I have wanted to dedicate all my academic/working/life victories to him.

I would like to dedicate each Chapter to specific people:

- Chapter (2) is dedicated to my supervisor, (dr.) Gabriel Lopes. After three months spent on a very specific topic (for my Literature colloquium): *Modeling and Control of Rigid Impacts*, it was decided to aim for a higher challenge. This chapter is the result of a new three months Literature survey. I kept in my mind your concept of "big picture". Although it was very hard to achieve, I hope that this chapter posses some characteristics of this concept.
Chapter (3) is dedicated to more than one person: Gijs van der Veen, (dr.ir.) Jan-Willem van Wingerden and (prof.dr.ir.) Michel Verhaegen. This dedication is not based on the fact of their comments on this chapter. In fact, they did not have the opportunity to see this chapter concluded before the publication of this thesis. Although this chapter could not elicit their satisfaction, I would like to dedicate it to them.

- Jan-Willem. Our only meeting showed me your expertise on the field of system identification. And it truly rewarded my little understanding, in such short space of time (Dec. 2015 - Jan. 2016), of interesting details of recursive least-squares with forgetting factor. And, only with your technical network, was I able to contact one of the kindest guy in 3me, Gijs.

- Gijs. This chapter took inspiration from your Phd thesis. I started this chapter with the intention of not taking these algorithms for granted, but to try to demonstrate them. I started as you started and I expanded the demonstration. Apart from this, without you I could not have believed in what I was doing and I could not have pushed myself in a good direction. You really are a good person as well as a valuable teammate.

- Michel. You were the right person at the right moment (and fortunately, you were available at that moment). So, although the interpretation of recursive least-squares with forgetting factor as a weighted least-squares problem could seem banal to you. To me, it was the completion of a path that could have not been reached without you. The 15-minute-discussions with you were essential for me to complete this chapter and to be happy about what I had done.

Chapter (4) is dedicated to my unconditional determination to seek the right knowledge for a problem which was difficult for me.

Chapter (5) is dedicated to Andrea Simonetto. I met you in February 2016 and for the first time ever, after eight months struggling with my thesis you wrote me equations that I believed could finally solve my problem. You proved your indisputably formal attitude to approach a new, unknown problem. And without you, this chapter could not been completed. I really pushed myself to understand what we were discussing so I could implement it the same day.

Chapter (6) is dedicated to (dr.ing.) Jens Kober. I would have continued my first thesis on Reinforcement Learning if and only if you had been part of this university. Unfortunately, you arrived a month later and I started a different topic. "Can I use your robot?". It was the craziest question that I have ever asked in my academic life. And you did not only allowed me to use the platform, but you directly helped me in the lab. This was the first moment where I felt to be part of a research team. Without your expertise with the robotic platform, this chapter would not exist and I would not have had a lot of fun in the lab!

I would like to dedicate my hard work to my friends. This thesis is the result of your indirect and direct influences on me.

I really want to thank all the people that I met here in Delft. From the ones that spent only six months here to the ones that will complete their entire master. You are
extraordinary people. Starting from my roommates: from 2013 you made my journey lighter and easier. A single day does not go by where I regret having chosen you as roommates. In particular, I would like to thank the first person that I met here in Delft, A. Šen. All the conversations that we had, they were inspirational speeches.

I would like to thank my international family. You really proved to be a really great family and I am so happy to be part of it. I would like to thank all the people that surprised me. It is hard to explain but sometimes, in life, some people surprise you in a very positive way.

I would like to thank A. Berry, C. Verdier and T. Bruin: your unconditional support was essential to finish this journey.

- I would like to thank all my friends/classmates in Politecnico di Milano. They really never stop supporting me. I am really sorry to have left Milan back in 2013. In some ways it was a tremendous mistake, but in other ways that decision, at that time, was one of the best decisions in my life. In particular, my hard work is dedicated to E. Sita and E. Merlo. With the hope that one day I will be as good as you are.

- I would like to thank you my friends in Tuscany. For the past five years I did not spend to much time in Tuscany, but every time that I went back, I really felt at home with you. I would like to thank A. Noferi. This academic journey could not have existed without that day five years ago; the day that you indirectly challenged me to aim high. I would like to dedicate my past five years (and more) to my three milestones that have supported me from very far away: A. Bucciolini, S. Salvi and T. Bocci. You represent several stages of my life. In so many ways, you are basically my history. Without you, my life would have been completely meaningless. Because you were the reasons I woke up every day and gave my best.

Delft, University of Technology
March 18, 2016

Matteo Ciocca

Master of Science Thesis Matteo Ciocca
“Il sacrificio perfetto è sempre una vittoria futura”
— Gabriele D’Annunzio
Chapter 1

Introduction

1-1 Interactive Tasks in Robotics

Robotic platforms would not have reasons to exist without an interactive purpose. Robots must deal with tasks of interaction.

Interactive tasks involve always at least two subjects: an actuated system (in our case a robotic platform) and the so-called environment. Every interactive task starts with the decision on the way to interaction: a visual interaction, a tactile interaction, a sound interaction, a localization task and so on. These ways of interaction can be combined too, but they are usually possible thanks to the sensor (or mechanical parts) with which the robotic platform is equipped. A set of sensors should be selected to re-create or resemble the human senses, including: sight, hearing, touch. The sensors selection is twofold: they should be used to control the robotics platform and their selection should be based on the type of interaction. Example of sensors are: force sensors, laser scans, camera system and so on. Given the set of sensors and the way of interaction, the word environment should be defined. Environment in robotics is used to define a general external thing, not actuated, with which the robot will interact. To define the word environment, some categories can be listed: a terrain where a robot walks, a surface, an object, a sound, a liquid in which a robot is immersed, an unexplored place and so on. The selection of a specific environment of with a proper set of sensors, defines a specific interaction task. Some interaction tasks (with the respective environment) are: walking (terrain), speech recognition (sound), mapping (an unexplored place), swimming (liquid), grasping (object). Last (regarding the mechanical parts), but not least, the robotics platform should be suitable for the interaction task. A KUKA robot manipulator is not suitable for a swimming task, an automated car cannot grasp an external object, on the other hand, a robot equipped with two legs is suitable for a walking task and a robot equipped with a camera is suitable for object detection.

The decision of a specific robotic platform, a way of interaction and a set of sensors do not completely define an interactive task. To fully define an interactive task, a goal, or from a control theory prospective, a control goal is needed:
The goal states how the robotic platform should interact with the environment.

The goal helps the user to select a particular algorithm suitable for the task. Walking cycle algorithms (walking task) \[10\], SLAM algorithms (exploration task) \[11\], computer vision algorithms (object detection) \[12\].

To summarize, an interactive task is fully defined with the following elements: way of interaction + type of robotics platform + set of sensors + defined environment + a proper algorithm. From the complete description of the interactive task several challenges arise. Challenges are related with the control goal. General challenges are: tracking (of a specific trajectory), rejection (of a general disturb), regulation (of the robot during the interaction) and estimation (in general: of state or properties).

These challenges regard the robotic platform and the environment coupled together during the interaction. In general:

The challenge arises from the unknown (or partially unknown) dynamics of the environment with which the robot will interact.

In fact, if the dynamics of the environment were known a priori, it would be easy to understand its behavior during an interaction and possibly to create a strategy that could predict and/or anticipate the environment response, feeding this information to a general control architecture that could help the robotics platform during the interaction. This lack of information renders the robotic platforms incapable of fully "understanding the environment behavior" and it leads them to adopt different intelligent control strategies.

1-2 Motivation

From the previous section it is clear that interactive tasks cover most of the fields in robotics. The thesis decided to focus on a particular interaction task with a specific environment. In our case, the interaction is effectuated by contacts between the robotics platform and the environment: a tactile interaction. The environment is chosen to be an unknown (or partially unknown) external object. The robotics platform is a robotics hand (or more generally a robotics end-effector). Because of an interaction task trough touch, the sensors used are tactile sensors: the type of sensors are force sensors. In general, tactile interaction tasks with external objects are becoming popular in literature, which has the particular goal of studying these objects \[13, 14, 15\]. Some purposes for this study are: grasping an object \[16, 17\] or the pushing an object to a particular position \[18, 19, 20\]. In general, the purposes listed above always have a control part working side-by-side with an exploration task. In our case instead, the interaction task focuses only on the exploration of the unknown object (in the study itself): a tactile exploration. The thesis focuses on the exploration aspects only.

A tactile exploration with an external object aims to learn few important characteristics:

• **The geometry of the object**

• **The mobility of the object**

• **The properties of the object**
Although geometry and mobility are well-defined: the first concerns the shape of the object and the second concerns the possibility of the object to move, the third must be explained. In our case, the properties of the objects are restricted to the following categories:

- **Stiffness of the object**
- **Mass of the object**
- **Friction characteristic between the object and the surface where it sits**

From a literature survey, these object properties were found to be the most popular because of their use in practical application. For example, the stabilization between the robot and external surface can be achieved successfully through the knowledge of the stiffness of the surface [21, 22, 23]. If a robot must push an object along a trajectory, its friction description, between the object and the surface where it sits, is useful to understand if the object will slide away in an undesirable direction [24]. If a robot must lift an object, the knowledge of its mass can help to understand if the "lift-action" is possible or not.

However, in the literature, sometimes this estimation is avoided using instead intelligent algorithms [25, 26, 27]. The robot tries to perform a task and, thanks to a learning process, (Reinforcement learning, Imitation Learning [25], Kernel methods [28]), it tries to learn after each trial. These methods deal with feature space, classification and data in general. Those algorithms are powerful tools. But the thesis took another direction. The thesis focuses on model-based techniques. Some of these techniques are not so far from intelligent algorithms, because both deal with a cost function: based on some information gathered from an interactive experience. However, model-based techniques always start with a model, possibly a dynamical model. In our case, a model of the object. And these techniques try to shape the object model parameters while performing an interaction with it, because they assumed that the model is a good mathematical representation of the object.

Models for object exist in literature. These models are not usually global, but they describe the local characteristic of an object. The most famous ones are models for compliant surfaces [29, 30, 31]. These models describe local deformation of an object or surface. Thus, they try to estimate stiffness property. Actually, only these kinds of dynamic models were found in literature. To my knowledge, there was no mention of a continuous time model to estimate simultaneously (with a dynamical model) friction and mass of an object. Although several friction models were found in literature, all of them were widely used to compensate the internal friction characteristic of robot manipulators [32, 33, 34, 35]. On the other hand, none of them were used to actually estimate friction properties of an external object.

Nevertheless, when something is not found in scientific literature, it could mean three things.

(i) The literature research process was not deep enough and what was not found had already been discovered and studied.
(ii) The question is banal and/or cannot lead to any good scientific improvements, and therefore, there was no reason to cite it in the literature.
(iii) It is a new scientific question that could finally close some theoretical/practical gap or solve some technological problems.

Despite the fear of being in (ii), (iii) was assumed and the following research question was asked.
1-3 Research Question

Given a robotic platform, in our case a manipulator with a compliant end-effector, but no visual sensors. Is it possible to design a simple model for a rigid movable object and through a tactile interaction (using the robot arm’s end-effector), use this model to identify relevant object properties? In our case the mass of the object and the friction between the object and the (unknown) surface where it sits.

1-4 Thesis Overview

The thesis focuses on the only case that was not found in the literature: a rigid movable object. This object is assumed to be rigid: with a high stiffness and movable - with friction. Because of the assumption of its high value, the stiffness property is not the goal of the estimation. On the other hand, the estimation aim is to estimate the mass and the friction of the object. The thesis understands the common characteristic of model-based parameter estimation techniques and selects the techniques considered suitable for such a task.

The thesis aims to understand which characteristic a simple object model should represent for the given task, and state which measurements are important. Furthermore, after the selection of the object model and an identification algorithm, a novel experiment for the properties extraction of a rigid movable mass is presented.

The thesis brought under a different perspective friction and mass estimation and tried to exploit touching capabilities, without the use of a vision system, to understand limitations and performance of a tactile estimation with the selected type of object.

The object model and hence the experiment are limited to the one dimension. The model-based algorithms used belong to the family of Recursive Least-squares. Both choices were inspired by the literature. The dimensionality of the model was proved to be effective and successful for the case of compliant surfaces. And the identification framework was found a perfect match for the research question asked.
1-5 Outline of the Thesis

Background on Model-Based Parameter Estimation Techniques. A new literature survey on model-based techniques equipped with a parameter estimation mechanism. A classification of local dynamical environment models to describe external surfaces or objects. The models do not take into consideration the geometry of the environment. The parameter estimation mechanism tries to estimate specific properties described by those models.


Object Model. Modeling of a rigid movable object w.r.t. the strategies selected. Test of the strategies on the designed models. Insights on algorithms used. Particular attention on the persistent excitation problem.

Experiment. A novel simulated experiment where the model and the strategies selected are tested. A robot finger will push and excite the dynamics of a rigid movable object and it will try to estimate online its mass and the friction between the object and the (unknown) surface where it sits. Different approaches are simulated. Results are fully discussed.

Experimental Setup and Results. A robotics platform: KUKA LBR iiwa 7 R800, is used to test the simulated strategies. The Chapter is the first attempt to test these strategies with a real rigid movable object, but the experiments are far from being complete. However, results and observation were found interesting to be reported.
Chapter 2

Background on Model-Based Parameter Estimation Techniques

This Chapter opens the thesis with an investigation on Environment models used for model-based robot-interaction techniques. Environment models are abstracted in a general dynamical system description and, afterwards, they are classified into two classes: compliant surfaces and rigid movable objects (here introduced). The latter is novel, in the sense that it is not explicitly used for any reviewed applications. The second part of the chapter introduces three model-based strategies: Parametric System Identification, Adaptive Control and Adaptive Observer. These techniques are used to achieve a stable interaction between a robotic platform and the environment meanwhile estimating environment properties. Our focus is to understand how to combine environment models and model-based techniques: aiming to extract relevant environment property with the help of a model, hence exploring the interacting environment.

2-1 Environment models

Interaction always involves impact between two objects. Theoretically speaking, the description of impacts/collisions involves an instantaneous change of velocity [36]. However, this approach has never used for practical application because it is not advantageous to have an instantaneous change of physical quantities (Figure (2-1)).

![Figure 2-1: Rigid Collision](image-url)
At a certain time, called $t_c \in \mathbb{R}$ the collision time, an instantaneous change of a continuous value (like the velocity signal) occurs. This means a *jump* and a creation of a derivative of the signal in that instant $t_c$ that goes to infinity $\rightarrow \infty$. *Infinities*, occurring when analyzing a system with the goal of control, mean incomplete problem description! [37]. In order to overcome the problem of infinities, a compliant element is added between the colliding bodies to achieve a continuous interaction, without creating discontinuous signals (Figure (2-2)).

![Figure 2-2: Compliant Collision](image)

What it is obtained from a simulation (and application) prospective is the addition of a new state during the interaction that is the elongation of the spring.

*This model is at the base of every interaction model used in both simulation and real experiments for robotics platforms*

The scheme in Figure (2-2) it also respects the famous *Hogan's rule* [38]. In fact, Hogan distinguishes two elements: impedance and admittance element. The first element has an input a position (or velocity) and as output a force. The second has as input a force and it imposes a position (or velocity). In our case, the two masses are admittance elements and the spring in the middle is the impedance element. The spring gives back a linear force that allows a compliant continuous motion between the colliding masses. These elements are strictly related to the concept of *flow* and *effort* and causality (input/output); for a complete understanding of this theory the reader is advised to read [38, 39].

Thanks to the new description, that avoids the *problems of infinities* (causing signal discontinuities), the environment can be additionally described as a dynamical model. A general form of a dynamical environment model is the following:

$$\Xi := \begin{cases} \dot{\eta}(t) = f_e(\eta(t), \zeta) + g_e(\eta(t), \zeta)v(t) \\ z(t) = h_e(\eta(t), \zeta) + w_e(\eta(t), \zeta)v(t) \end{cases} \quad (2-1)$$

The environment model $\Xi$ is defined as a nonlinear time invariant system with four functions: $f_e$, $g_e$, $h_e$, $w_e$ nonlinear in the states vector $\eta \in \mathbb{R}^n$, in the parameters vector $\zeta \in \mathbb{R}^p$ and in time $t$; with a control input $v \in \mathbb{R}^m$ and an output variable vector $z \in \mathbb{R}^q$.

This modeling structure is faithful to the general control system theory [40, 41, 42]. On the other hand, we can assign a physical interpretation of this nonlinear system model that has nothing to deal with the well-known concept of plant. A *plant is a combination of process and actuators*. The system ($\Xi$) does not belong to this

---

1Parameters can also depend in some environment characterisitic or being time dependet, but in this description they are assumed constant.
definition. The system does not have any actuators, the input of the system \((v)\) is, in our description, an external force that influences the system by an interaction with another system (or plant). The state vector of this system \((\eta)\) is not directly accessible (or measurable) and the parameter vector \((\zeta)\) is unknown. Take for example a rolling sphere (on a table). Position and a velocity of the sphere can belong to the state vector \((\eta)\), and friction (between the table and the sphere) coefficients and deformability of the sphere can belong to the parameter vector \((\zeta)\).

We should imagine that the system does not evolve in time until an interaction force does not influence it.

Defined a general description of the environment model, we now introduce a general description of a plant (in our case a robotic platform) that will interact with \(\Xi\). The plant is an actuated dynamical system:

\[
\Sigma := \{ \dot{x}(t) = f_s(x(t), \sigma) + g_s(x(t), \sigma)u(t) \\
y(t) = h_s(x(t), \sigma) + w_s(x(t), \sigma)u(t) \}.
\] (2-2)

The dynamical nonlinear system \(\Sigma\) is defined by four nonlinear functions: \(f_s, g_s, h_s, w_s\) dependent in the states vector \(x \in \mathbb{R}^a\), in the parameters vector \(\sigma \in \mathbb{R}^c\); and the dynamical system has a vector of actuators: control input \(u \in \mathbb{R}^b\) and a output variable vector \(y \in \mathbb{R}^d\).

When the system \(\Sigma\) and \(\Xi\) interact, a new general dynamical system description can be defined as:

\[
\Upsilon := \{ \dot{\rho}(t) = f_{s,e}(\rho(t), \nu) + g_{s,e}(\rho(t), \nu, u) \\
\iota(t) = h_{s,e}(\rho(t), \nu) \}.
\] (2-3)

Domains of vectors are: \(\rho = [\eta, x] \in \mathbb{R}^{n \times a}, \nu = [\zeta, \sigma] \in \mathbb{R}^{p \times c}, \iota = [z, y] \in \mathbb{R}^{q \times d}\) and \(u \in \mathbb{R}^b\).

Notice that the nonlinear function \(g_{s,e}\) is the only one that contains the input (actuator) vector of this new dynamical system. The nonlinear functions \(f_{s,e}\) and \(h_{s,e}\) are instead, created from the interaction.
2-1-1 Compliant Surface Models

The general model description introduced before can interpret correctly a famous class of environment models found in the literature study. A simple spring or a spring and a damper in parallel are widely used [43, 30, 29, 44, 45] to describe a compliant (or rigid!) environment model. Precisely, a compliant or rigid surface. The compliant model showed in Figure (2-2) assumes a small amount of local deformation at the contact, which allows the contact forces to be expressed as functions of local position (in general state variables). This feature makes it relatively easy to incorporate a compliant contact model into a dynamics simulator [30]. Two models were found extremely used in literature: Kelvin-Voigt (linear) and the Hunt-Crossley (nonlinear) models [46, 29, 47, 46].

Given the position \( a \) and velocity coordinates \( \left[ \eta_1(t), \eta_2(t) \right] \) and the position of the surface in \( \eta_1(t) = 0 \); the two models are described as follows:

\[
F_{KV}(t) = \begin{cases} 
K\eta_1(t) + B\eta_2(t) & , \eta_1 \geq 0 \\
0 & , \eta_1 < 0 
\end{cases}, \quad \text{and} \quad F_{HC}(t) = \begin{cases} 
k\eta_1^n(t) + b\eta_1^n(t)\eta_2(t) & , \eta_1 \geq 0 \\
0 & , \eta_1 < 0 
\end{cases}.
\] (2-4)

Where \( K \) and \( k \) are the elastic coefficients, \( B \) and \( b \) viscous coefficients and \( n \) is a real number, usually close to unity, that takes into account the geometry of contact surfaces.

It is interesting to see that the two descriptions can be related to our general environment system model.

\[
z_{KV}(t) = h_c(\eta(t), \zeta) : z_{KV}(t) = F_{KV}(t) , \quad \zeta = \begin{bmatrix} K \\ B \end{bmatrix} , \quad \eta(t) = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \end{bmatrix} \\
z_{HC}(t) = h_c(\eta(t), \zeta) : z_{HC}(t) = F_{HC}(t) , \quad \zeta = \begin{bmatrix} k \\ b \\ n \end{bmatrix} , \quad \eta(t) = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \end{bmatrix}.
\] (2-5)

The two models have been developed in order to achieve a continuous description of the relation between the body’s penetration and the consequent reaction force [29, 48]. Therefore, these models can be applied to describe and to estimate the contact dynamics between the end-effector of a robotic manipulator and surrounding objects. The main advantage of the Hunt-Crossley nonlinear model over the Kelvin-Voigt formulation is its consistency with the coefficient of restitution, and for this reason, it is more suitable to describe energy dissipation effects, especially for soft materials.
2-1-2 Rigid Movable Object Models

The previous example can be physically simplified. In fact, theoretically speaking, when two bodies are in contact and they remain in a contact condition, their contact position and velocity physically coincide. Meaning that the position of the end-effector of a robot $x(t)$ and the position of the object $\eta_1(t)$ coincide during the interaction, as well as the velocities $\dot{x}(t) = \dot{\eta}_2(t)$. The purpose of the generalization is instead to understand that certain state variables of a general object cannot be related to the state variables of our controlled platform. Hence, they have to be explicitly called in the state vector during an interaction.

As next class of object models, I considered models that represent rigid movable objects. The simplest description of this class of models is the following\(^2\):

$$
\Xi := \begin{bmatrix}
\dot{\eta}_1 \\
\dot{\eta}_2
\end{bmatrix} = 
\begin{bmatrix}
0 & 1 \\
0 & -\frac{b_v}{m}
\end{bmatrix}
\begin{bmatrix}
\eta_1 \\
\eta_2
\end{bmatrix} + 
\begin{bmatrix}
0 \\
\frac{1}{m}
\end{bmatrix} F 
\quad \Leftrightarrow \quad \dot{\eta} = f_e(\eta, \zeta) + g_e(\eta, \zeta)v
$$

(2-6)

$$
\eta^T = [\eta_1 \ \eta_2]^T, \ \zeta^T(t) = [b \ m]^T, \ v = F .
$$

The one dimensional model represents a mass $m$ with a viscous friction coefficient $b_v$, with a state vector composed respectively by the position and velocity $[\eta_1, \eta_2]$ of the object. The input ($v$) is an interacting force: it is not an actuator because we are considering this model as an object model.

The friction force represents the force that is created when the object slides on the surface where it sits. In this case, thanks to the linearity of the friction force with the velocity state variable, it is physically understandable that when this object interact with our robot end-effector (for example the robot finger push the object), the velocity of the object will coincide as the end-effector velocity.

However, in the general description, some state variables of the object can be different from our actuated system even during the interaction. This can happen when the object has an internal dynamics. An example of this kind is to consider a dynamical model for the object’s friction. After a review of several friction models [32], we can categorize the one used above as a static friction model. A well-known complete static friction model is the following:

$$
F_{\text{static}}(\eta_2) = \left( F_c + (F_s - F_c)e^{-\frac{v_s}{\delta_s}} \right) \text{sgn}(\eta_2) + b_v \eta_2.
$$

(2-7)

Where $F_c$ is the Coulomb friction, $F_s$ is the stiction force and the term $v_s$ and $\delta_s$ accounts for the Stribeck velocity, ($\delta_s$ usually equals to 2). These models relate the friction force to the

\(^2\)Omitted the time variable $t$. 

Master of Science Thesis

Matteo Ciocca
current velocity values. However, the friction phenomena related to non-stationary velocities, variations in the break-away force, small displacements occurring during the stiction phase, and hysteresis effects can be captured only by dynamic models, which describe the pre-sliding behavior representing the microscopic asperities of the contact surfaces by means of elastic bristles, whose deflections give rise to the friction forces. This phenomena is captured by dynamical models.

A dynamical well-known friction model is the LuGre (Lund-Grenoble) friction model [49]:

\[
\begin{aligned}
\dot{z}_1 &= \eta_2 - \frac{|\eta_2|}{g(\eta_2)} a_0 z_1 \\
F_{fric} &= a_0 z_1 + b_0 \dot{z}_1 + f(\eta_2)
\end{aligned}
\]

(2-8)

Where \(z_1\) is an internal state variable (describing the bristles), and \(g(\eta_2)\) and \(f(\eta_2)\) model the Stribeck effect and the viscous friction, respectively.

\[
F_{static}(\eta_2) = g(\eta_2) \text{sgn}(\eta_2) + f(\eta_2).
\]

(2-9)

Substituting the dynamical friction model in our movable rigid object, we obtain:

\[
\Xi := \begin{cases}
\dot{\eta}_2 = -F_{fric} + F/m \\
\dot{z}_1 = \eta_2 - \frac{|\eta_2|}{g(\eta_2)} a_0 z_1 = \Phi_1(\eta(t), \zeta) \\
F_{fric} = a_0 z_1 + b_0 \dot{z}_1 + f(\eta_2)
\end{cases}
\]

(2-10)

Where,

\[
f_e(\eta(t), \zeta)^T(t) = \begin{bmatrix} \eta_2 & -F_{fric}(\eta(t), \dot{\eta}(t), \zeta) & \Phi_1(\eta(t), \zeta) \end{bmatrix}, \quad g_e(\eta(t), \zeta)^T = \begin{bmatrix} 0 & 1/m & 0 \end{bmatrix}, \quad v = F
\]

\[
\zeta^T = \begin{bmatrix} m & a_0 & \dot{a}_1 & F_s & F_c & b_v \end{bmatrix}, \quad \eta^T(t) = \begin{bmatrix} \eta_1(t) & \eta_2(t) & z_1(t) \end{bmatrix}
\]

(2-11)

The model mathematically tells us that the object has an internal dynamics that cannot coincide with any state variables of our actuated system.

The one dimensional model of a rigid movable object is not theoretically new. In fact, a mass-damper model can be found in every control theory book. However, the interpretation to use this model in order to describe an external object has never been questioned in the literature, hence it was found interesting to explore.

### 2-2 Online Model-Based Parameter Estimation Techniques

Three widely used model-based techniques were reviewed in the literature. The scenario is a robot interacting with the environment. The robot incorporate a model of the environment and, in order to interact correctly with it, an algorithm estimates particular (and most of the time local) environment properties. Three techniques achieve different goals:

- **Parametric System Identification**: parameter adaptation.
- **Adaptive Control**: parameter adaptation and controller design.
- **Adaptive Observer**: parameter adaptation, state estimation and controller design.

Meanwhile the second technique is based on the first, are somehow related (Figure (2-5)), the third technique is a combination of Observer Design and Adaptive Laws.

![Euler-Venn Diagrams on the distinction between Parametric System Identification and Adaptive Control](image)

Using the dynamical description in Equation (2-3), we will distinguish (categorize) these three techniques based on the parameters that they will estimate.

### 2-2-1 Parametric System Identification

System identification [50] is a widely field that I could not explore entirely and, for the purposes of this thesis it was not required. The main focus of my research was to find all the model-based techniques of this field used for interaction tasks. In the case of System Identification, these techniques belong to Parametric System Identification\(^3\) (PSI). A complete detailed theory (and applications) of these techniques are the main focus of several books [51, 52, 53, 54]. PSI techniques, in general, lack a controller, but they are at the base of all control algorithms presented in this Chapter. The main goal of these techniques is to estimate a vector of parameters. Referring to Equation (2-2), we can transform it in:

\[
\Sigma := \begin{cases} 
\dot{x} & = f_s(x, \hat{\sigma}) + g_s(x, \hat{\sigma})u \\
y & = h_s(x, \hat{\sigma}) + w_s(x, \hat{\sigma})u 
\end{cases}.
\]

Where the parameters vector is unknown (or partially unknown) and the signal used for the estimation are \(u\) and \(y\). The scheme block of a PSI techniques is showed in Figure (2-6).

\(^3\)I named it.
Few elements are fundamental in this scheme.

- The unknown or partially unknown plant. The partial knowledge of the plant is the main reason to use these techniques. The knowledge of the plant is referred to the parameters of the model only.

- The Identifier. Its the main architecture that allows the algorithm to estimate the unknown parameters. Its includes all the variables previously introduced.

- The input signal and the output signal. Without them there could not be any identification.

- The controller\(^4\). It could be used to drive the input signal for the plant.

An important step is to understand how these techniques are used to estimate external properties. In fact, we are not interested in plant parameters, but in object parameters.

\[
\Xi := \begin{cases} 
\dot{\eta}(t) = f_e(\eta(t), \hat{\zeta}) + g_e(\eta(t), \hat{\zeta})v \\
z(t) = h_e(\eta(t), \hat{\zeta}) + w_e(\eta(t), \hat{\zeta})v 
\end{cases}
\]  

(2-13)

The conventional PSI scheme (Figure (2-6)) is not useful to represent a robot-environment interaction situation. Instead, It is reasonable to use the scheme in Figure (2-7). This scheme does not compromise any prior assumption or theorem (Chapter (3)) because the structure of the identifier remain unchanged. The change is based on the selection of input and output signal for the identifier and what the identifier is actually estimating. In general, if it is possible to measure the input and the output signals of a plant, those signals will be use for the identification. In the case of an interaction task, if the environment is under analysis of identification, we do not have "access" to its input and output signal. In fact, the only signals that we can measure are the signals from our robotic platform. Example of this scheme is presented later.

\(^4\)Not showed in Figure (2-6)
In both Figures (2-6-2-7), the identifier contains the algorithm that will estimate parameter. In our case, the requirements of these algorithms are: (i) to estimate parameters online and they must be (ii) model-based techniques. Recursive Least Square Estimation (RLS) are powerful and widely used techniques that full-fill our requirements. In Chapter (3) this kind of technique will be introduced. The main concept is to describe a system with unknown parameter in an output form. This form should be an expression linear in the parameters. This form can be expressed as a multiplication of a vector/matrix of parameters $\Theta$ and a vector/matrix of signals $\Phi$ up to some time instant $k$.

In general, the main goal of the Least-Squares (LS) algorithms is to determine the parameters in such a way that an estimated output form, $z = \Phi \hat{\Theta}$, agrees as closely as possible with a measured variable $Y$ in the sense of least squares. Mathematically speaking, the estimated parameter $\hat{\Theta}$ should be chosen to minimize a least-square loss function\[5\]:

$$V(\hat{\Theta}, t) = \frac{1}{2} \sum_{i=1}^{t} \epsilon(i)^2 = \frac{1}{2} \sum_{i=1}^{t} \left( Y(i) - \Phi(i) \hat{\Theta}(i) \right)^2. \quad (2-14)$$

The Recursive-Least-Squares (RLS) algorithms [52, 54], recursively updates information to keep bounded the dimension of the regressor and output signals.

These techniques were used for experiments found in the literature to explore and extract properties from the environment [55, 29, 56, 57]. The basic idea was to use available signals such $F$ (from a force sensor) and $[x, \dot{x}]$ position and velocity, to create an RLS algorithm to identify property of contact surface model (Section (2-1-1)). These experiments exploit the three continuous signals $[F, x, \dot{x}]$ during the contact with a soft/rigid material and try to extract the environment properties: stiffness, damping and $n$-coefficient; properties given by a chosen model to describe the environment (Section (2-1-1)). For example in [55], the RLS algorithm uses the following variables:

- $\hat{\Theta}^T(t) = [\hat{K}, \hat{B}]$: adaptive identifier parameter. From the Kelvin-Voigt model (Introduced in Equation (2-4)).
- $\Phi^T(t) = [x_e - x, -\dot{x}]$: regressor vector. The state variables of the actuated platform: in this case the end-effector’s position and velocity of a manipulator and the known position of the environment surface $x_e$.

\[5\]Here in discrete time. Continuous version in [51].

Master of Science Thesis
Matteo Ciocca
• The output function $y$:

$$Y(t) = F(t) = \left[ x_e - x - \dot{x} \right] = \Phi^T \Theta.$$  \hspace{1cm} \text{(2-15)}

The experiment consider only normal contact forces, extracted with a force sensor on the tip of the platform.

2-2-2 Adaptive Control

Most of the PSI techniques are one of the essential component in the design of Adaptive control [58, 51, 53, 59]. Adaptive control is based on a fundamental hypothesis and a precise definition:

**Definition 2-2.1.** Fundamentals of Adaptive Control:

• **When Adaptive Control?** For any possible values of plant (disturbance) model parameters there is a controller with a fixed structure and complexity such that the specified performances can be achieved with appropriate values of the controller parameters.

• **Adaptive Control:** Technique of applying some system identification techniques to obtain a model of the process and its environment from input-output experiment and using this model to design a controller - controller based on automatic parameter adjustment. Theoretically, adaptive control is used for unknown but fixed plant. In practice, for slowly time-varying and unknown plant.

Thus, Adaptive control is "not so far" from RLS techniques. However, it deals with control. In fact, the scope of Adaptive control is to design an intelligent model-based controller that will adapt online his structure to make a controlled plant follows a determined trajectory. Re-connecting to a general dynamical system:
The controller is dependent on some unknown plant parameters that will be estimated online thanks to an identifier. The general plant for PSI (Equation (2-12)) coincides with the plant used for adaptive control purposes. But, the estimated parameters are fed to a control architecture that will use them to drive the system trajectory $y(t)$ toward the desired one $r(t)$. A general complete scheme of adaptive control is showed in Figure (2-8). These scheme have several blocks that if removed in a certain manner, different types of adaptive control are obtained:

- **Direct Adaptive Control.** The reference model gives the desired time trajectory of the plant output, aimed at finding an effective control law, (the controller is directly estimated). Scheme in Figure (2-8). The reference model has the same structure of the plant but has reference parameters. This techniques shapes a controller to drive the parameter of the plant to the same parameter of the reference model in order to obtain a reference performance.

- **Indirect Adaptive Control.** Also called *self-tuning control*. Lyapunov-based algorithms strategies are used in indirect adaptive control. These strategies can be re-casted as PSI techniques. Scheme in Figure (2-9a). The Identifier has a model reference structure and it feeds the controller with the estimated parameter of the plant. The latter has to converge to the model reference of desired performance.

- **Gain Scheduling.** The gains of different parameters are changed online with some reference parameters in order to achieve a precise performance: a set of parameters for a set of reference models. Scheme in Figure (2-9b).

Adaptive control has a long history, starting with an attempt to further the design of autopilots in 1950s [51] and used for many others control applications [60]. Between them, the use of adaptive control is found for interaction tasks. The important characteristic of adaptive control for interaction is the fact to incorporate the model of the environment in the controller. The controller should find a good estimate of environment parameters during the interaction. These architectures differ for some characteristic in literature, but most of them agree in the
use of force signal $F$ (from a force sensor) to start an adaptation. Most important, in many adaptive controller architectures fundamental elements of PSI techniques can be identified, such as the regressor and residual.

In [21, 61], a force control uses an impedance filter to change the desired position of the system in order to match stability during a possible interaction. In this architecture the environment model is a simple linear spring. The spring is adapted online during the interaction and, in this case, can also assume negative values.

Another interesting approach was found in [62, 63, 64, 44, 65, 66] regarding Adaptive regulation for impacts. The interesting elements of this approach were: the use of a compliant model (or a simple spring) and the use of the back-stepping to design the control law.

However, Adaptive control approaches, rather than to explore the environment, they aim to achieve a stable contact. Most of the adaptive control architectures, used to interact with the environment, fall in the set of Indirect Adaptive Control. These schemes aim to estimate the parameters of the environment, but they are used directly for the control design. Thus, the exploring factor of the environment is constrained by the control goal.

In our case, because we have to explore the environment, a control objective could lead to a different perspective of our problem hence probably Adaptive Control seems to do not be a suitable architecture for our goal.

![Figure 2-10: Variable Impedance Filter scheme [21]](image)

### 2-2-3 Adaptive Observer

*Adaptive Observer* could be thought as a combination of adaptive controller and observer design. The first was explained in Section (2-2-2), the latter is introduced in this section. Here, the word observer means all these model-based techniques that try to estimate a number of not measurable states. Using the generalized plant model:

$$\text{Observer} := \begin{cases} \dot{\hat{x}} = f_s(\hat{x}, \sigma) + g_s(\hat{x}, \sigma)u + W(y, \hat{y}) \\ \hat{y} = h_s(\hat{x}, \sigma) + w_s(\hat{x}, \sigma)u \end{cases} \quad (2-17)$$

Instead of the parameter vector, the unknown variables are in the state vector. The $W$ function is usually a correction function with the so-called observer gain that is used to correct
the online estimation of the state thanks to the measurable output signal and the estimated one.

The model-based observer design can be combined with an adaptive algorithm for some unknown plant parameters and this combination converge in the category of Adaptive Observer. The goal of this architecture, if used for interaction task, is to adapt online: the estimate of the parameters of the environment and the estimate of the state of the environment model. What was found in the literature study were two main architectures: COBA (Candidate OBserver Algorithm) [67, 68, 69, 70, 71, 23, 31] and the famous Extended Kalman Filter [43, 72, 22, 73].

COBA is an Optimal Force-based Observer (Kalman). One example of COBA uses two force observers with two candidate (one high and one low) environment stiffnesses $K_s$. A linear interpolations is computed between the observers output to obtain an estimate of the environment stiffness.

![Figure 2-11:](image)

**Figure 2-11:** $[K_{sc_1}, K_{sc_2}]$ two stiffness candidates, $\Delta x_{c,k} = \Delta x_c(k)$ for error between the force sensor and the output of the observer. $\hat{K}_{s,k} = \hat{K}_s(k)$ final estimation of environment stiffness

Online stiffness estimation allows adaptation of the control laws to reduce the stiffness mismatch. This is done by adjusting the DC gain, the feedback gains and the nominal AOB (Active OBserver) model [74, 75] to conform to the estimated stiffness.

Another example is the use of *Extended Kalman Filter* (EKF) with the Kelvin-Voigt linear model to estimate online the stiffness parameter of an external environment. The form of the EKF is the following:

$$\begin{cases}
\dot{x}_a = f_s(x_a, \nu) + K_{EKF}(y - C_a \hat{x}_a) \\
\dot{\hat{y}} = h_s(x_a, \nu)
\end{cases},

f_s(x_a, \nu) = \begin{bmatrix}
\dot{x}_e \\
\dot{K}_e \\
\dot{D}_e \\
\dot{f}
\end{bmatrix} = \begin{bmatrix}
D_e^{-1}(-K_e x_e + f + \nu_x) \\
\nu_{K_e} \\
\nu_{D_e} \\
\nu_f
\end{bmatrix}.

(2-18)

Where $\nu^T = [\nu_{x_e}, \nu_{K_e}, \nu_{D_e}, \nu_f]$ is the vector of parameter uncertainties. The Kelvin-Voigt model is now nonlinear based on the fact that the parameters $(K_e, D_e)$ are unknown variables as well as the state variables, in particular, the algorithm tries to estimate the environment state variable representing the surface equilibrium position $x_e$ (in one dimensional case). The Kalman gain $K_{EKF}$ is computed covariance matrix and noise measurement. Here, the environment location $x_e$ is estimated through a force threshold.
2-3 Conclusions

From the survey conducted on model-based environment exploration, all the applications used compliant surface models. It is not a case that these models are widely used. If fact, most of the time, a common control goal (for an interaction task) is to stabilize a robotic platform with an external environment. Take for example an adaptive controller strategy. If the control goal is to stabilize with an external environment, it does not make sense to consider the external environment as a rolling ball (Movable Rigid Object). The environment models selected for interaction tasks in literature are models that allow an equilibrium situation, a stable position between the controlled platform and the environment.

In the case of experiments finalized to explore the environment properties, few cases where found in literature and all of them used compliant surface models. Digging in the literature survey, there were no experiments where a model of an object, that could move, was taken into account to represent the environment, not even in the so-called control-by-pushing [76, 20] strategies. Except in one case, where a 3D movable object exploration was taken into account by [14, 24, 77]. Their research and practical approach, to my knowledge, is the one that goes closer to my research question. What it found interesting in their experiment was:

- A mathematical compact model for planar sliding motion of an object. This 3D object model was very complex and based on the shape of the object (assumed known).

- How a robot acquires the parameters of such a model. Fundamentally, two parameters were interesting to me: the mass and the friction. Although the first could be estimated by lifting the object, they decided to measure the necessary force to start tilting the object (detect the starting of tilting with movement detection from the hand/arm and/or with vision), then use the contact and force/torque model to calculate the weight. The estimation of the static friction coefficient was conducted by applying an increasing force until the object starts sliding, the force just before this happens is the maximum static friction force. They used the hand/arm to measure when the object starts to move. To assure the object’s slip but not rotation, the robot pushed the object on a very low point of the object height.

Although in their approach, a vision system was used as well as the geometry of the object, they used model-based techniques. Our approach would like to:

- Use a reduced mathematical model side-by-side with an estimation algorithm that does not require to combine geometry and friction or geometry and stiffness. In fact, estimation techniques, for compliant surfaces, model the environment with just essential properties needed to estimate: without taking into consideration the geometry of the surface.

- Explore the class of models casted as: rigid movable objects, because we found these to be very unpopular in the literature. Creating an experiment where one (or many) of the model-based techniques explained in this Chapter can be used and be combined with the object model that is characterized by his essential properties to estimate.

- No vision system.
The technique that must be selected to explore the environment should be a model-based techniques, precisely:

*A model-based technique that can estimate online external parameter.*

The techniques reviewed in this Chapter fulfill the requirement for a *model-based environment exploration* and they present advantages and disadvantages: in terms of control goal. However, the first step is to select one of these strategy (or a family of strategies). Thus, posing the basis for a future model structure that will cooperate with the strategy selected.
Chapter 3

Identification Framework

The Chapter introduces the online model-based framework used in this thesis. A derivation of the well-known standard form of Recursive Least-Squares is presented with the expansion of the forgetting factor. Both derivation are presented in a general matrix form, not found in the literature. Afterwards, the stability analysis for parameter convergence completes the framework of Recursive Least-Squares with forgetting factor. Furthermore, the stability analysis for LTI systems using parametric system identification techniques is introduced and a mention on techniques not used closed the chapter.

3-1 Recursive Least-Squares Framework

The framework needs the introduction of three key elements:

\[
Y_k \triangleq \begin{bmatrix} y_{k-1} \\ y_{k-1+i} \\ \vdots \\ y_k \end{bmatrix} \in \mathbb{R}^n, \quad \Phi_k \triangleq \begin{bmatrix} \varphi_{k-1}^T \\ \varphi_{k-1+i}^T \\ \vdots \\ \varphi_k^T \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad \Theta \triangleq \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^n, \quad n_y = dim \mathbb{R} \Theta. \tag{3-1}
\]

The index \( k \) signifies that data up to time instant \( k \) is available and used. The vector \( Y_k \) is the output vector\(^1\), the matrix \( \Phi_k \) is the regressor matrix and \( \Theta \) is called the parameter vector\(^2\). The components of \( \Phi_k \) can be called regressors or regressor vectors.

\[
\varphi_k^T \triangleq \begin{bmatrix} \varphi_{k}^1 \\ \varphi_{k}^2 \\ \vdots \\ \varphi_{k}^n \end{bmatrix}.
\tag{3-2}
\]

The components \( \varphi_{k}^i \) are known function, called regressor variables, that may depend on other known variables. Given the time instant \( k \) define the regressor model the following expression:

\[
y_k \triangleq \varphi_{k}^1 \theta_1 + \varphi_{k}^2 \theta_2 + \cdots + \varphi_{k}^n \theta_n.
\tag{3-3}
\]

\(^1\)A vector and not a matrix, because the framework is limited to the case of a single output signal for each time instant \( k \).

\(^2\)A vector and not a matrix, because the framework is limited to the case of a vector of parameters.
A general least-squares problem aims to find the identifier parameter vector, $\hat{\Theta}_k$, with the following standard form[3]:

$$\hat{\Theta}_k = \arg \min_{\Theta} \left\{ \frac{1}{k} ||Y_k - \Phi_k \Theta||^2 \right\}.$$ (3-4)

The full-rank least-squares solution of (3-4) is:

$$\hat{\Theta}_k = \left( \Phi_k^T \Phi_k \right)^{-1} \Phi_k^T Y_k.$$ (3-5)

**Insight 3-1.1 (the full-rank least-squares solution).** The full-rank least-squares solution of a general least-squares problem (3-4) for the case where the output function is a single scalar and the regressor matrix is a single vector is derived. Define:

$$Y_k = y_k, \quad \Phi_k = \varphi_k^T.$$ (3-6)

**Proof.** Expand (3-4):

$$\frac{1}{k} ||y_k - \varphi_k^T \Theta||^2 = \frac{1}{k} (y - \Theta^T \varphi_k)(y - \varphi_k^T \Theta) = \frac{1}{k} \left( y_k^2 + \Theta^T \varphi_k \varphi_k^T \Theta - 2y_k \varphi_k^T \Theta \right).$$ (3-7)

The minimum can be found when the partial derivative w.r.t. $\Theta$ of (3-4) is equal to zero:

$$\frac{\partial}{\partial \Theta} \left( \frac{1}{k} ||y_k - \varphi_k \Theta||^2 \right) = 0.$$ (3-8)

$1/k$ can be eliminated. Substituting the expansion:

$$\frac{\partial}{\partial \Theta} \left( y_k^2 + \Theta^T \varphi_k \varphi_k^T \Theta - 2y_k \varphi_k^T \Theta \right) = 0$$

$$2(\varphi_k \varphi_k^T)\Theta - 2\varphi_k y_k = 0$$

$$(\varphi_k \varphi_k^T)\Theta = \varphi_k y_k.$$ (3-9)

Concluding that the $\Theta$ vector that minimize (3-9) is:

$$\Theta = (\varphi_k \varphi_k^T)^{-1} \varphi_k y_k.$$ (3-10)

The estimate consider the data available up to the time instant $k$. Suppose that a new sample becomes available for the estimation $[y_{k+1}, \varphi_{k+1}]$. Obviously, the data vector and matrix one time instant later, i.e., $Y_{k+1}$ and $\Phi_{k+1}$ contain the new samples and these elements therefore grow in time.

$$Y_k \triangleq \begin{bmatrix} y_{k-i} \\ y_{k-i+1} \\ \vdots \\ y_k \\ y_{k+1} \end{bmatrix}, \quad \Phi_k \triangleq \begin{bmatrix} \varphi_{k-i} \\ \varphi_{k-i+1} \\ \vdots \\ \varphi_k \\ \varphi_{k+1} \end{bmatrix}.$$ (3-11)

---

3The symbol $|| \cdot ||$ is the Euclidean 2-norm: given $x \in \mathbb{R}^n$, $||x|| = \sqrt{x^T x}$.  

Matteo Ciocca  
Master of Science Thesis
The least-squares solution with the new sample becomes:

\[
\hat{\Theta}_{k+1} = \left( \Phi_k^T \Phi_k + \varphi_{k+1} \varphi_{k+1}^T \right)^{-1} \left( \Phi_k^T Y_k + \varphi_{k+1} y_{k+1} \right) .
\]  

(3-12)

From (3-12), we define two new variables called respectively (up to the time instant \( k \)) the **information matrix** \( \mathcal{I}_k \) and the **covariance matrix** \( \mathcal{C}_k \):

\[
\mathcal{I}_k = \Phi_k^T \Phi_k , \quad \mathcal{C}_k = \left( \Phi_k^T \Phi_k \right)^{-1} .
\]  

(3-13)

Making use of this notation, the \( \mathcal{I}_{k+1}^{-1} \) in (3-12) can be simplified using the matrix inversion lemma, in order to avoid inverting a large matrix in each time step.

**Lemma 3-1.1 (Matrix Inversion Lemma).** Let \( A, C \) and \( C^{-1} + DA^{-1}B \) be nonsingular square matrices. Then \( A + BCD \) is invertible, and:

\[
(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} .
\]  

(3-14)

**Proof.** By direct multiplication we find that

\[
(A + BCD) \left( A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \right)
\]

\[
= I + BCDA^{-1} - B \left( C^{-1} + DA^{-1}B \right)^{-1}DA^{-1} - BCDA^{-1}B \left( C^{-1} + DA^{-1}B \right)^{-1}DA^{-1}
\]

\[
= I + BCDA^{-1} - (B + BCDA^{-1})(C^{-1} + DA^{-1}B)^{-1}DA^{-1}
\]

\[
= I + BCDA^{-1} - BC \underbrace{(C^{-1} + DA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}}_{I} DA^{-1}
\]

\[
= I .
\]  

(3-15)

Naming \( \mathcal{I}_k = A, \varphi_{k+1} = B, \varphi_{k+1}^T = D \) and \( I_{n_y} = C \) (an identity matrix), we arrive to the following expression:

\[
\mathcal{I}_{k+1}^{-1} = \left[ \Phi_k^T \Phi_k + \varphi_{k+1} \varphi_{k+1}^T \right]^{-1}
\]

\[
= \mathcal{I}_k^{-1} - \mathcal{I}_k^{-1} \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathcal{I}_k^{-1} \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \mathcal{I}_k^{-1}
\]

\[
= \mathcal{C}_k - \mathcal{C}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathcal{C}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \mathcal{C}_k ,
\]  

(3-16)

substituting this expression in (3-12):

\[
\hat{\Theta}_{k+1} = \left[ \mathcal{C}_k - \mathcal{C}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathcal{C}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \mathcal{C}_k \right] \left( \Phi_k^T Y_k + \varphi_{k+1} y_{k+1} \right) .
\]  

(3-17)
A final transformation can be done for the following expression:

\[
\hat{\Theta}_{k+1} = \left[ I_{n_y} - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \right] \cdot \mathbf{c}_k \left[ \hat{\Phi}_k^T \mathbf{Y}_k + \varphi_{k+1} y_{k+1} \right] \\
= \left[ I_{n_y} - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \right] \cdot \mathbf{c}_k \hat{\Phi}_k^T \mathbf{Y}_k + \mathbf{c}_k \varphi_{k+1} y_{k+1} \\
= \left[ I_{n_y} - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \right] \cdot \left[ \left( \hat{\Phi}_k^T \mathbf{c}_k \right)^{-1} \hat{\Phi}_k^T \mathbf{Y}_k + \mathbf{c}_k \varphi_{k+1} y_{k+1} \right] ,
\]

substituting the least-square solution (3-5),

\[
\hat{\Theta}_{k+1} = \left[ I_{n_y} - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \right] \cdot \left[ \hat{\Theta}_k + \mathbf{c}_k \varphi_{k+1} y_{k+1} \right] \\
= \hat{\Theta}_k - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \hat{\Theta}_k \\
+ \left[ I_{n_y} - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \right] \left[ \mathbf{c}_k \varphi_{k+1} y_{k+1} \right] \\
= \hat{\Theta}_k - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \hat{\Theta}_k \\
+ \left[ \mathbf{c}_k - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \mathbf{c}_k \right] \left[ \varphi_{k+1} y_{k+1} \right] ,
\]

substituting the covariance matrix update (3-16),

\[
\hat{\Theta}_{k+1} = \hat{\Theta}_k - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \hat{\Theta}_k + \mathbf{c}_{k+1} \varphi_{k+1} y_{k+1} .
\]

A final transformation can be done with for the following expression:

\[
\mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} - \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right) \\
= \mathbf{c}_k \varphi_{k+1} \left( \cdots \right)^{-1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right) - \mathbf{c}_k \varphi_{k+1} \left( \cdots \right)^{-1} \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \\
= \left[ \mathbf{c}_k - \mathbf{c}_k \varphi_{k+1} \left( I_{n_y} + \varphi_{k+1}^T \mathbf{c}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1}^T \mathbf{c}_k \right] \varphi_{k+1} \\
= \mathbf{c}_{k+1} \varphi_{k+1} ,
\]

and substituting in the identifier parameter vector update law:

\[
\hat{\Theta}_{k+1} = \hat{\Theta}_k + \mathbf{c}_{k+1} \varphi_{k+1} \left( y_{k+1} - \varphi_{k+1}^T \hat{\Theta}_k \right) .
\]
Finally, using the expression in (3-16) and (3-22), the algorithm is expressed by the following formula:

\[
\Pi := \begin{cases} 
\mathcal{C}_k &= \mathcal{C}_{k-1} - \mathcal{C}_{k-1} \varphi_k \left( I_{n_y} + \varphi_k^T \mathcal{C}_{k-1} \varphi_k \right)^{-1} \varphi_k^T \mathcal{C}_{k-1} \\
\hat{\Theta}_k &= \hat{\Theta}_{k-1} + \mathcal{C}_k \varphi_k (y_k - \varphi_k^T \hat{\Theta}_{k-1}) 
\end{cases} \tag{3-23}
\]

One important characteristic that distinguishes a recursive least-squares solution from a least-squares is the following: the size of the regressor matrix \( \Phi \) is reduced to the current instant, \( \varphi_k \). In fact, meanwhile a least-squares solution should collect the data and increases the dimension of \( \Phi \) (and \( Y \)), the recursive approach allows to update the covariance matrix in real-time with the information of the previous regressor vector as well as the identifier parameter vector.

To conclude, important elements of Parametric System Identification algorithms are summarized.

- \( \hat{\Theta}_k \in \mathbb{R}^n \): adaptive identifier parameter vector. Its the parameter vector that the algorithm has to estimate: dependent on time indeed.
- \( \phi_k \in \mathbb{R}^n \): parameter error vector, \( \phi_k = \hat{\Theta}_k - \Theta \). \( \Theta \) real parameters. Identify the error between the real parameter and the estimated ones.
- \( \Phi_k \in \mathbb{R}^{m \times n} \): regressor matrix. The output of a linear system can be written as \( z_k = \Phi_k \hat{\Theta}_k \). Its a vector containing measured signals, usually dependent on the state. It is different from the real output: \( Y_k \).
- \( \epsilon_k \in \mathbb{R} \): identifier error (or residual or linear error equation), \( \epsilon_k = \phi^T(t) \Phi_k = \Phi_k \hat{\Theta}_k - Y_k \).
- \( I_k \in \mathbb{R}^{n \times n} \): information matrix.
\[
I_k = \Phi_k^T \Phi_k \tag{3-24}
\]
- \( \mathcal{C}_k \in \mathbb{R}^{n \times n} \): covariance matrix.
\[
\mathcal{C}_k = \left[ \Phi_k^T \Phi_k \right]^{-1} \iff \mathcal{C}_k = \left[ \sum_{i=1}^{k} \varphi_i \varphi_i^T \right]^{-1} \tag{3-25}
\]

In Continuous time:
\[
\mathcal{C}(t) = \left[ \int_0^t \varphi(\tau) \varphi^T(\tau) d\tau \right]^{-1} \tag{3-26}
\]

### 3-1-1 Recursive Least-Squares Algorithms with Forgetting factor: \( RLS_{\kappa} \)

A factor \( \kappa \in (0, 1) \) is introduced to maintain a finite memory in the least-squares problem as opposed to an infinite memory. This ensures that the parameters remain adaptive;
furthermore this is required to maintain a finite (nonzero) covariance matrix. The new formulation of the least-squares assumes the characteristic of a weighted least-squares \[78\]. The new formulation to find the identifier parameter vector \( \hat{\Theta}_k \) is the following:

\[
\hat{\Theta}_k = \arg \min_{\Theta} \left\{ \frac{1}{k} (Y_k - \Phi_k \Theta)^T \mathcal{X} (Y_k - \Phi_k \Theta) \right\},
\]

\[\mathcal{X} = \text{diag}(\kappa^k, \kappa^{k-1}, \ldots, \kappa^2, \kappa, 1)\].

With \( \kappa = 1 \) the recursive least-squares standard form is recovered. The solution of (3-27) is:

\[
\hat{\Theta}_k = \Phi_k^T \mathcal{X} \Phi_k \frac{1}{\Phi_k^T \mathcal{X} \Phi_k} \Phi_k^T \mathcal{X} Y_k.
\]  

(3-28)

**Insight 3-1.2 (the full-rank weighted least-squares solution).** Given the weighted least-square problem in (3-27), the solution (3-28) is demonstrated in the following proof.

**Proof.** Expand (3-27):

\[
\frac{1}{k} (Y_k - \Phi_k \Theta)^T \mathcal{X} (Y_k - \Phi_k \Theta) = \left( Y_k^T - \Theta^T \Phi_k^T \right) \mathcal{X} \left( Y_k - \Phi_k \Theta \right) = \frac{1}{k} \left( Y_k^T \mathcal{X} - \Theta^T \Phi_k^T \mathcal{X} \right) \left( Y_k - \Phi_k \Theta \right) = \frac{1}{k} \left( Y_k^T \mathcal{X} Y_k - \Theta^T \Phi_k^T \mathcal{X} Y_k - Y_k^T \mathcal{X} \Phi_k \Theta + \Theta^T \Phi_k^T \mathcal{X} \Phi_k \Theta \right) = \frac{1}{k} \left( Y_k^T \mathcal{X} Y_k - 2\Theta^T \Phi_k^T \mathcal{X} Y_k + \Theta^T \Phi_k^T \mathcal{X} \Phi_k \Theta \right).
\]  

(3-29)

The minimum can be found when the partial derivative w.r.t. \( \Theta \) of (3-27) is equal to zero:

\[
\frac{\partial}{\partial \Theta} \left\{ \frac{1}{k} (Y_k - \Phi_k \Theta)^T \mathcal{X} (Y_k - \Phi_k \Theta) \right\} = 0.
\]  

(3-30)

1/\(k\) can be eliminated. Substituting the expansion:

\[
\frac{\partial}{\partial \Theta} \left( Y_k^T \mathcal{X} Y_k - 2\Theta^T \Phi_k^T \mathcal{X} Y_k + \Theta^T \Phi_k^T \mathcal{X} \Phi_k \Theta \right) = 0
\]

\[- 2 \Phi_k^T \mathcal{X} Y_k + 2 \Phi_k^T \mathcal{X} \Phi_k \Theta = 0 \]

\[
\Phi_k^T \mathcal{X} \Phi_k \Theta = \Phi_k^T \mathcal{X} Y_k.
\]  

(3-31)

Concluding that the \( \Theta \) vector that minimize (3-31) is:

\[
\Theta = \left[ \Phi_k^T \mathcal{X} \Phi_k \right]^{-1} \Phi_k^T \mathcal{X} Y_k.
\]  

(3-32)
The new weighted information matrix and weighted covariance matrix are introduced:

\[
\mathcal{I}_k = \Phi_k^T \mathcal{X} \Phi_k, \quad \mathcal{P}_k = \left[ \Phi_k^T \mathcal{X} \Phi_k \right]^{-1}.
\]  

(3-33)

Calling the information update law (3-16), we obtain the new covariance update as follows:

\[
\mathcal{I}_{k+1}^{-1} = \left[ \mathcal{X} \Phi_k + \varphi_{k+1} \varphi_{k+1}^T \right]^{-1}.
\]

(3-34)

Applying the Matrix Inversion Lemma (3-1.1),

\[
\hat{\Theta}_k = \hat{\Theta}_k + 1 \varphi_{k+1} \mathcal{P}_k \mathcal{X} \mathcal{I}_{k+1}^{-1} \mathcal{X}^T \varphi_{k+1}.
\]

Next, given with a new measurement sample, the following is obtained:

\[
\hat{\Theta}_{k+1} = \left[ \mathcal{X} \Phi_k + \varphi_{k+1} \varphi_{k+1}^T \right]^{-1} \left[ \mathcal{X} \Phi_k + \varphi_{k+1} \varphi_{k+1}^T \right] \cdot \mathcal{P}_{k+1} = \mathcal{I}_{k+1}^{-1}.
\]

(3-36)

Proceeding with a similar derivation as in (3-20) to obtain the final update parameter law:

\[
\hat{\Theta}_{k+1} = \frac{1}{n} \left[ I_{n_y} - \mathcal{P}_k \varphi_{k+1} \left( \mathcal{X} I_{n_y} + \varphi_{k+1}^T \mathcal{P}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1} \right]
\]

\cdot \left[ \mathcal{X} \hat{\Theta}_k + \mathcal{P}_k \varphi_{k+1} y_{k+1} \right]

\left[ \mathcal{X} \hat{\Theta}_k + \mathcal{P}_k \varphi_{k+1} y_{k+1} \right] + \mathcal{P}_{k+1} \varphi_{k+1} y_{k+1}

\hat{\Theta}_{k+1} = \hat{\Theta}_k - \mathcal{P}_k \varphi_{k+1} \left( \mathcal{X} I_{n_y} + \varphi_{k+1}^T \mathcal{P}_k \varphi_{k+1} \right)^{-1} \varphi_{k+1} T \mathcal{I}_{k+1} \hat{\Theta}_k + \mathcal{P}_{k+1} \varphi_{k+1} y_{k+1}

(3-37)
The transformation in (3-21), this time with the forgetting factor, can be applied to the last expression:

\[
P_k \varphi_{k+1} \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} \right)^{-1}
\]

\[
= P_k \varphi_{k+1} \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} \right)^{-1} \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} - \varphi_k^T P_k \varphi_{k+1} \right)
\]

\[
= \left\{ P_k \varphi_{k+1} \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} \right)^{-1} - P_k \varphi_{k+1} \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} \right)^{-1} \right\}
\]

\[
\hat{\Theta}_{k+1} = \hat{\Theta}_k - \frac{1}{\kappa} \varphi_k^T \left( \kappa I_n + \varphi_k^T P_k \varphi_{k+1} \right)^{-1} P_k \varphi_{k+1} \varphi_{k+1}
\]

\[
\Pi_{R} := \left\{ P_k = \frac{1}{\kappa} \left[ P_{k-1} - \frac{1}{\kappa} \varphi_k^T \left( \kappa I_n + \varphi_k^T P_{k-1} \varphi_k \right)^{-1} \varphi_k^T P_{k-1} \right] \right\}
\]

\[
\hat{\Theta}_{k+1} = \hat{\Theta}_{k-1} + \varphi_k \left( y_k - \varphi_k^T \hat{\Theta}_{k-1} \right).
\]

**3-1-2 Parameter estimation convergence for** \( RLS_{\kappa} \)

Derived the standard form of \( RLS_{\kappa} \), the study of the convergence of the parameter estimate vector is conducted through the Lyapunov stability. In what follow the demonstration is restricted to the convergence of the estimates only. The persistency excitation of \( \varphi_k \) is assumed.

**Definition 3-1.1 (Persistent Excitation (PE) discrete time [79])**. The vector sequence \( \varphi_k \) is persistently exciting if for some constant integer \( S \) and all \( j \) there exist positive constants \( \alpha \) and \( \beta \) such that:

\[
0 < \alpha I \leq \sum_{i=j}^{j+S} \varphi_i \varphi_i^T \leq \beta I < \infty.
\]

**Change of instant time,** \( (k+1, k) \rightarrow (k, k-1) \)
And the following is assumed (demonstrated in [79]):

**Definition 3-1.2 (Bounds on \( P^{-1} \)).** Suppose that \( \varphi_k \) is persistently exciting then for all \( k \geq S \):

\[
0 < \alpha \left( \frac{1}{\kappa} - 1 \right) I_{n_y} \leq P_k^{-1} \leq \beta \frac{1}{1 - \kappa^{s+1}} I_{n_y} .
\]  

(3-42)

\[\triangle\]

With these assumptions the covariance matrix, \( P_k \), converges in time to a bounded (positive definite) nonsingular matrix\[6\]. Moreover, given the persistent excitation of \( \varphi_k \), it is assumed that \( P_{k-1}^{-1} \) is uniformly bounded above and uniformly positive definite (i.e. bounded away from zero). The convergence of parameters to the nominal ones for a RLS\(\kappa\) algorithm is proved through the Lyapunov theorem [80, 79, 51, 81]. The difference from [79], is the initialization of the covariance matrix. This framework initialize the algorithm with \( P_0 \) instead of \( P_{-1} \).

Before proceeding with the Lyapunov stability theorem, the parameter estimation error\[7\] is defined with useful transformations:

**Definition 3-1.3 (Parameter Error Vector).** The parameter estimation error

\[
\tilde{\Theta} = \Theta - \hat{\Theta} .
\]  

(3-43)

Given (3-40), we can write:

\[
\Theta - \hat{\Theta}_k = \Theta - \hat{\Theta}_{k-1} + P_k \varphi_k (y_k - \varphi_k^T \hat{\Theta}_{k-1}) .
\]  

(3-44)

The output coincide to the following expression:

\[
y_k = \varphi_k^T \Theta .
\]  

(3-45)

Back to (3-43), the parameter estimation error update equation is:

\[
\begin{align*}
\Theta - \hat{\Theta}_k &= \Theta - \hat{\Theta}_{k-1} - P_k \varphi_k (\varphi_k^T \Theta - \varphi_k^T \hat{\Theta}_{k-1}) \\
\hat{\Theta}_k &= \hat{\Theta}_{k-1} - P_k \varphi_k (\varphi_k^T \hat{\Theta}_{k-1}) \\
\hat{\Theta}_k &= \left( I_{n_y} - P_k \varphi_k \varphi_k^T \right) \hat{\Theta}_{k-1} \\
\left\{ \hat{\Theta}_k^T = \hat{\Theta}_{k-1}^T \left( I_{n_y} - \varphi_k \varphi_k^T P_k \right) \right\} .
\end{align*}
\]  

(3-46)

\[\triangle\]

**Insight 3-1.3 (Useful Transformations).** Thanks to (3-38), the parameter estimation error update equation can be written as:

\[
\begin{align*}
\hat{\Theta}_k &= \left( I_{n_y} - P_{k-1} \varphi_k \left( \varphi I_{n_y} + \varphi_k^T P_{k-1} \varphi_k \right)^{-1} \varphi_k^T \right) \hat{\Theta}_{k-1} \\
\left\{ \hat{\Theta}_k^T = \hat{\Theta}_{k-1}^T \left( I_{n_y} - \varphi_k \left( \varphi I_{n_y} + \varphi_k^T P_{k-1} \varphi_k \right)^{-1} \varphi_k^T P_k \right) \right\} .
\end{align*}
\]  

(3-47)

\[6\]the reader is advised to read [79] for the complete proof that concern *persistent excitation*.

\[7\]Assumed to be different from the *parameter estimation error* previously defined.
Last, but crucial transformation is the following. The final Equation in (3-35) can be transformed into:

\[
\mathcal{P}_k \mathcal{P}_k^{-1} = \left( I_{ny} - \mathcal{P}_{k-1} \varphi_k \left( \mathcal{X} I_{ny} + \varphi_k^T \mathcal{P}_{k-1} \varphi_k \right)^{-1} \varphi_k^T \right).
\]  

(3-48)

\[ \triangle \]

**Theorem 3-1.1 (Lyapunov Stability for RLS,).** If the measurement vector sequence \( \varphi_k \) is persistently exciting, then the RLS, is exponentially stable.

**Proof.** Select the following Lyapunov function:

\[
V_k = \tilde{\Theta}_k^T \mathcal{P}_k^{-1} \tilde{\Theta}_k.
\]

(3-49)

The Lyapunov function (3-49) is positive definite and has its minimum for \( \tilde{\Theta}_k^T = 0 \). The derivative is studied as follows:

\[
V_k - V_{k-1} = \tilde{\Theta}_k^T \mathcal{P}_k^{-1} \tilde{\Theta}_k - \tilde{\Theta}_{k-1}^T \mathcal{P}_{k-1}^{-1} \tilde{\Theta}_{k-1},
\]

(3-50)

thanks to (3-47) and (3-48) the derivative is written as:

\[
V_k - V_{k-1} = \tilde{\Theta}_k^T \left( \mathcal{P}_k^{-1} \mathcal{P}_{k-1} \right) \tilde{\Theta}_k - \tilde{\Theta}_{k-1}^T \mathcal{P}_{k-1}^{-1} \tilde{\Theta}_{k-1}
\]

(3-51)

then, substituting the transpose of the equation (3-47) in the derivative, the following is obtained:

\[
V_k - V_{k-1} = \tilde{\Theta}_k^T \left( \mathcal{P}_{k-1}^{-1} \tilde{\Theta}_k - \mathcal{P}_k^{-1} \mathcal{P}_{k-1} \tilde{\Theta}_{k-1} \right)
\]

(3-52)

and therefore

\[
V_k \leq \tilde{\Theta}_k^T \mathcal{P}_k^{-1} \tilde{\Theta}_k = \cdots \leq \mathcal{X}^{k} V_0 = \mathcal{X}^{k} \tilde{\Theta}_0^T \mathcal{P}_0^{-1} \tilde{\Theta}_0.
\]

(3-53)

Hence \( V_k \) is nonincreasing. Thanks to the Defintion (3-1.2), for \( k \geq s \):

\[
||\tilde{\Theta}_k||^2 \leq \left( \frac{1}{\mathcal{X}} \right)^{s+1} \frac{1}{\alpha} \mathcal{X}^s \lambda_{max} \left( \mathcal{P}_0^{-1} \right) ||\tilde{\Theta}_0||^2.
\]

(3-54)

Concluding,

\[
\tilde{\Theta}_k^T \mathcal{P}_k^{-1} \tilde{\Theta}_k \rightarrow 0, \quad \tilde{\Theta}_k \rightarrow O_{ny \times 1}.
\]

(3-55)

Matteo Ciocca  
Master of Science Thesis
3-2 Convergence properties for LTI systems

Concluded the derivations and convergence properties for RLS\(_\kappa\) it was interesting to find relationship with the estimation properties for a SISO LTI systems [53]. First, consider two important assumptions:

**Remark 3-2.1 (A(1-2)). Plant And Signal Assumption:**

- Assumption on the plant: the plant is a SISO LTI system described by a transfer function:
  \[
  \frac{\hat{y}}{\hat{r}} = \hat{P} = k_p \frac{\hat{n}_p}{\hat{d}_p}.
  \] (3-56)
  \(\hat{n}_p\) and \(\hat{d}_p\) are monic, coprime polynomials\(^\text{8}\) of degree \(m\) and \(n\). \(m\) is unknown but the plant is strictly proper \(m \leq (n - 1)\).
- Assumption on the signal: the input \(r(\cdot)\) piecewise continuous and bounded on \(\mathbb{R}^+\).

Although, with only these two assumptions the convergence of the identifier error is guaranteed, nothing can be said about the convergence of the estimated parameters. A new assumption must be introduced.

**Remark 3-2.2 (A(3)). Bounded Output Assumption:**

- Assume that the plant is either stable or located in a control loop such \(r\) and \(y_p\) are bounded.

With these three assumptions the identifier error is guaranteed to converge to zero \(\epsilon(t) \to 0\), for \(t \to \infty\); and the derivative of the parameter error is guaranteed to go to zero \(\dot{\phi}(t) \to 0\), for \(t \to \infty\) [53]. However, the derivative does not say anything about the convergence of the parameter to the real estimate. It can be the case where the distance between \(\hat{\Theta}\) and \(\Theta\) is maintained constant for \(t \to \infty\).

To finally guarantee that the estimated parameter vector will converge to the real parameter vector, a definition must be introduced:

**Definition 3-2.1 (Persistency of Excitation (PE) for continuous signals).** A vector: \(\varphi: \mathbb{R} \to \mathbb{R}^{2n}\) is persistently exciting (PE) if there exist \(\alpha_1 > 0, \alpha_2 > 0, \delta > 0\) such that:

\[
\alpha_2 I \geq \int_{t_0}^{t_0+\delta} \varphi(\tau) \varphi(\tau)^T d\tau \geq \alpha_1 I , \forall t_0 \geq 0 .
\] (3-57)

\(\triangle\)

The definition means that the matrix (the integral of the vector product in time), should be positive semi-definite for two conditions:

\[
\int_{t_0}^{t_0+\delta} \varphi(\tau) \varphi(\tau)^T d\tau = X
\]

\[
\alpha_2 I - X \geq 0
\]

\[
X - \alpha_1 I \geq 0 .
\] (3-58)

\(^8\) Two polynomials are coprime iff they share no roots.
Although the matrix $\varphi(\tau)\varphi(\tau)^T$ is singular $\forall \tau$, the PE condition requires that $w$ rotates sufficiently in space that integral of the matrix $\varphi(\tau)\varphi(\tau)^T$ is uniformly positive definite over any integral of some length $\delta$. And as a final theorem (proof directions in [53]):

**Theorem 3-2.1 (PE and Exponentially Stability).** Let $w : \mathbb{R} \rightarrow \mathbb{R}^{2n}$ a piecewise continuous. If $w$ is PE, Then $\dot{\phi}(t)$ is globally exponentially stable. △

With PE the convergence of the $\hat{\Theta}$ vector to its nominal $\Theta$ is guaranteed: the parameter error $\phi$ converges to zero.

### 3-3 Mention of Subspace identification and offline algorithms

Some Parametric System Identification techniques are not useful for our purpose in online environment estimation. This types of PSI are the ones that work offline. They use batches of data, previously collected, and they try to adapt a parametric model to the input output data. The model is based not only on the input output data at a certain instant $[u(t), y(t)]$, but it uses the previous data too. The algorithm uses models with a vector of data for the input and the output. $[y(t), y(t-1), y(t-2), \ldots, y(t-n), u(t), u(t-1), \ldots, u(t-n)]$. The previous data during the estimation is a strong and powerful element that definitively increase the accuracy of the estimation. However, these methods present advantages and disadvantages that will not be discussed because they are used usually **offline**.

A Non-Parametric System Identification technique is, for example, **Subspace Identification**. Subspace identification methods [78]. These methods are based on the fact that, by storing the input and output data in structured block *Hankel* matrices, it is possible to retrieve certain subspaces that are related to the system matrices of the signal-generating state-space model.

One of the fundamental point of subspace identification is that there is no need to parameterize the model. Hence, subspace methods are not useful to explicitly estimate parameters. However, If the model becomes too complex these methods may be used, because Subspace identification is also used to identify the order of the model and it could be the case that some parameters of the model are lumped together and they cannot be retrieved separately. But for simple model we would like to achieve precise parameters estimation.
3-4 Conclusions

The Chapter aimed to gather knowledge about parametric system identification techniques from the literature focusing on: a clear derivation to the standard form of $RLS_\kappa$ and the important stability analysis of these algorithms. The Lyapunov-based stability analysis determines that parametric system identification can be based on the same stability analysis of Adaptive Control and Adaptive Observer. It is not surprising to find Lyapunov-based stability analysis for Adaptive control too. In fact, most of Adaptive control techniques present always an identifier block - see Chapter (2). In the case of model-based techniques with parameter estimation, the Lyapunov function, presented in (3-49), is always part of closed-loop Lyapunov function. In fact, common Adaptive Control techniques use the derivative of Lyapunov function to find the parameter update law. And, once again, it is not surprising that the (continuous time) parameter update law\(^9\) for Adaptive Control schemes, is usually composed by the identifier error (in the case of tracking, with the reference trajectory) and the regressor.

With the framework selected (and introduced) the direction for the modeling part is clear. A model with unknown parameter should be transformed in an output function linear in the parameter. Doing so, parametric system identification techniques can be used and all the properties explained in this Chapter can be verified and tested.

\(^9\)So the parameter update becomes from $\dot{\Theta}_k - \dot{\Theta}_{k-1}$ to $\dot{\Theta}_k$. 

Master of Science Thesis Matteo Ciocca
Chapter 4

Model

The intent of the Chapter is twofold. It justifies the object model selection and, based on Chapter (3), model-based estimation strategies are tested for the future tactile experiment. The rigid movable object model is at the base of this thesis. The model represents local properties of a one dimensional rigid movable object: mass and friction. To estimate the properties of this object through interaction, based on the framework in Chapter (3), Recursive Least-Squares techniques are tested. The chapter concludes with simulations of the algorithms and important observations about the signals in order to estimate correctly parameters online.

4-1 Introduction

Environment (in this case object) exploration is not an easy task to define. First of all, it is necessary to understand the models to define external objects. Thanks to the literature review in Chapter (2), the purpose of an object model used in practice were understood. An important feature of the object model, given our exploration goal, is the explicit description of the parameters to estimate. In the case of model-based parameter estimation techniques, if the object model does not contain the parameters at the fundamental of our exploration, the object model has no reason to be used.

In the case of compliant surfaces, the local object model describes the surface with its stiffness, $k$, and damping, $b$. Although the model could be created using the coefficient of restitution for an impact (or others mathematical models), the chosen one is characterized by the two properties at the base of the estimation process. True is the fact of a local estimation, but
the experiment conducted in [43, 30, 29] showed remarkable estimation results.

**Model Selection:** Based on this study and on an wide literature review, it was decided that it was worth to try the other class of object models: Rigid Movable Objects (Figure (4-1)). Although the model is simple, its definition does not break any physical laws. The model is described for one dimensional case. At this point, one may doubt that such model cannot work for a higher dimension. This is correct, the model works only for one dimensional case: in reality, a sliding movement it is not constraint in $\mathbb{R}$. However, from the experiments reviewed in the literature, one dimension models are used for the exploration. Examples, where one dimensional model was used for environment exploration, are several [43, 48, 68]. This choice limits the estimation reducing from global to local estimation\(^1\), on the other hand it is a very effective approach.

**Signals Selection:** We anticipate here that the velocity signals will have a key role in the proposed experiments. It was assume that position and velocity signals are measurable. Furthermore, because of an interaction task, the $F$ signal should be available as well: signal from an interaction between a general robotic platform and a touched object. Last, it was decided that acceleration signals were not measurable.

- Position, velocity and force signal (from a force sensor placed on the tip): available during the experiments.

**Strategy Selection:** Three main control strategies for environment exploration were reviewed in Chapter (2). The selection came down to the one that was explicitly created to estimate parameters: PSI. This family of techniques were found suitable to be the first estimation strategy to be tried for the model selected. The goal of PSI techniques it is exactly our goal.

\[
\text{Estimation of unknown parameters.}
\]

---

\(^1\)Local properties are explained in Section (5-1).
4-2 Object Model

Referring to Figure (4-1), the two parameters to estimate are: object mass and object sliding friction (on the surface where it sits). The mathematical object model, introduced in Chapter (2), in our case second order model, is the following:

\[ \Xi := \{ \dot{\eta}(t) = f_e(\eta(t), \varsigma) + g_e(\eta(t), \varsigma)v(t) \} . \] (4-1)

With the vector parameter \( \eta \in \mathbb{R}^2 \) and force input \( v \in \mathbb{R} \) as:

\[ \eta^T(t) = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \end{bmatrix} , \quad v(t) = F(t) . \] (4-2)

Where \( \eta_1(t) \) and \( \eta_2(t) \) are, respectively, the position and the velocity of the object. The symbolic functions are precisely:

\[ f_e^T(\eta(t), \varsigma) = \begin{bmatrix} \eta_2(t) \\ \frac{-F_{friction}(\eta(t), \varsigma)}{m} \end{bmatrix} \] \[ g_e^T(\varsigma) = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} . \] (4-3)

4-2-1 Viscous Friction coefficient and Filter Design

To begin with a simple friction model, a viscous coefficient was chosen for a simple sliding friction force. The friction model is chosen as:

\[ F_{friction}(\eta(t), \varsigma) = b_v \eta_2(t) \Rightarrow f_e^T(\eta(t), \varsigma) = \begin{bmatrix} \eta_2(t) \\ \frac{-b_v \eta_2(t)}{m} \end{bmatrix} . \] (4-4)

Thus, we can define the following parameter vector:

\[ \varsigma^T = \begin{bmatrix} b_v \\ m \end{bmatrix} , \quad b_v : \left[ \frac{Ns}{m} \right] , \quad m : [Kg] . \] (4-5)

In our model, the friction and the mass parameters appear only in the dynamical equation dependent on the velocity and acceleration signals. Thanks to this fact, for the purposes of estimation, the model could be also reduced to one order system. With a change of coordinates: \( \eta_2(t) = y(t) \) and \( \dot{\eta}_2(t) = \dot{y}(t) \), the dynamical model could be transformed into:

\[ \dot{y}(t) = -\frac{b_v}{m}y(t) + \frac{1}{m}F(t) . \] (4-6)

In order to use PSI techniques, we have to define an input and an output signal.\(^3\) This is a very crucial point because the selection must bring to an output signal linear in the parameter to estimate. Our choice for the input signal is the force signal \( F(t) \) (obtained with a force sensor) and the choice of the output signal is the velocity \( y(t) \). The acceleration could not be chosen as output signal because of the assumption on the available signal for the experiment. The next step is to create the output signal linear in the parameter. Because of the assumption on the acceleration, a low-pass filter seems to be a suitable choice to overcome

\(^2\) Always assumed signals as continuous signals in time.

\(^3\) See Chapter 3. We will only consider the algorithms that have a single input and a single output signal.
the problem of acceleration. The goal is to isolate the output signal \( y(t) \) on one side, and to have a function linear in the parameters on the other side.

The reduced order model can be generalized to a dynamical model of two unknown parameter with the following form:

\[
\dot{y}(t) = -a_o y(t) + b_o F(t) , \quad a_o = \frac{b_o}{m} : [\frac{1}{s}] , \quad b_o = \frac{1}{m} : [\frac{1}{Kg}]. \tag{4-7}
\]

Where \((\cdot)_o, o : \text{object}\). Let’s design two simple filters in frequency domain\[4\][53] for the force and velocity signal respectively:

\[
W_1(s) = \frac{1}{s + \lambda} F(s) , \quad W_2(s) = \frac{1}{s + \lambda} Y(s) : \lambda > 0. \tag{4-8}
\]

Or in time domain,

\[
\dot{w}_1(t) = -\lambda w_1(t) + F(t) , \quad \dot{w}_2(t) = -\lambda w_2(t) + y(t). \tag{4-9}
\]

Starting with the object model in frequency domain we can derive the following:

\[
sY(s) = -a_o Y(s) + b_o F(s) \quad sY(s) + a_o Y(s) = b_o F(s). \tag{4-10}
\]

Multiplying the filter structure to both members we obtain:

\[
\frac{1}{s + \lambda} (sY(s) + a_o Y(s)) = \frac{1}{s + \lambda} (b_o F(s)) \quad Y(s) - \left( \frac{\lambda - a_o}{s + \lambda} \right) Y(s) = \frac{1}{s + \lambda} (b_o F(s)) \quad Y(s) = \left( \frac{\lambda - a_o}{s + \lambda} \right) Y(s) + \frac{1}{s + \lambda} (b_o F(s)) . \tag{4-11}
\]

Substituting the filters, we obtain the final expression in frequency or time domain:

\[
Y(s) = (\lambda - a_o) W_2(s) + b_o W_1(s) \iff y(t) = (\lambda - a_o) w_2(t) + b_o w_1(t). \tag{4-12}
\]

This final expression fulfill the important requirement (discussed before) of PSI. To connect Equation (4-12) to the variables of PSI, we can identify (in time domain) the regressor vector up to the instant \( t \):

\[
\varphi^T(t) = \begin{bmatrix} w_2(t) & w_1(t) \end{bmatrix}. \tag{4-13}
\]

The regressor is a vector and not a matrix because of the model selection. And the nominal identifier parameters, corresponding to the real (unknown) parameters:

\[
\Theta^T = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} = \begin{bmatrix} \lambda - a_o & b_o \end{bmatrix}. \tag{4-14}
\]

Hence, the output signal can be re-written as:

\[
y(t) = \varphi^T(t) \Theta = \Theta^T \varphi(t). \tag{4-15}
\]

\[\text{Or Laplace domain [42].}\]

Matteo Ciocca  
Master of Science Thesis
Remark 4-2.1 (Output Dimension). It is clear that the output signal from a general vector description (Chapter 3) is restricted to a single value. The reason is inherent in the choice of model: for example with an ARX model predictor, the output function would become a vector containing previous time output signals. \(\triangle\)

The correctness of this estimate should minimize a standard linear error equation:

\[
\epsilon(t) = \varphi^T(t) \Theta - \varphi^T(t) \hat{\Theta}(t) .
\]

(4-16)

If \(\theta(t)\) is correctly estimated, we are able to retrieve the parameters:

\[
\hat{\theta}_2(t) = \hat{b}_o \Rightarrow \hat{\theta}_2(t) = \frac{1}{\hat{m}} \Rightarrow \hat{m} = \frac{1}{\hat{\theta}_2(t)} .
\]

(4-17)

And,

\[
\hat{\theta}_1(t) = \lambda - \hat{a}_o \Rightarrow \hat{\theta}_1(t) = \lambda - \hat{b}_v \hat{\theta}_2(t) \Rightarrow \hat{b}_v = \frac{\lambda - \hat{\theta}_1(t)}{\hat{\theta}_2(t)} .
\]

(4-18)

Given a certain model, it could be the case that some estimates are functions of two or more parameter (coupled together), and the estimation cannot give explicitly their single (explicit) parameter value. Fortunately, our model does not present this problem. Furthermore, because of the choice of a continuous dynamical model, another approach can be followed as in [51]. This approach justifies (again) the choice of a filter to overcome the problem of the acceleration. Given a general model (SISO) in frequency domain, it can be written as:

\[
A(s)y(t) = B(s)u(t) .
\]

(4-19)

The general structure can be applied to the reduced model in \(y\)-coordinate (Equation (4-7)). Given the model, a filter is applied to the input and output signal:

\[
A(s)H_f(s)y(t) = B(s)H_f(s)u(t)
\]

(4-20)

\[
A(s)y_f(t) = B(s)u_f(t) .
\]

And the following form of the filter is chosen:

\[
H_{f_2}(s) = \frac{1}{(s + p)^n} .
\]

(4-21)

Where \(n\) is the order of the filter. The regressor and the parameter vectors look different from [53]:

\[
\varphi^T(t) = \begin{bmatrix} -y_f(t) & u_f(t) \end{bmatrix} , \quad \hat{\Theta}^T(t) = \begin{bmatrix} \hat{b}_v & \frac{1}{\hat{m}} \end{bmatrix} .
\]

(4-22)

However, this little different on the regressor and parameter vector does not change the overall derivation and algorithm. Hence, we can proceed with [53]. Because of the assumption on the accelerations, we should design the low-pass filter in such a way that it will track exciting signals in order to be useful for a PSI algorithm. Without diving into unnecessary simulations, we should notice that the chosen filter cannot track...
asymptotically any kind of signal. Taking individually the transfer function of the filter naming the input and the output \( U \) and \( Y \) respectively:

\[
H_f(s) = \frac{1}{s + \lambda} = \frac{Y(s)}{U(s)}.
\] (4-23)

It is obvious that the regime value of this function is \( 1/\lambda \). The reason is given by the \textit{final value theorem} (FVT) [42]. The gain, the type\(^5\) and the poles of the transfer function give us the sufficient and necessary information to understand if the transfer function 'can track', for example a constant signal \( u = \bar{u} \). Using the FVT we can write:

\[
\bar{y} = \lim_{t \rightarrow +\infty} y(t) = \lim_{s \rightarrow 0} sH_f(0) \frac{\bar{u}}{s} = \lim_{s \rightarrow 0} \frac{1}{\lambda} \bar{u}.
\] (4-24)

It is evident that the filter cannot track any kind of signal. Complete version of a first-order filter can substituted \( H_f \).

\[
H_f_n(s) = \frac{\mu}{\frac{s}{\lambda} + 1} = \frac{Y(s)}{U(s)}.
\] (4-25)

This transfer function has two tuning variables that work as follow:

1. \( \lambda \) : it is used to define the frequency region for which we can track a signal: define a cut-off frequency, only the signals with less frequency can be tracked by the low pass filter.

2. \( \mu \) : it is used to tune the gain of the filter, thus permitting to decide to completely track or not a (trackable) signal.

Given this new filter, it is necessary to see if it is possible to obtain a regressor and a parameter vector; or better, to understand how the Equation (4-12) changes with new filter. Given the reduced system in frequency domain:

\[
sY(s) + a_o Y(s) = b_o F(s)
\]

\[
\frac{\mu}{\frac{s}{\lambda} + 1} (sY(s) + a_o Y(s)) = \frac{\mu}{\frac{s}{\lambda} + 1} (b_o F(s))
\]

\[
\mu Y(s) - \frac{\mu - \frac{\mu}{\lambda} a_o}{\frac{s}{\lambda} + 1} Y(s) = \frac{\mu}{\frac{s}{\lambda} + 1} (b_o F(s))
\]

\[
\mu Y(s) = \frac{\mu}{\frac{s}{\lambda} + 1} (\lambda - a_o) Y(s) + \frac{\mu}{\frac{s}{\lambda} + 1} b_o F(s)
\]

\[
\mu Y(s) = (\lambda - a_o) W_1(s) + b_o W_2(s)
\]

\[
Y(s) = \left( \frac{\lambda - a_o}{\mu} \right) W_1(s) + \frac{b_o}{\mu} W_2(s).
\] (4-26)

The regressor vector remained the same, but the parameter vector changed accordingly with the filter:

\[
\varphi^T(t) = \begin{bmatrix} w_1(t) & w_2(t) \end{bmatrix}, \quad \hat{\Theta}^T(t) = \begin{bmatrix} \frac{\lambda - \hat{a}_o}{\mu} & \frac{\hat{b}_v}{\mu} & \frac{1}{\hat{m} \mu} \end{bmatrix}. \] (4-27)

With the new filter the \textit{linear-in-parameter} characteristic was not compromised.

\(^5\)The number of integrator define the type of a transfer function: type 0, no integrators.
4-2-2 From a Viscous friction model to a Static friction model

Until now, a viscous friction coefficient was considered as friction model. Such a model is considerably easy to simulate, but the representation of the friction is often poor. Although the simulation is easy to perform, the validity of the viscous model is doubtful. However, in some cases, such full film contacts, the viscous model may offer the best representation of the behavior. In other cases, the viscous model is a poor representation but, by tuning the viscous coefficient, the model can represent such things as damping, in our case "damping of the object", rather well under particular running conditions. It is then necessary to expand the model with a more reliable one. Friction models in literature are divided between static friction models and dynamical friction models. In this section, the friction model passes from a simple coefficient to the addition of Coulomb friction. The main interest of this model expansion, it is the understanding if the parameters of the model can be still decoupled (to be explicitly retrieved), given an output function.

Remark 4-2.2. The Coulomb friction model was preferred instead of the coefficient of dynamic friction: $\mu_d$. The latter is an empirical measurement - it has to be measured experimentally, and cannot be found through calculations. Some value for $\mu_d$ could be assumed from literature, but the use of the Coulomb friction model was preferred because it could fit in a continuous dynamical model.

Coulomb friction, named after Charles-Augustin de Coulomb, is an approximate model used to calculate the force of dry friction. With Coulomb friction only, the present of stiction is not considered. Stiction, i.e. a higher\(^6\) friction force $F_s$ acting while the object is at rest: motion can start only when the applied external forces. The model is expanded to:

$$F_{\text{friction}} = F_c \text{sign}(y) + b_v y$$

(4-28)

Re-calling the reduced model in Equation (4-7), we had:

$$m \ddot{y} + b_v y = F$$

(4-29)

where the $y$ is the velocity of the mass. Coulomb friction is added to the friction force creating the simple Coulomb+viscous friction model. Substituting into the isolated reduced-object model:

$$\begin{cases} F_{\text{friction}} = F_c \text{sign}(y) + b_v y \\ m \ddot{y} + F_{\text{friction}} = F \end{cases}$$

(4-30)

Let’s use the low-pass filter created and let’s see how to re-phrase the continuous system in a linear regressor form: In frequency domain, the system is written as follows\(^7\):

$$sY(s) + \frac{1}{m} [F_c \mathcal{L}\{\text{sign}(y(t))\} + b_v Y(s)] = \left(\frac{1}{m}\right) U(s)$$

$$sY(s) + \left(\frac{F_c}{m}\right) \mathcal{L}\{\text{sign}(y(t))\} + \left(\frac{b_v}{m}\right) Y(s) = \left(\frac{1}{m}\right) U(s)$$

(4-31)

\(^6\)Higher than Coulomb friction force.

\(^7\)\(\mathcal{L}\{\cdot\}\) defines LaPlace transform of \((\cdot)\).
\[ sY(s) + \left( \frac{b_v}{m} \right) Y(s) = -\left( \frac{F_c}{m} \right) \mathcal{L}\{\text{sign}(y(t))\} + \left( \frac{1}{m} \right) U(s) \]

\[ sY(s) + \left( \frac{b_v}{m} \right) Y(s) = -\left( \frac{F_c}{m} \right) \mathcal{L}\{\text{sign}(y(t))\} + \left( \frac{1}{m} \right) U(s) \] (4-32)

\[ (s + a_o)Y(s) = c_o \mathcal{L}\{\text{sign}(y(t))\} + b_o U(s) \]

Applying the same complete low-pass filter used before:

\[ \frac{\mu}{s + \frac{1}{\lambda}} (s + a_o)Y(s) = \frac{\mu}{s + \frac{1}{\lambda}} \left[ c_o \mathcal{L}\{\text{sign}(y(t))\} + b_o U(s) \right] \] (4-33)

After few mathematical steps we can write:

\[ \mu Y(s) = (\lambda - a_o) W_1(s) + (c_o) W_3(s) + (b_o) W_2(s) \]

\[ Y(s) = \left( \frac{\lambda - a_o}{\mu} \right) W_1(s) + \left( \frac{c_o}{\mu} \right) W_3(s) + \left( \frac{b_o}{\mu} \right) W_2(s) \] (4-34)

With the new filter and the new model, the nominal parameter vector, \( \Theta \), is changed to:

\[ \Theta^T = \left[ \frac{\lambda - a_o}{\mu} \frac{b_o}{\mu} \frac{c_o}{\mu} \right] \]

\[ a_o = \frac{b_v}{m}, \quad b_o = \frac{1}{m}, \quad c_o = -\frac{F_c}{m} \] (4-35)

Completed the estimation, a vector of estimated parameter is \( \hat{\Theta}(t) \). The single parameters can be found, after the estimation, solving the following system:

\[ \frac{\hat{b}_o}{\mu} = \hat{\theta}_2 \Rightarrow \frac{1}{\mu \hat{m}} = \hat{\theta}_2 \Rightarrow \frac{1}{\hat{\theta}_2 \mu} = \hat{\mu} \rightarrow \hat{\mu} \]

\[ \frac{\hat{c}_o}{\mu} = \hat{\theta}_3 \Rightarrow -\hat{F}_c \frac{\mu}{m \hat{m}} = \hat{\theta}_3 \Rightarrow \hat{F}_c = -\frac{\hat{\theta}_3}{\hat{\theta}_2} \rightarrow \hat{F}_c \]

\[ \frac{\lambda - \hat{a}_o}{\mu} = \hat{\theta}_1 \Rightarrow \frac{\lambda - \frac{b_o}{m}}{\mu} = \hat{\theta}_1 \Rightarrow \hat{b}_v = \frac{\lambda}{\hat{\theta}_2} - \hat{\theta}_1 \rightarrow \hat{b}_v \] (4-36)

And the regressor becomes:

\[ \varphi^T(s) = \begin{bmatrix} W_1(s) & W_3(s) & W_2(s) \end{bmatrix} \]

\[ W_1(s) = H_f(s)Y(s), \quad W_2(s) = H_f(s)U(s), \quad W_3(s) = \frac{\mu}{s + \frac{1}{\lambda}} \mathcal{L}\{\text{sign}(y(t))\} \] (4-37)

Although the improper use of the third filter, the construction in frequency domain of \( W_3(s) \) does not compromise the velocity input signal, that, for its case, is the output of the \textit{signum function}. The output signal is then again linear in the parameter to estimate and it can be written as:

\[ y(t) = \varphi^T(t) \hat{\Theta} \] (4-38)
4-3 Test with Parameter-Estimation Algorithms

The algorithm test was conducted using an idealized simulation where:

1. The force applied to the object is driven freely to any desired trajectory.

2. The velocity of the object model and the force applied to the object can be directly estimated for each time instant.

Based on Section (4-2-1)-(4-2-2), two experiments were conducted. The first experiment, called Experiment-1, uses the object model with only the viscous coefficient. The second experiment instead, called Experiment-2, uses the object model with Coulomb friction.

The object model nominal (real) parameters were decided as follow. Because a robotic platform will be introduced to interact with an external object, a wood object and a steel object were selected suitable for the interaction. Both objects are considered to have a volume of $1 \text{ [dm}^3\text{]}$. The wood type is: *Picea abies*, having a weight of 0.4 [Kg] and the steel cube has a weight of 7.8 [Kg]. Looking at some empirical tables, it was assumed that the wood cube slides on a concrete surface with a kinetic friction coefficient of $\mu_w = 0.6$ and the steel cube slides on a lubricated steel surface $\mu_d = 0.05$. These coefficients are used to calculate an empirical Coulomb friction force for both objects.

\[
\begin{aligned}
F_c &= m_s g \mu_w^1 = (7.8) \cdot (9.8) \cdot (0.05) \simeq 3.8 [N], \text{ Steel on lubricated steel surface} \\
F_c &= m_w g \mu_d^2 = (0.4) \cdot (9.8) \cdot (0.6) \simeq 2.4 [N], \text{ Wood on concrete surface}
\end{aligned}
\]  

For the viscous coefficient instead, a realistic choice was harder. From an experiment conducted in the literature [82], where a robot slides his finger on a surface to understand the friction properties of the surface, the friction model contained the viscous coefficient. On the other hand, another experiment found in the literature [83] considers the estimation of friction coefficients of several objects, hence, without considering a static or dynamical model. In general, friction models are empirical structures to represent the friction phenomena, dictated by friction coefficients. In our experiments, it was decided to use a model with the viscous coefficient and to try to use a reasonable number for it. Based on [49], the viscous friction coefficient was chosen as $b_v = 0.4 \text{ [Ns/m]}$, for both cubes. Furthermore, The algorithms were tested on a third model with parameters not related to any real object.

The parameters of the filter were fixed for both experiments and the input Force signal was chosen differently for each experiment.

For the Experiment-2, a step $A \cdot \text{step}(t)$ with a certain magnitude $A$ (to be sure that the object will move) was added to the force signal: without movement, the properties of the object (mass and friction) cannot be calculated. The values are showed in Table (4-1).

<table>
<thead>
<tr>
<th>Table 4-1: Force input signal and Filter parameters for Experiment-1 and Experiment-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
</tr>
<tr>
<td>Experiment-1</td>
</tr>
<tr>
<td>Experiment-2</td>
</tr>
</tbody>
</table>
Table 4-2: Experiment-1: object model parameters and nominal \( \Theta \) vector

<table>
<thead>
<tr>
<th></th>
<th>( m ) : [Kg]</th>
<th>( b ) : [Ns/m]</th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>wood object</td>
<td>0.4</td>
<td>0.4</td>
<td>0.99</td>
<td>0.025</td>
</tr>
<tr>
<td>steel object</td>
<td>7.8</td>
<td>0.4</td>
<td>0.9994</td>
<td>0.0012</td>
</tr>
<tr>
<td>third object</td>
<td>10</td>
<td>20</td>
<td>0.98</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 4-3: Experiment-2: object model parameters and nominal \( \Theta \) vector - composed with the Coulomb friction force

<table>
<thead>
<tr>
<th></th>
<th>( m ) : [Kg]</th>
<th>( b ) : [Ns/m]</th>
<th>( F_c ) : [N]</th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \theta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>wood object</td>
<td>0.4</td>
<td>0.4</td>
<td>2.4</td>
<td>0.99</td>
<td>0.025</td>
<td>-0.06</td>
</tr>
<tr>
<td>steel object</td>
<td>7.8</td>
<td>0.4</td>
<td>3.8</td>
<td>0.9994</td>
<td>0.0012</td>
<td>-0.0048</td>
</tr>
<tr>
<td>third object</td>
<td>10</td>
<td>20</td>
<td>2</td>
<td>0.98</td>
<td>0.001</td>
<td>-0.002</td>
</tr>
</tbody>
</table>

**Gradient Law:** Although the simple mathematical model, it was found important to start with the simplest estimation algorithm. It does not exactly belong to the RLS algorithms but it can give an intuition if the filtered model can converge to the right estimates. In fact, this method is a steepest descent approach that try to minimize \( \epsilon(t)^2 \), based on a simple gradient law [53]. The gradient law is related to the identifier error and to the identifier vector.

\[
\dot{\hat{\Theta}}(t) = -g\epsilon(t)\varphi(t) : g > 0, \text{ gradient gain} \quad (4.40)
\]

After a bit of tuning, the estimation could be achieved. An improvement could have been done with the so called normalized gradient law [53], but it was not considered relevant for the purposes of the thesis.

Table 4-4: Test Set-up, Parameters for gradient law

<table>
<thead>
<tr>
<th>( g )</th>
<th>( m ) : [Kg]</th>
<th>( b ) : [Ns/m]</th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \hat{\theta}_1 )</th>
<th>( \hat{\theta}_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Law</td>
<td>10</td>
<td>0.4</td>
<td>0.4</td>
<td>0.99</td>
<td>0.025</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Figure 4-2: Simulation using the gradient law technique. Estimation for the object model: mass and viscous friction. Experiment-1, wood object.
4-3-1 Recursive Least-Squares techniques

Recursive Least-Squares (RLS) algorithms\(^8\) will cover most of the experiments conducted in this Thesis. Generally, RLS algorithm works in discrete time. At first, it was doubted if it was licit to use a discrete algorithm for a continuous time simulation. In fact, in [52], the main processes analyzed were discrete process of ARMA, ARMAX type, as well as other references [42, 54]. Although these algorithms can work with batches of data collected in time, the concern was to use the same techniques for a continuous plant. The problem was related to the sampled input and output data. If the object model was kept as it was without the use of the filters, these algorithms could not be applied to a continuous time model. For example, in [29], the compliant surface model is a continuous model. However, because all input and output signals could be gathered at the same sampled time: \( [F(hT), x(hT), \dot{x}(hT)] \), the discrete algorithms could be used. With the Rigid Movable Object, thanks to the use of the filters, the same situation was achieved: in the sense that input and output signals \( [F(hT), y(hT)] \) \((= [F(hT), \dot{x}(hT)])\) can be gathered at the same sampled time. Without the filters, keeping the acceleration term, \( \ddot{y} (= \ddot{x}) \), the model should have been discretized using Euler or Tustin [84].

In general, RLS algorithms try to minimize a least-square loss function - Chapter (2). What is important to understand is the use of the covariance matrix\(^9\).

**Insight 4-3.1 (Particular Initialization of the Covariance Matrix).** The matrix \( P \) in each algorithm represents an important characteristic of the parameter vector. The matrix can be initialized with the famous inversion matrix lemma [42], see Chapter (3). However, this initialization is avoided if a conventional initialization is used.

\[
P(0) = \alpha I : I \text{ identity matrix, } \alpha \in \mathbb{R} \\
\hat{\Theta}(0) = 0 .
\]

(4-41)

In order to preserve the consistence characteristic of the matrix \( P(t) \), \( \alpha \) must be assume positive. Given this initialization, the \( \alpha \) value embeds an important characteristic related to the parameter vector. The \( P(t) \) is now the variance matrix of the unknown parameter vector \( \hat{\Theta}(t) \). The \( \alpha \) can be chosen with a large or small value: if \( \alpha \) is a small value, it means that we retain the initial parametrization of \( \hat{\Theta}(0) \) not uncertain \((= certain)\). If \( \alpha \) is a big value, it means that we retain the initial parametrization of \( \hat{\Theta}(0) \) very uncertain, and the algorithm will diverge very quick from the initial values of \( \hat{\Theta}(0) \). Given this initialization the cost function should empirically becomes:

\[
V(\hat{\Theta}, t) = \sum_{i=1}^{t} \left[ \epsilon(i)^2 + (\Theta - \hat{\Theta}(0))^T P(0)(\Theta - \hat{\Theta}(0)) \right] .
\]

(4-42)

A important extension of the Recursive Least Squares algorithm is the inclusion of a new variable called the forgetting factor \((\kappa)\) [52]. In Chapter (3) Recursive Least-Squares with Forgetting Factor algorithms were referred with RLS\(_\kappa\).

---

\( ^8 \) The interested reader is referred to [53, 42, 54, 52]

\( ^9 \) pag. 66 [52], pag.48 [42].

Matteo Ciocca  
Master of Science Thesis
Property 4-3.1 (Forgetting Factor). The forgetting factor is a real variable $\kappa \in \mathbb{R}$, and it falls in the interval of values\[^{10}\] between 0 and 1: $\kappa \in (0; 1)$. This coefficient can be thought as a weighing factor. This weight is applied to the past data in the following manner\[^{11}\]: at the instant $t$ the signals have weight 1, the signals at the instant $t-1$ have weight $\kappa$ and the signals at instant $t-2$ have weight $\kappa^2$. The influence of the previous, past, data will decrease on the online estimation. Quantitatively, the interested algorithm data window $(W_d)$ corresponds to the following:

$$W_d = \frac{1}{1 - \kappa}. \quad (4-43)$$

This window represents the relevant signal data that will not be, in principle, attenuated. If $\kappa = 0.99$, the $W_d = 100$. If $\kappa = 0.5$, the $W_d = 2$.

The first $RLS_\kappa$ structure selected was the one for dynamical continuous time systems [51]:

$$\Pi_\kappa := \begin{cases} 
\epsilon(t) = Y(t) - \varphi^T(t)\hat{\Theta}(t) \\
\frac{d}{dt}P(t) = \left(\kappa I - \mathcal{P}(t)\varphi(t)\varphi^T(t)\right)P(t) \\
\frac{d}{dt}\hat{\Theta}(t) = \mathcal{P}(t)\Phi(t)\epsilon(t)
\end{cases}. \quad (4-44)$$

Although the update law resembles characteristic of a gradient law, this structure is casted as least-square estimation with forgetting factor for continuous time systems. The Equation (4-44) is the representation of a Recursive Least Square with Forgetting Factor $RLS_\kappa$ in continuous time. The discrete version (using the time instant $k$) from [51] is:

$$\Pi_\kappa := \begin{cases} 
\epsilon_k = Y_k - \varphi_k^T\hat{\Theta}_{k-1} \\
K_k = \mathcal{P}_k\varphi_k = \mathcal{P}_{k-1}\Phi_k \left(\kappa I + \varphi_k^T\mathcal{P}_{k-1}\varphi_k\right)^{-1} \\
\mathcal{P}_k = \frac{1}{\kappa} \left( I - K_k\varphi_k^T \right)\mathcal{P}_{k-1} \\
\hat{\Theta}_k = \hat{\Theta}_{k-1} + K_k\epsilon_k
\end{cases}. \quad (4-45)$$

The $RLS_\kappa$ representation used in [29], comes from [54]:

$$\Pi_\kappa := \begin{cases} 
\epsilon_k = Y_k - \varphi_k^T\hat{\Theta}_{k-1} \\
\mathcal{P}_k = \frac{1}{\kappa} \left[ \mathcal{P}_{k-1} - \mathcal{P}_{k-1}\varphi_k \left(\kappa I + \varphi_k^T\mathcal{P}_{k-1}\varphi_k\right)^{-1} \varphi_k^T\mathcal{P}_{k-1} \right] \\
L_k = \mathcal{P}_k\varphi_k \\
\hat{\Theta}_k = \hat{\Theta}_{k-1} + L_k\epsilon_k
\end{cases}. \quad (4-46)$$

\[^{10}\]The values 1 and 0 not included.
\[^{11}\]Consistently explained in Section (3-1-1) with the weighting matrix $X$. 

Master of Science Thesis 
Matteo Ciocca
And the last, but not least from [52]:

\[
\Pi_x := \begin{dcases}
\epsilon_k = Y_k - \varphi_k^T \hat{\Theta}_{k-1} \\
r_k = \varphi_k^T \mathcal{P}_{k-1} \varphi_k \\
\mathcal{P}_k = \frac{1}{\mathcal{X}} \left[ \mathcal{P}_{k-1} - \frac{\mathcal{P}_{k-1} \varphi_k \varphi_k^T \mathcal{P}_{k-1}}{1 + r_k} \right] \\
K_k = \frac{\mathcal{P}_{k-1} \varphi_k}{1 + r_k} \\
\hat{\Theta}_k = \hat{\Theta}_{k-1} + K_k \epsilon_k
\end{dcases}
\]  

(4-47)

**Insight 4-3.2.** The formulation found in [52] changes the standard form of the \( RLS_x \) because it does not discount the information at the time instant 0 and it uses a new forgetting matrix from time instant 1:

\[
\mathcal{X} = \text{diag}(x^{k-1}, x^{k-2}, \ldots, x^2, x) .
\]  

(4-48)

The information at time instant \( k \) is discounted by a \( x \) factor. The covariance update becomes:

\[
\mathcal{I}_{k+1} = \left[ x \Phi_k^T \mathcal{X} \Phi_k + x \varphi_{k+1} \varphi_{k+1}^T \right]^{-1} .
\]  

(4-49)

And the parameter update remain invariant:

\[
\hat{\Theta}_{k+1} = \left[ x \Phi_k^T \mathcal{X} \Phi_k + x \varphi_{k+1} \varphi_{k+1}^T \right]^{-1} \left[ x \Phi_k^T \mathcal{X} Y_k + x \varphi_{k+1} y_{k+1} \right] .
\]  

(4-50)

Leading to (4-47).

\( \triangle \)

The experiment took into consideration a reasonable value of \( x \). Reasonable algorithm data windows were used with a forgetting factor falling into the interval of values: \( x \in (0.95; 1) \).

As we previously understood, a \( x = 0.9 \) correspond to \( W_d = 10 \): not much useful.

The simulations showed in this Section are: Experiment-1 for the third object model using the algorithm in Equation (4-44) and Experiment-2 for the steel object model using the algorithm in Equation (4-47). The discrete \( RLS_x \) use a sampling time of \( h = 0.01 \) [s].

**Table 4-5:** Experiment-1: (third) object model parameters and nominal \( \Theta \) vector

<table>
<thead>
<tr>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \hat{\theta}_1 )</th>
<th>( \hat{\theta}_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous ( RLS_x )</td>
<td>0.98</td>
<td>0.001</td>
<td>9.800</td>
</tr>
<tr>
<td>Discrete ( RLS_x )</td>
<td>0.98</td>
<td>0.001</td>
<td>0.9800</td>
</tr>
</tbody>
</table>

**Table 4-6:** Experiment-2: (steel) object model parameters and nominal \( \Theta \) vector - composed with the Coulomb friction force

<table>
<thead>
<tr>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \theta_3 )</th>
<th>( \hat{\theta}_1 )</th>
<th>( \hat{\theta}_2 )</th>
<th>( \hat{\theta}_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous ( RLS_x )</td>
<td>0.9994</td>
<td>0.0012</td>
<td>-0.0048</td>
<td>0.9995</td>
<td>0.0013</td>
</tr>
<tr>
<td>Discrete ( RLS_x )</td>
<td>0.9994</td>
<td>0.0012</td>
<td>-0.0048</td>
<td>0.9995</td>
<td>0.0013</td>
</tr>
</tbody>
</table>

Matteo Ciocca  
Master of Science Thesis
Figure 4-3: Experiment-1, third object. Simulation using the Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS$, both with a forgetting factor $\alpha = 0.99$. The Figure shows the force and velocity profile and below the evolution of the covariance matrix, $P$, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 100$. 
The Figure shows the estimation of the \( \theta \) vector, for both the algorithms. The Figure 4-4: Experiment 1, third object. Simulation using the Continuous RLS and the discrete version RLS, both with a forgetting factor \( \kappa = 0.99 \). The Figure shows the estimation of the \( \theta \) vector, for both the algorithms.
Figure 4-5: Experiment-2, steel object. Simulation using the Continuous $RLS_{Cont}$ and the discrete version $RLS$, both with a forgetting factor $\kappa = 0.99$. The Figure shows the force and velocity profile and below the estimation of the $\hat{\Theta}$ vector, for both the algorithms.
Figure 4-6: Experiment-2, steel object. Simulation using the Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS$, both with a forgetting factor $\kappa = 0.99$. The Figure shows the evolution of the covariance matrix, $\mathcal{P}$, initialized for both algorithms with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. 

Matteo Ciocca  
Master of Science Thesis
4-4 The Problem of Persistent Excitation (PE)

An important aspect of these simulations was the concept of persistent excitation (PE). As explain in Chapter (3), the PE property is the essential characteristic in order to obtain the correct parameter estimate using Parametric System Identification techniques, in our case: online model based estimation techniques under the name of Recursive Least Square with forgetting factor.

Referring to the Chapter (3), the definition of PE takes into account the regression matrix, in our case a vector: \( \varphi \in \mathbb{R}^m \). Two different regression vectors appear for two experiment. The Experiment-1 uses a regression vector of dimension \( m = 2 \), the vector is composed by the input, force signal (\( F \)) and the output velocity signal (\( y \)), both at the current instant time. The Experiment-2 has a regression vector of dimension \( m = 3 \), it contains the same signal of Experiment-1 but also it contains the signum function of the velocity sign (\( y \)).

Definition (3-2.1) in Chapter (3), can be used for Experiment-1 because of the linearity of my system, Experiment-2 deals with a nonlinear system and the definition no longer holds. In fact, being \( m = 3 \), it does not satisfy the characteristic of \( \varphi : \mathbb{R} \rightarrow \mathbb{R}^{2n} \), with \( n \) order of the system. However, assumed to have a system linear in the parameters and to use an RLS technique, the Theorem (3-2.1), for the stability and convergence of the parameters, has the same characteristic of the theorem of exponential stability for RLS \( \kappa \) [79].

**Theorem 4-4.1 (Exponentially Stability for RLS\( \kappa \)).** If the sequence \( \Phi_k \) is PE, Then RLS\( \kappa \) is exponentially stable. △

PE it is not a simple concept, and it could be challenging to accomplish in our scenario\(^{13}\).

An explanation of PE is the following:

Suppose the the information matrix is an n-by-n square matrix, \( I \in \mathbb{R}^{n \times n} \), then over time \( \Phi_k \) should uniformly "visit" all directions in the space \( \mathbb{R}^n \). This effect is coupled to the notion of persistent excitation [79].

An interesting aspect was found in [79]. An interesting condition states:

if \( \Phi_k \) is bounded, persistency excitation of \( \Phi_k \) is necessary for \( P_k \) to be bounded.

Assumed that \( \Phi_k \) is bounded, then the necessary condition for \( P_k \) to be bounded can be studied. Instead checking the PE of \( \Phi_k \), we can directly check, thanks to the construction of any RLS\( \kappa \) algorithms, if \( P_k \) is bounded during the experiment. If the latter is true, we can state that the \( \Phi_k \) is PE. If \( P_k \) is not bounded we are not able to declare if \( \Phi_k \) is PE or not, thus we cannot say if the estimated parameters of a selected algorithm will converge to the nominal value of the parameters (the real parameter value). It is powerful then to check if the matrix \( P_k \) is bounded.

In our experiment, a signal in \( \Phi_k \) is generated, the force input: the signal is chosen bounded in time. The velocity signal, belonging to the regressor vector, should be bounded in time as well. This is correct thanks to the object model properties. In fact, the object model has friction, and because we are pushing it backward and forward, we are able to state that the \( \Phi_k \) vector will be always bounded in time.

---

\(^{12}\)In this section, the dimension is not coherent with the one stated in Section (3-1).

\(^{13}\)It will be observed in Chapter (5)
Because the boundedness condition of $\Phi_k$ is always satisfied, we can directly check the boundedness condition on $P_k$. As a practical example, a single step input force signal is used. Furthermore, as output signal (that will compose the vector $\Phi_k$), we will expect a low-pass filter kind of behavior, hence $\Phi_k$ bounded.

Satisfied the condition on $\Phi_k$, the estimation of the nominal parameter vector of the object model can be studied through the $P_k$. A new simulation was conducted for Experiment-1. There are several methods to check if a matrix is bounded. The one used here is the analysis of the trace, $\text{tr}(\cdot)$, of $P$ (here in continuous time): if the trace goes to infinity $\to +\infty$, then the matrix $P$ is unbounded\(^{14}\). Infinity is assumed to be an unbounded condition for the duration of the experiment: matching the unboundedness condition will state that for a longer interval of time (after then end of the experiment) the $\text{tr}(P)$ will continue to grow. Both Figures (4-8) and (4-9) show that the trace of $P_k$ is unbounded and for this reason "nothing can be said about the PE of the $\Phi_k$": $\Phi_k$ can or cannot be PE. In fact, we obtain different results changing the parameter of the object model. It can happen that for a model the $\Phi_k$ is PE and for another model $\Phi_k$ is not. Although a single step is well-know to be a not PE signal \([85]\), It can happen for some unknown system that a single step is rich enough to estimate its parameters.

Concluding, in case $P_k$ is unbounded, it is be better to repeat the experiment until $P_k$ is bounded: using for example different signals that will compose $\Phi_k$.

\(^{14}\)Trace as the sum of eigenvalues: no eigenvalue should go to infinity for the covariance matrix.

**Table 4-7:** Given the boundedness condition on $\Phi$, called $C$. The PE of $\Phi$, called $A$, is a necessary condition for $P$ to be bounded, called $B$. Highlighted in red are the conditions where I would like not to be. Because if $B$ is not satisfied, I do not know if $A$ is correct or not: because in both cases (when $A$ is true or false), the necessary condition is satisfied (true).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>$B \rightarrow A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
Figure 4-7: Simulation for the PE problem. Force input is a single unit step. Two cases: above the wood object, below the third object.
Figure 4-8: Simulation for the wood cube using the continuous version of $RLS_\kappa$. Force input is a single unit step. The trace of covariance matrix $P(t)$ is unbounded, but the estimates converge to their nominal/real value. The single unit force step was persistent excited enough for the wood cube model. This is the third case in the logic table.
Figure 4-9: Simulation for the third object using the continuous version of \( RLS_{\kappa} \). Force input is a single unit step. The trace of covariance matrix \( P(t) \) is unbounded and the estimates do not converge to their nominal/real value. The single unit force step was *NOT* persistent excited enough for the third object. This is the fourth case in the logic table.
4-5 Conclusions

The modeling part is crucial for a good parameter estimation. In our case, the model defines the parameters to estimate. If they do not represent real object properties, convergence of the estimates to some values will not represent any real physical property. In our case, these two models are assumed to represent two (or three) real physical properties.

The one dimensional model of a Rigid Movable object focuses on two properties: its mass and its friction. There is no description on the geometry of the object because this could increase the complexity of the model. Lacking of a geometry description and because of its reduction to one dimensional case, the estimation of its properties are restricted to a local estimation, instead of a global estimation. Nevertheless, it was explained the friction models presented are not complete, in a dynamical sense, and far from a complete description of the friction phenomena. However, it was decided to do not expand the model description, but instead to focus on the estimation strategy.

The problem of identification was restricted to the use Parametric System Identification algorithms only. Tests, on the model (alone), proved the convergence of the estimation for the mass and the friction, given an excited signal. The techniques belong to the family of Recursive Least Square (RLS) techniques. The experiments were conducted to understand the implementation and the effectiveness of the algorithms. Last, but not least, the problem of Persistent Excitation (PE) was discussed. Although this problem was studied in the case of RLS, the PE is, most of the time, the fundamental problem of the parameter convergence (to their real value) for several algorithms that regards parameter estimation; between them: Adaptive Control, Adaptive Observer [51, 53, 86, 87].

In the next Chapter, the algorithms studied will be tried on a novel simulated experiment for the extraction of the object properties, in our case: its mass and 'its' friction.
Chapter 5

Experiment

During the literature study several experiments were conducted to extract properties through a tactile interaction using the compliant surface model. On the other hand, no model-based experiments with a continuous estimation for a rigid movable object were found. Experiments that combine object models and online model-based techniques. It is then the scope of this Chapter to introduce the characteristics, assumptions, results and observations of a novel simulated experiment to extract properties of a rigid movable object through a continuous tactile interaction. The estimation aims to extract two properties from the object: its mass and its sliding friction (between the object and the unknown surface where it sits).

5-1 Experiment Description

Accomplished the test phase, for the algorithms chosen in Chapter (3)-(4), a simulated experiment is needed. In this experiment a controlled system is used to simulate a tactile interaction with a rigid movable object, and, during the interaction, an identification algorithm should extract object properties such as: mass and friction (between the object and the surface which is in contact with it). To reduce the complexity of this experiment and to focus on the property extraction, relevant parts of the controlled platform that will interact (through touching) with the object will be modeled. A robot finger was selected as reasonable robotic part to interact with the selected model: rigid movable object model. The experiment is strictly related to the dimension of the object model chosen. Being the model of the object in one dimension, it is reasonable to choose to model the robot finger in one dimension too. However, in a general picture, the finger was assumed to be part of a n-link manipulator.

Without creating modeling complexity on the side of the robotics platform, the finger was modeled as an end-effector: the finger is a simple actuated mass. Truth is, to be consistent, that the mass should contain a damping factor to represent the internal friction of the robotic platform [88]. This is also helpful from the point of the stability of the single finger only:
being a simple mass presents a double integrator and hence is unstable, which is false in reality\footnote{Imagine to give a push to a robotics structure: an \( n \)-link manipulator. It will not travel an endless trajectory.}.

An important assumption on the end-effector of the robot is the measurable signals. Usually, in a real-time robotic platform, position signals should be always available (measurable). However, sometimes, velocity signals cannot be directly measured: even worse for acceleration signals. In our case, because the velocity signal possesses a key role in the proposed experiments, it is assumed the velocity signal to be measurable. Furthermore, the force signal \( F \) is assumed to be available as well: signal from an interaction between the robotic finger and object touched. Concluding, it was decided the impossibility to measure the acceleration signals.

**Remark 5-1.1 (Measurable Signals).** End-effector signals:

- Position signal, \( x(t) \).
- Velocity signal, \( \dot{x}(t) \).
- Force signal (from a force sensor at the tip of the end-effector), \( F_{\text{ext}}(t) \).

\[ \triangle \]

**5-1-1 How the experiment works**

The choice of one dimensionality has an important feature: the rigid movable object can slide in an hypothetical surface only in two directions, backward and forward. In our scenario, the object is placed on the right side of the finger. The right side is chosen to be a positive direction. If the finger moves to the right, it will encounter the object and it will push it. A single pushing finger was chosen to create a contact/non-contact phase. The single finger choice allowed to focus on the quality of the identification algorithms. It is true that a single pushing finger is completely unlikely in reality. Chosen a robotic platform as a \( n \)-link manipulator, it should be reasonable to assume a defined workspace: where the finger is allowed to move (and to reach possible positions). The pushing finger cannot push the object for an infinite path. Thus, with a pushing finger, a simulation time limits the pushing action and tries to re-create a semi-real scenario.

**Definition 5-1.1 (Object and Interaction Assumptions).** Because the experiments were conducted in a simulation environment \footnote{89}, some assumptions were made on the interaction with the object and on the structure of the object:

- The object is made by compact material: not hollow.
- The object is made of one material only.
- The force applied to the object is perpendicular to its flat (assumed) surface: \( F(t) = F_n(t) \), a normal force applied only.
- The force is applied in such a way that the object cannot tilt.
The object is modeled in one dimension, \( \mathbb{R} \). It is allowed to move in two directions only: forward and backward.

The object is modeled in adimension. However, geometrical conditions can be made to establish contact/non-contact.

Insight 5-1.1 (On Local Properties Thorough a Tactile Interaction). Imagine, with closed eyes, to touch a wall with one hand. If this hypothetical wall is touched in a single point, mathematically speaking it is possible to estimate local properties of the wall. In fact, being with eyes closed, it is not possible to establish where these estimated properties of the wall, i.e. its stiffness, are unique for the entire surface. It is possible that the wall is composed by different characteristics in different zones and our estimation represents one of them: the stiffness of the wall could change in location. It is mathematically correct to call our object model a local model. Because environment informations are extracted given a single point. Although a global model estimation would be preferable, local properties are not an disadvantage. They are less computationally expensive and they can be eventually collected after few explorations and consequently, fitted in a global object model.
5-2 Hybrid Architecture, Trajectory and Controller Selection

![Diagram of a one-finger pushing experiment](image)

**Figure 5-1:** One-finger-pushing experiment. The reference frame $\Psi_0$ on the left. The positive movement direction to the right. The position of the finger $x$ and the position of the rigid movable object $y$. When the robotic finger and the object pass into the contact phase, a force sensor (spring and damper in parallel) is added to generate a force signal $F$.

**Hybrid Architecture.** As explained before, to test the quality of the algorithms, a single finger is chosen. The scenario is showed in Figure (5-1). To simulate an interaction with the object, an hybrid architecture is constructed as such:

![Diagram of contact and non-contact phases](image)

The position condition to switch from a contact to a non-contact phase is sufficient, but it is not complete. Given the model of the force sensor and the position condition, it is expected that there will be two additional internal phases during the contact condition. The force sensor will pass through a compression and a release phase. The compression is the key phase when the force signal is generated, meanwhile the release phase should not generate a
sticky effect. It means that the switching position condition can generate force value that will enforce the object and the finger to stick together. This is a well-known simulation problem [90], and to solve it, the switching condition should incorporate the force signal. The latter decision on the force signal was not incorporated into the switching condition because it did not improve any experiment. On the other hand, a careful attention on the force signal was conducted to ensure that it would not take absurd value.

**Force Sensor.** The force sensor was modeled (in one dimension) with a spring and a damper in parallel. This modeling was justified by the literature [88, 91]. Furthermore, as explained in Chapter (2), at least a spring would be added between two colliding masses to avoid infinities and to create a continuous behavior. With a damper, a complete continuous description is achieved taking into account the velocity signal. The decision of the spring and damper values were not easy. Empirical parameters for a soft compliant sensor should be selected. From [23], the stiffness of a compliant surface was estimated to be $\approx 400 [N/m]$. Based on that, it was decided that the soft force sensor should be a little bit stiffer than that: $\approx 600 [N/m]$. The damper was decided on the base of a mass-spring-damper system with a unit damping factor and a unit mass.

$$K_{sensor} = 600 \ [N/m], \ B_{sensor} = 2\sqrt{K_{sensor}} \ [Ns/m]. \quad (5-1)$$

**Robot Finger Mass.** A not trivial choice was the robot finger mass. Without the use of a real platform, the choice can lead to an unreal situation. However, because of the restriction to a simulated environment, it was decided to have a free choice on the mass of the robot finger. With the hope to guarantee a semi-real scenario where the robot can bump and push the object, it was decided the mass did not represent only a finger, but a concentrated parameter representing a sum of components to arrive at the established weight.

$$m_{robot} = 2 \ [Kg]. \quad (5-2)$$

**Trajectory Selection.** Trajectory selection was a very delicate step. First, the trajectory should be created to ensure a tactile interaction with the object. Being in one dimensional case, referring to Figure (5-1), it is obvious that a simple forward movement onto the right direction will ensure the contact with the object. Second, the trajectory should stimulate the identification algorithm to converge to the nominal parameters of the object model. The stimulation, or persistent excitation, is one essential requirement with the use of Parametric System Identification techniques. To ensure that the tactile interaction does not become a single ’bump’ or a brutal hit to the object, a polynomial (position) trajectory was considered a good candidate. The important characteristic of a polynomial trajectory is its approach to the object. Increasing the order of the polynomial, the smoothness of the trajectory increases. The trajectory was chosen starting from an $7th$ order acceleration profile as described in [92, 93]:

$$a(t) = -p \cdot t^2 \cdot (t - 0.5t_f)^3 \cdot (t - t_f)^2. \quad (5-3)$$
Where \( t_f \) is the time when the motion ends. The acceleration is then integrated to obtain the velocity profile:

\[
v(t) = \left[ \frac{1}{8} t^8 - \frac{1}{2} t^7 + \frac{19}{24} t^6 - \frac{5}{8} t^5 + \frac{1}{4} t^4 - \frac{1}{24} t^3 \right] .
\] (5-4)

And, as last step, to obtain a position profile:

\[
S(t) = \left[ \frac{1}{72} t^9 - \frac{1}{16} t^8 + \frac{19}{168} t^7 - \frac{5}{48} t^6 + \frac{1}{20} t^5 - \frac{1}{96} t^4 \right] .
\] (5-5)

The powerful feature of the polynomial trajectory is the choice of a coefficient \( p \), irrespective of the polynomial order. The coefficient allows to choose the maximum velocity, \( v_{\text{max}} \), and the final displacement, \( \Delta S \), given the motion-end-time \( t_f \).

\[
v_{\text{max}} = \frac{pt_f^8}{6144} , \quad \Delta S = \frac{pt_f^9}{10080} .
\] (5-6)

Manipulating the Equations (5-6), a desired displacement \( \Delta S_d \) can be chosen: generating the value of the \( p \) coefficient. The trajectory generated by choosing a desired displacement: \( \Delta S \) and a desired \( t_f \). The polynomial trajectory is showed in Figure (5-2) with a phase divisor, representing the division between the Run-up phase (first phase) and the Brake phase (second phase). However, an important question can arise:

Is the polynomial profile a good trajectory that will satisfy the persistent excitation requirement, during the interaction, to guarantee the convergence of the identification algorithm?

The question is not so trivial to answer. Meanwhile a sinusoidal (or a random) motion profile (and velocity profile too) is proved to be effective in [56, 57], satisfying the persistent excitation requirement, to estimate properties of an external compliant surface, in our experiment, the "external environment" is able to move and to lose indefinitely contact with the robotics platform used for the estimation. Without digressing on possible selections, the polynomial trajectory was considered a good starting point because it will allow to move the object. However, given the velocity profile, highlighted in Figure (5-2), one may ask if it will be sufficient to be persistently excitated with a force signal coming from the interaction. The velocity is a simple Plateau profile and it may not be able to generate (from the interaction) a good force profile: used in the regressor vector.

Inspired by the experiment in [29], it was decided to expand the polynomial trajectory with the addition of a sinusoidal.

\[
S_2(t) = Asin(\omega t) .
\] (5-7)

Two set of parameters were chosen for the sinusoidal:

<table>
<thead>
<tr>
<th>Table 5-1: Parameter of the sinusoidal</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>( S_1^2(t) )</td>
</tr>
<tr>
<td>( S_2^2(t) )</td>
</tr>
</tbody>
</table>

Matteo Ciocca Master of Science Thesis
Adding the sinusoidal to the position profile:

\[ S(t) = -p \left[ \frac{1}{72} t^9 - \frac{1}{16} t f t^8 + \frac{19}{168} t^7 f t^7 - \frac{5}{48} t^5 f t^6 + \frac{1}{20} t^4 f t^4 - \frac{1}{96} t^2 f t^2 \right] + A\sin(\omega t) \] \quad (5-8)

Successively, the sinusoidal was differentiated w.r.t. time for the velocity and the acceleration:

\[ v(t) = -p \left[ \frac{1}{8} t^8 - \frac{1}{2} t f t^7 + \frac{19}{24} t^6 f t^6 - \frac{5}{8} t^4 f t^4 + \frac{1}{4} t^2 f t^2 - \frac{1}{24} t^2 f t^2 \right] + A\omega \cos(\omega t) \]

\[ a(t) = -p \cdot t^2 \cdot (t - 0.5 t_f)^3 \cdot (t - t_f)^2 - A\omega^2 \sin(\omega t) \] \quad (5-9)

The complete profile is showed in Figure (5-2). Sinusoidal movement could allow a relaxation and compression phases of the force sensor during the interaction: these two phases could generate a sinusoidal profile for the force signal too, a sinusoidal that cannot be achieved by the simple polynomial trajectory. Because the object is assumed to slide and to not roll, both trajectories are suitable for the experiment with only one finger.
Figure 5-2: The polynomial trajectory. The generation of the trajectory (position, velocity and acceleration) was achieved by choosing $\Delta S_d = 3 \text{ [dm]}$ and $t_f = 20 \text{ [s]}$. The plots show the simple polynomial trajectory and the same trajectory with the addition of a selected sinusoidal: $S_1^2(t)$ and $S_2^2(t)$. 

Matteo Ciocca
Master of Science Thesis
Controller Design. In literature (given a simple mass \( m \) interacting with the environment) a position-based *impedance controller* is widely used for such kind of tasks \([94, 95]\):

\[
    u = \frac{m}{M_d} [\ddot{x}_d + k_d (\dot{x}_d - \dot{x}) + k_p (x_d - x)] + \left( \frac{m}{M_d} - 1 \right) F_{ext}.
\]  
(5-10)

The impedance controller aims to use the force signal, gathered from a force sensor, in order to stabilize the system with an external environment or to use the force extracted from the environment to push it. An impedance control architecture is showed in Chapter 2, Figure (2-10). The impedance strategy is widely used for *compliant* not movable environments. A force-based *impedance controller* has similar control purposes \([94]\):

\[
    u = \frac{m}{M_d} [k_f (f_d - F_{ext}) - k_d \dot{x}] - F_{ext}.
\]  
(5-11)

Where the error taken into consideration is not based on a position profile, but on a force profile: the error is minimized when the external force, from the force sensor, matches the desired force. A third architecture for interaction tasks is the so-called *Damping Injection* architecture \([96, 97]\). Its dynamic controller corresponds to:

\[
    u := \begin{cases} 
    \dot{x}m + F_{ext} - k(x_c - x) = 0 \\
    \ddot{x}_c m - k(x - x_c) + b \dot{x}_c - k_c (x_d - x_c) = 0
    \end{cases}
\]  
(5-12)

Where \((x_d - x)\) is the error used in the feedback-loop. The latter does not take into account the velocity profile \((\dot{x})\), and try to stabilize the system using a *virtual spring-damper-mass* \((k_c, b_c, m_c)\), given a possible interaction described by the force signal \(F_{ext}\). This interacting architecture does not aim to excite the environment but to stabilize with it. Others architectures, described in Section 2-2 of Chapter 2, try to estimate external parameters but for stabilization purposes. Example for compliant surfaces, the stiffness value is used to deform the virtual spring of the controller.

In our case, the external environment must be excited. The object must move and the robot finger must move with it. A motion control strategy is suitable for this kind of task. In general, position controllers are extremely avoided in practical interaction tasks. An important reason is the safety of the robotic platform. With a simple position control, there is no guarantee to interact without damaging the interacting robot components. The reason it is inherent in the control architecture.

\[
    u = m[K_p (x_d - x) + K_d (\dot{x}_d - \dot{x}) + \ddot{x}_d].
\]  
(5-13)

A position control takes into consideration the internal dynamics of the system and it does not use any external environment properties. In fact, the error between the desired position and the measured one does not relate to an external behavior, such as the force signal \(F_{ext}\). A second important drawback of position control with an interaction task is the saturation of the actuators. Given a desired unreachable position in space, the position error, \(e = x_d - x\), could increase in time and the control action could take high unwanted values saturating the actuators, i.e. reaching the maximum voltage input value allowed for an actuator.

On the other hand, the position control can be expanded with the architecture in Equation
(5-10). In general, position-based impedance controllers contain the same goal of a position based controller but with the addition of a force component used to interact with an unknown environment. Because the trajectory is essential to characterize a possible exciting signals, impedance based controller was selected. Furthermore, the experiment will consider some important assumptions.

Property 5-2.1 (Assumption for the control architecture). The chosen control architecture has the following assumptions:

1. Saturation of the actuators will never been reached. $u \in (u_{\min}, u_{\max})$.

2. The robotic platform will be able to move the object with a admissible control input, given the first assumption.

3. Although the movable objects are stiff, they will not damage the robotic platform during the interaction, even with the maximum control input.

△

Seems that the established assumptions make the simulated experiment far from a practical implementation. However, it is important to remember the dimension of the objects used during the experiment. Only $[dm]$ dimensional objects are used and they do not weight extremely much. Given an industrial manipulator with a compliant end-effector, the task is to move an object with a $\approx 10\%$ of its entire mass: although in the experiment only the robot finger is modeled. Control input values will be not extremely high, the control action needed to move them will not damage the robotics platform.
5-3 Pushing finger Experiment

The first experiment uses a single pushing finger to explore the rigid movable object properties. Two different object models are tested: the model with only the viscous coefficient and the model with the addition of the Coulomb friction. The actuated system and the object model are formally introduced:

\[ \Sigma_{\text{finger}} := \{ \dot{x}(t) = f_s(x(t), \sigma) + g_s(x(t), \sigma)u(t) \} \]

\[ \Xi_{\text{object}} := \{ \dot{\eta}(t) = f_e(\eta(t), \zeta) + g_e(\eta(t), \zeta)v(t) \} \].

Where \((\eta, x) \in \mathbb{R}^2, (u, v) \in \mathbb{R}\) and:

\[
\begin{bmatrix}
\eta^T(t) = \left[ \eta_1(t) \ \eta_2(t) \right], \ v(t) = F(t) \\
x^T(t) = \left[ x_1(t) \ x_2(t) \right], \ u(t) : [N] \\
\sigma = \left[ M \right], \ \zeta = \left[ m \ \zeta_{\text{friction}} \right]
\end{bmatrix}
\]

\[
\begin{align*}
& f_s^T(x(t), \sigma) = \begin{bmatrix} x_2(t) \ 0 \end{bmatrix}, \ g_s^T(x(t), \sigma) = \begin{bmatrix} 0 & 1 \\ M & 0 \end{bmatrix} \\
& f_e^T(\eta(t), \zeta) = \begin{bmatrix} \eta_2(t) - \frac{F_{\text{friction}}}{m} \end{bmatrix}, \ g_e^T(\zeta) = \begin{bmatrix} 0 & 1 \\ m & 0 \end{bmatrix}.
\end{align*}
\]

When the finger and the object are in contact, the force sensor is added to between them and formally the following system is obtained:

\[ \Upsilon_{\text{coupled}} := \{ \dot{\rho}(t) = f_{s,e}(\rho(t), \nu) + g_{s,e}(\rho(t), \nu, u) \} \].

Where,

\[
\begin{bmatrix}
\rho^T(t) = \left[ \eta_1(t) \ \eta_2(t) \right], \ \nu^T = \left[ \sigma \ \zeta \right] \\
x_2(t)
\end{bmatrix}
\]

\[
\begin{align*}
& f_{s,e}(\rho(t), \nu, u) = \begin{bmatrix}
\frac{k_s}{M}[\eta_1(t) - x_1(t)] + \frac{b_s}{M}[\eta_2(t) - x_2(t)] \\
\frac{k_s}{m}[x_1(t) - \eta_1(t)] + \frac{b_s}{m}[x_2(t) - \eta_2(t)] - \frac{F_{\text{friction}}}{m}
\end{bmatrix}, \ g_{s,e}(\rho(t), \nu, u) = \begin{bmatrix} 0 & u \\ M & 0 \end{bmatrix}.
\end{align*}
\]

The force sensor, modeled with a spring \((k_s)\) and a damper \((b_s)\) in parallel, is the coupling element between the object and the actuated system. Two functions are chosen (for the experiments conducted) to express the friction behavior:

\[ F_{\text{friction}} : F_f = b_v \eta_2(t), \ F_f = F_e \text{sign}(\eta_2(t)) + b_v \eta_2(t). \]
Both simple static friction models will be evaluated. The control architecture is chosen as position-based impedance controller and the desired path is generated with a polynomial trajectory - Section (5-2). Two algorithms from Chapter (4) are tested on three different combinations of mass and friction models. Two Recursive Least Squares (with forgetting factor) algorithms were used. One in continuous time:

\[
\Pi_{\kappa} := \begin{cases}
\epsilon(t) = Y(t) - \Phi^T(t)\hat{\Theta}(t) \\
\frac{d}{dt}P(t) = \left(\kappa I - P(t)\Phi(t)\Phi^T(t)\right)P(t) \\
\frac{d}{dt}\hat{\Theta}(t) = P(t)\Phi(t)\epsilon(t)
\end{cases}.
\] (5-19)

And another in discrete time:

\[
\Pi_{\kappa} := \begin{cases}
\epsilon_k = Y_k - \Phi_k^T\hat{\Theta}_{k-1} \\
r_k = \Phi_k^TP_{k-1}\Phi_k \\
P_k = \frac{1}{\kappa}\left[P_{k-1} - \frac{P_{k-1}\Phi_k\Phi_k^TP_{k-1}}{1 + r_k}\right] \\
K_k = \frac{P_{k-1}\Phi_k}{1 + r_k} \\
\hat{\Theta}_k = \hat{\Theta}_{k-1} + K_k\epsilon_k
\end{cases}.
\] (5-20)

In discrete time. The experiment consider the robot starting from the origin of the reference frame and the object positioned at 0.3 \textit{dm} distance from it.

With respect to Table (4-5) and (4-6), the following figure shows two experiments: Experiment-1 with the steel object and Experiment-2 with the wood object. The controller parameters were chosen as:

<table>
<thead>
<tr>
<th>Table 5-2: Filter and Controller Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
</tr>
<tr>
<td>Experiment-1</td>
</tr>
<tr>
<td>Experiment-2</td>
</tr>
</tbody>
</table>

Both Experiments use the polynomial trajectory with the addition of $S_2^\circ(t)$ from Table (5-1) and for the discrete version of $RLS_{\kappa}$ the sampling time was chosen $h = 0.01 \text{ [s]}$. 

Matteo Ciocca

Master of Science Thesis
Figure 5-3: Experiment-1, steel object. Relevant measurements (above) of the robot finger (\(\cdot\))_r, the object (\(\cdot\))_o and desired trajectory (\(\cdot\))_d: position, velocities. Filtered signals and control input (below).
Experiment

Figure 5-4: Experiment 1, steel object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ for Continuous RLS and the discrete version $\text{RLS}_\kappa$ algorithms, both with a forgetting factor $\kappa = 0.99$.
Figure 5-5: Experiment-1, steel object. Evolution of the trace of the covariance matrix (ZOOM-IN: plots on the right), $P$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\text{np}}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 100$. 
Figure 5-6: Experiment-2, wood object. Relevant measurements (above) of the robot finger (·)ₚ, the object (·)ₒ and desired trajectory (·)ₚ: position, velocities. Filtered signals and control input (below).
5-3 Pushing finger Experiment 2, wood object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for Continuous RLS and the discrete version RLS algorithms, both with a forgetting factor $\kappa = 0.99$. 

Figure 5-7:
Figure 5-8: Experiment-2, wood object. Evolution of the trace of the covariance matrix, $P$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\kappa}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 0.01$. 
Although the results of Experiment-1 are a disaster, they are extremely important and give us useful insights to improve the estimation.

**Simulated interaction Evaluation.** From Figure (5-6) the force profile highlights the choice of the position-based hybrid structure. In fact, little negative force peaks during the interaction are evident. Physically speaking, these peaks represent the force sensor that is not releasing the two masses (robot and object), instead the sensor is trying to keep them together. This is absolutely not correct, the force sensor is used to extract a force profile and not to be a connecting part between mechanical components. To overcome this problem a simple threshold is applied to the filtered force profile: $F \geq 0$. The force profile is used for two elements: the estimation architecture (in the regressor vector) and for the impedance controller. In practice, these two elements should use a reasonable and possibly realistic force profile so the threshold is fundamentally useful.

![Graph 1](image1.png)

![Graph 2](image2.png)

**Figure 5-9:** Experiment-1, steel object. Closer look at the estimation of the $\hat{\Theta}$ vector.

**Conclusions on Experiment-1.** Although the sliding friction model for this experiment could be questionable, the interaction behavior obtained by the simulation could resemble the interaction between the robot finger and a rolling sphere (modeled in one dimension) - in this case made of steel. Aware of this, the Experiment-1 gives us the opportunity to understand the performance of the estimation algorithms in the present of a rolling object. It is clear that the estimations in (5-4) are not good. For the first estimate, the estimation seems to be correct, however this is not entirely true. If we look closer to the estimation, Figure (5-9), the $\hat{\theta}_1$ estimates start the convergence before the interaction. The reason is simple, the estimation uses always the velocity signal. But, if there is no interaction, this signal does not make sense for the error function.
\[ \epsilon = \dot{x}_r - \dot{\hat{\theta}}_1 \varphi^1. \]  

(5-21)

The first estimate, in the absence of interaction \( \varphi^2 = F = 0 \), will tend to the value 1. The second aspect of the parameter is their fluctuation around some values, i.e. for \( \hat{\theta}_2 \). The fluctuation is related to the error function \( \epsilon \). The function fluctuates around the zero value but does not asymptotically converges to it. The reason is a combination of the effect of the filter and the covariance matrix. Although, theoretically speaking, in the absence of the force signal the update of the second estimate should be zero, numerical approximation of the filter and the simulation can bring to have a small value of the order of \( 10^{-6} \) for \( \varphi_2 \). Assumed so, this value is multiplied or used in the update of the covariance matrix and successively in the gain expression \( K_k \). The gain, given a small error, will update the second estimate.

**Conclusions on Experiment-2.** This simulation had brought apparently good results. During the pushing action there is no lose of contact and there are no several bounced between the robot and the object (as in Experiment-1). A continuous interaction with a 'long' contact phase allows the estimates to converge to their nominal values. At the very end of the simulation, the estimates diverge away from a good estimation. This is because the object no longer 'moves'. The stiction simulated behavior of the Coulomb friction is acting at the very end. Because the movement of the finger does not require to push the object, object and finger remain in contact (because of the force sensor), but there is not a good estimation. Once again, at the very beginning, with no contact, the estimates cannot mathematically converge and their evolution is based on some errors as in Experiment-1.

**Conclusions on the Covariance Matrix \( \mathcal{P} \).** Another important aspect of these simulations is the evaluation of the covariance matrix. The covariance matrix, as explained in Section (4-4), plays an important role in the parameter estimation and it gives insights on the PE of the regressor vector. Using a continuous estimation for both RLS, the covariance matrix cannot physically represent what is really happening. In fact, the velocity signal makes the covariance matrix evolves in time even when there is no force (no contact). For this reason, the analysis of the covariance matrix is not coherent with the theory explained in Section (4-4) and Chapter (3). Nevertheless, for Experiment-2 the trace of the covariance matrixes was bounded during the contact interaction phase and it confirmed us to rely on the estimation of the \( \hat{\Theta} \) vector during that period of time.

**Trajectory Evaluation.** The trajectory selected was the one with the addition of second sinusoidal, \( S_2^2(t) \). One could doubt that a simple polynomial trajectory was sufficient enough to estimate parameter. This is partially correct. In fact, with just a polynomial trajectory some bounces could be probably avoided. However, if the Cruising Phase\(^2\) is prolonged for a long period of time, it could be showed that, the signals (force and velocity) resembled a damped step response and from Section (4-4) it is known that we could fall in one of the unwanted cases: Table (4-7).

With the addition of the first sinusoidal, \( S_1^1(t) \), some first attempts were made for these experiments. The results for Experiment-1 were worst than with the second sinusoidal. The

\(^2\)The phase between the run-up phase and the brake phase, where the velocity is constant.
reason was the amplitude of the sinusoidal. The amplitude is an important factor for the interaction: if it is taken too big many bounces can occur and, as proven for Experiment-1 they are far from a good interaction with a rigid movable object. The frequency of the sinusoidal was decided to be reasonably fast, but not too fast for the same reason explained for the amplitude.

**Parameter Validation.** The validation of the parameters was something not trivial at all. Imagine the case where the estimated parameters were part of an internal model. A validation of these parameter should be the following: a simulation with the same input to both real and estimated model and check the error between the two outputs. Several methods exist to validate the parameter; a popular one is $VAF$ [78]: it gives a quantity of the quality of the estimated model. With abuse of notation, consider a general plant output $y$ and the estimated one $\hat{y}$ (of the same plant).

$$VAF(y(k), \hat{y}(k, \Theta)) = \max \left\{ 0, \left( 1 - \frac{1}{N} \sum_{k=1}^{N} ||y(k) - \hat{y}(k, \Theta)||_2^2 \right) \right\} \times 100\%$$ \hspace{1cm} (5-22)

The $VAF$ has a value between 0% and 100%; the higher the $VAF$, the lower the prediction error and the better the model. However, in our case the parameters are estimated for an external model: a rigid movable object. The parameter estimates cannot be directly compared with the real object model because in practice, the object velocity as measurable signal is not available. The velocity of the object can be measured only when the robot touches it. Furthermore, with the assumption of no vision system, the object position could be lost after the experiment, i.e. Experiment-1, concluding in the impossibility to measure its signals: object position and velocity.

In simulation, a validation could be done using the real object model and compare its output with the estimated model. In this way, the same validation procedure could be similar to the one explained above. However, to be consistent with a practical situation, another approach was used. This approach is far from being theoretical. And it does not claim to be the best or a robust validation. For each experiment, the *good estimates* should be the ones that remains bounded (with an upper bound and a lower bound) for a certain amount of time during the tactile interaction. This practical validation of the parameters does not take into consideration the covariance matrix. However, the second stage of this validation is to look at the covariance matrix and decide to trust or not the *good estimates* - remember Table (4-7)!

A pseudo-code of the validation algorithm is represented in Algorithm (1). The bounds for a single estimate are defined by a $\delta$ displacement, from the estimate itself at instant $k$, $\dot{\theta}(k)$. At each time step $k$, the bounds are updated if the experiment situation is not suitable for a good estimate: there must be interaction $F(k) \geq 0$ and $T_x \leq T_p$. The second condition defines the certain amount of time for which the estimate should be bounded in order to be gathered. If the situation is matched, the bounds are not updated and the time $T_x$ is incremented. The estimate is decided to be a good one only when $T_x > T_p$. The time when the *good estimates* will be gathered is stored as LastTime. In our case, because of a deterministic scenario, it was decided to take the very last update as the *good estimates*. For a real implementation, the sample mean of the estimate could be gathered. The tuning parameters are $T_p$ and $\delta$. 

**Master of Science Thesis** Matteo Ciocca
Algorithm 1 Validation for a $\Theta$ vector of $n_y = 1$

1: procedure SINGLE PARAMETER VALIDATION
2: $\hat{\theta}_{\text{final}} = []; \text{LastTime} = 0; T_x = 0; \theta_{\text{down}} = \hat{\theta}(0) - \delta; \theta_{\text{up}} = \hat{\theta}(0) + \delta;$
3: for $k \in (0, t_f), t_f$ duration of the experiment do
4: if $T_x \leq T_p$ then
5: if $F(k) \geq 0$ then
6: if $\hat{\theta}(k) > \theta_{\text{down}}$ and $\hat{\theta}(k) < \theta_{\text{up}}$ then
7: if Deterministic then
8: $\hat{\theta}_{\text{final}} \leftarrow \hat{\theta}(k)$
9: else
10: $\hat{\theta}_{\text{final}} \leftarrow \left[\hat{\theta}_{\text{final}} + \hat{\theta}(k)\right]$ 
11: end if
12: unitary increment of $T_x$
13: else
14: $\theta_{\text{down}} \leftarrow \hat{\theta}(k) - \delta, \theta_{\text{up}} \leftarrow \hat{\theta}(k) + \delta$
15: reset: $\hat{\theta}_{\text{final}} = [], T_x \leftarrow 0$
16: end if
17: else
18: $\theta_{\text{down}} \leftarrow \hat{\theta}(k) - \delta, \theta_{\text{up}} \leftarrow \hat{\theta}(k) + \delta$
19: reset: $\hat{\theta}_{\text{final}} = [], T_x \leftarrow 0$
20: end if
21: else
22: if LastTime is 0 then
23: LastTime = $k$
24: end if
25: end if
26: end for
27: if $\hat{\theta}_{\text{final}}$ is not empty then
28: if Deterministic then
29: Gather $\hat{\theta}_{\text{final}}$
30: else
31: Gather $\hat{\theta}_{\text{final}} \leftarrow \left[\hat{\theta}_{\text{final}}\right] / T_p$
32: end if
33: else
34: No correct estimation.
35: end if
36: end procedure

In the following tables, the results are summarized. Both Experiment 1 and 2, as explained before, use the velocity signal continuously and the covariance matrix evolves based on that information. Parameters are listed in the Tables (5-3) and (5-9) according to Algorithm (1), with explicit $T_p$ and $\delta$. The LastTime is expressed close to each estimated label at the top of each Table.
### Table 5-3: Pushing Finger, Experiment-1: (steel) object model parameters and nominal $\Theta$ vector.

$T_p = 2 \, [s], \, \delta = 10^{-2}$.

<table>
<thead>
<tr>
<th></th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\hat{\theta}_1$</th>
<th>$\hat{\theta}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous $RLS_{\kappa}$</td>
<td>0.9994</td>
<td>0.0012</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>Discrete $RLS_{\kappa}$</td>
<td>0.9994</td>
<td>0.0012</td>
<td>~</td>
<td>~</td>
</tr>
</tbody>
</table>

$m : [Kg] \quad b : [Ns/m] \quad \hat{m} \quad b$

### Table 5-4: Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 10 \, [s], \, \delta = 10^{-3}$.

<table>
<thead>
<tr>
<th></th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\hat{\theta}_1(33.6s)$</th>
<th>$\hat{\theta}_2(33.5s)$</th>
<th>$\hat{\theta}_3(35s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete $RLS_{\kappa}$</td>
<td>0.99</td>
<td>0.025</td>
<td>-0.06</td>
<td>0.9898</td>
<td>0.0249</td>
<td>-0.0597</td>
</tr>
</tbody>
</table>

$m : [Kg] \quad b : [Ns/m] \quad F_c : [N] \quad \hat{m} \quad b \quad \hat{F_c}$

### Table 5-5: Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 15 \, [s], \, \delta = 10^{-3}$.

<table>
<thead>
<tr>
<th></th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\hat{\theta}_1(33.7s)$</th>
<th>$\hat{\theta}_2(37.6s)$</th>
<th>$\hat{\theta}_3(39.5s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous $RLS_{\kappa}$</td>
<td>0.99</td>
<td>0.025</td>
<td>-0.06</td>
<td>0.9898</td>
<td>0.0249</td>
<td>-0.0597</td>
</tr>
</tbody>
</table>

$m : [Kg] \quad b : [Ns/m] \quad F_c : [N] \quad \hat{m} \quad b \quad \hat{F_c}$
5-4 Discontinuities Problem

A real problem of a tactile interaction with a movable object is the loss of contact during the estimation. If the object used for the experiment is a rolling sphere, the polynomial trajectory could make the robot finger hit the sphere once and contact could be lost for the entire duration of the experiment. Assuming that more than a single bump with the object will be achieved by the robot finger, the force signal could present some discontinuities: the force profile would present only several spikes (as in Figure (5-6)). A force profile of this kind cannot help the parameter convergence to estimate external object properties.

From a general picture, looking at the previous results of Experiment 1 and 2, it is clear that less discontinuities on the force profile enrich the regressor vector and definitively help the convergence of the parameters.

Discontinuities as lack of PE. At first, these discontinuities were casted as a lack of PE. As explained in Section (4-4), the assumption of PE can be hard to obtain. Because this assumption is at the base of the convergence properties of RLS algorithms, Chapter (3), it was found interesting to understand the case when the excitation was incomplete, namely, the data did not yield sufficient information along all the directions in the parameter space. In fact, in our case, the information of the regressor vector should be said to "not be exciting" when there is no contact: $F = 0 [N]$. The concept of excitation subspace [81] was taken under account. The concept divides theoretically the parameter vector into two subspace: the excitation subspace, $\Omega_e$, and the subspace of decreasing excitation or underexcited subspace, $\Omega_u$: their components are orthogonal to each other $\Omega_e \perp \Omega_u$. The concept try to discriminate the parameter vector belonging to $\Omega_u$ with a modification to the discrete algorithm (5-20) must be made.

$$
K_k = \frac{P_{k-1}\Phi_k}{1 + r_k} \\
\hat{\Theta}_k = \hat{\Theta}_{k-1} + a_k \frac{\epsilon_k}{r_k} \\
P_k = \frac{1}{a_k} \left[ P_{k-1} - \frac{P_{k-1}\Phi_k\Phi_k^T P_{k-1}}{1 + r_k} \right]
$$

(5-23)

In [81] is proved that if there exists a scalar $c \geq 0$, a practical convergence of the parameter estimation even in poor excitation cases is obtained. The parameter $a_k$ is then chosen as:

$$
a_k = \frac{c + cr_k}{1 + cr_k}
$$

(5-24)

A similar algorithm in the case of lack of PE was found in the literature. The Directional forgetting algorithm is based on the same decomposition of the excitation subspace. The main drawback of RLS with forgetting algorithm is that all information is forgotten uniformly. As a consequence, the matrix $I$ may become singular over the course of time if the observations $\varphi_k$ do not excite all directions. The idea of directional forgetting is to split the information matrix in two parts and apply the forgetting factor to the new data according to:

$$
\tilde{I}_k = I_k \Pi_{\varphi_k}^T + \kappa \tilde{I}_k \Pi_{\varphi_k},
$$

(5-25)
The split is in accordance with: *excitation subspace* and *underexcited subspace*. And a new update of the covariance matrix:

\[ \overline{P}_k = P_k + \alpha_k \varphi_k \varphi_k^T , \]  
\[(5-26)\]

where the scalar \( \alpha_k \) given by:

\[ \alpha_k = \frac{1 - \kappa}{\kappa} \frac{1}{\varphi_k \mathcal{I}_k \varphi_k^T} \geq 0 . \]  
\[(5-27)\]

A significant drawback of this directional forgetting algorithm is that the matrix \( \mathcal{I}_k \) must be always available to compute (5-27).

**The Switching solution.** Both algorithms named above proved to do not solve the discontinuity problem. The reason is simple. There is not a lack of PE: there is *no excitation at all*. Being the \( F_k \) signal part of the regressor vector \( \varphi_k \), when the signal will be zero, part of the covariance matrix \( P_k \) will be updated incorrectly. This lack of signal information allowed to investigate on a property of the covariance matrix. Usually, the use of the RLS techniques deal with noise and with excitation subspace (as explained before). Our case is the exact opposite. Taking the \( RLS_{\kappa} \) in discrete time (5-20) and posing \( \Phi_k = [0; 0] \). The covariance update will be:

\[ P_k = \frac{1}{\kappa} P_{k-1} . \]  
\[(5-28)\]

Being \( \kappa \in (0, 1) \), this expression tell us clearly that the covariance matrix will increase in time by a \( 1/\kappa \) factor. If the start condition on the covariance matrix is \( P_0 = \alpha \mathcal{I} \) with a large \( \alpha \), it is not a surprise that some of its eigenvalues will go to infinity in time and \( P \) will be unbounded. Without further investigations a practical implementation to overcome this problem was the following:

*An if-else condition was applied to both RLS algorithms. Based on the absence of the force signal.*

The condition works as follow: if a force signal difference from zero is present, the velocity signal is eligible to be considered in the estimation process and the covariance matrix is allowed to evolve in time, because based on right informations. In the absence of force signal, the following updates are made:

\[ P_k = P_{k-1} \]
\[ \epsilon_k = 0 \]
\[ \hat{\Theta}_k = \hat{\Theta}_{k-1} , \]  
\[(5-29)\]

for the discrete \( RLS_{\kappa} \). For the continuous version of \( RLS \), in absence of the force signal, the conditions were:

\[ \frac{d}{dt} P(t) = \mathcal{O}_{ny \times ny} \]
\[ \epsilon(t) = 0 \]
\[ \frac{d}{dt} \hat{\Theta}(t) = \mathcal{O}_{ny} . \]  
\[(5-30)\]
It is worth to mention two approaches could have been taken to overcome the discontinuity problem. A well-known approach is the covariance resetting \cite{53}. This approach is needed in case of covariance wind-up. Relating this event to our problem, when one eigenvalue of $P$ tend to an high number (selected), the covariance matrix is reset to its initial condition $P(0)$. In our case, a reset condition could have been taken on the eigenvalue of the force signal. However, covariance resetting has an undesirable burst consequence on the parameters estimation.

$$\text{if} \; \lambda_{\text{max}}(P_k) > k, \; \text{then} \; P_k = P_0. \quad (5-31)$$

Another approach can be called variable forgetting factor \cite{52}. The effect of this modification has the same purposes of covariance resetting: this modification force the trace of the covariance matrix to stay around a certain upper-value, but does not allow burst consequences on the parameter estimation.

However, both algorithms were implemented and they did not bring any better results than the ones showed here. Reading carefully, these two techniques do not stop the parameter update $\hat{\Theta}$. So, for example, although the covariance matrix is reset, its informations will be used to update the parameter vector. But, if no force signal is present, the informations are wrong.

Same experiments were conducted as in Section (5-3).
Figure 5.10: Experiment 1, steel object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for Continuous $RLS_{\text{Cont}}$ and the discrete version $RLS_{\kappa}$ algorithms, both with a forgetting factor $\kappa = 0.99$. 

Master of Science Thesis  
Matteo Ciocca
Figure 5-11: Experiment-1, steel object. Evolution of the trace of the covariance matrix, $P$, for Continuous $RLS_{cont}$ and the discrete version $RLS_{al}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 100$. 
Figure 5.12: Experiment-2, wood object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for Continuous $RLS_{Cont}$ and the discrete version $RLS$ algorithms, both with a forgetting factor $\kappa = 0.99$. 
Figure 5-13: Experiment-2, wood object. Evolution of the trace of the covariance matrix, $P$, for Continuous $RLS_{Cont}$ and the discrete version $RLS_{\kappa}$ algorithms, initialized for both algorithms with: $P(0) = \alpha I$, $\alpha = 0.01$. 
Strategy Evaluation. The effect of the switching condition on the force signal is evident from Experiment-1. The estimates stop their adaptation when there is no contact as well as the trace of covariance matrix. This strategy was not beneficial for Experiment-1, Figure (5-10), and did not change so much the results of Experiment-2, Figure (5-12). For the Experiment-1 there is a better adaptation for the second estimate w.r.t. the discrete \( RLS_k \), on the other hand did not improve the adaptation for the continuous algorithm. The few, fast bounces are highlighted by the identifier error (\( e \)), Figure (5-10), that becomes different from zero for few little intervals of time.

For the analysis of the covariance matrix instead, this strategy allows to keep bounded the covariance matrix in case of no interaction. However, the theory on the parameters convergence is not eligible for the entire duration of the experiment but for small interval of time, i.e. in Experiment-1. This strategy ensures to avoid numerical errors due to the unboundedness of the covariance matrix, on the other hand, nothing can be said about the parameters convergence.

Some tuning has been made for different experiments: tuning on the forgetting factor and on the initialization of the covariance matrix. However, something fundamental cannot be changed:

> these algorithms selected work with a continuous estimation.

Based on this assumption, the switching strategy has the big limitation that no formal convergence proof can be said. For this reason, in the following section a different strategy will try to tackle this discontinuity problem from a difference prospective. The following table shows the result of Experiment-1-2. Meanwhile for Experiment-2 the Algorithm (1) brought good estimates, for Experiment-1 it was decided to take the last estimated value, regardless Algorithm (1).

**Table 5-6:** Pushing Finger (switching law), Experiment-1: (steel) object model parameters and nominal \( \Theta \) vector

<table>
<thead>
<tr>
<th></th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \hat{\theta}_1(\text{final}) )</th>
<th>( \hat{\theta}_2(\text{final}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous ( RLS_k )</td>
<td>0.9994</td>
<td>0.0012</td>
<td>0.5140</td>
<td>0.0259</td>
</tr>
<tr>
<td>Discrete ( RLS_k )</td>
<td>0.9994</td>
<td>0.0012</td>
<td>0.9878</td>
<td>( 7.0243 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( m : [Kg] )</td>
<td>( 7.8 )</td>
<td>0.4</td>
<td>( 0.3866 )</td>
<td>( 18.7895 )</td>
</tr>
<tr>
<td>( b : [Ns/m] )</td>
<td>( 7.8 )</td>
<td>0.4</td>
<td>( 14.2362 )</td>
<td>( 17.3366 )</td>
</tr>
</tbody>
</table>
5-5 Time-varying parameters

Given the RLS structure decided in Chapter (4), the parameter estimation with the switching law was, to my knowledge, the best way to deal with discontinuities - explained in Section (5-4). The reason was the one related with the signals. If the signals contain wrong informations, there is no reason to use them for the estimation strategy.

In sight of this, another approach was considered. Because of the unknown instant time of these discontinuities, the new approach is to do not consider the discontinuities at all. The discontinuities are the reason that all the estimation algorithms can fail: because these algorithms work with continuous signals. An approach to avoid that is to consider the interacting systems as in Figure (5-14).

![Figure 5-14: Different modeling prospective for a tactile interaction, in the case of a rigid movable object.](image)

The physical interaction is modeled as it should be in reality: when the robot finger and the rigid movable object are in contact, the two masses are coupled together. With this new model/s, the estimation algorithm must be revisited.

5-5-1 A different approach

Experiment-1 is taken under analysis. The $RLS_\kappa$ uses the following elements:

$$\varphi^T(t) = \begin{bmatrix} w_1(t) & w_2(t) \end{bmatrix}, \quad \hat{\Theta}^T = \lambda \frac{\hat{b}_i}{\mu} \frac{1}{\hat{m} \mu},$$

(5-32)

where the regressor vector filters the following signals:

$$W_1(s) = \frac{\mu}{\frac{s \hat{\lambda}}{\hat{\lambda}} + 1} F(s), \quad W_2(s) = \frac{\mu}{\frac{s \hat{\lambda}}{\hat{\lambda}} + 1} Y(s),$$

(5-33)

$F(s)$ and $Y(s)$ representing respectively the force and velocity signals. With this structure the force signal is one of the component of the regressor vector. Before, a switching condition was used to stop the estimation. In fact, without force signal, the velocity (of the robot) does not represent the velocity of the object.

Matteo Ciocca

Master of Science Thesis
However, with the limitations explained before, the new regressor vector and the new theta vector are chosen as the followings:

$$\varphi^T(t) = \begin{bmatrix} w_1(t) & w_2(t) \end{bmatrix}, \quad \hat{\Theta}^T = \begin{bmatrix} \lambda - \frac{\hat{B}}{\hat{M}} & 1 \mu \\ \mu \hat{M} & \mu \end{bmatrix}.$$  

where the new filtered signals are:

$$W_1(s) = \frac{\mu}{s + 1} U(s), \quad W_2(s) = \frac{\mu}{s + 1} Y(s), \quad Y(s) = sX_r(s).$$

There are two important characteristics that has been changed: the parameter to estimate and the signals. With the new regressor the signal chosen is no more the force signal but the actuation signal for the robot finger ($u$). The parameters represent both systems in Figure (5-14) as follows. When the robot finger is not in contact:

$$\hat{B} = 0, \quad \hat{M} = \hat{m}_{robot},$$

and when the robot is in contact,

$$\hat{B} = \hat{b}_v, \quad \hat{M} = \hat{m}_{robot} + \hat{m}.$$  

The $\Theta$ vector should converge for the two different situation as:

$$\hat{\Theta}_{\text{no-touch}}^T = \begin{bmatrix} \lambda - \frac{\hat{b}_v}{\hat{m}_{robot} + \hat{m}} & 1 \\ \mu \hat{m}_{robot} + \hat{m} & \mu \end{bmatrix}, \quad \hat{\Theta}_{\text{touch}}^T = \begin{bmatrix} \lambda - \frac{\hat{b}_v}{\hat{m}_{robot} + \hat{m}} & 1 \\ \mu \hat{m}_{robot} + \hat{m} & \mu \end{bmatrix}.$$  

Mathematically speaking, the estimation algorithm became a $RLS_\kappa$ with time-varying parameters, $\Theta(t)$.

$$\hat{\Theta}_k = \arg \min_{\hat{\Theta}} \left\{ \frac{1}{k} (Y_k - \Phi_k \Theta_k)^T \hat{X} (Y_k - \Phi_k \Theta_k) \right\}.$$  

And the error function becomes:

$$\epsilon(t) = \varphi^T(t) \Theta - \varphi^T(t) \hat{\Theta}(t) \Rightarrow \epsilon(t) = \varphi^T(t) \Theta(t) - \varphi^T(t) \hat{\Theta}(t).$$

The most important thing about this transformation is the fact that the model does not deal with discontinuities of signals in the regressor. In fact, the actuator signal will be always be present during both Experiments. On top of that, the actuation signal can be driven to be exciting enough following a particular trajectory.

The velocity signal is coherent, all the time instant, with the parameter estimation. Does not matter if the robot is in contact or not, because this hybrid description is taken into account by the time-varying parameters.

**Insight 5-5.1 (The variation of the forgetting factor).** For time-varying parameters the tuning of the forgetting factor $\kappa$ was essential. In fact, $\kappa$ influences the reactivity of the $RLS_\kappa$ in the presence of variation of parameters. If $\kappa$ is decreased, the parameter estimation tracking is faster. On the other hand, for constant parameter some fluctuations can occur. Because of the assumption of time-varying parameters, the $\kappa$ should be decreased. △
For the second Experiment-2, with the presence of the Coulomb friction, the same procedure is applied.

\[
\hat{\Theta}_{\text{no-touch}}^T = \begin{bmatrix}
\frac{\lambda}{\mu} & 1 & \frac{1}{\mu} & 0
\end{bmatrix}
\]
\[
\hat{\Theta}_{\text{touch}}^T = \begin{bmatrix}
\lambda - \frac{\hat{b}_v}{(m_{\text{robot}} + \hat{m})\mu} & 1 & \frac{\hat{F}_c}{(m_{\text{robot}} + \hat{m})\mu}
\end{bmatrix},
\]

and for the regressor,

\[
\varphi^T(s) = \begin{bmatrix} W_1(s) & W_2(s) & W_3(s) \end{bmatrix}
\]
\[
W_1(s) = \frac{\lambda}{s} Y(s), \quad W_2(s) = \frac{\lambda}{s} U(s), \quad W_3(s) = \frac{\lambda}{s} \text{sign}(Y(s)), \quad Y(s) = sX_r(s)
\]

\[
U(s), \text{ the actuator signal instead of the force signal. The Experiment-1 and 2 were simulated for the same respectively objects with this new interpretation. The parameters for the filter and the controller were not changed as well as the use of the same trajectory.}

<table>
<thead>
<tr>
<th>Table 5-7: Filter and Controller Parameters</th>
</tr>
</thead>
</table>
| \hline
| \( \lambda \) & \( \mu \) & \( m_d \) & \( k_p \) & \( k_d \) |
| Experiment-1 | 100 & 100 & 1 & 5 & 3 |
| Experiment-2 | 100 & 100 & 1 & 80 & 10 |
Figure 5-15: Experiment-1, steel object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ - discrete version RLS algorithms, forgetting factor $\kappa = 0.75$. 

Master of Science Thesis Matteo Ciocca
Figure 5-16: Experiment-1, steel object. Above, the signal filtered: \( F = u \) and velocity of the robot \( \dot{r}_r \). Below, the evolution of the trace of the covariance matrix, \( P \) - discrete version \( RLS_\infty \) algorithm, initialized with: \( P(0) = \alpha I, \alpha = 100 \).
Figure 5-17: Experiment 2, wood object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$ - discrete version RLS algorithm, forgetting factor $\kappa = 0.99$. 
Figure 5-18: Experiment-2, wood object. Evolution of the trace of the covariance matrix, $\mathcal{P}$ - discrete version $RLS_\infty$ algorithm, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. 
Figure 5-19: Experiment 1, steel object, changed robot mass \( m_{\text{robot}} = 1 \text{ [Kg]} \). Estimation of the \( \hat{\Theta} \) vector and the error function \( \epsilon \) - discrete version RLS algorithm, forgetting factor \( \kappa = 0.75 \).
Figure 5-20: Experiment-1, steel object, changed robot mass $m_{\text{robot}} = 1\ [Kg]$. Evolution of the trace of the covariance matrix, $\mathcal{P}$ - discrete version $RLS_{\kappa}$ algorithm, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. 
Simulation Evaluation and Results. The new formulation of the $RLS_{w}$ improved the estimation for Experiment-1. Comparing Figure (5-4), (5-10) and (5-15), the second estimate proved his convergence to the right estimate, in this case the mass. The first parameter struggles to converge to the nominal estimate in such a short time: the short time is dictated by the "bump" between the object and the robot. Not surprisingly, a third simulation, where the mass of the robot finger was changed to $1\ [Kg]$ was conducted to prove that a longer time of interaction is needed to understand the dynamics of the object friction, in case of Experiment-1 a viscous damper.

Regarding Experiment-2 the results did not change so much. Experiment-2 is a confirmation that with a long interaction between the robot and the object, where the robot pushes the object forward, a good estimation can be accomplished regardless the algorithm used: see Figure (5-17), (5-12) and (5-7). Interesting, for all the Experiment-2 is the behavior at the end of the simulation. The robot finger decreases its control action and it cannot break the Coulomb friction, hence, it is not able to move the object. The chattering of the velocity of the object, $t \in (45s, 50s)$, is visible and it has an impact on the parameter estimation. The object and the robot finger are still in contact thanks to the deformable dynamics of the force sensor\[3\], but the parameters "do not understand the behavior" and they diverge from the right estimate. This event is perceived by the trace of the covariance matrix that goes unstable at the very end.

A very crucial characteristic must be pointed out from the explicit form of the error function. Take the error function of Experiment-2.

$$\epsilon = \dot{x}_r - \hat{\theta}_1\varphi^1 - \hat{\theta}_2\varphi^2 - \hat{\theta}_3\varphi^3$$, \hspace{1cm} (5-43)

During the event of no-touch, the friction force should be zero as well as the damping coefficient. Being $\lambda = \mu$, the (5-43) becomes:

$$\epsilon = \dot{x}_r - \varphi^1 - \hat{\theta}_2\varphi^2$$, \hspace{1cm} (5-44)

From Equation (5-42), in the error function (5-44), the only two signals remained are the filtered velocity and input signal. At this point, it should be evident that, in order to minimize the error function ($\epsilon$), $\hat{\theta}_2 = 0$ because, asymptotically, the filter should track the velocity signal. Given a constant signal the previous statement should be correct. However, even with a step input on the actuator, the nature of a double integrator for the robot finger make impossible the perfect tracking/filtering of the velocity signal. Between the velocity and the filtered velocity will be always a small error that allows the parameter $\hat{\theta}_2$ to evolve to the right estimate. Furthermore, the parameter $\hat{\theta}_2$, representing the inverse of mass of the robot finger ($m_{robot}$) multiplied by the gain of the filter ($\mu$), will be always a very small parameter.

The reason is inherent not only on the choice of the robot mass, but also on the choice of $\mu$, in our case 100.

On the covariance matrix. Proved (by an implementation point of view) the reason of the parameter convergence during the non-contact phase, it is important to point out the Lyapunov stability explained in Chapter (3) is eligible for our case. The continuous estimation and parameter adaptation was the fundamental hypothesis of these estimation techniques. The time-variation of the parameter does not compromise the Lyapunov theory. In fact,

\[3\]Used still in simulation, but not for the estimation algorithm.
the stability theory can be applied correctly to intervals of time where the nominal/real parameters do not change. Nevertheless, if the variation is extremely fast and periodic, theoretically speaking, a good tuning and/or sampling time could still converge online to the new set of parameters.

The PE problem can finally be investigated thanks to the trace of the covariance matrix - Section (4-4). Thus, looking at Figure (5-17)-(5-18)-(5-20), the trace of the Covariance matrix is bounded and the estimation can be said to be reliable.

time-varying Parameter Evaluation. The nature of the new strategy does not compromise Algorithm (1). In fact, the parameter are stored only when a force signal is present. However, the two parameters present a different precision. For the first parameter, a small variation of $10^{-5}$ [1/s] could determine a contact condition. This small variation is due to the decision of both masses: object and robot as well as the viscous parameter and the filter parameter. On the other hand, the second parameter, that regards only masses and filter gain, can change of "bigger" variation $10^{-3}$ [1/kg]. On the basis of this, the different scale on the variation was represented by a different $\delta$ for each parameter.

Table 5-8: Pushing Finger ($m_{robot} = 1$ [Kg]), Experiment-1: (steel) object model parameters and nominal $\Theta$ vector. $T_p = 1$ [s], $\delta \hat{\theta}_i = 10^{-5}$ and $\delta \hat{\theta}_2 = 10^{-3}$.

<table>
<thead>
<tr>
<th>From $\Theta_{touch}$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\hat{\theta}_1(28s)$</th>
<th>$\hat{\theta}_2(13s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete RLS$_m$</td>
<td>0.9995</td>
<td>0.0011</td>
<td>0.9996</td>
<td>0.0011</td>
</tr>
<tr>
<td>$m_{robot} + m$ : [Kg] $b_v$ : [Ns/m] $M$ $B$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete RLS$_m$</td>
<td>8.8</td>
<td>0.4</td>
<td>9.4309</td>
<td>0.4116</td>
</tr>
</tbody>
</table>

Table 5-9: Pushing Finger, Experiment-2: (wood) object model parameters and nominal $\Theta$ vector. $T_p = 15$ [s], $\delta = 10^{-3}$.

<table>
<thead>
<tr>
<th>From $\Theta_{touch}$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\hat{\theta}_1(30.9s)$</th>
<th>$\hat{\theta}_2(27.3s)$</th>
<th>$\hat{\theta}_3(27.3s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous RLS$_m$</td>
<td>0.9983</td>
<td>0.0041</td>
<td>-0.01</td>
<td>0.9983</td>
<td>0.0042</td>
<td>-0.0100</td>
</tr>
<tr>
<td>$m$ : [Kg] $b$ : [Ns/m] $F_c$ : [N] $\tilde{m}$ $\tilde{b}$ $\tilde{F}_c$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous RLS$_m$</td>
<td>2.4</td>
<td>0.4</td>
<td>2.4</td>
<td>2.380</td>
<td>0.408</td>
<td>2.38</td>
</tr>
</tbody>
</table>
This Chapter represents the first attempt for an online continuous model-based estimation of mass and friction properties for a rigid movable object.

The simulation environment created a good continuous interaction architecture avoiding discontinuities. Two types of experiments were conducted. The first, with a questionable friction model revealed to be the most important one. The discontinuities during the interaction where the key to understand the used algorithms in depth. In fact, the estimation problems arise from the first experiment pushed the investigation on specific estimation techniques and a deeper analysis on the results. From Section (5-3), a direct use of $RLS_\kappa$ revealed to be effective for the Experiment-2; and the performance did not change with the addition of a switching law and in Section (5-5).

On the other hand, the Experiment-1 was the reason to pursue a deeper investigation. As concluded, a continuos parameter estimation did not bring any good result. The signals used for the online estimation were meaningless without interaction. This fact, brought to explore new solutions. With the switching mechanism, a proper selection of signals was finally obtained. However, the "switching estimation" did not any convergence improvements. But, the accurate analysis of the covariance matrix and the knowledge gained from Chapter (3), proved that the switching mechanism had limitations: from a theoretical point of view, and in practice too. The decision to create a new global output function that could include both situations of touching and not touching was decisive. In fact, in Section (5-5), an improvement on the online adaptation of the parameter for the Experiment-1 finally was achieved. At this point, the decision to change for an online time-varying parameter optimization problem solved the lack of assumption on the continuous estimation. The Lyapunov theorem proved in Chapter (3) could be finally taken into consideration and the covariance matrix could be finally studied.

Regarding the approach with time-varying parameters, it lacks of the informations from the force sensor. As future work, it could be interesting to expand the architecture with multiple cost functions:

$$
\hat{\Theta}_k = \arg \min_{\Theta} \left\{ \frac{1}{k} \alpha(k) (Y_k - \Phi_k \Theta_{no\text{-touch}})^T \mathcal{X} (Y_k - \Phi_k \Theta_{no\text{-touch}}) + \frac{1}{k} [1 - \alpha(k)] (Y_k - \Phi_k \Theta_{touch})^T \mathcal{X} (Y_k - \Phi_k \Theta_{touch}) \right\} .
$$

(5-45)

Where a continuous weighting parameter, $\alpha(k)$, could be designed to focus the online estimation for a contact or a non-contact situation only. An example could be:

$$
\alpha(k) = \frac{1}{2} \left\{ \tanh(K[F(k) - \varpi]) + 1 \right\} .
$$

(5-46)

The term $\varpi$ is a parameter to be tuned: it represents the threshold on the real force signal that should state when the robotic platform is in contact or not. And for a higher switching adaptation gain, $K > 1$, a faster switching condition is obtained: $K$ should be chosen very high.
Experimental Setup and Results

This Chapter concludes the thesis with a test of the algorithms on a real set-up. The robotic platform used is a "KUKA LBR iiwa 7 R800". The platform tries to achieve what was performed in the previous simulated experiments: push an object and estimate its properties such its mass and its friction. The non-deterministic nature of the experiment is given by the noise on the signals. Because the thesis focused only on deterministic scenario, the observations and insight on the results are limited to the knowledge acquired until now. However, important aspects of this first attempt were considered to be reported.

6-1 Introduction

The Chapter was not intended to study a new robotics platform or a particular control law. The experience aimed to use the end-effector of a robotics platform and try to re-create the scenario designed in simulation. The laboratory experience was far to be complete: only few tests were conducted and not all the characteristic were taken under analysis because lack of time. On top of that, few behaviors were not accurately analyzed.

However, it was considered important to show the practical aspects of the algorithms implemented in Chapter (5). Because their characteristics could be rightly doubted to have a good practical effects. The Chapter presents few properties on the robotic platform as well as few observations on the gathered signals. The goal is to re-create the experiment in Chapter (5) and use the gathered signals to infer on the properties of an external object, in this case a rigid movable mass.
6-2 Relevant informations on the platform and object

The robotics platform used for the real-time experiment is a KUKA LBR iiwa 7 R800. The LBR iiwa is classified as a lightweight robot and is a jointed-arm robot with 7 axes. The actuation to track a trajectory consists of an impedance control law with a gravitational compensation factor. To my knowledge, it was not possible to access directly to the actuated torques as well as the control mechanism. However, there was the possibility to shape the stiffness and the damping ratio of the control law.

The platform was equipped with joint torque sensors. The sensors can calculate reaction torques, $\tau_r$. Meanwhile the user has not access to the active torques, $\tau$, the robotics platform uses the active torques to calculate the total torque $\tau_{tot}$. If the robot collides, the different between the applied torques and the total torques generate reaction torques.

$$ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_r = \tau_{tot} $$

$$ \tau_r = J_r^T(q)F_r $$

In our case, the $J_r(q)$ Geometric Jacobian was the one to derive the forces applied to the end-effector. The robotic platform allowed to gathered the force signal in a particular direction at the end effector. The direction was the one towards the object.

The platform was not equipped with velocity sensors, so the velocity measurement was derived by differentiated the position signal.

The object was a hexagon polygon made of lead. The dimension of the object are: the height 4 [cm], the distance from vertex to vertex 17 [cm] and the distance from side to side 15 [cm]. The object weighs 5 [Kg].

![Robot and object](image1.png)

**Figure 6-1**: Robot and object (left). Model of the real-time experiment (centre). Object (right).
6-3 Understanding the signals

Before trying to re-create the experiment in Chapter (5), two important steps were taken. The first decision was to run the platform without any movement, without any contact. The reason was to understand the presence of some offset in the measured signals or some strange behavior in general. The Figure (6-2) shows the position of the end-effector of a single direction and the force signal in the same direction applied to the end-effector. The irrelevant offset on the initial position can be seen. On the other hand, the force signal does not converge to zero at the final time instant: 15 [s]. The force signal seems to converge around $\sim -1.58$ [N]. The error could be possibly related to some mechanical uncertainties not taken into account for the gravity compensation. Given the heavy robotics platform, the error around a Newton can be considered a very small error. However, for the purposes of our experiment, the error could be high enough to do not trust the force signal.

Another experiment was to make the robot follow a trajectory: a polynomial trajectory and the same trajectory with the addition of a sinusoidal. The experiment with the single polynomial trajectory was conducted with and without contact to understand the difference in the force signal. From Figure (6-3), the force signal drops around $\sim -9$ [N] because the robotics platform is trying to push an object. On the other hand, when the robot tries to follow the polynomial trajectory in the free motion, the force does not remain constant. When the robot starts the movement the force signal drops around $\sim -3$ [N] and during the Cruising Phase and the Brake Phase it converges back to $\sim -1.58$ [N].

The force behavior, was tried to be investigated with a java function to get the force inaccuracy: getForceInaccuracy(). In unfavorable positions, the calculated force/torque values can deviate from the actual forces and torques applied. In particular near singularities, several of the calculated values are highly unreliable and can be invalid. Depending on the axis position, this only applies to some of the calculated values. The value of this function gives back a quantity of inaccuracy. However, due to the lack of time, it was impossible to understand if this inaccuracy could be related somehow to the force signal gathered in free motion. Hence, the inaccuracy was not taken into account and the force signal gathered was used for the estimation without modification. Last, in Figure (6-4), the robotic platform tries to follow a polynomial trajectory with the addition of a sinusoidal.

<table>
<thead>
<tr>
<th>Table 6-1: Selected trajectory Parameters: polynomial+sinusoidal, and Controller parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta S_d$ [dm]</td>
</tr>
<tr>
<td>Free Motion</td>
</tr>
</tbody>
</table>

In Figure (6-4), the trajectory is not perfectly followed. However, the scope of the experiment will not to track a certain movement but to push an object and excite its dynamics. Same conclusions for the velocity signal, when the robot will be in contact with the object, the velocity of the object and the robotic platform will be the same so the velocity signal does not have to track any reference, but it must be used in the estimation strategy.

$^1$All the experiment consider the position on the $y$-axes of the end-effector frame, $\Psi_e$. 

Master of Science Thesis Matteo Ciocca
Figure 6-2: Robot in a stand-still position to understand some possible offset of the signals.
Figure 6-3: Robot following a polynomial trajectory. The figure shows the position in free motion, \( \cdot \)\text{\textsuperscript{\text{f}}} \), and the position pushing the object \( \cdot \)\text{\textsuperscript{\text{p}}} \). Below, the force profiles and the force inaccuracy.
Figure 6-4: Free motion. Robot following a polynomial trajectory with the addition of a sinusoidal: $A = 10\ [mm], \ w = 4.5\ [rad/s]$. 
6-4  Experiment Results

In this Section, the gathered data is used in order to estimate the parameter of the rigid movable object. While the mass of the object is known: $5 \, [Kg]$, the friction force it is hard to define. From some empirical tables online, the kinetic friction coefficient $\mu_d$ could be assumed to be 0.2. The object (made of lead) slides on a waxed wood. The viscous coefficient is (again) assumed from [49]. However, these nominal parameters were selected with a reasonable manner, without being extracted from another experiment.

The end-effector of the manipulator was posed in touch with the external object. The initialization allowed to use all the data gathered from the experiment. At the beginning of the movement and at the very end there are some discontinuities, but they do not take long period of time. So, practically, the entire force profile is eligible for the estimation algorithm. The sampling time for the experiment was $h = 0.005 \, [s]$.

Table 6-2: Filter and Impedance Controller Parameters.

<table>
<thead>
<tr>
<th>Experiment-(a)</th>
<th>$\lambda$</th>
<th>$\mu$</th>
<th>$k_p$ [N/m]</th>
<th>$D_{ratio}$</th>
<th>$A$ [mm]</th>
<th>$w$</th>
<th>$t_f$ [s]</th>
<th>$\Delta S_d$ [dm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment-(b)</td>
<td>20</td>
<td>20</td>
<td>500</td>
<td>1</td>
<td>5</td>
<td>4.5</td>
<td>20</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 6-3: Selected Empirical Parameters

<table>
<thead>
<tr>
<th>Object Properties</th>
<th>$b_v$ [Ns/dm]</th>
<th>$\mu_d$</th>
<th>$m$ [Kg]</th>
<th>$F_c$ [N]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.04</td>
<td>0.2 5</td>
<td>9.8</td>
<td></td>
</tr>
</tbody>
</table>

Table 6-4: Nominal (Empirical) Estimates, $\Theta$

<table>
<thead>
<tr>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96</td>
<td>0.01</td>
<td>-0.098</td>
</tr>
</tbody>
</table>
Figure 6-5: Experiment-(a). Above, the matrix entries of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version $RLS_\kappa$ algorithms, initialized with: $P(0) = \alpha I$, $\alpha = 0.01$. 
Figure 6.6: Estimation of the $\hat{\theta}$ vector and the error function $\epsilon$, discrete version RLS algorithms, with a forgetting factor $\kappa = 0.99$. 

Master of Science Thesis Matteo Ciocca
Figure 6-7. Experiment (a). Force and velocity signal: real and filtered. Below the three components of the regressor vector.
Figure 6-8: Experiment-(b). Above, the matrix entries of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version $RLS_{\kappa}$ algorithms, initialized with: $\mathcal{P}(0) = \alpha I$, $\alpha = 0.01$. 
Experimental Setup and Results

Figure 6-9: Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, discrete version RLS algorithms, with a forgetting factor $\lambda = 0.99$.

Matteo Ciocca
Master of Science Thesis
Figure 6.4: Experiment Results

Figure 6.10: Experiment (b). Force and velocity signal: real and filtered. Below the three components of the regressor vector.
On the real-time Experiment. The experiment tries to re-create the simulated situation in Chapter (5). It is not a surprise that the situation is hard to achieve. The disadvantage characteristic of this scenario is that the object can slide in several directions, not only in one. This real situation cannot coincide with the selected model. However, thanks to the particular contact between the object and the platform, Figure (6-11), several simulations were conducted where the object slides correctly along with the robot movement. The experiment showed in this Section are the ones where the sliding was considered correct. In fact, if the robot applies a torque to the object and it slides away, the model for a rigid movable object is not eligible for that situation. The goal was to achieve the situation in Chapter (5) because of the simplicity model designed.

![Image](image_url)

**Figure 6-11:** Experiment (a)-(b). Closer look at the contact point between the robot and the object (left). Distance between the robot end-effector and the table (right).

On the other hand, a good result was the absence of discontinuities during the interaction. When the robot touches the object and starts the pushing action, the robot and the object remained coupled. This is confirmed by the force profile for each experiment. This effect was somehow predicted from the simulated experience in Chapter (5) with the friction model containing Coulomb friction.

In addition, both experiments start with the contact condition. Although some first movements create discontinuities for the contact condition, in general, for the entire duration of the experiment it was assumed that the robotic platform remained in contact with the object. With this assumption, the entire force signal was used for the estimation.

On the Covariance Matrix $\mathcal{P}$. Between time instant 0 [s] and 2 [s], the trace of the covariance matrix goes unbounded. This case probably does not related to a poor excitation but to the noise\(^2\). However, poor excitation could be reasonable due to some bounces at the beginning of the experiment. There may be the same behavior simulated in Section (5-3), when the $RLS_\kappa$ was implemented without any switching law. After 2 seconds the covariance matrix remains bounded. In a deterministic scenario, the estimation can be trusted. Here, the case can be questionable due to the noise factor.

\(^2\)Not investigated in this Thesis.

Matteo Ciocca

Master of Science Thesis
**Parameter validation.** The parameter validation part was hard to achieve. With respect to Algorithm (1), the $\delta$ value was not a trivial choice as well as the $T_p$. Moreover, kinetic friction coefficient was assumed $\sim 0.2$ and an empirical assumption was made on $b_v$. From a practical point of view, it was very interesting to understand the difficulty of a correct estimate. All the estimated in a deterministic scenario are precise values. $\hat{\theta}_1$ for example has its nominal value around 1. But, as showed in Chapter (5), its value could be very close to one, i.e $\sim 0.9995$, so in a real scenario it is hard to tell if this parameter is converging to a right value or it is converging to 1. On the other hand, $\hat{\theta}_2$ can be trusted. In fact, this estimate is based only in the filter gain and the mass of the object. The values of this estimate are again very precise, and very small for bigger mass (or increasing the filter gain). However, from Figure (6-6)-(6-9), it can be seen that $\hat{\theta}_2$ is stabilizing above the nominal value. It means that the mass of the object is estimated to be smaller than its real value. On the other hand, it is interesting that the third estimate, $\hat{\theta}_3$, is stabilizing to a negative value proving to estimate correctly the sign of the Coulomb force.
6-5 Conclusions

This experience in the DCSC laboratory would not have been conducted without the help of (dr.ing.) Jens Kober that implemented several trajectories the robot should perform as well as helping me to gather the signals. In total, only 1h and 15min were spent in the laboratory. This information is not intended to justify the results obtained. But to make the reader aware that several investigations could have been conducted in a better scientific manner. In fact, most of the experiment were conducted trusting the simulated behavior in Chapter (5).

Starting from the understanding of the platform and signals. The force signal could have been studied from a free motion point of view and the 'force inaccuracy' could have been related to the force signal itself to bring better results. The setup could have been decided in a proper manner understanding friction characteristic of the surface where the object would have slide to validate the parameter estimated. Furthermore, several objects could be tested. Object with known mass. Different object shapes could achieve a better sliding motion in one direction. Another robot configuration to push the object could have been tested. Last, but not least, (at the beginning of the experiments) the surface of the table was not at the same height at every (surface) point. So, a provisory solution was taken in order to obtain the same height but the solution was not validated\[^3\]. The same height (in each point of the surface) is very important because it confirms to apply a force perpendicular to the surface of the object, which fortunately, in this case, has a flat surface.

Given the time for the laboratory experience and several unsolved problems explained before, no validation was conducted on the estimates obtained. In addition, no exact validation procedure, to my knowledge, was found very effective. An idea could be to apply a filter to the estimate themselves. It could probably reduce the fast variation around the values where the estimate converge.

---

\[^3\]Two legs of the table were raised, but no spirit level was used to validate the new height.
Conclusions

7-1 Thesis Results

An extended literature survey (2) posed the basis of the research question stated in the Introduction (1). The goal of the literature survey was twofold. The survey focused on the classification of environment dynamical models found in the literature. The survey focused on those models describing local properties of external object such as stiffness, mass and friction. Only two models were found, both to describe compliant surfaces. The first model is composed of a spring and a damper in parallel. The description was inspired by the Kelvin-Voigt (linear) and the Hunt-Crossley (nonlinear) models. On the other hand, a single spring was widely useful to describe the same compliant surfaces. A single spring is effective in practical situations because it reduces the environment description to one parameter only: the level of difficulty of parameter identification is directly proportional to the number of parameters. The second scope of the literature survey was the understanding of which model-based techniques were used to adapt and identify parameters of an external environment. Adaptive observer as well as Parametric system identification techniques were found to be the most popular ones. The classification of these techniques revealed the motivation of their use. In general, model-based adaptive techniques aim to identify a set of parameters. But the adaptation can work side-to-side with control to accomplish some goals stated in Section (1-1). A general understanding of these techniques and models was essential in order to identify a good research question and take the first steps in a precise direction.

The selected identification technique belongs to the family of Parametric system identification algorithms. The selection was not made blindly. First, a new derivation of two algorithms, RLS and RLS_κ, in a matrix form was proposed. Although, these two algorithms are widely used and demonstrated in the literature, a matrix formulation was not found and it gave the possibility to comprehend that RLS_κ could be re-cast as a weighted least-squares algorithm. On top of both derivations, in Section (3-1) few elements were listed. These elements were found in common in scientific papers among model-based parameter adaptation techniques. Second, the Lyapunov stability of these techniques closed the picture on the stability of general model-based adaptive techniques. Take for example Adaptive Control: (i) it is based on
Lyapunov stability and (ii) it is based on Parametric system identification techniques. Because Adaptive Control was assumed to be a subset of Parametric system identification, a Lyapunov stability theorem was expected to be found. In conclusion, the study of Parametric system identification is at the core of model-based adaptation strategies and its understanding could pose fundamental bases for future experiments and improvements with complex adaptation laws.

The model of a rigid movable object was aimed to fit with the identification strategy selected. Velocity and force signal were assumed available for future experiments. With this assumption, and with the elements listed in Section (3-1), two simple models of rigid movable objects were derived. Although the simplicity of these two models, with a not novel dynamical friction description, the interpretations of these models, where the input force is the force from an interaction, changed the perspective of their use. The dynamical friction description could be expanded to a more complex model. However, it would have led to a different research question, because the focus is on simple models.

Two algorithms, belonging to the selected identification framework, were tested on the models. Furthermore, few well-known insights were given on the forgetting factor and on the covariance matrix. Finally, the problem of the persistent excitation was studied. In [79], a fundamental statement, on the relationship between the covariance matrix and the persistent excitation assumption of the regressor signal, gave useful information for future practical experiments.

Chapter (5) focused entirely on a novel simulated experiment aimed at answering the research question asked. A robot uses a finger to push a rigid movable object. Thanks to this interaction with it, the robot tries to estimate the object’s mass and the friction between the surface and the friction between the object and the (unknown) surface where it sits. The pushing finger experiment was tested under different algorithms. First, a straightforward implementation of $\text{RLS}_\kappa$ was found to be not effective. Furthermore, it was found that to be impossible to verify the correctness estimation from a theoretical perspective, Section (3-1-2). Afterwards, a switching law was tested to improve the performance of the experiment. Despite a better stability of parameter, it was concluded that both strategies could not solve our parameter estimation. A good understanding of the covariance matrix in Section (4-4) helped the explanation of the results achieved. (a) The switching law did not respect a basic assumption of $\text{RLS}_\kappa$: they work with continuous signals. (b) the covariance matrix for the straightforward implementation of $\text{RLS}_\kappa$ proved to fall in one of the two cases of Table (4-7), for which the experiment should be re-conducted. But, on top of that when there was no interaction, the velocity signal could be interpreted as the velocity of the object. To overcome these two difficulties regarding the discontinuities of the signals and the problem of the covariance matrix, a new time-varying model was used instead. This new strategy proved to be effective not only from a simulated point of view, but it could also finally respect the parameter convergence property as well as the use of only continuous signals.

Chapter (6) was the first attempt to implement simulated strategies in a real-time robotic platform. Few problems were encountered: from a good understanding of the set-up itself to the surface where the object slides. However, it was very interesting to test the strategies and see their effect on a real situation. Some effects were predicted by the simulation and others were new. Important assumptions, stated in Chapter (5), were tried to be held in the real experiment.
7-2 Thesis Contributions

On the basis of the results explained in (7-1), the following contributions are:

1. The research question itself. The question "if a simple continuous dynamic model could be use for an external rigid movable object" was not found in the literature and therefore the question poses new problems for the scientific community, regardless the bad or good results. In fact, if the question had already been asked and bad results achieved, it should have been studied and published to make the community aware of a wrong scientific direction.

2. A general understanding of parameter estimation techniques. Chapter (2) and (3) aimed at generalizing three different kinds of model-based parameter estimations strategies.
   - Parametric system identification techniques aims at an adaptation of plant parameter only. Adaptive control does not directly aim to identify parameter of the plant, but adapt online these unknown parameters and it feeds the adaptation in a control law used to drive the system to a general desired performance. Adaptive Observer is a combination of adaptive control and Observer design: it uses the same characteristic of adaptive control but, at the same time, tries to estimate unknown states of the plant.
   - Common elements are identified between these techniques. Because they all try to estimate parameters online.

3. Demonstration (in a matrix form) of recursive least-squares with forgetting factor as weighted least-squares. Although a clear relationship between $RLS_\kappa$ and weighted least-squares formulation, to my knowledge, the interpretation and demonstration of $RLS_\kappa$ in a matrix form as a weighted least-squares was never been explicitly explained: not found in [54, 53, 42, 52]. The interpretation $RLS_\kappa$ was inspired by [78]. But in [78] there is no mention (at all) of forgetting factor in general. And, to be clear, in [78], $RLS_\kappa$ was not found (and hence not demonstrated) as a weighted recursive least-squares.

4. A novel simulated experiment, with a first attempt on a robotic platform. The realization of the simulated experiment combined: the identification techniques selected and the local model characteristic. The goal was an online continuous estimation of the mass and the friction of the object. Although, the friction characteristic were limited to the Coulomb friction model, it was found effective to describe simulated behavior of a rigid movable object. The experiment voluntarily divided control architecture and online parameter architecture. The reason is inherent to the research question: a tactile exploration. Thus, the parameter estimation is not restricted to a control performance, i.e. follow a desired trajectory. However, the trajectory is fundamental in order to estimate object properties: it should allow the robot to push the object and it should excite the dynamics of the object.

Interesting behavior and estimation results of a robotic platform (performing the simulated experiment) were important to conclude the experience of the rigid movable object model. However, due to lack of time, the first attempt is far from concluding precise practical results.
Perspectives and Future Work

As was mentioned in the introduction, a vision system was not intended to be used. This decision should not be seen as a rejection of a future vision system, on the other hand, it should exploit exploration from a tactile point of view. The goal was to understand how effective a tactile interaction can be. When best results are achieved, then a future integration with a vision system could not only bring new challenges but also help the tactile interaction.

A first future work, from a simulation point of view, is the expansion of the experiment. The idea is to use a second finger on the other side of the object. In a simulated experiment this could only bring to a symmetry of the pushing finger experiment. But in reality, this could ensure a longer (or endless) interaction with the external object to obtain more information from the pushing experience. The goal should be to push the object forward and backward and try to better excite its friction dynamics.

Talking about dynamics, one may ask if it would be necessary to expand the friction model. The friction model considered can be enough. The reason for this decision is related to the simplicity of the model. The model should remain simple. A Coulomb+viscous description is not complete but probably effective for friction characteristic. Internal friction characteristic should be considered different from external ones: meanwhile the first does not allow a proper movement, creating the impossibility to track certain trajectories, for a pure exploration purpose too many values are not profitable in terms of estimation and a complex description of a friction model for a possible pushing action is not essential.

Figure 7-1: The two fingers represented by two actuated (u) masses (M) and the object (m) that can slide in two directions.

Figure 7-2: Compliant surface model (above) and Rigid Movable Object Model (below).
With our model, a new class of *model-based local environment* description was added to the ones reviewed in the literature. Assuming the absence of a vision system, a possible future architecture could include these two models. It would be interesting if the robot could identify which, between these two class of models, it is interacting with. The architecture should include a good decision process based on possible force velocity and position signal. The online decision should sequentially shape the desired trajectory to obtain the right interaction based on the model selected.

Even more interesting should be a mechanism that, while the robot interacts with the external object, does not decide which model the robot is interacting with but shapes *stiffness*, *friction* and *mass* properties in order to obtain a complete local description of the object. This mechanism could then identify a third class of object such as *deformable movable object*.

All this future work keeps in mind that the models are local. So a general description will never be achieved. However, their complexity could be increased to describe global properties, but this is outside the scope of the thesis.

The hope is that model-based techniques will be useful for a tactile-based interaction and possibly with the help of a vision system. Adaptive model-based techniques proved a theoretical global parameter convergence with Lyapunov theory, which is not possessed by intelligent algorithms.
Errata Corrige

• Errors in Chapter (5). All the Figures of the simulated experiment present the units [m]: position, velocity, error in velocity. It is correct but the position should be scaled by 0.1 as well as the velocity. Because the experiment consider a trajectory from 0 [m] to 0.3 [m]. This mistake can be found in Chapter (5) and in Appendix (A). Another mistake in Chapter (5)-(4) is the unit of measure for the $\hat{\Theta}$ vector. Taking the error function with the Coulomb friction:

$$
\epsilon(t) = y(t) - \left[ \left( \frac{\lambda - \frac{b_v}{m}}{\mu} \right) w_1(t) + \left( \frac{1}{m\mu} \right) w_2(t) + \left( \frac{-F_c}{m\mu} \right) w_3(t) \right] . \tag{7-1}
$$

The unit of measure of the error function is [m/s]. Thus, $\theta_1$ value should be dimensionless, $\theta_2$ should have [s/kg] dimension and $\theta_3$ value should have [m/s] dimension (cause of the filter on the sign). With this in mind, $\lambda$ and $\mu$ should have dimension [1/s]. The plots in the Thesis show wrong dimension for the thetas. In fact, taking the filter for the velocity signal:

$$
\dot{w}_1(t) = -\lambda w_1(t) + \mu y(t) , \tag{7-2}
$$

it is reasonable to give a unit of measure to $\mu$ of [1/s] as well as for $\lambda$, because $\dot{w}_1$ represents an acceleration. And the same unit can be given to the force filter.

• Error in Chapter (6). Because the position and velocity data was converted in [dm], then then the Coulomb friction should use [dm/s²]. The viscous coefficient, $b_v$, is independent on the velocity, so it should be considered 0.4 [Kg/s]. In Chapter (6), only the Coulomb friction model was taken into account to model the friction of the external object: this was not explicitly written. The unit of the $\Theta$ vector should change accordingly to the choice of [dm] and [dm/s].

• 'Table (??)' refers to Table (4-7).

• In Appendix (A-3), Experiment-1 $\rightarrow$ Experiment-2.
Appendix A

Additional Simulations and Experiments

A-1 Filter and Controller design

Remember the chosen values.

Table A-1: Filter and Controller Parameters

<table>
<thead>
<tr>
<th></th>
<th>$\lambda$</th>
<th>$\mu$</th>
<th>$m_d$</th>
<th>$k_p$</th>
<th>$k_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment-1</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Experiment-2</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>80</td>
<td>10</td>
</tr>
</tbody>
</table>
A-1-1 Filter Design

The low-pass filter selected has a structure of a first order system. The selection of the gains was not driven by any deep analysis. The reason is inherent with the use of the filter itself. For instance, the force profile of interaction cannot be known a priori. And it was not expected that the force profile could be an "exciting movement" with an high frequency. Few gains were tried, and the choice converged to choose both parameter $\mu = \lambda = 100$.

$$H_f(s) = \frac{\mu}{s\lambda + 1} = \frac{Y(s)}{U(s)}. \quad (A-1)$$

The only pole of the filter is $p = 100$. The maximum frequency to track a signal is $f_{\text{max}} = \frac{w_n}{2\pi} = 15.9 \,[Hz]$.

**Insight A-1.1.** If $U(s)$ is a step:

$$Y(s) = H_f(s)U(s) = \frac{H_f(s)}{s} \iff y(t) = \frac{\mu}{\lambda} \left(1 - e^{-t\lambda}\right) \quad (A-2)$$

**Proof.**

$$Y(s) = \frac{H_f(s)}{s} = \frac{\mu}{\lambda} \left(\frac{s}{s\lambda + 1}\right) \quad (A-3)$$

applying Heaviside’s method:

$$Y(s) = \frac{\mu}{\lambda} \left(\frac{s}{s\lambda + 1}\right) = \frac{A}{s} + \frac{B}{s\lambda + 1} = \frac{\mu}{\lambda} - \frac{\mu\lambda^2}{s\lambda + 1} = \frac{\mu}{\lambda} \left[\frac{1}{s} - \frac{1}{s + \lambda}\right] \quad (A-4)$$

In time domain

$$\frac{1}{s} \rightarrow \text{step}(t) \rightarrow 1, \quad \frac{1}{s - \alpha} \rightarrow e^{\alpha t} \text{step}(t) \rightarrow e^{\alpha t}$$

$$\Rightarrow y(t) = \frac{\mu}{\lambda} \left[1 - e^{-t\lambda}\right]. \quad (A-5)$$

**Table A-2:** Filter and Controller Parameters

<table>
<thead>
<tr>
<th>$y_\infty$</th>
<th>$T_{sett}$</th>
<th>$T_{rise}$</th>
<th>$w_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter characteristic</td>
<td>$\mu/\lambda$</td>
<td>$2.2/\lambda$</td>
<td>$0.7/\lambda$</td>
</tr>
</tbody>
</table>

Matteo Ciocca

Master of Science Thesis
A-1-2 Impedance Controller Design

*position-based impedance controller* for a single mass in one dimension is characterized by the following expression:

\[ u = \frac{m}{M_d} [\ddot{x}_d + k_d (\dot{x}_d - \dot{x}) + k_p (x_d - x)] + \left( \frac{m}{M_d} - 1 \right) F_{ext}. \]  

(A-6)

For \( k_d^2 < 4 k_p M_d \) (with our choices - always), the second-order linear system is characterized by a pair of asymptotically stable complex poles with *natural frequency* and *damping ratio* given by:

\[ w_n = \sqrt{\frac{k_p}{M_d}}, \quad D_{ratio} = \frac{k_d}{2 \sqrt{k_p M_d}}. \]  

(A-7)

In our experiments:

Experiment-1: \( w_n = 2,236 \ [rad/s], \ D_{ratio} = 0,67 \).

Experiment-2: \( w_n = 8,944 \ [rad/s], \ D_{ratio} = 0,55 \).

(A-8)

The second order system is given by the following transfer function:

\[ G(s) = \frac{1}{M_d s^2 + k_d s + k_p} \sim \frac{w_n^2}{s^2 + 2 D_{ratio} w_n s + w_n^2} \]  

(A-9)
Figure A-1: Above: Bode plots, step response and tracking of a sinusoidal $u = \sin(10t)$, frequency $f = 10/2\pi = 1.59 \ [Hz]$. Below: Bode plots, poles and zeros map and step response. $G_i(\cdot)$ representing the mass-spring-damper transfer function for the $i$ Experiment.
A-2 Third and Wood object for viscous friction coefficient

Here are reported the simulation plots for the case of the objects from Experiment-1 not showed in Chapter (5) with $RLS_\kappa$ using the different approach in Section (5-5).
Figure A-2: Experiment-1, third object. Relevant measurements (above) of the robot finger ($\cdot r$), the object ($\cdot o$) and desired trajectory ($\cdot d$): position, velocities. Filtered signals and control input (below).
Figure A-3: Experiment-1, wood object. Relevant measurements (above) of the robot finger ($\cdot$) <sub>r</sub>, the object ($\cdot$) <sub>o</sub> and desired trajectory ($\cdot$) <sub>d</sub>: position, velocities. Filtered signals and control input (below).
Figure A-4: Experiment 1, third object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for the discrete version RLS algorithm, with a forgetting factor $\kappa = 0.85$. 

Forgetting factor $\kappa = 0.85$. 

Figure A-5: Experiment-1, third object. Estimation of the \( \hat{\Theta} \) vector and the error function \( \epsilon \), for the discrete version \( RLS_\kappa \) algorithm, with a forgetting factor \( \kappa = 0.85 \).
Figure A-6: Experiment-1, third object (above), wood object (below). Evolution of the trace of the covariance matrix, \( P \), for the discrete version \( RLS_\infty \) algorithm, initialized for both algorithms with: \( P(0) = \alpha I, \alpha = 100 \).
From Figure (A-3) is clear that when the object and the robot are in contact, the error of their velocity does not converge to zero, hence good results are not expected. The error in the velocity is due to the hybrid architecture with the force sensor modeled as a spring and damper in parallel. In reality, the velocity of the object and the robot should be equal during the touching phase (without considering the noise factor).

Table A-3: Experiment-1: wood and third object models parameters and nominal $\Theta_{\text{touch}}$ vector. $T_p = 1\ [s]$, $\delta_{\hat{\theta}_1} = 10^{-5}$ and $\delta_{\hat{\theta}_2} = 10^{-3}$.

<table>
<thead>
<tr>
<th>$\Theta_{\text{touch}} + RLS_{\theta}$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\hat{\theta}_1$</th>
<th>$\hat{\theta}_2(13.0\ s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>third</td>
<td>0.0819</td>
<td>0.00083</td>
<td>$\sim$</td>
<td>0.00074</td>
</tr>
<tr>
<td>$\Theta_{\text{touch}} + RLS_{\theta}$</td>
<td>$\theta_1$</td>
<td>$\theta_2$</td>
<td>$\theta_1(13.4\ s)$</td>
<td>$\theta_2(13.0\ s)$</td>
</tr>
<tr>
<td>wood</td>
<td>0.9983</td>
<td>0.0041</td>
<td>0.9983</td>
<td>0.00416</td>
</tr>
</tbody>
</table>

A-3 Third and Steel object for Coulomb friction

Here are reported the simulation plots for the case of the objects from Experiment-2 not showed in Chapter (5) with $RLS_{\theta}$ using the different approach in Section (5-5).
Figure A-7: Experiment-2, third object. Relevant measurements (above) of the robot finger \((\cdot)_r\), the object \((\cdot)_o\) and desired trajectory \((\cdot)_d\): position, velocities. Filtered signals and control input (below).
Figure A-8: Experiment-2, steel object. Relevant measurements (above) of the robot finger (r), the object (o) and desired trajectory (d): position, velocities. Filtered signals and control input (below).
Forgetting factor $\kappa = 0.99^8$.

Figure A-9: Experiment 2, third object. Estimation of the $\hat{\theta}$ vector and the error function $\epsilon$ for the discrete version RLS algorithm, with a forgetting factor $\kappa = 0.99^8$. 

Matteo Ciocca
Master of Science Thesis
Figure A-10: Experiment-2, steel object. Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, for the discrete version $RLS_\kappa$ algorithm, with a forgetting factor $\kappa = 0.85$. 

Master of Science Thesis Matteo Ciocca
Figure A-11: Experiment-2, third object (above), steel object (below). Evolution of the trace of the covariance matrix, $\mathcal{P}$, for the discrete version $RLS_{\kappa}$ algorithm, initialized for both algorithms with: $\mathcal{P}(0) = \alpha I$, $\alpha = 100$. 

Matteo Ciocca

Matteo Ciocca

Master of Science Thesis
From Figure (A-7)-(A-8), the same error in velocity, due to the interaction model, is visible. The error is the cause of the incorrect estimation for the parameters. The covariance matrices (A-11) present the same problem at the end of the simulation as explained in Chapter (5).

Table A-4: Experiment-2: steel and third object models parameters and nominal $\Theta_{touch}$ vector.

<table>
<thead>
<tr>
<th>$\Theta_{touch}+RLS_{\kappa}$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\hat{\theta}_1(27.6s)$</th>
<th>$\hat{\theta}_2(27.2s)$</th>
<th>$\hat{\theta}_3(27.2s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>steel</td>
<td>0.999</td>
<td>0.0010</td>
<td>-0.0038</td>
<td>0.9981</td>
<td>0.0009</td>
<td>-0.0033</td>
</tr>
<tr>
<td>third</td>
<td>0.0819</td>
<td>0.00083</td>
<td>-0.001</td>
<td>0.9749</td>
<td>0.0082</td>
<td>-0.00078</td>
</tr>
</tbody>
</table>

A-4 Estimation with only a polynomial trajectory

Although the covariance matrix is bounded and a not precise validation strategy for the parameters, it could be inferred that the parameter estimation is worst that the ones achieved in Chapter (6).

From a deterministic point of view it was explained that a simple polynomial trajectory could converge to an undesired case of the Table (4-7). However, looking at Figure (A-13), behavior of the estimates were found interesting. During the **Cruising Phase** (around 15 [s]) seems to be that the estimate converge to the empirical parameters given in Chapter (6). It is instead assumed that the parameter converge as follows: $\hat{\theta}_1 \to 1, \hat{\theta}_2 \to 0$ and $\hat{\theta}_3 \to 0$, concluding a wrong estimation. (i) Before 15 [s], $\hat{\theta}_1$ stabilize around values greater than 1 and (ii) the pronounced fluctuation of the estimate vector $\hat{\Theta}$ could be additional due to the wrong excitation. In this case, techniques such as **excitation subspace** would be found interesting to test.

The only use of a polynomial trajectory was assumed to do not explore the limited state-space (a state $[y_l]$) of a rigid movable object model with a tactile interaction. Hence, it was assumed not a useful trajectory, although the boundedness of the covariance matrix.
Figure A-12: Experiment-\((b)\). Above, the matrix entries of the covariance matrix. Below, the evolution of the trace of the covariance matrix (with ZOOM-IN), the discrete version $RLS_\kappa$ algorithms, initialized with: $P(0) = \alpha I$, $\alpha = 0.01$. 
Figure A-13: Experiment-(b). Estimation of the $\hat{\Theta}$ vector and the error function $\epsilon$, discrete version $RLS_\kappa$ algorithms, with a forgetting factor $\kappa = 0.99$. 
Figure A.14: Experiment (b). Force and velocity signal: real and filtered. Below the three components of the regressor vector.
Mistakes in the Literature

Although it is not the main focus of the thesis, during the (new) literature survey several papers were reviewed. In what follows, two papers has been found to have few mistakes. I want to report here the two corrections. I think the observations done by me are correct. If this is the case, this appendix aims to show where the mistakes are. The reason to highlight these mistake is the following. In Chapter (2) a wide literature survey was conducted. In this survey, several papers where reviewed. On top of that, some papers were taken into consideration because found very interesting. At first, a simple read could be enough, but when the algorithms must be implemented for testing, a deeply understanding should be conducted. The understanding shed light on few mistakes. And, i.e. [98], with the impossibility of the implementation. Thus, bad for the scientific community. Still in [98] questions are open, in fact, in the paper there is not a single simulation.

B-1 Adaptive regulation of impact induced forces for three degree of freedom collisions: a backstepping approach [98]

System used:

\[
\begin{align*}
m_0 \ddot{x}_0 + a_0 (x_0 - x_1 + r_{01}) &= u \\
m_1 \ddot{x}_1 - a_0 (x_0 - x_1 + r_{01}) &= d
\end{align*}
\]  

(B-1)

Where:

\[
d = \theta \psi(x_1)m_1 , \quad \psi(x_1) = -\frac{b_1}{m_1}x_1
\]  

(B-2)

Where \( \theta \) represent the binary index of contact. Use the change of coordinates:

\[
\begin{align*}
z_1 &= x_1 \\
z_2 &= \dot{x}_1 \\
z_3 &= \frac{a_0 (x_0 - x_1 + r_{01})}{m_1} \\
z_4 &= \frac{a_0 (\dot{x}_0 - \dot{x}_1)}{m_1}
\end{align*}
\]  

(B-3)
Deriving the dynamical system form this change of coordinates:

\[
\begin{align*}
\dot{z}_1 &= \dot{x}_1 \\
\dot{z}_2 &= \dot{x}_1 \\
\dot{z}_3 &= \frac{a_0(x_0-x_1)}{m_1} \\
\dot{z}_4 &= \frac{a_0(x_0-x_1)}{m_1} 
\end{align*}
\]  

(B-4)

\[
\begin{align*}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= \frac{a_0(x_0-x_1) + d}{m_1} = \frac{a_0(x_0-x_1 + r_01)}{m_1} + \frac{d}{m_1} = z_3 + \frac{\theta \psi(x_1)m_1}{m_1} = z_3 + \theta \psi(z_1) \\
\dot{z}_3 &= z_4 \\
\dot{z}_4 &= \frac{a_0(x_0-x_1)}{m_1} = \frac{a_0}{m_1} \left( \left[ \frac{u-a_0(x_0-x_1+1)}{m_0} \right] - \left[ \frac{d+a_0(x_0-x_1+1)}{m_1} \right] \right) \\
&= \frac{a_0}{m_1m_0}u - \frac{a_0}{m_1}d - \frac{a_0}{m_1} \left[ \frac{1}{m_0} + \frac{1}{m_1} \right] (x_0 - x_1 + r_01) \\
&= \frac{a_0}{m_1m_0}u - \frac{a_0}{m_1} \theta \psi(z_1)m_1 - \frac{a_0}{m_1} \left[ \frac{1}{m_0} + \frac{1}{m_1} \right] (x_0 - x_1 + r_01) \\
&= \frac{a_0}{m_1m_0}u - \frac{a_0}{m_1} \theta \psi(z_1) - \frac{a_0}{m_1} \left[ \frac{1}{m_0} + \frac{1}{m_1} \right] (x_0 - x_1 + r_01) \\
\end{align*}
\]  

(B-5)

\[
\begin{align*}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 + \theta \psi \\
\dot{z}_3 &= z_4 \\
\dot{z}_4 &= h_uu + h_v \theta \psi + g(x_0, x_1) 
\end{align*}
\]  

(B-6)

Where,

\[
\begin{align*}
h_u &= \frac{a_0}{m_0m_1} \\
h_v &= -\frac{a_0}{m_1} \\
g &= -\frac{a_0}{m_1} \left( \frac{m_1+m_0}{m_1m_0} \right) (x_0 - x_1 + r_01)
\end{align*}
\]  

(B-7)

**B-1-1 Design procedure: Adaptive Backstepping**

**Step 1:** Define \( \bar{z}_1 = x_1 \) calling \( y_1 = z_1 - \bar{z}_1 \) and \( y_2 = z_2 - \alpha_1(z_1) \). Choose for a subsystem in \( y \)-coordinates, the following Lyapunov function to prove stability:

\[
V_1 = \frac{1}{2} y_1^2 
\]  

(B-8)

Deriving the Lyapunov function:

\[
\dot{V}_1 = y_1 \dot{y}_1 \\
= y_1 (\dot{z}_1) \\
= y_1 (z_2) \\
= y_1(y_2 + \alpha_1) \\
= y_1 y_2 + y_1 \alpha_1
\]  

(B-9)

In the subsystem where \( y_2 = 0 \), the stabilizing function is:

\[
\alpha_1 = -c_1 y_1 
\]  

(B-10)

Making:

\[
\dot{V}_1 = -c_1 y_1^2 
\]  

(B-11)
Step 2: Define \( y_3 = z_3 - \alpha_2(z_1, z_2, \dot{\theta}) \). Choose for a subsystem in \( y \)-coordinates, the following Lyapunov function to prove stability:

\[
V_2 = V_1 + \frac{1}{2} y_2^2 + \frac{1}{2\gamma} (\dot{\theta} - \theta)^2
\]  
\[ \text{(B-12)} \]

Deriving the Lyapunov function:

\[
\dot{V}_2 = \dot{V}_1 + y_2 \dot{y}_2 + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= y_1 y_2 + y_1 \alpha_1 + y_2 (\dot{z}_2 - \dot{\alpha}_1) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 + y_1 y_2 + y_2 \left( z_3 + \theta \psi(z_1) - \frac{\partial \alpha_1}{\partial z_1} \dot{z}_1 \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 + y_1 y_2 + y_2 \left( y_3 + \alpha_2 + \theta \psi(z_1) - \frac{\partial \alpha_1}{\partial z_1} \dot{z}_2 \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]  
\[ \text{(B-13)} \]

Now choose the second stabilizing function as:

\[
\alpha_2 = -c_2 y_2 - y_1 + \frac{\partial \alpha_1}{\partial z_1} \dot{z}_2 - \dot{\theta} \psi(z_1)
\]

\[ \text{(B-14)} \]

Substituting the new stabilizing function:

\[
\dot{V}_2 = -c_1 y_1^2 + y_1 y_2 + y_2 \left( y_3 + \alpha_2 + \theta \psi(z_1) - \frac{\partial \alpha_1}{\partial z_1} \dot{z}_2 \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 + y_1 y_2 + y_2 \left( y_3 - c_2 y_2 - y_1 + \frac{\partial \alpha_1}{\partial z_1} \dot{z}_2 - \dot{\theta} \psi(z_1) + \theta \psi(z_1) - \frac{\partial \alpha_1}{\partial z_1} \dot{z}_2 \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 + y_1 y_2 + y_2 \left( y_3 - c_2 y_2 - y_1 - \dot{\theta} \psi(z_1) + \theta \psi(z_1) \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 - y_2 \dot{\theta} \psi(z_1) + y_2 \theta \psi(z_1) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 - y_2 \psi(z_1) \left( \dot{\theta} - \theta \right) + \frac{1}{\gamma} (\dot{\theta} - \theta) \dot{\theta}
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma} (\dot{\theta} - \theta) \left( \dot{\theta} - y_2 \gamma \psi(z_1) \right)
\]  
\[ \text{(B-15)} \]

The first tuning function is:

\[
\tau_1 = y_2 \gamma \psi(z_1)
\]

\[ \text{(B-16)} \]

If \( y_3 = 0 \):

\[
\dot{V}_2 = -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma} (\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right)
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 + \frac{1}{\gamma} (\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right)
\]  
\[ \text{(B-17)} \]

Choosing \( \dot{\theta} = \tau_1 \), we could stabilizing the new system to \((y_1, y_2) = (0, 0)\),

\[
\dot{V}_2 = -c_1 y_1^2 - c_2 y_2^2
\]  
\[ \text{(B-18)} \]

But the choice of \( \dot{\theta} \) is postponed to the very last step.
Step 3: Let introduce a new variable $y_4 = z_4 - \alpha_3(z_1, z_2, z_3, \hat{\theta})$ and a new Lyapunov function:

$$V_3 = V_2 + \frac{1}{2}y_3^2$$ (B-19)

Taking the time derivative of $V_3$:

$$\dot{V}_3 = \dot{V}_2 + y_3 \dot{y}_3$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) + y_3 \dot{y}_3$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) + y_3 (z_4 - \alpha_2)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) + y_3 (z_4 - \alpha_2)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) + y_3 (y_4 + \alpha_3 - \alpha_2)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) +$$

$$+ y_3 \left( y_4 + \alpha_3 - \frac{\partial \alpha_2}{\partial z_1} z_2 - \frac{\partial \alpha_2}{\partial z_2} z_3 - \frac{\partial \alpha_2}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_2}{\partial \theta} \dot{\theta} \right)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) +$$

$$+ y_3 \left( y_4 - c_3 y_3 - y_2 + \frac{\partial \alpha_2}{\partial z_1} z_2 + \frac{\partial \alpha_2}{\partial z_2} z_3 + \frac{\partial \alpha_2}{\partial \theta} \dot{\theta} \psi(z_1) + \frac{\partial \alpha_2}{\partial \theta} \dot{\theta} \right)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) +$$

$$+ y_3 \left( y_4 - c_3 y_3 - y_2 + \dot{\theta} \frac{\partial \alpha_2}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_2}{\partial \theta} \dot{\theta} \right)$$

$$= -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 \right) +$$

$$+ y_3 \left( y_4 - c_3 y_3 - y_2 + \dot{\theta} \frac{\partial \alpha_2}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_2}{\partial \theta} \dot{\theta} \right)$$

$$\dot{V}_3 = -c_1 y_1^2 - c_2 y_2^2 + y_2 y_3 + \frac{1}{\gamma}(\dot{\theta} - \theta) \left( \dot{\theta} - \tau_1 + \gamma y_3 \frac{\partial \alpha_2}{\partial \theta} \psi(z_1) \right) - y_3 \frac{\partial \alpha_2}{\partial \theta} \left( \dot{\theta} - \tau_2 \right)$$ (B-22)
Let’s define the second tuning function:

\[ \tau_2 = \tau_1 - \gamma y_3 \frac{\partial \alpha_2}{\partial z_2} \psi(z_1) \] (B-23)

Substituting:

\[ \dot{V}_3 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) \] (B-24)

Again, if \( y_4 = 0 \) and \( \dot{\theta} = \tau_2 \),

\[ \dot{V}_3 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 \] (B-25)

**Step 4:** Let’s introduce the last Lyapunov function:

\[ V_4 = V_3 + \frac{1}{2} y_4^2 \] (B-26)

Deriving the Lyapunov function:

\[ \dot{V}_4 = \dot{V}_3 + y_4 \dot{y}_4 \]

\[ = \dot{V}_3 + y_4 (\dot{z}_4 - \dot{\alpha}_3) \]

\[ = \dot{V}_3 + y_4 \left( h_u u + h_\psi \theta \psi(z_1) + g(x_0, x_1) - \dot{\alpha}_3 \right) \]

\[ = \dot{V}_3 + y_4 \left( h_u u + h_\psi \theta \psi(z_1) + g(x_0, x_1) - \frac{\partial \alpha_3}{\partial z_2} z_2 - \frac{\partial \alpha_3}{\partial z_3} z_3 - \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_3}{\partial \theta} z_4 - \frac{\partial \alpha_3}{\partial \theta} \dot{\theta} \right) \]

\[ = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) \]

\[ + y_4 \left( h_u u + h_\psi \theta \psi(z_1) + g(x_0, x_1) - \frac{\partial \alpha_3}{\partial z_2} z_2 - \frac{\partial \alpha_3}{\partial z_3} z_3 - \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_3}{\partial \theta} z_4 - \frac{\alpha_3}{\partial \theta} \dot{\theta} \right) \] (B-27)

Let’s choose the control function:

\[ u = h_u^{-1} \left[ -h_\psi \dot{\psi}(z_1) - g(x_0, x_1) - y_3 - c_4 y_4 + \frac{\partial \alpha_3}{\partial z_2} z_2 + \frac{\partial \alpha_3}{\partial z_3} z_3 + \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) + \frac{\partial \alpha_3}{\partial \theta} \tau_3 + u_2 \right] \] (B-28)
Substituting this control law:

\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) + y_4 \left( h_u u + h_\phi \theta \psi(z_1) + g(x_0, x_1) - \frac{\partial \alpha_3}{\partial z_2} z_2 - \frac{\partial \alpha_3}{\partial z_3} z_3 - \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_3}{\partial \theta} \dot{z}_4 - \frac{\partial \alpha_3}{\partial \theta} \dot{\theta} \right)
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) + y_4 \left( -h_\phi \dot{\theta} \psi(z_1) - y_3 - c_4 y_4 + \frac{\partial \alpha_3}{\partial z_2} \theta \psi(z_1) + \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_3}{\partial \theta} \dot{\theta} \right)
\]

\[
= -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) + y_4 \left( -h_\phi \dot{\theta} \psi(z_1) - y_3 - c_4 y_4 + \frac{\partial \alpha_3}{\partial z_2} \theta \psi(z_1) + \frac{\partial \alpha_3}{\partial \theta} \psi(z_1) - \frac{\partial \alpha_3}{\partial \theta} \dot{\theta} \right)
\]

\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2 - \gamma y_4 \dot{\theta} \psi(z_1) + \gamma y_4 \frac{\partial \alpha_3}{\partial z_2} \psi(z_1)) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) + y_4 u_2 - y_4 \frac{\partial \alpha_3}{\partial \theta} (\dot{\theta} - \tau_3)
\]

\[ \text{(B-29)} \]

I can now introduce the last (third) tuning function:

\[
\tau_3 = \tau_2 + \gamma y_4 \dot{\theta} \psi(z_1) - \gamma y_4 \frac{\partial \alpha_3}{\partial z_2} \psi(z_1)
\]

\[ \text{(B-30)} \]

Substituting:

\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 + y_3 y_4 + \frac{1}{\gamma} (\dot{\theta} - \theta) (\dot{\theta} - \tau_2 - \gamma y_4 \dot{\theta} \psi(z_1) + \gamma y_4 \frac{\partial \alpha_3}{\partial z_2} \psi(z_1)) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\theta} - \tau_2) + y_4 u_2 - y_4 \frac{\partial \alpha_3}{\partial \theta} (\dot{\theta} - \tau_3)
\]

\[ \text{(B-31)} \]

As we see (in the scientific paper), \( \tau_2 \) still appear in the derivation, in order to cancel it, \( u_2 \) must be used\(^1\):

\[
u_2 = y_3 \gamma \dot{\theta} \psi(z_1) \frac{\partial \alpha_2}{\partial \theta} \left( h_\phi - \frac{\partial \alpha_3}{\partial z_2} \right)
\]

\[ \text{(B-32)} \]

\(^1\) My addition
Substituting this last equation:
\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 - c_4 y_4^2 + \frac{1}{\gamma}(\hat{\theta} - \theta)(\dot{\hat{\theta}} - \tau_3) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\hat{\theta}} - \tau_2) + y_4 u_2 - y_4 \frac{\partial \alpha_2}{\partial \theta} (\dot{\hat{\theta}} - \tau_3)
\]
\[
= -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 - c_4 y_4^2 + \frac{1}{\gamma}(\hat{\theta} - \theta)(\dot{\hat{\theta}} - \tau_3) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\hat{\theta}} - \tau_2) + y_4 \left( y_3 \gamma \psi(z_1) \frac{\partial \alpha_2}{\partial \theta} \left( h - \frac{\partial \alpha_3}{\partial \theta} \right) \right) - y_4 \frac{\partial \alpha_3}{\partial \theta} (\dot{\hat{\theta}} - \tau_3)
\]
\[
= -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 - c_4 y_4^2 + \frac{1}{\gamma}(\hat{\theta} - \theta)(\dot{\hat{\theta}} - \tau_3) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\hat{\theta}} - \tau_2) + y_4 \frac{\partial \alpha_2}{\partial \theta} \left( \dot{\hat{\theta}} - \tau_2 - y_4 \gamma \psi(z_1) \left( h - \frac{\partial \alpha_3}{\partial \theta} \right) \right) - y_4 \frac{\partial \alpha_3}{\partial \theta} (\dot{\hat{\theta}} - \tau_3)
\]
\[\text{(B-33)}\]

Finally:
\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 - c_4 y_4^2 + \frac{1}{\gamma}(\hat{\theta} - \theta)(\dot{\hat{\theta}} - \tau_3) - y_3 \frac{\partial \alpha_2}{\partial \theta} (\dot{\hat{\theta}} - \tau_2) - y_4 \frac{\partial \alpha_3}{\partial \theta} (\dot{\hat{\theta}} - \tau_3) \quad \text{(B-34)}
\]

It is now the time to choose \( \dot{\hat{\theta}} \), we choose it equals to \( \tau_3 \), and the time derivative of \( V_4 \) becomes:
\[
\dot{V}_4 = -c_1 y_1^2 - c_2 y_2^2 - c_3 y_3^2 - c_4 y_4^2 \quad \text{(B-35)}
\]

### B-1-2 Summary of the Adaptive Backstepping control law

System (Real Plant):
\[
\begin{align*}
  m_0 \ddot{x}_0 + a_0 (x_0 - x_1 + r_{01}) &= u \\
  m_1 \ddot{x}_1 - a_0 (x_0 - x_1 + r_{01}) &= d
\end{align*}
\]

Where:
\[
d = \theta \psi(x_1) m_1, \quad \psi(x_1) = -\frac{b_1}{m_1} x_1
\]

Change of Coordinates:
\[
\begin{align*}
  z_1 &= x_1 \\
  z_2 &= \dot{x}_1 \\
  z_3 &= \frac{a_0 (x_0 - x_1 + r_{01})}{m_1} \\
  z_4 &= \frac{a_0 (x_0 - x_1)}{m_1}
\end{align*}
\]

All the tuning functions:
\[
\begin{align*}
  \tau_1 &= y_2 \gamma \psi(z_1) \\
  \tau_2 &= \tau_1 - \gamma y_3 \frac{\partial \alpha_2}{\partial \theta} \psi(z_1) \\
  \tau_3 &= \tau_2 + \gamma y_4 \psi(z_1) \left( h - \frac{\partial \alpha_3}{\partial \theta} \right)
\end{align*}
\]

Master of Science Thesis  Matteo Ciocca
Choosing: $\dot{\theta} = \tau_3$.

All the stabilizing functions:

$$
\begin{align*}
\alpha_1 &= -c_1 y_1 \\
\alpha_2 &= -c_2 y_2 - y_1 + \frac{\partial \alpha_1}{\partial z_1} z_2 - \dot{\psi}(z_1) \\
\alpha_3 &= -c_3 y_3 - y_2 + \frac{\partial \alpha_2}{\partial z_1} z_2 + \frac{\partial \alpha_2}{\partial z_2} z_3 + \frac{\partial \alpha_2}{\partial z_3} \dot{\psi}(z_1) + \frac{\partial \alpha_2}{\partial \theta} \tau_2 
\end{align*}
$$

(B-40)

The Control law:

$$
u = h_u^{-1} \left[ -h_\psi \dot{\psi}(z_1) - g(x_0, x_1) - y_3 - c_4 y_4 + \frac{\partial \alpha_3}{\partial z_2} z_2 + \frac{\partial \alpha_3}{\partial z_3} z_3 + \frac{\partial \alpha_3}{\partial z_4} \dot{\psi}(z_1) + \frac{\partial \alpha_3}{\partial \theta} \tau_3 + u_2 \right]$$

(B-41)

Where (My addition):

$$
u_2 = y_3 \gamma \psi(z_1) \frac{\partial \alpha_2}{\partial z_2} \left( h_\psi - \frac{\partial \alpha_3}{\partial z_2} \right)
$$

(B-42)

and,

$$
\begin{align*}
h_u &= \frac{a_0}{m_0 m_1} \\
h_\psi &= -\frac{a_0}{m_1} \\
g &= -\frac{a_2}{m_1} \left( \frac{m_1 + m_2}{m_1 m_0} \right) (x_0 - x_1 + \tau_0)
\end{align*}
$$

(B-43)

B-2 Adaptive Regulation of Impact Induced Forces for One-Degree-of-Freedom Collisions [45]

I found the paper where the case of a single DOF is analyzed. The system is the following:

$$
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{m} (-d(x_1) + u)
\end{align*}
$$

(B-44)

Where $d(x_1)$ is:

$$
\begin{align*}
d(\xi) &= 0 \quad , \text{if } \xi < 0 \\
d(\xi) &= k\xi \quad , \text{if } \xi \geq 0
\end{align*}
$$

(B-45)

The problem wants to find an equilibrium point defined by a constant reference force value on this deformable/rigid wall: $\phi_r = k x_{1e}$. For each $\phi_r \geq 0$, should be a dynamic feedback law that stabilize the system in contact and non contact:

$$
\begin{align*}
\dot{\theta} &= g(\theta, x_1, x_2, \phi_r) \\
u &= h(\theta, x_1, x_2, \phi_r)
\end{align*}
$$

(B-46)

The control law is the following\footnote{copied from the paper}:

$$
u = -m_2 x_2 - \frac{k}{\beta} x_2 + k x_{1e} - k \theta - k_\psi (x_2 + \beta (x - x_{1e}))
$$

(B-47)
Where $\beta$, $k_v$ are two scalar gains. The update law $\dot{\theta}$ not yet defined. Closing the loop:

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{m}_x &= -m\beta x_2 + \tau(x_1, x_2, x_{1e}) \\
\tau &= -k_v(x_2 + \beta(x_1 - x_{1e})) - \frac{k}{\beta} x_2 - d(x_1) + k x_{1e} - k \theta
\end{align*}$$

(B-48)

Now, we achieve the first change of coordinates:

$$\begin{align*}
\dot{q}_1 &= x_1 - x_{1e} \\
\dot{q}_2 &= x_2 + \beta(x_1 - x_{1e}) \\
\dot{\theta} &= \begin{cases} 
\theta - x_{1e}, & \text{if } x_1 < 0 \\
\theta, & \text{if } x_1 \geq 0
\end{cases}
\end{align*}$$

(B-49)

Whose inverse is:

$$\begin{align*}
x_1 &= \dot{q}_1 + x_{1e} \\
x_2 &= \dot{q}_2 - \beta \dot{q}_1 \\
\theta &= \begin{cases} 
\dot{\theta} + x_{1e}, & \text{if } \dot{q}_1 < -x_{1e} \\
\dot{\theta}, & \text{if } \dot{q}_1 \geq -x_{1e}
\end{cases}
\end{align*}$$

(B-50)

Closing the loop with these new variables $(q^T, \dot{\theta})$:

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{d}(\dot{q}_1 + x_{1e}) &= -\beta \dot{q}_1 + \dot{q}_2 \\
\dot{\dot{q}}_1 &= -\beta \dot{q}_1 + \dot{q}_2
\end{align*}$$

(B-51)

The second equation (contact in blue):

$$\begin{align*}
m \dot{x}_2 &= -m\beta x_2 - k_v(x_2 + \beta(x_1 - x_{1e})) - \frac{k}{\beta} x_2 - d(x_1) + k x_{1e} - k \theta \\
m \dot{d}(\dot{q}_2 - \beta \dot{q}_1) &= -m\beta(\dot{q}_2 - \beta \dot{q}_1) - k_v [(\dot{q}_2 - \beta \dot{q}_1) + \beta \dot{q}_1] - \frac{k}{\beta} (\dot{q}_2 - \beta \dot{q}_1) - k (\dot{q}_1 + x_{1e}) + k x_{1e} - k \theta \\
m \dot{q}_2 - m\beta \dot{q}_1 &= -m\beta \dot{q}_2 - m\beta^2 \dot{q}_1 - k_v \ddot{q}_2 - \frac{k}{\beta} \ddot{q}_2 + k \ddot{q}_1 - k \ddot{q}_1 - k x_{1e} + k x_{1e} - k \ddot{\theta} \\
m \dot{q}_2 - m\beta(\beta \dot{q}_1 + \ddot{q}_2) &= -m\beta \dot{q}_2 - m\beta^2 \dot{q}_1 - k_v \ddot{q}_2 - \frac{k}{\beta} \ddot{q}_2 - k \ddot{\theta} \\
m \dot{q}_2 + m\beta^2 \ddot{q}_1 - m\beta \ddot{q}_2 &= -m\beta \dot{q}_2 - m\beta^2 \ddot{q}_1 - k_v \ddot{q}_2 - \frac{k}{\beta} \ddot{q}_2 - k \ddot{\theta} \\
m \dot{q}_2 &= -k_v \ddot{q}_2 - \frac{k}{\beta} \ddot{q}_2 - k \ddot{\theta}
\end{align*}$$

system in contact

$$\begin{align*}
m \dot{q}_2 &= -k_v \ddot{q}_2 - \frac{k}{\beta} \ddot{q}_2 - \sigma(\ddot{q}, \ddot{\theta}, x_{1e}) \\
\sigma(\ddot{q}, \ddot{\theta}, x_{1e}) &= k \ddot{\theta}
\end{align*}$$

(B-52)
When the system is not in contact:

$$m\dot{q}_2 = -k_0q_2 - \frac{k}{\beta} q_2 + k\bar{q}_1 + kx_{1e} - k(\bar{q} + x_{1e})$$

they are not cancelled anymore

$$m\dot{q}_2 = -k_0q_2 - \frac{k}{\beta} q_2 + k\bar{q}_1 + kx_{1e} - k\bar{q} - kx_{1e}$$  \hfill (B-53)

system in contact

$$\begin{align*}
m\dot{q}_2 &= -k_0q_2 - \frac{k}{\beta} q_2 - k\bar{q}_1 + k\bar{q} + kx_{1e} - kx_{1e} \\
\sigma(q, \bar{q}, \bar{q}_1) &= k\bar{q} - kq_1
\end{align*}$$

Secondly, we apply another change of coordinates:

$$\begin{align*}
\bar{z}_1 &= \bar{q}_1 \\
\bar{z}_2 &= \bar{q}_2 \\
\bar{z}_3 &= \bar{q}_2 + \lambda \bar{\theta}
\end{align*}$$  \hfill (B-54)

Where $\lambda > 0$, and writing the closed-loop again:

$$\begin{align*}
\dot{\bar{z}}_1 &= -\beta \bar{z}_1 + \bar{z}_2 \\
m\dot{\bar{z}}_2 &= -k_0\bar{q}_2 - \frac{k}{\beta} \bar{z}_2 - \rho(\bar{z}, x_{1e}) \\
\rho(\bar{z}, x_{1e}) &= \begin{cases} \\
\frac{k}{\beta} (\bar{z}_3 - \bar{z}_2) - k\bar{z}_1, & \text{if } \bar{z}_1 < -x_{1e} \\
\frac{k}{\beta} (\bar{z}_3 - \bar{z}_2), & \text{if } \bar{z}_1 \geq -x_{1e}
\end{cases} \\
\dot{\bar{z}}_3 &= -\left( \frac{k_0}{m} + \frac{k}{m\beta} \right) \bar{z}_2 - \frac{1}{m} \rho(\bar{z}, x_{1e}) + \lambda \bar{\theta}
\end{align*}$$  \hfill (B-55)

A Lyapunov function to determine the stability of the system will give me the answer to find an update law for $\dot{\bar{\theta}}$:

$$V = \frac{1}{2} k \bar{z}_1^2 + \frac{1}{2} m \bar{z}_2^2 + \frac{1}{2\gamma} \bar{z}_3^2$$  \hfill (B-56)

Deriving the Lyapunov function:

$$\dot{V} = -k\beta \bar{z}_1^2 + k\bar{z}_1 \bar{z}_2 \left( k_v + \frac{k}{\beta} \right) \bar{z}_2^2 - \left( k_v + \frac{k}{\beta} \right) \bar{z}_2 \bar{z}_3 \left( m \gamma + \frac{k}{m\gamma} \right) \bar{z}_3^2 + \frac{\lambda}{\gamma} \bar{z}_3 \bar{\theta}$$  \hfill (B-57)

During the non-contact phase, $\bar{z}_1 < -x_{1e}$:

$$\dot{V} = -k\beta \bar{z}_1^2 + 2k \bar{z}_1 \bar{z}_2 \left( k_v + \frac{k}{\beta} \right) \bar{z}_2^2 + \frac{k}{m\gamma} \bar{z}_1 \bar{z}_3 \left( \eta + \frac{k}{\lambda} \right) \bar{z}_3^2 + \frac{k}{m\gamma} \bar{z}_1 \bar{\theta}$$  \hfill (B-58)

Where:

$$\eta = \left( k_v + \frac{k}{\beta} - \frac{k}{\lambda} \right)$$

$$\dot{\bar{\theta}} = \frac{\gamma}{\lambda} \left[ \left( \frac{\eta}{m\gamma} + \frac{k}{\lambda} \right) \bar{z}_2 - \frac{k}{m\gamma} \bar{z}_1 \right]$$  \hfill (B-59)
I added this term in order to arrive at the same solution, the term was not showed in the paper! Substituting:

\[
\dot{V} = -k\beta \ddot{z}_1^2 + 2k\ddot{z}_1 \dddot{z}_2 - \left(k_v + \frac{k}{\beta}\right) \dddot{z}_2^2 + \frac{k}{m\gamma} \ddot{z}_1 \dddot{z}_3 - \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \dddot{z}_3 \dddot{z}_2 + \\
+ \frac{k}{\lambda} \dddot{z}_2 - \frac{k}{m\gamma\lambda} \ddot{z}_1 \ddot{z}_3^2 + \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \ddot{z}_2^2 - \frac{k}{m\gamma} \ddot{z}_1 \dddot{z}_3
\]

\[
\dot{V} = -k\beta \ddot{z}_1^2 + 2k\ddot{z}_1 \dddot{z}_2 - \left(k_v + \frac{k}{\beta}\right) \dddot{z}_2^2 + \frac{k}{m\gamma\lambda} \ddot{z}_1 \dddot{z}_3 - \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \dddot{z}_3 \dddot{z}_2 + \\
+ \frac{k}{\lambda} \dddot{z}_2 - \frac{k}{m\gamma\lambda} \ddot{z}_1 \ddot{z}_3^2 + \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \ddot{z}_2^2 - \frac{k}{m\gamma} \ddot{z}_1 \dddot{z}_3
\]

\[
\dot{V} = -k\beta \ddot{z}_1^2 + 2k\ddot{z}_1 \dddot{z}_2 - \left(k_v + \frac{k}{\beta}\right) \dddot{z}_2^2 + \frac{k}{m\gamma\lambda} \ddot{z}_1 \dddot{z}_3 - \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \dddot{z}_3 \dddot{z}_2 + \\
+ \frac{k}{\lambda} \dddot{z}_2 - \frac{k}{m\gamma\lambda} \ddot{z}_1 \ddot{z}_3^2 + \left(\frac{\eta}{m\gamma} + \frac{k}{\lambda}\right) \ddot{z}_2^2 - \frac{k}{m\gamma} \ddot{z}_1 \dddot{z}_3
\]

In a matrix form:

\[
\dot{V} = -\dddot{z}^T \begin{bmatrix} k\beta & -k & 0 \\ -k & \eta & 0 \\ 0 & 0 & \frac{k}{m\gamma\lambda} \end{bmatrix} \dddot{z}
\] (B-61)

The matrix is negative definite if an only if:

1. \(k\beta > 0 \Rightarrow \beta > 0\)
2. \(\Delta_1 = \beta \left(k_v + \frac{k}{\beta} - \frac{k}{\lambda}\right) - k^2 > 0 \Rightarrow \lambda > 0\)
3. \(\frac{k}{m\gamma\lambda} > 0 \Rightarrow k_v > \frac{k}{\lambda}\)

Since \((m, \beta, \gamma) > 0\). Moreover, from the calculus of the contact phase (not showed), a lower bound can be derived for \(\beta\):

\[
\beta > \max \left\{ \left(\frac{k}{m\gamma\lambda}\right)^2 \frac{\eta' - \frac{3k^2}{4}}{k \left(k_v - \frac{k}{\lambda}\right)}, 1 \right\}
\] (B-62)

Where \(\eta' = \left(k_v + k - \frac{k}{\lambda}\right)\)
Bibliography


Master of Science Thesis

Matteo Ciocca


Master of Science Thesis  Matteo Ciocca


Matteo Ciocca Master of Science Thesis


