POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering

Impedance shaping and Model Predictive Sliding Mode Control for stable bilateral robot teleoperation

Supervisor: Prof. Paolo Rocco
Co-supervisor: Davide Nicolis

Master Thesis dissertation of:

Fabio Allevi ID 850142

Academic year 2016-2017

Ai miei nonni

A Martina

Ringraziamenti

Giunto alla conclusione di questa tesi e di questo percorso accademico, mi resta poco tempo (come al solito) da *dedicare* all'ultima pagina del mio elaborato. In essa vorrei riassumere la mia gratitudine verso tutte le persone che mi hanno accompagnato e sostenuto in questi anni, senza le quali tutto ciò che ho realizzato e sto realizzando avrebbe un'importanza marginale. Innanzitutto, vorrei ringraziare mio padre e mia madre, ai quali dedico tutto l'impegno e la fatica di questi anni universitari. Senza il loro aiuto, sostegno e guida non sarebbe stato possibile raggiungere nessuno dei traguardi che mi sono posto. Un sentito ringraziamento va al prof. Paolo Rocco, per avermi permesso di prendere parte alle attività del laboratorio Merlin e ad avermi concesso l'utilizzo di apparecchiature all'avanguardia per poter realizzare la parte sperimentale della mia tesi. Un *grazie* va anche a Davide Nicolis, che mi ha aiutato sia nell'approfondimento dei concetti legati alla teleoperazione sia nella successiva realizzazione sperimentale. *Grazie* ai miei amici di università, con i quali ho condiviso cinque/sei bellissimi anni e tanti (tantissimi) esami. *Grazie* a Davide Bazzi, Dario Savaresi, Stefano Dattilo, Giulio dell'Oro, Sebastian Guzman e al buon Liuk, che mi ha dato una mano (anzi un braccio) nella realizzazione di molti degli argomenti di questa tesi. *Grazie* a tutti i miei famigliari, ai miei nonni, zii e cugini, per il sostegno che mi hanno dato. *Grazie* ai miei amici Alberto, Claudio, Barbara, Andrea, Laura, Benedetta, Luca, Federico, Erika e a tutti gli altri, per la pazienza che hanno avuto nel sopportarmi tutti questi anni. *Grazie* alla mia squadra di calcio, ai miei allenatori e a tutte quelle persone che hanno creduto in me e in quello che ho fatto, faccio e farò.

Contents

Abstract **1**

1 Introduction **5**

1.1 Historic review 6

1.2 Motivation and objective 8

1.3 Contributions 8

1.4 Thesis outline 9

2 Teleoperation system analysis **13**

2.1 Teleoperation concept scheme 13

2.1.1 Nomenclature 14

2.1.2 Performance indicators 14

2.2 Teleoperation constitutive elements 16

2.2.1 Master device 17

2.2.2 Slave device 18

2.2.3 Communication channel 19

2.2.4 Environment 20

2.2.5 Operator 22

2.3 Control scheme architectures 24

2.3.1 Generalized control scheme 25

2.3.2 2 - channel architectures 26

2.3.3 3 - channel architectures 28

2.3.4 4 - channel architectures 29

2.3.5 Control architecture choice 29

3 Local controller 33

3.1 Inverse dynamics control . . . 33

3.2 Impedance control . . . 35

3.2.1 Master device . . . 37

3.2.2 Slave device . . . 38

3.2.3 Stability limits . . . 41

3.3 Sliding mode control . . . 43

3.3.1 Sliding mode design procedure . . . 46

3.3.2 Sliding mode master device . . . 48

3.3.3 Sliding mode slave device . . . 52

3.3.4 Stability analysis . . . 53

3.3.5 Sliding mode drawbacks . . . 58

3.4 SMC continuous approximations . . . 61

3.5 Higher order sliding mode control . . . 62

3.5.1 Second order SMC . . . 65

3.6 Simulation comparison . . . 69

3.7 Integral sliding mode . . . 75

3.8 Sliding mode MPC . . . 80

4 Global scheme analysis 85

4.1 Teleoperation matrices representations . . . 85

4.2 Stability analysis . . . 89

4.2.1 Llewellyn's analysis . . . 91

4.2.2 Llewellyn's curve sensitivity . . . 99

4.2.3 Llewellyn's curve optimization . . . 102

4.3 Transparency analysis . . . 106

4.4 Passivity observer . . . 110

4.4.1 Energy storage saturation . . . 114

4.4.2 Local oscillations . . . 115

4.4.3 Time delay implementation . . . 117

4.5 Variable impedance controller . . . 120

4.5.1 Continuous adaptation . . . 121
4.5.2 Two - state adaptation . . . 123

5 N - DOFs extension 127
5.1 Decentralized SMC . . . 128
5.2 Operational space SMC . . . 130
5.2.1 Centralized sliding surface . . . 132
5.2.2 Centralized equivalent control . . . 135
5.3 Redundant manipulator . . . 137
5.3.1 MPC equivalent control . . . 137
5.3.2 Null space sliding surface . . . 139
5.4 2 - DOF simulation . . . 141
5.4.1 Operational space SMC without redundancy . . . 144
5.4.2 Operational space SMC with redundancy . . . 145
5.5 7 - DOF simulation . . . 149
5.6 Predictive SMC . . . 154

6 Experimental results 159
6.1 Experimental set-up . . . 159
6.2 External force estimator . . . 161
6.3 External controller . . . 163
6.3.1 FIR identification . . . 165
6.3.2 Friction compensation . . . 169
6.4 Impedance tracking . . . 172
6.4.1 Step reference . . . 172
6.4.2 Interpolated TVP reference . . . 181
6.5 Teleoperation experiment . . . 190

7 Conclusions 197
7.1 Future developments . . . 198

A Differentiation procedures 201
A.1 Introduction . . . 201

A.2 Levant differentiator 202
A.3 Higher order differentiators 203
A.4 Algorithm testing results 204

B Friction identification **211**

B.1 Friction - velocity map identification 211
B.2 Friction model choice 213
B.3 Joint elasticity 216
B.4 Parameter identification 218
B.5 Implementation and validation 221

Bibliography **229**

List of Figures

1.1 Patented slave - master bilateral teleoperator by Görtz 6

1.2 Auxiliary system trajectory for STA sliding mode control 10

1.3 Three dimensional representation of the robotic arm used for 7 DOFs simulation 11

1.4 Network scheme of a generic bilateral teleoperation system 12

1.5 Frequency spectra for the filtered joints input torque 12

2.1 General bilateral teleoperation scheme (distributed representation) . . 14

2.2 Maximum available performance curve for a teleoperation system . . 16

2.3 Transmitted impedance 16

2.4 Simplified 1 Degree of Freedom (DOF) modelling for slave and master devices 18

2.5 Two possible operator models: fixed impedance with imposed force reference, McRuer crossover with speed tracking configuration 24

2.6 Block diagram representation for a generic bilateral teleoperation control scheme 25

3.1 Inverse dynamics control scheme block diagram 35

3.2 Impedance control scheme block diagram 36

3.3 Impedance - impedance controllers signals flow through teleoperation architecture 40

3.4 General block diagram for a variable structure control system 43

3.5 Standard phase plane evolution for an Sliding Mode Controller (SMC) controlled process 45

3.6 1 DOF uncertain mechanical system affected by uncertainty: with and without SMC control . . . 46
3.7 Representation of the sliding manifold in master device phase space diagram . . . 50
3.8 Convergence of the expanded state trajectory onto the sliding plane . 56
3.9 Three examples of asymptotically stable control systems . . . 57
3.10 Control signal for a standard SMC controller. We can notice in the left plot high frequency oscillation due to chattering . . . 58
3.11 Detail of system position and velocity trajectories when chattering on the control signal occurs . . . 59
3.12 Uncontrolled reaching phase can generate very different initial transient response, although once sliding is enforced controlled system behaves as the desired one . . . 60
3.13 Graphical representation of the two most common continuous approximations for the signum function, plotted with various δ . . . 62
3.14 Chattering removal effect of quasi - sliding control techniques compared to standard SMC . . . 63
3.15 Numerical differentiator implemented in Suboptimal Second Order Sliding Mode Controller (SSOSMC) to find the extremal points of σ . 68
3.16 Auxiliary system trajectories for two different higher order sliding mode algorithms in the σ, $\dot{\sigma}$ plane. Changing contraction criteria generate peculiar evolutions . . . 70
3.17 Comparison between the various type of chattering - avoidance methods while performing an impedance tracking of a system affected by uncertainty. Top: position, bottom: error . . . 72
3.18 Comparison between the various type of chattering - avoidance methods while performing an impedance tracking of a system affected by uncertainty - control force and sliding variable . . . 73
3.19 Detail of control effort for the three tested algorithms after transient phase . . . 74

3.20 Transient response of simplified master device with initial position x =1 m, with and without Integral Sliding Mode Controller (ISMC) . 78

3.21 Phase portraits of standard SMC and ISMC. As we can notice, reaching phase is absent for the second one 79

3.22 Sliding Mode Model Predictive Controller (SMPC) block diagram, where equivalent control information is fed to the SMC in order to avoid constraint violation 81

3.23 Improvement of adaptive reaching function with respect to the fixed nominal one - sliding variable 82

3.24 Improvement of adaptive reaching function with respect to the fixed nominal one. Top: control action, bottom: position response 84

4.1 Network representation for a generic bilateral teleoperation scheme . 87

4.2 Llewellyn's curve for a generic set of impedance parameters and a fixed time delay 93

4.3 Bilateral teleoperation network when the communication channel is not affected by time delay 94

4.4 Llewellyn's curve and stability limit, with underlined early crossing (red dot) and high frequency crossing (blue dot) 96

4.5 Necessary condition to avoid early crossing of Llewellyn's curve 97

4.6 Lower envelope of the Llewellyn's curve compared with the real one . 98

4.7 Sensitivity analysis with respect to $\bar{b}_m$ variations 100

4.8 Sensitivity analysis with respect to $\bar{m}_s$ variations 101

4.9 Sensitivity analysis with respect to $\bar{k}_s$ variations 103

4.10 Sensitivity analysis with respect to $\bar{b}_s$ variations 104

4.11 Llewellyn's curve for the ideally transparent hybrid matrix 107

4.12 Evaluation of the transmitted impedance frequency response for different parameters tuning 109

4.13 Evaluation of the transmitted force scaling frequency response for different parameters tuning 111

4.14 2 port network energy flows scheme 112

4.15 Energy flows monitored by the passivity observer during stable teleoperation task . . . 113
4.16 Saturation of the observed energy inside the passivity observer 114
4.17 Observed energy at the master and slave sides with and without the conservative energy contributions . . . 118
4.18 Split of energy flows at the master and slave ports . . . 119
4.19 Variable impedance control scheme architecture . . . 121
4.20 Continuously adaptive impedance control: slave and master position profiles and Llewellyn's curve for two different operative conditions . 122
4.21 Two state adaptation control logic for the variable impedance slave controller . . . 123
4.22 Contact force comparison between different slave impedance tunings: dashed lines represent wall impacts . . . 124
4.23 2 - state adaptive impedance control: slave and master position profiles and Llewellyn's curve for two different operative conditions . . . 125

5.1 Control scheme of a decentralized SMC controller for multi - DOFs manipulators . . . 129
5.2 Centralized control structure for the operational space impedance controller . . . 132
5.3 Mapping between the manipulator joint space and the task space for a redundant manipulator . . . 139
5.4 Graphic visualization of the 2 DOFs planar manipulator implemented 143
5.5 Task space end - effector dynamics with and without operational space sliding surface . . . 146
5.6 Sliding surface and control signal dynamics for operational space SMC 147
5.7 Constraint fulfilment of the free y coordinate with and without $\boldsymbol{\sigma_0}$ . . 150
5.8 Three dimensional representation of the robotic arm used for 7 DOFs simulation in its initial pose $\boldsymbol{q_0}$ . . . 151
5.9 Joint position response with and without decentralised SMC 152
5.10 Torque control effort for decentralised SMC . . . 153

5.11 Sliding variable evolution in the σ - $\dot{\sigma}$ plane. A limit cycle is established around the origin due to a control action delay of 12 ms 155

5.12 Standard sliding surface and its 2-step ahead prediction 157

5.13 predictive SMC compared to standard one applied on a system affected by control variable delay 158

6.1 Dual arm robot adopted for the experiments 160

6.2 Block diagram representation of the `YuMi` control scheme. The dashed box indicates the internal proprietary control unit 164

6.3 Feed - forward torque signal vs. actual torque applied to the joint motor 166

6.4 Identification of the first joint Finite Impulse Response (FIR) transfer function from experimental input - output response 168

6.5 Frequency spectra for the seven joints input torque during sliding mode motion. Allowed control bandwidth is significantly smaller than the Nyquist frequency 125 Hz 169

6.6 Torque control signal applied to joint 3 and the amount of effort required to overcome friction 170

6.7 Experimentally computed friction for joint 3, tested for different robot configurations. The identified simplified model is shown with the black line 173

6.8 When friction at low speed is very high, integral action can cause stick - slip phenomena to occur 176

6.9 1^{st}, 2^{nd}, 3^{rd} joint positions for the step reference impedance tracking experiment 177

6.10 4^{th}, 5^{th}, 6^{th} joint positions for the step reference impedance tracking experiment 178

6.11 Joint position of the 7^{th} joint for the step reference impedance tracking experiment 179

6.12 Feed - forward torque applied to the first four joints by (a) sigmoid SMC controller (b) Super Twisting Algorithm controller 179

6.13 Sliding surfaces for the first four joints of the step reference impedance tracking experiment 180
6.14 Impedance tracking with Trapezoidal Velocity Profile (TVP) profile for the first joint 183
6.15 Impedance tracking with TVP profile for the second joint 184
6.16 Impedance tracking with TVP profile for the third joint 185
6.17 Impedance tracking with TVP profile for the fourth joint 186
6.18 Impedance tracking with TVP profile for the fifth joint 187
6.19 Impedance tracking with TVP profile for the sixth joint 188
6.20 Impedance tracking with TVP profile for the seventh joint 189
6.21 Conceptual interaction between the master and slave manipulators in order to perform the teleoperation tracking task 190
6.22 Bilateral motion, joint positions: first, second, third 192
6.23 Bilateral motion, joint positions: fourth, fifth, sixth 193
6.24 Bilateral motion, tracking error: first, second, third 194
6.25 Bilateral motion, tracking error: fourth, fifth, sixth 195

A.1 Time evolution of input signal (A.7) and its theoretical derivative (A.8)205
A.2 Bode diagrams of $H_{HF}(z)$ and $H_{LF}(z)$ compared to the ideal differentiator frequency response 206
A.3 Simulation comparison between linear and non-linear algorithms without input noise 208
A.4 Simulation comparison between linear and non-linear algorithms with band-limited white noise on $f(t)$ 209

B.1 Reference torque profile and measured joint speed during friction identification experiment 212
B.2 Experimental friction curve obtained with the procedure described in section B.1 213
B.3 Static friction models that match the properties required by the experimental data 215

B.4 LuGre model estimated friction evolution. Difference between increasing and decreasing speeds is clearly visible 216

B.5 Friction experiment performed to analyse joint elasticity 217

B.6 Graphical representation of the Stribeck curve with added torsional spring, as shown in Equation (B.4) 218

B.7 Identification of viscous coefficients and Coulomb frictions 220

B.8 Identification of stiction torque τ_s and inverse Stribeck speed α. Red dots shows the stiction for each cycle, while red area represents the inverse response phase 221

B.9 Identification of joint elastic stiffness and comparison with the experimental result 222

B.10 Effect of the linear approximation on the original friction curve 223

B.11 Comparison between estimated and experimental friction values . . . 224

B.12 Random spline trajectory used for friction model validation 225

B.13 Comparison between validation friction curve and estimated one . . . 226

B.14 Friction estimation error analysis 227

List of Tables

2.1 Nomenclature for the generic control scheme architecture 26

2.2 Overview table of the main teleoperation architecture's advantages - disadvantages 31

3.1 Master device parameters used for the simulation comparison of chattering - avoidance methods 69

3.2 Root Mean Square (RMS) errors for the step tracking performance of the tested sliding mode control algorithm 71

3.3 Variability of control effort after transient phase 74

5.1 Nominal and estimated parameters for the 2 DOFs planar manipulator 143

6.1 Identified coefficient for the simplified friction expression 172

6.2 Tuning parameters chosen for the sigmoid SMC approximation and the Super Twisting Algorithm in the step experiment 174

6.3 RMS error associated to impedance control, sigmoid SMC control and Super Twisting Algorithm for step response trajectories 175

6.4 Tuning parameters chosen for the impedance controller and the sigmoid SMC approximation in the TVP trajectory experiment 181

6.5 RMS error associated to impedance and sigmoid SMC control for TVP 182

A.1 Tuning parameters for the two implemented Levant differentiators . . 207

A.2 Comparison of RMS errors for the four tested differentiators, with and without noise 210

B.1 Experimental identified parameter values 221

List of Symbols

C	MPC cost function
D	Discrete time delay
E^m	Energy flow at the master port
E^s	Energy flow at the slave port
$F(j\omega)$	FIR experimental transfer function
I	Desired impedance profile
K	Sliding mode discontinuous term gain
K_t	Force feedback fidelity transfer function
$S_{xx}(j\omega)$	Input power spectrum
$S_{xy}(j\omega)$	Cross power spectrum
T	Sampling time
$U_{\max}$	Auxiliary variable control amplitude for SSOSMC
$V(\sigma)$	Lyapunov function
Z_e	Environmental impedance
Z_h	Human operator impedance
Z_m	Master mechanical impedance
Z_s	Slave mechanical impedance
Z_t	Transmitted impedance
Γ	Sliding surface external signals (master side)
$\Lambda(\omega)$	Llewellyn's curve
$\Lambda_{\min}(\omega)$	Llewellyn's curve lower envelope

Ψ	Sliding surface external signals (slave side)
Σ	Integral sliding variable
α	Contraction constant for SSOSMC
$\boldsymbol{B(q)}$	Inertia matrix
$\boldsymbol{C(\dot{q},q)}$	Coriolis and centrifugal matrix
$\boldsymbol{F_v(q)}$	Estimated friction vector
$\boldsymbol{F}$	Sliding surface coefficients vector (slave side)
$\boldsymbol{G}$	Sliding surface coefficients vector (master side)
$\boldsymbol{H}$	Hybrid matrix
$\boldsymbol{I_\tau}$	Joint projection of the operational impedance profile vector
$\boldsymbol{I}$	Operational space impedance profile vector
$\boldsymbol{J(q)}$	Jacobian matrix
$\boldsymbol{P}$	Null space sliding surface projector
$\boldsymbol{Q}$	Weight matrix for cost function optimization procedures
$\boldsymbol{S}$	Integral of the operational space impedance profile vector
$\boldsymbol{T(q)}$	Forward kinematic expression
$\boldsymbol{Z}$	Impedance matrix
$\boldsymbol{\sigma_0}$	Null space sliding surface
$\boldsymbol{\sigma_{op}}$	Operational space sliding surface
$\boldsymbol{\sigma}$	Decentralised sliding manifolds
$\boldsymbol{\tau_m}$	Master device control torque
$\boldsymbol{\tau_s}$	Slave device control torque
$\boldsymbol{\tau_{ext}}$	External torques contribute
$\boldsymbol{\tau_{fw}}$	Feed - forward torque signal
$\boldsymbol{e}$	Tracking error vector
$\boldsymbol{g(q)}$	Gravitational vector
$\boldsymbol{h_e}$	Forces and moments applied by the environment on the manipulator
$\boldsymbol{h_h}$	Forces and moments applied by the human on the manipulator

$\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})$	Compressed expression of Coriolis, cnetrifugal and gravitational terms
$\boldsymbol{p}$	Generalized momentum vector
$\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}$	Position, speed and acceleration of joint variables vector
$\boldsymbol{r}$	Residual vector
$\boldsymbol{v}^{\boldsymbol{eq}}$	Decentralised equivalent control laws
$\boldsymbol{w}$	Reference input vector
$\boldsymbol{x}$	Expanded state vector (master side)
$\boldsymbol{z}$	Expanded state vector (slave side)
χ	Grönwall's lemma coefficients
δ	Approximation degree for continuous sliding surface approximations
η	Additive uncertainty
$\hat{\sigma}(k+2\|k)$	2 - steps sliding surface prediction
$\lambda(t)$	Reaching function
μ	Robustness degree parameter for μ - reachability
ω_1	Lower bound for the stability region
ω_2	Upper bound for the stability region
ρ	Sliding surface sliding order
σ	Sliding variable
ξ	Extended μ - reachability parameter for higher order sliding mode
ζ	Multiplicative uncertainty
b_e	Environmental damping
d	Continuous time delay
f_e, F_e	Force exerted by the environment on the teleoperation system
f_h, F_h	Force exerted by the operaton on the teleoperation system
f_m, F_m	Control force of the master device
f_s, F_s	Control force of the slave device
$h(\boldsymbol{v})$	Cost function which minimizes the control effort amplitude
k	Super Twisting Algorithm tuning parameters

k_e	Environmental stiffness
k_f	Force scaling factor
k_p	Kinematic scaling factor
r	System relative degree
$s(\sigma)$	Continuous approximation of the signum function
t_0	Initial time instant
t_f	Sliding mode starting time
v	Acceleration control input for the inverse dynamics controller
v^{eq}	Equivalent control law
v^{smc}	Discontinuous control law
v_e, V_e	Environmental speed for the unidirectional interaction model
v_h, V_h	Operator speed for the 1 DOF model
v_m, V_m	Master speed for the 1 DOF model
v_s, V_s	Slave speed for the 1 DOF model
w	Auxiliary system control input
x_e	Wall position
z	Auxiliary system state variable

List of Abbreviations

ARMA Auto-Regressive Moving Average

DOF Degree of Freedom

DTFT Discrete Time Fourier Transform

FIR Finite Impulse Response

HOSMC Higher Order Sliding Mode Controller

ISMC Integral Sliding Mode Controller

MIMO Multiple Input Multiple Output

MPC Model Predictive Control

PD Proportional Derivative

PID Proportional Integral Derivative

PN Petri Net

PSD Power Spectral Density

RHP Right - Half Plane

RMS Root Mean Square

RTT Round Trip Time

SISO Single Input Single Output

SMC Sliding Mode Controller

SMO Sliding Mode Observer

SMPC Sliding Mode Model Predictive Controller

SOSMC Second Order Sliding Mode Controller

SSOSMC Suboptimal Second Order Sliding Mode Controller

STA Super Twisting Algorithm

TVP Trapezoidal Velocity Profile

VSCS Variable Structure Control System

Abstract

Nowadays, automation techniques are more oriented towards a complete autonomy of supervision and control systems in order to take the human out of the loop. This phenomenon is spreading from the consumer area (autonomous vehicles, Ambient Assisted Living) to the industrial one (machine learning, industry 4.0). Nevertheless, recent studies have shown that the cooperation between humans and machines has a better adaptability and greater productivity with respect to fully autonomous solutions. Moreover, this choice is preferable when production processes involve the interaction with dangerous or expensive substances. To better combine the peculiarities of the operator and the machine, they need to share as much information as possible. For this reason, robot bilateral teleoperation has been considered in this thesis. In this framework, the user is able to control a slave manipulator at a remote location from a master station, while at the same time being able to perceive the robot external environment thanks to accurately designed feedback forces. These networked systems are notoriously affected by stability and transparency issues. To solve them, in this thesis a model predictive nonlinear sliding mode control technique has been developed and generalised to multi DOFs teleoperation systems. Furthermore, the proposed controller is able to guarantee stable environment interaction while retaining good telepresence (even if affected by delay) thanks to impedance shaping techniques. Simulation results prove the validity of the approach, improving the robustness of the system with respect to external disturbances and uncertainties. The formulated theory has been applied to an anthropomorphic manipulator, underlining the main simulation limits as well as showing new possible improvements of the proposed theory.

Sommario

Oggigiorno, le tecniche di automazione sono sempre più orientate verso la completa autonomia dei sistemi di supervisione e di controllo, così da poter escludere il più possibile l'intervento dell'uomo. Questo fenomeno si sta diffondendo sia nell'ambito "consumer" (veicoli autonomi, domotica) che in quello industriale (machine learning, industria 4.0). Ciononostante, recenti studi hanno mostrato che la cooperazione tra uomo e macchina possiede più capacità di adattamento e miglior produttività rispetto ad una soluzione totalmente autonoma. Inoltre, questa scelta è preferita nel caso in cui i processi produttivi prevedano la manipolazione di sostanze pericolose o costose. Per poter conciliare al meglio le peculiarità di operatore e macchina, essi devono condividere più informazioni possibili. Per questa ragione, nella presente tesi è stato analizzato un sistema di teleoperazione bilaterale. Grazie ad esso, l'utente è in grado di controllare, dalla propria stazione di comando, un manipolatore remoto e allo stesso tempo di percepire l'ambiente esterno del robot grazie alla presenza di un feedback di forza adeguato. A tali sistemi distribuiti sono notoriamente associati problemi di stabilità e trasparenza. Per poterli risolvere, in questa tesi sono state sviluppate tecniche di controllo non lineare di tipo model predictive sliding mode, generalizzandone l'analisi a sistemi di teleoperazione a più gradi di libertà. Inoltre, il controllore proposto è in grado di garantire la stabilità al contatto con l'ambiente pur mantenendo una buona tele - presenza, persino se affetto da ritardo. Ciò è ottenuto grazie a modifiche online delle relazioni d'impedenza. I risultati ottenuti in simulazione dimostrano la bontà dell'approccio scelto, migliorando la robustezza del sistema nei confronti di disturbi esterni e incertezze. La teoria formulata è stata applicata ad un manipolatore antropomorfo, evidenziando i limiti della simulazione e mostrando nuovi possibili sviluppi della teoria proposta.

Chapter 1

Introduction

According to the dictionary definition, *teleoperation* is the capability of controlling a device, or machine, remotely. Indeed, it is composed by the ancient Greek prefix $\tau\tilde{\eta}\lambda\varepsilon$- (tèle-), which literally means “far from” and it is usually referred to tasks where information is transmitted over a significant distance. If the machine remotely located is constituted by a robotic manipulator, it is possible to talk about *telerobotics*. The essence of these systems is given by the remote interconnection between two separate devices, which are the operator haptic interface (also known as *master device*) and the remote environment manipulator (also known as *slave device*). Therefore, teleoperation systems are realised to extend human capability to interact with external environments without the need of being in the same location [38]. To realise such interconnection, it is necessary that master and slave devices share information on their respective environments. This is done in order to provide the operator with similar conditions as the ones he/she would experience by interacting directly with the remote location. With a strong and coherent connection, the operator is able to perform the remote task at his/her best, which is the ultimate goal for a reliable teleoperation system. Several external factors (communication delay, uncertain remote environment, etc.) can degrade the performance of teleoperation systems. Thus, several strategies have been studied to deal with these problems.

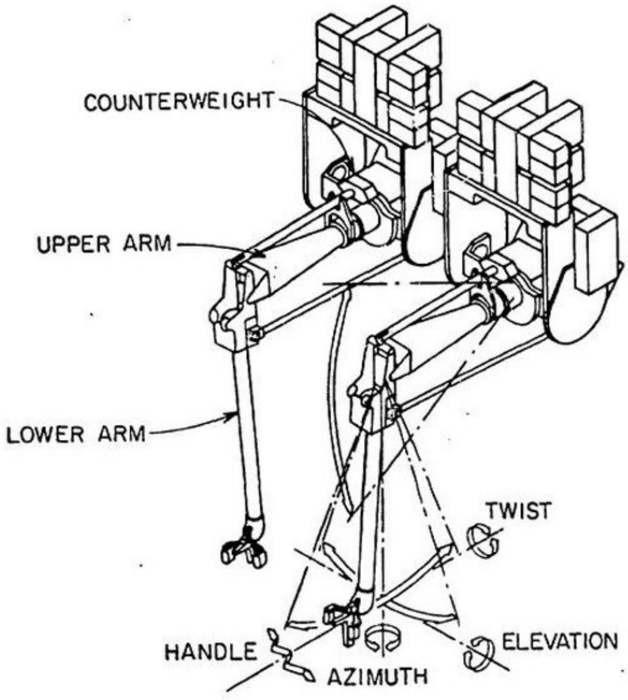


Figure 1.1: Patented slave - master bilateral teleoperator by Görtz

1.1 Historic review

The concept of teleoperation as it is intended nowadays was born together with the industrial nuclear field, where there was the need of handling radioactive materials without exposing human operators. One of the pioneers of this new research topic was Raymond Görtz [50], who developed master - slave systems which were firstly interconnected via mechanical links (end of '40s) and then via electrical signals (1954). His prototype is visible in Figure 1.1. Then, evolution of new application fields (space and underwater exploration, microsurgery, etc.) pushed research towards more precise and reliable teleoperation systems.

Master devices started to differentiate from slave devices and new haptic interfaces were born in order to improve the operator performance as much as possible [75]. Together with these technical improvements, new problems arose. Delay in the communication channel was firstly addressed by Sheridan [71], who observed significant performance degradation as well as instability of the control devices. This was highlighted by the necessity of returning feedback information from the slave to the master device (*bilateral teleoperation*). Another source of instability is generated when the slave device enters in contact with hard surfaces [31]. According to [38], to deal with these problems, three types of teleoperation strategies have been proposed:

- **Move and wait**: first intuitive approach implemented when significant delay affects the communication channel. Operator performs the desired task with the master and waits until the slave has performed it before continuing the task execution. Although intrinsically stable, this approach has extremely low performance levels;
- **Supervisory control**: with the introduction of the first calculators, some teleoperation activities started to be scheduled and the operator acted as a supervisory entity, who defined the predetermined operation that the slave should execute [84]. Obviously this technique is applicable only when the interaction is extremely predictable;
- **Predictive control**: this strategy makes use of virtual reality devices, which feed the operator with an undelayed version of the slave environment. In this way the operator interacts with a prediction of the remote environment. Also this approach requires a good knowledge of the slave site;

In [27, 28], Valdovinos et al. developed a higher order sliding mode control theory for bilateral teleoperation systems. Their focus, however, was on 1 DOF networks only and their stability analysis performed off - line. Park and Cho [10, 59] proposed a similar approach based on a different sliding surface, but also their experimental set - up was based on 1 DOF systems. A different approach has been pursued by Spong et al. [45] and it consists in an on - line energetic analysis of the network, in order to guarantee its passivity through proper damping injection. Although effective, this approach significantly deteriorates teleoperation performance (by introducing conservatism). Analogous reasonings have been performed by Hannaford et al. [68, 32] and Ortega et al. [55]. The work of Salcudean at al. [35] is inspired by the results obtained by Lawrence about 4 - channel optimization of transparency and stability. Although it guarantees the optimal trade - off between these two indicators, tuning of controllers parameters is complicated and dependent on the human - environment behaviour.

1.2 Motivation and objective

The main objective of this thesis is to analyse the stability and transparency properties of a bilateral teleoperation system, in particular when the remote device enters in contact with unknown environmental conditions. To perform this task, we impose certain impedance relationships to the master and slave device, which are proved to realise stable teleoperation in spite of human - environment behaviour. To robustly impose such dynamics, it has been chosen to adopt sliding mode control technique, which are able to reject unmodelled dynamics that could interfere with the desired one and lead to instability.

This objective is motivated by the lack of literature concerning the stability of bilateral teleoperated systems composed by robotic manipulators. Indeed, results treated in this thesis are usually implemented on simplified mechanisms, which have little practical applications. In this dissertation instead, these methodologies have been extended to a significant real world apparatus, in order to lay the groundwork for a more in - depth application of bilateral controllers.

1.3 Contributions

This thesis dissertation offers the following contributions with respect to the existing theory about stable bilateral teleoperation:

- An innovative control scheme for master and slave devices is proposed, based on MPC minimization process joined with robust higher order sliding mode techniques. An intuitive modification of the sliding variable in order to apply Integral Sliding Mode Controller has been proposed as well. A detailed comparison between different chattering reduction techniques and quantitative simulation of their performance is conduced, along with an adaptive integral sliding surface in order to guarantee MPC constraint fulfilment;

- An analysis of the master and slave impedance parameters variation and their effect on the overall system stability is performed, with the definition of the maximum stable bandwidth for a teleoperation system affected by communication time delay and computation of the optimal impedance parameters. Passivity control theory is extended to impedance shaping techniques in order to optimize transparency during free motion while retaining stable contact with the environment. Different impedance adaptation logics are proposed (2 - state switching, continuous parameter variation), with an analysis of their pros and cons;

- Operational space sliding mode theory is extended to multi DOFs redundant robotic systems, with completely assignable end effector impedance in Cartesian space. A validation of the robustness of this approach with respect to model uncertainties and disturbances is carried out through numerical simulation. A robust null motion and constraint fulfilment for multi - DOFs systems (theoretical proofs and simulation validation), and an innovative predictive sliding surface to deal with systems affected by actuator delays are proposed;

- The theoretical results are implemented on an industrial `ABB IRB14000 YuMi` dual arm manipulator. Theoretical results are validated experimentally using standard impedance control as a benchmark, and by realizing a teleoperation system using one of the arm as a master device while the other as a slave device. Free motion teleoperation tracking with the proposed local controllers is tested.

1.4 Thesis outline

This thesis dissertation is organized based on the following structure:

- **Chapter 2 «Teleoperation system analysis»**: the main teleoperation components as well as their interaction are presented. A brief introductory model for each element that composes the standard teleoperation scheme is derived, as well as the reasonings which lead to their choice. Main performance

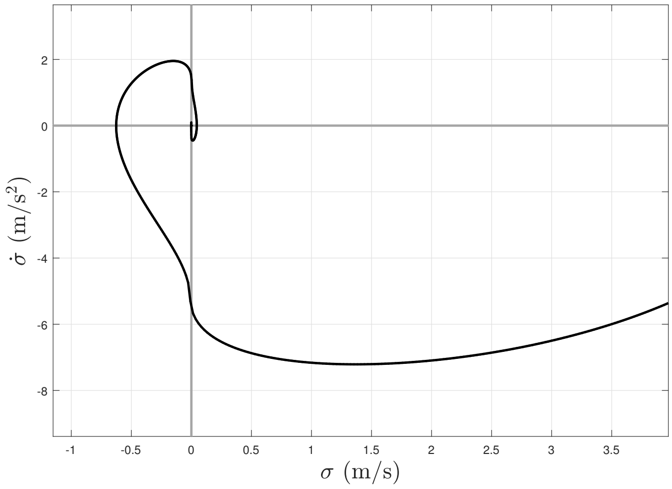


Figure 1.2: Auxiliary system trajectory for STA sliding mode control

indicators are underlined and the trade - off between stability and transparency introduced. Finally, some of the control scheme architectures available in literature are presented and their pros and cons analysed in order to choose the most adequate one for the thesis objective;

- **Chapter 3 «Local controller»**: focus of this chapter is the detailed analysis of the device controllers which are located at the master and slave sides of the teleoperation system. Here, stability issues related to uncertainties and disturbances acting on the manipulators models have been tackled. In particular, nonlinear sliding mode techniques have been developed in order to robustly guarantee desired impedance relationships between the forces acting on the system and the kinematic quantities associated to the device. Main drawbacks related to SMC technique are then presented and the main solutions to them have been reported. A simulation comparison has been adopted in order to choose the best strategy to design master and slave controllers. SMPC is also presented at the end of this section;

- **Chapter 4 «Global controller»**: starting from the results obtained in the previous chapter, the system analysis moves towards the interconnection between the master and slave devices. Here, the main stability issues have been addressed thanks to criteria belonging to network theory. The impedance val-

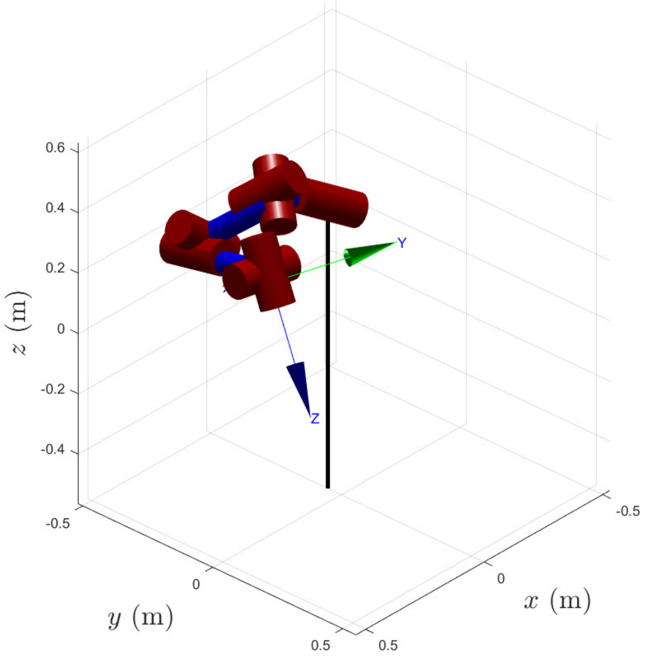


Figure 1.3: Three dimensional representation of the robotic arm used for 7 DOFs simulation

ues for the master and slave devices required to retain stability have been derived, as well as the maximum operative bandwidth in the case of teleoperation systems affected by communication delay. Then, transparency analysis has been performed, deriving also for this case the optimal impedance requirements. The trade - off between these two design principles is shown as well as a possible solution based on on - line impedance shaping;

- **Chapter 5 «N - DOFs extension»**: this chapter generalises the results found in the local controller section in order to apply them to complex multi - DOFs robotic systems. Two possible strategies are presented (decentralised and centralised controllers) and their advantages and disadvantages listed. The concept of operational space impedance is introduced and methods to control it robustly are presented. The discussion has been then extended from standard to redundant manipulators, introducing concepts like null space robust control and optimal constraints fulfilment. These concepts have been tested through simulation on 2 DOFs and 7 DOFs models in order to verify the theoretical

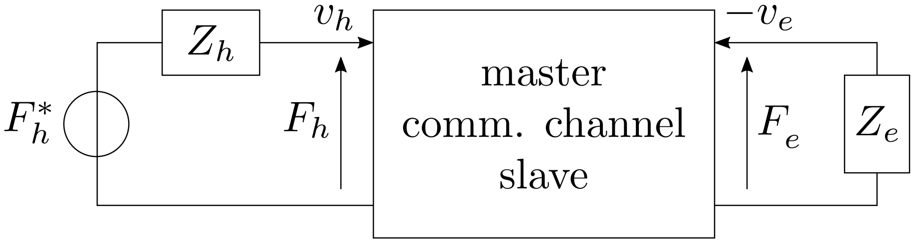


Figure 1.4: Network scheme of a generic bilateral teleoperation system

results. At the end of the section, a way to deal with delays on the actuator channel is also presented;

- **Chapter 6 «Experimental results»**: some of the control techniques presented in the thesis are applied to the experimental apparatus, represented by the `ABB IRB14000 YuMi` robot. Before doing so, a brief description of the available signals and sensors is done. An external torque estimator is derived to provide perception to the robotic arms. Then, the internal proprietary control structure is described and the main operative modes listed. Nonideality of the actuation channel (input filtering) as well as joint friction have been identified and their effects have been compensated. Tracking at the slave joint level is performed, with different reference profiles. Eventually, bilateral tracking motion with the proposed local controllers has been realized and its performance analysed.

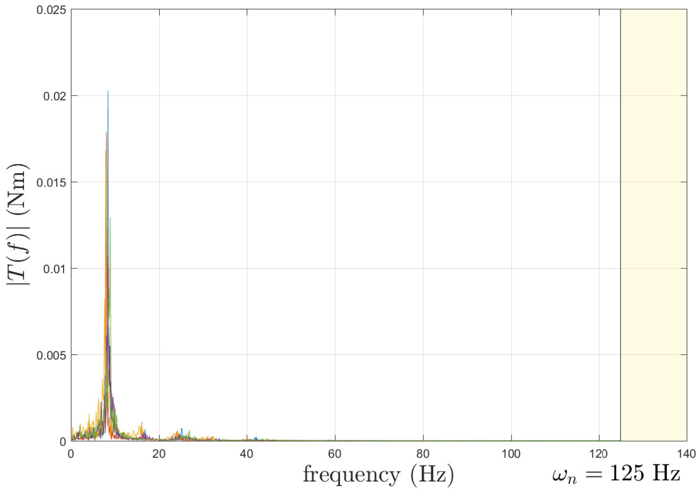


Figure 1.5: Frequency spectra for the filtered joints input torque

Chapter 2

Teleoperation system analysis

This chapter introduces the main theoretical concepts regarding teleoperation in its *more general* sense. In the next section, a generic *teleoperation picture* is introduced. Then, the *primary actors*, which recite their role in the teleoperation drama, will be described. A simplified mathematical models of such entities will be introduced. In the last section, a brief recall on the *main control architectures* will be drawn, as well as their pros and cons.

2.1 Teleoperation concept scheme

Since the main objective of bilateral teleoperation is the remote connection between environments that are physically separated, it seems logical to model the system structure in a *distributed way.* This allows to clearly understand the physical components of the teleoperator structure and the criticalities that such distribution generates. Another advantage of this approach is the possibility of considering the single elements which constitutes the teleoperation scheme separately and focus on the *information flows* that occur between different areas of the overall system.

Thanks to this choice, we can take advantage from concepts brought by network theory in order to put together the skeleton of our teleoperation scheme. In particular, each element can be represented as a 2 - port network, exchanging forces and velocities with its mutual blocks. A schematic representation of this idea is visible at Figure 2.1.

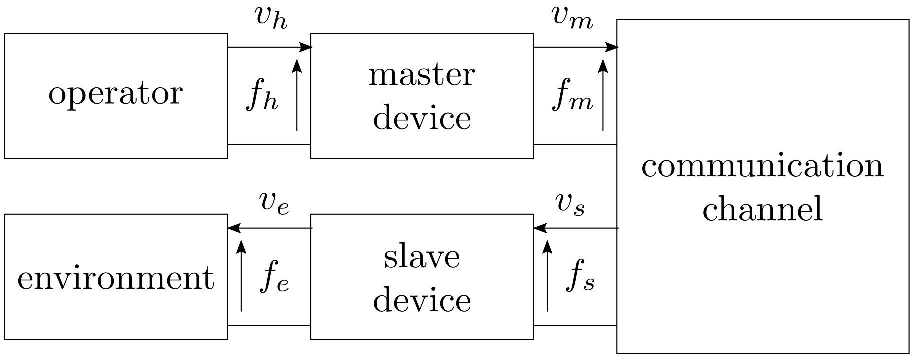


Figure 2.1: General bilateral teleoperation scheme (distributed representation)

2.1.1 Nomenclature

Before considering the main aspects of teleoperation, it's useful to introduce the reader to some terminology.

All the apparatus which belong to the operator environment, such as haptic interfaces, local controllers, etc. will be referred to as located at *master side*. On the other hand, all the devices belonging to the teleoperated environment, such as actuation's plant, manipulated objects, etc. will be located at *slave side*. Since teleoperation devices allow interactions between these two realities, they are equivalently called master-slave controllers.

When the information flow along the communication channel is directed from the master to the slave only, we talk about *unilateral* teleoperation schemes. Instead, when feedback coming both from the master and the slave side are added to the control loop, teleoperation scheme is said to be *bilateral*.

The *mechanical impedance* of a system represents the dynamical relationship between force and velocity exchanged with the external world. According to [83], an *haptic display* is an electromechanical device which is able to properly render impedance characteristics to the operator.

2.1.2 Performance indicators

As we said in the introductory part, the main objective of a teleoperation system is to *perfectly* and *robustly* render the remote environment to the operator, in order to accomplish the assigned tasks in the most natural and efficient way available. This in spite of the uncertainties affecting the system as well as the time delay caused by the communication channel.

All these aspects can be summarized more formally by introducing two main concepts [38]:

- **Stability**: all the variables on both sides have to be bounded and converge to specific reference values, irrespectively of operator - environment behaviour;
- **Transparency**: capability of the teleoperation system to reproduce faithfully the slave environment, as if there were no interfaces between the two remote sites. Such parameter is directly associated with **telepresence**, which represents qualitatively operator's ability to perform tasks in a natural way, without intensive train and adaptively with respect to unknown and variable environment [60];

Thanks to these new definitions, the above objective can be rephrased in this way: an *ideal* master-slave device will be the one which is able to maximize transparency while retaining closed-loop stability.

As we can clearly see from [11, 44, 81], there exists an important trade-off between the impedance range that can be displayed to the operator and the stability of the whole system. If we increase the bandwidth of our controllers in order to render better impedance to the operator, we decrease stability margins, allowing high frequency information to travel around the network. On the other hand, slow system responses may cause haptic interfaces to feel sluggish and provide poor force reflection to the operator. This reasoning can be visually interpreted by looking at Figure 2.2.

A possible measure for transparency can be done evaluating the *transmitted impedance* Z_t, i.e. the impedance of slave environment seen through the master teleoperator port (Figure 2.3). Using this quantity we can compute the dynamic range of impedances that can be shown to the operator, namely the *Z-width* of the teleoperation system [12]

$$Z_{t,width} = \frac{1}{\omega_{max} - \omega_{min}} \int_{\omega_{min}}^{\omega_{max}} || \log Z_{t,max}(j\omega)| - |\log Z_{t,min}(j\omega)|| d\omega \tag{2.1}$$

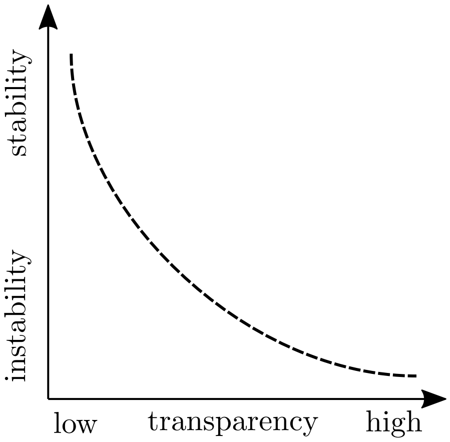


Figure 2.2: Maximum available performance curve for a teleoperation system

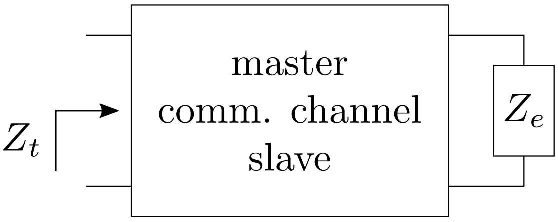


Figure 2.3: Transmitted impedance

where $Z_{t,max}$ and $Z_{t,min}$ are defined as the maximum and minimum transmitted impedances and their values are:

$$Z_{t,max}(j\omega) = Z_t(j\omega)|_{Z_e \to \infty}, \qquad Z_{t,min}(j\omega) = Z_t(j\omega)|_{Z_e \to 0} \tag{2.2}$$

with Z_e equals to the environmental impedance.

According to the definition, teleoperation system A is more transparent than B when $Z^A_{t,width}$ is grater than $Z^B_{t,width}$. A more rigorous and detailed analysis on these performance indicators will be done in chapter 4.

2.2 Teleoperation constitutive elements

As shown in Figure 2.1, constitutive blocks of a teleoperation scheme are sequentially combined in order to realize the information flow between master and slave sites. In order to accommodate such diagram, it's necessary to model them as *dynamic systems* which depend on velocities and forces at their ports. Here we are not making any assumptions on the characteristics of such dynamics (can be non-linear or time-variant). In the next subsections inner behaviour of each 2-port element is described and the operator and environment models are introduced.

2.2.1 Master device

This element needs to model the mechanical interface which is seen by the operator during any teleoperation task. It can be chosen as a generic N DOFs manipulator, with joint positions and velocities available for measure as well as the possibility of estimating the force exerted by the human. To control the robot we can impose a joint torque vector $\boldsymbol{\tau_m}$ in order to:

1. Provide *good haptic feedback* to the operator;
2. Display *proper impedance characteristic* for smooth manipulation;

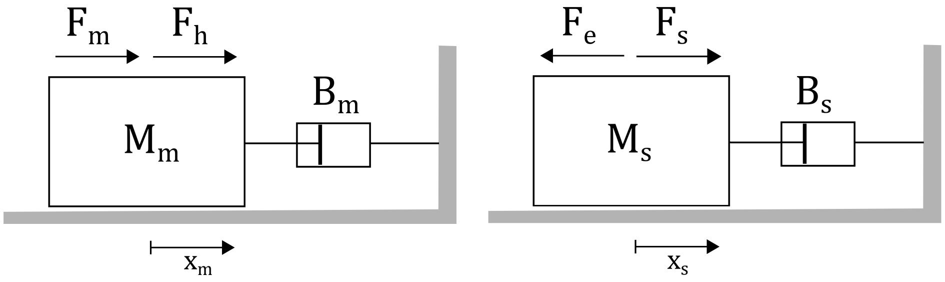
Applying standard Euler-Lagrange procedure [73], the dynamic model for the master system in the joint space is straightforwardly derived

$$\boldsymbol{B_m(q_m)\ddot{q}_m + C_m(q_m, \dot{q}_m)\dot{q}_m + g_m(q_m) = \tau_m + J_m^\top(q_m)h_h} \tag{2.3}$$

where $\boldsymbol{q_m, \dot{q}_m, \ddot{q}_m} \in \mathbb{R}^n$ are, respectively, position, velocity and acceleration vectors, $\boldsymbol{B_m(q_m)} \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C_m(q_m, \dot{q}_m)} \in \mathbb{R}^{n\times n}$ contains all Coriolis' and centrifugal terms, $\boldsymbol{g_m(q_m)} \in \mathbb{R}^n$ represents the gravitational component, $\boldsymbol{\tau_m} \in \mathbb{R}^n$ are the controlled joint torques and $\boldsymbol{J_m^\top(q_m)h_h} \in \mathbb{R}^n$ are the joint-equivalent torques due to a generalized force vector $\boldsymbol{h_h} \in \mathbb{R}^m$ exerted by the human operator at the manipulator's end effector. $\boldsymbol{J_m^\top(q_m)} \in \mathbb{R}^{n\times m}$ represents the Jacobian matrix of master system. Dependency on time of joint variables and torques is here omitted to simplify equation notation.

Equation (2.3), although very general, is rather difficult to analyse when the master device is connected to the network. This is mainly due to the coupled dynamics of each joint and the necessity to export quantities from the operational space (human force, impedance characteristics) to the joint one. That is why it is rather common [27, 10] to consider a *simplified model* for the master device, constituted by a simple 1 DOF mass-damper system, which captures the essential elements to properly describe teleoperation network properties.

$$m_m\ddot{x}_m + b_m\dot{x}_m = f_m + f_h \tag{2.4}$$



(a) *Master device* (b) *Slave device*

Figure 2.4: Simplified 1 DOF modelling for slave and master devices

where the dependency on time ($\ddot{x}_m(t)$, $\dot{x}_m(t)$, $f_m(t)$, $f_h(t)$) is omitted for the sake of readability. f_h represents the force exerted by the human operator on the master system while f_m is the control force acting on the master device. A schematic representation of equation (2.4) is shown at Figure 2.4a.

For network analysis purposes, it is useful to define also the master mechanical impedance Z_m (s is the Laplace operator)

$$Z_m = \frac{F_h + F_m}{V_m} = m_m s + b_m \tag{2.5}$$

where F_h, F_m and V_m are, respectively, the human force, the master control force and the master speed signals expressed in the Laplace domain.

2.2.2 Slave device

An analogous reasoning can be performed on the slave side, where the teleoperated device used to perform the remote task is usually a robotic manipulator. Also in this case, a generic N DOFs dynamic model is introduced, with sensors or estimators that allow access to positions, velocities and end-effector forces.

$$\boldsymbol{B_s}(\boldsymbol{q_s})\ddot{\boldsymbol{q}}_s + \boldsymbol{C_s}(\boldsymbol{q_s}, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{q}}_s + \boldsymbol{g_m}(\boldsymbol{q_s}) = \boldsymbol{\tau_s} - \boldsymbol{J_s}^\top(\boldsymbol{q_s})\boldsymbol{h_e} \tag{2.6}$$

where $\boldsymbol{q_s}, \dot{\boldsymbol{q}}_s, \ddot{\boldsymbol{q}}_s \in \mathbb{R}^l$, $\boldsymbol{B_s}(\boldsymbol{q_s}) \in \mathbb{R}^{l\times l}$, $\boldsymbol{C_s}(\boldsymbol{q_s}, \dot{\boldsymbol{q}}_s) \in \mathbb{R}^{l\times l}$, $\boldsymbol{g_s}(\boldsymbol{q_s}) \in \mathbb{R}^l$, $\boldsymbol{\tau_s} \in \mathbb{R}^l$, $\boldsymbol{J_s}^\top(\boldsymbol{q_s}) \in \mathbb{R}^{l\times k}$ are analogous to master device quantities and $\boldsymbol{h_e} \in \mathbb{R}^k$ is the generalized force vector exerted by the environment on the slave system at the contact point. In this case, the slave control torque $\boldsymbol{\tau_s}$ has to be able to:

1. Provide *stable contact* with the environment;

2. Ensure *good tracking* capabilities with respect to the master position;

Similarly to the master device, for analysis purposes, it is convenient to consider also in this case a simplified 1 DOF model, which allows to understand the main features of the slave system while keeping manageable the overall scheme structure.

$$m_s\ddot{x}_s + b_s\dot{x}_s = f_s - f_e \tag{2.7}$$

f_e represents the force exerted by the environment to the slave system, while f_s is the force applied by the manipulator controller. A schematic representation of equation (2.7) is available at Figure 2.4b. Also in this case it is possible to define the slave mechanical impedance Z_s as

$$Z_s = \frac{F_s - F_e}{V_s} = m_s s + b_s \tag{2.8}$$

where F_s, F_e and V_s are, respectively, the slave control force, the environmental force and the slave speed signals expressed in the Laplace domain.

2.2.3 Communication channel

As reported in [4, 80], one of the main difficulties in describing teleoperation systems is given by the proper modelling of the interconnection between master and slave sides. This because the main objective of teleoperated devices is to realize a *remote connection*, implying that information may need to travel very long distances before reaching targets. This introduces a *time delay* in the channel that can vary from tenth of milliseconds (industrial teleoperation) to seconds (space/deep ocean transmission lines [70]).

Based on these considerations, to keep our next developments as general as possible, we can assume the time delay to be an *unknown quantity*, chosen externally according to physical constraints and network characteristics. Moreover, usual communication channel's latency doesn't keep fixed with time, but varies according to network traffic, adopted protocols, etc. Particularly in the last decades, teleoperation via internet has become a growing field of research and this aspect has become more relevant [57].

Each signal that has to be exchanged between master and slave can be obtained through the equation

$$x_m^d(t) = x_m(t - d(t)) \qquad x_s^d(t) = x_s(t - d(t)) \tag{2.9}$$

where $d(t)$ is the time delay function. To maintain a compact notation, from now on the delay will be indicated with the apex d; if a variable is delayed twice apex becomes dd, and so on.

Since master and slave devices have a priori different structures, it may be useful to include constant *scaling factors* between velocities and forces transmitted through the network $k_p, k_f \in \mathbb{R}^+$. Subscript p indicates kinematic scaling, while f stands for force scaling.

2.2.4 Environment

Concerning the environment faced by the slave manipulator, we need to model the interaction as if it was a component of the closed teleoperation control loop. This is necessary because we want to provide to the operator a force feedback coming from measurements done at the slave contact point in order to realize a bilateral control architecture.

There exists a broad and detailed literature about contact mechanics [58], as well as their effects on teleoperated systems [51, 31]. In accordance with the thesis' objectives, only a brief recall of the most used contact models is here introduced:

- **Hertzian theory** (1882): first contact model, developed by Heinrich Hertz to describe the interaction between two spherical objects. It considers both surfaces to behave in an *elastic* way, thus neglecting any dissipative effect due to the plastic deformation of the object as well as adhesion phenomena;

- **Bradley models** (1936): opposed to Hertz contact concept, it neglects deformations at the interaction point, using only Van der Waals forces to explain attraction and repulsion of bodies;

- **JKR theory** (1970): extension of the hertzian studies to include adhesive contact between objects. It is able to consider also pulling forces during the unload situation;

- **DMT theory** (1975): general theory of contact adhesion of an elastic spherical particle on a flat and rigid surface. It considers both the Van der Waals forces and the JKR model. It is rather complete and complicate, useful to simulate precisely pressure and force distributions at the interaction point;

Coming back to our modelling case, we need to select the most appropriate one to fit our contact interaction. Since the environment belongs to the slave side, we should assume that its properties are *mostly unknown* to the teleoperation system. This means that we cannot choose too sophisticated models, which would require identification procedures to determine their parameters. Moreover, contact environment dynamically evolves based on performed tasks and/or unmodelled entities. Due to these considerations, we can choose a simple *elastic model*, with dissipative element to model material deformation.

Let's consider a *unidirectional interaction* between slave and environment. If the contact depth is limited, it is possible to linearize the elastic behaviour of the surfaces, obtaining the following contact law

$$f_e(x) = \begin{cases} k_e(x - x_e) + b_e\dot{x} & x \geq x_e \\ 0 & x < x_e \end{cases} \tag{2.10}$$

where k_e is the environmental stiffness, while b_e models dissipative forces due to material deformations. x_e represents the wall position and defines the discontinuity between free motion and constrained motion.

This model requires few parameters, but it neglects slave flexibility and nonlinearities. Therefore it is useful for a qualitative analysis of control robustness with respect to very different environmental conditions. Also in this situation, it is possible to express the environment impedance Z_e as

$$Z_e = \frac{F_e}{V_e} = b_e + \frac{k_e}{s} \tag{2.11}$$

where F_e and V_e are, respectively, the force exerted by the environment on the teleoperation system and the environmental speed (which coincides with the slave speed except for the sign).

2.2.5 Operator

The last entity that is left to describe is also the most critical one, since it introduces the human component inside the loop. Due to the fact that the operator is physically connected through a master haptic interface to the teleoperation chain, its impedance can critically affect loop stability. In particular, the human tends to *adapt* his/her behaviour according to the task he/she has to perform and the tactile feedback perceived.

According to literature [21], two different strategies can be pursued to identify a meaningful model for the operator:

- Classical approaches [1, 34] consider human limbs as *static* and *passive* components. The operator is characterized by an ideal force source F_h^{des}, which models the desired effort he/she wants to apply on master device, plus a fixed relation Z_h which models the mechanical impedance of the human hand. Such relation identifies the operator bandwidth, limiting the range of frequencies that he/she can introduce into the system. Since operator and master device are connected through the haptic interface, master speed v_m is equal to the human speed v_h. Thus, Z_h is defined as:

  $$Z_h = \frac{F_h^{des} - F_h}{V_h} = \frac{F_h^{des} - F_h}{V_m} = m_h s + b_h + \frac{k_h}{s} \tag{2.12}$$

  where F_h is the human force effectively applied to the teleoperator system. Parameters m_h, b_h and k_h need to be identified through empirical operator frequency diagrams. Since hand impedance depends on the performed task as well as the operator training, experiments won't generate constant impedance parameters. Therefore they should rely on average results (approximation);

- More sophisticated models [85, 53] have been developed to improve human representation for control purposes. They rely on the operator capability to *adapt* his/her impedance as a consequence of the information perceived through the haptic interface. While classical methods are substantially *open loop* descriptions, in these new models operator actively take part to system stability, varying the control signal according to task requirement and personal experience (*closed loop* interaction). Such behaviour has been intensively studied in the aeronautical field, where pilot's reactions against plants characteristics is fundamental to guarantee safe flight conditions.

If we consider the operator as a feedback controller in pursuit of a tracking task, we can assume that it tries to stabilize the teleoperation system, closing the loop with a good phase margin. According to McRuer [53], it's possible to define a *crossover model* as in Figure 2.5b, where human behaviour is modelled with the transfer function:

$$C_h(s) = K\frac{e^{-\tau s}}{1+Ts}\frac{1+T_z s}{1+T_p s} \tag{2.13}$$

τ and T represent, respectively, neuromuscular activation latency and limited actuation bandwidth of the human controller. Reasonable values for τ are around [30,100] ms, while T is approximately [0.15,0.25] s. Gain K and lead-lag compensator T_z and T_p are tuned adaptively by the operator in order to maintain a stable crossover frequency ω_c for the loop transfer function

$$L(s) = C(s)G(s) \approx K_c\frac{e^{-\tau_c s}}{s}, \qquad s = j\omega_c$$

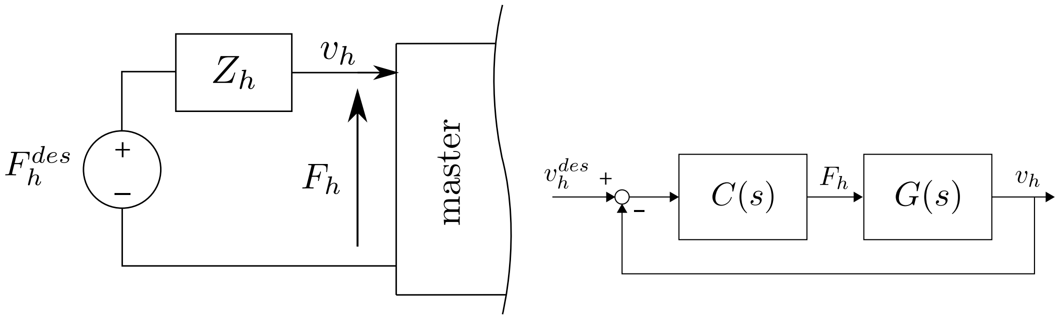
where $G(s)$ represents the master dynamics seen by the human. Input reference signal for this loop can be either *position error* (free motion) or *force error* (contact with environment) according to operator task requirements. Criticalities arise in the transition phase, when switching between position to force tracking is required. If the system is affected by time delay, incoherency between force feedback and master position/velocity can easily destabilize the operator's control loop (loss of causality between perceived and applied forces);

The operator's model choice relies on the following considerations: if the human is well trained and the task is known, he uses only little information coming from the external environment. He takes advantage of his experience and performs the operation almost as an open loop controller, showing a quite constant impedance. For this situation, the simplified model (Figure 2.5a) is enough to describe his behaviour. On the other hand, if the task is almost unknown and/or there is uncertainty in the teleoperation environment, the human needs to highly consider sensory information to improve/accomplish his objective. In this case, it is better to adopt more complex models, like the McRuer one (Figure 2.5b).

2.3 Control scheme architectures

Once the main teleoperation components have been introduced, it is necessary to analyse their mutual interaction. This is determined by the *teleoperator control structure*, which elaborates the information coming from the master and slave systems in order to fulfil the performance requirements stated in subsection 2.1.2.

According to literature [44, 80, 34], there are several ways to represent the MIMO system which identifies the teleoperation plant, depending on the available measurements and the required degree of complexity. In the following subsections, a generic



(a) *Static operator model*

(b) *McRuer crossover model*

Figure 2.5: Two possible operator models: fixed impedance with imposed force reference, McRuer crossover with speed tracking configuration

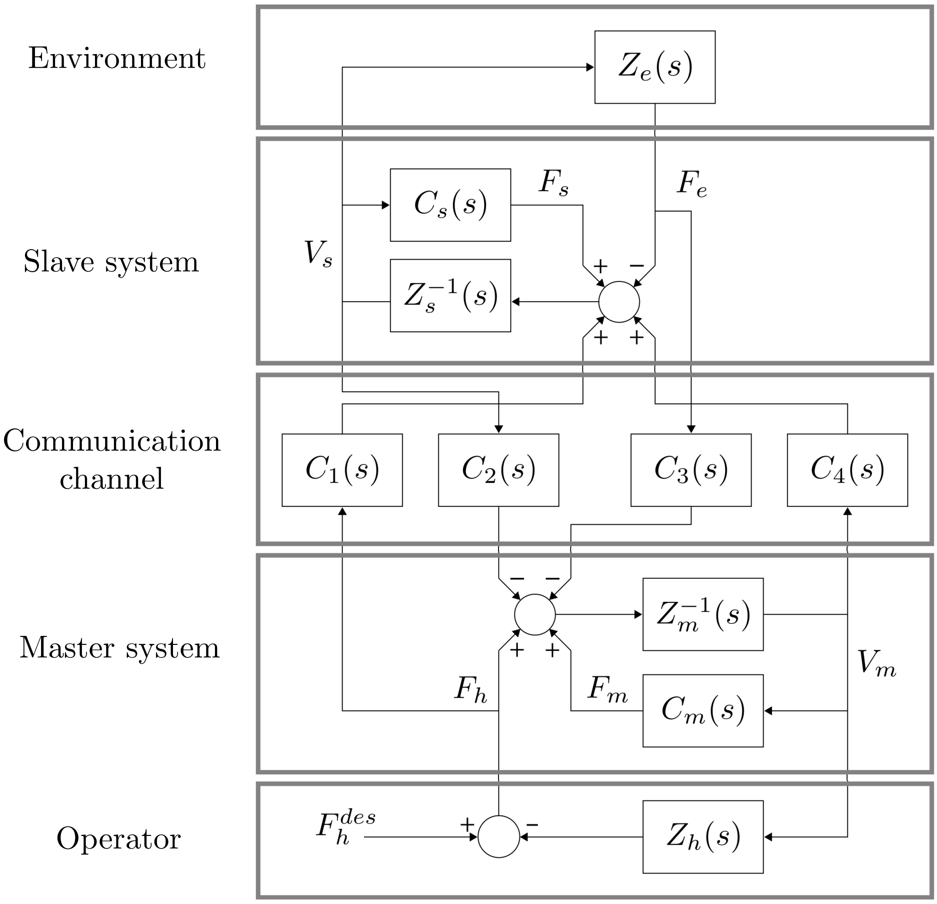


Figure 2.6: Block diagram representation for a generic bilateral teleoperation control scheme

control structure, introduced by Lawrence [44], has been proposed. Then, the architecture is specialized in order to generate the major number of teleoperation control schemes currently available. In the last subsection, previously proposed controllers are evaluated according to their pro and cons, choosing the most suitable for our application.

2.3.1 Generalized control scheme

Starting from Figure 2.1, a more rigorous scheme has been developed using *block diagram* formalism. This representation keeps a clear physical meaning for the teleoperation components, while it underlines the interactions generated by the network architecture. The resulting diagram is visible in Figure 2.6 and the meaning of each symbol is described in Table 2.1.

Symbol	Meaning
$Z_e(s)$	Environmental impedance
$C_s(s)$	Local slave controller
$Z_s(s)$	Slave device model
$C_{1-4}(s)$	Master-slave communication channel descriptors
$Z_m(s)$	Master device model
$C_m(s)$	Local master controller
$Z_h(s)$	Operator impedance

Table 2.1: Nomenclature for the generic control scheme architecture

We can notice that each subsystem (operator, master system, etc.) is described using a single transfer function, which defines the impedance relationship between its conjugate variables. For master and slave devices, it is possible to control the value of such impedance, in order to obtain the desired behaviour required by the teleoperation system.

In the general control architecture, *all* information available at the master side can be sent through the communication channel to the slave side and vice versa. Since we are describing our mechanical system based on an equivalent representation (effort - flow formalism), the meaningful quantities that we can broadcast through the channel are *four*: master-slave velocity and master-slave force.

Sent signals can be processed using communication channel descriptors, in order to correctly interface master and slave habitats. Usually, *scaling factors* are required; this because master devices are designed to be operator compliant, while slave manipulators are task dependent and could generate incompatible forces and speeds. Other common descriptors are used to define *channel properties*, such as time delays, packet loss and signal degradation.

The desired human force F_h^{des} is used by the operator to impose a given trajectory to the master device and, consequently, to the entire teleoperation scheme.

2.3.2 2 - channel architectures

First proposed teleoperation control schemes aimed at emulating early bilateral manipulators, already developed for nuclear applications and hazardous material processing. These systems were constituted by similar master and slave devices, mechanically coupled in order to reproduce the same displacement in the remote environment.

To replicate their behaviour, only the information about robots position needs to be transmitted; consequently, this architecture takes the name of *position - position controller* (PP). Doing so we are creating a *virtual* mechanical coupling between master and slave devices; $x_m, \dot{x}_m$ act as path reference for the slave controller, while $x_s, \dot{x}_s$ generates the haptic feedback for the master controller.

Control signals for the master and slave systems have the following form:

$$\begin{aligned} k_{bm}(\dot{x}_s - \dot{x}_m) + k_{km}(x_s - x_m) &= f_m \qquad \text{Master device} \\ k_{bs}(\dot{x}_m - \dot{x}_s) + k_{ks}(x_m - x_s) &= f_s \qquad \text{Slave device} \end{aligned} \tag{2.14}$$

where k_{bm}, k_{km} are the proportional and derivative parameters for the master side, while k_{bs}, k_{ks} plays the analogous role at the slave side. These terms define the stiffness of the virtual coupling as well as the amount of compliance showed to the operator/environment.

This control strategy is totally *symmetrical* and aims at synchronizing master and slave movements by means of local feedback regulators. Since it aims at emulating a mechanical spring-damper link, position - position controllers are *intrinsically passive* if controller coefficients are positive [38].

From the Lawrence point of view (Figure 2.6), this controller can be obtained by imposing $C_1(s)$ and $C_3(s)$ equal to zero, while $C_2(s)$ and $C_4(s)$ are equal to $1/s$. Velocity is retrieved from position via numerical differentiation.

Another well known 2 - channel architecture is the so called *position - force controller* (PF). According to [3], this is the most intuitive teleoperation control scheme, because it transmits position reference from master to slave (achieve good tracking) and external forces from slave to master (achieve haptic feedback).

Also in this case $x_m, \dot{x}_m$ represent the slave controller's reference, while f_e instead of x_s is transmitted to the master. Obviously this implies the presence of a *force sensor* on the slave side or, at least, an estimator which is able to produce such signal. Controllers equations become:

$$\begin{aligned} -k_f f_e &= f_m && \text{Master device} \\ k_{bs}(\dot{x}_m - \dot{x}_s) + b_{ks}(x_m - x_s) &= f_s && \text{Slave device} \end{aligned} \tag{2.15}$$

where k_{bs} and k_{ks} have the same meaning of position - position architecture, while k_f is the reflected force scaling from the slave to the master device.

This control strategy is *asymmetrical*, because resulting closed-loop equations have a different structure. In order to write this scheme according to the general structure proposed in Figure 2.6, this time $C_1(s)$ and $C_2(s)$ should be equal to zero, while $C_3(s)$ and $C_4(s)$ define communication structure (scaling factors, delays, etc.).

2.3.3 3 - channel architectures

These architectures can be seen as an evolution of the 2 - channel control schemes in order to overcome their intrinsic limitations . As mentioned in [37], every control strategy which requires slave position/speed feedback suffers from *delay induced forces*. To show this phenomenon, it's necessary to apply control laws (2.14) when communication channel is affected by time delay:

$$\begin{aligned} k_{bm}(\dot{x}_s^d - \dot{x}_m) + k_{km}(x_s^d - x_m) &= f_m && \text{Master device} \\ k_{ks}(\dot{x}_m^d - \dot{x}_s) + k_{ks}(x_m^d - x_s) &= f_s && \text{Slave device} \end{aligned} \tag{2.16}$$

where apex d stands for delayed variable ($x^d(t) = x(t-d)$). Focusing on the master equation, its expression can be rewritten by adding and subtracting $\dot{x}_s$ and x_s:

$$k_{bm}(\dot{x}_s - \dot{x}_m) + k_{km}(x_s - x_m) + \underbrace{k_{bm}(\dot{x}_s^d - \dot{x}_s) + k_{km}(x_s^d - x_s)}_{\text{delay induced forces}} = f_m \tag{2.17}$$

if k_{bm} and k_{km} have too high values, lag between master and slave positions generates strong reaction forces reflected to the operator.

To gain advantage from the direct slave force reflection and avoid delay induced forces, an interesting extension of the position - force architecture can be represented by the *position, force - force* (PF-F) architecture. In this case also the operator force is sent to the slave device, requiring additional information with respect to the 2 - channel control structure. However, this disadvantage is compensated by increased teleoperation performance, as well as the possibility of implementing an absolutely stable controller [86].

From the Lawrence point of view, this kind of architecture can be represented by setting $C_1(s)$ and $C_3(s)$ equals to one (or a scaling factor in case of telefunctioning), while $C_4(s)$ is still $1/s$ and $C_2(s)$ is not used.

2.3.4 4 - channel architectures

As the name already suggests, this category of bilateral controllers considers *all* the information available at master - slave sides to perform the control action. Although 4 - channel architectures are more complex than the previous ones (they require knowledge of external forces on both sides), they can be tuned in order to optimize system's transparency while retaining stability [44].

Specific transfer functions are associated at each channel descriptor, according to the bandwidth at which the teleoperation system will operate. In this case, it's not immediate to derive closed-loop equations, since they strongly depend on the chosen values assigned to $C_1(s)$, $C_2(s)$, $C_3(s)$ and $C_4(s)$.

As mentioned in [29], Lawrence optimal tuning will require slave/master acceleration transmission. This is usually not possible, therefore only an approximated behaviour can be obtained.

2.3.5 Control architecture choice

Now that the major teleoperation control architectures have been presented, we can analyse their pros and cons in order to decide which is the best strategy to realize our bilateral control scheme.

The position - position architecture, as already mentioned, has been the first one to be implemented thanks also to its simplicity. In fact, it doesn't require external torque-force sensors to perceive environment, while position and speed signals are immediately available on every robotic system. Moreover, choosing a sufficiently high $\bar{b}_m$ and $\bar{b}_s$ guarantees system's global stability [56].

Main drawbacks are related to the overall bad performance of these controllers, in terms of telepresence and operator's handling. As reported in [37], virtual mechanical coupling generated by position - position controllers causes the *reflection* of the slave impedance to the master; this implies that the operator should overcome also the remote dynamics to perform the required task. Eventually, the ultimate consequence is an altered perception of the environment, which is generally felt "softer" than it actually is. Another unpleasant effect associated with PP controllers is the already mentioned *delay induced force*, that causes the teleoperation system to be sluggish in free space motion.

For what concern position - force architectures, they tend to better perform with respect to the position - position ones, since environment is directly perceived through the remote measured force. Nevertheless, this kind of schemes tends to suffer from communication channel delay and stability is a critical factor [82]; when the slave enters in contact with stiff environments, it bounces, generating high frequency reaction forces. Thanks to direct reflection, this signal couples with operator's dynamics, which becomes an integral part of the control loop [31]. Ultimately, the operator should deal with this impulsive response correctly to regain system's stability, which is not intuitive when there's time delay.

In view of the problems associated with 2 - channel architectures, it seems reasonable to look at more complete schemes to improve system's performance. In particular, 3 - channel architecture represents a good compromise between control complexity and exploitation of information. Indeed, adding the operator force feedforward to the scheme (PF-F configuration) allows us to *completely* assign master and slave dynamics, while retaining all the good properties of P-F architecture.

Unfortunately, also in this case we have instability associated with hard contact reflection. To mitigate this drawback, teleoperation network can be stabilized by means of different techniques (damping injection, passivity-based approach, wave variables, etc. [38]).

The remaining question is now represented by the convenience of increasing again architecture's order, choosing a 4 - channel structure. Although it guarantees op-

	P-P	P-F	PF-F	PF-PF
requires force sensors		●	●	●
intrinsically stable	●			
slave dynamics reflection	●			●
completely assignable impedance			●	●
delay induced forces	●			●
optimized for transparency				●
stiff contact instability		●	●	●

Table 2.2: Overview table of the main teleoperation architecture's advantages - disadvantages

timal transparency, resultant bilateral controller needs to be treated as a complete MIMO system, which implies more involved parameter tuning as well as more complex and computational demanding control techniques (H_∞,μ-synthesis). Moreover, Lawrence developments require a good knowledge of master and slave devices, which is often not the case (model errors, friction, etc.). Eventually, introduction of slave position feedback will generate again delay induced forces [37].

For all these reasons, the control architecture chosen for implementing the bilateral teleoperation scheme is the position/force - force one, since it represents a good trade-off between too simplified strategies (PP, PF) and too general ones (PF-PF). Moreover, we assume that both environmental and human forces are measurable quantities. We summarize the main concepts of this section in Table 2.2. For each column, pro and cons of the presented architectures have been reported. In this way, it's possible to visually compare them in order to choose the most effective one.

Chapter 3

Local controller

In this chapter, a detailed analysis on system local controllers has been conducted. With the term *local controller*, we indicate the control structure, attached to the master and slave devices, which is in charge of determining their desired mechanical properties.

First of all, a brief recall on impedance control is presented, describing how it can be applied to our ideal 1 DOF model for master and slave manipulators. Then, its limitations are shown and a new robust variable structure controller is introduced, called Sliding Mode Controller (SMC). Also in this case, the design strategy is developed for both systems. The main SMC drawbacks are then reported. To overcome such limitations, evolutions of the standard SMC has been considered (Integral Sliding Mode Controller (ISMC), continuous approximations of signum function, Higher Order Sliding Mode Controller (HOSMC)). Standard SMC is then compared with these techniques via a `MATLAB - Simulink` simulation. At the end, a brief introduction to Sliding Mode Model Predictive Controller (SMPC) is done, with emphasis again on the 1 DOF teleoperation set-up.

3.1 Inverse dynamics control

To deal with multivariable non-linear systems, a very useful control technique is represented by *feedback linearization* [43]. It consists in the removal of the system undesired dynamics through a non-linear feedback signal and it is particularly useful when the control input can be easily made explicit from system's dynamic equations. In the robotic field, feedback linearization goes under the name of *inverse dynamics control* [73].

Let's consider the dynamic equations of the master manipulator (analogous reasoning can be done for the slave device):

$$\boldsymbol{B}_m(\boldsymbol{q}_m)\ddot{\boldsymbol{q}}_m + \boldsymbol{n}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m) = \boldsymbol{\tau}_m + \boldsymbol{J}_m^\top(\boldsymbol{q}_m)\boldsymbol{h}_h \tag{3.1}$$

where, for simplicity, we have assumed $\boldsymbol{n}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m) = \boldsymbol{C}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m)\dot{\boldsymbol{q}}_m + \boldsymbol{g}_m(\boldsymbol{q}_m)$ to be the main non-linearity source of our system. Dependency on time of kinematic quantities here is omitted for clarity. Note that equation (3.1) has always a linear structure with respect to the control input and the inertia matrix is positive definite [73].

Therefore, we can choose $\boldsymbol{\tau}_m$ such that:

$$\boldsymbol{\tau}_m = \hat{\boldsymbol{n}}_m(\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m) - \boldsymbol{J}_m^\top(\boldsymbol{q}_m)\hat{\boldsymbol{h}}_h + \hat{\boldsymbol{B}}_m(\boldsymbol{q}_m)\boldsymbol{v} \tag{3.2}$$

where $\hat{\boldsymbol{n}}_m, \hat{\boldsymbol{B}}_m, \hat{\boldsymbol{h}}_h$ are the estimated values of the dynamic parameters of master manipulator, while $\boldsymbol{v}$ can be considered as the new control input for the controlled scheme. Since, usually, $\hat{\boldsymbol{n}}_m \neq \boldsymbol{n}_m, \hat{\boldsymbol{B}}_m \neq \boldsymbol{B}_m$ and $\hat{\boldsymbol{h}}_h \neq \boldsymbol{h}_h$, we indicate the estimation error $\boldsymbol{\Delta n} = \hat{\boldsymbol{n}}_m - \boldsymbol{n}_m$ and similarly for the other terms.

Substituting expression (3.2) into (3.1), the resulting closed-loop equation has the following form:

$$\ddot{\boldsymbol{q}}_m = \boldsymbol{\zeta}_m \boldsymbol{v} + \boldsymbol{\eta}_m \tag{3.3}$$

where $\boldsymbol{\zeta}_m = \boldsymbol{B}_m^{-1}\hat{\boldsymbol{B}}_m$ and $\boldsymbol{\eta}_m = \boldsymbol{B}_m^{-1}[\boldsymbol{\Delta n} - \boldsymbol{J}_m^\top \boldsymbol{\Delta h}_h]$.

If we perfectly know the manipulator's dynamics, $\boldsymbol{\zeta}_m = \mathbf{1}$ and $\boldsymbol{\eta}_m = \mathbf{0}$. Therefore, expression (3.3) can be simplified and the resulting controlled system is represented by:

$$\ddot{\boldsymbol{q}} = \boldsymbol{v} \tag{3.4}$$

Equation (3.4) describes a *linear* and *decoupled* n DOF system with respect to the new input $\boldsymbol{v}$. Based on this result, control specifications now can be easily fulfilled by

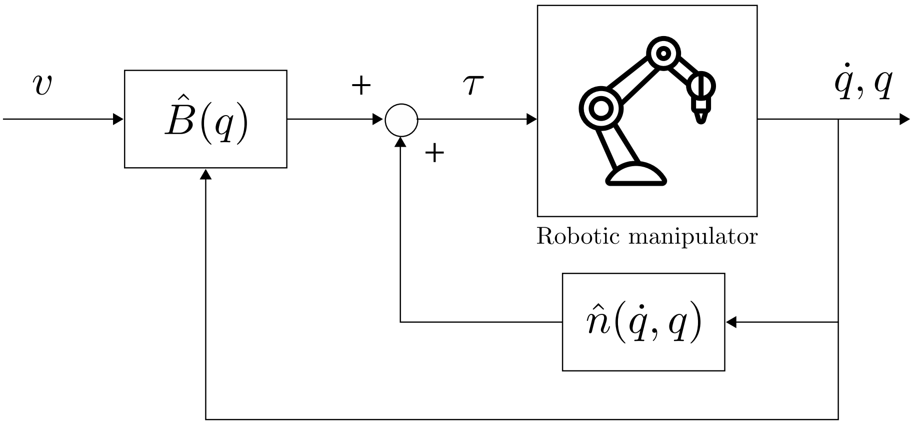


Figure 3.1: Inverse dynamics control scheme block diagram

means of any control strategy which is able to deal with a chain of n decoupled double integrators. Typical solutions to this problem are decentralized control schemes, such as Proportional Derivative (PD) controllers, that allow a complete tracking error definition.

The control law represented in equation (3.2) is non-linear, coupled and it represents the main drawback of the inverse dynamics scheme. Indeed, the flawed knowledge of manipulator's structure, joint with variable operative conditions, underline the poor robustness of this control concept. Moreover the computational burden required for $\hat{\boldsymbol{n}}_{\boldsymbol{m}}, \hat{\boldsymbol{B}}_{\boldsymbol{m}}, \hat{\boldsymbol{J}}_{\boldsymbol{m}}$ can be significant and it needs to be performed for each sample time (each different configuration) [73].

For a generic manipulator, inverse dynamics architecture is reported at Figure 3.1.

3.2 Impedance control

According to Colgate [83], *"haptic displays can be considered to be devices which generate mechanical impedances (relationship between velocity and force)"*. Since our aim in bilateral teleoperation is to make the operator *aware* of what happens at the remote site (where the slave manipulator is operating), we need to design master and slave controllers that are able to trustfully represent virtual environments.

In order to reach this goal, it's necessary to control the *dynamic interaction* between the manipulator and its environment. To do so, we need to impose a given

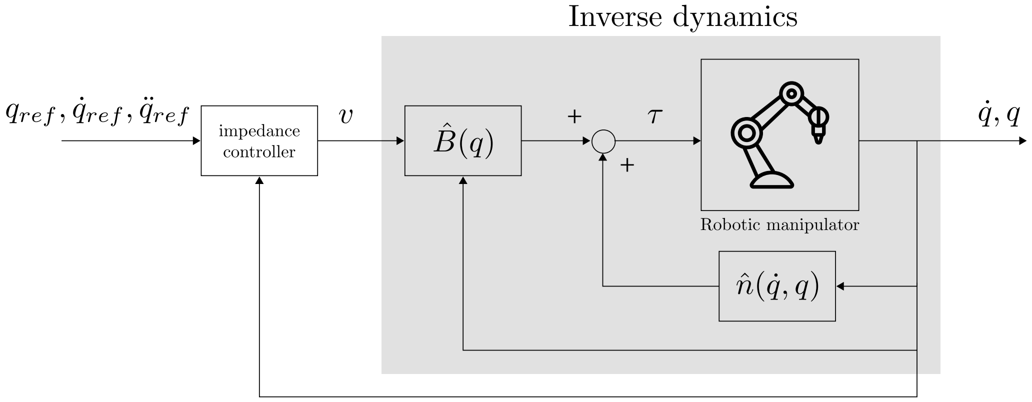


Figure 3.2: Impedance control scheme block diagram

mechanical impedance to the system, so that it can react to external stimuli in a predictable and safe way.

Control schemes which are able to impose a given mechanical relationship between force and speed are called *impedance controllers*. Through them, it's possible to fulfil the main requirements both for master and slave manipulators:

- From the *operator* point of view, impedance control allows to set up desired mass - spring - damper values, resulting in a smooth and pleasant teleoperation task. Moreover, force feedback generation is straightforward (it can be seen as an additional force acting on the haptic interface);

- From the *slave device* point of view, impedance control determines the link between external environmental forces and master - to - slave tracking error. In this way, we can explicitly take into account the manipulator's degree of compliance as well as guaranteeing asymptotic error convergence during free motion;

To properly apply an impedance controller to a mechanical device (manipulator, 1 DOF system, etc.), it is necessary to *cancel* its intrinsic impedance through an appropriate inverse dynamics controller, of the type introduced in section 3.1. This inner control loop is reported in the grey box of Figure 3.2. Once inverse dynamics has been performed, the new impedance feedback regulator is connected to the linearized system and the desired mechanical impedance is reached.

In the next subsections, slave and master analysis has been performed on the simplified 1 DOF model, in order to avoid difficult expressions and focus on the intrinsic properties and drawbacks of impedance control schemes.

3.2.1 Master device

Recalling equation (2.4), the master dynamic system for the 1 DOF model is represented by:

$$m_m \ddot{x}_m + b_m \dot{x}_m = f_m + f_h \tag{3.5}$$

According to [10, 28], master device should show to the operator a desired impedance characteristic of the following type:

$$\bar{m}_m \ddot{x}_m + \bar{b}_m \dot{x}_m + \bar{k}_m x_m = f_h - k_f f_e^d \tag{3.6}$$

where $\bar{m}_m, \bar{b}_m, \bar{k}_m$ are the desired values for the inertia, damping and stiffness coefficients and k_f represents a constant force scale factor, useful to adapt master and slave different structures. Eventually, f_e^d is the delayed environmental force coming from the slave device.

To obtain the relationship (3.6) from master's dynamics model (3.5), the following control law needs to be applied:

$$f_m = \hat{b}_m \dot{x}_m - f_h + \frac{\hat{m}_m}{\bar{m}_m}(f_h - k_f f_e^d - \bar{b}_m \dot{x}_m - \bar{k}_m x_m) \tag{3.7}$$

where "hat" terms indicate, as usual, estimated quantities, while "bar" terms indicate the desired ones. Here, the inverse dynamics control is represented by $f_m = \hat{b}_m \dot{x}_m - f_h + \frac{\hat{m}_m}{\bar{m}_m} v$, while the impedance control is given by $v = f_h - k_f f_e^d - \bar{b}_m \dot{x}_m - \bar{k}_m x_m$.

Writing equation (3.7) in the Laplace domain using the Lawrence formalism leads to the transfer function:

$$C_m(s) = \frac{F_m}{V_m} = \hat{b}_m - \frac{\hat{m}_m}{\bar{m}_m}\left(\bar{b}_m + \frac{\bar{k}_m}{s}\right) \tag{3.8}$$

where F_m and V_m represent, respectively, the master force control and the master speed expressed in the frequency domain. Estimated parameters can be substituted with the real ones if there's no uncertainty in the modelled dynamics (i.e. perfect cancellation of master's impedance).

3.2.2 Slave device

Similarly to the master device, also for the slave manipulator we can rewrite the dynamic system for the 1 DOF model:

$$m_s \ddot{x}_s + b_s \dot{x}_s = f_s - f_e \tag{3.9}$$

This time, the main target for the slave system is to properly track reference position and speed coming from the master device. Therefore, it's necessary to define the manipulator *tracking error* $\tilde{x}_s = x_s - x_s^{des}$, where x_s^{des} is the desired slave position and it's computed based on the communicated master coordinate $x_s^{des} = k_p x_m^d$.

To take advantage of the impedance control formalism introduced in section 3.2, we need to design a controller that guarantees a closed - loop error equation of the following type [10, 27]:

$$\bar{m}_s \ddot{\tilde{x}}_s + \bar{b}_s \dot{\tilde{x}}_s + \bar{k}_s \tilde{x}_s = -f_e \tag{3.10}$$

By imposing a second order dynamics to the tracking error, we are assuring that, in free motion ($f_e = 0$), $\tilde{x}_s \to 0$. Furthermore, $\tilde{x}_s$ will follow a trajectory defined by the design parameters $\bar{m}_s, \bar{b}_s, \bar{k}_s$.

When the slave device enters in contact with the environment ($f_e \neq 0$), equation (3.10) behaves like an impedance controller between the tracking error and the external force. Also in this case, the degree of compliance is dictated by the desired mass - spring - damper parameters.

To realize equation (3.10), we can apply to the slave device the same reasoning done for the master haptic system. Our controller will be composed of an inverse dynamics term, which will cancel the intrinsic slave model, and an impedance control term, which will impose the desired dynamics to the remote manipulator. This turns out to be the following control equation [9]:

$$f_s = \hat{b}_s \dot{x}_s + f_e + \frac{\hat{m}_s}{\bar{m}_s} \left(-f_e + \bar{m}_s k_p \boxed{\ddot{x}_m^d} - \bar{b}_s \dot{\tilde{x}}_s - \bar{k}_s \tilde{x}_s \right) \tag{3.11}$$

Unfortunately, equation (3.11) cannot be directly applied to the slave device, due to its dependency from the *master acceleration term* $\ddot{x}_m^d$. This information typically is not available, even at the master side, because acceleration measurements are difficult to be performed and heavily affected by noise. A possible workaround for this problem is reported in [9]. It consists in replacing the unknown term with the master dynamic equation, which is known thanks to the impedance controller designed in subsection 3.2.1.

If we write the delayed closed - loop master dynamics and we make it explicit with respect to the acceleration signal, we obtain:

$$\ddot{x}_m^d = \frac{1}{\bar{m}_m} \left(f_h^d - \bar{b}_m \dot{x}_m^d - \bar{k}_m x_m^d - \boxed{k_f f_e^{dd}} \right) \tag{3.12}$$

where the superscript *dd* is used to label signals that have been delayed twice, i.e. they travelled along the communication channel backward and forward (Round Trip Time (RTT)). If the link between master and slave is *symmetric*, it can be seen as the double of the standard delay introduced in unilateral communication.

We can notice that the boxed term in (3.12) represents the haptic feedback given to the master by the slave, which is sent back to the slave itself to define the remote site controller.

If we substitute the acceleration term in (3.11) with the master controlled dynamics (3.12), we finally obtain the control law that should be applied to the master:

$$\begin{aligned} f_s =& \hat{b}_s \dot{x}_s + f_e + \frac{\hat{m}_s}{\bar{m}_s} \left[\frac{\bar{m}_s k_p}{\bar{m}_m} \left(f_h^d - \bar{b}_m \dot{x}_m^d - \bar{k}_m x_m^d - k_f f_e^{dd} \right) - \bar{b}_s \dot{\tilde{x}}_s - \bar{k}_s \tilde{x}_s - f_e \right] \\ =& \left(\hat{b}_s - \frac{\hat{m}_s}{\bar{m}_s} \bar{b}_s \right) \dot{x}_s - \frac{\hat{m}_s}{\bar{m}_s} \bar{k}_s x_s + \left(1 - \frac{\hat{m}_s}{\bar{m}_s} \right) f_e + k_p \hat{m}_s \left(\frac{\bar{b}_s}{\bar{m}_s} - \frac{\bar{b}_m}{\bar{m}_m} \right) \dot{x}_m^d \\ & + k_p \hat{m}_s \left(\frac{\bar{k}_s}{\bar{m}_s} - \frac{\bar{k}_m}{\bar{m}_m} \right) x_m^d + \frac{\hat{m}_s}{\bar{m}_m} k_p f_h^d - \frac{\hat{m}_s}{\bar{m}_m} k_f k_p f_e^{dd} \end{aligned} \tag{3.13}$$

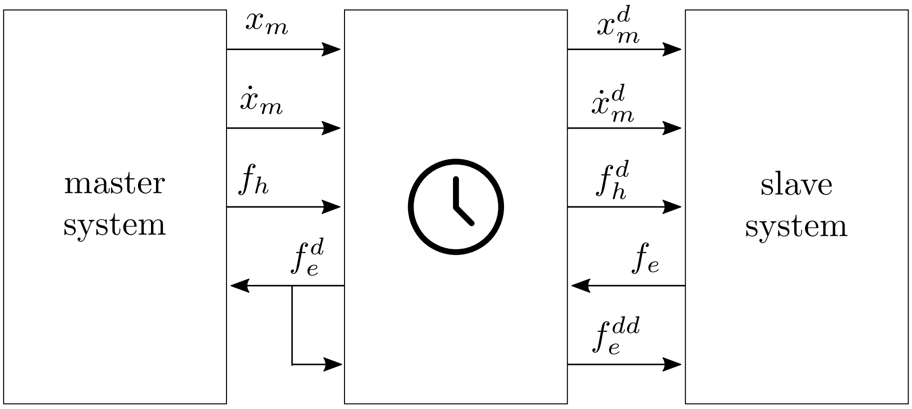


Figure 3.3: Impedance - impedance controllers signals flow through teleoperation architecture

where the second equation is obtained from the first by substituting the tracking error expression and collecting the parameters associated to each signal. Since we are using the information about operator's force (f_h^d), this controller can be listed as a 3 - channel architecture. Signal flows along the bilateral teleoperation system are shown in Figure 3.3.

To fit control law (3.13) in Lawrence teleoperation scheme (Figure 2.6), we need to write the transfer functions associated to $C_s(s), C_1(s)$ and $C_4(s)$:

$$\begin{aligned} C_s(s) &= \frac{F_s}{V_s} = \hat{b}_s - \frac{\hat{m}_s}{\bar{m}_s}\bar{b}_s + \frac{\hat{m}_s}{\bar{m}_s}\bar{k}_s\frac{1}{s} \\ C_1(s) &= \frac{F_h^d}{F_h} = \frac{\hat{m}_s}{\bar{m}_m}k_p e^{-\tau s} \\ C_4(s) &= \frac{V_m^d}{V_m} = k_p\hat{m}_s\left[\left(\frac{\bar{b}_s}{\bar{m}_s} - \frac{\bar{b}_m}{\bar{m}_m}\right) + \left(\frac{\bar{k}_s}{\bar{m}_s} - \frac{\bar{k}_m}{\bar{m}_m}\right)\frac{1}{s}\right]e^{-\tau s} \end{aligned} \tag{3.14}$$

If the delay is a-priori known, double delayed environmental force can be directly computed inside the slave regulator and we can still maintain the generalized control scheme formalism. If we need to get back the force feedback signal from the master, an *additional channel* should be added. This is usually done because in most cases delay is unknown and moreover it may vary with time.

3.2.3 Stability limits

Once the control laws for the master and slave devices have been derived, we need to assess properties and limits of the overall teleoperation scheme. As reported in [10], performance of impedance controllers which are based on an inverse dynamics process *may deteriorate severely when uncertainty exists in model's parameters.*

If we analyse the controlled system during ideal functioning ($\hat{m}_{m,s} = m_{m,s}$, $\hat{b}_{m,s} = b_{m,s}$), master and slave dynamics are perfectly cancelled and the two closed - loop equations matched the desired ones (3.6) and (3.10).

Both represent second order systems, with assignable poles. Thus, all the performance and stability requirements can be simply imposed by using standard pole placement techniques.

Let's now analyse the *robust behaviour* of the impedance controller, by considering imperfect knowledge of model parameters as well as unmodelled - unknown dynamics (e.g. friction forces, measurement noise, etc.). Let's suppose that these effects generate a mismatch between the real manipulator's coefficients and the estimated ones; for now, neglect differences between the measured force values and the actual ones. Applying the control laws (3.7) and (3.13) to the real systems (3.5) and (3.9) generates the following closed - loop expressions:

$$\frac{m_m}{\hat{m}_m}\bar{m}_m\ddot{x}_m + \left[\bar{b}_m + \frac{\bar{m}_m}{\hat{m}_m}(b_m - \hat{b}_m)\right]\dot{x}_m + \bar{k}_m x_m = f_h - k_f f_e^d \tag{3.15}$$

$$\bar{m}_s\left(\frac{m_s}{\hat{m}_s}\ddot{x}_s - k_p\ddot{x}_m^d\right) + \frac{\bar{m}_s}{\hat{m}_s}(b_s - \hat{b}_s)\dot{x}_s + \bar{b}_s\dot{\tilde{x}}_s + \bar{k}_s\tilde{x}_s = -f_e \tag{3.16}$$

Without dwelling on cumbersome computations for the slave device equation, we can look at the master one to understand the main limitations of impedance control approach. For what concern stability issues, let's focus on the coefficients that describe the dynamic equation:

$$\underbrace{\frac{m_m}{\hat{m}_m}\bar{m}_m}_{\text{a}}\ddot{x}_m + \underbrace{\left[\bar{b}_m + \frac{\bar{m}_m}{\hat{m}_m}(b_m - \hat{b}_m)\right]}_{\text{b}}\dot{x}_m + \underbrace{\bar{k}_m}_{c} x_m = f_h - k_f f_e^d$$

To guarantee that the system is stable, all the roots of the characteristic polynomial as^2+bs+c must have negative real part. Thanks to the Descartes' rule of signs, the previous condition can be verified by imposing the following set of inequalities:

$$\begin{cases} \frac{m_m}{\hat{m}_m}\bar{m}_m > 0 \\ \bar{b}_m + \frac{\bar{m}_m}{\hat{m}_m}(b_m - \hat{b}_m) > 0 \\ \bar{k}_m > 0 \end{cases} \tag{3.17}$$

Since all parameters are positive by definition (they represent physical masses, damping and spring constants), the first and last equations are obviously verified $\forall \hat{m}_m, \bar{m}_m, \bar{k}_m$. The same cannot be said for the middle one. If we isolate the estimated quantities from the desired ones, we obtain the subsequent stability condition:

$$\frac{b_m - \hat{b}_m}{\hat{m}_m} > -\frac{\bar{b}_m}{\bar{m}_m} \tag{3.18}$$

From inequality (3.18) we can draw some interesting considerations. First of all, the right side of the condition is always negative; this implies that an underestimation of the friction force ($b_m \geq \hat{b}_m$) is sufficient to gain system stability. If this is not the case ($b_m < \hat{b}_m$), choice of the desired parameters becomes critical. If we want a heavy and lightly damped master device, $\frac{\bar{b}_m}{\bar{m}_m} \to 0$ and the stability region reduces. Also the estimated mass $\hat{m}_m$ plays an important role in scaling up or down the error generated by the damping factor.

Beyond the stability issue, also in the case where we are able to guarantee the fulfilment of inequality (3.18), the desired dynamic behaviour for the master system is strongly compromised, since coefficient's a, b, c are very different from the designed ones. Moreover, the real system may undergo further changes, resulting in a variation of b_m and m_m. Eventually, this variation may violate stability limit and the system could become unstable.

Thanks to all these considerations, we can state that:

- impedance controller, joint with inverse dynamics, is a good control strategy for applying a defined dynamic model to a mechanical device, such as master - slave manipulators;

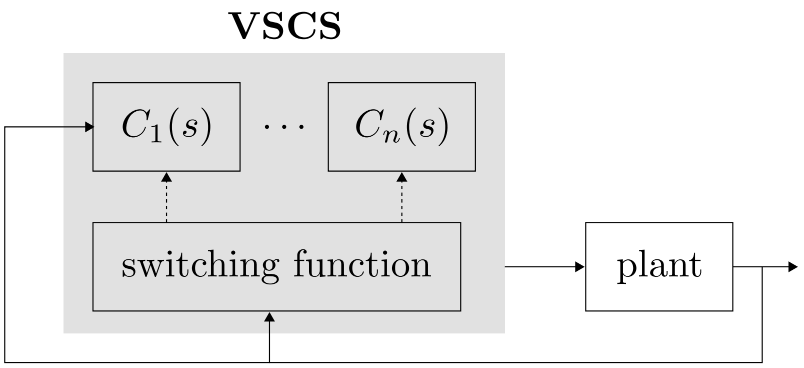


Figure 3.4: General block diagram for a variable structure control system

- inverse dynamics strongly depends on model's structure and parameters, making the system very sensitive to uncertainties;
- wrong choice of controller parameters and variations in the system's real dynamic model may cause the controlled device to drift from its desired impedance and start to behave unpredictably, even generating closed - loop instability;

3.3 Sliding mode control

In subsection 3.2.3 we have considered the main limitations that arise from a direct implementation of impedance controllers. Thus, it's necessary to find a control strategy which guarantees the desired dynamics imposition *robustly* with respect to the uncertainties that affect the system.

As reported in [14], to deal with the differences between the mathematical model and the real process, a particular control structure, called *sliding mode control* (SMC), has been introduced.

SMC belongs to the class of controllers which goes under the name of Variable Structure Control Systems (VSCSs). They have been developed thanks to the pioneering work performed by Soviet Union scientists at the end of 1970. Particular relevance should be given to the researches of U.Itkis [40] and V.I.Utkin [77].

The basic philosophy of a variable structure approach is to have a number of feedback control laws plus a decision rule, generally called *switching function* [74] (Figure 3.4). Through a proper choice of the control law that should be applied to the process at each time instant, it's possible to drive robustly the system near to the desired operative region. Using this kind of switching non-linear controllers, it is possible to achieve results which are not available with standard linear techniques (e.g. impedance control) [43].

To properly design a VSCS, we need to determine two components: the set of control actions to be applied to the plant and the decision rule that allows the switch between them. In sliding mode control theory, the switching function is usually called *sliding variable* and it's indicated with the symbol σ. A typical decision rule adopted for σ is the following:

$$C(s) = \begin{cases} K, & \sigma > 0 \\ 0, & \sigma = 0 \\ -K, & \sigma < 0 \end{cases} \tag{3.19}$$

where K represents a constant coefficient. The main goal of a sliding mode controller is to *drive in finite time* and *maintain* the controlled system as near as possible to the manifold $\sigma = 0$, which takes the name of *sliding surface* [14, 77]. If that happens, the controlled system will inherit all the characteristics of $\sigma = 0$ once it reaches it. Therefore, we can design σ so that it provides the desired properties we want to impose on the process, which will be transferred on the controlled plant thanks to the SMC . This strategy is clearly visible from equation (3.19); since we are changing the control structure each time σ swaps its sign, we are trying to make $\sigma = 0$ an *attractor* for the system's trajectories.

During SMC, the system can be in two different states:

- **Reaching state**: controlled system is not on the sliding surface yet. Its trajectory is dictated by the closed - loop equation generated by the regulator with fixed gain K (or $-K$);

- **Sliding mode**: controlled system is *attached* to the sliding surface $\sigma = 0$, moving along its trajectory thanks to the SMC switching logic;

When the controlled process is in sliding mode condition, it's usually called *equivalent system*. To better visualize how SMC operates, for second order differential

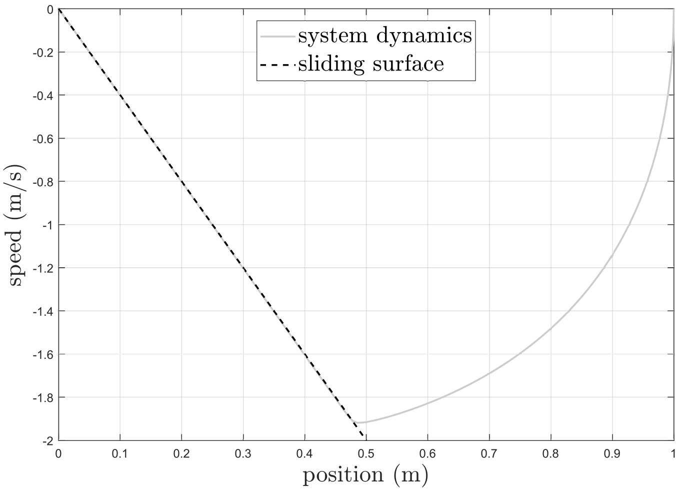


Figure 3.5: Standard phase plane evolution for an SMC controlled process

equations it is possible to represent state evolution using a *phase diagram* (or phase plane). Axis coordinates indicate the two system's state variables, while trajectories show the path followed by the model through time. In Figure 3.5 we can see a phase plane evolution for an SMC controlled process. Here, the two states previously listed are clearly noticeable: up to a given time t_f, the system is far from the sliding surface and thanks to the proportional controller it moves towards it. At time t_f, $\sigma = 0$ is reached and the system goes into sliding mode, following the dotted line imposed dynamics towards the equilibrium point $(0, 0)$.

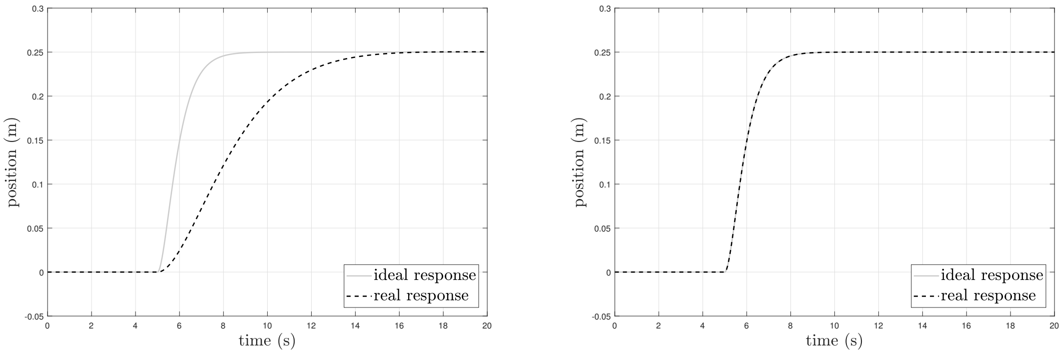
SMC has numerous advantages:

1. It **decouples** the design of system's desired dynamics from the project of a stable controller that is able to impose such dynamics;

2. As will be shown in subsection 3.3.4, the controlled system is able to reach the condition $\sigma = 0$ even in the presence of **disturbances** or **uncertainties** in the dynamic model;

3. Process dynamics associated to the sliding mode can be **arbitrarily assigned** through a proper choice of σ;

Thanks to points 2 and 3, SMC represents a good solution to alleviate the problems that affect the impedance control. As an example, in Figure 3.6 is reported



(a) *uncertainty without SMC*

(b) *uncertainty with SMC*

Figure 3.6: 1 DOF uncertain mechanical system affected by uncertainty: with and without SMC control

the position trajectory of a 1 DOF mechanical system (similar to the ones used for master and slave modelling), whose model is affected by uncertainty. On the left, we can see that the standard impedance controller generates a response dynamics which is very far from the desired one. The same system is then controlled via sliding mode techniques and the model mismatch rejection is clear.

3.3.1 Sliding mode design procedure

To properly design an SMC controller, it's necessary to define the model of the dynamic system we want to control, written in state - space form. Thus, we need to rewrite equations (3.5) and (3.9) choosing as state variables $\boldsymbol{x} = [\dot{x}_m \;\; x_m]^\top$, $\boldsymbol{z} = [\dot{x}_s \;\; x_s]^\top$.

$$\begin{aligned} \dot{x}_1 &= x_2 & \dot{z}_1 &= z_2 \\ \dot{x}_2 &= -\frac{b_m}{m_m}x_2 + \frac{1}{m_m}(f_h + f_m) & \dot{z}_2 &= -\frac{b_s}{m_s}z_2 + \frac{1}{m_s}(f_s - f_e) \end{aligned} \tag{3.20}$$

Since the main objective of a sliding mode controller is to assign a desired impedance to the system in spite of uncertainties which may affect it, we need to perform SMC *after* the inverse dynamic controller (3.2). Doing so, the resultant systems will be double integrators affected by model mismatches.

$$\begin{aligned} \dot{x}_1 &= x_2 & \dot{z}_1 &= z_2 \\ \dot{x}_2 &= \underbrace{\frac{\hat{b}_m - b_m}{m_m}}_{\eta_m(x_2)} x_2 + \underbrace{\frac{\hat{m}_m}{m_m}}_{\zeta_m} v_m & \dot{z}_2 &= \underbrace{\frac{\hat{b}_s - b_s}{m_s}}_{\eta_s(z_2)} z_2 + \underbrace{\frac{\hat{m}_s}{m_s}}_{\zeta_s} v_s \end{aligned} \tag{3.21}$$

where ζ_m, ζ_s, $\eta_m(x_2)$ and $\eta_s(z_2)$ are the parametric uncertainties which affect the models after inverse dynamics procedure. v_s and v_m are the new control inputs of the new dynamic models.

To deal with the systems in (3.21), a standard sliding mode is designed. Its control action can be written in the following form:

$$v = v^{eq} + v^{smc} \tag{3.22}$$

where v^{eq} is the *equivalent control law* and v^{smc} is the *discontinuous term*.

As reported in [14], v^{eq} is the control action which is required to maintain the system on the sliding surface. By definition, it is the *unique* solution of the algebraic equation $\dot{\sigma} = 0$.

For a more "practical" interpretation, we can think to the equivalent control as the *exact* controller which should be applied to the system in order to perform the desired task. Since our process is affected by model mismatches, v^{eq} will also be a function of such uncertainties (otherwise it would not be able to perform the task correctly). From this consideration, it turns out that the true equivalent controller *cannot* be directly applied as a control law, because it depends on unknown quantities.

Therefore, the best we can do is to design v^{eq} as if it was acting on the *nominal plant* instead of the real one. Then, the true equivalent controller will be realized by adding the discontinuous control term, so that we reach $\sigma = 0$.

For what concerns v^{smc}, it is responsible of compensating imperfect model knowledge and it is structured similarly to equation (3.19). More formally, it can be written as:

$$v^{smc} = -K \operatorname{sgn} \sigma = -K \frac{\sigma}{|\sigma|} \tag{3.23}$$

To find v^{eq}, v^{smc} and eventually u, the following design points need to be followed:

1. **Choice of the sliding manifold**: we need to assign the proper value of σ such that it owns the properties we are interested in;

2. **Equivalent control derivation**: by applying the v^{eq} definition, we use the value of $\dot{\sigma}$ to compute the nominal equivalent control law;

3. **Definition of the attraction region**: now that we have completed the computation of sliding mode control action, we need to check if the closed - loop system is stable or not: this will give us insights on the values to be assigned to K;

3.3.2 Sliding mode master device

Let's start computing the control signal for the master device. First of all, we need to assign a proper dynamic model to σ that fulfils our requirement. As we have done for the impedance controller, the ideal behaviour we want to obtain is represented by equation (3.6). Therefore, it seems logic to impose as sliding surface:

$$I = \bar{m}_m \ddot{x}_m + \bar{b}_m \dot{x}_m + \bar{k}_m x_m - f_h + k_f f_e^d \tag{3.24}$$

Doing so, if we are able to impose $I = 0 \ \forall t > t_f$ we are guaranteeing that the system is behaving correctly. If we write equation (3.24) using the state - space notation introduced in equation (3.20) we obtain:

$$I = \bar{m}_m \boxed{\dot{x}_2} + \bar{b}_m x_2 + \bar{k}_m x_1 - f_h + k_f f_e^d \tag{3.25}$$

Here, we can see why we *cannot apply* I as a sliding surface. By definition [19], the sliding manifold must have an order which is *less* than the one of the system. Since we are imposing a desired *second order* dynamics, the order of equation (3.25) is *equal* to the order of the mechanical system. Thus, we cannot choose it as a sliding surface due to its dependency from the master acceleration (boxed term).

To express σ as a function of the system's state ($\sigma(\boldsymbol{x})$), we need to get rid of

$\dot{x}_2$. A possible strategy to do so is reported in [10] and it consists in choosing the following sliding surface:

$$\sigma = \frac{1}{\bar{m}_m} \int_0^t I(\tau)\, d\tau \tag{3.26}$$

If we substitute expression (3.25) into equation (3.26), we obtain:

$$\begin{aligned} \sigma &= \dot{x}_m + \frac{\bar{b}_m}{\bar{m}_m} x_m + \frac{\bar{k}_m}{\bar{m}_m} \int_0^t x_m(\tau)\, d\tau + \frac{1}{\bar{m}_m} \int_0^t (k_f f_e^d(\tau) - f_h(\tau))\, d\tau \\ &= x_2 + \frac{\bar{b}_m}{\bar{m}_m} x_1 + \frac{\bar{b}_m}{\bar{m}_m} \boxed{\int_0^t x_1(\tau)\, d\tau} + \frac{1}{\bar{m}_m} \int_0^t (k_f f_e^d(\tau) - f_h(\tau))\, d\tau \end{aligned} \tag{3.27}$$

Again, we have solved the problem associated to the acceleration term but we haven't expressed σ as a function of the state due to the boxed integral of x_1. Before continuing, we have to notice that the following statement holds:

Theorem 3.1. *(Sliding mode enforcement) If $\sigma = 0$ $\forall t > t_f$, then $I = 0$ and we are correctly imposing our desired dynamics*

Proof of theorem 3.1 is trivial if we look at equation (3.26) ($\int I = 0 \Rightarrow I = 0$). Moreover, theorem 3.1 is only a *sufficient condition*, since the opposite assertion is not true ($I = 0 \not\Rightarrow \sigma = 0$).

Thanks to transformation (3.26), we have "fictitiously" reduced the sliding surface order by adding a new state - space variable ($\int x_1$). Thus, to express $\sigma(\boldsymbol{x})$, it's necessary to *expand* the master state - space model, adding a new equation which defines $\int x_1$:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -\frac{b_m}{m_m} x_3 + \frac{1}{m_m}(f_h + f_m) \end{aligned} \tag{3.28}$$

where the new state vector is $\boldsymbol{x} = [\int x_m \;\; x_m \;\; \dot{x}_m]^\top$ and the first equation represents the additional integrator which relates the new state variable with the remaining model. Substituting the new state - space formulation (3.28) inside the expression

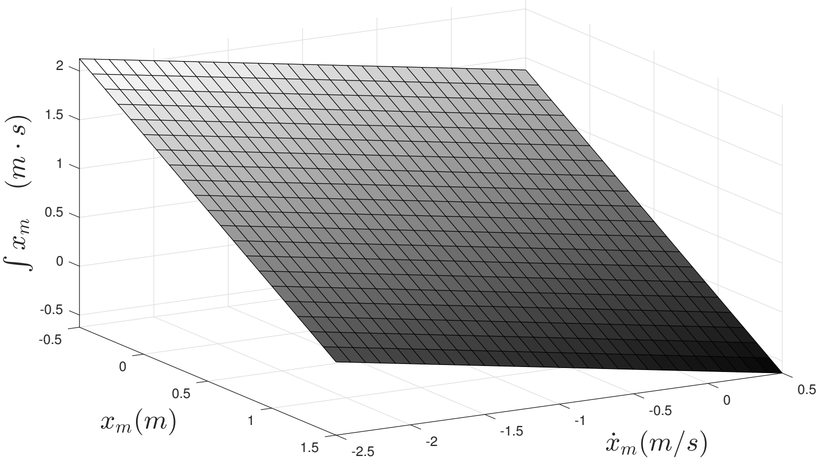


Figure 3.7: Representation of the sliding manifold in master device phase space diagram

of σ (3.27), we obtain:

$$\begin{aligned}\sigma &= x_3 + \frac{\bar{b}_m}{\bar{m}_m}x_2 + \frac{\bar{k}_m}{\bar{m}_m}x_1 + \frac{1}{\bar{m}_m}\int_0^t \left(k_f f_e^d(\tau) - f_h(\tau)\right) d\tau \\ &= \boldsymbol{G}\boldsymbol{x} + \Gamma\end{aligned} \tag{3.29}$$

where $\boldsymbol{G} = \begin{bmatrix} \frac{\bar{k}_m}{\bar{m}_m} & \frac{\bar{b}_m}{\bar{m}_m} & 1 \end{bmatrix}$ is the row vector of sliding surface coefficients, while $\Gamma = \int_0^t (k_f f_e^d(\tau) - f_h(\tau))\, d\tau$ contains all the external signals which affect σ. As we can see, now σ can be expressed as a function of the system's state. Moreover, it's a *linear combination* of $\boldsymbol{x}$, which is a favourable condition because it allows an easy representation of the manifold [76].

Since we have extended the state vector of master model, now we can no more represent state evolutions using a phase plane diagram. Thus, we need to adopt a *phase space* environment, in which the three coordinates stand for the three components of vector $\boldsymbol{x}$. If we assume no external forces acting on the system ($\Gamma = 0$), $\sigma = 0$ can be represented as a plane in the master device phase space (Figure 3.7). When the controlled system is in sliding mode, its trajectory lays on the plane, moving according to the desired parameters that define it.

Now that we have our sliding surface, we can apply the equivalent control definition and compute v^{eq}. First of all, we need $\dot{\sigma}$:

$$\dot{\sigma} = \dot{x}_3 + \frac{\bar{b}_m}{\bar{m}_m}\dot{x}_2 + \frac{\bar{k}_m}{\bar{m}_m}\dot{x}_1 + \frac{1}{\bar{m}_m}(k_f f_e^d - f_h) \tag{3.30}$$

Since we are applying a *first order* SMC, the relative degree between the sliding surface and the controlled system is one (degree of $\sigma = 2$ and degree of modified master device = 3). Therefore, $\dot{\sigma}$ will have the same order of the initial mechanical system, allowing to substitute the master model after dynamic inversion (equation (3.21)).

$$\dot{\sigma} = \eta_m(\dot{x}_m) + \zeta_m \boxed{v_m} + \frac{\bar{b}_m}{\bar{m}_m}\dot{x}_m + \frac{\bar{k}_m}{\bar{m}_m}x_m + \frac{1}{\bar{m}_m}(k_f f_e^d - f_h) \tag{3.31}$$

where we have already replaced state vector items with their physical counterparts. The boxed term represents the new control variable, that we need to make explicit in order to compute the equivalent control law.

If we are in sliding mode, $\sigma = 0$. So, we must also have $\dot{\sigma} = 0$ to remain on the sliding surface. This leads to the following equality:

$$\begin{aligned}
&\eta_m(\dot{x}_m) + \zeta_m v_m + \frac{\bar{b}_m}{\bar{m}_m}\dot{x}_m + \frac{\bar{k}_m}{\bar{m}_m}x_m + \frac{1}{\bar{m}_m}(k_f f_e^d - f_h) = 0 \\
\rightarrow v_m^{eq} = &-\frac{1}{\zeta_m}\left[\eta_m(\dot{x}_m) + \frac{\bar{b}_m}{\bar{m}_m}\dot{x}_m + \frac{\bar{k}_m}{\bar{m}_m}x_m + \frac{1}{\bar{m}_m}(k_f f_e^d - f_h)\right]
\end{aligned} \tag{3.32}$$

where v_m^{eq} is the equivalent control, i.e. the control signal that would be applied by SMC once the sliding surface is reached. As we can see, it intercepts all the model mismatches generated by non perfect dynamic cancellation, obtaining the desired impedance model.

If we could apply v_m^{eq} as it is written in equation (3.32), we wouldn't need to use SMC any more. Unfortunately, both ζ_m and $\eta_m(\dot{x}_m)$ are *unknown* and *unpredictable*, therefore equation (3.32) is not feasible. The best we can do is to choose the equivalent control as if it were be applied to the *nominal* plant, neglecting all terms which are not at our disposal:

$$v_m^{eq} \approx -\frac{\bar{b}_m}{\bar{m}_m}\dot{x}_m - \frac{\bar{k}_m}{\bar{m}_m}x_m - \frac{1}{\bar{m}_m}(k_f f_e^d - f_h) \tag{3.33}$$

If we compare equation (3.33) with the impedance controller expression (3.7), we can notice that they are analogous. This is reasonable because, using a standard linear controller, the best we can do to impose a given dynamics is represented precisely by the impedance control.

3.3.3 Sliding mode slave device

Similarly to what we have done for the master device, also for the slave manipulator we can follow the same approach highlighted in subsection 3.3.1. First of all, let's define the sliding surface based on the desired dynamics (3.10):

$$I = \bar{m}_s \ddot{\tilde{x}}_s + \bar{b}_s \dot{\tilde{x}}_s + \bar{k}_s \tilde{x}_s + f_e \tag{3.34}$$

Again, to eliminate the dependency from the acceleration term, use transformation (3.26) to write the feasible sliding surface:

$$\sigma = \dot{\tilde{x}}_s + \frac{\bar{b}_s}{\bar{m}_s}\tilde{x}_s + \frac{\bar{k}_m}{\bar{m}_m}\int_0^t \tilde{x}_s(\tau)\,d\tau + \frac{1}{\bar{m}_m}\int_0^t f_e(\tau)\,d\tau \tag{3.35}$$

By extending the dynamic state - space model for the slave, it's possible to rewrite expression (3.35) as a linear combination of the tracking error vector $\boldsymbol{e} = \boldsymbol{z} - k_p \boldsymbol{x}^d$.

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ \dot{z}_3 &= -\frac{b_s}{m_s} z_3 + \frac{1}{m_s}(f_s - f_e) \end{aligned} \tag{3.36}$$

$$\begin{aligned} \sigma &= e_3 + \frac{\bar{b}_s}{\bar{m}_s} e_2 + \frac{\bar{k}_s}{\bar{m}_s} e_1 + \frac{1}{\bar{m}_s}\int_0^t f_e(\tau)\,d\tau \\ &= \boldsymbol{F}\boldsymbol{e} + \Psi \end{aligned} \tag{3.37}$$

This time, the plane which describes $\sigma = 0$ doesn't belong to the phase space of the slave manipulator, but it can be represented in the tracking error phase space $\langle \dot{\tilde{x}}_s, \tilde{x}_s, \int \tilde{x}_s \rangle$.

For what concerns the equivalent control law, we can compute $\dot{\sigma}$ and equate it to zero in order to find v_s^{eq}:

$$\begin{aligned}
&\dot{\sigma} = \eta_s(\dot{x}_s) + \zeta_s v_s - k_p \ddot{x}_m^d + \frac{\bar{b}_s}{\bar{m}_s}\dot{\tilde{x}}_s + \frac{\bar{k}_s}{\bar{m}_s}\tilde{x}_s + \frac{f_e}{\bar{m}_s} = 0 \\
&\rightarrow v_s^{eq} = -\frac{1}{\zeta_s}\left[\eta_s(\dot{x}_s) + \frac{\bar{b}_s}{\bar{m}_s}\dot{\tilde{x}}_s + \frac{\bar{k}_s}{\bar{m}_s}\tilde{x}_s + \frac{f_e}{\bar{m}_s} - k_p \ddot{x}_m^d\right]
\end{aligned} \tag{3.38}$$

In the same way for the master device, we need to remove the unknown contributes ζ_s and $\eta_s(\dot{x}_s)$ in order to obtain a feasible control signal for the slave manipulator. Resulting v_s^{eq} has the following form:

$$v_s^{eq} \approx -\frac{\bar{b}_s}{\bar{m}_s}\dot{\tilde{x}}_s - \frac{\bar{k}_s}{\bar{m}_s}\tilde{x}_s - \frac{f_e}{\bar{m}_s} + k_p \ddot{x}_m^d \tag{3.39}$$

if we make $\ddot{x}_m^d$ explicit by substituting master controlled dynamics (3.12), we finally end up with the same expression of the ideal impedance controller (3.13).

3.3.4 Stability analysis

According to subsection 3.2.1, the last thing we have to examine is that the control law:

$$v_i = v_i^{eq} - K_i \operatorname{sgn} \sigma_i \qquad i = m, s \tag{3.40}$$

generates closed - loop dynamics that are *asymptotically stable*. A common procedure to check system stability in SMC [14, 10, 59] is to rely on *Lyapunov theory*. It is based on the definition of an arbitrary energy storage function $V(z)$ (Lyapunov function), related to system's state, which must have the following two properties:

1. $V(z)$ must be *positive definite*:

$$V(z) \geq 0 \quad \forall z \quad \wedge \quad V(z) = 0 \Leftrightarrow z = 0$$

2. $\dot{V}(z)$ must be *negative definite*:

$$\dot{V}(z) \leq 0 \quad \forall z \quad \wedge \quad \dot{V}(z) = 0 \Leftrightarrow z = 0$$

If conditions 1 and 2 are satisfied, equilibrium point $z = 0$ is globally asymptotically stable.

We already know that once the controlled system has reached the sliding mode, it has an equilibrium point ($\boldsymbol{x} = \mathbf{0}$ or $\boldsymbol{e} = \mathbf{0}$) which is asymptotically stable. This is a direct consequence of the desired dynamics choice and can be verified easily by computing the poles of equations (3.6) and (3.10).

Instead, what we need to verify is the enforcement of a sliding mode for every initial condition of our dynamic system. In other words, we need to prove that $\sigma = 0$ is a global attractor for the system.

A good Lyapunov function candidate is represented by the quadratic expression:

$$V(\sigma) = \frac{1}{2}\sigma^2 \tag{3.41}$$

Choosing $V(\sigma)$ as in (3.41), condition 1 of the theorem is verified, while we need to check the second one. To do so, let's compute firstly $\dot{V}(\sigma)$:

$$\dot{V}(\sigma) = \sigma\dot{\sigma} \tag{3.42}$$

now substitute the expression of $\dot{\sigma}$ (3.31) and rewrite it as a function of σ and system's parameters. For simplicity, let's consider only the master case (slave device demonstration is analogous):

$$\dot{V}(\sigma_m) = \sigma_m \left[\eta_m(\dot{x}_m) + \zeta_m v_m + \frac{\bar{b}_m}{\bar{m}_m}\dot{x}_m + \frac{\bar{k}_m}{\bar{m}_m}x_m + \frac{1}{\bar{m}_m}(k_f f_e^d - f_h)\right] \tag{3.43}$$

the last terms of (3.43) correspond to the nominal equivalent control of the master device retrieved in (3.33), that can be substituted:

$$\begin{aligned}
\dot{V}(\sigma_m) &= \sigma_m \left[\eta_m(\dot{x}_m) + \zeta_m v_m - v_m^{eq}\right] \\
&= \sigma_m \left[\eta_m(\dot{x}_m) + \zeta_m \left(v_m^{eq} - K_m \operatorname{sgn}\sigma_m\right) - v_m^{eq}\right] \\
&= \sigma_m \left[\eta_m(\dot{x}_m) + (\zeta_m - 1)v_m^{eq} - \zeta_m K_m \operatorname{sgn}\sigma_m\right]
\end{aligned} \tag{3.44}$$

Now, by imposing condition 2 to expression (3.44), we eventually find the following inequality:

$$|\sigma_m| \left\{ \left[\eta_m(\dot{x}_m) + (\zeta_m - 1) v_m^{eq} \right] \operatorname{sgn} \sigma - \zeta_m K_m \right\} < 0 \tag{3.45}$$

where we have applied signum properties $\sigma = |\sigma| \operatorname{sgn} \sigma$ and $|\sigma| = \sigma \operatorname{sgn} \sigma$. Since sliding mode is usually introduced to deal with systems affected by uncertainty, condition on K_m which can be derived from (3.45) is not sufficiently strict to guarantee asymptotic stability despite disturbances. That's why Utkin at al. [77, 14] advise to impose a *stronger* condition on $\dot{V}(\sigma)$, known as $\mu -$ reachability. It allows us to ensure the sliding mode attraction region with an assignable degree of robustness. Based on this new definition, theorem's condition 2 is rewritten as:

$$\dot{V}(z) \leq -\mu|z| \quad \forall z \quad \wedge \quad \dot{V}(z) = 0 \Leftrightarrow z = 0 \tag{3.46}$$

where μ is a positive parameter which defines the degree of robustness. Higher is its value, stricter becomes the inequality.

If we perform the same computations shown in (3.44) with the new definition (3.46), we obtain:

$$\begin{gathered} |\sigma_m| \left\{ \left[\eta_m(\dot{x}_m) + (\zeta_m - 1) v_m^{eq} \right] \operatorname{sgn} \sigma - \zeta_m K_m \right\} < -\mu|\sigma_m| \\ \rightarrow K_m > \frac{1}{\zeta_m} \left[\mu + \eta_m(\dot{x}_m) + (\zeta_m - 1) v_m^{eq} \right] \end{gathered} \tag{3.47}$$

From inequality (3.47), we can state the following important theorem:

Theorem 3.2. *(Sliding mode stability) Let's consider system (3.21) at which control law (3.40) is applied. If the uncertainty affecting the system is bounded ($|\eta_m(\dot{x}_m)| \leq \eta_m^{\max}$, $\zeta_m^{\min} \leq \zeta_m \leq \zeta_m^{\max}$, $|v_m^{eq}| \leq v_m^{\max}$), then we can choose a value of K_m:*

$$K_m > \frac{1}{\zeta_m^{\min}} \left[\mu + \eta_m^{\max} + (\zeta_m^{\max} - 1) v_m^{\max} \right] \tag{3.48}$$

which guarantees that the closed loop system is asymptotically stable for every initial condition $\boldsymbol{x}$.

The proof of this theorem is based on the observation that, if we choose K_m

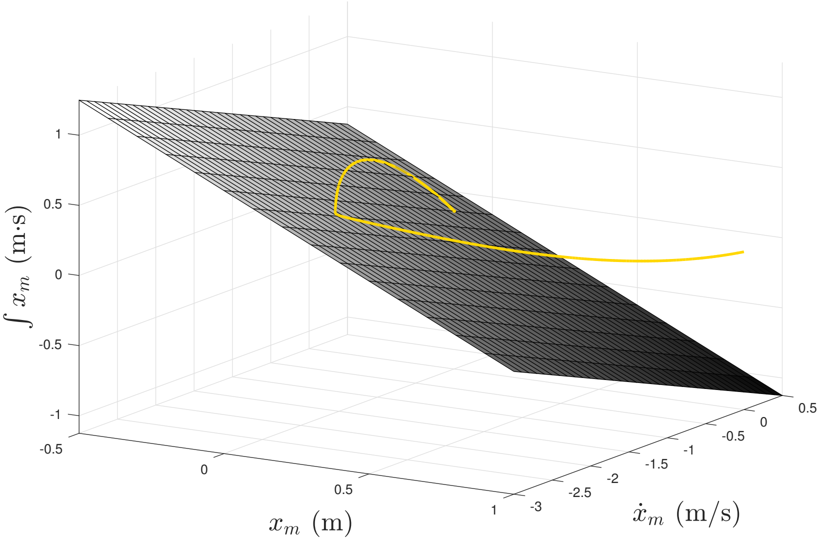


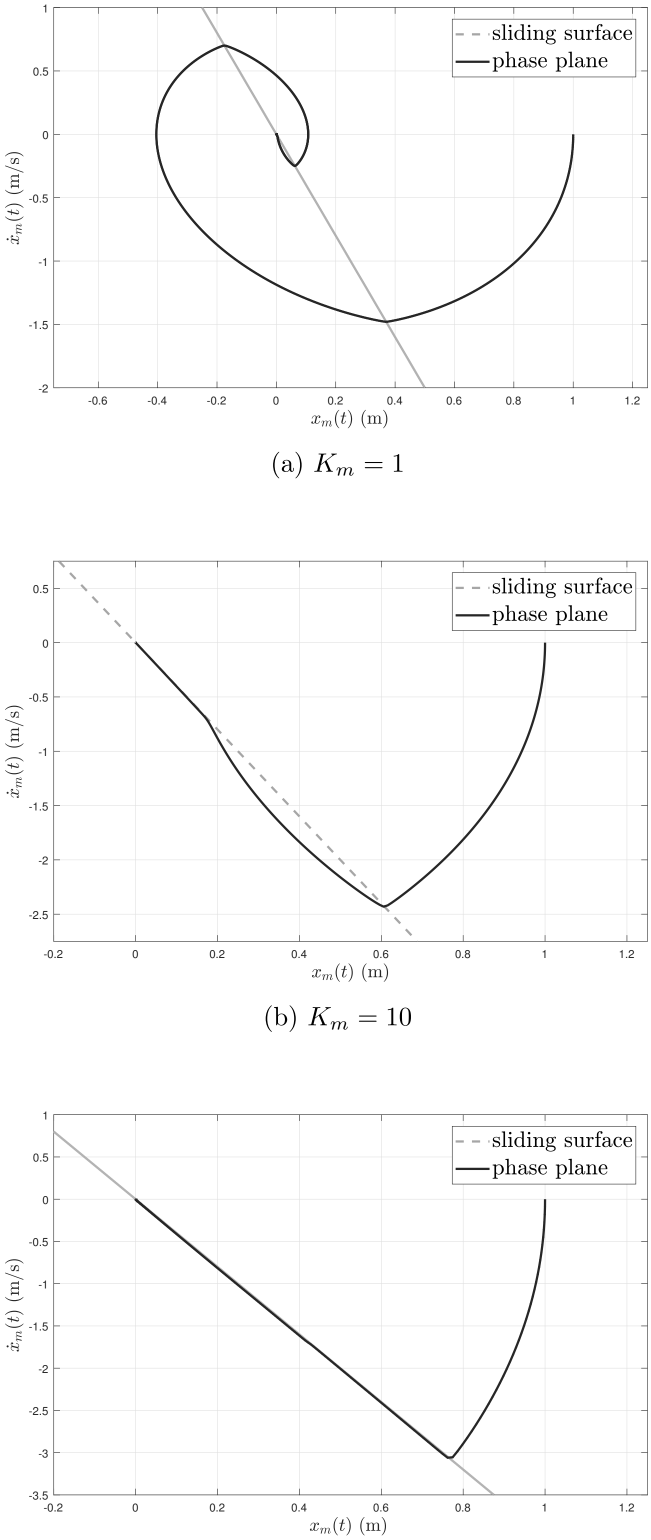
Figure 3.8: Convergence of the expanded state trajectory onto the sliding plane

higher enough to overcome the maximum uncertainty affecting the system, Lyapunov stability holds and we are ensuring the attractiveness of the sliding manifold.

It is important to notice that all hypothesis done to derive this statement are reasonable. We need to know only the upper and lower bounds of the disturbances, while condition on the equivalent control is usually verified thanks to the structure of v^{eq} (linear controller with stable closed - loop poles generates a *bounded* control signal). A typical example of sliding mode convergence can be seen in Figure 3.8, where system's trajectory (yellow line), after a finite transient, remains on the sliding plane plotted in the expanded state phase space, until the origin (equilibrium) is reached.

Observation 3.1. *If condition* (3.48) *is not satisfied, the sliding mode is not necessarily enforced in the controlled system, but the closed - loop state evolution may be, however, asymptotically stable.*

Theorem 3.2 guarantees the convergence of system trajectories onto the sliding surface and, thanks to the properties of σ, it ultimately assures that the origin is an asymptotically stable equilibrium. Nevertheless, observation 3.1 remarks that Lyapunov theorem is only a *sufficient condition* and we can have closed - loop trajectories which converges into the origin although no or partial sliding mode is

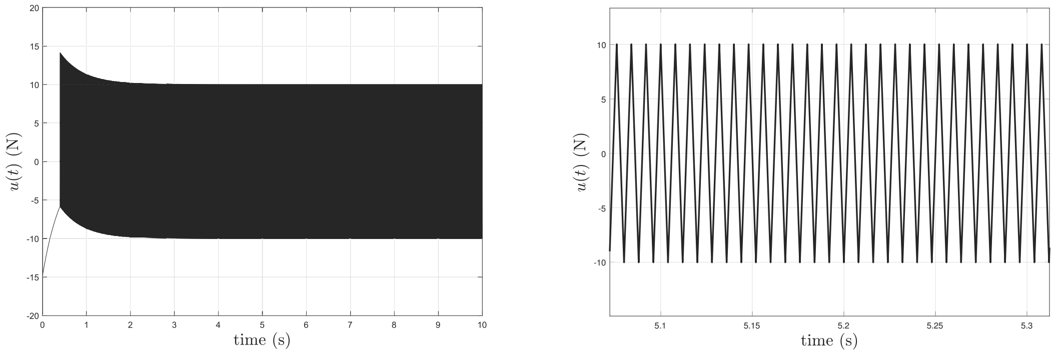


(a) $K_m = 1$

(b) $K_m = 10$

(c) $K_m = 30$ *(satisfies theorem 3.2)*

Figure 3.9: Three examples of asymptotically stable control systems



(a) *control signal*

(b) *chattering detail*

Figure 3.10: Control signal for a standard SMC controller. We can notice in the left plot high frequency oscillation due to chattering

generated. To clarify this fact, an illustrative example is reported in Figure 3.9.

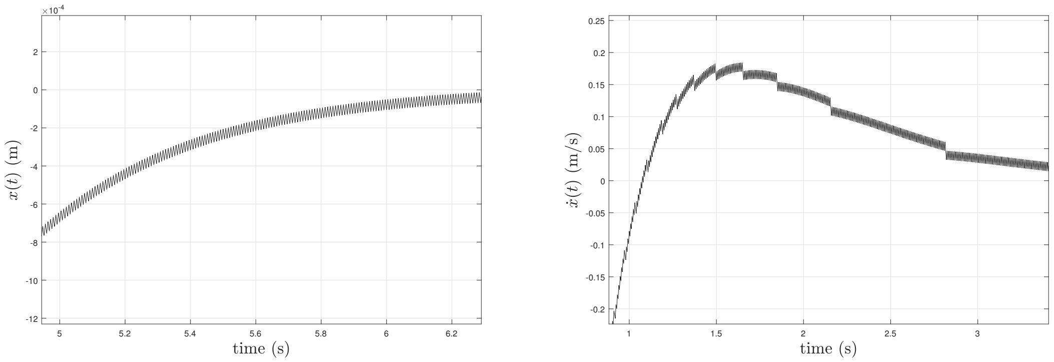
3.3.5 Sliding mode drawbacks

Until now, we have praised the positive properties of sliding mode control, both in terms of disturbance rejection and enhanced system robustness. Unfortunately, this favourable behaviour comes at the price of some *undesired phenomena*, which occurs mainly when we reach the sliding mode operative condition.

Excessive control signal First of all, we have to notice that the total control signal (3.40) is proportional to the parameter K associated to the SMC control contribute. In the previous subsection, we derived an inequality which constrains to choose a value of K sufficiently high in order to guarantee the sliding mode to happen. If the uncertainty affecting the process is significant, the sliding gain could become very large, overcoming the equivalent control signal and saturating actuators.

The only way to overcome this problem is to acquire more information about system's dynamics, in order to reduce uncertainty and bring K boundary inside the feasibility region.

Chattering Another, more subtle, phenomenon is intrinsic to the nature of SMC and it's called *chattering*. According to [78], it corresponds to undesired stationary



(a) *position response*

(b) *speed response*

Figure 3.11: Detail of system position and velocity trajectories when chattering on the control signal occurs

oscillations, with finite frequency and amplitude, which affects the control signal as well as the other system variables. Principal cause of chattering phenomena is the presence of a discontinuity in SMC control action.

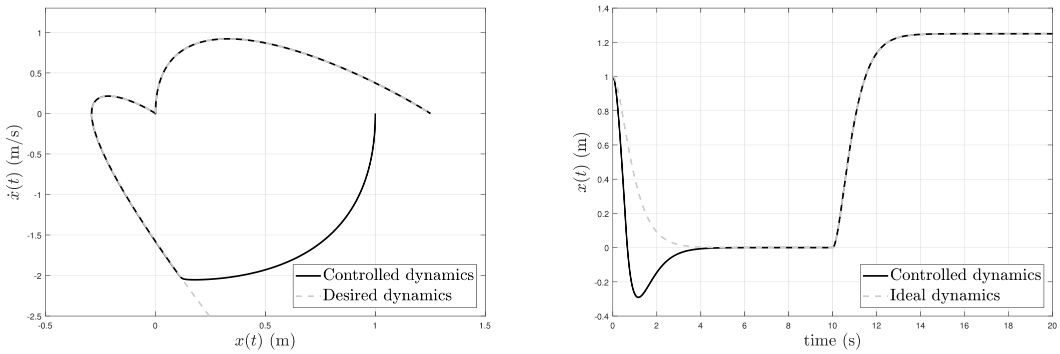
When we approach condition $\sigma = 0$, the discontinuous control action starts to change sign each time we cross the sliding manifold. Ideally, this should happen at *infinite frequency*, ensuring the permanence of the phase portrait onto the sliding curve. However, for practical implementations, ideal switching is *not* achievable:

- Actuators and sensors have finite bandwidths and their unmodelled dynamics becomes relevant when we require an infinitely fast switching;

- Control logic is usually implemented on a digital platform, which works with a finite sample rate. Discretization of input and output signals limit the maximum switching frequency of our controller;

We can see the effect of discretization on standard SMC by looking at Figure 3.10. In this case, we are using a simulation step of 4 ms, which constrains the maximum switching frequency to be below 250 Hz. Once sliding mode is enforced, the signum function starts to change sign at each iteration, making the control variable oscillate between $\pm K$ at the maximum available frequency.

Chattering phenomenon is disruptive for mechanical mechanisms [72], because it causes high wear of moving parts (actuators), and high heat losses in power circuits. Moreover, it results in a poor tracking accuracy, as it can be noticed in Figure 3.11.



(a) *phase portrait*

(b) *time response*

Figure 3.12: Uncontrolled reaching phase can generate very different initial transient response, although once sliding is enforced controlled system behaves as the desired one

In section 3.4 and 3.5 some techniques will be presented in order to eliminate or, at least, attenuate chattering phenomena in sliding mode control.

Uncontrolled reaching phase The last problem that affects standard sliding mode is related to the transient phase required by the system before reaching sliding mode condition. We know that the desired impedance dynamics is valid only during the sliding phase, which is reached in finite time t_f thanks to theorem 3.2. Until that time, the control scheme has *no authority* on the system behaviour and the only assertion we can make is that its trajectory is becoming closer to $\sigma = 0$.

Since we are designing a robust control law, we would be glad to enforce the desired dynamic behaviour from the *initial time* t_0. Doing so, we are able to guarantee that the system is behaving in a controlled and predictable way during its entire state evolution.

An example of how different can be the reaching dynamics from the desired one is given in Figure 3.12. As we can see, after the initial mismatch, tracking of the desired dynamics is excellent once sliding mode is reached.

With respect to the previous two caveats, this is easier to solve and a solution will be detailed in section 3.7.

3.4 SMC continuous approximations

In many practical applications, like robotic servomechanisms, it is necessary to attenuate chattering in order to guarantee precise positioning and decreased wear of reduction stages. To do so, the control signal imposed by the SMC *should be smoothed*, while retaining all the desirable properties of disturbance rejection and uncertainty insensitivity.

As said in subsection 3.3.5, the main source of chattering is the discontinuity of signum function when the system state crosses the sliding manifold. As reported in [72, 7], the most obvious and immediate way to overcome signum nonlinearity consists in substituting it with a *continuous function approximation* $s(\sigma)$. It should mimic the behaviour of $\operatorname{sgn}\sigma$ except for a neighbour of the origin, where the two extremities are blended to obtain a smooth transient.

To this end, most common choices are:

- **Saturation function** [10]:

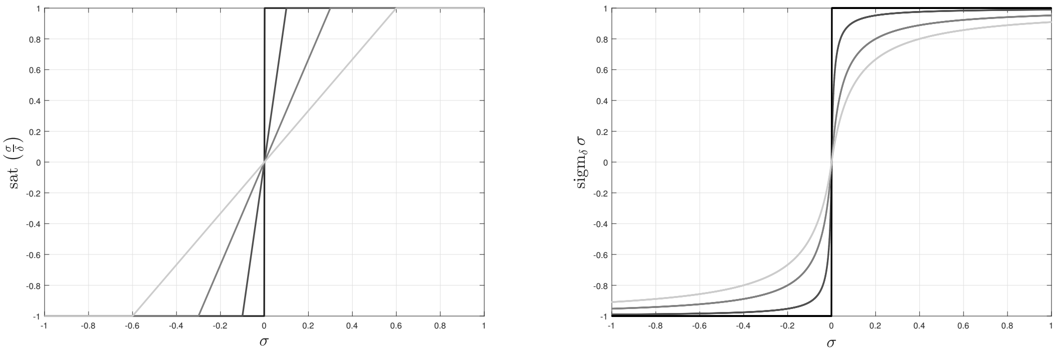
$$s(\sigma) = \operatorname{sat}\left(\frac{\sigma}{\delta}\right) \tag{3.49}$$

- **Sigmoid function** [72]:

$$s(\sigma) = \operatorname{sigm}_\delta \sigma = \frac{\sigma}{|\sigma| + \delta} \tag{3.50}$$

where δ is a constant positive parameter which can be used to increase or decrease the approximation degree of $s(\sigma)$ with respect to the ideal signum. Higher is δ, larger is the linearised region (worse approximation) and vice versa. It represents also the inverse of the derivative of $s(\sigma)$ calculated in the origin. An example of equations (3.49) and (3.50) can be seen in Figure 3.13, for different values of parameter δ.

The main advantage of this approach consists in its simplicity, since we have to change only the discontinuous part of our controller. Moreover, it's easy to prove that, with $\delta \to 0$, $s(\sigma) \to \operatorname{sgn}\sigma$ and we recover standard sliding mode control. The



(a) *saturation function*

(b) *sigmoid function*

Figure 3.13: Graphical representation of the two most common continuous approximations for the signum function, plotted with various δ

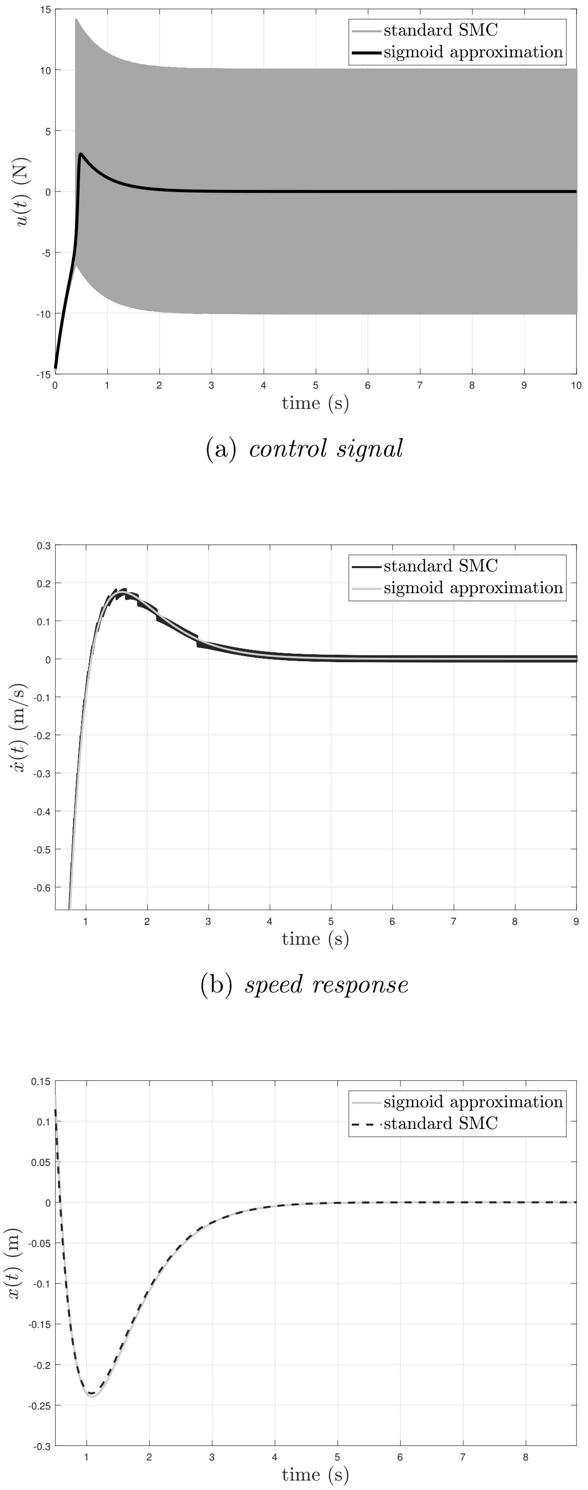
additional parameter can be seen as a new tuning knob, useful to obtain a trade off between low chattering response and acceptable control error.

Unfortunately, introduction of $s(\sigma)$ causes the loss of the main properties and theorems enunciated in section 3.3. In particular, it is not valid any more that the controlled system reaches $\sigma = 0$ in finite time, loosing the robustness of standard SMC. It can be proved [72] that we are able to remain *in the vicinity* of σ, without actually converging to it. That is why we usually refer to these control systems as *quasi - sliding mode controllers.*

In many applications, this lack of theoretical correctness is compensated by the great improvement of control signal's dynamics. Moreover, by choosing a value of δ "sufficiently small", convergence performance of approximated methods are comparable to the ideal sliding mode. Typically, the sigmoid approximation is preferred, because it is continuous also in its derivative and allows a smoother control signal with respect to saturation. A visual comparison of sigmoid benefits can be observed in Figure 3.14.

3.5 Higher order sliding mode control

When dealing with systems affected by chattering, continuous approximations do not represent the unique solutions that allow us to overcome the problem. During the last decades, several studies have been performed in order to *generalize* the main



(a) *control signal*

(b) *speed response*

(c) *position response*

Figure 3.14: Chattering removal effect of quasi - sliding control techniques compared to standard SMC

concepts of standard SMC and realize a more complete theory on VSCS [47, 46, 5]. Ultimately, this research leads to the introduction of *higher order* sliding mode controllers (Higher Order Sliding Mode Controller (HOSMC)). According to [47], they keep the main advantages of standard SMC, removing the chattering effect and providing even higher tracking accuracy.

Before starting with these new control techniques, a brief introduction on the main symbols and definitions is required.

Definition 3.1. The *sliding order* ρ is the total number of continuous derivatives of σ that cancel during sliding mode operative condition

$$\sigma = \dot{\sigma} = \ddot{\sigma} = \cdots = \sigma^{(\rho-1)} = 0 \tag{3.51}$$

Definition 3.2. The *relative degree* r of a Single Input Single Output (SISO) system is the minimum order of the time derivative of σ in which the control signal v explicitly appears

$$\forall i \geq r \qquad \sigma^{(i)} = \sigma^{(i)}(v) \tag{3.52}$$

According to these definitions, standard SMC has a sliding order equal to one (only $\sigma = 0$) and also its relative degree is equal to one ($\dot{\sigma}$ is a function of the control input v, see (3.30)).

Due to the property of order reduction, the sliding manifold must have a lower dynamic order with respect to the system it is applied to. This can be translated in a condition on ρ and r. Especially, the following inequality is verified:

$$\rho \geq r \tag{3.53}$$

For standard SMC, condition (3.53) is always valid, since ρ by definition cannot be less than one. This is no more true when we have to control a system using a sliding manifold with relative degree greater than one. For example, if we remove the dependency on speed from equation (3.27), we need to derive σ twice before v appears. In this case, we *cannot design* a first order sliding mode and we are constrained to use a value of $\rho \geq 2$.

By sticking to our case ($r = 1$), higher order sliding modes can be realized by imposing not only $\sigma = 0$, but also its derivatives up to $\rho - 1$. Therefore, $\rho \geq r + 1$ and the new control variable for the system is no more v but $v^{(\rho-1)}$. The advantage is that the discontinuity now is confined to the time derivative of v, while the *true control input* has continuous derivatives up to order $\rho - r \geq 1$. This smoothness condition proves that HOSMC attenuates chattering, guaranteeing at the same time fulfilment of the sliding requirements on σ (disturbance rejection, etc.).

Higher order methods are a formal extension of standard first order sliding mode, therefore they inherit all the theorems we have derived in section 3.3. This represents their superiority with respect to continuous approximation, which was a “practical” solution based on empirical reasonings. Enhanced control capabilities come at the price of an increased complexity in the control law definition and a greater number of information on the system state.

3.5.1 Second order SMC

Among all the various higher order control techniques, the most simple to implement are represented by Second Order Sliding Mode Controller (SOSMC). They can be seen as a good trade off between computational complexity in control law derivation and reasonable chattering attenuation, thanks to the continuity of v. Condition (3.53) specializes in this case, becoming the equality $\rho = r + 1$. Thus, in order to obtain a second order sliding movement on σ, we need to choose a value of ρ equals to 2, which ultimately amounts to impose $\sigma = \dot{\sigma} = 0$.

Considering the generalised uncertain SISO system (valid both for master and slave models)

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \eta(\boldsymbol{x}) + \zeta(\boldsymbol{x})v \end{aligned} \tag{3.54}$$

design procedure for SOSMC pursues these main steps:

1. Definition of a new **auxiliary system** associated to the original one (3.54). It is constituted by a chain of integrators and the new state variables are the sliding surface σ and its derivative. If we define vector $\boldsymbol{z} = [\sigma \;\; \dot{\sigma}]^\top$ and $w = \dot{v}$,

the auxiliary system can be written as:

$$\begin{aligned}\dot{z}_1 &= z_2 \\ \dot{z}_2 &= H(\boldsymbol{z}) + p(\boldsymbol{z})w\end{aligned} \tag{3.55}$$

where $H(\boldsymbol{z})$ and $d(\boldsymbol{z})$ can be easily derived by computing $\ddot{\sigma}$ and collecting all the terms associated to $\dot{v}$;

2. Using standard control techniques, find a value for the new control signal w that stabilizes the equilibrium point $[0\ \ 0]^\top$. If we are able to do so, we are guaranteeing a second order sliding enforcement for our initial SISO system;

3. Based on the definition of w, apply the following control input to the original plant:

$$v = \int w\,dt \tag{3.56}$$

Observation 3.2. *To solve point* 2 *of SOSMC design steps, we cannot use standard SMC procedures. This is due to the fact that state* $\boldsymbol{z}$ *is not fully known, because* $\dot{\sigma}$ *generally depends on uncertain quantities. Therefore, difficulty of SOSMC lays in finding a control law* w *able to steer state* $\boldsymbol{z}$ *to zero with little or no knowledge of* $\dot{\sigma}$.

Based on observation 3.2, different strategies have been elaborated to tackle the design of an effective second order sliding mode algorithm. Here, we limit our dissertation to two of the most relevant implementations of SOSMC:

Suboptimal controller [5, 61] In order to guarantee convergence of the auxiliary system state to the origin, we need a switching logic both on σ and on $\dot{\sigma}$. Moreover, to obtain an asymptotically stable system, it is necessary that the distance between the equilibrium point $(0, 0)$ and the system state *reduces* at each crossing of $\sigma = \dot{\sigma} = 0$. According to [61], this condition can be verified by imposing the following inequality (*extended* μ *- reachability*):

$$\ddot{\sigma}\sigma \leq -\xi(t)|\sigma| \qquad \xi(t) \geq \xi_1 > 0 \tag{3.57}$$

Without entering too much into details, we can simply say that inequality (3.57) generates a sequence of singular points $[z_1 \;\; z_2] = [\sigma_{Mi} \;\; 0] \quad i = 1, 2, \ldots,$ that converge towards zero if $\xi(t)$ satisfies [5]:

$$\begin{cases} \xi(t) \leq \xi^*(t) & (\sigma - \frac{1}{2}\sigma_{Mi}) > 0 \\ \xi(t) \geq \alpha\xi^*(t) & \text{otherwise} \end{cases}$$

where $\xi^*(t) \geq \xi_1 > 0$, $\alpha \geq 1$ and σ_{Mi} is the closer previous singular point.

As usual, if we can guarantee boundedness of uncertainty affecting the auxiliary system (3.55), at least in a neighbour of the origin:

$$\begin{aligned} |H(\boldsymbol{z})| &\leq H_0 \\ 0 \leq P_1 \leq p(\boldsymbol{z}) &\leq P_2 \end{aligned} \tag{3.58}$$

it can be proved that the system converges asymptotically towards $\sigma = \dot{\sigma} = 0$ by applying algorithm 1, with $U_{\max}$ that satisfies:

$$U_{\max} > \max\left(\frac{H_0}{\alpha^* P_1}, \frac{4H_0}{3P_1 - \alpha^* P_2}\right)$$

Algorithm 1 Suboptimal control algorithm

```
Require: $\alpha^* \in (0,1) \cap \left(0, \frac{3P_1}{P_2}\right)$
procedure SSOSMC($z_1(t)$)
    $z_{\max} \leftarrow z_1(0)$                                   ▷ initialization
    for $t > 0$ do                                                ▷ iterate at each sample
        if $\left[z_1(t) - \frac{1}{2}z_{\max}\right]\left[z_{\max} - z_1(t)\right] > 0$ then   ▷ switching logic
            $\alpha \leftarrow \alpha^*$
        else
            $\alpha \leftarrow 1$
        end if
        if $z_1(t)$ is an extremal point then                     ▷ $\dot{z}_1(t) = 0$
            $z_{\max} \leftarrow z_1(t)$                          ▷ update switching threshold
        end if
        $w(t) \leftarrow -\alpha U_{\max}\,\mathrm{sgn}\left(z_1(t) - \frac{1}{2}z_{\max}\right)$   ▷ control law
    end for
end procedure
```

As we can notice, the suboptimal controller does not require the full knowledge of $\dot{\sigma}$, which makes it a good candidate for our control problem. Unfortunately, we need

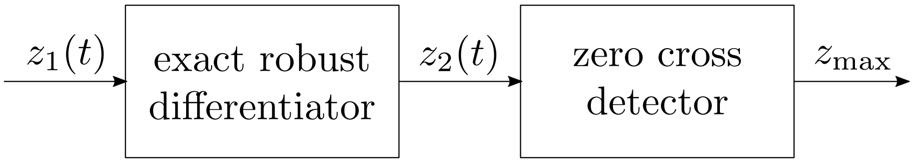


Figure 3.15: Numerical differentiator implemented in SSOSMC to find the extremal points of σ

to compute σ_{Mi}, which are the extremal points of σ. Therefore, we need to find when the derivative of sliding surface changes its sign in order to switch properly. Since $\dot{\sigma}$ is unknown, a possible solution is to rely on *numerical differentiation techniques* and zero crossing detectors according to the structure shown in Figure 3.15. To differentiate σ various strategies can be followed. Here, taking inspiration from [17], a robust differentiation technique which goes under the name of *Levant differentiator* has been adopted. A detailed analysis on its properties and a comparison with other differentiation techniques can be found in Appendix A. A visual representation of SSOSMC convergence is given in Figure 3.16a

Super twisting controller [47, 65] This class of higher order sliding mode algorithms represent an evolution of the early studies done by Levant [47], which were intended to define a contraction criterion independent from the knowledge of $\dot{\sigma}$. A first attempt to solve (3.57), called *twisting controller*, has been proposed in [49], but it still requires the knowledge of $\dot{\sigma}$ sign to perform the switching logic. Indeed, the contraction criterion can be written as [61]:

$$\begin{cases} \xi(t) \leq \xi^*(t) & \sigma\dot{\sigma} < 0 \\ \xi(t) \geq \alpha\xi^*(t) & \sigma\dot{\sigma} > 0 \end{cases}$$

where $\xi^*(t) \geq \xi_1 > 0$, $\alpha \geq 1$.

It seems that $\dot{\sigma}$ is fundamental for the evaluation of second order sliding modes, but this is not true. In fact, Levant proposed an evolution of the twisting algorithm, called *Super Twisting Algorithm (STA)*, which requires only the knowledge of σ. Its

	B_m	M_m	K_m
real parameters	2	1	0
estimated parameters	2.6	1.2	0
desired parameters	4	1	4

Table 3.1: Master device parameters used for the simulation comparison of chattering - avoidance methods

control law has the following structure:

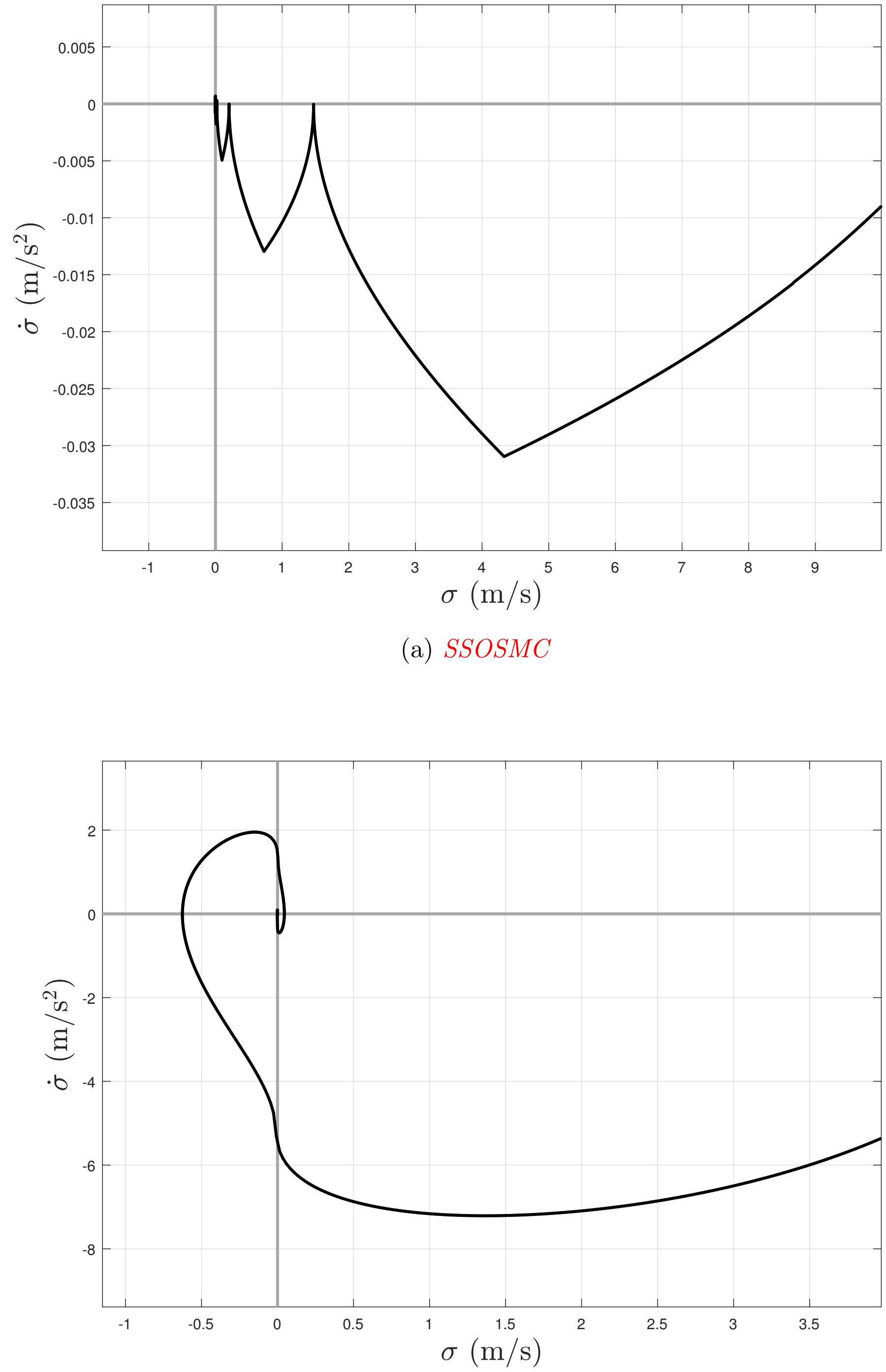
$$v = -k_1\sqrt{|z_1|}\,\operatorname{sgn} z_1 - k_2\int \operatorname{sgn} z_1 \tag{3.59}$$

where k_1 and k_2 are constant tunable parameters. Also in this case it's possible to derive uncertainty boundaries to guarantee finite time convergence of STA to the origin. However, this leads generally to an overestimation of k_1 and k_2 which ultimately turn out in a degradation of control performance. That is why it is better to tune these kinds of algorithms based on computer - numerical simulations. Trajectories generated by STA are similar to the ones of twisting algorithm, but smoothed thanks to the contribution of $\sqrt{|z_1|}$ (Figure 3.16b).

3.6 Simulation comparison

Now that we have presented all the main methodologies to overcome chattering problems, we need to compare them in order to find the one which offers the best compromise between tracking precision and control moderation. To do so, we use `MATLAB - Simulink`® environment to realize the master model (3.5) and standard SMC control law (3.40). To simulate the uncertainty in estimated parameters and disturbances acting on the system, the damping coefficient B_m is overestimated by 30%, while device mass is overestimated by 20% (Table 3.1). Although they are still in the stability region for standard impedance controller, this level of uncertainty severely degrade tracking performance of the desired impedance.

Simulation has been run for 20 s and a step of 5 N in the human force has been given at 10 s. In this way we avoid transient effects due to the reaching phase and



(a) *SSOSMC*

(b) *STA*

Figure 3.16: Auxiliary system trajectories for two different higher order sliding mode algorithms in the σ, $\dot{\sigma}$ plane. Changing contraction criteria generate peculiar evolutions

	Sigmoid SMC	STA	SSOSMC
RMS (m)	1.3534×10^{-4}	6.4577×10^{-5}	4.5434×10^{-4}

Table 3.2: RMS errors for the step tracking performance of the tested sliding mode control algorithm

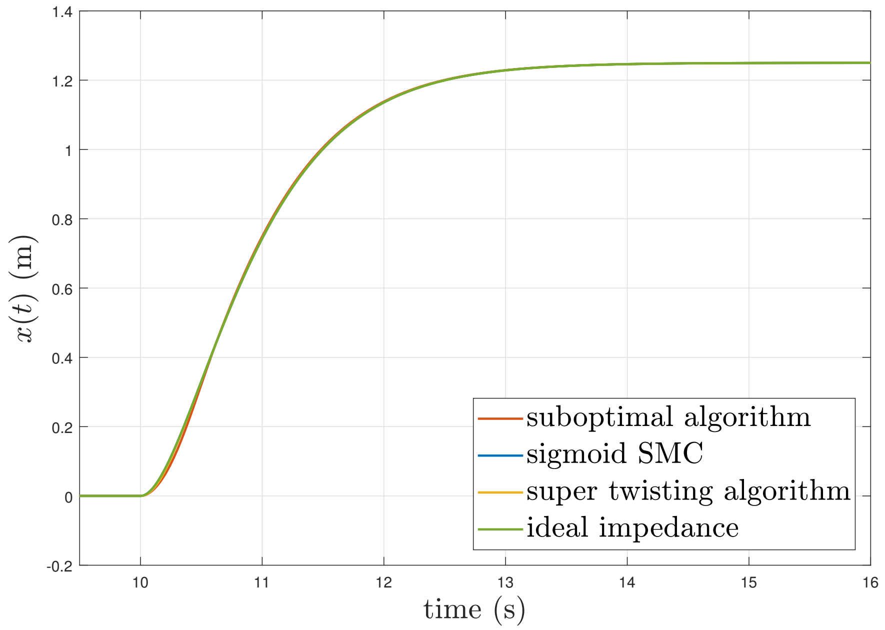
we can compare all the algorithms when they are already in sliding mode condition. Sampling time is equal to 4 ms. Ideally, the control action of SMC should be strong enough to keep system along the desired trajectory throughout the response. Instead, real sliding mode shows little discrepancies at the beginning and the end of the transient.

The position response profiles for the three tested algorithm is visible in Figure 3.17a, together with the ideal transient generated using a system tuned with the desired parameters. Since the four curves are not clearly distinguishable, an error evaluation is executed with respect to the desired dynamic response (Figure 3.17b).

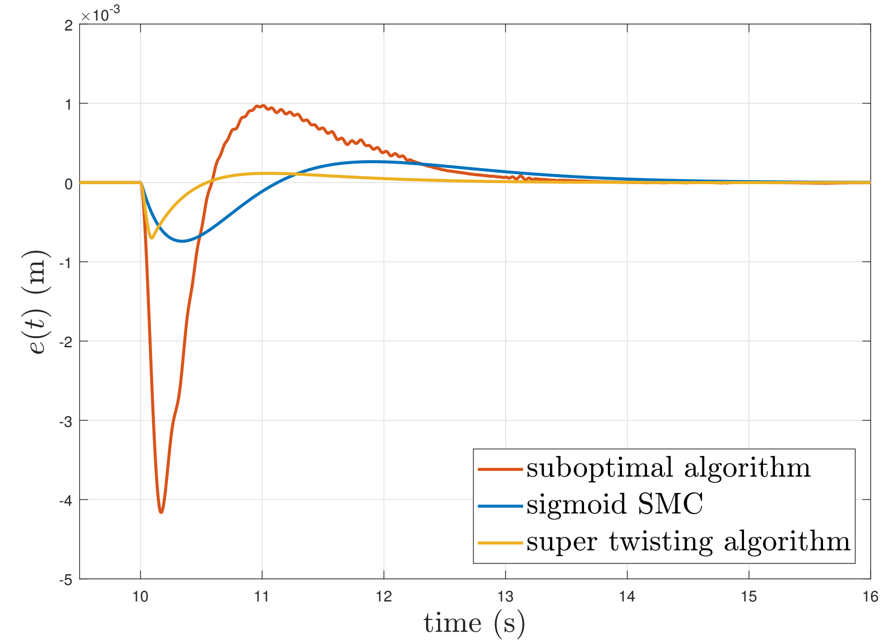
From these plots we can already draw some interesting conclusions:

- All chattering - avoidance methods correctly reject model uncertainties, tracking almost perfectly the reference dynamics;
- From the error evaluation chart, we can underline that the suboptimal control algorithm differs most from the ideal profile with respect to the sigmoid and super twisting ones;
- Settling time required for sigmoid SMC and suboptimal algorithm is longer (≈3 s) compared to the super twisting one (≈2 s);

For a more quantitative analysis, in Table 3.2 the values of RMS errors for all the tested algorithms are reported. As we have already noticed in our preliminary considerations, best performance is achieved by STA, with a tracking accuracy of 99.9935%. Up to now we have seen only one side of the coin. To verify that our modifications to standard SMC are effective, we need to check also the control effort required and the deviation with respect to the sliding surface $\sigma = 0$. These plots are visible, respectively, in Figure 3.18a and 3.18b.

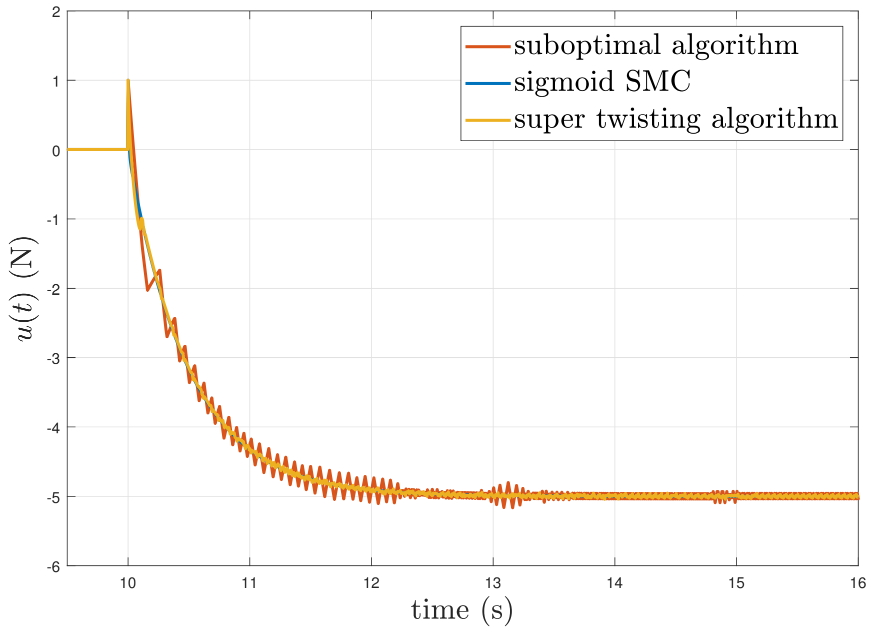


(a) *position response*

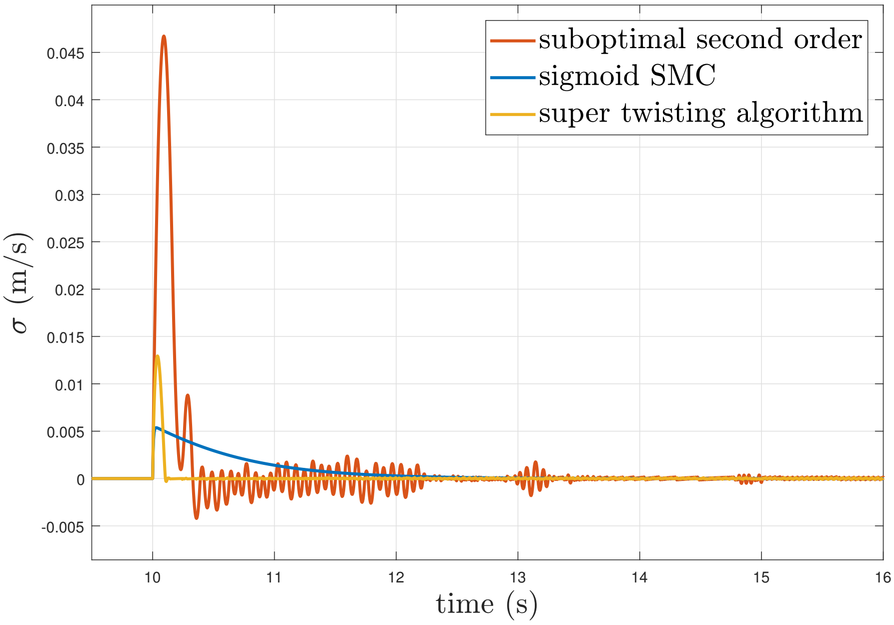


(b) *error with respect to the ideal trajectory*

Figure 3.17: Comparison between the various type of chattering - avoidance methods while performing an impedance tracking of a system affected by uncertainty. Top: position, bottom: error



(a) *control force*



(b) *sliding variable*

Figure 3.18: Comparison between the various type of chattering - avoidance methods while performing an impedance tracking of a system affected by uncertainty - control force and sliding variable

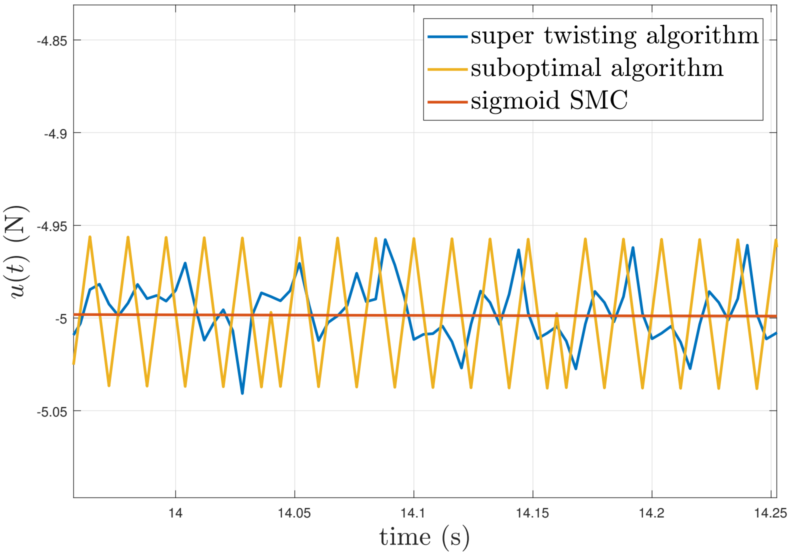


Figure 3.19: Detail of control effort for the three tested algorithms after transient phase

	Sigmoid SMC	STA	SSOSMC
standard deviation (N)	4.7595×10^{-5}	0.0151	0.0282

Table 3.3: Variability of control effort after transient phase

At first glance, we notice that all the methods strongly attenuate chattering with respect to standard SMC (Figure 3.10). Nevertheless, there are some residual ripples in the suboptimal algorithm, which are due to numerical differentiation of the sliding surface and strong discontinuity in the auxiliary control action $w(t)$. These oscillations around sliding manifold are totally absent for sigmoid approximation, while STA places in between these two cases. Chattering effect can be evaluated more clearly by looking at the control effort after the transient period (Figure 3.19). Standard deviation for the three signals is reported in Table 3.3.

Another important aspect that is worth remarking is the oscillation amplitude irregularity which affects the suboptimal algorithm, with bad sliding mode enforcement. On the contrary, STA, while having a discrete amount of residual ripples (Table 3.3), shows a much faster convergence towards $\sigma = 0$. From Figure 3.18b we can appreciate the quickness of super twisting to converge to zero, compared to both sigmoid and suboptimal.

This agrees with our theoretical results. Indeed, sigmoid guarantees only error boundedness, therefore it allows sliding detachment for longer periods. Different is the case of suboptimal algorithm, since its reaction time is as fast as STA but it experiences a greater error and, consequently, higher oscillations.

Eventually, the best chattering attenuation method is STA, which represents a good compromise between long settling times (sigmoid approximation) and too large dynamic error (suboptimal algorithm).

3.7 Integral sliding mode

The last obstacle we need to tackle is the presence of an uncontrolled reaching phase during the initial transient of standard SMC. This problem is well known in literature [30, 17, 79] and it is usually tackled through an adjustment of SMC which goes under the name of *Integral Sliding Mode Control (ISMC)*.

The main feature of this new control technique is the enforcement of sliding mode since the initial time instant t_0. Therefore, we need that condition $\sigma = 0$ is verified throughout our control period, independently from the initial state of our system $\boldsymbol{x}(0)$. A standard procedure to perform this is to consider a new modified sliding variable, called *integral sliding surface* $\Sigma(\boldsymbol{x}(t), \Gamma(t))$, which is based on the following definition:

$$\Sigma(\boldsymbol{x}(t), \Gamma(t)) = \sigma(\boldsymbol{x}(t), \Gamma(t)) - \lambda(t) \tag{3.60}$$

where $\sigma(\boldsymbol{x}(t), \Gamma(t))$ is the standard sliding surface retrieved from equation (3.29), while $\lambda(t)$ is a suitable *reaching function* which will be used to impose sliding mode since $t = t_0$. To compute its value, we can set the time derivative of $\Sigma(\boldsymbol{x}(t), \Gamma)$ equal to zero, considering the system as it is sliding on the new integral manifold [79]:

$$\begin{aligned} \dot{\Sigma}(\boldsymbol{x}(t), \Gamma(t)) &= 0 \\ \dot{\sigma}(\boldsymbol{x}(t), \Gamma(t)) - \dot{\lambda}(t) &= 0 \\ \frac{\partial \sigma}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} + \frac{\partial \sigma}{\partial \Gamma} \dot{\Gamma} &= \dot{\lambda}(t) \end{aligned} \tag{3.61}$$

if we integrate both sides starting from initial condition $\sigma(t_0)$, we finally obtain:

$$\lambda(t) = \int_{t_0}^{t} \left(\frac{\partial \sigma}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}(\tau) + \frac{\partial \sigma}{\partial \Gamma} \dot{\Gamma}(\tau) \right) \, d\tau + \sigma(t_0) \tag{3.62}$$

Since $\Sigma(t_0) = \sigma(t_0) - \sigma(t_0) = 0$, if we choose it as sliding surface, we are starting directly in sliding mode condition. Moreover, we inherit the convergence properties of standard SMC, because the control law is unchanged as well as uncertainties (theorem 3.2 hypotheses are still verified, we just need to adapt the Lyapunov function to the new Σ).

By putting everything together, we come up with a sliding mode controller which is able to totally eliminate reaching phase. Moreover, in our case, the integral sliding surface can be further simplified.

Theorem 3.3. *(Integral sliding mode) If we choose the equivalent control law v^{eq} as the nominal solution of the equation $\dot{\sigma}(\boldsymbol{x}(t), \Gamma(t)) = 0$, the integral sliding surface formulation becomes:*

$$\Sigma(\boldsymbol{x}(t), \Gamma(t)) = \sigma(\boldsymbol{x}(t), \Gamma(t)) - \sigma(\boldsymbol{x}(t_0), \Gamma(t_0)) \tag{3.63}$$

For master device, hypothesis is verified by choosing v^{eq} according to equation (3.33), while for the slave device (3.39) can be selected.

Proof. First of all, let's write the generic device dynamic model, expressed in its vectorial form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g} v \tag{3.64}$$

where $\boldsymbol{f}(\boldsymbol{x}) = [x_2 \;\; x_3 \;\; \eta(\boldsymbol{x})]^\top$ and $\boldsymbol{g} = [0 \;\; 0 \;\; \zeta]^\top$.

Then, we need to substitute equation (3.64) into the reaching function formula (3.62):

$$\lambda(t) = \int_{t_0}^{t} \left(\frac{\partial \sigma}{\partial \boldsymbol{x}} \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g} v \right) + \frac{\partial \sigma}{\partial \Gamma} \dot{\Gamma}(\tau) \right) \, d\tau + \sigma(t_0) \tag{3.65}$$

Now, we can apply to v the standard SMC law (3.40), supposing that perfect disturbance rejection (i.e. sliding mode condition) is achieved by v_{smc}. This assumption is verified because, using (3.60), we are laying on the sliding surface since the initial time instant t_0. If sliding mode is enforced, we are left with the equations of a double integrator generated by the ideal inverse dynamics:

$$\lambda(t) = \int_{t_0}^{t} \left(\frac{\partial \sigma}{\partial \boldsymbol{x}} (\bar{\boldsymbol{f}}(\boldsymbol{x}) + \bar{\boldsymbol{g}} v^{eq}) + \frac{\partial \sigma}{\partial \Gamma} \dot{\Gamma}(\tau) \right) d\tau + \sigma(t_0) \tag{3.66}$$

where $\bar{\boldsymbol{f}}(\boldsymbol{x}) = [x_2 \;\; x_3 \;\; 0]^\top$ and $\bar{\boldsymbol{g}} = [0 \;\; 0 \;\; 1]^\top$. If we now substitute $\sigma = \boldsymbol{G}\boldsymbol{x} + \Gamma$ (Equation (3.29)), we can compute the partial derivatives, obtaining:

$$\lambda(t) = \int_{t_0}^{t} \left(\boldsymbol{G}(\bar{\boldsymbol{f}}(\boldsymbol{x}) + \bar{\boldsymbol{g}} v^{eq}) + \dot{\Gamma}(\tau) \right) d\tau + \sigma(t_0) \tag{3.67}$$

To conclude the proof, it is sufficient to apply our hypothesis to compute v^{eq} and substitute its value into equation (3.67):

$$\begin{aligned} \dot{\sigma} &= 0 \\ \boldsymbol{G}\dot{\boldsymbol{x}} + \dot{\Gamma} &= 0 \\ \boldsymbol{G}(\bar{\boldsymbol{f}}(\boldsymbol{x}) + \bar{\boldsymbol{g}} v^{eq}) + \dot{\Gamma} &= 0 \\ v^{eq} &= \frac{-\boldsymbol{G}\bar{\boldsymbol{f}}(\boldsymbol{x}) - \dot{\Gamma}}{\boldsymbol{G}\bar{\boldsymbol{g}}} \end{aligned} \tag{3.68}$$

$$\lambda(t) = \int_{t_0}^{t} \left(\boldsymbol{G} \left(\bar{\boldsymbol{f}}(\boldsymbol{x}) + \bar{\boldsymbol{g}} \frac{-\boldsymbol{G}\bar{\boldsymbol{f}}(\boldsymbol{x}) - \dot{\Gamma}}{\boldsymbol{G}\bar{\boldsymbol{g}}} \right) + \dot{\Gamma} \right) d\tau + \sigma(t_0) = \sigma(t_0)$$

□

Thanks to theorem 3.3, in order to apply ISMC to our set-up, it is sufficient to *translate* the sliding manifold σ by a constant quantity related to the initial condition $\boldsymbol{x}(t_0)$ of the controlled system.

Since $\sigma(t_0)$ can be a-priori unknown, its value cannot be hard coded in the control algorithm. To overcome this problem, the initial value of σ can be measured and memorized. Then, in order to obtain Σ, it can be subtracted from the sliding variable and used to apply both standard SMC and higher order algorithms.

To verify all these properties of ISMC, a simulation comparison has been performed, with the same set-up used for the chattering - avoidance comparison. In

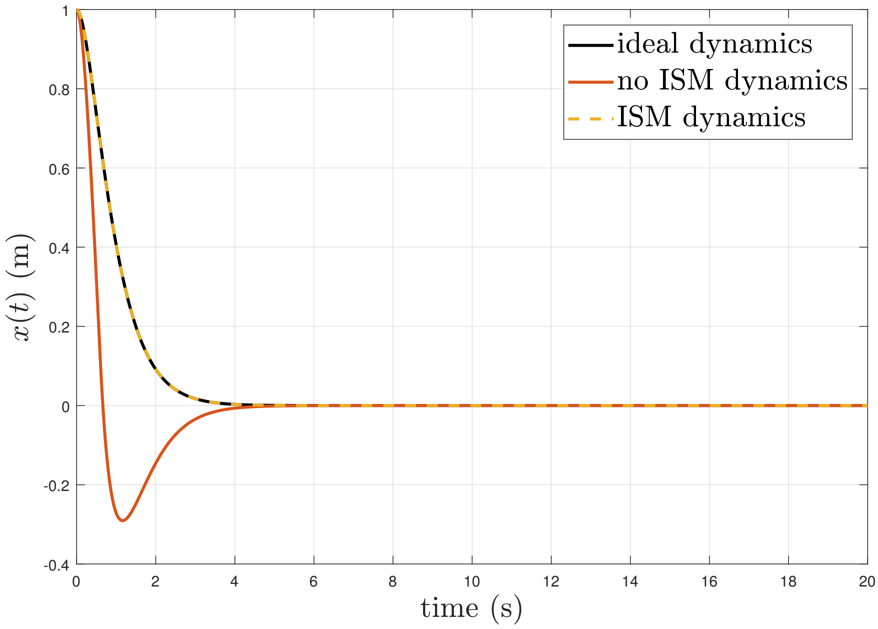
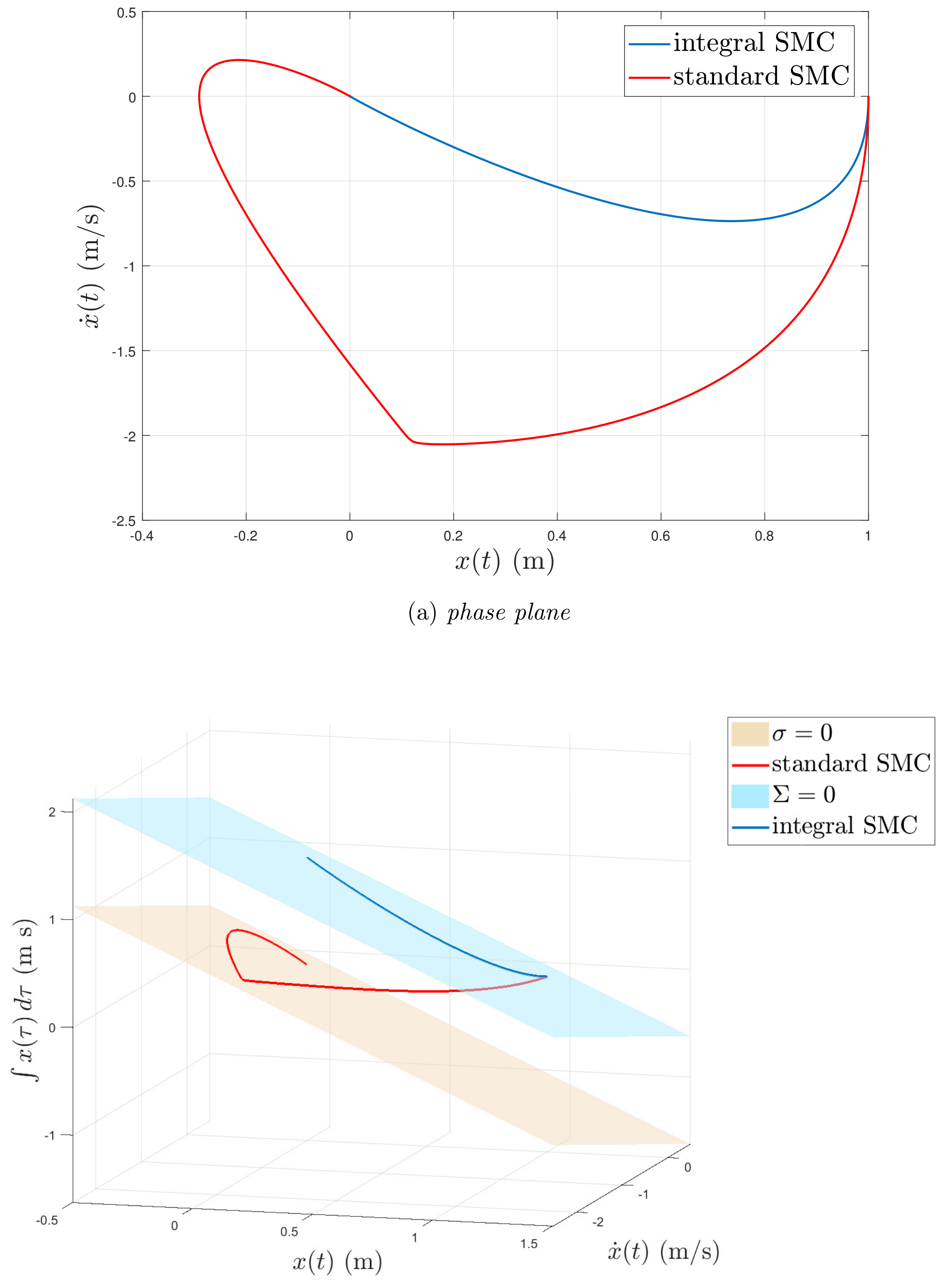


Figure 3.20: Transient response of simplified master device with initial position $x = 1\,\mathrm{m}$, with and without ISMC

this case, we adopt standard SMC to control the system with given initial condition $\boldsymbol{x}(0) = [1 \;\; 0]^\top$, obtaining a clear reaching phase before sliding mode is engaged (Figure 3.21a). Then, the integral sliding surface control logic is implemented, removing the offset from σ. As it can be seen at Figures 3.20 and 3.21a, this time the reaching phase is totally removed and the ideal dynamics is tracked from the initial time instant despite $\boldsymbol{x}(0)$. To clearly visualize the translation of the sliding manifold, the same trajectories have been plotted in the phase space in Figure 3.21b. As it can be noticed, convergence point of the new trajectory is different from the one obtained with standard σ. Indeed, they share the same physical coordinates x and $\dot{x}$, while the one used for state expansion, $\int x$, is different. This quantity represents a "fictitious" state, that has been added to the system in order to apply SMC theory. Therefore, its value is not related with the other two physical quantities and the system evolution, in both cases, converges towards the correct equilibrium as in Figure 3.21a.



(a) *phase plane*

(b) *phase space*

Figure 3.21: Phase portraits of standard SMC and ISMC. As we can notice, reaching phase is absent for the second one

3.8 Sliding mode MPC

Up to now, the control objective for our master and slave devices has been simply represented by tracking an imposed dynamic profile. This can be accomplished using an impedance control law of the type we have introduced in the earlier sections. With this choice, we are applying a linear relationship between the control signal v^{eq} and the measured positions and speeds.

Sometimes, however, tracking a desired impedance profile is not the only requirement which we would like to impose on our system. Typical examples are represented by moderation of control effort, upper and lower saturation of actuators, etc. Moreover, find a closed form solution of $\dot{\Sigma} = 0$ could be not trivial if we are dealing with multi - DOFs and redundant manipulators (see chapter 5).

These requirements and situations *cannot* be fulfilled by the standard linear impedance control, calling for a more advanced control strategy. To this aim, Model Predictive Control (MPC) has been introduced. This family of algorithms allow to reformulate the control problem as an *optimization* one, where different goals can be considered.

If we choose MPC as a control principle to design v^{eq}, while maintaining the same structure proposed for sliding mode controller in order to reject uncertainties (equation (3.40)), the final scheme takes the name of Sliding-mode Model Predictive Control SMPC [66, 39, 18]. Doing so we can exploit the advantages of both techniques, since MPC is very versatile but it requires a good model to be applied, while SMC can robustly guarantee the permanence of the system evolution onto the sliding surface but without additional constraints.

In order to design a MPC controller, we need to define a *cost function* J that will be minimized and the *finite horizon* k for which to compute the future control sequence. Since the optimal v^{eq} was directly calculated by imposing $\dot{\Sigma} = \dot{\sigma} = 0$ in nominal conditions, it seems logical to choose $J = \dot{\sigma}$ in order to obtain the best equivalent control with respect to optimization constraints. Finally, k has been chosen equal to 1 due to the following two remarks:

- Computational power of the experimental set-up is limited and minimization

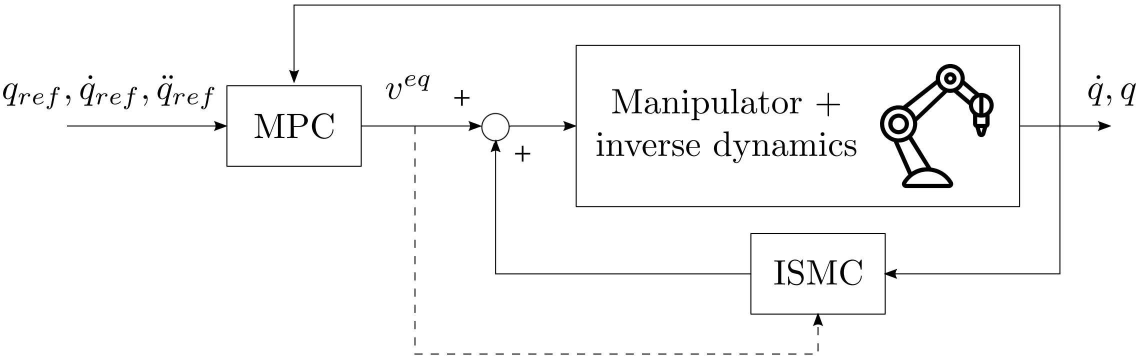


Figure 3.22: SMPC block diagram, where equivalent control information is fed to the SMC in order to avoid constraint violation

procedure on a larger time window could require more than the available cycle time to be performed;

- Although the major uncertainties have been rejected by SMC, the prediction of sliding surface $\Sigma(t+k|t)$ proposed in section 5.6 has revealed poor experimental accuracy;

$$\begin{aligned} \min_{v} \quad & \left|\left|v + \frac{\bar{b}}{\bar{m}}x_3 + \frac{\bar{k}}{\bar{m}}x_2 + \frac{1}{\bar{m}}(k_f f_e^d - f_h)\right|\right|_Q^2 \\ \text{s.t.} \quad & A_{low} \le Av \le A_{high}, \\ & v_{low} \le v + v_{smc} \le v_{high} \end{aligned} \tag{3.69}$$

where $Q > 0$ is a weight scalar. Written in this form, optimisation (3.69) represents a *quadratic programming* problem, which can be easily solved via open source algorithms, like `qpOASES` [20].

It is worth noting that if no constraint is active, the optimum solution of (3.69) coincides with the standard impedance control law v^{eq}. When constraints activate, MPC imposes an equivalent control law that is *different* from the theoretical one. This mismatch is minimized by the cost function structure, which however becomes different from zero. If SMPC is simply realized by *superimposing* the sliding mode action with the new control signal, the mismatch caused by constraint activation is seen as a sliding surface detachment for SMC. Without other information, this state movement is interpreted as an additional disturbance acting on the system, which the controller will try to compensate in order to bring back σ to zero. By doing so, the constraint imposed by MPC is violated, making one of the most appealing

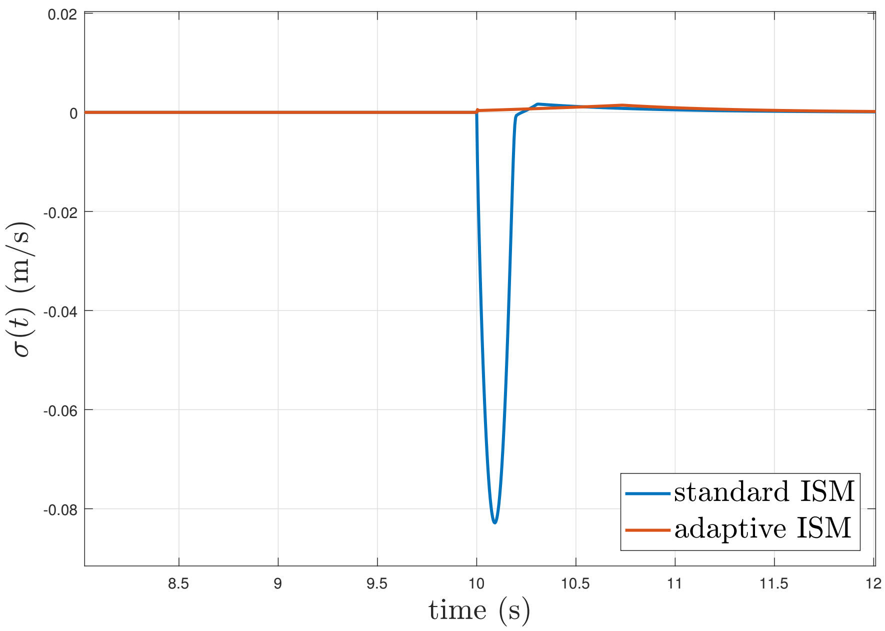


Figure 3.23: Improvement of adaptive reaching function with respect to the fixed nominal one - sliding variable

features of this control class totally useless.

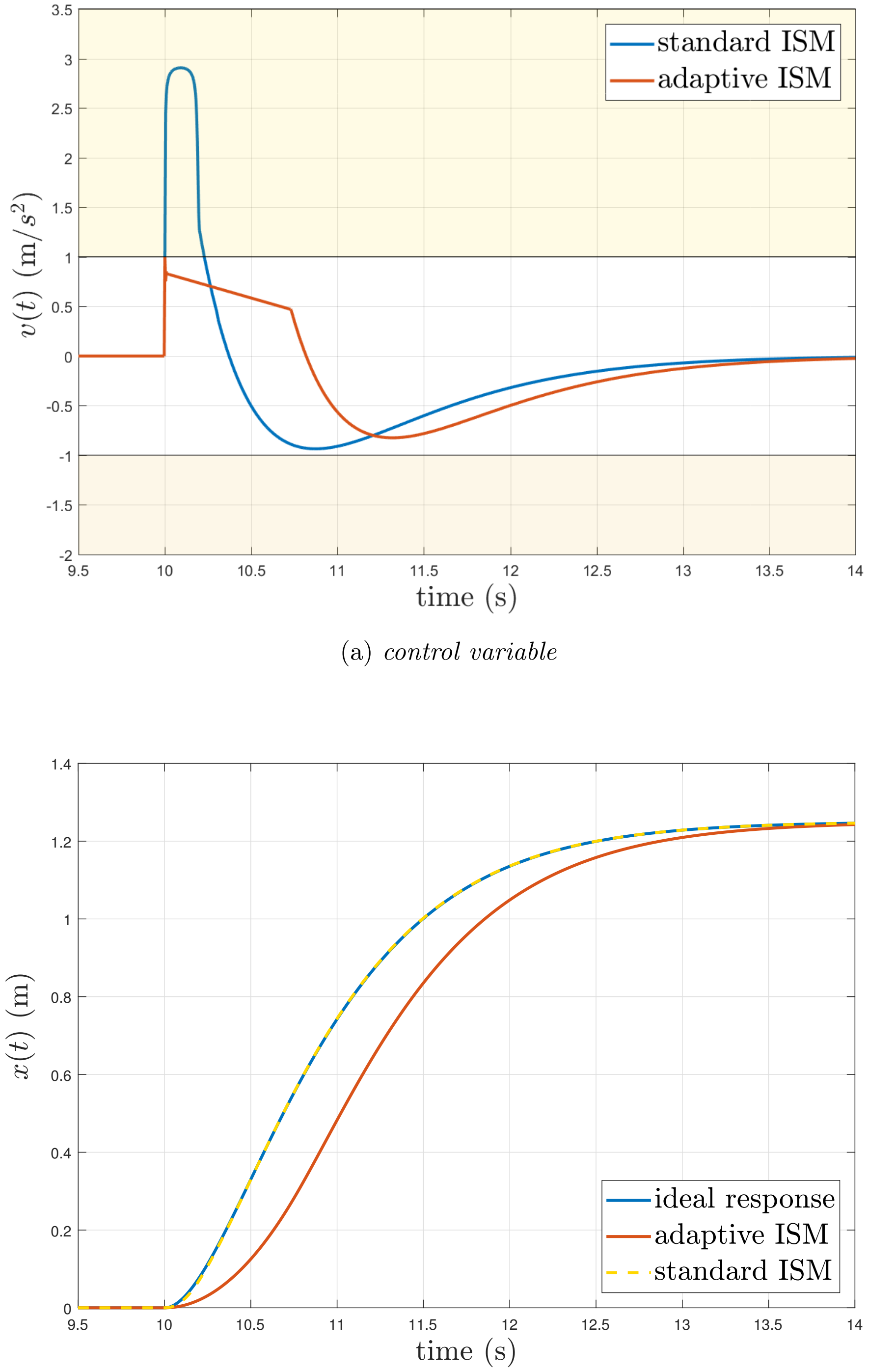
To solve this problem, the scheme modification proposed in Figure 3.22 is adopted [39]. Equivalent control applied by MPC is not always equal to the optimal one, therefore we need to communicate its variation also to the sliding mode controller. Doing so, it will be able to distinguish sliding surface drifts caused by actual disturbances or internal control strategies. More precisely, if $v^{eq} \neq v^{opt}$, theorem 3.3 is not valid any more. Therefore, to properly design an integral sliding surface that will not reject the new v^{eq}, it is necessary to solve the reaching function:

$$\lambda(t) = \int_{t_0}^{t} \Big(\boldsymbol{G}(\bar{\boldsymbol{f}}(\boldsymbol{x}) + \bar{\boldsymbol{g}} v^{eq}) + \dot{\Gamma}(\tau) \Big) \, d\tau + \sigma(t_0) \tag{3.70}$$

with the equivalent control v^{eq} coming directly from MPC minimization procedure. This choice of $\lambda(t)$ converges to the already analysed case when no constraint is active, while adapting its value when the MPC controller behaves differently from the standard impedance one.

To verify these statements, a simulation has been performed, with a set-up similar to that of subsection 3.6. This time, a model predictive controller is in charge of giving the equivalent control and when, at 10 s, a step change in the human force is given, v^{eq} is saturated using a proper constraint. In Figure 3.23 it is possible to observe the response of standard "fixed" ISMC and adaptive one with reaching function (3.70). It is clearly visible that mismatch between theoretical and applied equivalent control causes a drift in σ, which is compensated by SMC violating the constraint. Instead, with the correct expression of $\lambda(t)$, this drift is no more present since SMC is aware that the change in v^{eq} is due to a constraint activation in MPC.

If $\lambda(t)$ is changed on-line according to the current v^{eq}, resultant sliding surface applied to the system *does not* represent the desired impedance relationship any more. This is reasonable because, due to limitations in the control action, it is not possible to reject all the uncertainty and tracking performance worsen consequently. This effect is visible in Figure 3.24b, where the position response of adaptive ISMC is compared to the standard one. When the control signal exceeds $[-1,1]\,\mathrm{m\,s^{-2}}$, the modified SMPC saturates its action, guaranteeing constraints fulfilment. Instead, if $\lambda(t)$ is kept fixed, constraint is treated as a disturbance and rejected, violating the maximum imposed acceleration (Figure 3.24a).



(a) *control variable*

(b) *position response*

Figure 3.24: Improvement of adaptive reaching function with respect to the fixed nominal one. Top: control action, bottom: position response

Chapter 4

Global scheme analysis

The goal of this chapter is to move the analysis from the single master/slave device to the entire teleoperation scheme. To do so, it is necessary to introduce some elements of *network theory*, which allow a correct description of teleoperation transparency and stability properties. Then, some matrix descriptions of the overall scheme are introduced and the most useful ones selected. Using them, a systematic stability check is performed, based on different criteria. Particular relevance has been given to the *Llewellyn's stability criterion*, from which the conditions for network stability have been retrieved. Next, a *transparency analysis* has been performed and its trade - off with stability has been assessed. To conclude, previous considerations have been verified in simulation using a 1 DOF master - 1 DOF slave teleoperation system and a coupled control logic has been implemented. The criterion chosen is the maximisation of stability when slave is in contact with the environment, while transparency is maximised when it moves freely. To change teleoperation behaviour, *network passivity* is assessed in real-time. The two proposed strategies used to realize the impedance switching have been called *continuous adaptation* and *two - state adaptation*.

4.1 Teleoperation matrices representations

In order to shift the analysis from the single architecture element to the network, it is necessary to introduce a *new formalism* to model the bilateral teleoperation system, which will be useful in the following sections. First of all, let us consider the models introduced in chapter 2 for the master and slave devices, as well as the operator and environment representations. For simplicity, let us focus on the single DOF components; doing so, the complexity of the network is contained. Furthermore, the extension to more difficult cases can be treated straightforwardly by considering decoupled coordinates dynamics in the operational space (superimposition of multiple 1 DOF networks).

A generic mechanical system can be represented as a combination of *multi - poles*, which exchange *energy* at their ports through *conjugate variables.* These two functions of time are usually identified with the name of *effort* and *flow* and their product represents the amount of *power* that is given/taken by the multi - pole to the rest of the network it is connected to. This descriptive method goes under the name of *equivalent modelling* and it is useful to analyse the energy flows in distributed systems.

In the case of interest, the effort variable is represented by the resultant of the forces exerted on the mechanic device, while the flow variable (describing the motion of the system) is the velocity. The mathematical relationship between effort and flow is called *impedance.* It defines the behaviour of the device when a force or a velocity acts on it. In chapter 2, impedance values for the principal teleoperation components have been already computed:

$$
\begin{aligned}
Z_h &= \frac{F_h^{des} - F_h}{V_m} = m_h s + b_h + \frac{k_h}{s} \\
Z_e &= \frac{F_e}{V_e} = b_e + \frac{k_e}{s} \\
Z_s &= \frac{F_s - F_e}{V_s} = m_s s + b_s \\
Z_m &= \frac{F_h + F_m}{V_m} = m_m s + b_m
\end{aligned}
\tag{4.1}
$$

where Z_h is the operator arm impedance relation, Z_e is the simplified environmental contact model and Z_s, Z_m are the mechanical models, respectively, for the master and slave devices. According to the definition of multi - poles and equivalent model given before, it is possible to represent the network generated by these components as

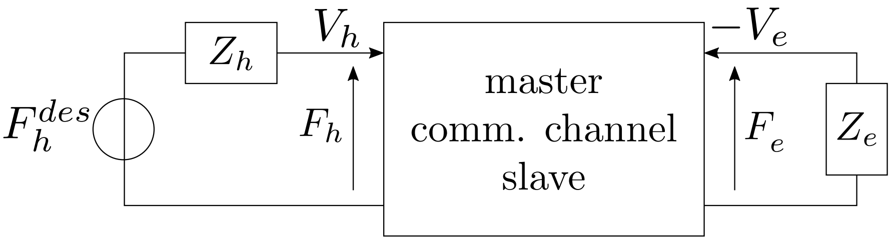


Figure 4.1: Network representation for a generic bilateral teleoperation scheme

depicted in Figure 4.1. The principal source of energy for the system is represented by the operator, through the input F_h^{des}. However, the force he/she desires to impose to the haptic interface is filtered by the arm impedance, represented by Z_h. Therefore, the force effectively applied on the teleoperation network is equal to F_h. After passing through the master - slave path, the energetic flow reaches the environment, which is represented by Z_e. The force exerted on it (F_e) defines the slave speed $V_s = V_e$.

Together with this energetic flow, another one exists in the opposite direction, which reflects the energy from the environment to the human arm (bilateral teleoperation). Consequently, the aggregated system composed by master and slave devices, together with the communication channel, constitutes a *2 - port network* [64], which interfaces the operator to the environment and vice versa (central block in Figure 4.1). Sharing concepts from circuit theory, the mathematical model of the teleoperation block can be described in the frequency domain using a 2 × 2 matrix, which relates the effort and flows at each port. According to the chosen couples of conjugate variables, this matrix can take 6 different formulations. Among them, two will be the most relevant for the following studies:

- **Impedance matrix $\boldsymbol{Z}$:**

$$\begin{bmatrix} F_h \\ F_e \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} V_h \\ -V_e \end{bmatrix} \tag{4.2}$$

- **Hybrid matrix $\boldsymbol{H}$:**

$$\begin{bmatrix} F_h \\ -V_e \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} V_h \\ F_e \end{bmatrix} \tag{4.3}$$

The first representation has the advantage of describing the interaction between master and slave ports using *homogeneous parameters* (all the terms are impedances). However, the off - diagonal elements, which define the cross correlation between the two ports, have not immediate interpretation. If the reader is interested in their expressions, a full description of the teleoperation system using the impedance matrix can be found in [63].

According to [50], hybrid representation is the favourite description of bilateral teleoperation systems. Although H_{11}, H_{12}, H_{21} and H_{22} have different unit of measures, they all have a *clear physical meaning*. Indeed, their expressions can be straightforwardly derived from the definitions of master and slave impedances.

Observation 4.1. *If the local controllers developed in chapter 3 work correctly, it is reasonable to assume that the impedance relationships that describe master and slave devices have the following form:*

$$\bar{m}_m \ddot{x}_m + \bar{b}_m \dot{x}_m + \bar{k}_m x_m = f_h - k_f f_e^d \tag{4.4}$$

$$\bar{m}_s \ddot{\tilde{x}}_s + \bar{b}_s \dot{\tilde{x}}_s + \bar{k}_s \tilde{x}_s = -f_e \tag{4.5}$$

where $\bar{m}_m$, $\bar{b}_m$, $\bar{k}_m$, $\bar{m}_s$, $\bar{b}_s$ and $\bar{k}_s$ are design parameters.

If this observation holds, the terms of matrix $\boldsymbol{H}$ are:

$$H_{11} = \left.\frac{F_h}{V_h}\right|_{F_e=0} = s\bar{m}_m + \bar{b}_m + \frac{\bar{k}_m}{s} \tag{4.6}$$

$$H_{12} = \left.\frac{F_h}{F_e}\right|_{V_h=0} = k_f e^{-ds} \tag{4.7}$$

$$H_{21} = \left.\frac{-V_e}{V_h}\right|_{F_e=0} = -k_p e^{-ds} \tag{4.8}$$

$$H_{22} = \left.\frac{-V_e}{F_e}\right|_{V_h=0} = \frac{s}{s^2\bar{m}_s + s\bar{b}_s + \bar{k}_s} \tag{4.9}$$

where the values for F_h, V_h, F_e and V_e have been derived from the Laplace domain transformation of (4.4) and (4.5). Expressions (4.6) and (4.9) represent, respectively, the master impedance and the slave admittance, while the off - diagonal adimensional terms (4.7) and (4.8) identify the force and position scaling of the teleoperation system. The hybrid representation is convenient because it confines the communication channel description on H_{12} and H_{21}, without modifying the diagonal terms related to master and slave sides.

4.2 Stability analysis

The first property that the bilateral teleoperation scheme must satisfy is *stability*. According to [16, 10], a 2 - port network is said to be *absolutely stable* if it is stable for every possible couple of passive dipoles connected to its extremities. A dipole is said to be *passive* if and only if [32]:

$$\int_0^t f(\tau)v(\tau)\,d\tau + E(0) \geq 0 \tag{4.10}$$

where $f(t)$ and $v(t)$ are the conjugate variables acting on the dipole and $E(0)$ is the energy initially stored in the dipole. If in (4.10) the inequality is replaced by an equality, the element is said to be *lossless*.

Network passivity is a more tight requirement with respect to absolute stability. Indeed, it can be proved that passivity implies absolute stability, but not vice versa. A network which is not absolutely stable can be potentially unstable. This means that there exists passive termination dipoles which destabilize the network.

Equation (4.10) defines the energy balance of a passive dipole, which cannot be negative. The 1 - port element that describes the environment (independently on how complex its representation is) cannot generate energy, therefore it can be considered as a passive component. Also the operator's arm is described as an impedance relationship, inheriting passivity. Moreover, there are evidences that *"humans remain stable when interacting with passive systems"* [83]. This is particularly true if the teleoperation signals interact with the haptic interface at frequencies which are *high* compared to the voluntary motion bandwidth.

That is why it will be sufficient to prove the absolute stability of the 2 - port master/slave network to guarantee the overall bilateral teleoperation system stability, independently of the possible environments or operators that interact with it. To do so, several criteria are available in literature, which can be applied to any of the four matrix representations:

- **Raisbeck's passivity criterion** [36]: the 2 - port network described in (4.3) is passive if and only if:

  1. All $\boldsymbol{H}$ terms have no poles in the Right - Half Plane (RHP)
  2. Each pole of $\boldsymbol{H}$ which lays on the imaginary axis is simple and its residual fulfil the following inequalities (R_{ij} is the residual of H_{ij}, R_{ij}^* is the complex conjugate of R_{ij}):
     - $R_{11} \geq 0$
     - $R_{22} \geq 0$
     - $R_{11}R_{22} - R_{12}R_{21} \geq 0 \quad \text{with } R_{21} = R_{12}^*$
  3. All $\boldsymbol{H}$ terms satisfy $\forall\omega$:

  $$\operatorname{Re} H_{11} \operatorname{Re} H_{22} - (\operatorname{Re} H_{12} + \operatorname{Re} H_{21})^2 - (\operatorname{Im} H_{12} - \operatorname{Im} H_{21})^2 \geq 0 \quad (4.11)$$

- **Llewellyn's absolute stability criterion** [33, 10]: the 2 - port network described in (4.3) is absolutely stable if and only if:

  1. H_{11} and H_{22} have no poles in the RHP
  2. Each pole of H_{11} and H_{22} which lays on the imaginary axis is simple and its residual is real and positive
  3. All $\boldsymbol{H}$ terms satisfy $\forall\omega$:
     - $\operatorname{Re} H_{11} \geq 0$
     - $\operatorname{Re} H_{22} \geq 0$
     - $2 \operatorname{Re} H_{11} \operatorname{Re} H_{22} - \operatorname{Re} H_{12}H_{21} - |\operatorname{Re} H_{22}H_{11}| \geq 0 \quad (4.12)$

- **Scattering approach** [50]: it is based on the definition of *incident* and *reflected* waves, that are a linear combination of the conjugate variables. These waves are related by the *scattering operator* S:

$$f(t) - v(t) = S(f(t) + v(t))$$

Using the frequency domain approach as for the hybrid and impedance matrices, it is possible to define the *scattering matrix* as a function of $\boldsymbol{H}$ [50]:

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (\boldsymbol{H} - \boldsymbol{I})(\boldsymbol{H} + \boldsymbol{I})^{-1} \tag{4.13}$$

Once $\boldsymbol{S}$ has been derived, the 2 - port network associated to it is passive if and only if $||\boldsymbol{S}||_\infty \leq 1$

Out of these three theorems, the first and the last ones assess the network passivity, while the second one defines the conditions for absolute stability. Since passivity is a more conservative property with respect to absolute stability, the *Llewellyn's criterion* has been adopted to find the values of $\boldsymbol{H}$ that guarantee network stability. Doing so, a larger parameter set should be found with respect to the passivity criteria, that are more stringent.

4.2.1 Llewellyn's analysis

By substituting (4.6, 4.7, 4.8, 4.9) into Llewellyn's absolute stability criterion, it is possible to find the value of matrix $\boldsymbol{H}$ parameters that guarantee teleoperation network absolute stability:

1. H_{11} has a pole in zero, therefore it has no RHP poles. H_{22} poles are defined as the solutions of the second order equation $s^2\bar{m}_s + s\bar{b}_s + \bar{k}_s = 0$. Since all slave impedance parameters are positive, by applying Descartes' sign rule it is possible to conclude that all poles of H_{22} have negative real part;

2. Poles of H_{22} do not lay on the imaginary axis, while H_{11} pole (which is located in zero) is simple and its residual is given by:

$$R_{11} = \lim_{s \to 0}[sH_{11}] = \bar{k}_m \qquad \bar{k}_m \in \mathbb{R}^+$$

In order to verify the conditions imposed by the third point of Llewellyn's criterion, it is necessary to find the frequency response of the hybrid matrix $\boldsymbol{H}$, which is simply obtained by substituting s with $j\omega$:

$$\boldsymbol{H}(j\omega) = \begin{bmatrix} \bar{b}_m + j\left(\bar{m}_m\omega - \dfrac{\bar{k}_m}{\omega}\right) & k_f e^{-j\omega d} \\ -k_p e^{-j\omega d} & \dfrac{j\omega}{-\bar{m}_s\omega^2 + j\omega\bar{b}_s + \bar{k}_s} \end{bmatrix} \tag{4.14}$$

- ∘ $\operatorname{Re} H_{11} = \bar{b}_m \geq 0 \qquad \forall\omega$
- ∘ $\operatorname{Re} H_{22} = \dfrac{\omega^2\bar{b}_s}{(\bar{k}_s - \bar{m}_s\omega^2)^2 + \omega^2\bar{b}_s^2} \geq 0 \qquad \forall\omega$

These two properties are always verified thanks to the positiveness and realness of the master and slave impedance parameters. Therefore, absolute stability of the network represented by $\boldsymbol{H}$ is verified $\forall\omega$ if and only if condition (4.12) holds. After having properly made explicit all its terms, the inequality that needs to be fulfilled has the following form:

$$\frac{2\omega^2\bar{b}_s\bar{b}_m}{\omega^2\bar{b}_s^2 + (\omega^2\bar{m}_s - \bar{k}_s)^2} + k_p k_f(\cos 2\omega d - 1) \geq 0 \qquad \forall\omega \tag{4.15}$$

The left hand side of the inequality is a function of ω and it is called *Llewellyn's curve* $\Lambda(\omega)$. It is composed by the sum of a rational term and a trigonometrical one, which generate the behaviour shown in Figure 4.2. It is an adimensional quantity and it positiveness defines the absolute stability of the analysed network. As said before, it is possible to achieve the same inequality (4.15) using interchangeably any network matrix representation. To check whether the Llewellyn's curve is greater than zero, different teleoperation operative conditions have been analysed.

No time delay First of all, let's consider the case where there is no time delay affecting the communication channel between master and slave. Consequently, $d = 0$ and the trigonometric term of $\Lambda(\omega)$ cancels out. The remaining expression is:

$$\Lambda(\omega) = \frac{2\omega^2\bar{b}_s\bar{b}_m}{\omega^2\bar{b}_s^2 + (\omega^2\bar{m}_s - \bar{k}_s)^2} \tag{4.16}$$

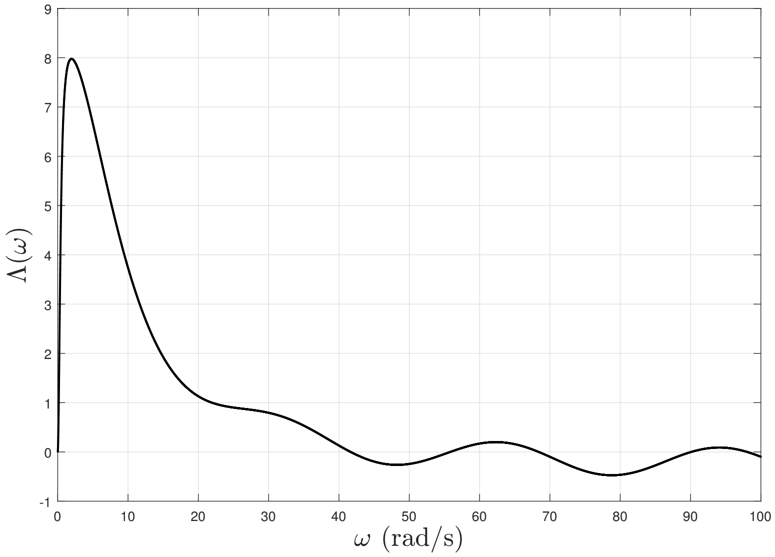
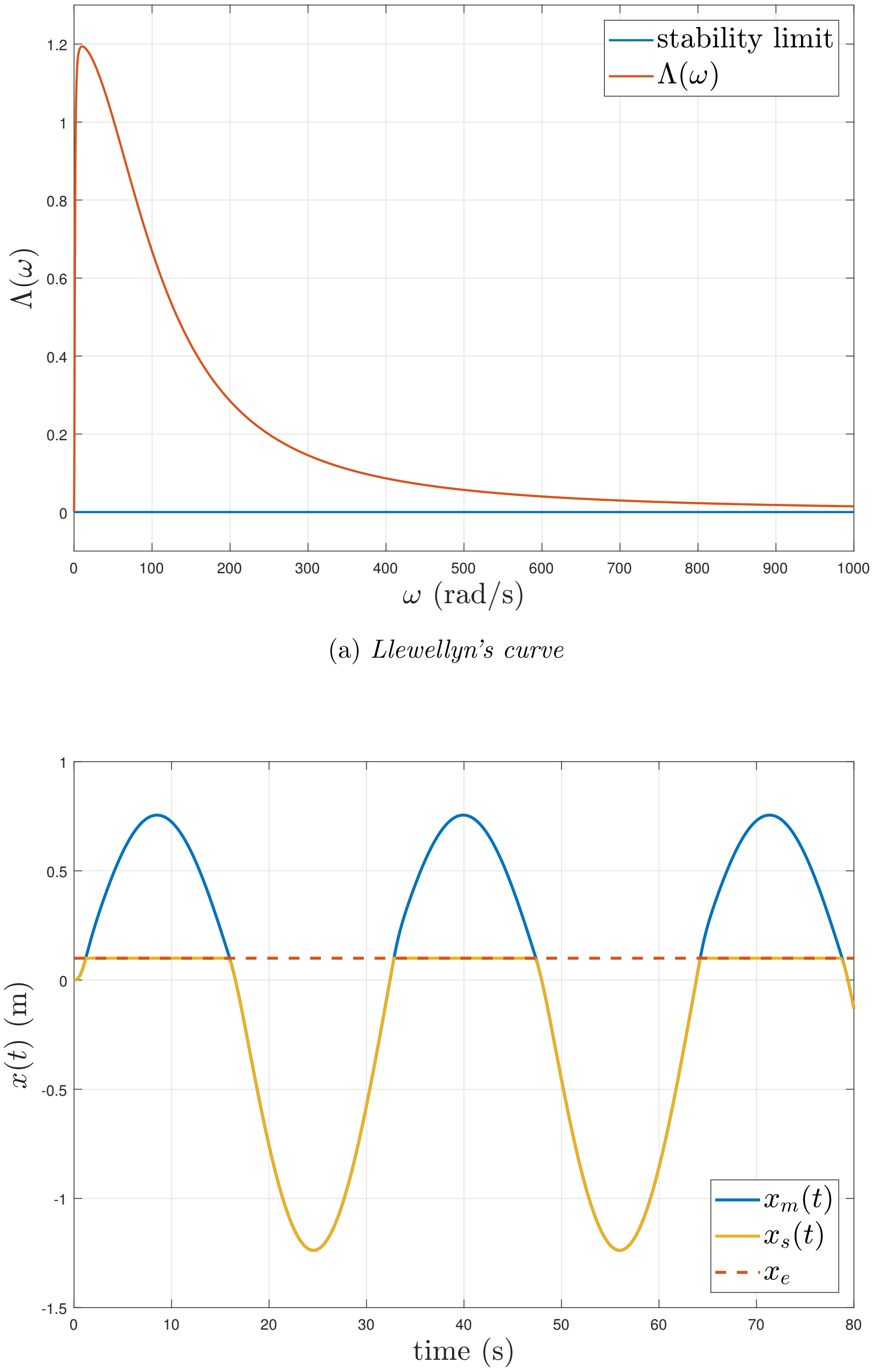


Figure 4.2: Llewellyn's curve for a generic set of impedance parameters and a fixed time delay

which is positive for every value of ω.

This means that if the teleoperation system is not affected by time delay, then it is absolutely stable independently of the frequencies of the signals transmitted through the communication channel and the design parameters choice. This can be appreciated by looking at the Llewellyn's curve plot in Figure 4.3a, which is always positive.

To verify the previous statement, a simple simulation has been set - up. 1 DOF master and slave systems are controlled using SMC and they are connected without time delay. The master device is fed with a sinusoidal force reference of amplitude 5 N and frequency 0.2 rad s^{-1} to simulate the operator task movement. The slave device interacts with the environment, that is modelled as (2.10) with wall position x_e equal to 0.1 m, wall stiffness $\bar{k}_e$ 6×10^5 N m^{-1} and wall damping $\bar{b}_e$ 100 N s m^{-1}. The profiles of master and slave positions during the teleoperation task have been reported in Figure 4.3b, as well as the wall position. As it can be noticed, no oscillation occurs neither on the master side nor on the slave side. The master trajectory is deformed during contact motion thanks to the force feedback coming from the slave device. When there is no contact with the wall (i.e. free motion), the tracking error between master and slave devices is negligible.



(a) *Llewellyn's curve*

(b) *teleoperation task*

Figure 4.3: Bilateral teleoperation network when the communication channel is not affected by time delay

General case Let's now consider a generic communication time delay d (symmetric with RTT equal to $2d$), between the master and slave devices. By looking at the *asymptotic* behaviour of $\Lambda(\omega)$, the following theorem can be drawn.

Theorem 4.1. *(Delay instability) If the teleoperation system is subject to time delay d different from zero, then it is not possible to guarantee absolute stability for all frequencies of the signals transmitted through the communication channel*

Proof. Let's consider the limit of the Llewellyn's curve for increasing ω:

$$\lim_{\omega\to\infty}\left[\frac{2\omega^2\bar{b}_s\bar{b}_m}{\omega^2\bar{b}_s^2+(\omega^2\bar{m}_s-\bar{k}_s)^2}+k_pk_f(\cos 2\omega d-1)\right] \tag{4.17}$$

this limit can be split into two parts:

- The first part is rational and tends to cancel out for increasing ω

$$\lim_{\omega\to\infty}\left[\frac{2\omega^2\bar{b}_s\bar{b}_m}{\omega^2\bar{b}_s^2+(\omega^2\bar{m}_s-\bar{k}_s)^2}\right]=0$$

- The second part is trigonometric and the limit for $\omega\to\infty$ is not defined, because the function continues to oscillate between 0 and $-2k_pk_f$ at a frequency which is related to d

$$\lim_{\omega\to\infty}\left[k_pk_f(\cos 2\omega d-1)\right]=\text{not defined}$$

Therefore, if ω is chosen large enough so that the effect of the rational part is negligible, there exists a frequency value ω_0 at which the Llewellyn's curve becomes negative, violating the absolute stability criterion. □

Since Llewellyn's curve *always* becomes negative at high frequencies, it makes sense to compute the values of $\omega \geq 0$ for which the crossing happens. These will define the available bandwidth in which the teleoperation system can be absolutely stable also in presence of time delay. Obviously, this bandwidth will depend on the amount of time delay d affecting the system. Indeed, high delay implies very fast oscillations in Λ, generating earlier crossings of the stability threshold $\Lambda(\omega)=0$.

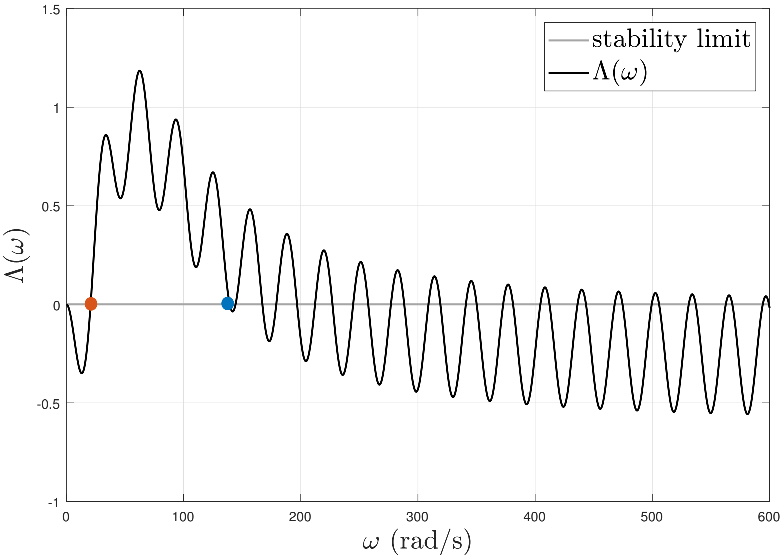


Figure 4.4: Llewellyn's curve and stability limit, with underlined early crossing (red dot) and high frequency crossing (blue dot)

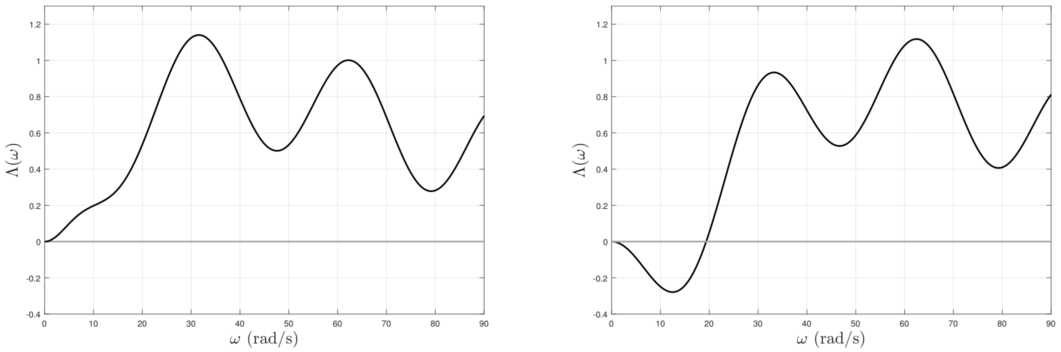
According to Figure 4.4, the bandwidth at which the network is stable is determined by the interval of frequencies $[\omega_1, \omega_2]$ rad s^{-1}, where ω_1 identifies the *initial crossing* (red dot) and ω_2 the *high frequency crossing* (blue dot).

Initial crossing It is due to the fact that the Llewellyn's curve, while starting at zero, has negative concavity. Therefore, it goes below the stability threshold, becoming positive only after ω_1. These networks are generally *potentially unstable*, because when the system reaches an equilibrium condition, signals frequencies tend to zero.

To *avoid* such type of crossing, it is necessary to constrain Λ to have initial positive concavity. In this way, the first oscillation is over the stability threshold. In order to impose such condition, Llewellyn's curve second derivative has been computed and evaluated in the origin. `Maxima` symbolic computation program has been adopted to perform this passage:

$$\left.\frac{\partial^2\Lambda(\omega)}{\partial\omega^2}\right|_{\omega=0} \geq 0 \qquad \rightarrow \qquad \frac{\bar{b}_m\bar{b}_s}{\bar{k}_s^2} - d^2k_fk_p \geq 0 \tag{4.18}$$

Inequality (4.18) constitutes a relationship between three design parameters ($\bar{b}_m$,



(a) $\bar{k}_s$ *fulfils* (4.19)

(b) $\bar{k}_s$ *does not fulfil* (4.19)

Figure 4.5: Necessary condition to avoid early crossing of Llewellyn's curve

$\bar{b}_s$ and $\bar{k}_s$). Therefore, if two of them have been chosen according to other design principles, the third one must fulfil the inequality in order to guarantee no initial crossing. As it will be seen in the next subsections, $\bar{b}_m$ and $\bar{b}_s$ have a stronger impact on $\Lambda(\omega)$ with respect to $\bar{k}_s$. Thus, it seems reasonable to explicit condition (4.18) on $\bar{k}_s$:

$$\bar{k}_s \leq \sqrt{\frac{\bar{b}_m \bar{b}_s}{d^2 k_f k_p}} \tag{4.19}$$

Once $\bar{b}_m$ and $\bar{b}_s$ are chosen, it is necessary to fulfil (4.19) to avoid early crossing. As it can be seen, the inequality depends on the amount of delay that affects the network. In presence of a very long time delay, $\Lambda(\omega)$ has a highly oscillating behaviour and $\bar{k}_s$ needs to be reduced consequently. That is why, to improve the robustness of the boundary with respect to d, it is preferable to substitute $\leq$ in (4.19) with $\ll$ (worst case scenario for $\bar{k}_s$). In Figure 4.5, it is possible to check the effect of condition (4.19) of removing the initial down - slope in the Llewellyn's curve.

High frequency crossing Thanks to the asymptotic property proved in theorem 4.1, there exists a frequency ω_2 at which the Llewellyn's curve crosses the stability limit for the first time. If condition (4.19) holds, there is no lower bound for the absolute stability bandwidth and $\Lambda(\omega)$ is positive starting from 0 up to ω_2. This high frequency limit should depend on time delay, because the higher is the frequency of oscillations, the lower is the bandwidth. However, d is generally *unknown* and/or *not constant*. Thus, it would be better to derive a frequency boundary that

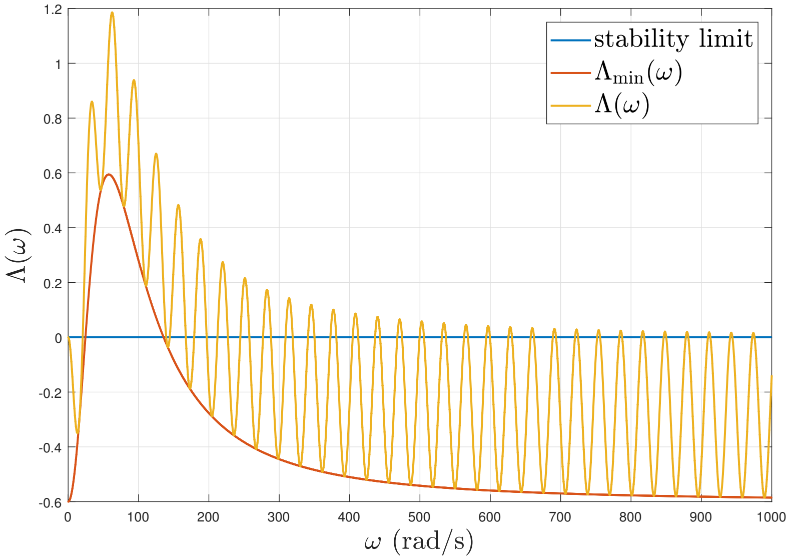


Figure 4.6: Lower envelope of the Llewellyn's curve compared with the real one

is *independent* from the value of d. That is why the following analysis will be focused on the *worst case scenario*, that is represented by the *lower envelope* of Llewellyn's curve. This is simply obtained by imposing $\cos 2\omega d = -1$ in (4.15):

$$\Lambda_{\min}(\omega) = \frac{2\omega^2 \bar{b}_s \bar{b}_m}{\omega^2 \bar{b}_s^2 + (\omega^2 \bar{m}_s - \bar{k}_s)^2} - 2k_p k_f \tag{4.20}$$

A visual representation of Llewellyn's curve and $\Lambda_{\min}$ can be seen in Figure 4.6. It is possible to notice that the crossing frequency defined by $\Lambda_{\min}$ is more conservative than the actual one. If $\hat{\omega}_2$ is the value of ω for which $\Lambda_{\min}(\omega) = 0$ at high frequency, $\hat{\omega}_2 \to \omega_2$ if $d \to \infty$. Moreover, the starting point of $\Lambda_{\min}$ is the negative constant $-2k_p k_f$. Therefore $\Lambda_{\min}$ intersect the stability limit always twice and it is necessary to select only the second relevant one.

$\Lambda_{\min}$ has also the great advantage of being a *rational* function of ω. Consequently, the number of crossings with the stability limit is equal to 2 whatever impedance parameter tuning is chosen. Moreover, it is possible to write $\Lambda_{\min}(\omega) = 0$ as a *biquadratic* equation in ω:

$$\underbrace{k_f k_p \bar{m}_s^2}_{a} \omega^4 + \underbrace{(k_f k_p \bar{b}_s^2 - 2k_f k_p \bar{k}_s \bar{m}_s - \bar{b}_s \bar{b}_m)}_{b} \omega^2 + \underbrace{k_f k_p \bar{k}_s^2}_{c} = 0 \tag{4.21}$$

Out of the 4 solutions, 2 identify negative ω and need to be discarded. For the remaining 2 solutions, the one with greater modulus is picked:

$$\hat{\omega}_2 = \sqrt{\frac{-b + \sqrt{b^2 - 4ac}}{2a}} \tag{4.22}$$

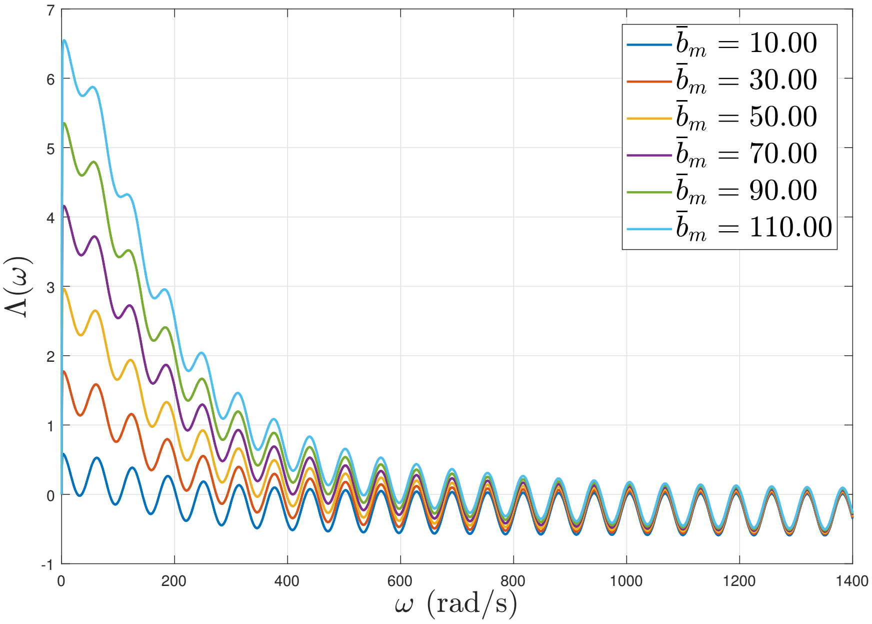
4.2.2 Llewellyn's curve sensitivity

Since it is quite complex to analyse the effect of the various impedance parameters on the Llewellyn's curve absolute stability bandwidth (4.22), it is convenient to perform a sensitivity analysis to roughly understand how the curve modifies its shape when parameters vary. For this procedure, a fixed time delay d of 0.1 s has been considered, as well as fixed position and force scaling factors $k_f = 0.3$ and $k_p = 1$. Their variations will not be considered because their choice is not possible (time delay) or it is dictated by external motivations (scaling factors).

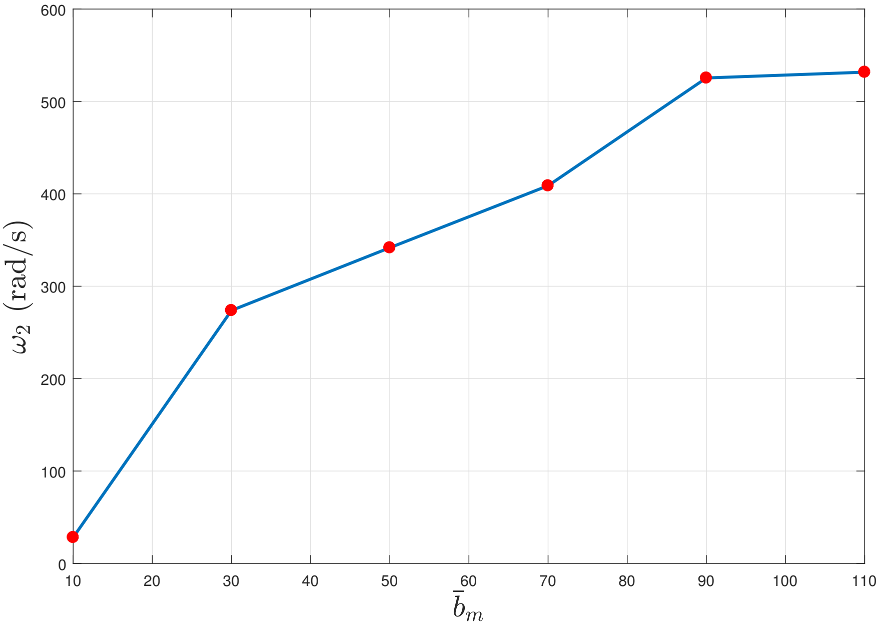
The first significant observation is that $\bar{m}_m$ and $\bar{k}_m$ do not appear in the definition of the Llewellyn's curve (4.15). Therefore, their values do not influence the stability property of the network and their values can be tuned in order to obtain the desired master dynamics. Doing so, it is possible to optimise the manipulability of the haptic interface or the operator's comfort without modifying the stability of the network.

Then, $\bar{b}_m$ variations have been checked. As it can be seen in Figure 4.10a, high values of master damping raise the amplitude of $\Lambda(\omega)$ at low frequencies. This increases the stability bandwidth of the system, as it can be appreciated in Figure 4.7b. Therefore, $\bar{b}_m$ should be set as high as possible to improve absolute stability. Unfortunately, high damping values on the haptic device interferes with operator manoeuvres, causing bad feeling and low reactivity. Therefore, this parameter is quite critical and its variation should be managed with caution. Moreover, on - line changes of $\bar{b}_m$ may confuse the operator perception of the environment.

For what concerns the variations of $\bar{m}_s$, it is possible to notice the opposite effect with respect to the previous case. Here, the lower is the parameter value, the higher is the available stability bandwidth of the teleoperation system (Figures 4.8a and 4.8b). However, also in this case it is not recommendable to choose $\bar{m}_s$ to close

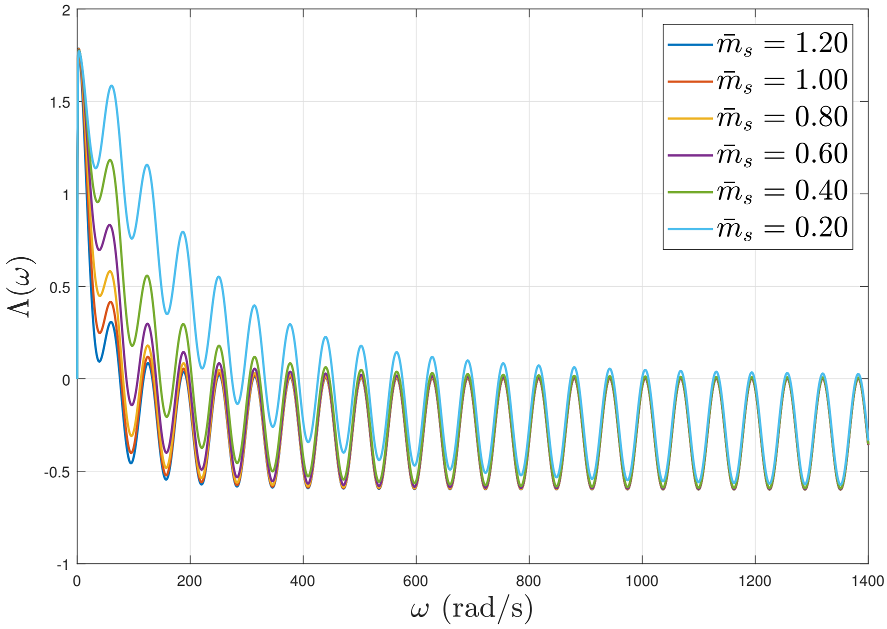


(a) *Llewellyn's curve*

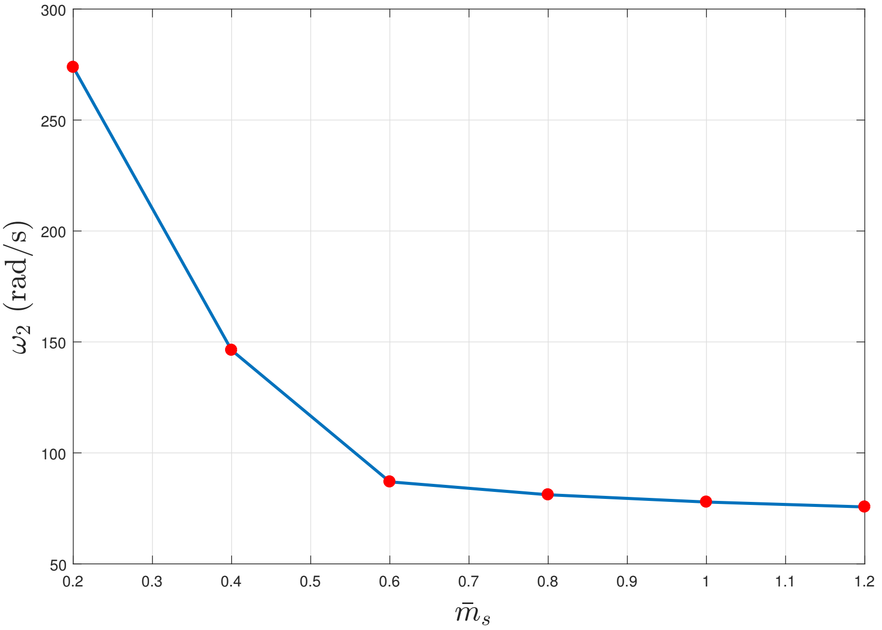


(b) *crossing frequency*

Figure 4.7: Sensitivity analysis with respect to $\bar{b}_m$ variations



(a) *Llewellyn's curve*



(b) *crossing frequency*

Figure 4.8: Sensitivity analysis with respect to $\bar{m}_s$ variations

to zero, because this would generate a low impedance tracking profile, with the consequence of a poor disturbance rejection during free motion operations.

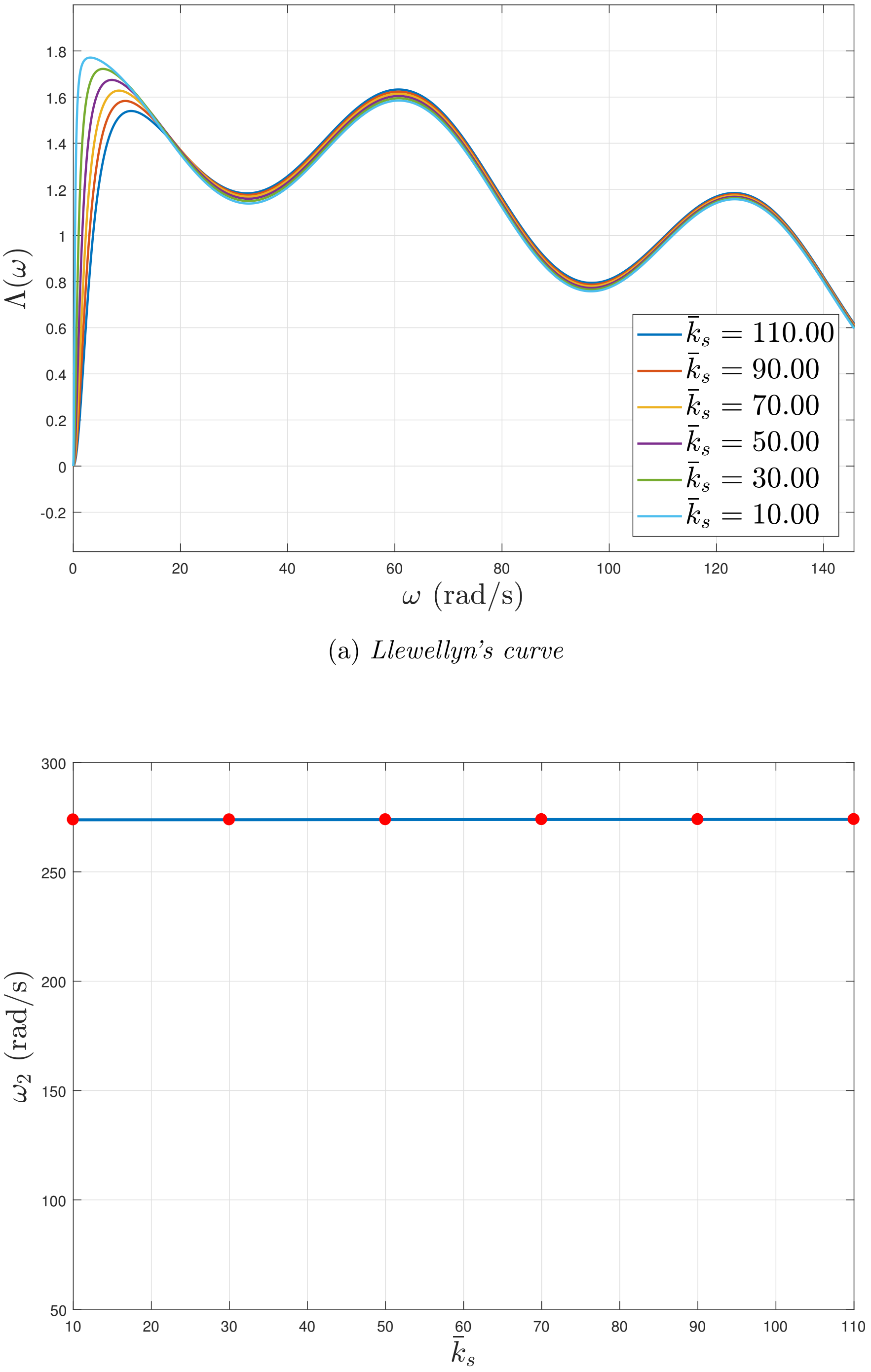
The less significant parameter for the definition of ω_2 is represented by $\bar{k}_s$. As it has been anticipated in the initial crossing subsection, its effect is limited to the low frequency range, having almost no effect on the position of the high frequency bandwidth (Figures 4.9a and 4.9b). That is why it has been used to derive inequality (4.19). By constraining its value, Llewellyn's curve is not modified at high frequency, decoupling the tuning of the parameters and making it more simple.

The last parameter is represented by $\bar{b}_s$ and it is the one with the most complex effect. Its variation mainly modifies the central area of the Llewellyn's curve and its peak shape. For high values of $\bar{b}_s$, $\Lambda(\omega)$ is flattened and the crossing happens early. Instead, if $\bar{b}_s$ is low, the curve has a high peak value, which has however a very narrow frequency range (Figure 4.10a). This means that the crossing frequency also in this situation is low. For this parameter, it makes sense to talk about a trade - off between a low and flat curve with respect to a high and narrow one.

4.2.3 Llewellyn's curve optimization

In the previous subsection it has be roughly understood how the impedance parameters should be tuned in order to obtain a wider absolute stability bandwidth. Now, it is necessary to find the tuning set which *maximizes* the Llewellyn's curve stability region, in order to be as robust as possible with respect to communication time delay. To perform this procedure, it is possible to consider the formulation of ω_2 proposed in (4.22) as a *cost function* for the optimization procedure. Since (4.22) is strongly non linear, it is preferable to perform this optimization off - line, by relying on nonlinear constrained program solvers. For computing the maximum ω_2, `MATLAB` command `fmincon` has been adopted. It is worth noticing that these kind of algorithms require a starting tuning to perform the optimization, and the result they provide is only *locally* optimal.

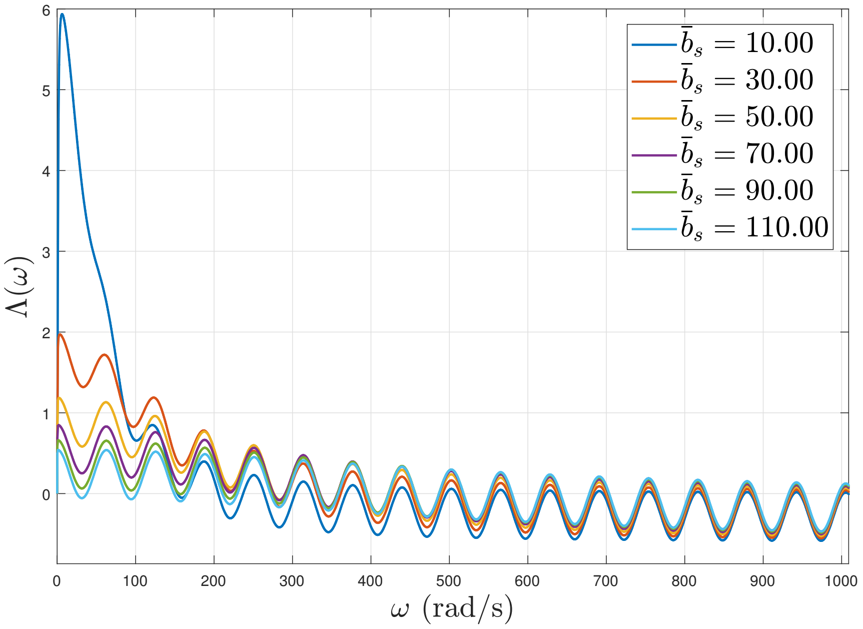
Thanks to the sensitivity analysis, it is possible to guess that the cost function proposed will be *unbounded* in the directions of $\bar{b}_m$ and $\bar{m}_s$. Therefore, to obtain a meaningful result, it is necessary to bound these directions via proper constraints.



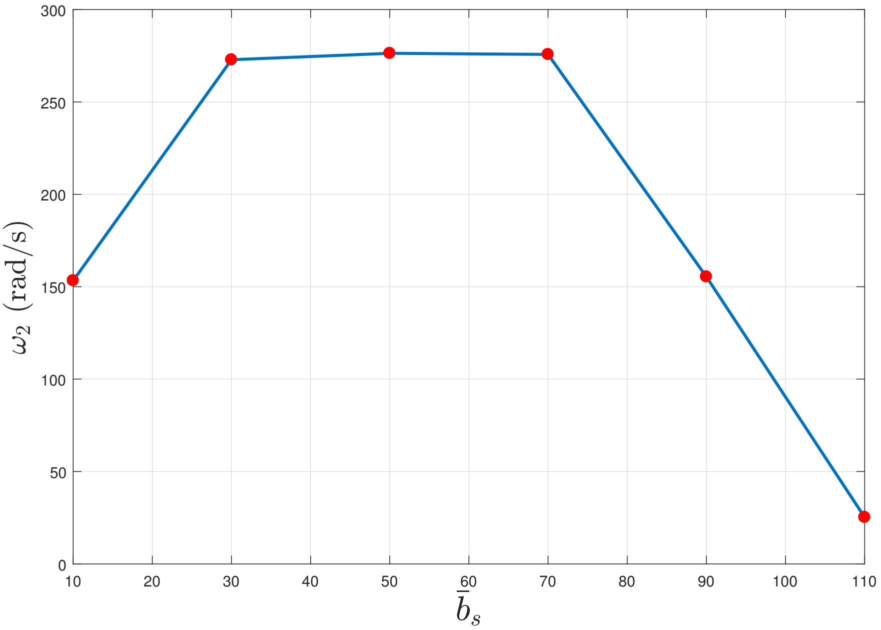
(a) *Llewellyn's curve*

(b) *crossing frequency*

Figure 4.9: Sensitivity analysis with respect to $\bar{k}_s$ variations



(a) *Llewellyn's curve*



(b) *crossing frequency*

Figure 4.10: Sensitivity analysis with respect to $\bar{b}_s$ variations

They will represent the minimum/maximum tuning choices that can be done in order not to compromise teleoperation functionalities and they will be indicated as $\bar{b}_m^{\max}$ and $\bar{m}_s^{\min}$. Furthermore, all the impedance parameters $\bar{b}_m$, $\bar{m}_s$, $\bar{k}_s$ and $\bar{b}_s$ have to be positive by definition.

Putting together these constraints and condition (4.19), it is possible to write the following maximization problem:

$$\begin{aligned} \max_{\bar{b}_m, \bar{b}_s, \bar{k}_s, \bar{m}_s} \quad & \omega_2 \\ \text{s.t.} \quad & 0 < \bar{b}_m \leq \bar{b}_m^{\max}, \\ & \bar{m}_s \geq \bar{m}_s^{\min} > 0, \\ & \bar{b}_s > 0, \\ & 0 < \bar{k}_s \leq \sqrt{\frac{\bar{b}_m \bar{b}_s}{d^2 k_f k_p}} \end{aligned} \tag{4.23}$$

In order to solve (4.23), it is required to perform a nonlinear optimization subject to nonlinear constraints, due to the upper limit of $\bar{k}_s$. Since the available solver does not allow the specification of this type of constraints, it is necessary to *substitute* a feasible value of $\bar{k}_s$ inside ω_2 (e.g. a percentage α of the upper limit). In this way, $\bar{k}_s$ is removed from the optimization parameters and the new problem has only linear constraints:

$$\begin{aligned} \max_{\bar{b}_m, \bar{b}_s, \bar{m}_s} \quad & \omega_2 \Big|_{\bar{k}_s = \alpha \sqrt{\frac{\bar{b}_m \bar{b}_s}{d^2 k_f k_p}}} \\ \text{s.t.} \quad & 0 < \bar{b}_m \leq \bar{b}_m^{\max}, \\ & \bar{m}_s \geq \bar{m}_s^{\min}, \\ & \bar{b}_s > 0 \end{aligned} \tag{4.24}$$

Since (4.24) has less degrees of freedom with respect to (4.23), its maximum solution should be different. However, thanks to the property of $\bar{k}_s$ underlined in Figure 4.9b, it is reasonable to assume that its variations have negligible effect on the position of the maximum bandwidth. Therefore, the optimum of (4.24) will almost coincide with the initial one of (4.23).

4.3 Transparency analysis

As it has been shown in subsection 2.1.2, one of the main performance indicators for a teleoperation system, together with stability, is represented by *transparency*. It indicates the capability of the master/slave network to reproduce faithfully the remote environment to the operator, in order to improve telepresence and task perception.

Taking advantage of the previous stability analysis, it is possible to observe that when there is *no force feedback* coming from the slave to the masted device (i.e. no environmental contact), the teleoperation network is absolutely stable *independently* of the parameters choice (as long as positive). This sentence can be easily proved by imposing k_f equal to zero in the hybrid matrix element H_{12}, obtaining the same expression for the Llewellyn's curve as for the no time delay case (4.16).

Therefore, when there is no contact with the environment, it is convenient to tune system parameters in order to *maximize* the level of transparency of the network. According to Lawrence [44], this can be done by choosing a transmitted impedance Z_t that is as close as possible to the environment one Z_e. In terms of network representations, the *ideal* hybrid matrix that achieves that result corresponds to [2]:

$$\begin{bmatrix} F_h \\ -V_e \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} V_h \\ F_e \end{bmatrix} \tag{4.25}$$

With this choice of $\boldsymbol{H}$, master device shows no impedance on the operator, while the slave tracking impedance is infinite. Doing so, the network *z - width* covers all the frequency spectrum and perfect match between operator and environmental forces and velocities is realised (as if the two entities interact directly). Indeed, $F_h = F_e$ and $V_e = V_h$, with $Z_t = Z_e$. Although this is a very appealing operative condition, it can be easily shown using the Llewellyn's curve that (4.25) is not absolutely stable [33]. If no delay affects the system, $\Lambda(\omega) = 0 \ \forall\omega$ (marginally stable). If the communication channel introduces a delay different from zero, the Llewellyn's curve depicted in Figure 4.11 is obtained, which is always lower or equal to the stability

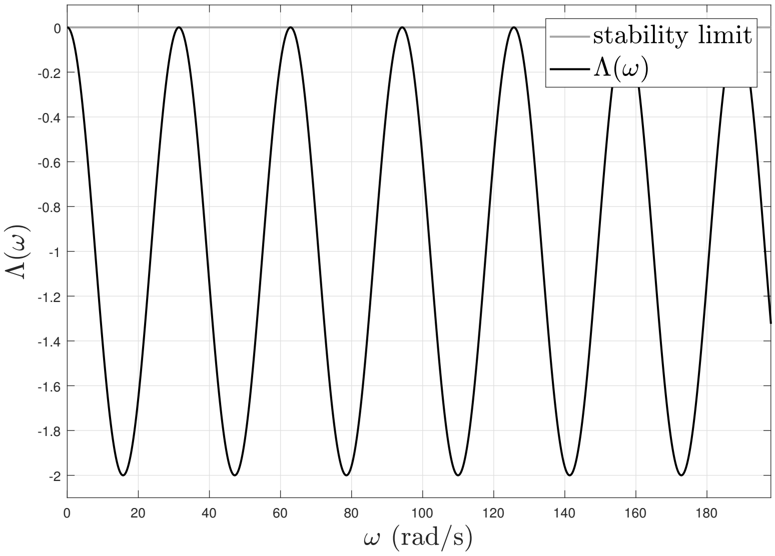


Figure 4.11: Llewellyn's curve for the ideally transparent hybrid matrix

limit:

$$\Lambda(\omega) = \cos 2\omega d - 1 \tag{4.26}$$

This reasoning shows the clear trade - off between transparency and stability that has been introduced in chapter 2.

In order to improve the transparency of the teleoperation network, it is necessary to compute the transmitted impedance Z_t and show its dependency from the hybrid matrix terms. Using the definition presented in subsection 2.1.2:

$$F_h = Z_t V_h \tag{4.27}$$

which is the impedance perceived by the operator when he/she interacts with the teleoperation system. Using network representation 4.3 and substituting F_e with $Z_e V_e$, it is possible to express 4.27 as:

$$F_h = [H_{11} - H_{12} Z_e (1 + H_{22} Z_e)^{-1} H_{21}] V_h \tag{4.28}$$

As an example, let's suppose now that the environment impedance Z_e has the following expression:

$$Z_e = 100 + \frac{600\,000}{s} \tag{4.29}$$

For simplicity, time delay has not be considered in the transparency analysis, since its effect on the magnitude of the transmitted impedance is marginal and therefore it has little influence on the computation of the *z - width*.

First of all, the transmitted impedance of the parameters selected by the optimization solver has been computed. This tuning represents the one which maximizes the network absolute stability bandwidth according to the Llewellyn's curve. As it can be noticed in Figure 4.12 (low impedance line), this choice of slave and master impedances generates a Z_t which is quite far from the ideal one (dashed line). Mismatch is particularly significant for the frequencies at which the teleoperation system usually operates ([0.1,10] $\mathrm{rad\,s^{-1}}$).

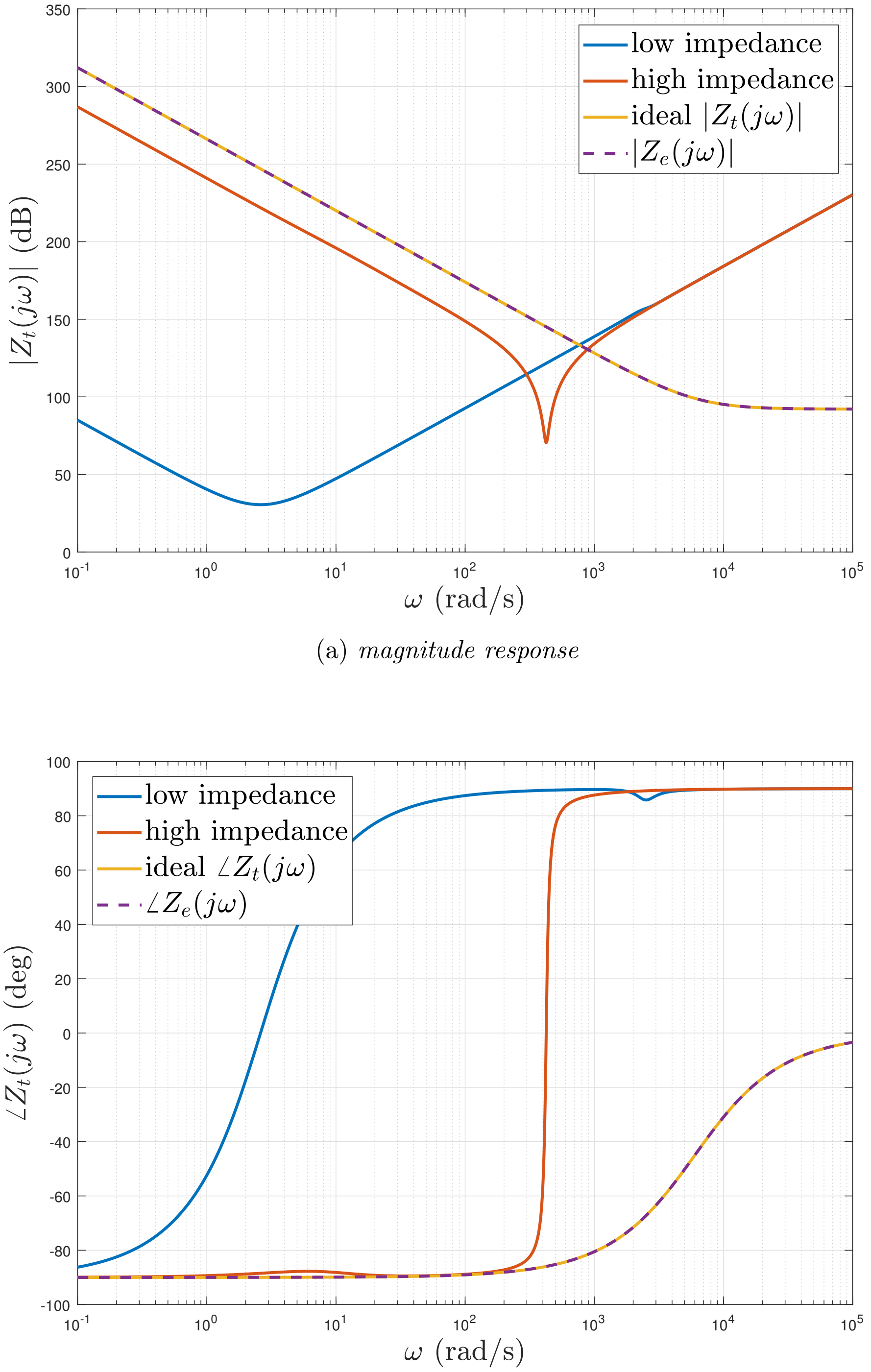
To improve the transparency, the hybrid matrix $\boldsymbol{H}$ should be closer to the ideal one 4.25. Since the haptic interface impedance relationship has been designed in order to fulfil operator's requirements, the best strategy is represented by having the slave admittance as low as possible (ideally zero). To do so, it is possible to *maintain* the same slave poles obtained in the stability optimization process and *scaling up* the entire impedance profile by a constant quantity. The new transmitted impedance is the red line in Figure 4.12. This time, frequency response of Z_t is close to the ideal one for a large set of frequencies, losing transparency only over 500 $\mathrm{rad\,s^{-1}}$ (out of the usual teleoperation excitation bandwidth).

Similar conclusions can be drawn by looking at the *force feedback fidelity*, which is another important parameter for evaluating teleoperation network performance. It is defined as the capability of the bilateral control scheme to replicate faithfully the environmental contact force to the operator:

$$F_h = K_t F_e \tag{4.30}$$

Ideally, it should be $K_t = 1 \ \forall \omega$ and preservation of the whole information generated by the contact force. As for the transparency, also in this case it is possible to express K_t as a function of the hybrid matrix terms:

$$F_h = \left[H_{12} - \frac{H_{11}}{H_{21}} \left(H_{22} + \frac{1}{Z_e} \right) \right] F_e \tag{4.31}$$



(a) *magnitude response*

(b) *phase response*

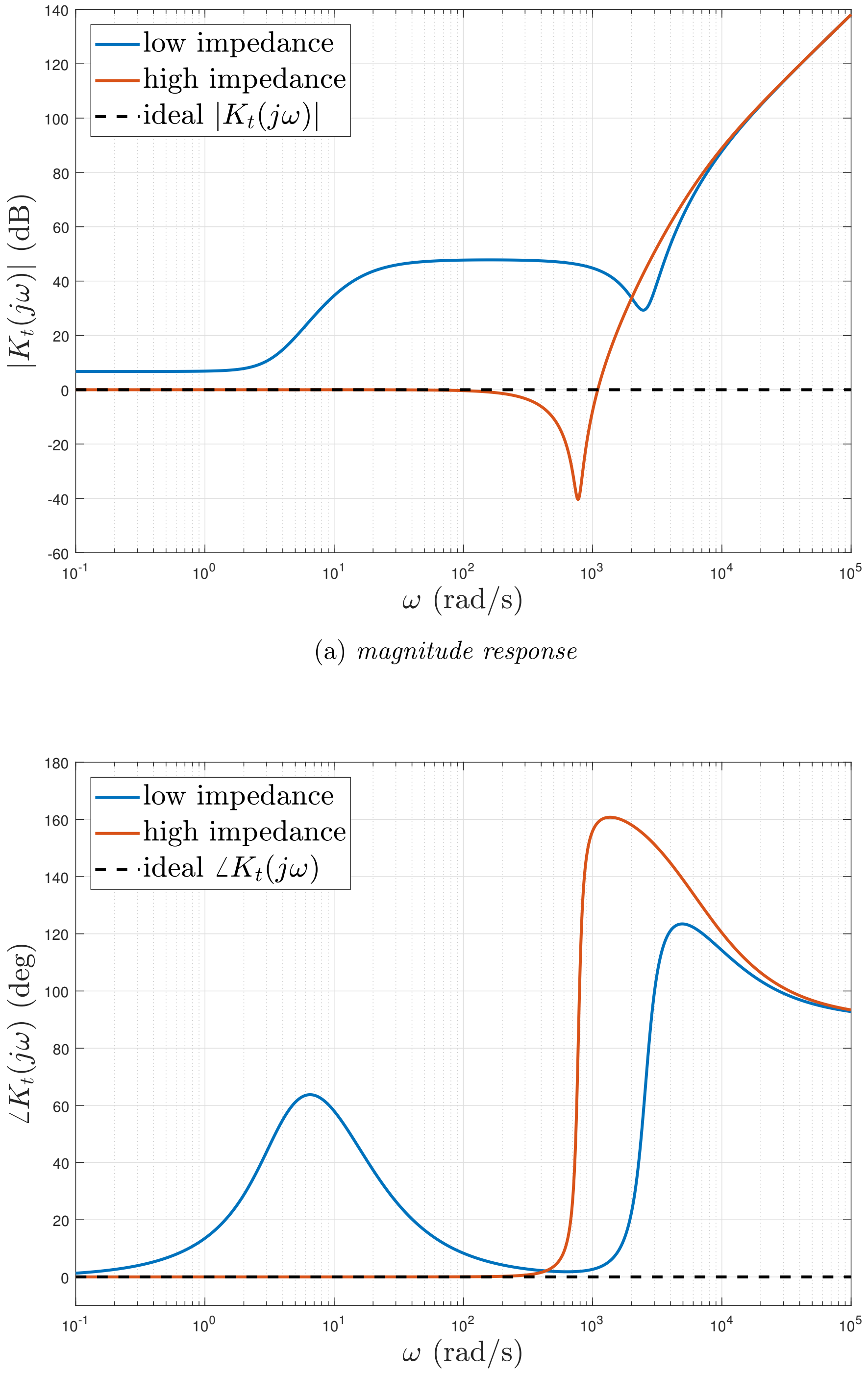
Figure 4.12: Evaluation of the transmitted impedance frequency response for different parameters tuning

The same set of impedance parameters used to compute Z_t has been adopted for evaluating the frequency response of K_t, which is reported in Figure 4.13. In this case, the ideal force transmission is represented by the 0 dB line. As expected, the stable profile impedance (blue line) has a poor force feedback fidelity compared to the one with scaled slave parameters (red line). The closer K_t is to the ideal response, the higher is the capability of distinguishing different environments for the operator (environmental feedback is not distorted).

4.4 Passivity observer

In the previous two sections, different tuning criteria for the slave and master parameters have been adopted, according to the chosen optimization indicator (stability or transparency). Since they are conflictual, it is not possible to achieve a tuning which maximizes both. The strategy of real - time variable impedance that will be implemented in section 4.5 requires *knowledge* of the network stability in order to be effectively applied. A successful method to obtain this information is the implementation of a *passivity observer* [32, 67, 23]. This block has the role of monitoring the energetic flows of the 2 - port network and check if the passivity definition (4.10) holds at each sampling time T. If the bilateral teleoperation network becomes active, the total energy of the system becomes negative and the network is no more guaranteed to be stable. Instability is not mandatory, because passivity is a sufficient but not necessary condition to system stability (introduces conservatism). However, this fact is compensated by the simplicity of implementing the observer.

Let's consider a bilateral teleoperation system which is not affected by time delay ($d = 0$). Thanks to this assumption, it is possible to measure the conjugate variables at the ports of the teleoperation network simultaneously. Then, an energy sign convention is required in order to correctly balance the flows. The one adopted is reported in Figure 4.14 and it agrees with the definitions of network forces and velocities. Now, it is possible to rewrite the passivity definition (4.10) for this 2 -



(a) *magnitude response*

(b) *phase response*

Figure 4.13: Evaluation of the transmitted force scaling frequency response for different parameters tuning

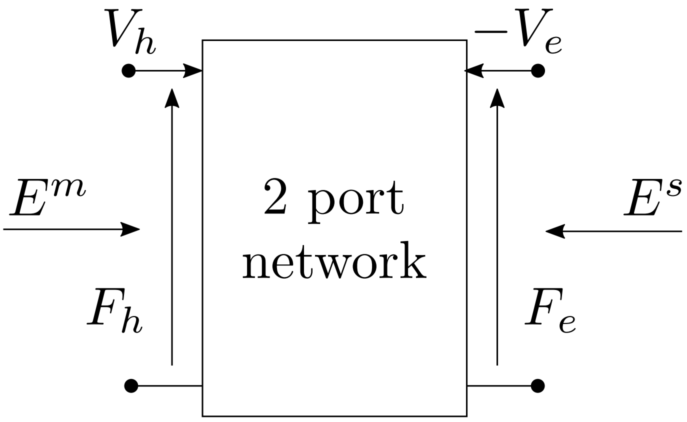


Figure 4.14: 2 port network energy flows scheme

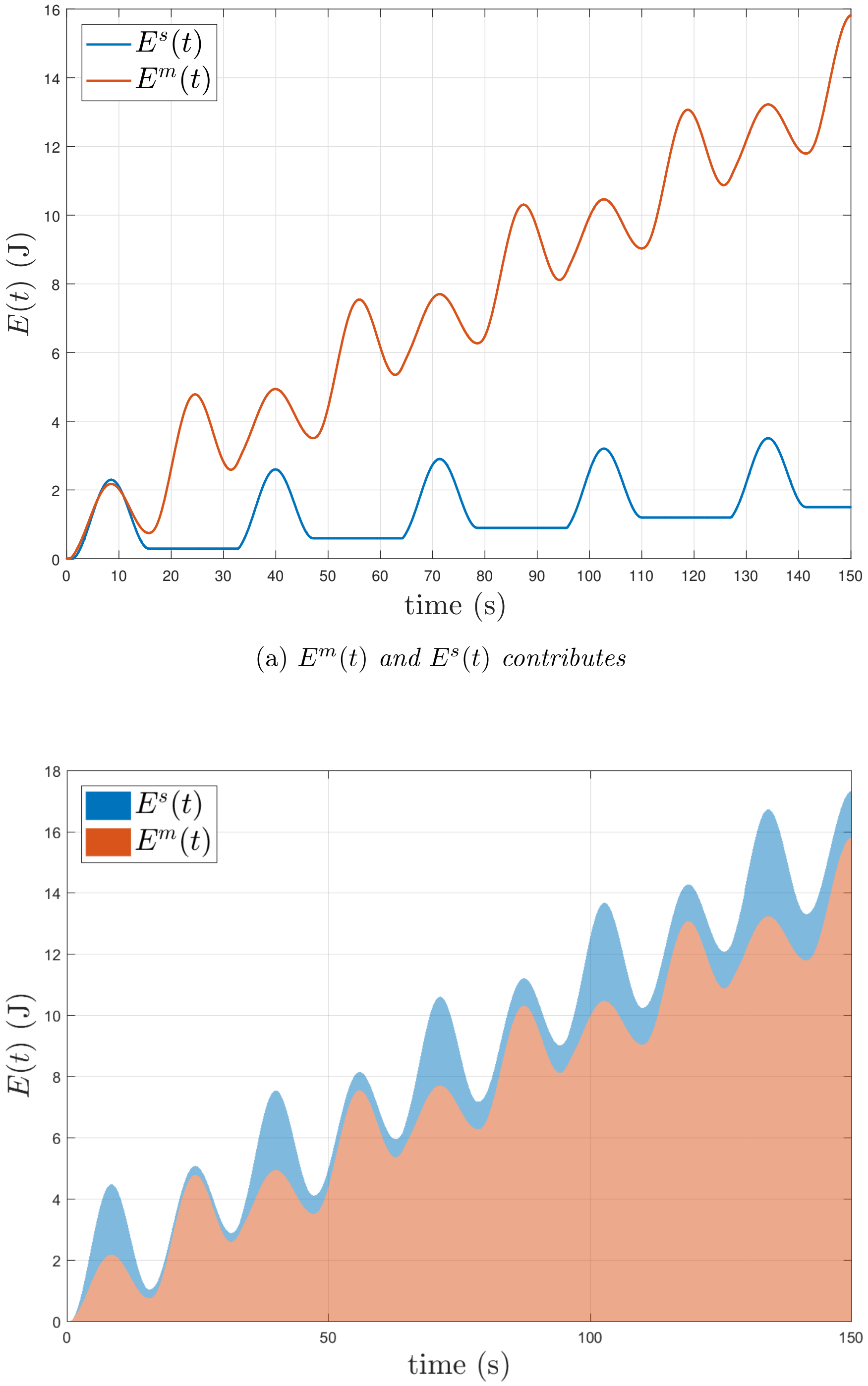
port network:

$$E^m(t) + E^s(t) = \int_{t_0}^{t} (f_h(\tau)v_h(\tau) - f_e(\tau)v_e(\tau))\, d\tau \geq E_0 \quad \forall t \geq t_0 \tag{4.32}$$

where $E^m(t)$ is the energy flow (entering from the master side) generated by the operator, $E^s(t)$ is the energy flow (entering from the slave side) reflected by the environment and E_0 is the initial energy storage of the network.

A visual representation of these two contributions to the total energy coming in the 2 - port network is visible in Figure 4.15a. From this graph, the passive behaviour of the operator and environment can be noticed. The energy reflected by the remote site to the master site is mainly composed by a peak of elastic energy and a constant dissipative term, which increases each time the slave manipulator enters contact with the environment. On the other side, the operator command force generates an energetic trend which is peculiar of the performed task.

Since there is no time delay, these two quantities can be compared in real - time, generating the *resultant* energy flow (Figure 4.15b). To better underline which component of this total flow belongs to the master or slave side, the energy contribution for each port has been shaded. As we would expect, the majority of the energy entering the network is due to the operator, while the reflected energy of the environment represents only a small portion. This is due to the fact that Z_e generates mainly an elastic effect, which is conservative and does not affect the total energy permanently.



(a) $E^m(t)$ *and* $E^s(t)$ *contributes*

(b) *total energy flow*

Figure 4.15: Energy flows monitored by the passivity observer during stable teleoperation task

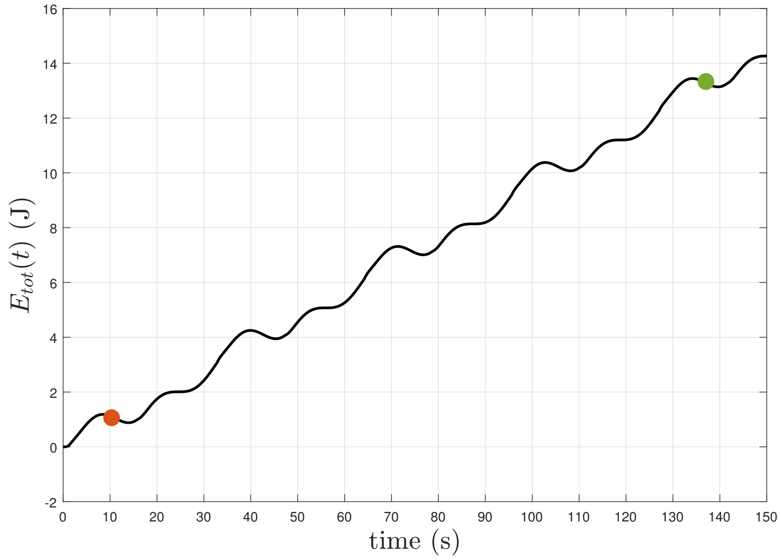


Figure 4.16: Saturation of the observed energy inside the passivity observer

4.4.1 Energy storage saturation

One of the main remarks that can be made by looking at the energetic behaviours observed in Figure 4.15a is the presence of an *increasing trend* generated by the dissipative terms, which causes the energy to drift if the network behaves passively for a long period of time. Consequently, the passivity observer *saturates*, becoming insensitive with respect to system activity. Indeed, let's consider the total energy flow plotted in Figure 4.16. If the active behaviour starts at 10 s, the observed energy is represented by the red dot. Since this point is very close to the stability limit, after a small decrease $E^{tot}(t)$ becomes negative, detecting system potential instability. If the same active behaviour occurs after 135 s (green dot), a greater energy level reduction needs to be observed before instability is detected. This ultimately produces a slow adaptation of the control impedances, leading to large oscillations. Although the system is not yet unstable, we reach a state of practical instability from the operative point of view.

To rapidly detect when the system becomes active, it is better to look at the *energetic trend* instead of the absolute value of $E^{tot}(t)$. When $E^{tot}(t)$ increases, the system is dissipating more energy than the one transmitted to the rest of network. On the other side, when $E^{tot}(t)$ decreases, the system is generating more energy that the one it dissipates. Therefore, saturation of the observed energy does not affect the detection of instability and the time at which it occurs becomes irrelevant. This technique has the advantage of obtaining a *fast detection* of unstable behaviour. Furthermore, lower control effort is required to recover stability, because the oscillatory unstable behaviour has just started. The main drawback of this idea is that it introduces conservatism, due to the stringent requirement with respect to the passivity definition 4.32. Moreover, it is necessary to add an *energy threshold* to avoid false instability detections

4.4.2 Local oscillations

Another issue associated to passivity observers is related to *oscillations* of the calculated total energy E^{tot}. These fluctuations are associated to the variations of the *conservative energy* component of master/slave devices. If a logic based on the energetic trend has been adopted to detect active behaviour of the network (previous subsection), oscillations may deceive the observer, be leading it to think that the system is behaving actively even when it is not in contact with the environment.

This is caused by the fact that a part of the energy injected by the operator into the teleoperation network is *stored* by the master and slave mechanical systems in the form of *kinetic* and *potential* energy. Indeed, the effort done by the human is partially used to accelerate master and slave inertias or load the springs introduced by their control laws.

According to the particular teleoperation task, the devices may decelerate and springs may be unloaded. In this situation, the teleoperation system acts as a *reservoir*, returning such energy to the operator. If this energy flow is added to the one reflected by the environment, it can be easily noticed that the operator observes an outcoming energy which is greater than the incoming one at the remote site. Therefore, the energy trend is decreasing, resembling a network active behaviour. This effect can be appreciated by looking at the oscillations of E^{tot} in Figure 4.15b, mainly generated by conservative energy flows.

To address this problem, it is necessary to look at the various energy contributions which constitute E^m and E^s:

$$\boxed{\frac{1}{2}\bar{m}_m\dot{x}_m^2} + \int \bar{b}_m\dot{x}_m^2 + \boxed{\frac{1}{2}\bar{k}_m x_m^2} + \int k_f f_e^d \dot{x}_m = \int f_h \dot{x}_m = E^m \tag{4.33}$$

$$\boxed{\frac{1}{2}\bar{m}_s\dot{\tilde{x}}_s^2} + \int \bar{b}_s\dot{\tilde{x}}_s^2 + \boxed{\frac{1}{2}\bar{k}_s \tilde{x}_s^2} - \int k_p f_e \dot{x}_m^d = -\int f_e \dot{x}_s = E^s \tag{4.34}$$

the boxed terms of (4.33) and (4.34) represent the kinetic and potential energies stored by the master and slave devices into the teleoperation network. Their contribution to the total energy exchanged at the two ports is *always positive*, therefore they preserve network passivity. Indeed, let's rewrite the total energy computed by the observer:

$$E^{tot} = E^m + E^s = E_c^m + E_{nc}^m + E_c^s + E_{nc}^s \tag{4.35}$$

where the subscript c stands for conservative (boxed) terms, while the remaining ones are labelled with nc. Since it holds that $E_c^m \geq 0\ \forall t \geq t_0$ and $E_c^s \geq 0\ \forall t \geq t_0$, the following implication is true:

$$E_{nc}^m + E_{nc}^s \geq 0 \qquad \Rightarrow \qquad E^m + E^s \geq 0 \tag{4.36}$$

Thus, it is possible to substitute the total observed energy with their non - conservative counterparts in order to verify that the 2 - port network is behaving passively $\forall t \geq t_0$. Ultimately, the inequality that needs to be verified at every time instant is:

$$\int \bar{b}_m\dot{x}_m^2 + \int \bar{b}_s\dot{\tilde{x}}_s^2 \geq \int k_p f_e \dot{x}_m^d - \int k_f f_e^d \dot{x}_m \tag{4.37}$$

where at the left side of the inequality there are the *dissipative terms* associated to master and slave devices, while at the right side there are the *coupling terms* generated by the position reference and the force feedback. It is interesting to notice that in case of no delay affecting the system and 1 - to - 1 teleoperation scaling ($k_f = k_p$), inequality (4.37) degenerates into:

$$\int \bar{b}_m\dot{x}_m^2 + \int \bar{b}_s\dot{\tilde{x}}_s^2 \geq 0 \tag{4.38}$$

which is always fulfilled if the damping coefficients are positive. This proves that if the system is not affected by time delay and it has no kinematic/force scaling, it preserves passivity.

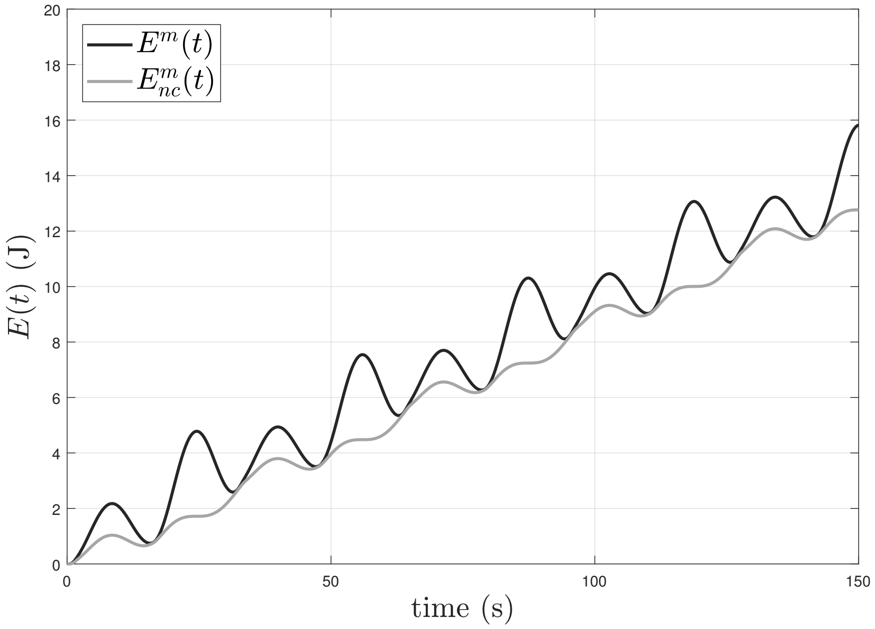
In the general case, condition (4.37) must be checked in real time to monitor the behaviour of the network. In Figures 4.17a and 4.17b it is possible to compare the total energy flow with the one of the non conservative contribution. As it can be noticed, the oscillations associated to the conservative terms are no more present and only the dissipative and coupling contributions affect the observer. Therefore, any drop in the new observed E_{nc}^{tot} is generated only when the environmental force transmits more energy than the one the master and slave devices are able to dissipate.

Since the implication (4.36) is only sufficient, the passivity condition (4.37) is more stringent than the theoretical one (4.10), but it allows a fast detection of active behaviour, which is crucial in order to guarantee system stability for every contact interaction.

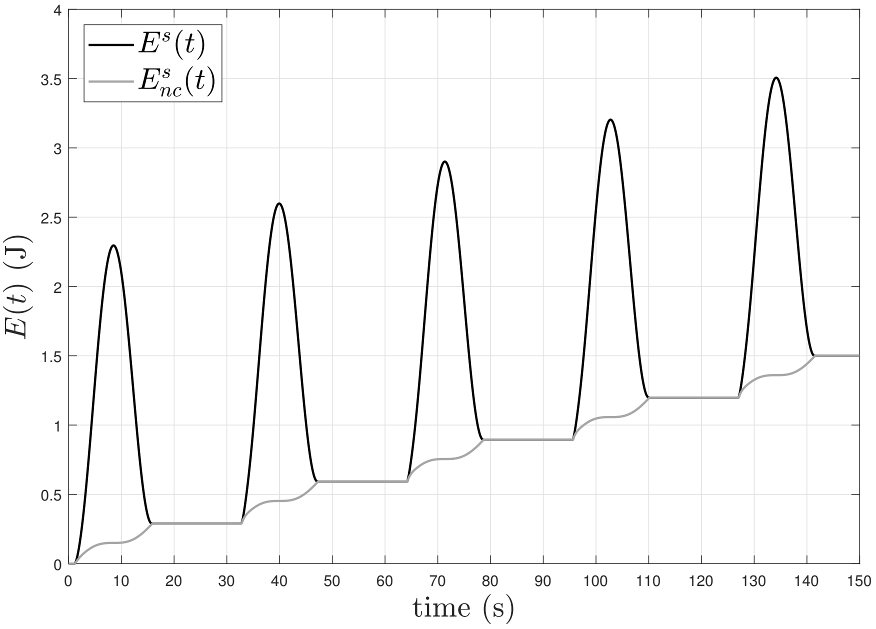
To evaluate (4.37) in presence of time delay, it is necessary to estimate the value of d, otherwise it is not possible to compare at the same time $k_p f_e \dot{x}_m^d$ and $k_f f_e^d \dot{x}_m$. If this is not feasible (variable, unknown time delay), the observed energy cannot be real - time estimated and it is necessary to adopt another strategy.

4.4.3 Time delay implementation

Up to now, it has been considered a centralised passivity observer, which is able to measure at the same time the conjugate variables at each port of the network. Unfortunately, this is usually not the case, because teleoperation systems are distributed and it is not possible to have access at both terminals simultaneously. Therefore, E^s is only available at the slave side while E^m at the master side. To overcome this issue, the solution proposed in [67] can been adopted. It consists in *splitting* the energy contributions at each port in an outcoming and an incoming flow, as depicted

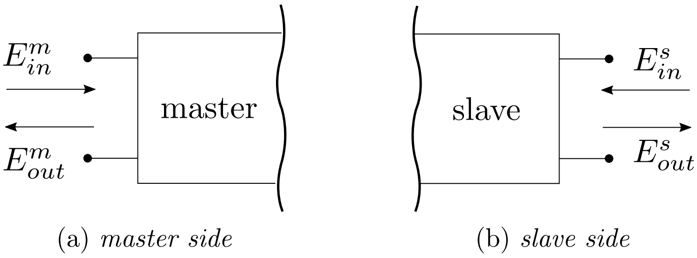


(a) *master side*



(b) *slave side*

Figure 4.17: Observed energy at the master and slave sides with and without the conservative energy contributions



(a) *master side* (b) *slave side*

Figure 4.18: Split of energy flows at the master and slave ports

in Figure 4.18. Doing so, it is possible to write the following two equalities:

$$\begin{aligned} E^m &= E^m_{in} - E^m_{out} \\ E^s &= E^s_{in} - E^s_{out} \end{aligned} \tag{4.39}$$

where $E^{m,s}_{in} \geq 0$ and $E^{m,s}_{out} \geq 0$ are the absolute values of the incoming and outcoming energies. Then, let's substitute them into the passivity definition:

$$E^m_{in} - E^m_{out} + E^s_{in} - E^s_{out} \geq 0 \tag{4.40}$$

Now, the main idea is to *divide* the single passivity observer into two blocks, one at the master side and one at the slave side. They are responsible for the passivity of their respective ports and the inequalities they need to satisfy are the followings:

$$\begin{aligned} E^s_{out} &\geq E^{md}_{in} \qquad \text{slave side} \\ E^m_{out} &\geq E^{sd}_{in} \qquad \text{master side} \end{aligned} \tag{4.41}$$

where the superscript d for the incoming energies means that they are delayed. The reasoning behind (4.41) is quite immediate. At each port, the outcoming energy is compared to the incoming one on the other port d instants before. If this energy is increased, it means that the network has generated it and therefore it behaves actively. More formally, it is possible to prove that conditions (4.41) represents a sufficient condition for the global network passivity [67].

These two passivity observers are very interesting because they can be implemented without any a - priori knowledge of the time delay affecting the system. In fact, incoming energy information can be sent through the communication channel together with the signals, and E_{in}^{md}, E_{in}^{sd} will be delayed by the exact amount of time that affects the network. The main drawback of this architecture is that the impedance shaping cannot be performed only on the slave side, but it is necessary to modify also the impedance profile of the haptic interface if the master observer detects an active behaviour of the port.

4.5 Variable impedance controller

After having analysed the stability and transparency of the teleoperation 2 - port network, it is possible to specify the main control objectives for the entire system in a more rigorous way. Due to the trade - off between stability and transparency, to achieve *safe* environmental interaction and *good* telepresence, it is not possible to use a single parameter tuning set. This means that, to obtain the maximum from both indicators, it is necessary to implement a *variable impedance* system, which is able to optimize one of the two indicators according to the given operative condition of the teleoperation system.

In the following simulations, it has been assumed that the network is affected by a time delay d of 1 s, constant and known. Therefore, it is possible to implement a centralised passivity observer using the strategies presented in subsections 4.4.2 and 4.4.1. Based on the information obtained through the passivity observer, it is possible to define in which condition the teleoperation system is working and decide the best tuning strategy that should be adopted in order to maximize one of the two performance indicators. Since it is desirable to keep constant the tuning parameters of the haptic interface to reduce the influence of the control strategy on the operator, impedance adaptation is realised on the slave one, according to the scheme proposed in Figure 4.19.

When the system is not in contact with the environment, slave parameters should be tuned in order to obtain high transparency. Indeed, during free motion, stability is guaranteed because there is no force feedback. Therefore, it is possible to scale up the slave impedance profile in order to obtain good tracking accuracy, external disturbance rejection, etc. When the slave enters contact with the environment, two

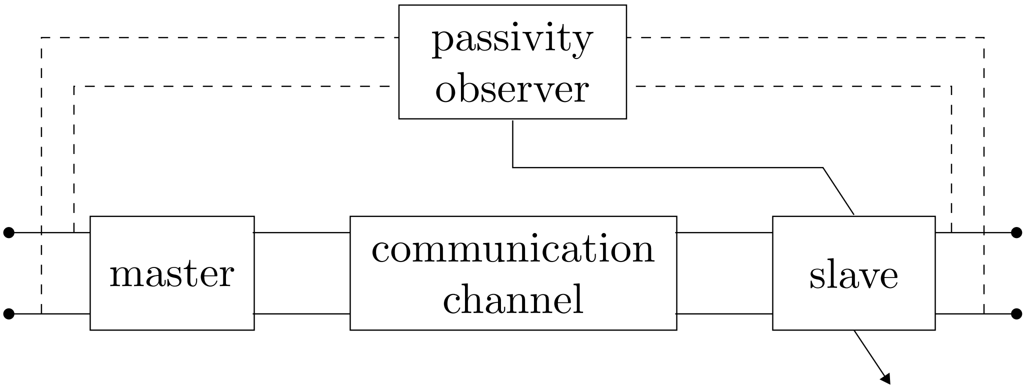


Figure 4.19: Variable impedance control scheme architecture

situations may occur:

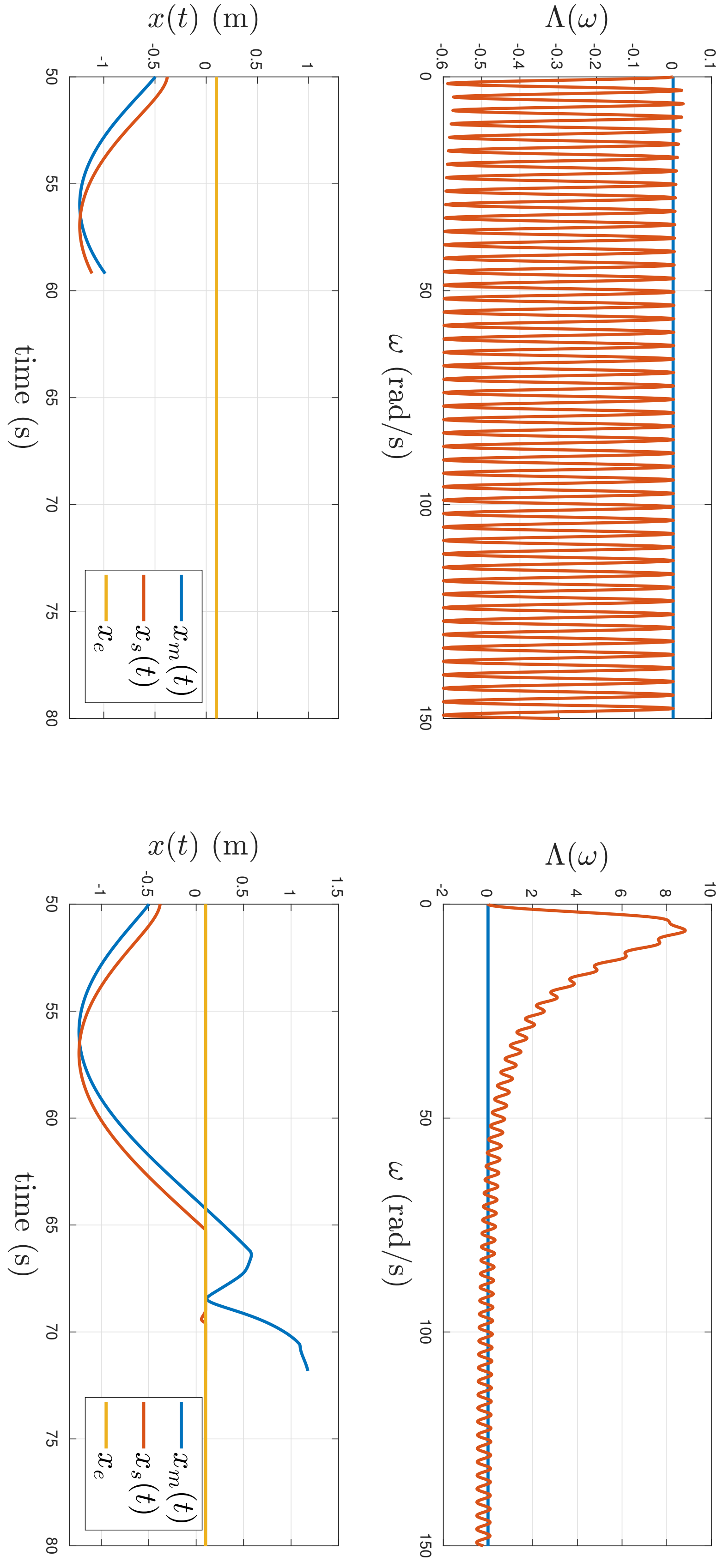
- The system is still passive, therefore it is possible to continue maximizing the transparency in order to render properly Z_e to the operator;
- The system starts to behave actively (typical when the slave manipulator enters in contact with stiff environments). In this case, it is mandatory to scale down slave impedance parameters in order to increase Llewellyn's ω_2 and regain stability as fast as possible;

To apply the previous reasoning, two paths have been followed.

4.5.1 Continuous adaptation

Slave impedance is downscaled *proportionally* to the degree of activity measured by the passivity observer. The slave device is initially tuned based on an high impedance profile in order to have a good transparency, and the value of its parameters is kept constant until the system starts to behave actively. After that, the main idea is to use the amount of active energy generated by the system as a reference signal for the on - line adaptation of the slave impedance. To do so, it is necessary to properly condition its value in order to obtain the correct reaction for each amount of active energy generated. When the system regains passivity, the high impedance profile is restored to regain transparency.

This method does not take into account if the slave device is in contact with the environment or not, therefore Z_e can be rendered correctly if the system is passive



(a) *improve network transparency*

(b) *improve contact stability*

Figure 4.20: Continuously adaptive impedance control: slave and master position profiles and Llewellyn's curve for two different operative conditions

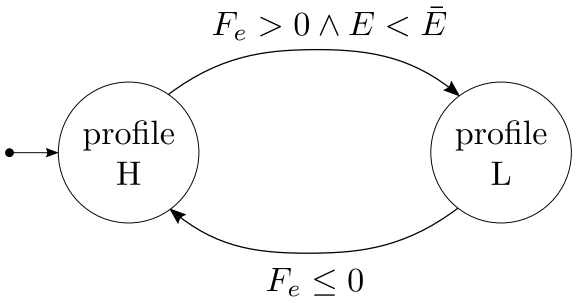


Figure 4.21: Two state adaptation control logic for the variable impedance slave controller

during the initial contact instants. Evolution of master and slave positions and Llewellyn's curve variable shape are plotted in Figure 4.20. In the first couple of figures, the system is performing a free motion task and transparency is maximized. This can be appreciated by the fact that the Llewellyn's curve is almost always below the stability limit. Instead, when the system starts behaving actively, tuning is changed and the Llewellyn's curve raises, obtaining a stable contact interaction. The backlash that occurs to the master device when the slave enters contact with the environment is due to the high transparency, which reflects the initial force impulse to the haptic interface.

4.5.2 Two - state adaptation

In this case, the number and shapes of the various slave impedance profiles is fixed. Two possible parameters tuning are generated off - line, following the two design principles for high transparency and stability:

- **Profile H**: it guarantees optimal telepresence and high fidelity free motion for the teleoperation network;

- **Profile L**: it assures stability during environmental contact and it damps potential active behaviour of the network;

Then, the profile choice is regulated using a *finite state machine* of the type proposed in Figure 4.21. The criteria chosen to switch between one profile to another reflect the desire of having a good transparency also for the initial contact instants. Indeed, profile H is maintained until the slave is in contact with the environment and

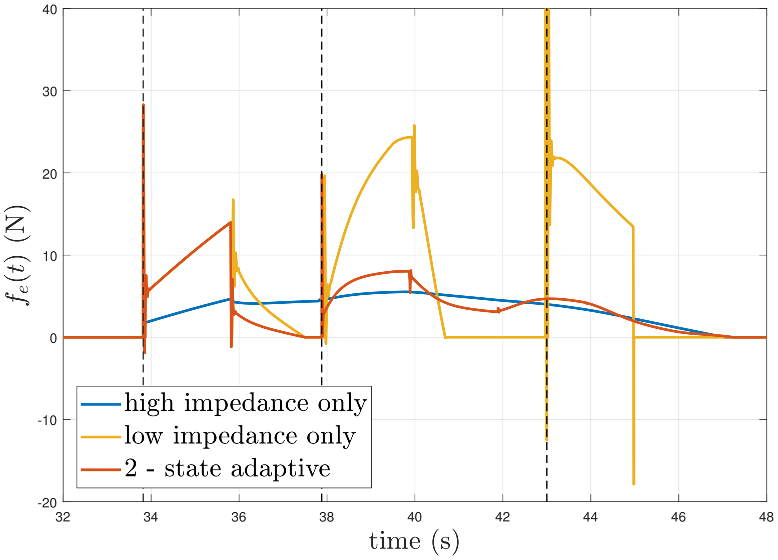
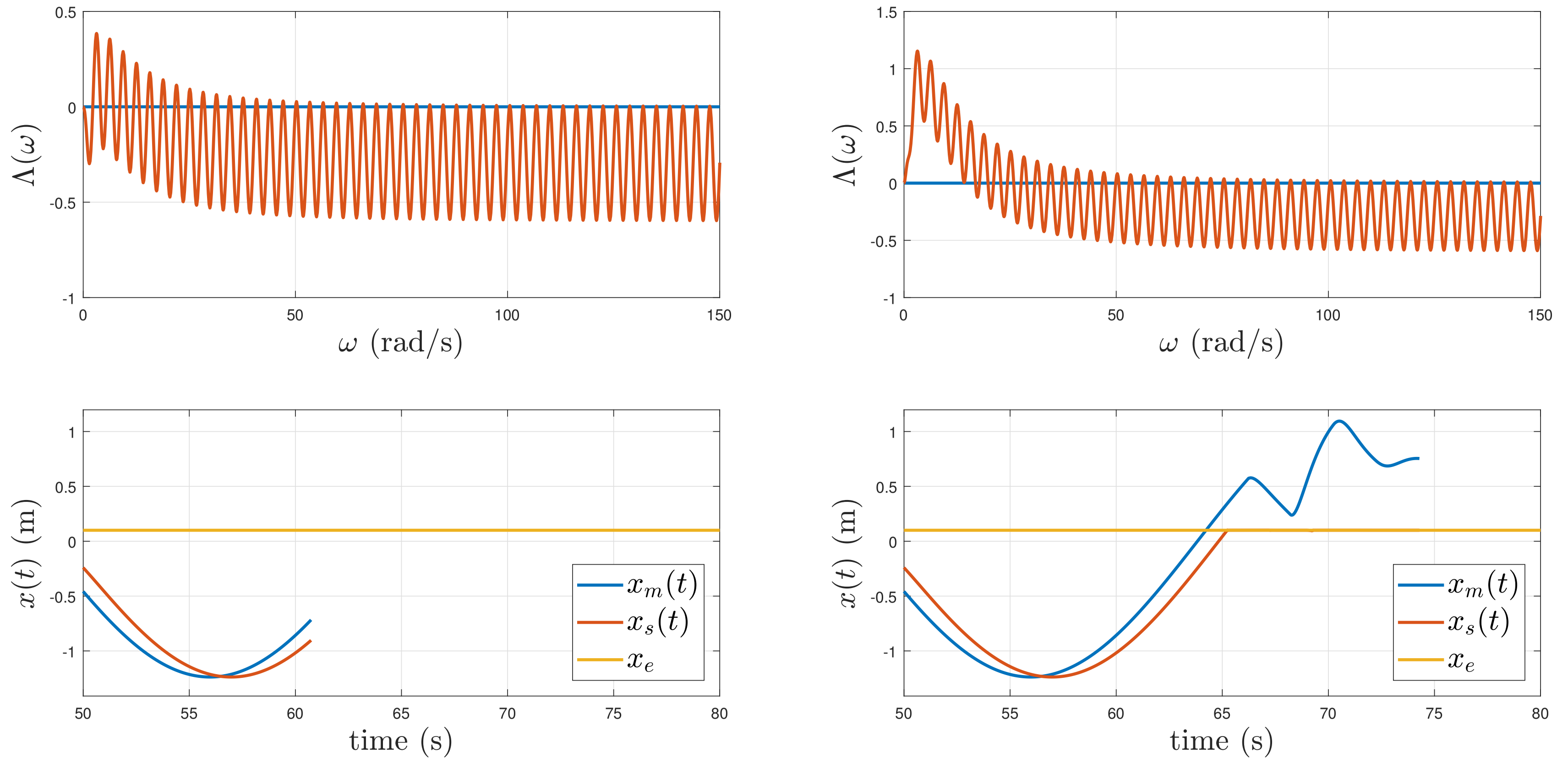


Figure 4.22: Contact force comparison between different slave impedance tunings: dashed lines represent wall impacts

the system is sufficiently passive. This last condition adds a new tuning parameter $\bar{E}$, that should be carefully chosen in order to avoid instability while maintaining a significant transparency phase. Once profile L is selected, passivity is no more checked. This has been done in order to avoid the oscillations observed in the continuous adaptive scheme, caused by the interaction between the energy flows and the variable impedance mechanism. Profile H is restored only when there is no more contact with the environment, assuring the stability of the network.

In Figure 4.23, it is possible to observe the same experiment performed for the continuous adaptation case using the 2 - state switching control logic. In this case, the two operative conditions shown in 4.23a and 4.23b describe the two different impedance profiles that have been selected off - line. The second one has been computed by optimizing ω_2 according to subsection 4.2.3, while the other is its scaled up version (optimized transparency). As it can be noticed, in this case the admissible tuning range is well defined, generating more predictable system response without modifying excessively the slave tuning.

The environmental force generated during the experiment is the red line shown in Figure 4.22. It has been compared with the ones that would be obtained if a fixed impedance profile were selected. The choice of an L profile for the entire teleopera-



(a) *improve network transparency*

(b) *improve contact stability*

Figure 4.23: 2 - state adaptive impedance control: slave and master position profiles and Llewellyn's curve for two different operative conditions

tion task generates obviously a stable response of the system, but the force seen by the operator is quite smoothed, with a bad rendering of the external environment. The opposite situation happens for the H profile, which renders correctly the impulsive initial force, but then it starts to bounce due to the low stable bandwidth. The 2 state approach takes advantage of both behaviour, correctly rendering the contact force at the initial time steps and then smoothing it out to avoid bounces and oscillations.

The two - state adaptation is better than continuous one under several aspects. It allows off -line optimization and better usage of the tuning parameters, which are defined a - priori. This results in an easy tuning process and a more controlled evolution of the network state. Moreover, it does not suffer from oscillations due to fast passive - active switching, which instead affect the continuous adaptation procedure. Different criteria can be easily implemented in the finite state machine to choose proper switching logics.

Chapter 5

N - DOFs extension

An extension of the discussion presented in chapter 3 for multi - DOFs mechanical systems is reported here. In particular, it is shown how to implement sliding mode controllers when inverse dynamics are applied to a robotic manipulator, generating a chain of double integrators affected by uncertainty.

The first possibility is to perform a *sequential SMC* for each joint, using a decoupled sliding surface to cope with joint uncertainties independently. Then, an *operational space* impedance controller is introduced, in order to directly define the end - effector desired behaviour. To deal with the uncertainties affecting this controller, a new coupled sliding surface has been implemented.

Finally, the analysis is extended to *redundant manipulators*, where there is not a unique mapping between joint and operational spaces. In this case, optimization procedures have been implemented to derive the equivalent control law, while the sliding surface has been extended to remove uncertainties also for the *null space motion*.

All these results have been tested in simulation on a 2 - DOF manipulator and then on a simulated model of `ABB YuMi` 7 - DOF right arm.

5.1 Decentralized SMC

First of all, let's consider the generic n-DOFs model for the slave manipulator introduced in subsection 2.2.2:

$$\boldsymbol{B}_s(\boldsymbol{q}_s)\ddot{\boldsymbol{q}}_s + \boldsymbol{n}_s(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s) = \boldsymbol{\tau}_s - \boldsymbol{J}_s^\top(\boldsymbol{q}_s)\boldsymbol{h}_e \tag{5.1}$$

where $\boldsymbol{n}_s(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s) = \boldsymbol{C}_s(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{q}}_s + \boldsymbol{g}_m(\boldsymbol{q}_s)$.

After applying the inverse dynamics control law (section 3.1), we come up with the following perturbed chain of double integrators:

$$\ddot{\boldsymbol{q}}_s = \boldsymbol{\zeta}_s\boldsymbol{v} + \boldsymbol{\eta}_s \tag{5.2}$$

where $\boldsymbol{\zeta}_s = \boldsymbol{B}_s^{-1}\hat{\boldsymbol{B}}_s$ and $\boldsymbol{\eta}_s = \boldsymbol{B}_s^{-1}[\boldsymbol{\Delta n}_s + \boldsymbol{J}_s^\top\boldsymbol{\Delta h}_e]$.

By looking at expression (5.2), it can be noticed that each joint acceleration $\ddot{q}_{si} \;\; i = 1 \dots n$ is related to the other ones through the coupling terms generated by imperfect dynamic cancellation. If it is possible to assume that $\boldsymbol{\zeta}_s \approx \boldsymbol{1}$ and $\boldsymbol{\eta}_s \approx \boldsymbol{0}$ (model knowledge is accurate enough), we can treat coupling terms as *disturbances*, obtaining the following "decoupled" sequence:

$$\begin{aligned} \ddot{q}_{s1} &= v_1 + d_1 \\ &\vdots \\ \ddot{q}_{sn} &= v_n + d_n \end{aligned} \tag{5.3}$$

The idea of splitting a single complex control design into several simple ones is known in literature as *decentralized control* [73]. Then, it is possible to design n independent control laws, based only on the error associated to each separate joint. Thus, it is sufficient to iterate the process seen for the 1 DOF case through all the equations in (5.3).

In order to reject disturbance d_i, we can take advantage of the SMC already introduced, while the desired joint dynamics can be specified independently for each joint:

$$\bar{\boldsymbol{B}}\ddot{\tilde{\boldsymbol{q}}} + \bar{\boldsymbol{C}}\dot{\tilde{\boldsymbol{q}}} + \bar{\boldsymbol{g}}\tilde{\boldsymbol{q}} = -\boldsymbol{J}_s(\boldsymbol{q})^\top\boldsymbol{h}_e \tag{5.4}$$

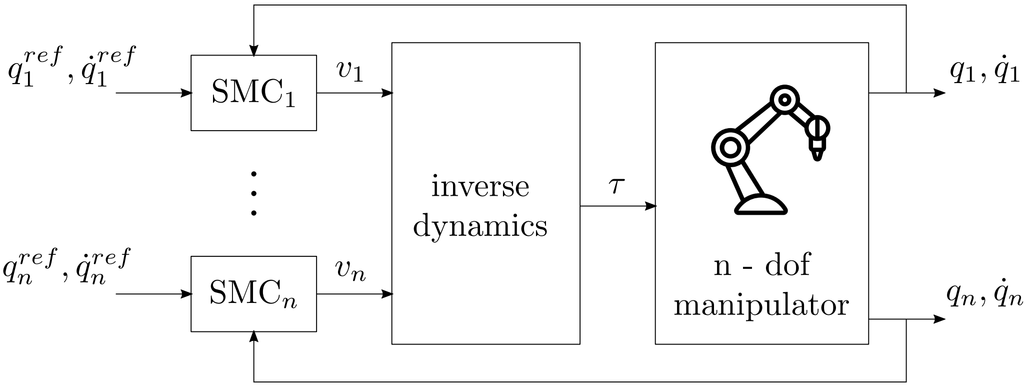


Figure 5.1: Control scheme of a decentralized SMC controller for multi - DOFs manipulators

where $\bar{\boldsymbol{B}} = \mathrm{diag}(b_1, \ldots, b_n)$, $\bar{\boldsymbol{C}} = \mathrm{diag}(c_1, \ldots, c_n)$ and $\bar{\boldsymbol{g}} = \mathrm{diag}(g_1, \ldots, g_n)$. Tracking error definition in joint space is totally analogous to the one proposed in subsection 3.2.2:

$$\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{k}\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{d}} \tag{5.5}$$

It can be noticed that equation (5.4) is not fully decoupled when the manipulator is in contact with the environment, due to the presence of $\boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{q})^\top \boldsymbol{h}_{\boldsymbol{e}}$. However, if differential kinematics is assumed to be known almost exactly, this contribution can be easily computed, resulting in an additional torque τ_i for each individual joint.

The resultant manipulator's sliding surface $\boldsymbol{\sigma}$ can be seen as the superimposition of n joints sliding manifolds:

$$\boldsymbol{\sigma} = \bar{\boldsymbol{B}}\dot{\tilde{\boldsymbol{q}}} + \bar{\boldsymbol{C}}\tilde{\boldsymbol{q}} + \int_0^t (\bar{\boldsymbol{g}}\tilde{\boldsymbol{q}}(\boldsymbol{\tau}) + \boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{q}(\boldsymbol{\tau}))^\top \boldsymbol{h}_e) d\tau \tag{5.6}$$

while the equivalent control is simply represented by the vectorial formulation of the standard impedance control law:

$$\boldsymbol{v}^{\boldsymbol{eq}} = -\bar{\boldsymbol{B}}^{-1}\bar{\boldsymbol{C}}\dot{\tilde{\boldsymbol{q}}} - \bar{\boldsymbol{B}}^{-1}\bar{\boldsymbol{g}}\tilde{\boldsymbol{q}} - \bar{\boldsymbol{B}}^{-1}\boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{q})^\top \boldsymbol{h}_e + \boldsymbol{k}_{\boldsymbol{p}}\ddot{\boldsymbol{q}}_{\boldsymbol{m}}^{\boldsymbol{d}} \tag{5.7}$$

Since each control loop is almost a standalone entity, convergence property can be proved for each joint *independently* and conditions on sliding gain matrix $\boldsymbol{K}$ similar to the one presented in theorem 3.2 can be drawn. The overall control scheme is shown in Figure 5.1.

Up to now, all the advantages of a decentralized architecture have been listed. Unfortunately, it presents also some major drawbacks:

- It can require **high values** of SMC gains in order to reject disturbances coming from inexact inverse dynamics. Moreover, uncertainties are coupled and vary with robot configuration, making the worst case scenario very difficult to figure out;

- The desired dynamic parameters b_i, c_i, g_i are expressed in the **manipulator's joint space**, therefore with a decentralized control scheme it is not possible to impose a determined impedance relationship between the end - effector and the external environment. This causes a *configuration dependant impedance*, because the behaviour seen in the operational space becomes the sum of all impedance contributes, weighted by the Jacobian matrix. End - effector dynamic relationship is *variable* with respect to the robot configurations, reaching very big values when robot is close to singularities. This can be considered a negative aspect if the objective is to keep constant end - effector impedance;

For these reasons, sequential SMC extension can be suitable only for *slave device*, because master haptic interfaces require known and constant end - effector impedance value. Instead, the remote manipulator desired parameters define an "error dynamics" with respect to the master reference joint positions, which can be uncorrelated with the operational space.

Nevertheless, the choice of sequential controllers for slave devices generates *stability issues* when environmental contact is taken into account. This because, in some configurations, variable end - effector impedance could cause very high interaction forces.

5.2 Operational space SMC

To overcome the decentralized architecture problems, it is necessary to define control objectives directly in the manipulator operational space. Doing so, it is possible to guarantee a known and constant end - effector dynamics, with clear physical - related parameters (mass, damping, stiffness).

As usual, we start from the inverse dynamics closed - loop equation (5.2), referred, this time, to the master device:

$$\ddot{\boldsymbol{q}}_{\boldsymbol{m}} = \boldsymbol{\zeta}_{\boldsymbol{m}} \boldsymbol{v} + \boldsymbol{\eta}_{\boldsymbol{m}} \tag{5.8}$$

where $\boldsymbol{\zeta}_{\boldsymbol{m}} = \boldsymbol{B}_{\boldsymbol{m}}^{-1} \hat{\boldsymbol{B}}_{\boldsymbol{m}}$ and $\boldsymbol{\eta}_{\boldsymbol{m}} = \boldsymbol{B}_{\boldsymbol{m}}^{-1}[\boldsymbol{\Delta n}_{\boldsymbol{m}} - \boldsymbol{J}_{\boldsymbol{m}}^{\top} \boldsymbol{\Delta h}_{\boldsymbol{h}}]$.

Contrary to the previous subsection, this time the assumption of joint decoupling is not taken into account. This does not mean that this scheme is able to compensate for a higher level of uncertainty with respect to the decentralised scheme, but it simply states that the operational control law will be a *centralized* one (i.e. $\boldsymbol{v}$ is defined using all joint information). The desired impedance profile required for the haptic interface can be extended from the scalar formulation (equation (3.6)) to the n - dimensional space:

$$\bar{\boldsymbol{M}} \ddot{\boldsymbol{x}} + \bar{\boldsymbol{D}} \dot{\boldsymbol{x}} + \bar{\boldsymbol{K}} \boldsymbol{x} = \boldsymbol{h}_{\boldsymbol{h}} - \boldsymbol{k}_{\boldsymbol{f}} \boldsymbol{h}_{\boldsymbol{e}}^{\boldsymbol{d}} \tag{5.9}$$

where $\bar{\boldsymbol{M}} = \operatorname{diag}(m_1, \ldots, m_n)$, $\bar{\boldsymbol{D}} = \operatorname{diag}(d_1, \ldots, d_n)$ and $\bar{\boldsymbol{K}} = \operatorname{diag}(k_1, \ldots, k_n)$. According to the degrees of freedom of the manipulator, it is feasible to impose a desired impedance profile to:

- Single spatial coordinates (1 DOF case already studied);
- x, y, z position (at least 3 DOFs);
- Full end - - effector position and orientation (at least 6 DOFs);

By choosing one of the previous three cases, the definitions of $\boldsymbol{x}$, $\boldsymbol{h}_{\boldsymbol{h}}$ and $\boldsymbol{h}_{\boldsymbol{e}}$ should be adapted consequently. For example, a full position and orientation profile requires:

$$\begin{aligned}
\boldsymbol{x} &= \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^{\top} \\
\boldsymbol{h}_{\boldsymbol{h}} &= \begin{bmatrix} f_{hx} & f_{hy} & f_{hz} & \tau_{h\phi} & \tau_{h\theta} & \tau_{h\psi} \end{bmatrix}^{\top} \\
\boldsymbol{h}_{\boldsymbol{e}} &= \begin{bmatrix} f_{ex} & f_{ey} & f_{ez} & \tau_{e\phi} & \tau_{e\theta} & \tau_{e\psi} \end{bmatrix}^{\top}
\end{aligned}$$

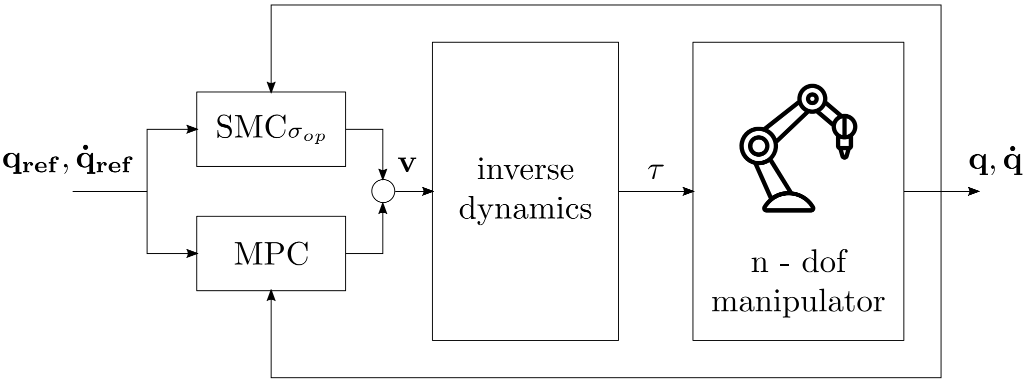

$\mathbf{q_{ref}}, \dot{\mathbf{q}}_{\mathbf{ref}}$
SMC$_{\sigma_{op}}$
MPC
$\mathbf{v}$
inverse dynamics
τ
n - dof manipulator
$\mathbf{q}, \dot{\mathbf{q}}$

Figure 5.2: Centralized control structure for the operational space impedance controller

where (ϕ, θ, ψ) are the Euler angles associated to the particular representation chosen to describe end - - effector orientation.

Given equation (5.9), the objective is to design a vectorial controller which is able to obtain the desired impedance while rejecting uncertainties coming from $\boldsymbol{\eta_m}$ and $\boldsymbol{\zeta_m}$. To perform this task, an SMPC architecture of the type presented in section 3.8 is implemented. It offers the advantage of a flexible equivalent control definition through a constrained minimization stage, while retaining SMC robustness properties. An overview of the control scheme architecture is visible in Figure 5.2. The sliding surface is a unique coupled vectorial expression and will be indicated with the symbol $\boldsymbol{\sigma_{op}}$.

5.2.1 Centralized sliding surface

Following the standard steps introduced in subsection 3.3.1 for 1 DOF sliding mode control design, it is necessary to introduce the vectorial sliding manifold $\boldsymbol{\sigma_{op}}$ starting from the desired operational space impedance profile (equation (5.4)):

$$\boldsymbol{I} = \boldsymbol{\bar{M}\ddot{x}} + \boldsymbol{\bar{D}\dot{x}} + \boldsymbol{\bar{K}x} - \boldsymbol{h_h} + \boldsymbol{k_f h_e^d} \tag{5.10}$$

To remove acceleration dependency, integration on both sides is performed, obtaining the vectorial form of equation (3.27):

$$\boldsymbol{S} = \int \boldsymbol{I} = \boldsymbol{\bar{M}\dot{x}} + \boldsymbol{\bar{D}x} + \boldsymbol{\bar{K}} \int \boldsymbol{x} - \int (\boldsymbol{h_h} - \boldsymbol{k_f h_e^d}) \tag{5.11}$$

Up to now, the same passages of the 1 DOF formulation have been performed. However, this time equation (5.11) cannot be directly implemented, because our measurements and control signals lay on the joint space, while the impedance profile is defined in the operational space. Therefore, we need to *make* $\boldsymbol{S}$ *explicit* with respect to the joint variables in order to implement it in our controller.

Let's consider the manipulator forward and differential kinematics. Thanks to them, we can express end - effector position, velocity and acceleration as a function of joint coordinates:

$$\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{q}) \qquad \dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad \ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{5.12}$$

where $\boldsymbol{T}(\boldsymbol{q})$ is the direct kinematics of the robotic manipulator, while $\boldsymbol{J}(\boldsymbol{q}) = \frac{d\boldsymbol{T}(\boldsymbol{q})}{d\boldsymbol{q}}$ is the Jacobian matrix. Substituting expressions (5.12) into (5.11) we obtain:

$$\boldsymbol{S} = \bar{\boldsymbol{M}}\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} + \bar{\boldsymbol{D}}\boldsymbol{T}(\boldsymbol{q}) + \bar{\boldsymbol{K}}\int \boldsymbol{T}(\boldsymbol{q}) - \int(\boldsymbol{h}_h - \boldsymbol{k}_f\boldsymbol{h}_e^d) \tag{5.13}$$

Although the sliding manifold $\boldsymbol{S}$ can now be computed using only joint position and velocity measurements, its output still belongs to the Cartesian environment. This means that each component of $\boldsymbol{S}$ describes the distance between the sliding surface and the manipulator's state in the *operational space*, while control effort needs to be given in joint space.

Let's suppose that manipulator's DOFs are equal to the number of impedance profiles. If we apply directly $\boldsymbol{S}$ as a sliding surface at joint level, first joint will be driven based on the Cartesian error of the first coordinate and so on for the remaining joints. This is not logical because error along a Cartesian direction should be distributed between all the joints.

To solve this mismatch, it is convenient to use the robot static relationship [73]:

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{h} \tag{5.14}$$

which maps forces and moments at the end - effector with joint torques through the transpose of the Jacobian matrix. Since the desired impedance profile is a vectorial torque and force balance at the end - effector, it is possible to apply equation (5.14) to derive the equivalent torque balance at the joint level:

$$\boldsymbol{I_\tau} = \boldsymbol{J}(\boldsymbol{q})^\top \left[\bar{\boldsymbol{M}}\ddot{\boldsymbol{x}} + \bar{\boldsymbol{D}}\dot{\boldsymbol{x}} + \bar{\boldsymbol{K}}\boldsymbol{x} - \boldsymbol{h_h} + \boldsymbol{k_f}\boldsymbol{h_e^d}\right] \tag{5.15}$$

Unfortunately, equation (5.15) *cannot* be used as a sliding surface, because of its dependency on $\ddot{\boldsymbol{x}}$ (i.e. $\ddot{\boldsymbol{q}}$). Indeed, if kinematic relationships (5.12) are substituted in (5.15), it is not possible to make $\int \left(\boldsymbol{J}(\boldsymbol{q})^\top \bar{\boldsymbol{M}}\ddot{\boldsymbol{q}}\right)$ independent from $\ddot{\boldsymbol{q}}$ (the Jacobian matrix is variable with time).

An available alternative consists in defining our operational sliding surface as:

$$\begin{aligned} \boldsymbol{\sigma_{op}} &= \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{S} \qquad (5.16) \\ &= \boldsymbol{J}(\boldsymbol{q})^\top \left[\bar{\boldsymbol{M}}\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} + \bar{\boldsymbol{D}}\boldsymbol{T}(\boldsymbol{q}) + \bar{\boldsymbol{K}}\int \boldsymbol{T}(\boldsymbol{q}) - \int \left(\boldsymbol{h_h} - \boldsymbol{k_f}\boldsymbol{h_e^d}\right)\right] \end{aligned}$$

which is, in general, different from the integral of (5.15).

Let's assume that expression (5.16) is a valid sliding surface candidate. Since the initial control objective was to track the desired operational space impedance profile by imposing $\boldsymbol{I}$ as a sliding manifold (equation (5.10)), it is necessary to verify that $\boldsymbol{\sigma_{op}}$ still guarantees that. To this aim, the following theorem has been enounced:

Theorem 5.1. *(Desired dynamics convergence) Let's consider an n - DOF manipulator, with Jacobian's rank equals to the impedance tracking task dimension ($\dim \boldsymbol{I} = n$) and far from kinematic singularities. If $\boldsymbol{\sigma_{op}} = \boldsymbol{0} \quad \forall t \geq t_f$, then $\boldsymbol{I} = \boldsymbol{0} \quad \forall t \geq t_f$ and the correct impedance profile is tracked.*

Proof. Let's assume that $\mathcal{Q}$ is a set of joint positions which do not contain kinematic singularities. By hypothesis, the following homogeneous system can be written:

$$\boldsymbol{J}(\boldsymbol{q})^\top \int \boldsymbol{I} = \boldsymbol{0} \tag{5.17}$$

thanks to Rouché–Capelli theorem, if matrix $\boldsymbol{J}(\boldsymbol{q})^\top$ fulfils the assumption:

$$\operatorname{rank} \boldsymbol{J}(\boldsymbol{q})^\top = n = \dim \boldsymbol{I} \quad \forall \boldsymbol{q} \in \mathcal{Q}$$

then, it exists a unique solution for the system (5.17) which corresponds to:

$$\int \boldsymbol{I} = \boldsymbol{0} \quad \forall t \geq t_f \quad \Rightarrow \quad \boldsymbol{I} = \boldsymbol{0} \quad \forall t \geq t_f$$

□

As we have already noticed for the 1 DOF case, this is only a *sufficient condition*, since $\boldsymbol{I} = \boldsymbol{0}$ does not imply that the operational sliding surface $\boldsymbol{\sigma_{op}}$ goes to zero. Theorem 5.1 is applicable also if the manipulator is structurally or functionally redundant (it is sufficient that the rectangular Jacobian matrix has the same rank of the desired impedance profile). These cases will be treated in detail in section 5.3.

Corollary 5.1. *If theorem 5.1 is true ($\boldsymbol{I} = \boldsymbol{0}$ $\forall t \geq t_f$), choosing equation (5.16) as sliding surface generates the same sliding motion of equation (5.15). Indeed, $\int \boldsymbol{I_\tau}$ is different from $\boldsymbol{\sigma_{op}}$, but they share the same sliding surface.*

$$\boldsymbol{J}(\boldsymbol{q})^\top \int \boldsymbol{I} = \boldsymbol{0} \quad \forall t \geq t_f \quad \Leftrightarrow \quad \int \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{I} = \boldsymbol{0} \quad \forall t \geq t_f$$

5.2.2 Centralized equivalent control

Once the sliding manifold has been defined, we are left with deriving the equivalent control law to be applied to the n - DOF manipulator. As usual, $\boldsymbol{v^{eq}}$ will be computed by equating to zero the sliding variable derivative and substituting it in the nominal dynamic system (i.e. perfect inverse dynamics achieved). Instead of computing $\boldsymbol{\dot{\sigma}_{op}}$, the desired impedance profile $\boldsymbol{I}$ can be used. This is motivated by the fact that, when system is in sliding mode condition ($\boldsymbol{\sigma_{op}} = \boldsymbol{0}$), not only $\boldsymbol{\dot{\sigma}_{op}} = \boldsymbol{0}$ but also $\boldsymbol{I} = \boldsymbol{0}$:

$$\boldsymbol{\bar{M}\ddot{x}} + \boldsymbol{\bar{D}\dot{x}} + \boldsymbol{\bar{K}x} - \boldsymbol{h_h} + \boldsymbol{k_f h_e^d} = \boldsymbol{0} \tag{5.18}$$

now substitute the kinematic relationships (5.12) to express the desired impedance profile as a function of joint variables:

$$\boldsymbol{\bar{M}}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{\ddot{q}} + \boldsymbol{\dot{J}}(\boldsymbol{q})\boldsymbol{\dot{q}}) + \boldsymbol{\bar{D}J}(\boldsymbol{q})\boldsymbol{\dot{q}} + \boldsymbol{\bar{K}T}(\boldsymbol{q}) - \boldsymbol{h_h} + \boldsymbol{k_f h_e^d} = \boldsymbol{0} \tag{5.19}$$

Since equivalent control should be computed when the system is in nominal conditions, let's consider the following manipulator model:

$$\ddot{\boldsymbol{q}} = \boldsymbol{v} \tag{5.20}$$

Eventually, substituting (5.20) into (5.19), we came up with the system:

$$\begin{gathered} \bar{\boldsymbol{M}}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{v} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}) + \bar{\boldsymbol{D}}\dot{\boldsymbol{q}} + \bar{\boldsymbol{K}}\boldsymbol{T}(\boldsymbol{q}) - \boldsymbol{h_h} + \boldsymbol{k_f}\boldsymbol{h_e^d} = \boldsymbol{0} \\ \boldsymbol{A}(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{b}(\boldsymbol{q}) \end{gathered} \tag{5.21}$$

where $\boldsymbol{A}(\boldsymbol{q}) = \bar{\boldsymbol{M}}\boldsymbol{J}(\boldsymbol{q})$ and $\boldsymbol{b}(\boldsymbol{q}) = \boldsymbol{h_h} - \boldsymbol{k_f}\boldsymbol{h_e^d} - \bar{\boldsymbol{M}}\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \bar{\boldsymbol{D}}\dot{\boldsymbol{q}} - \bar{\boldsymbol{K}}\boldsymbol{T}(\boldsymbol{q})$.

Here, it is necessary to make a distinction between two cases:

- $\boldsymbol{A}(\boldsymbol{q})$ is square and invertible;
- $\boldsymbol{A}(\boldsymbol{q})$ is under-determined (desired impedance dimension is lower with respect to the manipulator's DOFs);

Since the second case is typical for redundant manipulators, it will be treated in subsection 5.3.1. For the first one, it is possible to directly find the expression of $\boldsymbol{v^{eq}}$ by computing the solution of system (5.21):

$$\boldsymbol{v^{eq}} = \boldsymbol{A}(\boldsymbol{q})^{-1}\boldsymbol{b}(\boldsymbol{q}) \tag{5.22}$$

Computation of the equivalent control through the inversion of the Jacobian matrix, although feasible, does not allow to impose additional constraints on joint limits, control effort, etc. Moreover, this process causes a significant loss of precision when $\boldsymbol{q}$ is close to singular values, because matrix $\boldsymbol{A}(\boldsymbol{q})$ loses rank and becomes no more invertible [62].

Thus, an MPC - like structure can be useful to enhance control specification capabilities. Indeed, it is possible to rewrite system (5.21) as an optimization problem, with quadratic cost function:

$$C = ||\boldsymbol{A}(\boldsymbol{q})\boldsymbol{v} - \boldsymbol{b}(\boldsymbol{q})||^2 = \boldsymbol{q}^\top\boldsymbol{A}(\boldsymbol{q})^\top\boldsymbol{A}(\boldsymbol{q})\boldsymbol{q} - \boldsymbol{2}\boldsymbol{q}^\top\boldsymbol{A}(\boldsymbol{q})^\top\boldsymbol{b}(\boldsymbol{q}) + \boxed{\boldsymbol{b}^\top\boldsymbol{b}} \tag{5.23}$$

where the last term is removed because it represents only a translation of the cost function (i.e. it does not affect the minimization procedure).

Once C has been computed, it is possible to perform the minimization procedure with one step predictive horizon, as the one presented in section 3.8. Doing so, if there are no active constraints, we know already that the solution is the same as (5.22). Thus, in this case, the main advantage of performing this passage is given by the possibility of modifying the solution of 5.22 in order to constraint the state/control variable. Instead, for redundant manipulators, this strategy will become crucial since 5.22 is not applicable any more.

5.3 Redundant manipulator

If the joint space of a robotic device has an higher dimensionality than the task space required to perform a given operation, the manipulator is said to be *redundant* for that specific task. Since the operational space can have at most 6 DOFs, robots with more than 6 joints are *structurally* redundant. Instead, if the required task is simplified (e.g. only end - effector positioning), the operational space reduces and manipulators that have ≤ 6 DOFs can become *functionally* redundant.

Redundancy is very common when we are dealing with impedance specification in the operational space, because the required dynamic characteristics usually involve a *limited number* of dimensions. Therefore, it is necessary to analyse and modify the theory presented in the previous section in order to deal with under-determined systems which can accept possibly infinite solutions.

From now on, let's indicate with the symbol n the dimension of the desired impedance profile vector, while the number of robot's Degrees of Freedom (i.e. joint space dimension) will be labelled with symbol m. For redundant manipulators, the relationship $m > n$ holds.

5.3.1 MPC equivalent control

Recalling system (5.17), it is worth noticing that matrix $\boldsymbol{A}(\boldsymbol{q})$ has the same dimension of the Jacobian and it is equal to $\dim \boldsymbol{A}(\boldsymbol{q}) = n \times m$. Since $m > n$, there are more variables than constraints on our system, which is then under-determined. Consequently, there exist *infinite* equivalent control vectors that fulfil (5.17).

To choose one of them, a standard solution consists in minimizing an additional cost function $h(\boldsymbol{v})$ while retaining the optimality properties of (5.23). This allows to specify the behaviour of redundant DOFs. For example, a typical requirement is the penalisation of control effort amplitude, which is obtained by imposing:

$$h(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\top \boldsymbol{Q}\boldsymbol{v}$$

where $\boldsymbol{Q}$ is a positive diagonal weight matrix. This choice of $h(\boldsymbol{q})$ generates the following equivalent control vector [73]:

$$\boldsymbol{v^{eq}} = \boldsymbol{A(q)}^{+}\boldsymbol{b(q)} \tag{5.24}$$

where $\boldsymbol{A(q)}^{+} = \boldsymbol{A(q)}^{\top}\left(\boldsymbol{A(q)}\boldsymbol{A(q)}^{\top}\right)^{\boldsymbol{-1}}$ is the right Moore - Penrose pseudo-inverse of matrix $\boldsymbol{A(q)}$.

If it is desirable to obtain a different solution with respect to (5.24), it is necessary to select a proper $h(\boldsymbol{q})$ and to solve a new optimization problem. To perform this procedure on-line, 1 step prediction horizon MPC can be implemented.

Similarly to subsection 5.2.2, system (5.17) is used to write a quadratic cost function (5.23) and `qpOASES` solver is chosen to perform the optimization phase. Since there exist multiple solutions that guarantee $C = 0$, algorithm will select (if no constraint is applied) the closest one with respect to its initial condition, which will be different, in general, from (5.24). To exploit the redundancy, *constraints* can be added to the optimization (5.25). In this way, the best $\boldsymbol{v^{eq}}$ which satisfies the desired impedance and, at the same time, the constraints is selected (e.g. operational space limitations, maximum accelerations - velocities, etc.).

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & ||\boldsymbol{A(q)v} - \boldsymbol{b(q)}||^2 \\ \text{s.t.} \quad & \boldsymbol{A_{low}} \leq \boldsymbol{Av} \leq \boldsymbol{A_{high}}, \\ & \boldsymbol{v_{low}} \leq \boldsymbol{v} \leq \boldsymbol{v_{high}} \end{aligned} \tag{5.25}$$

5.3.2 Null space sliding surface

It has been already remarked that, for redundant manipulators, there is an infinite number of control laws that can guarantee $\boldsymbol{I} = \mathbf{0}$. This means that there exists a subset of the manipulator joint space, called *null space*, whose projection in the task space has no effect on the task (Figure 5.3). These "null motions" are responsible for constraints fulfilment, preserving the main impedance tracking objective. The number of auxiliary tasks that can be performed is equal to the null space dimension $p = m - n$.

If no uncertainty affects the system, $\boldsymbol{v^{eq}}$ computed in subsection 5.3.1 will be sufficient to drive correctly the manipulator and fulfil the constraints imposed by MPC. Since this is usually not the case, a sliding mode control action is required to improve the system robustness.

Taking advantage of the result obtained in subsection 5.2.1, the redundant manipulator sliding surface can be chosen equal to $\boldsymbol{\sigma_{op}}$ (5.16). Doing so, the desired impedance profile tracking is guaranteed during sliding mode.

Unfortunately, the chosen $\boldsymbol{\sigma_{op}}$ *has no knowledge* of the manipulator redundancy, since it has been derived from the Cartesian impedance expression. Indeed, we have that every null space motion of the robot, by definition, does not modify the impedance relation. Consequently, if we suppose that ISMC is implemented ($\boldsymbol{\sigma_{op}}(0) = \mathbf{0}$), $\boldsymbol{\sigma_{op}}(q^*, t) = \mathbf{0}$ $\forall \boldsymbol{q^*} \in \ker \mathcal{Q}$ ($\ker \mathcal{Q}$ is the joint null space).

Let's suppose that the manipulator is affected by a disturbance that acts in

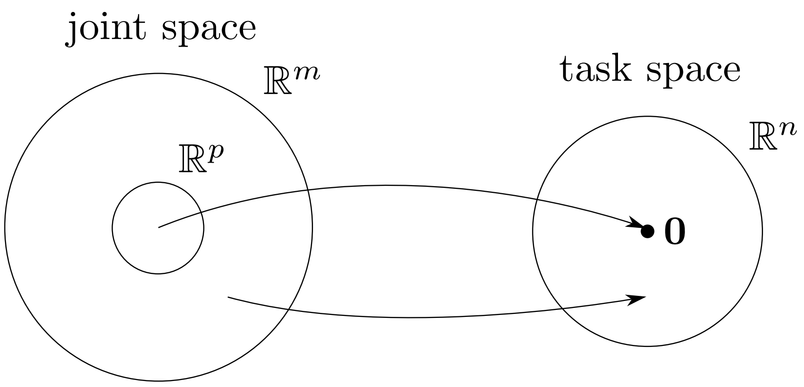


Figure 5.3: Mapping between the manipulator joint space and the task space for a redundant manipulator

the joint null space joint subset; this will generate deviations in the null space from the ideal motion trajectory computed by MPC. However, due to the previous considerations, $\boldsymbol{\sigma_{op}}$ remains zero during these motions, since it is *insensitive* to the disturbance. Ultimately, this may lead to constraint violations.

That is why it is necessary to design a new sliding variable which is able to cope with the uncertainties coming from null space motion.

As first step, we rewrite the standard sliding surface equation (5.16) as an underdetermined system:

$$\begin{aligned} \boldsymbol{\sigma} &= \boldsymbol{J}^\top \int \boldsymbol{I} \\ \boldsymbol{J}\boldsymbol{\sigma} &= \boldsymbol{J}\boldsymbol{J}^\top \int \boldsymbol{I} \\ \left(\boldsymbol{J}\boldsymbol{J}^\top\right)^{-1} \boldsymbol{J}\boldsymbol{\sigma} &= \int \boldsymbol{I} \end{aligned} \tag{5.26}$$

where $\boldsymbol{J}\boldsymbol{J}^\top$ is a $n \times n$ invertible matrix (full rank Jacobian hypothesis).

As usual for redundant manipulators, (5.26) admits infinite solutions, which can be composed by a *particular one* and a *generic one* projected in the null space of $(\boldsymbol{J}^\top\boldsymbol{J})^{-1}\boldsymbol{J}$:

$$\boldsymbol{\sigma} = \underbrace{\boldsymbol{J}^\top \int \boldsymbol{I}}_{\boldsymbol{\sigma_{op}}} + \underbrace{(\boldsymbol{I} - \boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{J}^\top)^{-1}\boldsymbol{J})}_{\boldsymbol{P}}\,\boldsymbol{\sigma_0} \tag{5.27}$$

Thanks to the initial step, the particular solution coincides with the expression of $\boldsymbol{\sigma_{op}}$ adopted in subsection 5.2.1, while the remaining term is the projection into the manipulator's null space of a *new* sliding surface $\boldsymbol{\sigma_0}$. This term will be responsible for the rejection of disturbances acting in the null space, without affecting the desired impedance tracking. Indeed, the projection matrix $\boldsymbol{P}$ is defined such that:

$$\left(\boldsymbol{J}\boldsymbol{J}^\top\right)^{-1}\boldsymbol{J}\boldsymbol{P} = \left(\boldsymbol{J}\boldsymbol{J}^\top\right)^{-1}\boldsymbol{J}(\boldsymbol{I} - \boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{J}^\top)^{-1}\boldsymbol{J}) = \boldsymbol{0} \tag{5.28}$$

In order to detect deviations from the ideal null space motion trajectory, it is necessary to check the mismatch between the MPC equivalent control signal we are currently applying and the signal that would have to be applied to the nominal nominal system to generate the observed state evolution. More formally, if we *reverse* the definition of the equivalent control law, we can choose:

$$\boldsymbol{\sigma_0} \quad \text{s.t.} \quad \dot{\boldsymbol{\sigma}}_{\boldsymbol{0}} = \boldsymbol{0} \quad \text{when} \quad \boldsymbol{v} = \boldsymbol{v}^{\boldsymbol{eq}} \tag{5.29}$$

A possible solution to (5.29) is $\boldsymbol{\sigma_0} = \boldsymbol{0}$, but this choice makes the new sliding surface coincide with the operational space one, that we have already seen to be not effective. Indeed, this choice does not take into account the null space state. Another option is represented by:

$$\dot{\boldsymbol{\sigma}}_{\boldsymbol{0}} = \boldsymbol{v} - \boldsymbol{v}^{\boldsymbol{eq}} \qquad \boldsymbol{\sigma_0} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^{\boldsymbol{eq}} \tag{5.30}$$

which indicates the difference between the actual manipulator speed and the one that it should have if the MPC control law were applied without uncertainty. This term is sensitive to disturbances acting on the null space motion and therefore it is a good candidate for $\boldsymbol{\sigma_0}$.

Now that we have specified all the terms of (5.27), we can apply SMC techniques based on $\boldsymbol{\sigma}$ in order to reject uncertainties in the whole manipulator joint space, not only in the image of the task space.

5.4 2 - DOF simulation

To verify the results obtained in sections 5.2 and 5.3, a simulation has been set up in `MATLAB - Simulink`®. A fixed step solver and sampling time of 4 ms have been adopted. To limit the complexity of the mechanical model and focus mainly on the new control strategies, a 2 DOFs planar manipulator has been implemented.

First of all, let's briefly recall the kinematic and dynamic relationships which define the robotic arm model:

- **Dynamic model matrices**

$$\boldsymbol{B} = \begin{bmatrix} m_1 l_1^2 + I_1 + m_2(a_1^2 + l_2^2 + 2a_1 l_2 \cos\theta_2) + I_2 & m_2(l_2^2 + a_1 l_2 \cos\theta_2) + I_2 \\ m_2(I_2^2 + a_1 l_2 \cos\theta_2) + I_2 & m_2 l_2^2 + I_2 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} -2m_2a_1l_2\dot{\theta}_2\sin\theta_2 & -m_2a_1l_2\dot{\theta}_2\sin\theta_2 \\ m_2a_1l_2\dot{\theta}_1\sin\theta_2 & 0 \end{bmatrix} \tag{5.31}$$

$$\boldsymbol{g} = \begin{bmatrix} (m_1l_1 + m_2a_1)g\cos\theta_1 + m_2gl_2\cos(\theta_1+\theta_2) \\ m_2l_2g\cos(\theta_1+\theta_2) \end{bmatrix}$$

- **Forward kinematics**

$$\begin{aligned} x &= a_1\cos(\theta_1) + a_2\cos(\theta_1+\theta_2) \\ y &= a_1\sin(\theta_1) + a_2\sin(\theta_1+\theta_2) \end{aligned} \tag{5.32}$$

- **Jacobian matrix**

$$\boldsymbol{J} = \begin{bmatrix} -a_1\sin\theta_1 - a_2\sin(\theta_1+\theta_2) & -a_2\sin(\theta_1+\theta_2) \\ a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) & a_2\cos(\theta_1+\theta_2) \end{bmatrix} \tag{5.33}$$

where θ_1 and θ_2 are the angles, respectively, of the first and the second joints. a_1 and a_2 represent the link lengths and l_1 and l_2 identify the centre of mass position for each link. m_1 and m_2 are the link masses and I_1, I_2 stand for link moments of inertia. Finally, g is the standard gravity ($9.81\,\mathrm{m\,s^{-2}}$). Joint coordinates are chosen positive counter-clockwise. A graphical representation of the 2 DOF manipulator is reported in Figure 5.4. The shaded space represents the boundary of the robotic arm operative region, while the asterisk is the goal position for the end - effector. To avoid starting too close to singularity, the initial joint positions have been set to $\boldsymbol{\theta_0} = [-0.5 \;\; 1]^\top (rad)$.

Nominal and estimated values adopted for the simulation are reported in Table 5.1, together with their percentage of uncertainty. It has been assumed that geometrical quantities (link lengths, centre of gravities) are known with sufficient accuracy, so that their estimated values coincide with the nominal ones. Moreover, in the manipulator model, a diagonal constant friction term is added to matrix $\boldsymbol{C}$ in order to simulate joint static friction ($\boldsymbol{C_{fric}} = \mathrm{diag}(18, 18)$).

Aware of the experience regarding higher order sliding mode controllers, to limit chattering and other undesired effects, STA has been implemented on both joint

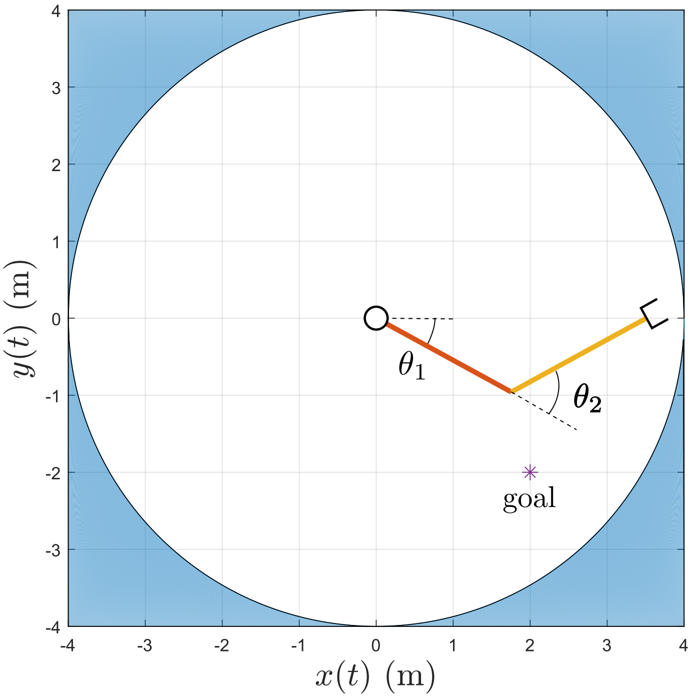


Figure 5.4: Graphic visualization of the 2 DOFs planar manipulator implemented

	nominal	estimated	uncertainty (%)
a_1 (m)	2	2	0.0
a_2 (m)	2	2	0.0
l_1 (m)	1	1	0.0
l_2 (m)	1	1	0.0
m_1 (kg)	5	4	-20.0
m_2 (kg)	3	4	33.3
I_1 (kg m^2)	1.5	1	-33.3
I_2 (kg m^2)	1	0.7	-30.0

Table 5.1: Nominal and estimated parameters for the 2 DOFs planar manipulator

channels, with parameters $\boldsymbol{k_1} = [5\ \ 5]^\top$ and $\boldsymbol{k_2} = [40\ \ 40]^\top$. To avoid the initial reaching phase, ISMC is added as well. It is important to notice that, for operational space sliding surfaces of type (5.16), removal of constant bias due to the initial offset must be performed *before* the projection into the joint space of the desired impedance. This is because theorem 3.3 is not applicable directly to $\boldsymbol{\sigma_{op}}$. Indeed, if we choose $\boldsymbol{v^{eq}}$ such that $\boldsymbol{I} = \boldsymbol{0}$, it holds that:

$$\dot{\boldsymbol{\sigma}}_{\boldsymbol{op}} = \boldsymbol{J}^\top \boldsymbol{I} + \dot{\boldsymbol{J}}^\top \int \boldsymbol{I} = \dot{\boldsymbol{J}}^\top \boldsymbol{c} \qquad \boldsymbol{c} \in \mathbb{R}^2 \quad \text{constant} \tag{5.34}$$

which is neither zero nor constant (violating theorem 3.3 hypothesis). This fact is a direct consequence of the only sufficient conditions used to derive the equivalent control in subsection 5.2.2. To overcome this issue, it is possible to apply ISMC on (5.11), for which the theorem still holds. Indeed, $\boldsymbol{v_{eq}}$ is computed based on $\boldsymbol{I} = \dot{\boldsymbol{S}} = \boldsymbol{0}$, therefore theorem 3.3 is valid.

Once initial bias of $\boldsymbol{S}$ has been removed, joint space mapping can be performed:

$$\boldsymbol{\Sigma_{op}}(\boldsymbol{q}(t), \boldsymbol{\Gamma}(t)) = \boldsymbol{J}(\boldsymbol{q}(t))^\top \left(\boldsymbol{S}(\boldsymbol{q}(t), \boldsymbol{\Gamma}(t)) - \boldsymbol{S}(\boldsymbol{q}(t_0), \boldsymbol{\Gamma}(t_0))\right) \tag{5.35}$$

this passage does not alter the sliding surface initial condition, which is:

$$\boldsymbol{\Sigma_{op}}(\boldsymbol{q}(t_0), \boldsymbol{\Gamma}(t_0)) = \boldsymbol{J}(\boldsymbol{q}(t_0))^\top \left(\boldsymbol{S}(\boldsymbol{q}(t_0), \boldsymbol{\Gamma}(t_0)) - \boldsymbol{S}(\boldsymbol{q}(t_0), \boldsymbol{\Gamma}(t_0))\right) = \boldsymbol{0} \tag{5.36}$$

5.4.1 Operational space SMC without redundancy

First simulation consists in forcing x and y coordinates of the 2 DOFs manipulator to the specified goal position $\boldsymbol{x}^* = [2\ \ -2]^\top$, starting from $\boldsymbol{x_0}$ and without external environmental forces. The state evolution in operational space must follow the desired impedance profile imposed for both coordinates:

$$\bar{\boldsymbol{M}}\ddot{\tilde{\boldsymbol{x}}} + \bar{\boldsymbol{D}}\dot{\tilde{\boldsymbol{x}}} + \bar{\boldsymbol{K}}\tilde{\boldsymbol{x}} = \boldsymbol{0} \tag{5.37}$$

where $\bar{\boldsymbol{M}} = \text{diag}(1, 1)$, $\bar{\boldsymbol{D}} = \text{diag}(4, 4)$ and $\bar{\boldsymbol{K}} = \text{diag}(4, 4)$.

In Figure 5.5, position responses of the end - effector in the Cartesian space are shown. When no sliding mode is enforced on the system, both coordinates suffer from the uncertainties generated by dynamic inversion, drifting from their desired profiles. This causes a significant static error in the end - effector final pose, equal to $\boldsymbol{x} = [1.608 \ -0.977]^\top$. Connecting operational space SMC to the same impedance controller, an important improvement can be noticed. In this case, disturbances on both profiles are rejected and theoretical impedance accurately tracked (dashed lines). Both components of $\boldsymbol{\sigma_{op}}$ start from zero and converge in finite time (Figure 5.6a); the control effort required by STA is “almost” chattering free and its amplitude is comparable with the control signal given by the equivalent controller (Figure 5.6b).

5.4.2 Operational space SMC with redundancy

In the second simulation, the 2 DOFs manipulator is considered to be functionally redundant. The impedance tracking task is specified only along the x coordinate, while constraints are applied to the y coordinates in order to select the equivalent control law. Therefore, the new Jacobian matrix $\boldsymbol{J_1}$ becomes the first row of $\boldsymbol{J}$ ($\dim \boldsymbol{J_1} = 1 \times 2$). The desired impedance dynamics for x is defined by the following scalar expression:

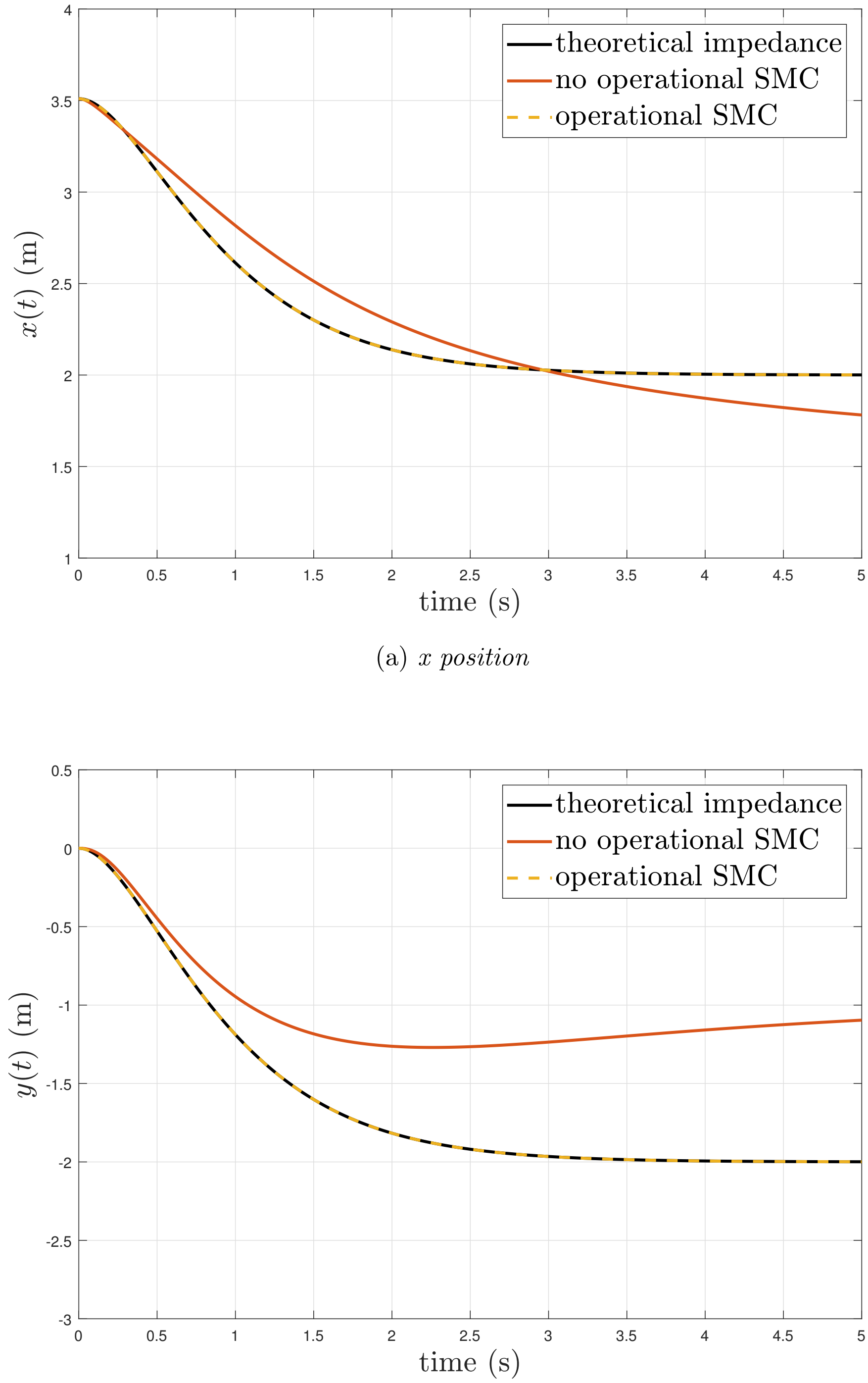
$$\bar{M}_x \ddot{\tilde{x}} + \bar{D}_x \dot{\tilde{x}} + \bar{K}_x \tilde{x} = 0 \tag{5.38}$$

while the MPC minimization problem has the following structure:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \left\| \boldsymbol{J_1 v} + \boldsymbol{\dot{J}_1 \dot{q}} + \frac{\bar{D}_x}{\bar{M}_x} \boldsymbol{J_1 \dot{q}} + \frac{\bar{K}_x}{\bar{M}_x} \left(T_1(q) - x^* \right) \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{A_{low}} \leq \boldsymbol{Av} \leq \boldsymbol{A_{high}}, \\ & \boldsymbol{v_{low}} \leq \boldsymbol{v} \leq \boldsymbol{v_{high}} \end{aligned} \tag{5.39}$$

where $T_1(q)$ is the first row of the 2 DOFs direct kinematics, while $x^* = 2$ is the target position only for x coordinates. To select one among the infinite number of $\boldsymbol{v}$ which satisfy (5.39), suitable constraints on y should be adopted. Two options have been considered:

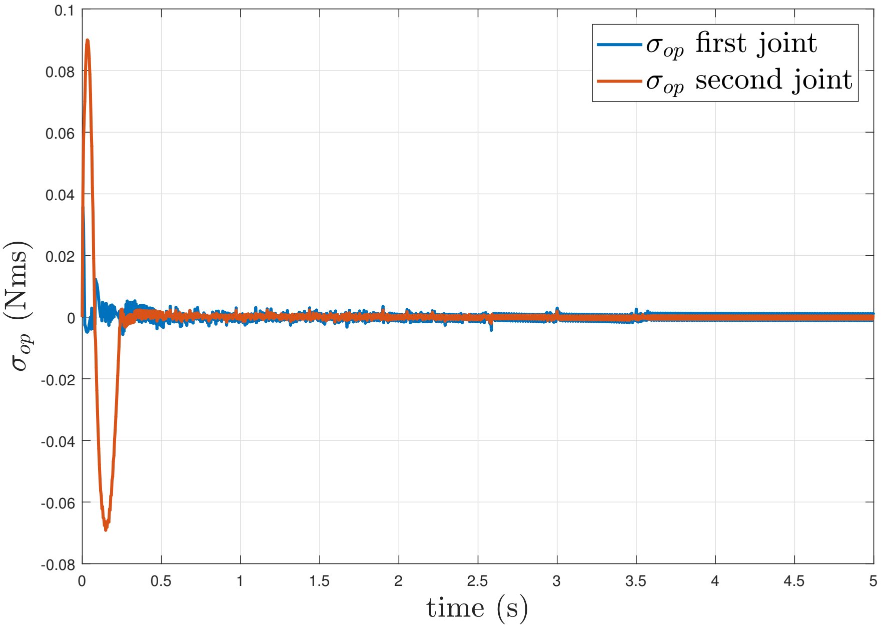
1. constrain y to lie on the x axis ($y(t) = 0 \quad \forall t \geq t_0$);



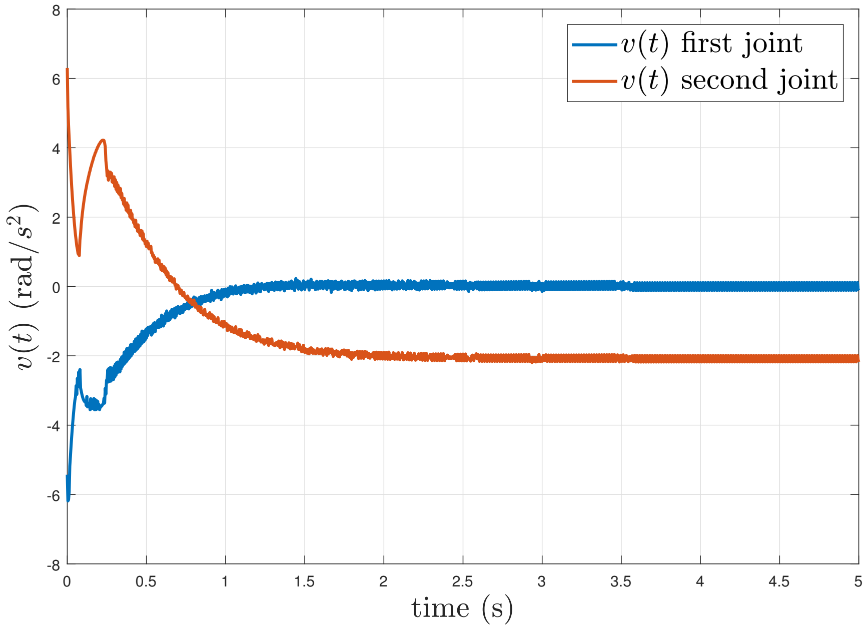
(a) *x position*

(b) *y position*

Figure 5.5: Task space end - effector dynamics with and without operational space sliding surface



(a) *sliding surface*



(b) *control signal*

Figure 5.6: Sliding surface and control signal dynamics for operational space SMC

2. constrain y to follow a TVP arbitrarily imposed;

In both cases, it is necessary to apply a constraint in the Cartesian space position to a minimization problem where the optimization variables are the joints angular accelerations. To perform this conversion, *Grönwall's lemma* has been employed.

First of all, let's consider to apply the second constraint to the minimization problem (5.39). Given the TVP position, velocity and acceleration for each sampling time $(y_p(t), \dot{y}_p(t), \ddot{y}_p(t))$, it is possible to write:

$$\begin{aligned} y(t) &= y_p(t) \\ \dot{y}(t) &= \dot{y}_p(t) \qquad \forall t \geq t_0 \\ \ddot{y}(t) &= \ddot{y}_p(t) \end{aligned} \tag{5.40}$$

Thanks to the Grönwall's lemma, equalities (5.40) can be written as the solutions of a unique stable differential equation of the following type:

$$\ddot{y}(t) = \ddot{y}_p(t) - (\chi_1 + \chi_2)(\dot{y}(t) - \dot{y}_p(t)) - \chi_1\chi_2(y(t) - y_p(t)) \tag{5.41}$$

where χ_1 and χ_2 are positive scalar coefficients. By imposing (5.41), we are ensuring that, after a reaching transient dictated by χ_1 and χ_2, constraints (5.40) are fulfilled.

The last step consists in making explicit the dependency on $\boldsymbol{v}$ of the constraint and write it as an inequality:

$$\begin{gathered} \boldsymbol{A_{low}} = \boldsymbol{A_{high}} = \ddot{y}_p - \dot{\boldsymbol{J}}_{\boldsymbol{2}}\dot{\boldsymbol{q}} - (\chi_1 + \chi_2)(\boldsymbol{J_2}\dot{\boldsymbol{q}} - \dot{y}_p) - \chi_1\chi_2(T_2(q) - y_p) \\ \boldsymbol{A} = \boldsymbol{J_2} \end{gathered} \tag{5.42}$$

where $\boldsymbol{J_2}$ and $T_2(q)$ are the second rows, respectively, of the Jacobian matrix and direct kinematic. If the first constraint $(y(t) = 0 \quad \forall t \geq t_0)$ is to be applied, it is sufficient to just substitute $(y_p(t), \dot{y}_p(t), \ddot{y}_p(t))$ with $(0, 0, 0)$.

Since we are forcing only one coordinate to have a desired impedance profile, $\boldsymbol{\sigma_{op}}$ is based only on equation (5.38). In Figure 5.7a, the first constraint on the y coordinate has been applied, with and without the null space sliding surface $\boldsymbol{\sigma_0}$. As it can be seen, uncertainty in null space motion generates a drift in the vertical direction when robust control is not active. However, from this plot one may think that this tracking error is marginal with respect to the main task objective of impedance tracking. Unfortunately, this is not always the case. As it is shown in Figure 5.7b, when the TVP is applied on y, constraint violation is significant. Depending on the degree of uncertainty affecting the system, such uncontrolled motion may even compromise the functionality of $\boldsymbol{\sigma_{op}}$ in the main channel. Here $\boldsymbol{\sigma_0}$ greatly improves null space motion, allowing fulfilment of MPC constraints.

5.5 7 - DOF simulation

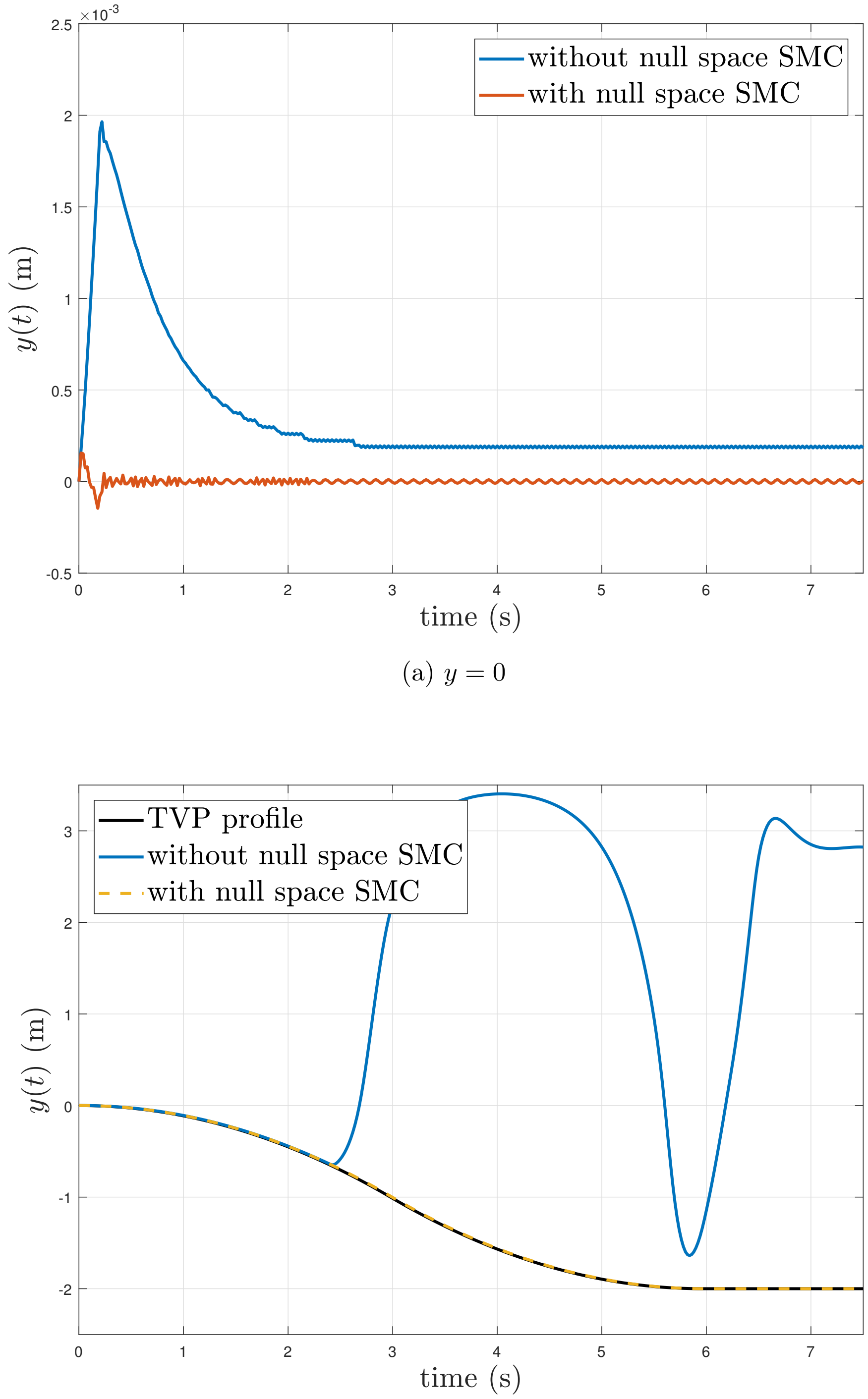
A last simulation step has been performed on a 7 DOFs manipulator model, replica of the `ABB IRB14000 YuMi`® right arm. To generate the robot model, *Peter Corke's Robotics Toolbox for MATLAB* [13] has been used. In particular, , since it interfaces with `MATLAB - Simulink` library.

For the sake of brevity, kinematic and dynamical data used for the implementation of the 7 DOFs manipulator are omitted. Dynamic matrices and vectors $\boldsymbol{B(q)}$, $\boldsymbol{C(\dot{q},q)}$ and $\boldsymbol{G(q)}$ are computed on-line given the simulated robot pose. A graphical representation of the robotic arm, with its seven revolution joints, can be seen in Figure 5.8.

The level of uncertainty given by the inverse dynamic process has been introduced by underestimating by 20% the value of $\boldsymbol{n(\dot{q},q) = C(\dot{q},q)\dot{q} + g(q)}$ in the feedback linearization ($\boldsymbol{\hat{n}(\dot{q},q) = 0.8n(\dot{q},q)}$).

To force joint variables to follow given reference impedance profiles, a decentralized SMC is designed. Doing so, we avoid dealing with the redundancy problem, which should be tackled if we consider operational space. Moreover, a decentralized control strategy does not suffer of kinematic singularities, which in this case can be difficult to be computed a-priori (on-line check of Jacobian rank).

Equation (5.4) is adopted of each joint, with diagonal matrices $\boldsymbol{\bar{B} = I_7}$, $\boldsymbol{\bar{C}} = 10\,\boldsymbol{I_7}$ and $\boldsymbol{\bar{g}} = 25\,\boldsymbol{I_7}$, where $\boldsymbol{I_7}$ is the 7×7 identity matrix. Model mismatches are simply modelled by changing the on-line computed value of $\boldsymbol{C(\dot{q},q)}$ in the inverse dynamic process.



(a) $y = 0$

(b) *TVP y profile*

Figure 5.7: Constraint fulfilment of the free y coordinate with and without $\boldsymbol{\sigma_0}$

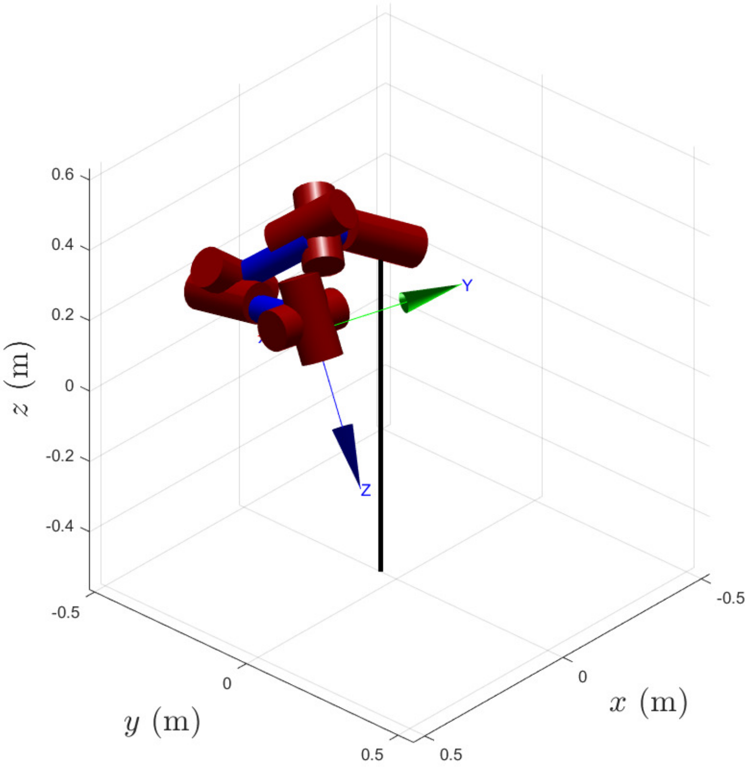


Figure 5.8: Three dimensional representation of the robotic arm used for 7 DOFs simulation in its initial pose $\boldsymbol{q_0}$

Similarly to the 2 DOFs simulation, STA and ISMC are implemented to improve the performance of the control scheme. In Figure 5.9, the position evolution for each of the seven joints is reported. It is noticeable that the coupled system uncertainty has a strong impact on the impedance controller. Indeed, except for the fifth and seventh joints, the others show a noticeable drift in the response when there is no SMC (blue lines). The situation significantly improves when disturbance is rejected by sliding mode contribution. The obtained profiles (dashed yellow lines) are almost identical to the ideal ones (black lines), except for a short initial transient.

For what concerns control effort of the decentralized SMC, torque profiles for each joint are reported in Figure 5.10. Thanks to higher order algorithms, the control signals are almost chattering free, with smooth transients and limited amplitude. The required action is inside maximum torque limits for each joint, which makes them feasible to be applied to the real manipulator without damaging it.

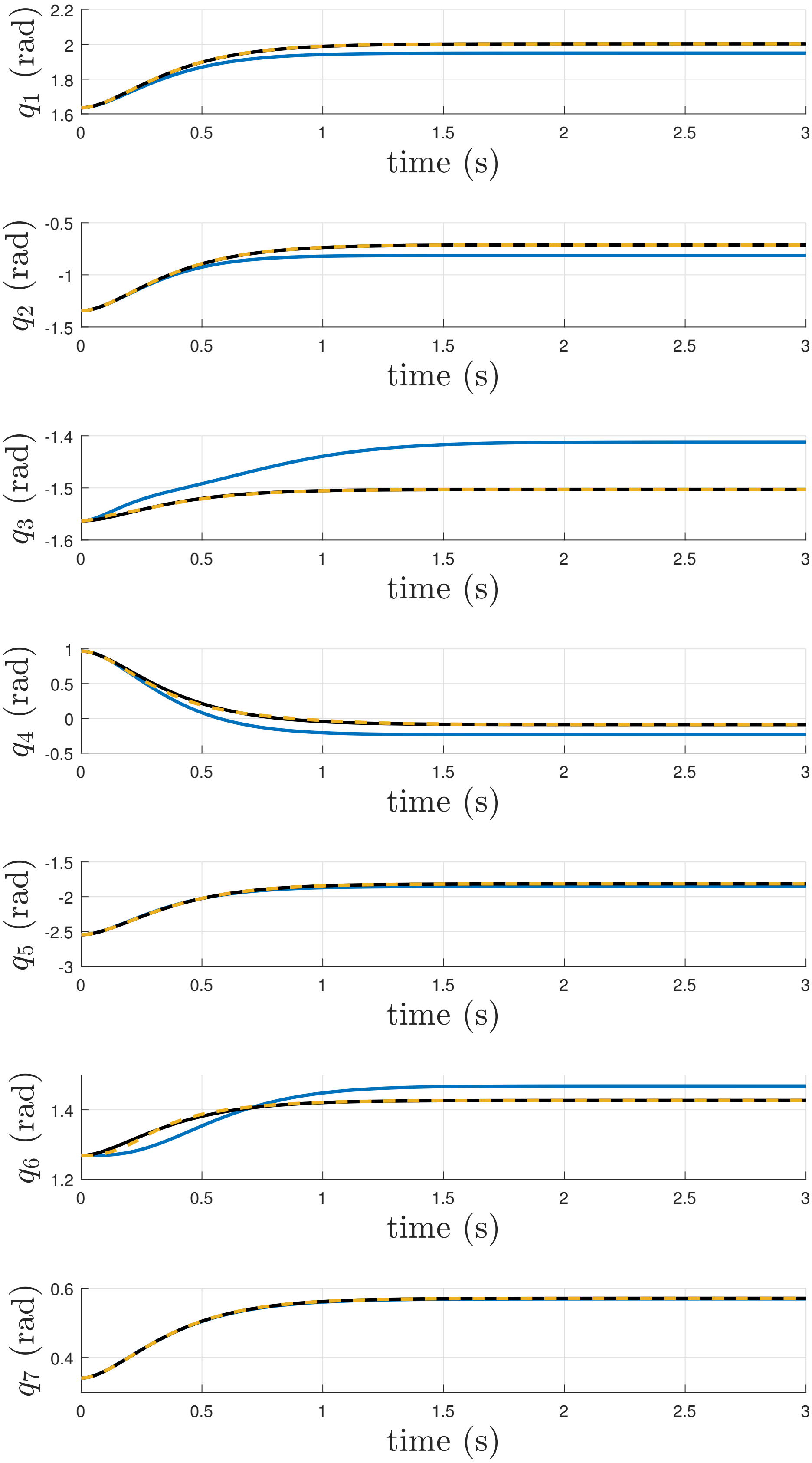


Figure 5.9: Joint position response with and without decentralised SMC

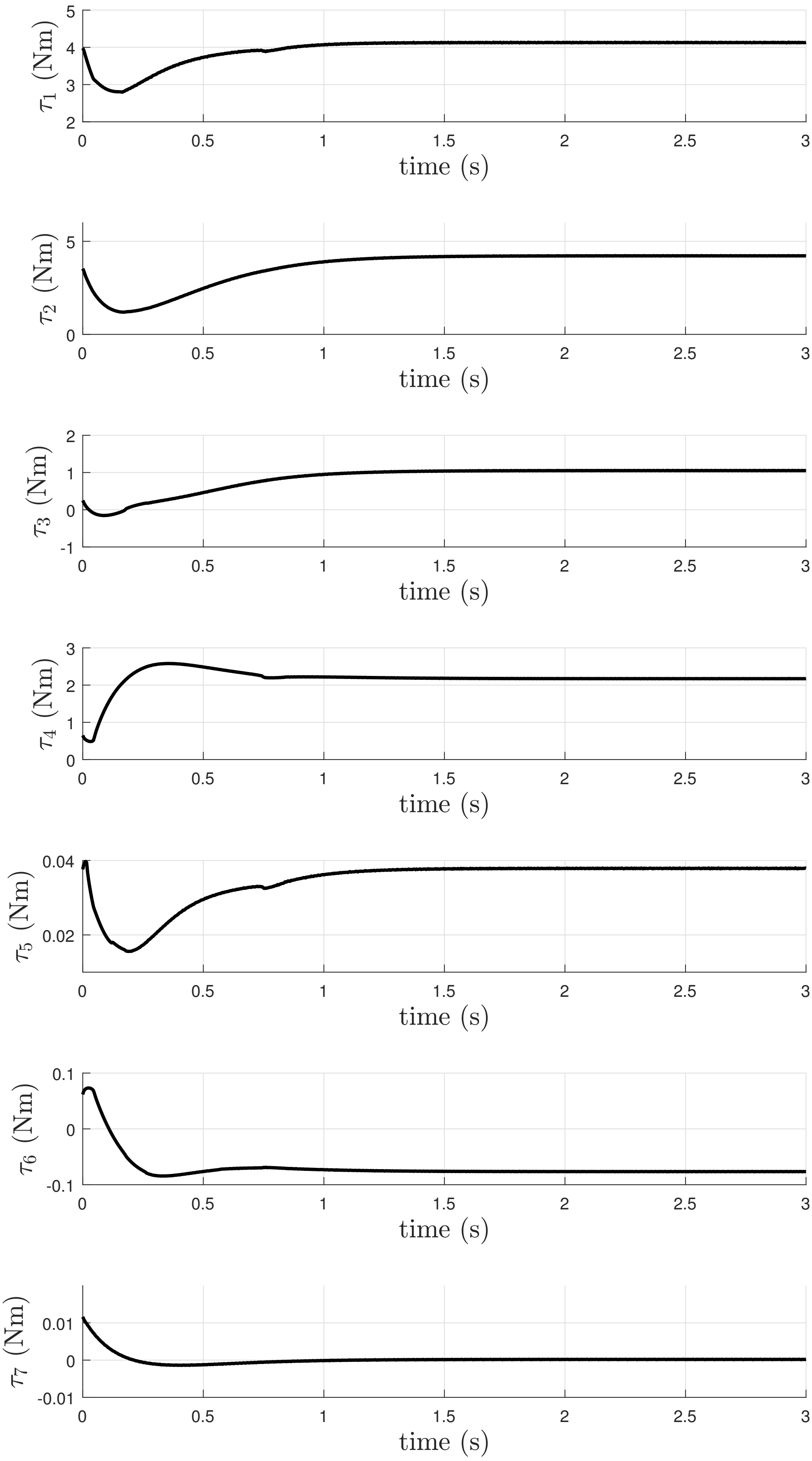


Figure 5.10: Torque control effort for decentralised SMC

5.6 Predictive SMC

In subsection 3.3.5, one of the main drawbacks associated with standard SMC was the *chattering effect*. Main cause of such phenomenon is the discontinuity in the control action along the sliding surface and solutions have been addressed to remove it (sections 3.4 and 3.5). Unfortunately, this is not the only source of persistent oscillations in the control action. Finite actuator dynamics and especially *time delay* lower the control frequency bandwidth and may introduce a significant loss of phase margin. Consequently, a *limit cycle* establishes around the origin of the sliding surface phase diagram (Figure 5.11) and we cannot converge towards the equilibrium point $(0, 0)$. Stable oscillatory behaviour is visible also in the control variable and in the position response.

If we know the amount of delay d which affects the control channel, it is possible to improve the performance of the controller by using, instead of the current value of $\sigma(t)$, its *d-step forward prediction* $\hat{\sigma}(t+d|t)$. This idea has been already exploited in literature [25, 26, 24] but it is generally applied to stochastic models (autoregressive, ARMA, etc.).

First of all, let's consider the general expression of σ for the slave device (3.37). Its d - step prediction is:

$$\hat{\sigma}(t+d|t) = \boldsymbol{F}\hat{\boldsymbol{e}}(t+d|t) + \hat{\Psi}(t+d|t) \tag{5.43}$$

where $\hat{\boldsymbol{e}}(t+d|t)$ and $\hat{\Lambda}(t+d|t)$ are, respectively, the d - step predictors for the tracking error and the external forces.

Since we have no information about the future evolution of Λ (it depends on external forces which are a-priori unknown), the best estimate we can perform is given by the current value of Λ:

$$\hat{\Lambda}(t+d|t) = \Lambda(t)$$

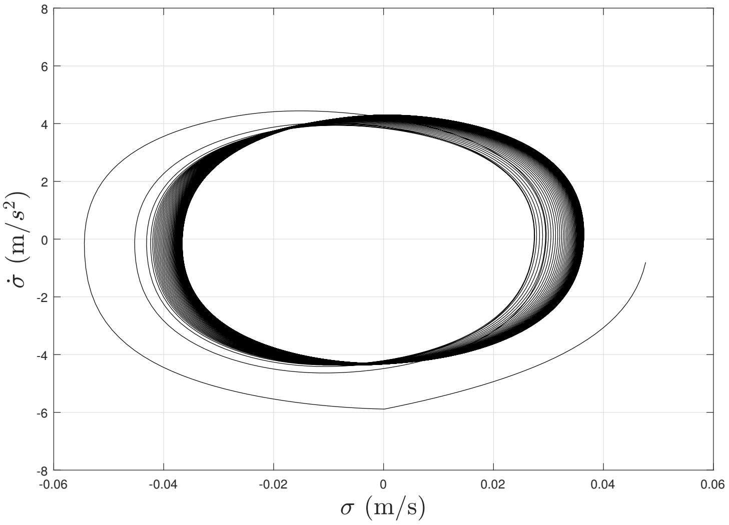


Figure 5.11: Sliding variable evolution in the σ - $\dot{\sigma}$ plane. A limit cycle is established around the origin due to a control action delay of 12 ms

For what concerns the error predictor, it is possible to write:

$$\hat{\boldsymbol{e}}(t+d|t) = \hat{\boldsymbol{x}}(t+d|t) - \hat{\boldsymbol{w}}(t+d|t)$$

where $\boldsymbol{x} = [\dot{q} \;\; q \;\; \int q]^\top$ is the extended state vector, written using joint coordinates, while $\boldsymbol{w} = [\dot{q}_{ref} \;\; q_{ref} \;\; \int q_{ref}]^\top$ is the reference input vector (for standard teleoperation they represent master joint coordinates).

To assess the future evolution of the state vector, it is necessary to recall the generic device dynamic model (3.64) and *discretize* it. To do so, *forward Euler method* has been adopted, because it is computationally efficient and does not require the solution of algebraic loops (it depends only on previous output time instants). Discrete time steps are labelled with symbol k, while the equivalent delay is $D = \frac{d}{T}$ (T is the sampling time). Then, the new discrete state has to be propagated up to D steps in the future and all the terms which depend on time instants between k and $k+D$ have to be made explicit by substituting the dynamic model in them. For the sake of clarity, passages for deriving the 2 - step predictor are here reported:

- Write the vectorial dynamic model (3.64), with control variable delayed by $d = 2T$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x})(t) + \boldsymbol{g}v(t-2T) \tag{5.44}$$

- Discretize (5.44) with forward Euler

$$\frac{1}{T}[\boldsymbol{x}(k+1)-\boldsymbol{x}(k)] = \boldsymbol{f(x)}(k)+\boldsymbol{g}v(k-2) \tag{5.45}$$

- Propagate state vector up to 2 steps in the future

$$\frac{1}{T}[\boldsymbol{x}(k+2)-\boldsymbol{x}(k+1)] = \boldsymbol{f(x)}(k+1)+\boldsymbol{g}v(k-1) \tag{5.46}$$

- Substitute $\boldsymbol{x}(k+1)$ in order to obtain $\boldsymbol{x}(k+2)$ as a function only of the current state

$$\boldsymbol{x}(k+2) = \boldsymbol{x}(k)+T[\boldsymbol{f(x)}(k)+\boldsymbol{f(x)}(k+1)]+\boldsymbol{g}[v(k-1)+v(k-2)] \tag{5.47}$$

- Expand the state vector to explicit $\boldsymbol{f(x)} = [x_2 \;\; x_3 \;\; \eta(\boldsymbol{x})]^\top$ and $\boldsymbol{g} = [0 \;\; 0 \;\; \zeta]^\top$

$$\begin{aligned} x_1(k+2) &= x_1(k)+T[x_2(k+1)+x_2(k)] \\ x_2(k+2) &= x_2(k)+T[x_3(k+1)+x_3(k)] \\ x_3(k+2) &= x_3(k)+\boxed{T[\eta(\boldsymbol{x})(k+1)+\eta(\boldsymbol{x})(k)]}+T\boxed{\zeta}[v(k-1)+v(k-2)] \end{aligned} \tag{5.48}$$

Boxed terms in (5.48) last equation represent uncertainties at current and future time steps. Since these quantities are a-priori unknown, the optimal 2 - steps predictor for x_3 is simply given by:

$$\hat{x}_3(k+2|k) = x_3(k)+T[v(k-1)+v(k-2)] \tag{5.49}$$

The convergence of such estimator towards the real value of $x_3(k+2)$ is dictated by the rejection capabilities of SMC. More the device model tends to the nominal one, smaller will be the prediction error. By substituting $x_3(k+1)$ and $x_2(k+1)$ with their respective expressions in the first two equations, formulations of $\hat{x}_2(k+2|k)$

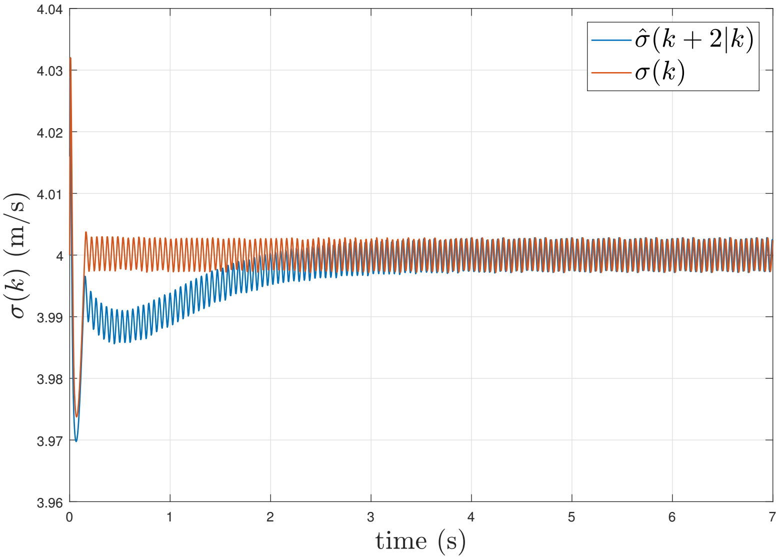


Figure 5.12: Standard sliding surface and its 2-step ahead prediction

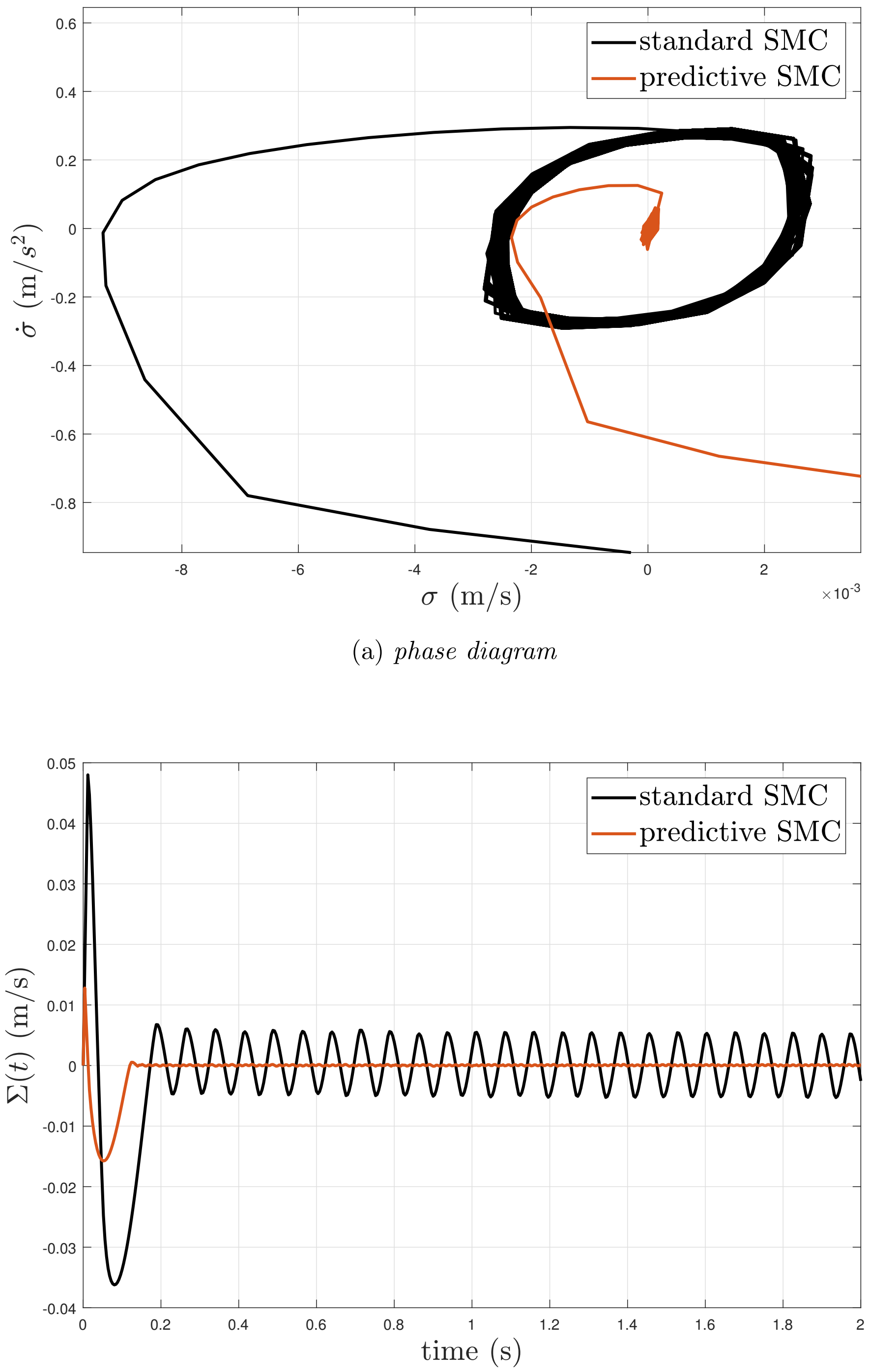
and $\hat{x}_1(k+2|k)$ are just a matter of trivial computations:

$$\begin{aligned}\hat{x}_1(k+2|k) &= x_1(k) + T[x_2(k-1) + x_2(k)] \\ &\quad + T^2[x_3(k-1) + x_3(k-2)] + T^3[v(k-3) + v(k-4)]\end{aligned} \tag{5.50}$$

$$\hat{x}_2(k+2|k) = x_2(k) + T[x_3(k-1) + x_3(k)] + T^3[v(k-2) + v(k-3)] \tag{5.51}$$

For what concerns $\hat{\boldsymbol{w}}(t+d|t)$, its expression strongly depends on the kind of input our system is dealing with. If it is possible to know in advance what will be the future desired trajectory, then $\hat{\boldsymbol{w}}$ could be set equal to that values. Instead, if there is no information on the future evolution of the reference signals, the best we can do is to set the predictor equal to the current value of $\boldsymbol{w}$ (as for the external inputs case).

In Figure 5.12, the comparison between the standard σ and its 2 steps ahead prediction is reported. As we can notice, the predictor takes some time to converge towards the sliding surface (approximately 2 s) but then it correctly tracks the future behaviour of σ. If we use $\hat{\sigma}(k+2|k)$ instead of σ *predictive SMC* is obtained. Improvement in sliding precision and stability is visible in Figure 5.13. The limit cycle has been almost removed.



(a) *phase diagram*

(b) *sliding surface*

Figure 5.13: predictive SMC compared to standard one applied on a system affected by control variable delay

Chapter 6

Experimental results

In this chapter, some of the control techniques presented for multi - DOFs manipulators will be applied to the `ABB IRB14000 YuMi`$^{\circledR}$ dual arm robot.

In the first section, a brief introduction of the experimental set-up will be given, as well as a description of the robot operative conditions. Then, the robot external controller architecture will be analysed, since it represents the interface between the simulation environment and the robot internal firmware. Here, the two main problems encountered in the implementation of the various control techniques are shown (low pass input filter and friction). Impedance tracking experiments have been performed on single joints, adopting a decentralised control strategy. Different sliding mode algorithms have been tested and their performance have been compared.

Finally, a teleoperation task has been performed, piloting the slave arm through the master arm. A joint level tracking has been implemented using the proposed algorithms. Pros and cons of this kind of approach will be reported at the end.

6.1 Experimental set-up

In order to implement the control strategies developed in previous chapters, a dual arm robotic manipulator has been adopted. Thanks to this choice, it is possible to simulate teleoperation activities by selecting one of the arm as *mater device*, while the other as *slave device*. Although this is not the usual operative condition for this kind of manipulator, it has the advantage of requiring *one - to - one* teleoperation

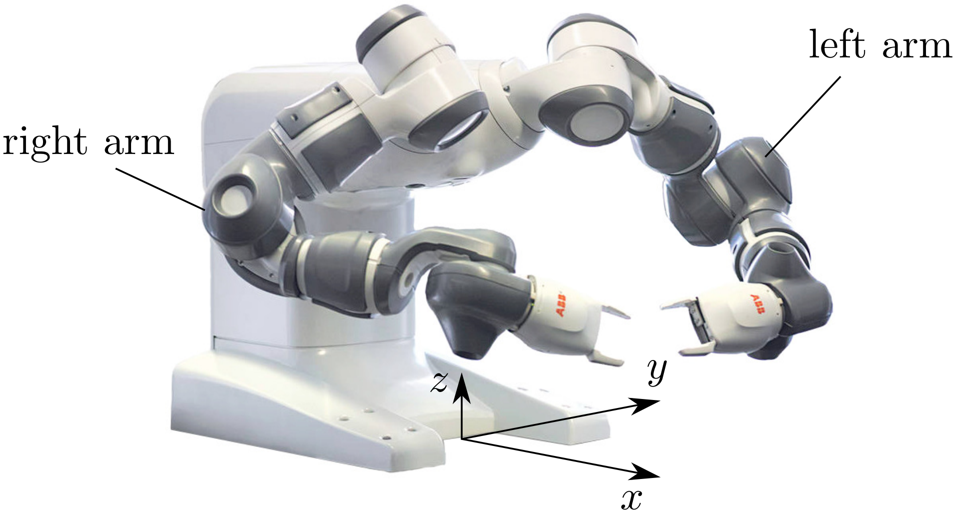


Figure 6.1: Dual arm robot adopted for the experiments

mapping (i.e. the possibility of replicating the exact joint movements of master device onto the slave arm). In this way, it is possible to apply an error definition of the decentralized type (see (5.5)), where the scaling factor can be used to magnify the master movements allowing better slave precision. A visual representation of the manipulator on which the experiments have been performed is shown in Figure 6.1. The right arm has been used as a master device, while the left arm acted as a slave device.

The user interfaces with the master arm mainly by moving the end effector in the operational space in front of the robot. Thanks to the decentralized architecture, he/she can move also single portions of the manipulator by acting directly on the joints of interest. At the slave end effector, a tool tip can be attached to interact with the external environment.

Let's suppose to divide the manipulator's operative region into two halves by a plane passing through the origin and parallel to the z axis. To *avoid self collisions*, master and slave have to operate in their respective spatial regions, otherwise they might interfere. Thus, joint mapping between the two arms has been defined in such a way that they are *specular* with respect to the middle plane dividing the two subspaces. With this choice, operative regions for the master and slave subsystem are maximized without the possibility of self collisions. If the operator finds *difficult* to operate the robotic arm in this configuration, it is possible to change joint mapping in order to obtain the same end effector position and orientation. In this case, particular attention has to be posed on the operative region intersection between master and slave arms.

6.2 External force estimator

Both slave and master devices *are not* equipped with neither joint torque sensors nor end effector force sensors, therefore potential external interactions should be detected via proper *force estimators*. The one implemented on the `YuMi` has been proposed in [52, 15] and it is based on the computation of the residual torque vectors.

Let's suppose that the left/right `YuMi` arm can be described by the following dynamic equation:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F_v}(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau_{ext}} \tag{6.1}$$

where $\boldsymbol{B}(\boldsymbol{q})$, $\boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, $\boldsymbol{F_v}(\dot{\boldsymbol{q}})$, $\boldsymbol{g}(\boldsymbol{q})$ are, respectively, the inertia and Coriolis - centrifugal matrices, the estimated joint friction contribute and the gravitational load. $\boldsymbol{\tau}$ is the input torque signal given at each joint, while $\boldsymbol{\tau_{ext}}$ is the external torque exerted by the environment. Now write the generalized momentum $\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$ and its derivative:

$$\begin{aligned} \dot{\boldsymbol{p}} &= \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} \\ &= \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\tau} + \boldsymbol{\tau_{ext}} - \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} - \boldsymbol{F_v}(\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) \end{aligned} \tag{6.2}$$

To remove the dependency on $\dot{\boldsymbol{M}}(\boldsymbol{q})$, it is possible to exploit the *skew-symmetric* property [73]:

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{C}^{\top}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \tag{6.3}$$

obtaining the final expression:

$$\dot{\boldsymbol{p}} = \boldsymbol{C}^\top(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\tau} + \boldsymbol{\tau_{ext}} - \boldsymbol{F}_v(\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) \tag{6.4}$$

From the momentum vector it is possible to derive a *residual* r:

$$\boldsymbol{r} = \boldsymbol{K}\left[\boldsymbol{p} - \int_0^t \left(\boldsymbol{C}^\top(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{\tau} + \boldsymbol{r} - \boldsymbol{F}_v(\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q})\right) dt\right] \tag{6.5}$$

where $\boldsymbol{K} = \text{diag}(k_1, \dots, k_7)$ is a constant positive definite matrix. Equation (6.5) represents 7 decoupled first order linear systems with stable poles given by $-k_1, \dots, -k_7$. The residual dynamics are excited by the external torque vector $\boldsymbol{\tau_{ext}}$, that serves as input. Indeed, deriving (6.5), it is possible to write $\boldsymbol{r}$ in a more familiar form:

$$\dot{\boldsymbol{r}} = -\boldsymbol{K}\boldsymbol{r} + \boldsymbol{K}\boldsymbol{\tau_{ext}} \tag{6.6}$$

Therefore, the computed residual is a low pass filtered version of the external torques acting on the manipulator. If we suppose that the environmental interaction is confined at the end effector, external torques measured at joint level can be projected using the inverse of relation (5.14).

$$\boldsymbol{h} = \boldsymbol{J}^+(\boldsymbol{q})\boldsymbol{r} \tag{6.7}$$

The proposed estimator does not require the knowledge of joint accelerations, using only the information coming from the input joint torques. Moreover, its stable dynamics guarantee that after the interaction has occurred, equilibrium $\boldsymbol{r} = \boldsymbol{0}$ is reached asymptotically.

However, the algorithm shown to compute $\boldsymbol{r}$ is strongly based on the correct knowledge of the manipulator. Thus, any unmodelled mismatch/disturbance will reflect into a "fictitious" external torque acting on the robotic arm. If we are using this information to control the manipulator impedance, these wrong torque estimations could become dangerous for the human operator.

That is why this residual external torque computation can be safely introduced only when the manipulator model is accurately known, in order to reduce as much as possible unpredictable robot behaviours.

As it will be shown in the next sections, YuMi left and right arms models suffer from strong unmodelled dynamics (first of all, joint friction), that will bias the residual estimator giving wrong external torque measurements. For this reason, experiments in the next sections have been based on *free motion* trajectories. Consequently, $\boldsymbol{\tau_{ext}}$ computed according to (6.5) is equal to zero (after a proper signal conditioning). However, to develop further trials, force estimate is fundamental to obtain the force feedback signal to be given to the master device.

6.3 External controller

For both arms, ABB YuMi is equipped with internal industrial controllers, that allow the robot to perform assigned motions both in the joint space and in the Cartesian space. Their structure is composed of independent PID controllers, one for each joint. They read the angular positions $\boldsymbol{q}(t)$ through encoders mounted on the motors and differentiate it to derive the angular speed vector $\dot{\boldsymbol{q}} = \frac{\Delta \boldsymbol{q}}{T}$. This information is used to close 7 independent nested P-PI control loops. A schematic representation of YuMi control unit is visible in Figure 6.2. The dashed box element constitutes the internal controller, whose structure cannot be modified. It outputs the reference torque signal to the motor controller, while it receives as inputs the position and velocity reference signals and a torque feed-forward τ_{fw} which is usually set to zero.

In order to implement the control techniques analysed in chapter 5, it is necessary to interact with the industrial ABB controller. To do so, a research interface developed by Lund university, called OPCOM, has been adopted [6]. This program allows to connect to the ABB YuMi an *external controller* located on a Linux device. Through this computer, it is possible to use proper MATLAB - Simulink libraries to read and write the signals coming from the internal controller. In particular, it is possible to read the joint angular positions $\boldsymbol{q}$ and the commanded torques to the joints $\boldsymbol{\tau}$. Parameters K_{pp}, K_{pv} and K_{iv} can be read and written in order to modify the behaviour of the internal PID controllers. Reference trajectories $\boldsymbol{q_{ref}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{ref}}$ are

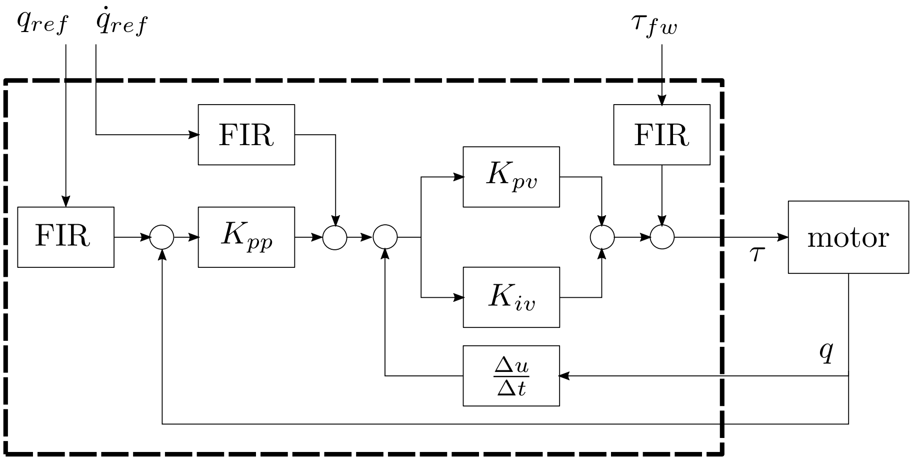


Figure 6.2: Block diagram representation of the YuMi control scheme. The dashed box indicates the internal proprietary control unit

accessible as well as the torque feed-forward $\boldsymbol{\tau_{fw}}$.

According to the previous considerations, when the external controller is attached to the robot, there are two possible operative conditions:

- Keep the internal ABB control **active** and generate reference signals for this "low level" controller ($\boldsymbol{q_{ref}}$ and $\boldsymbol{\dot{q}_{ref}}$). These signals will be chosen following an "high level" control strategy (force control, MPC, etc.);

- **Deactivate** the internal ABB controller by imposing all the PID gains equal to zero. Doing so, $\boldsymbol{q_{ref}}$ and $\boldsymbol{\dot{q}_{ref}}$ inputs become meaningless and the entire control action has to be given through the $\boldsymbol{\tau_{fw}}$ channel;

The first operative choice has the great advantage of guaranteeing great tracking accuracy of the reference positions and speeds thanks to the built in ABB control loops. This ultimately allows to focus on other control objectives without caring on how the chosen trajectory will be realised. Instead, the second operative condition leaves complete freedom to the external control action to generate the desired torque profile, except for the maximum input frequency (FIR filter on the channel). This means that $\boldsymbol{\tau_{fw}}$ should be computed by taking into account all the non linearities and couplings that were cancelled by PIDs.

According to the theory developed, the second option is the one coherent with the structure of SMC analysed up to now. Therefore, the external controller will turn off internal one and will generate the closed - loop control signal by acting on the feed - forward torque. The sampling time at which the external controller works is equal to 4 ms. That is why in previous simulation this fixed sampling time has been adopted, in order to avoid discretization issues during the implementation.

Proper routines have already been designed for the real - time computation of dynamic matrices $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{g}$ as well as Jacobian and forward kinematic expressions for each arm. Since the dynamic model is particularly complex, the real - time execution of that function is not feasible (it exceeds the sampling time). Therefore, the right arm dynamic parameters are calculated on the external computer (faster than the manipulator one) and then sent in real -time to the robotic arm.

Through the `OPCOM` interface, it is also possible to interact with some variables during its execution of the program. This mechanism will be mainly adopted to start the routine and control the internal program evolution.

All the measurements reported in this chapter have been logged through the external controller, therefore they are constrained to be the accessible inputs/outputs of the control scheme in Figure 6.2. Internal `ABB` quantities such as PID torques cannot be directly observed.

6.3.1 FIR identification

As we can notice, in scheme 6.2 there is a FIR filter which connects the feed-forward signal $\boldsymbol{\tau_{fw}}$ to the torque control applied to the motor $\boldsymbol{\tau}$. This block plays a *crucial* role in the stability of the SMC controller, because it introduces an actuator dynamics that lowers significantly the bandwidth of the desired control action. To visually analyse such effect, it is possible to look at Figure 6.3, where the feed - forward signal is compared with the real one applied to the system. If we exclude the one - step delay introduced by the reading procedure ($\boldsymbol{\tau}(t) = \boldsymbol{\tau_{fw}}(t-1)$), the remaining shift and the attenuation introduced in $\boldsymbol{\tau}$ are caused by the filter.

In order to compute the filter dynamics and understand the available bandwidth allowed for the external controller, a frequency response analysis has been performed. For the sake of brevity, only the first joint FIR identification has been reported. According to [54], one method of determining empirically the dynamic characteristics

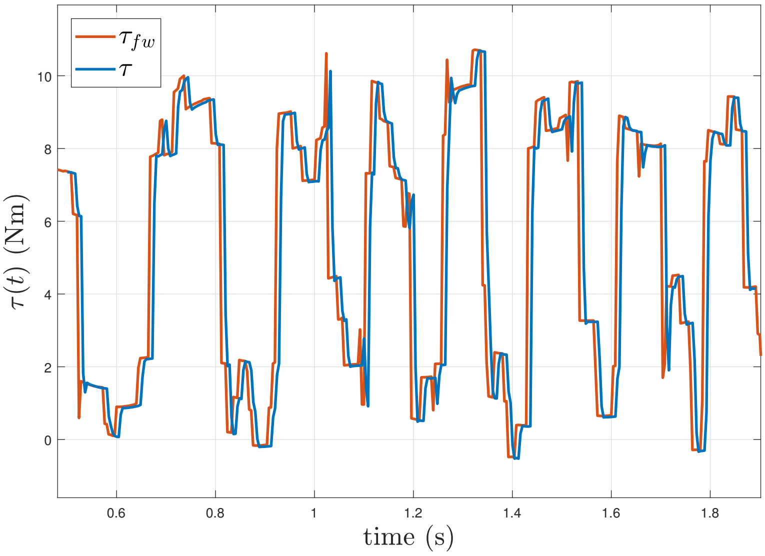


Figure 6.3: Feed - forward torque signal vs. actual torque applied to the joint motor

of a system is by *cross - correlating* the input with the output. To perform this, the unknown transfer function can be computed by calculating:

$$F(j\omega) = \frac{S_{xy}(j\omega)}{S_{xx}(j\omega)} \tag{6.8}$$

where subscripts $x = \tau_{fw1}$ and $y = \tau_1$ are, respectively, the input and output signals for the first filter. The term $S_{xy}(j\omega)$ represents the cross power spectrum between input and output signals, while $S_{xx}(j\omega)$ is the power spectrum of the input. Therefore, $F(j\omega)$ represents the amount of correlation between the input and output channels normalized with respect to the input power.

To compute $S_{xy}(j\omega)$ and $S_{xx}(j\omega)$, it is necessary to find the spectra of τ_{fw1} and τ_1 using Discrete Time Fourier Transform (DTFT). Once $T_{fw1}(j\omega)$ and $T_1(j\omega)$ have been calculated, the power spectra needed in (6.8) are:

$$\begin{aligned} S_{xy}(j\omega) &= T_1(j\omega)T_{fw1}(j\omega)^* \\ S_{xx}(j\omega) &= T_{fw1}(j\omega)T_{fw1}(j\omega)^* \end{aligned} \tag{6.9}$$

where superscript $*$ stands for the complex conjugate operator. The identified FIR transfer function magnitude $|F(j\omega)|$ can be then plotted on a logarithmic scale (blue line in Figure 6.4).

As we can notice, $F(j\omega)$ resemble the shape of a low pass filter, with cut off frequency at approximately 100 rad s^{-1}. To perform a more formal analysis, `MATLAB` *System Identification Toolbox* has been adopted. First of all, a FIR filter has been identified, since it should be the actual filter placed between $\boldsymbol{\tau_{fw}}$ and $\boldsymbol{\tau}$. To keep the complexity limited and avoid over - fitting phenomenon, a fourth order FIR has been adopted (yellow line in Figure 6.4):

$$\hat{F}_{FIR}(z) = 6.993 \times 10^{-3} + 8.846 \times 10^{-2}z^{-1} + 2.048 \times 10^{-1}z^{-2} + 2.327z^{-3} \tag{6.10}$$

However, with this choice fitting to the estimated data is poor (44.6%), especially at low frequency (offset with respect to the unitary gain). A better fitting could be obtained by increasing the number of coefficients, at the price of a complex filter structure. Moreover, fitting is extended also to high frequency ripples which are mainly due to noise.

Since we are not interested in high frequency dynamics, to detect the maximum allowed bandwidth a continuous transfer function can be used. According to the identification toolbox, a good fitting precision can be obtained with the second order transfer function (red line in Figure 6.4):

$$\hat{F}_2(s) = \frac{-24.42s + 2.486 \times 10^4}{s^2 + 304.3s + 2.486 \times 10^4} \tag{6.11}$$

which has 2 complex poles at $-152.14 \pm 41.28j$ and a fitting accuracy of 97.25%. The -3 dB bandwidth of this filter is approximately at 107 rad s^{-1}, which corresponds to 17 Hz. $\hat{F}_2(s)$ has a high frequency zero one decade faster than the two poles and unitary gain. Loosing a bit of accuracy, $F(j\omega)$ can be fitted quite well also with a single pole low pass filter of the following form (green line in Figure 6.4)

$$\hat{F}_1(s) = \frac{74.93}{s + 72.9} \tag{6.12}$$

Fitting accuracy is reduced but it is still quite high if compared with the FIR model (91.8%). The transfer function gain is almost unitary and the single pole is located

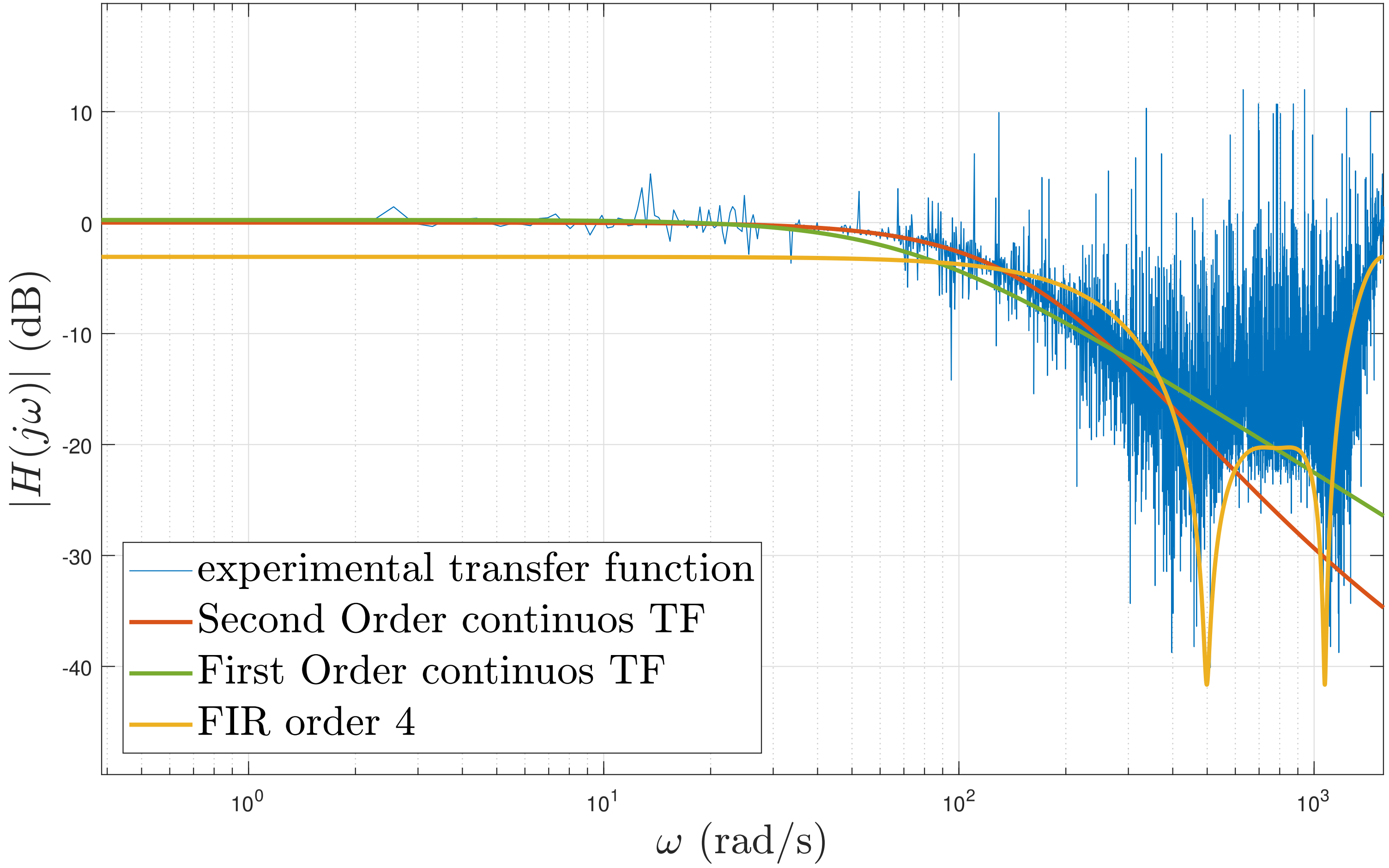


Figure 6.4: Identification of the first joint FIR transfer function from experimental input - output response

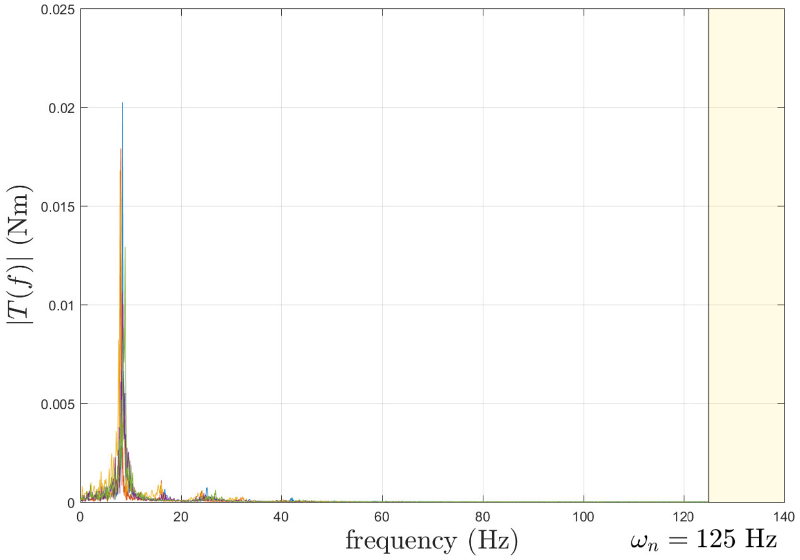


Figure 6.5: Frequency spectra for the seven joints input torque during sliding mode motion. Allowed control bandwidth is significantly smaller than the Nyquist frequency 125 Hz

at $-72.9\,\mathrm{rad\,s^{-1}}$. The $-3\,\mathrm{dB}$ bandwidth is equal to $79\,\mathrm{rad\,s^{-1}}$ ($\approx 12.5\,\mathrm{Hz}$).

Thanks to its simplicity, the last identified filter has been chosen to model the internal `ABB` FIR and to describe main actuator dynamic range. It is worth noticing that the maximum frequency limitation imposed by `ABB` firmware to external control signals is significantly lower than the sampling time at which the controller is operating (Nyquist frequency equal to 125 Hz). This conservationism has been probably introduced to prevent joint mechanisms from being damaged by torque references at too high frequency. If on the one hand this filter allows not to consider elastic effects due to transmission coupling and other high frequency dynamics, on the other it causes severe oscillations in the SMC joint control loops due to the low bandwidth available for the switching function. As it can be seen in Figure 6.5, this effect is common to all joints and it justifies the assumption of the same filter mounted on each feed - forward channel.

6.3.2 Friction compensation

When dealing with joint motion control, friction introduced by transmissions and links can be significant. Since we are designing the control law in order to be

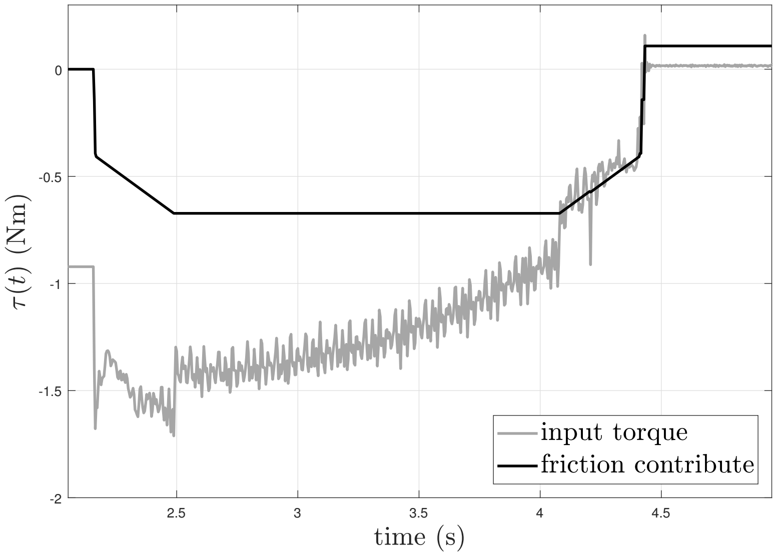


Figure 6.6: Torque control signal applied to joint 3 and the amount of effort required to overcome friction

robust with respect to model uncertainties, we should expect that friction effects will be properly rejected by SMC. Unfortunately, this is not completely true. If the unmodelled friction term is comparable, or even greater than the joint dynamic model, the required sliding action K needed to bring the system into sliding mode can be considerably high. This fact, together with the previous considerations about the maximum achievable control bandwidth, generates oscillations in the control signal with very high amplitude, that could damage motors and joint structures.

To better explain the relevance of the friction contribution inside the manipulator model, in Figure 6.6 the torque required to control the TVP trajectory of joint 3 is reported. As it can be noticed, the amount of friction that the controller needs to compensate is at least equal to half of the total amount of torque that should be supplied by $\boldsymbol{\tau_{fw}}$. This means that one of the main component of the manipulator model is uncertain and to compensate it SMC should be tuned too aggressively.

Situation is even worse when wrist joints are analysed, because `YuMi` is not equipped with motor brakes for them. Therefore, when the manipulator is off, it is possible to move the last three joints freely and the perceived friction is quite strong. If we add the fact that manipulator dynamics close to the end effector are not too relevant (low inertia and gravity), we easily come up with a dynamic model which is almost *totally* uncertain, since it is mainly composed by the friction term.

As it will be seen in the next section, applying SMC directly without a rough knowledge of the friction entity is not a good idea. That is why *joint friction estimation* has been performed. Through a series of experiments, a friction model has been fitted for each joint, with particular focus for low speed dynamics which are typical of teleoperation applications. For a detailed report of this procedure, the reader can refer to appendix B.

Once the friction model has been retrieved, it is used as an open loop *feed - forward contribute* inside the inverse dynamics scheme. Doing so, the effect of friction uncertainty is significantly reduced, allowing to reduce sliding mode gains and obtain a more stable and predictable joint behaviour. Obviously, as every open loop actions, it is fully based on the accuracy of the experiments used to identify friction (their values become unreliable if operative conditions change).

Although very precise friction models have been presented in appendix B, experiments proved that *overestimation* of the friction inverse response (decreasing viscous coefficient at low speeds) causes high frequency oscillations around the steady - state position. Indeed, no steady state can be reached and friction compensation becomes useless. To overcome this issue, a simpler model, despite being less precise, has been fitted:

$$\boldsymbol{F_v}(\dot{\boldsymbol{q}}) = \begin{cases} \boldsymbol{\tau_c^+} + \boldsymbol{\kappa}^+\dot{\boldsymbol{q}} & \dot{\boldsymbol{q}} > \dot{\boldsymbol{q}}_{\boldsymbol{th}} \\ \boldsymbol{\tau_c^-} + \boldsymbol{\kappa}^-\dot{\boldsymbol{q}} & \dot{\boldsymbol{q}} < -\dot{\boldsymbol{q}}_{\boldsymbol{th}} \\ \frac{1}{2}\left[\frac{\boldsymbol{\tau_{th}^+}-\boldsymbol{\tau_{th}^-}}{\dot{\boldsymbol{q}}_{\boldsymbol{th}}}\dot{\boldsymbol{q}} + \boldsymbol{\tau_{th}^+} + \boldsymbol{\tau_{th}^-}\right] & -\dot{\boldsymbol{q}}_{\boldsymbol{th}} \leq \dot{\boldsymbol{q}} \leq \dot{\boldsymbol{q}}_{\boldsymbol{th}} \end{cases} \tag{6.13}$$

where $\boldsymbol{\kappa}^+$ and $\boldsymbol{\kappa}^-$ are the viscous coefficients vectors, $\boldsymbol{\tau_c^+}$ and $\boldsymbol{\tau_c^-}$ are the Coulomb coefficients vectors and $\dot{\boldsymbol{q}}_{\boldsymbol{th}}$ is the speed threshold adopted to linearise the friction model ($0.001\,\mathrm{rad\,s^{-1}}$). Thanks to this parameter, it is possible to avoid the discontinuity in the standard friction response, ultimately decreasing chattering phenomena caused by speed measurement noise. For speed values between $\dot{\boldsymbol{q}}_{\boldsymbol{th}}$ and $-\dot{\boldsymbol{q}}_{\boldsymbol{th}}$, linear interpolation has been performed ($\boldsymbol{\tau_{th}^+} = \boldsymbol{\tau_f}|_{\dot{\boldsymbol{q}}_{\boldsymbol{th}}}$ and $\boldsymbol{\tau_{th}^-} = \boldsymbol{\tau_f}|_{-\dot{\boldsymbol{q}}_{\boldsymbol{th}}}$) (Figure 6.7).

Identified values for the seven joints of the slave (left) arm are reported in table 6.1. For the master arm this identification has not been performed, because joint

joint n.	τ_c^+	τ_c^-	κ^+	κ^-
1	1.9	−1.9	0.05	0.05
2	1.5	−1.5	1.4214	1.2527
3	0.4	−0.35	0.3188	0.5255
4	0.55	−0.54	0.75	0.8
5	0.2289	−0.2543	0.0273	0.0279
6	0.1414	−0.2156	0.0318	0.0213
7	0.1033	−0.1135	0.0474	0.0436

Table 6.1: Identified coefficient for the simplified friction expression

friction introduces a damping coefficient which *improves* the human manipulability.

6.4 Impedance tracking

In this section, the decentralized controller results for the slave arm are presented. These preliminary experiments are performed according to the following points:

1. Master (right) arm has been left with the internal `ABB` Proportional Integral Derivative (PID) controller in a given position far from the left arm, in order to not interfere with this one;

2. Slave (left) arm internal PID are disabled and an inverse dynamic control has been performed through $\boldsymbol{\tau_{fw}}$ in order to maintain it in the initial commanded position;

3. The impedance control is given as control input $\boldsymbol{v}$ to the inverse dynamic scheme, with error impedance profile defined as stated in equation 5.4;

4. According to the chosen parameters and the reference signal given as $\boldsymbol{q_{ref}}$, controlled joints should perform the desired movements reaching the stable final positions;

6.4.1 Step reference

In this case, $\boldsymbol{q_{ref}}$ is simply defined as the target joint positions that the manipulator should reach starting from the initial pose $\boldsymbol{q_0}$. Acceleration and speed references are

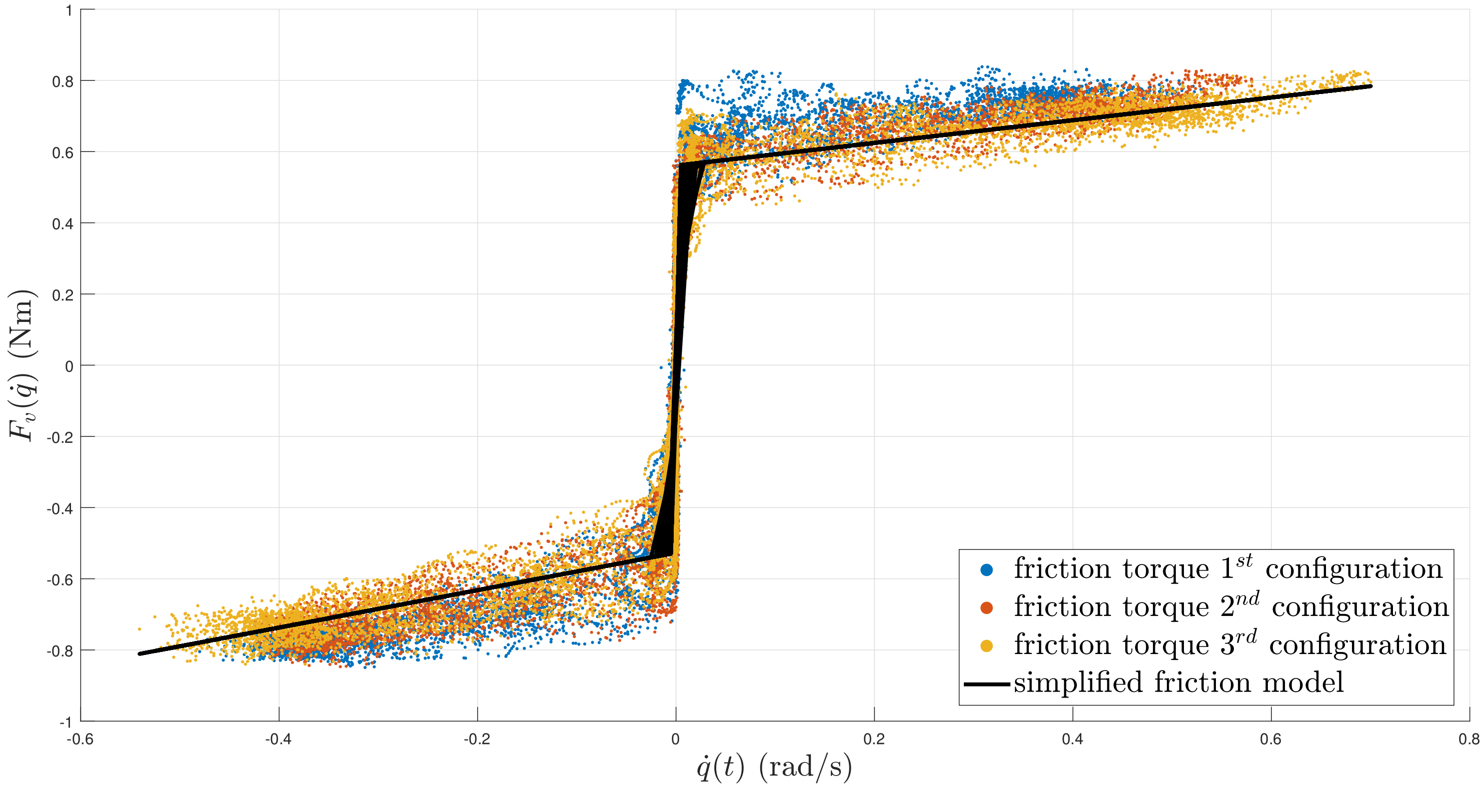


Figure 6.7: Experimentally computed friction for joint 3, tested for different robot configurations. The identified simplified model is shown with the black line

joint n.	K_{smc} $(\mathrm{rad\,s^{-2}})$	k_1 $(\mathrm{rad\,s^{-3}})$	k_2 $(\mathrm{rad\,s^{-3}})$
1	10	20	10
2	10	20	10
3	5	20	10
4	15	14	5
5	130	120	80
6	80	120	80
7	120	120	80

Table 6.2: Tuning parameters chosen for the sigmoid SMC approximation and the Super Twisting Algorithm in the step experiment

chosen equal to zero and the test is performed in order to check if the tracking impedance profile imposed at each joint is followed by the manipulator or not. The impedance controller is tuned so that the n decoupled closed - loop systems (5.4) should have 2 stable poles in -1 ($\bar{\boldsymbol{B}} = 10\boldsymbol{I_7}$, $\bar{\boldsymbol{C}} = 20\boldsymbol{I_7}$ and $\bar{\boldsymbol{g}} = 10\boldsymbol{i_7}$).

After a few trials, it has been noticed that the sliding variable defined in (5.6) suffers from integral drift due to the presence of steady - state error. This effect could make it insensitive to further reference changes or even leading to numerical overflow, causing the robot to stop working. While in the next subsection this problem has been removed by properly saturating the upper and lower bounds of $\boldsymbol{\sigma}$, for this earlier experiments it has been preferred to *deactivate* $\boldsymbol{\sigma}$ integral part. Doing so, joint dynamics converges to n decoupled first order linear systems, that can be found by imposing:

$$\boldsymbol{\sigma} = \bar{\boldsymbol{B}}\dot{\tilde{\boldsymbol{q}}} + \bar{\boldsymbol{C}}\tilde{\boldsymbol{q}} = 20\boldsymbol{I_7}\dot{\tilde{\boldsymbol{q}}} + 10\boldsymbol{I_7}\tilde{\boldsymbol{q}} \tag{6.14}$$

equal to zero. These systems have a single pole equal to -0.5 and their time evolution is represented in Figures 6.9, 6.10 and 6.11 (blue lines). In these figures, joint variables evolution in also presented, with different controllers applied to the scheme. Firstly, standard impedance controller has been tested (green line). Since it is not equipped with any type of disturbance rejection, the tracking performance is very poor. Then, sliding mode is added to the structure, improving the system robustness and, consequently, the tracking performance.

Given the bandwidth limitations due to the input filter, chattering will surely

joint n.	impedance (rad)	sigmoid SMC (rad)	STA algorithm (rad)
1	0.2718	0.0051	0.0036
2	0.2562	0.0034	0.0033
3	0.0248	0.0137	0.0016
4	0.7825	0.0434	0.0111
5	0.6120	0.0101	0.0061
6	0.1336	0.0050	0.0017
7	0.1426	0.0179	0.0118

Table 6.3: RMS error associated to impedance control, sigmoid SMC control and Super Twisting Algorithm for step response trajectories

occur if standard SMC is directly applied. Therefore, chattering avoidance methods have been implemented (STA and sigmoid SMC with $\delta = 0.5$). To remove the reaching phase, ISMC is implemented as well. The tuning parameters for each joint can be seen in Table 6.2.

The overall control torques imposed as $\boldsymbol{\tau_{fw}}$ are visible in Figure 6.12 for the first four joints. This choice has been done to empathize the capability of STA to maintain torque ripples confined in the initial response transient, reaching steady - state values. This is not the case for sigmoid SMC, whose ripples continue also when steady state is reached.

In Figure 6.13, evolution of the sliding surfaces for the first four joints is reported. Also in this case, it is interesting to notice that sigmoid SMC guarantees only boundedness of the surface, while STA imposes its value to zero. This is done at the price of increasing oscillations around the sliding surface.

For a more quantitative comparison, in Table 6.3 the RMS error between the three proposed controllers and the reference ideal trajectory have been computed. Thanks to SMC, it is possible to gain more than an order of precision with respect to the impedance controller alone. Between the two methods of chattering avoidance, STA is the best in terms of tracking accuracy, which is however very close to the one achieved by the sigmoid SMC.

However, when STA is applied to systems affected by high static friction levels (as in the case of `YuMi` robotic arm), it tends to suffer from *stick - slip phenomena* due to the integral of the sliding surface which composes its formulation (3.59). Although

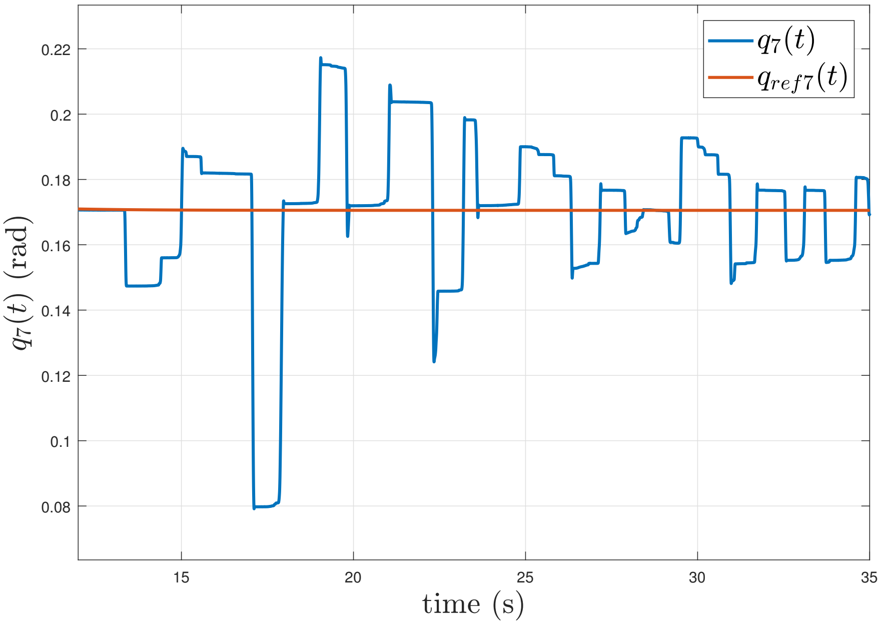
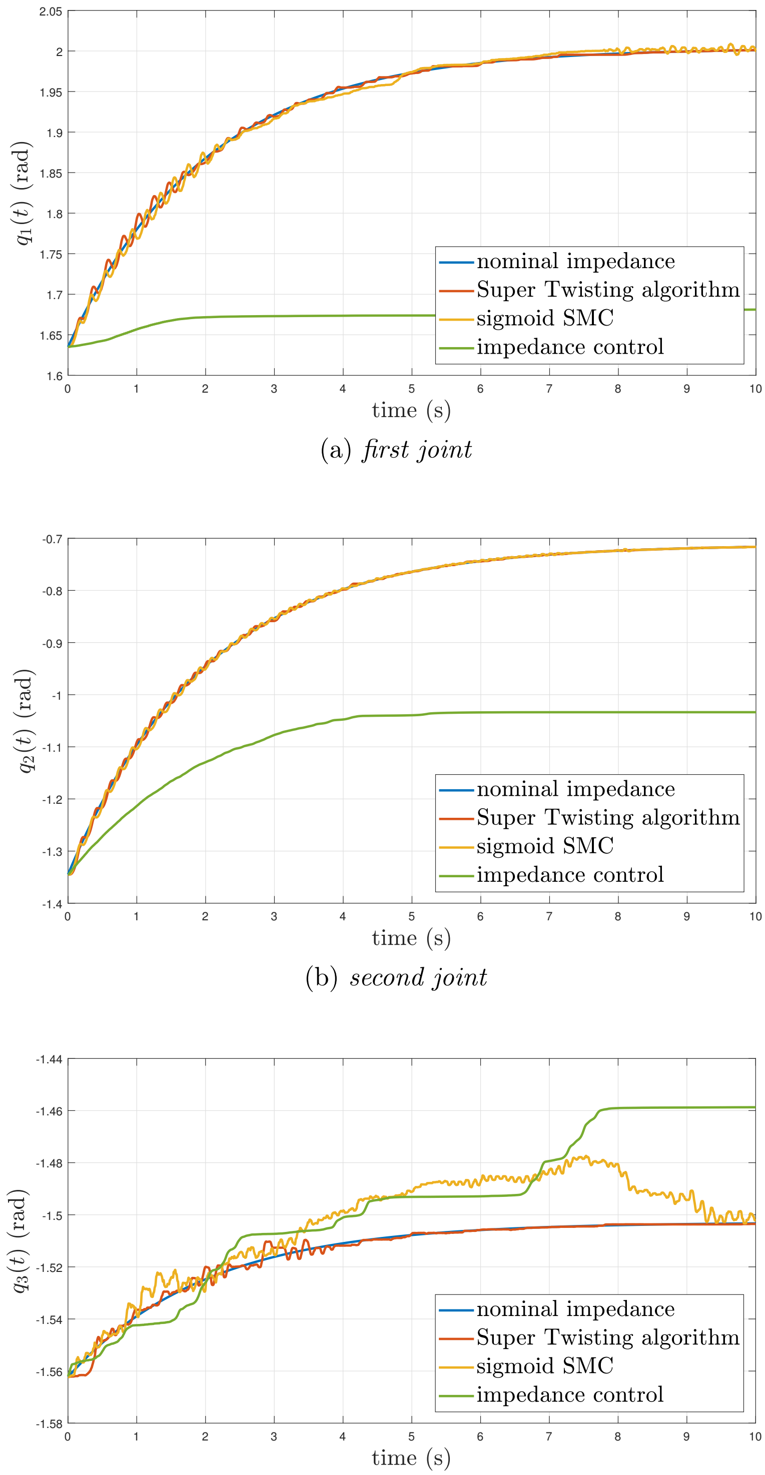


Figure 6.8: When friction at low speed is very high, integral action can cause stick - slip phenomena to occur

that term is the one which guarantees its asymptotic stability, the impossibility of reaching the exact steady - state value in the position response triggers a *limit cycle* around that value (Figure 6.8). This effect is more relevant for wrist joints, since they are the ones where friction contribution is dominant with respect to the rest of the dynamic model. For this reason, in the next subsection sigmoid approximation SMC has been adopted instead of STA. Despite its lower precision, it does not suffer from stick - slip (it does not contain any integral action). Moreover, it is simpler to tune. Since the tracking effort required for smooth trajectories is smaller than the one for step reference, K_{smc} can be raised enough to obtain a better tracking error than the one shown in Figure 6.9 for sigmoid SMC. This obviously at the price of a more oscillating response, which represents however the control trade - off for systems affected by high friction values (stick - slip vs chattering).

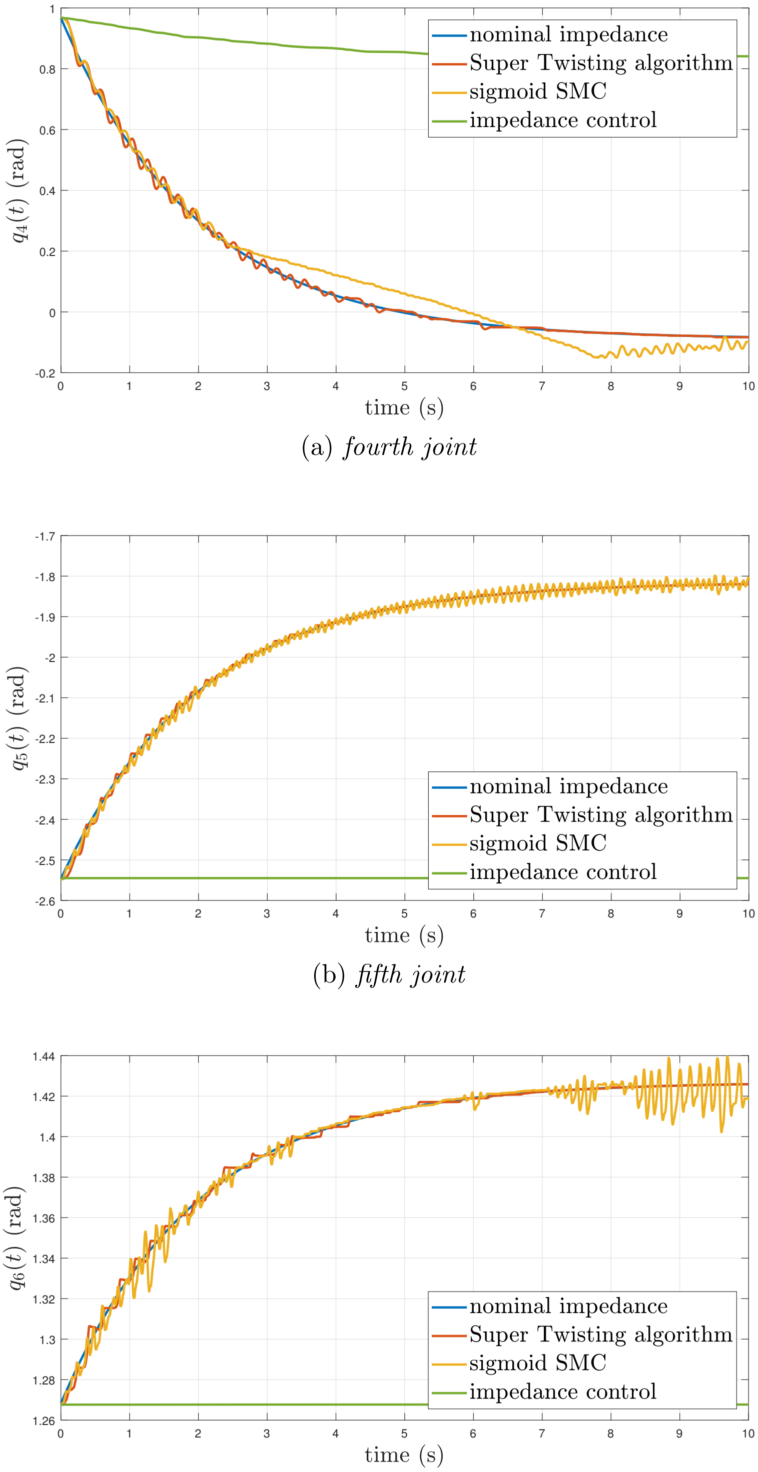


(a) *first joint*

(b) *second joint*

(c) *third joint*

Figure 6.9: 1^{st}, 2^{nd}, 3^{rd} joint positions for the step reference impedance tracking experiment



(a) *fourth joint*

(b) *fifth joint*

(c) *sixth joint*

Figure 6.10: 4^{th}, 5^{th}, 6^{th} joint positions for the step reference impedance tracking experiment

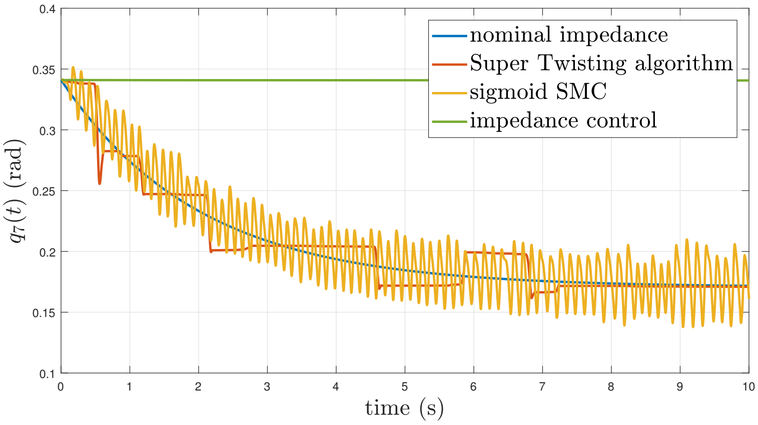
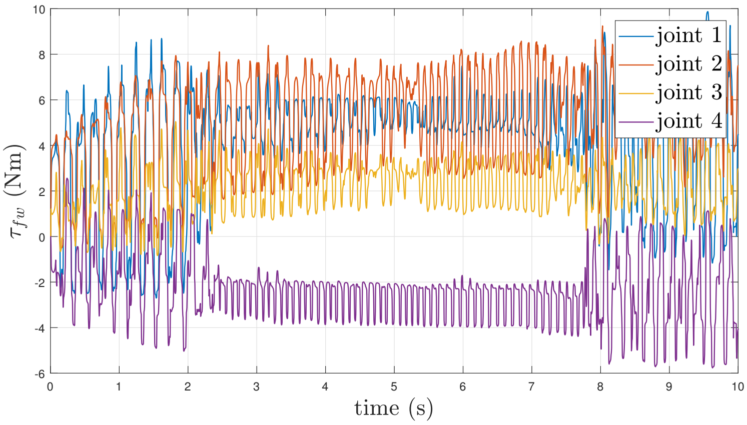
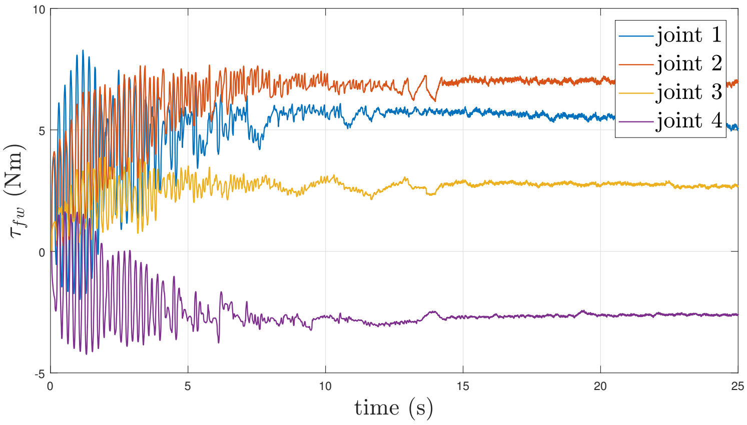


Figure 6.11: Joint position of the 7^{th} joint for the step reference impedance tracking experiment

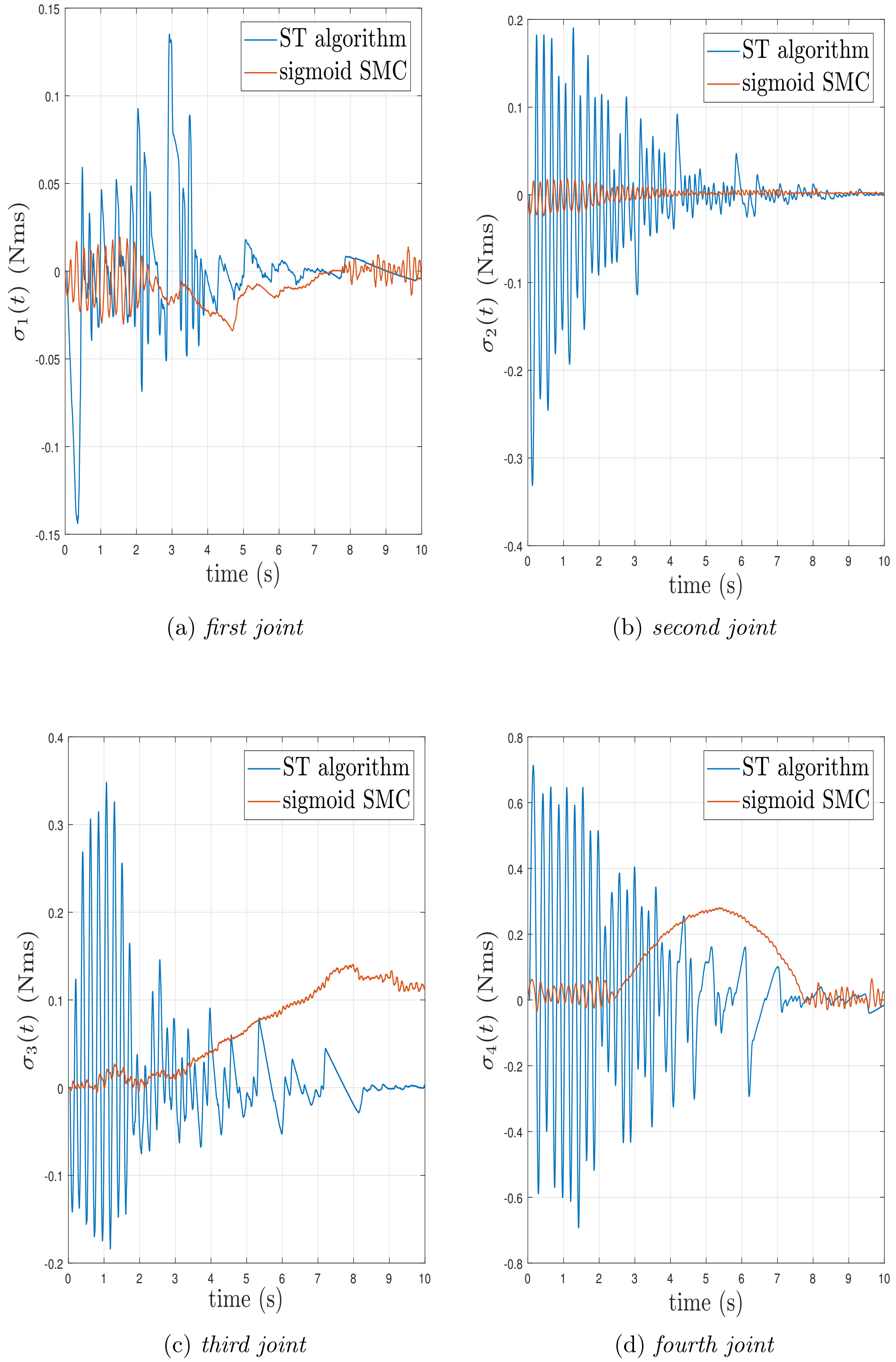


(a) *sigmoid SMC*



(b) *Super Twisting Algorithm*

Figure 6.12: Feed - forward torque applied to the first four joints by (a) sigmoid SMC controller (b) Super Twisting Algorithm controller



(a) *first joint*

(b) *second joint*

(c) *third joint*

(d) *fourth joint*

Figure 6.13: Sliding surfaces for the first four joints of the step reference impedance tracking experiment

joint n.	K_d (s^{-1})	K_p (s^{-2})	K_{smc} ($\mathrm{rad\,s}^{-2}$)
1	8	16	2
2	12	36	3
3	16	64	5
4	16	64	5
5	70	5	20
6	65	3	20
7	70	7	20

Table 6.4: Tuning parameters chosen for the impedance controller and the sigmoid SMC approximation in the TVP trajectory experiment

6.4.2 Interpolated TVP reference

By looking at Figure 6.12, it is possible to observe that the overall control torque required for the step response is very oscillating, also when adopting chattering prevention algorithm. This vibration, that is due to the limited bandwidth of the actuation channel, leads to gear wear as well as motors noise.

This effect is partially associated also to the performed test (step response), which causes a big initial error on the joint variables and consequently generates high amplitude torque commands. Moreover, this kind of input cannot happen when the teleoperation system is connected. Indeed, in this case the slave controller receives reference signals from the master manipulator at each sampling time, generating a "smooth" trajectory (this obviously depends on the operator's reactivity).

According to these considerations, a new set of experiments have been performed, where the new reference input trajectory is an interpolated TVP signal. Since error is updated every sample, impedance parameters associated to the equivalent control "lose" their error dynamics meaning, becoming PD tuning coefficients:

$$\boldsymbol{v^{eq}} = \ddot{\boldsymbol{q}}_{\boldsymbol{ref}} - \bar{\boldsymbol{B}}^{-1}\bar{\boldsymbol{C}}\dot{\tilde{\boldsymbol{q}}} - \bar{\boldsymbol{B}}^{-1}\bar{\boldsymbol{g}}\tilde{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_{\boldsymbol{ref}} - \boldsymbol{K}_d\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{K}_p\tilde{\boldsymbol{q}} \tag{6.15}$$

The chosen values for $\boldsymbol{K_p}$ and $\boldsymbol{K_d}$, as well as the SMC gain $\boldsymbol{K_{smc}}$ are reported in Table 6.4. For the sigmoid approximation, value of δ equal to 0.5 has been chosen. As usual, ISMC has been implemented to enforce the sliding mode at the beginning of the experiment.

joint n.	impedance (rad)	sigmoid SMC (rad)
1	0.0307	0.002 157
2	0.0301	0.000 510
3	0.0884	0.002 422
4	0.0331	0.000 364
5	0.1163	0.002 511
6	0.2274	0.002 344
7	0.2328	0.001 351

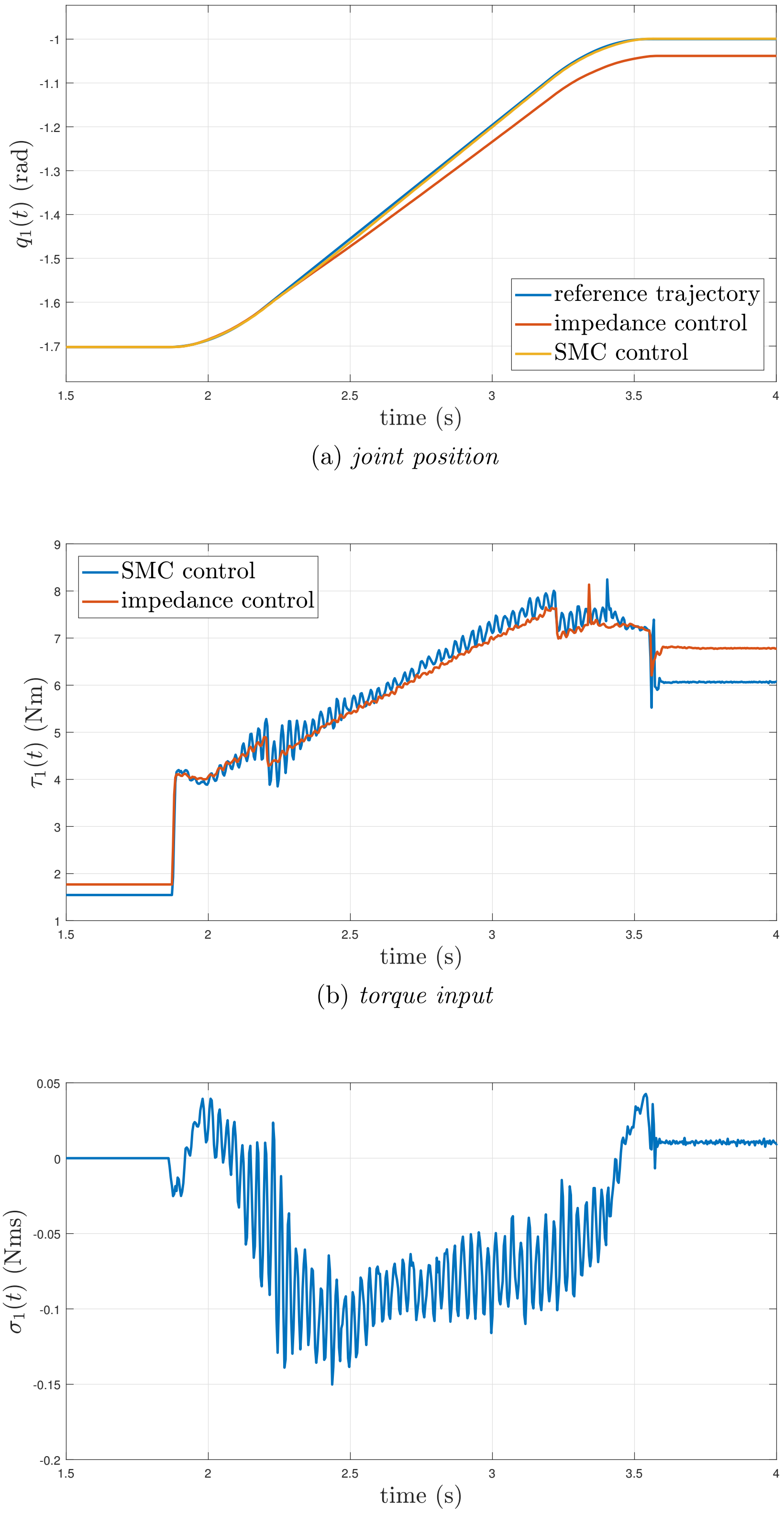
Table 6.5: RMS error associated to impedance and sigmoid SMC control for TVP

From Figure 6.14 to 6.20, the tracking performance of the proposed controller, fed with TVP reference signal, is evaluated. For each joint, the angular position response of the impedance controller is reported, together with the SMC one. Although less pronounced than the previous subsection, also in this case the improvement of sliding mode controllers with respect to impedance ones is significant.

As we would expect, now torque profiles are *smoother* than the ones generated by step reference, without strong oscillations associated to high values of K_{smc}. This has become possible thanks to the additional speed and acceleration feed - forward terms ($\dot{\boldsymbol{q}}_{\boldsymbol{ref}}$, $\ddot{\boldsymbol{q}}_{\boldsymbol{ref}}$) generated by the input trajectory, that allowed more aggressive impedance tuning (low disturbance rejection required). With a little increment in the control torque contribution, tracking accuracy is greatly improved.

The last plots represent the joints sliding variables during the response transient. Due to the approximation of the signum function, sliding mode is not perfectly enforced on the system. Moreover, steady - state deviation from the sliding surface can occur, due to the presence of high stiction values. By looking at Figures 6.18 and 6.19, it is possible to notice higher deviations of wrist joints sliding variables with respect to the arm joints; this effect confirms the increasing uncertainty that SMC needs to compensate when joints are close to the end effector (friction contribute becomes dominant in the model equations).

Ultimately, the evaluation of the RMS error between the standard impedance control and the sigmoid SMC is done in Table 6.5. Their values are significantly smaller than the step reference case. Moreover, tracking accuracy of SMC is quite high (tenths of degree).

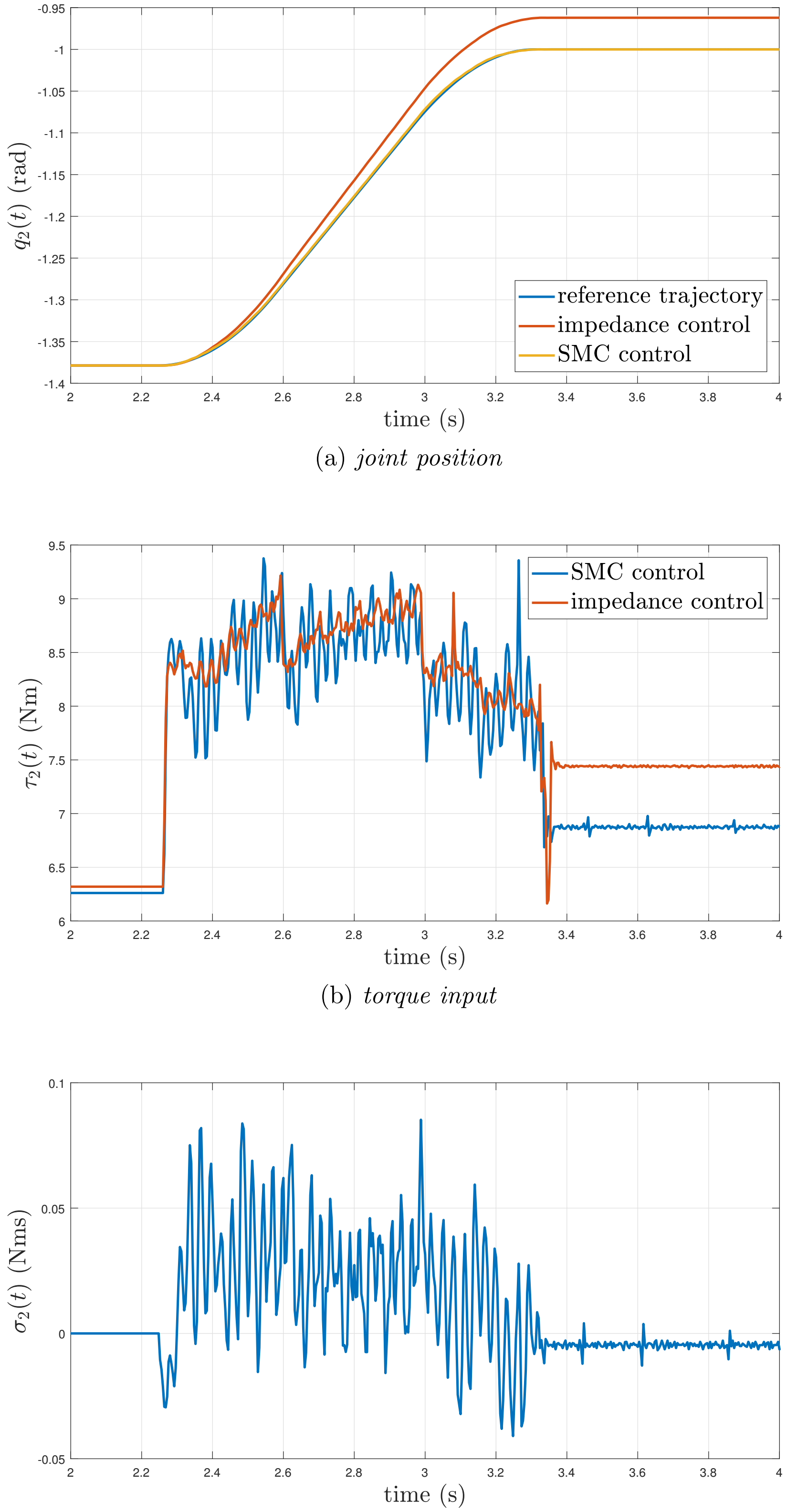


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.14: Impedance tracking with TVP profile for the first joint

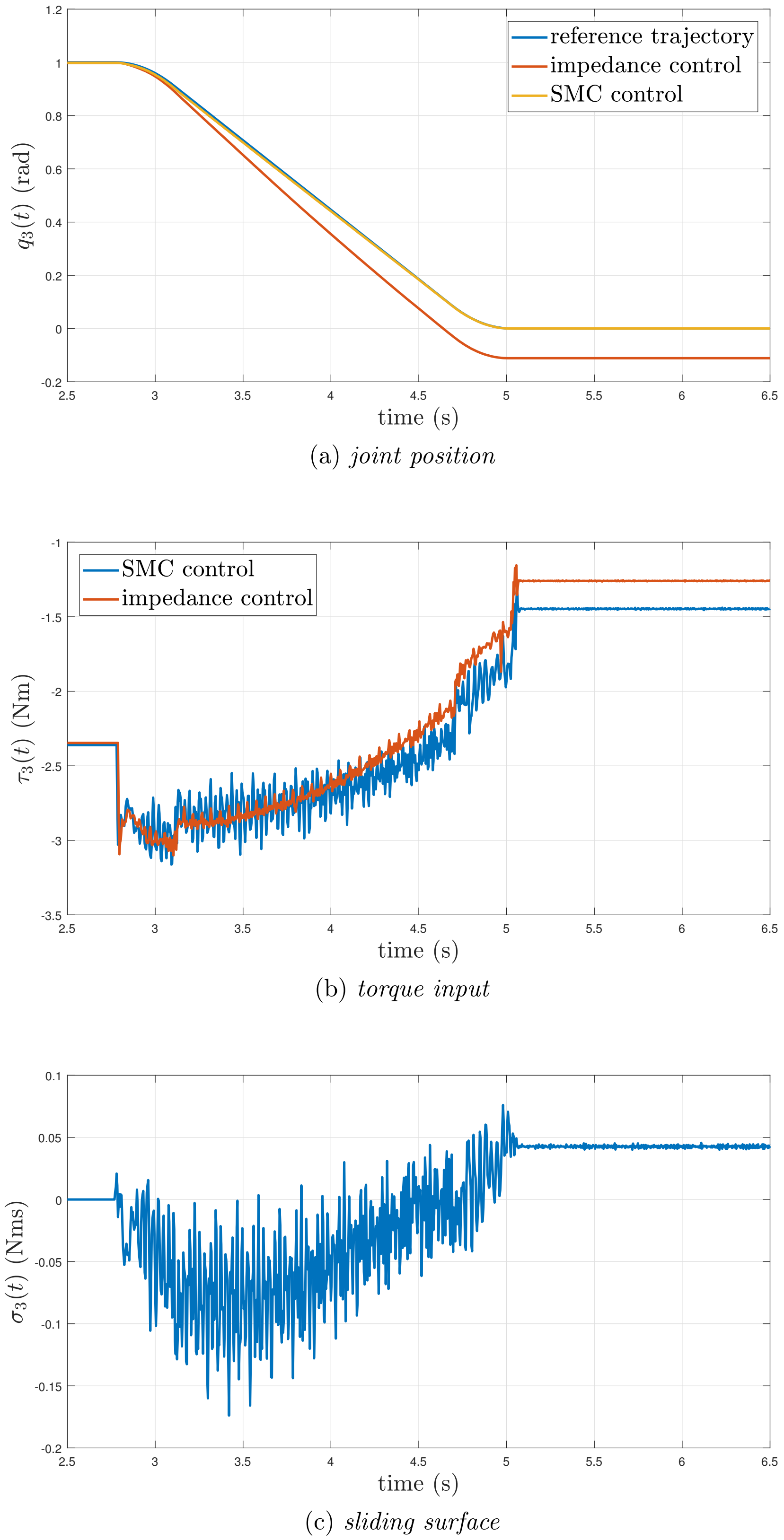


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.15: Impedance tracking with TVP profile for the second joint

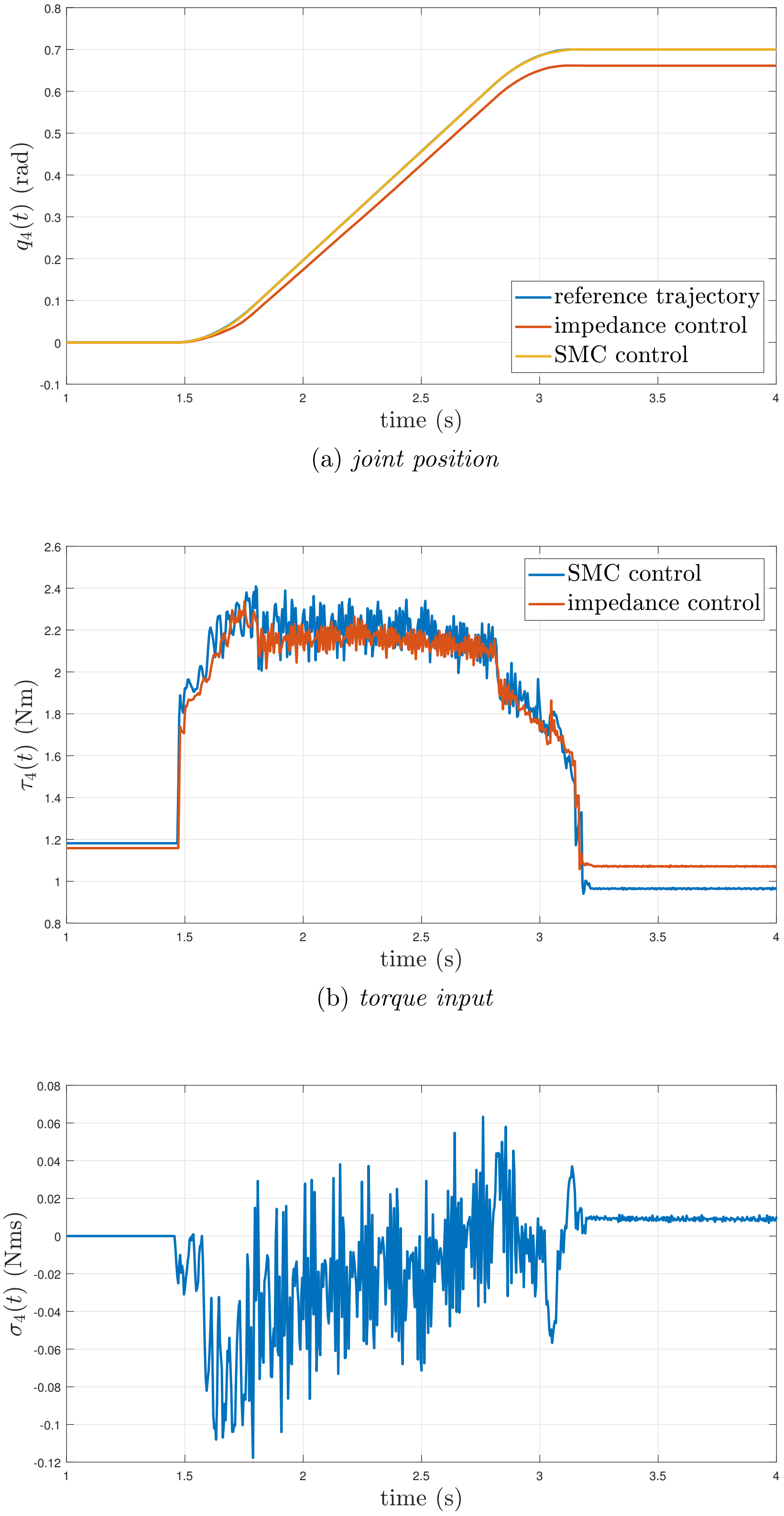


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.16: Impedance tracking with TVP profile for the third joint

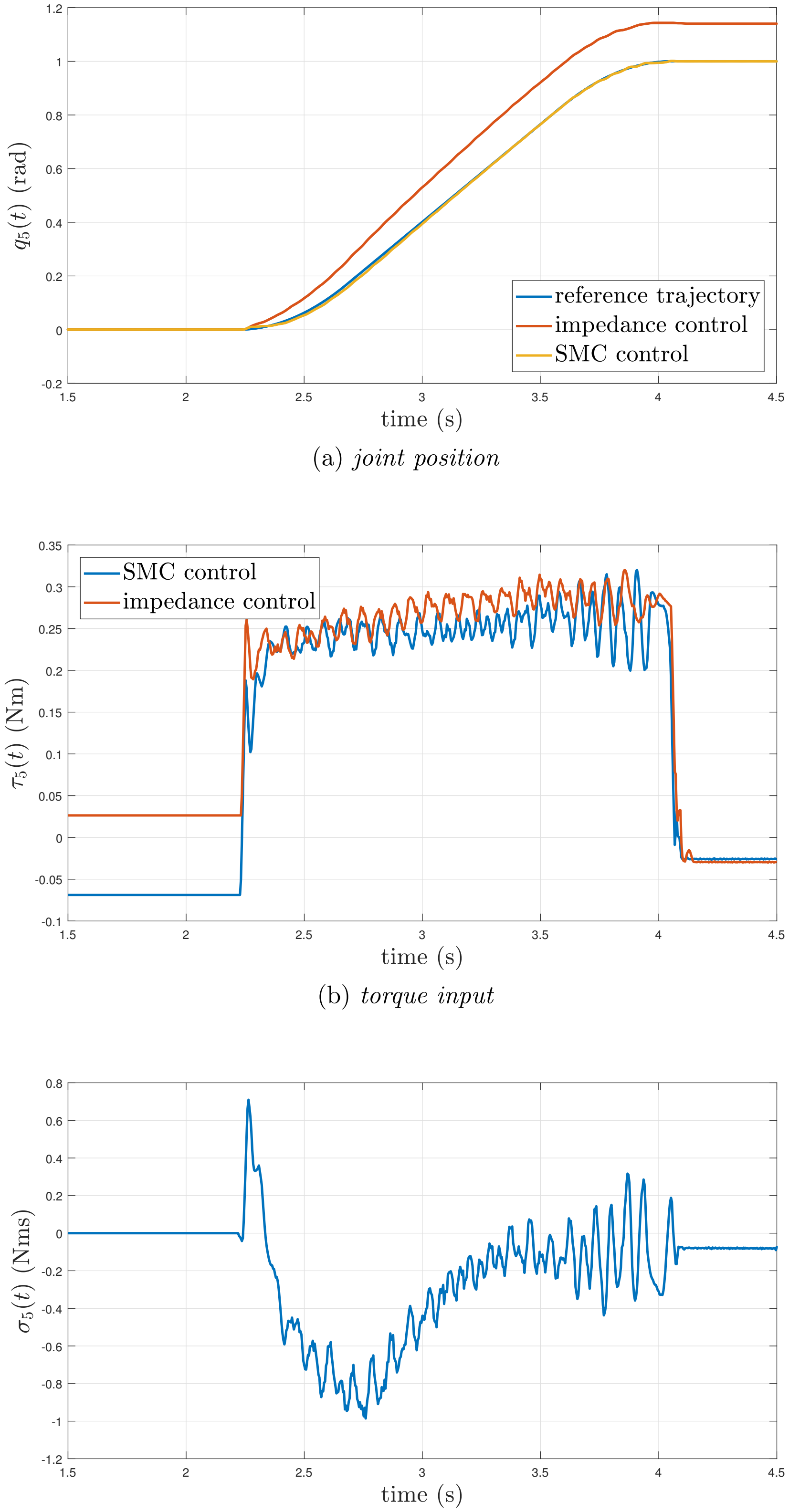


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.17: Impedance tracking with TVP profile for the fourth joint

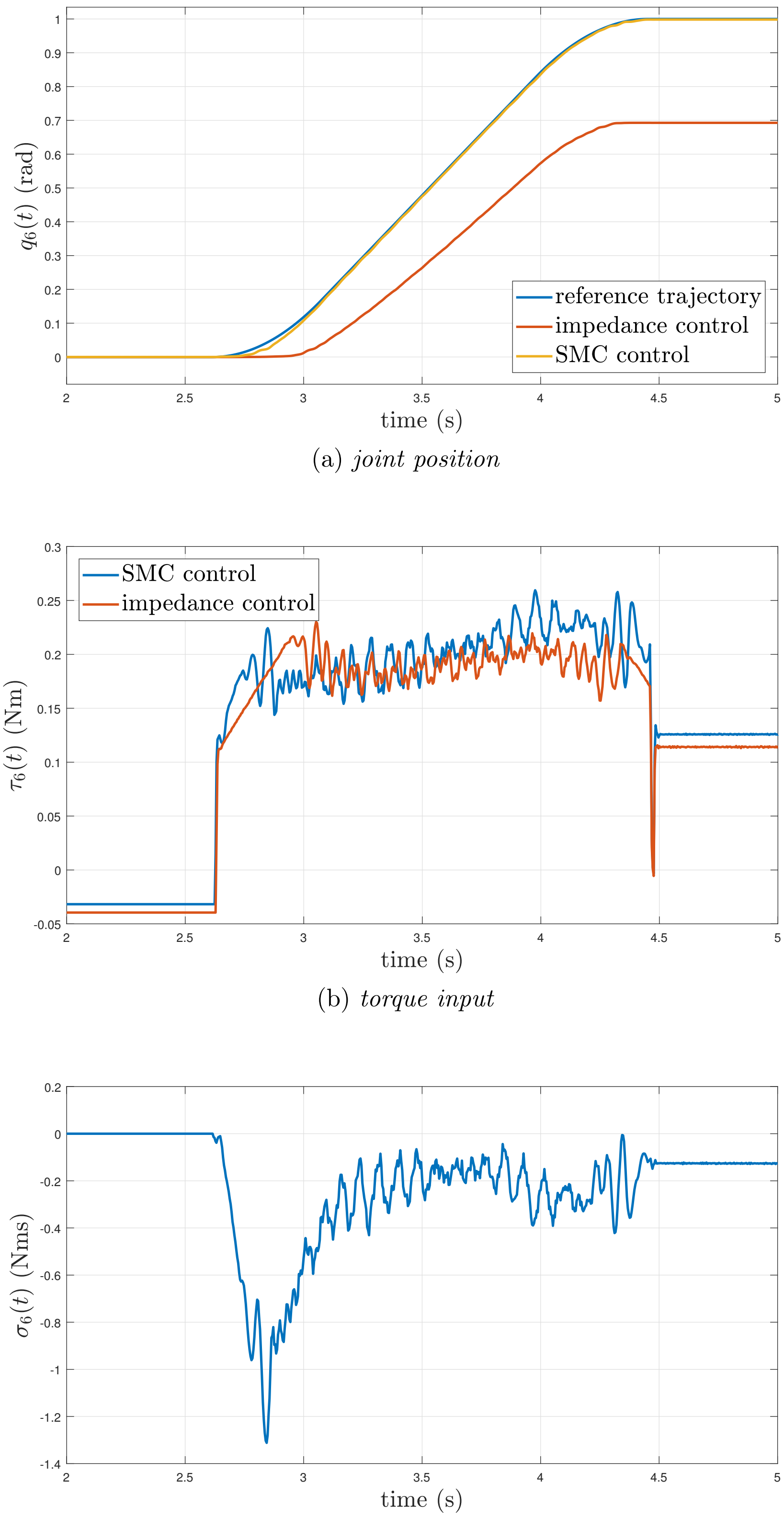


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.18: Impedance tracking with TVP profile for the fifth joint

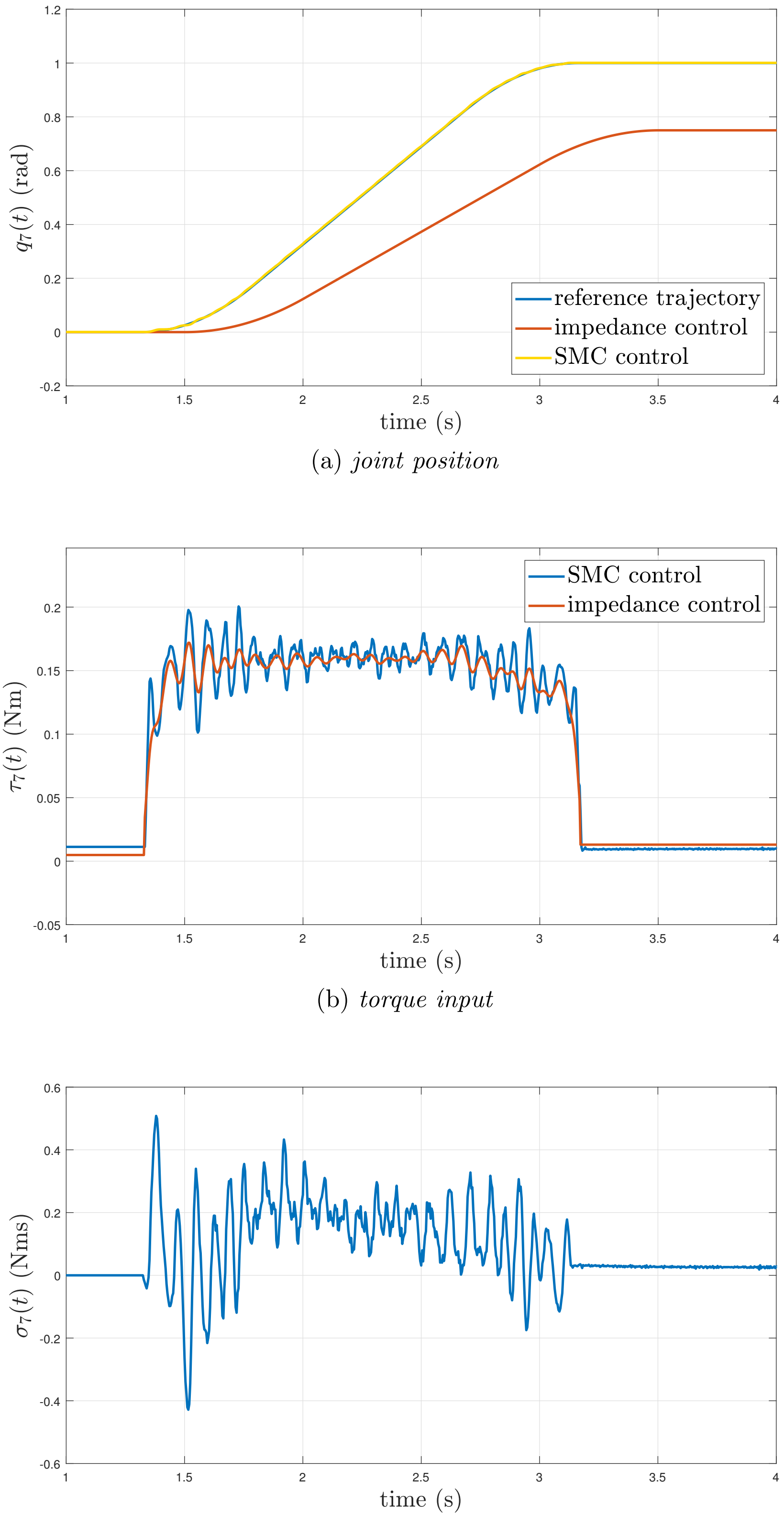


(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.19: Impedance tracking with TVP profile for the sixth joint



(a) *joint position*

(b) *torque input*

(c) *sliding surface*

Figure 6.20: Impedance tracking with TVP profile for the seventh joint

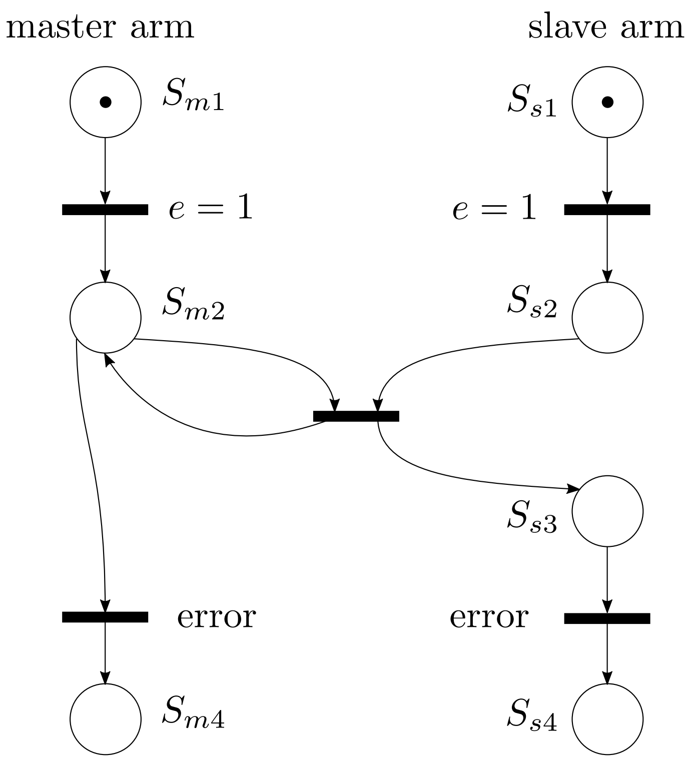


Figure 6.21: Conceptual interaction between the master and slave manipulators in order to perform the teleoperation tracking task

6.5 Teleoperation experiment

Now that a reasonable slave controller has been implemented, it is possible to perform teleoperation tasks by using the master arm as a reference generator. Since this device will be moved freely by the human, it should be controlled in order to compensate the main dynamics effects. That is why, for this experiment, inverse dynamics has been performed also on the master device. For the time being, no haptic feedback has been returned to the operator.

The conceptual operative scheme of the two manipulators has been defined by the Petri Net (PN) structure presented in Figure 6.21. The initial state of the two arms S_{m1} and S_{s1} are the systems rest position, when the robot is deactivated. When event e occurs, the two transitions associated to it are enabled and they can fire, leading the master and slave devices to a state where they share the same coordinates (according to the specular principle in section 6.1). Once both have reached such position (synchronization), effective teleoperation can start. The slave system goes in *tracking mode* (S_{s3}), while master system remains in S_{m2}, waiting to interact with the operator. Ultimately, if an *error* during the teleoperation task is detected (out of joint bounds, too high speeds, etc.), both systems switch into a *failure mode*, where they remain until the operator manually resets the experiment.

Tracking performance for each joint is reported in Figures 6.22 and 6.23. Reference signal is given by the master device angular positions, while the slave is controlled adopting the strategies presented in subsection 6.4.2. The tracking error between these two signals is shown in Figures 6.24 and 6.25. Bilateral motion control activation starts at 32 s (dashed line). Any mismatch between the master and slave positions before that time are due to poor tracking accuracy in the internal `ABB` controllers (as in Figure 6.22c).

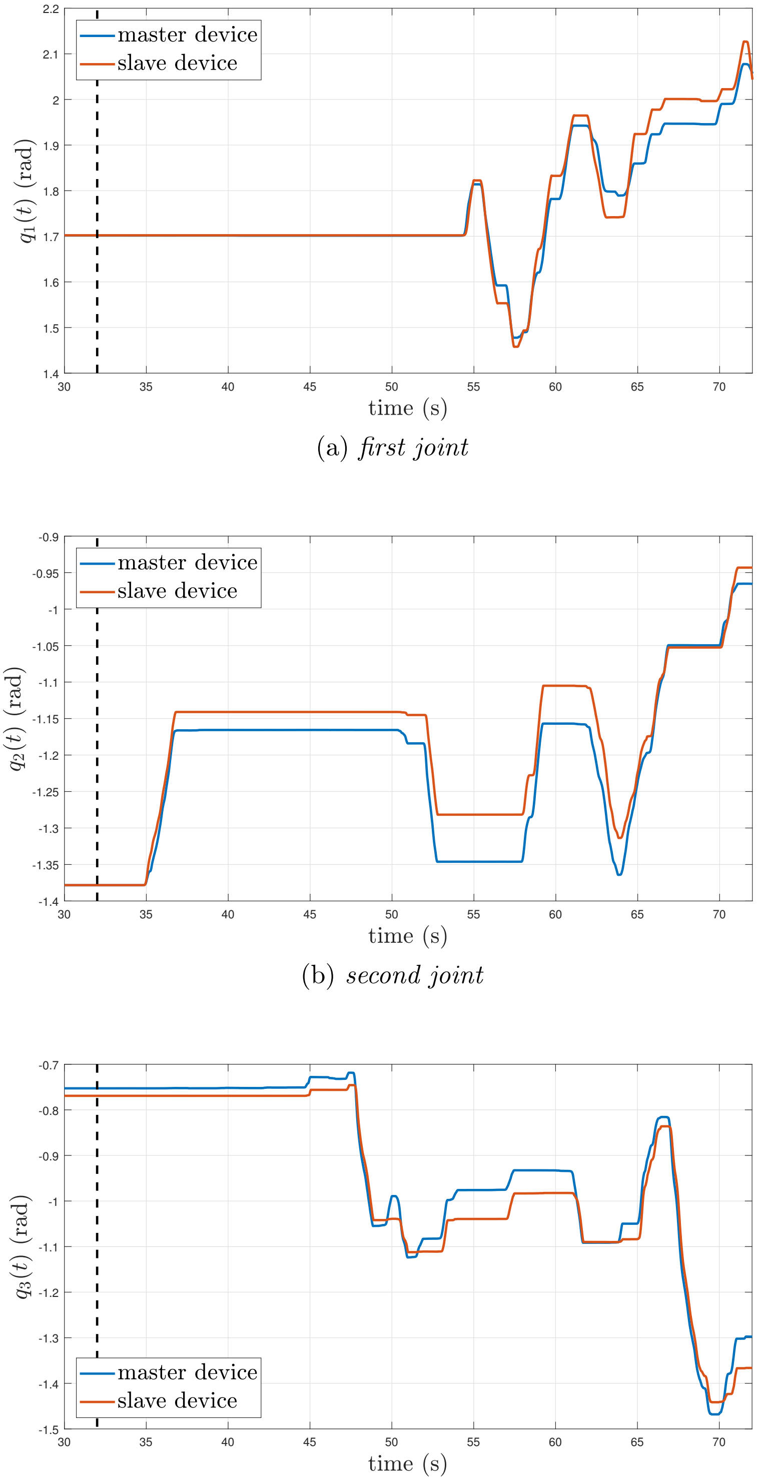
The first noticeable fact is that the error fluctuation (in particular for the first three joints) is much greater than the average error analysed in subsection 6.4.2. This is due to several factors:

- Big joints are the ones which suffer mostly from **joint ripples**, therefore their control action cannot be too aggressive (K_{smc} should be set lower than the required one). If more tracking accuracy is needed, limited actuator bandwidth causes oscillations in $\boldsymbol{\tau}_{\boldsymbol{fw}}$ which ultimately leads to vibration and noise;

- By looking closer at the time scale, teleoperated movements generates **slower reference signal** with respect to the TVP experiment. Thus, static friction is very relevant and open - loop compensation is not good enough for this low dynamic range. Ultimately, this causes loss of tracking accuracy;

- If joint gains are set to high values to cope with low speed uncertainty, **unstable behaviour** occurs if the operator moves the master very fast, due to high torque oscillations;

For all these reasons, a bit of accuracy has been sacrificed in order to obtain a smooth teleoperation device independently from operator capabilities. Moreover, this set - up grants a safer interaction with the external environment, allowing a certain tracking error tolerance without requiring high torques to compensate for it.

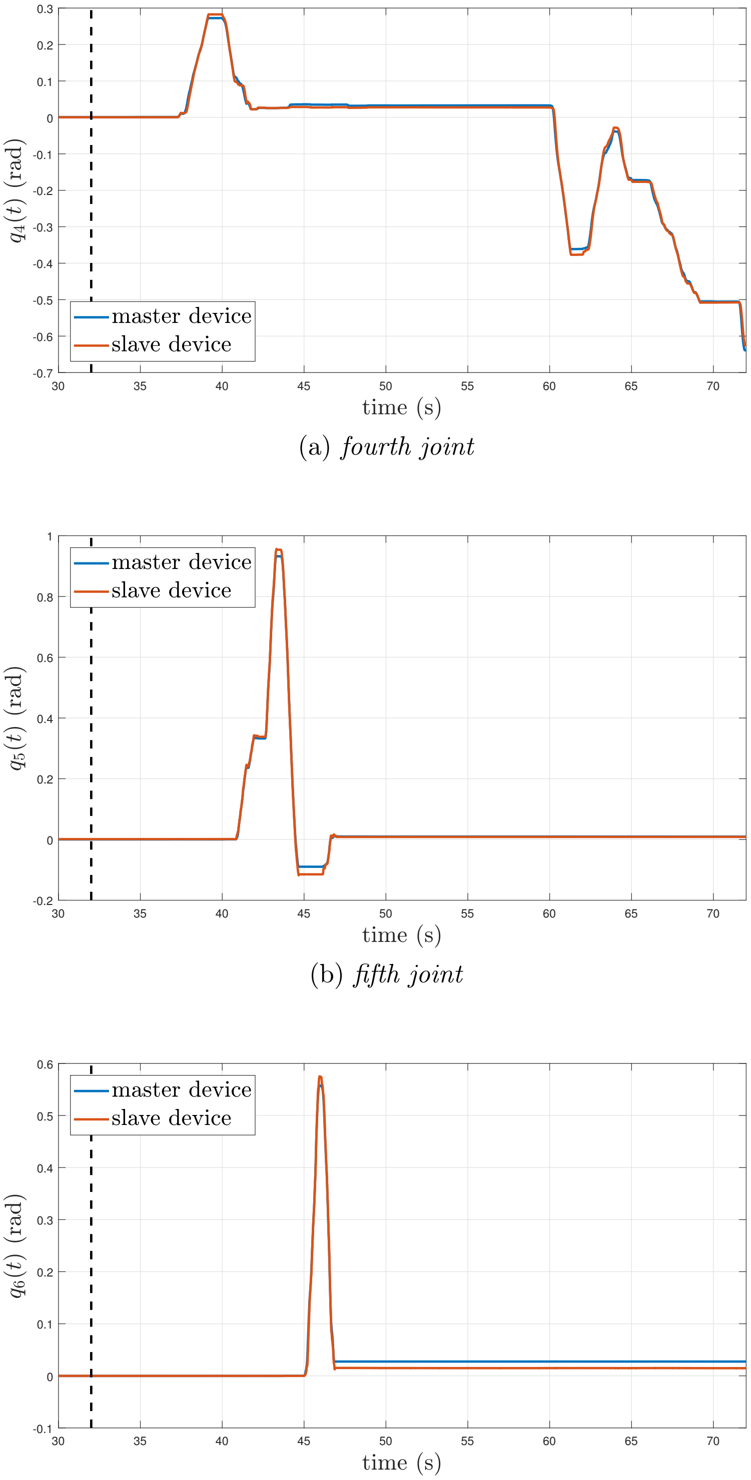


(a) *first joint*

(b) *second joint*

(c) *third joint*

Figure 6.22: Bilateral motion, joint positions: first, second, third

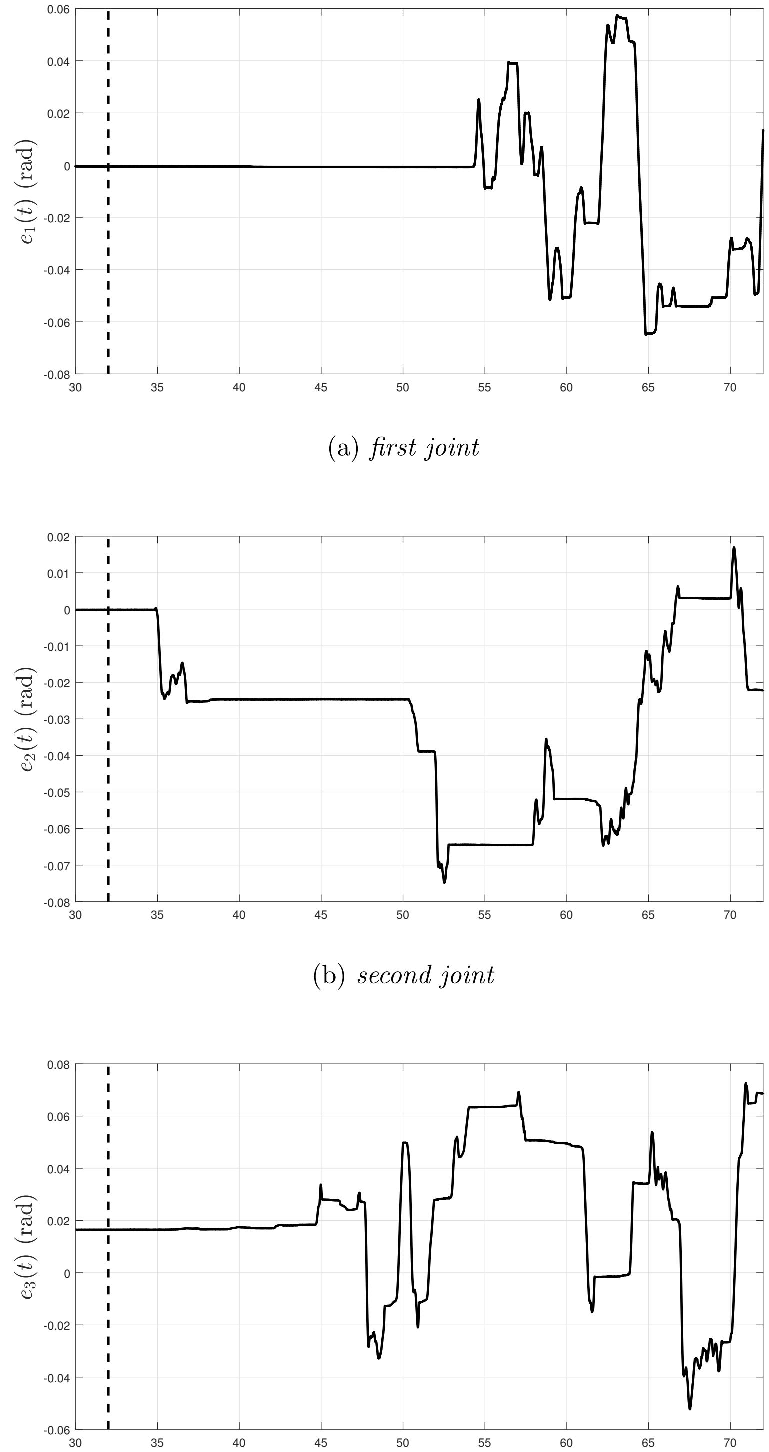


(a) *fourth joint*

(b) *fifth joint*

(c) *sixth joint*

Figure 6.23: Bilateral motion, joint positions: fourth, fifth, sixth

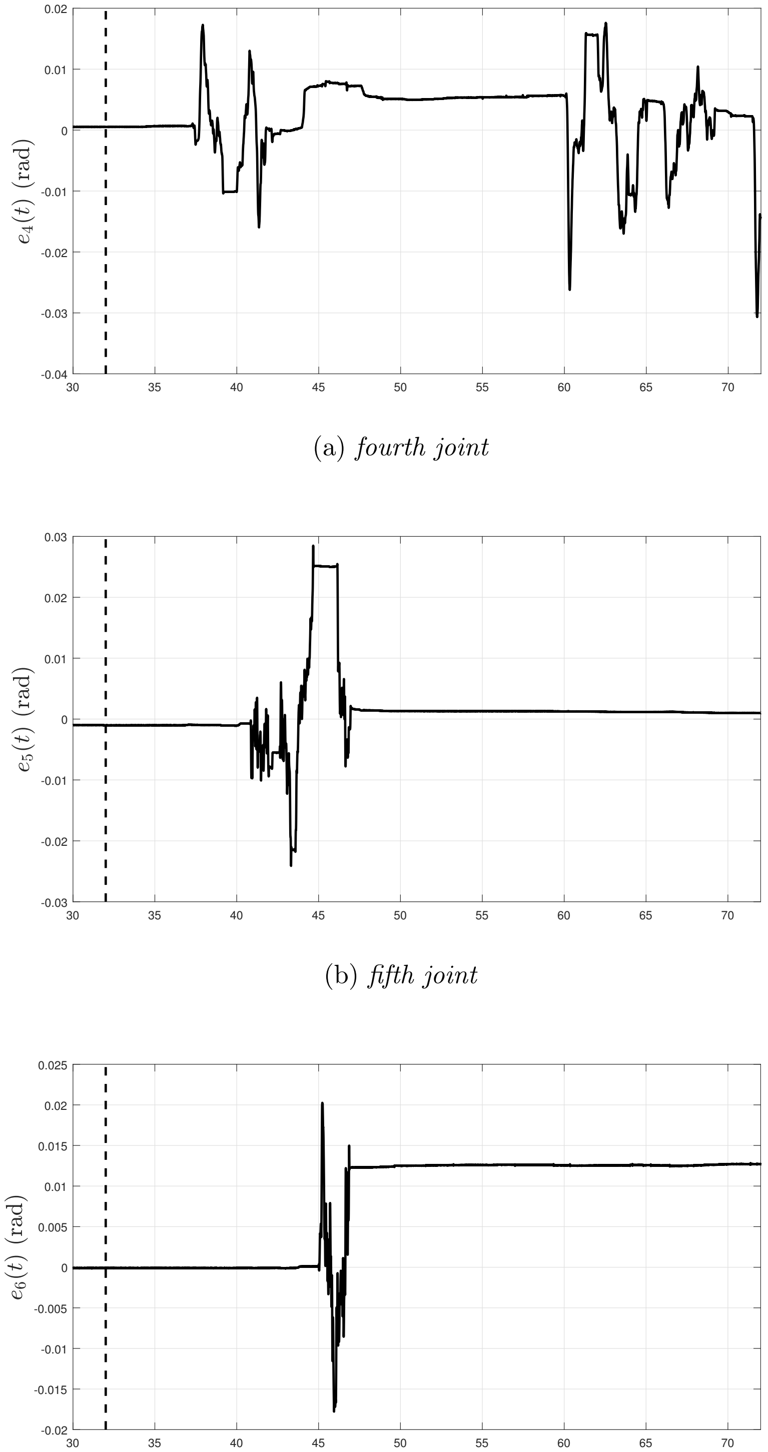


(a) *first joint*

(b) *second joint*

(c) *third joint*

Figure 6.24: Bilateral motion, tracking error: first, second, third



(a) *fourth joint*

(b) *fifth joint*

(c) *sixth joint*

Figure 6.25: Bilateral motion, tracking error: fourth, fifth, sixth

Chapter 7

Conclusions

In this thesis, a comprehensive overview of bilateral teleoperation nonlinear control strategies has been proposed. The theory validated through simulation has been developed in order to guarantee stable interaction between the slave device and the external environment, while performing teleoperation tasks. This system allows to improve the safety of the remote surroundings also if the operator has not perfect knowledge of the slave set - up. This is typically the case when only vision sensors are adopted to help the human to perform the remote task (blind sides). Through haptic feedback, the operator perception of the obstacles faced by the slave is enhanced. Thanks to this additional information, he/she will be able to take the correct decision without damaging the slave environment.

Due to the large scope of this thesis, not all the aspects have been completely dealt in this dissertation. In particular, experimental application has been limited due to the huge differences between the simplified simulated model and the real robotic manipulator. As it can be seen in chapter 6, only a simplified control scheme over the complex theoretical analysis has been implemented. This has been motivated by the great difficulties encountered in controlling the joint motion through the ABB feed - forward torque channel, disabling the internal manipulator's loop. This choice has introduced in the proposed schemes a lot of uncertainties with respect to the parametric mismatch assumed in simulation, requiring a strong simplification of the theoretical aspects in order to obtain reasonable results.

Nevertheless, it has been shown that SMC can be effectively applied to servomechanisms, improving significantly the performance of a linear controller. This represents a valid alternative with respect to the standard implementation of impedance controllers as a reference to the internal control loops. Furthermore, all the theory developed for the 1 DOF case can be extended to each operational space coordinates thanks to the results achieved in chapter 5.

However, these considerations do not invalidate the theoretical assets presented in chapters 3 and 4, which are valid for a wide variety of teleoperation schemes. Their results prove that it is possible to design a bilateral teleoperation system that maximize transparency while retaining stability properties.

Moreover, the capabilities of the decentralised SMC scheme to guarantee a stable free motion teleoperation over a significant temporal window has been tested experimentally. This can be seen as a starting point for a more detailed analysis of nonlinear control strategies applied to complex teleoperation systems, which nowadays are not very widespread in the industrial field.

7.1 Future developments

The experimental analysis performed on the robotic arm has been proved to be feasible but limited. In particular, the n DOFs extension properties have not been fully exploited. Therefore, future research steps can be the following:

- Implementation of the operational space control strategy on the real robotic arm and assessment of its robustness property. This should be done in order to extend the global stability results (analysed for the 1 DOF case) to each coordinate of the operational space. This is also fundamental in order to help the operator understand the external environment faced by the slave manipulator;

- Inclusion of the passivity observer presented in chapter 4 to monitor the energy flows between the master and slave robotic arms. During the experiments, it has been noticed a strong dependency between system oscillations and imposed impedance relationship. Consequently, desired dynamics have been forced to appropriate values of the parameters. If it is necessary to alter those values to guarantee passivity, SMC could not be able to track the desired impedance any more due to the excessive uncertainties;

- To reduce the pressure on SMC and allow impedance shaping, friction estimation could be done in real time and adaptively, instead of the static open - loop identification proposed in this thesis. If on - line adaptation is introduced, care must be taken in the computation of the external forces through the residual estimator, since they will be seen by the friction observer as friction contributes and it will try to cancel them out;

- Another interesting variation to the proposed algorithm could be represented by introducing adaptation also to the sliding gain K_{smc}. This consideration comes from the fact that the static friction uncertainty is much greater than the dynamic one. Since K_{smc} should be tuned according to the maximum level of disturbance acting on the system, often its value is overestimated, causing unneeded oscillations. In these situations it would be better to allow a large K_{smc} at low speed and then reduce its value according to the lower dynamic uncertainty;

- Stability of the environmental interaction has not been assessed experimentally, therefore in future studies it will be necessary to introduce external forces at the slave end effector and properly map them to the master device, in order to verify the theoretical results that have been exploited in chapter 4.

Appendix A

Differentiation procedures

In the numerical analysis field, great attention has been devoted to robust and computationally efficient ways to retrieve derivatives of a given function $f(t)$. Since $\dot{f}(t)$ is related only to the function local properties, such methods are easily implementable in their *on-line* versions, which are capable to compute real-time derivatives of an incoming (maybe noisy) signal.

A.1 Introduction

Let's start by assuming that the input signal to our differentiator is derivable at time t. More precisely, $\dot{f}(t)$ has to satisfy the *Lipschitz condition* with a value of $L > 0$

Definition A.1. (Lipschitz condition) A function $\dot{f} : [a, b] \to \mathbb{R}$ is said to satisfy the Lipschitz condition if there exists a constant $L > 0$ such that

$$|\dot{f}(t) - \dot{f}(t')| \leq L|t - t'| \qquad \forall t, t' \in [a, b] \tag{A.1}$$

The smallest constant L satisfying equation (A.1) is called *Lipschitz constant.*

If $f(t)$ is regular enough, Lipschitz condition holds for values of L greater than a given $\bar{L}$, which can be computed a priori if the signal bandwidth is known. If no information is available on $f(t)$, it is possible to choose $\bar{L}$ based on the Nyquist frequency at which the signal has been sampled. Doing so, we are sure that definition A.1 will hold $\forall t \geq t_0$ initial time.

Once differentiability of $f(t)$ is guaranteed, it's possible to apply a variety of *finite difference methods*, as proposed in [62], which can be rewritten as first order linear filters using Laplace transform. This way of proceeding coincides with the classical numerical differentiation approach and it leads to a series of problems. Such issues may particularly arise when the system has high frequency information or it is affected by noise [48]:

- Significant **phase lag** at the output derivative, as well as an **attenuation** due to the filtering effects of the differentiator;
- **Error doesn't tend to zero**, also in the presence of vanishing noise, at any fixed time;

Thanks to these observations, we can state that there exists a trade-off between noise (high frequency filters) and time delay (low frequency filters) affecting the output derivative [8]. This fact motivates the research of *exact* and *robust* differentiators, which can guarantee the convergence of $\dot{f}(t)$ with finite transient time.

A.2 Levant differentiator

Consider the problem of tracking a function $f(t)$, using as control variable its derivative $\dot{f}(t)$. This objective can be formalized by considering the following first order state equation

$$\dot{x} = u \tag{A.2}$$

where x represents our estimate of function $f(t)$. Let's define our *sliding surface* S as the tracking error between the function and its estimation. The idea is to find a control variable u which is capable to impose on system (A.2) (after a finite time transient process) the following **two-sliding modes**:

$$S = x - f(t) = 0 \qquad \dot{S} = \dot{x} - \dot{f}(t) = u - \dot{f}(t) = 0 \tag{A.3}$$

using only the on-line measure of function $f(t)$ ($\dot{f}(t)$ is considered to be an *unmeasurable quantity*).

If we look closely at the problem formulation, we can notice that the differentiation task is analogous to the standard second order sliding mode control one. Therefore, it makes sense to apply second order control techniques to obtain the desired estimate convergence. A possibility is to use the *modified 2-sliding algorithm* proposed by Levant and adapted for estimation processes [22]:

$$\begin{aligned} \dot{z}_0 &= -\lambda_0 L^{1/2}|z_0 - f(t)|^{1/2} sign(z_0 - f(t)) + z_1 \\ \dot{z}_1 &= -\lambda_1 L sign(z_0 - f(t)) \end{aligned} \tag{A.4}$$

If we desire to estimate only the first derivative of $f(t)$, it's possible to apply a very crude approximation to the general convergence criterion theorem [48] and derive a *sufficient condition* for the asymptotic stability of the previous non-linear dynamic system. If parameters λ_0, λ_1 and L are chosen such that the following inequalities hold

$$\lambda_1 > 1 \quad \wedge \quad \lambda_0^2 \geq 4\frac{\lambda_1 + 1}{\lambda_1 - 1} \tag{A.5}$$

the estimator state $\mathbf{z} = [z_0, z_1]^T$ tends in finite time to the input function $f(t)$ and its derivative $\dot{f}(t)$. A complete stability analysis of Levant differentiator lies outside to the aim of this work. If the reader is interested in a more detailed analysis of observer convergence, he/she can refer to [48, 22]. Let's just say that in more complex cases it is not possible to analytically find stability bounds for the Levant differentiator and it's necessary to rely on computer simulation.

A.3 Higher order differentiators

The Levant differentiator structure can become *recursive*. This means that it is possible to impose higher order sliding manifolds simply by enlarging the observer state up to the **n-th derivative** of the input function $f(t)$ ($f^{(n)}(t)$). This feature is noteworthy, because by using a standard linear differentiation, each step adds approximations and we cannot iterate our estimation for more than two - three times. Instead, Levant n-th order differentiator *inherits* the property of finite time convergence, guaranteeing vanishing error for the maximum estimated derivative.

The generic expression of the higher order Levant differentiator is:

$$\begin{aligned}
\dot{z}_0 &= \nu_0, & \nu_0 &= -\lambda_0 L^{\frac{1}{n+1}} |z_0 - f(t)|^{\frac{n}{n+1}} sign(z_0 - f(t)) + z_1 \\
\dot{z}_1 &= \nu_1, & \nu_1 &= -\lambda_1 L^{\frac{1}{n}} |z_1 - \nu_0|^{\frac{n-1}{n}} sign(z_1 - \nu_0) + z_2 \\
&\dots & &\dots \\
\dot{z}_{n-1} &= \nu_{n-1}, & \nu_{n-1} &= -\lambda_{n-1} L^{\frac{1}{2}} |z_{n-1} - \nu_{n-2}|^{\frac{1}{2}} sign(z_{n-1} - \nu_{n-2}) + z_n \\
\dot{z}_n &= \nu_n, & \nu_n &= -\lambda_n L sign(z_n - \nu_{n-1})
\end{aligned} \tag{A.6}$$

Advantages of Levant recursive definition can be resumed through the following two points:

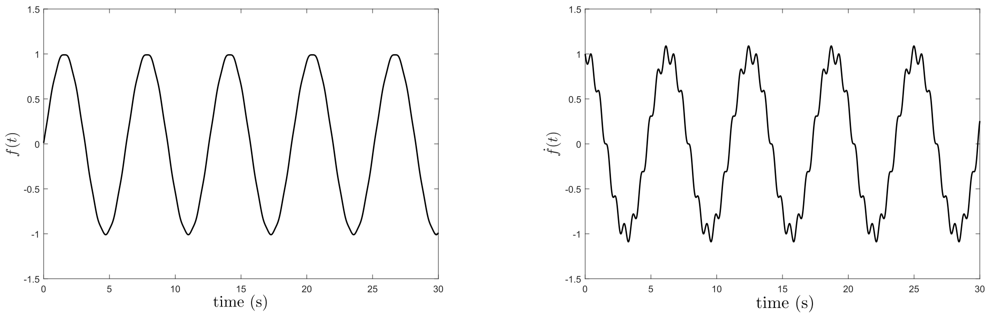
- each state can be tuned **independently** using a single parameter λ_i, without influencing other states' dynamics;
- it may be convenient to augment the differentiator's order to **improve the computation of lower degree derivatives**. This is because the last term of the non-linear state chain contains a sign function which may cause chattering in the estimated variable (signal deterioration). If another derivative is added to the differentiator, discontinuity is filtered out and estimation is smoothed;

The last statement may not be valid any more in the case of noise affecting the system, because increasing the derivative order may introduce more uncertainty in the estimation.

A.4 Algorithm testing results

In order to prove the superior performance of Levant differentiator with respect to standard linear observers, a simulation via `MATLAB - Simulink`® environment has been carried out. The chosen input function to be differentiated is

$$f(x) = \sin(t) + 0.01\cos(10t) \tag{A.7}$$



(a) $f(x)$

(b) $\dot{f}(x)$

Figure A.1: Time evolution of input signal (A.7) and its theoretical derivative (A.8)

and its *analytically exact* first derivative is equal to

$$\dot{f}(x) = \cos(t) - 0.1\sin(10t) \tag{A.8}$$

Levant differentiator is compared with discrete-time linear filters which emulate the behaviour of an ideal differentiator in a certain frequency range. They are retrieved from their dual continuous versions via Tustin discretization method. The region where a linear filter is a good approximation of the ideal differentiator will be called *differentiation bandwidth.*

- The first filter has a pole at $\omega_p = 100\,\text{rad/s}$, which is comparable to system's Nyquist frequency ($\omega_N \simeq 780\,\text{rad/s}$). Its differentiation bandwidth is $[0{,}50]\,\text{rad/s}$;
$$H_{HF}(z) = \frac{250}{3}\frac{z-1}{z-2/3} \tag{A.9}$$

- The second filter has a pole at $\omega_p = 1\,\text{rad/s}$, approximately one hundredth of system's Nyquist frequency. Its differentiation bandwidth is $[0{,}0.5]\,\text{rad/s}$;
$$H_{LF}(z) = 500\frac{z-1}{501z-499} \tag{A.10}$$

Levant differentiators are not tuned following inequalities (A.5) because they result into excessively conservative parameters choices. The chosen approach is to adapt Levant suggested values [48] verifying the generic convergence theorem validity via computer simulation. Before computing λ_i, it is necessary to give an estimation

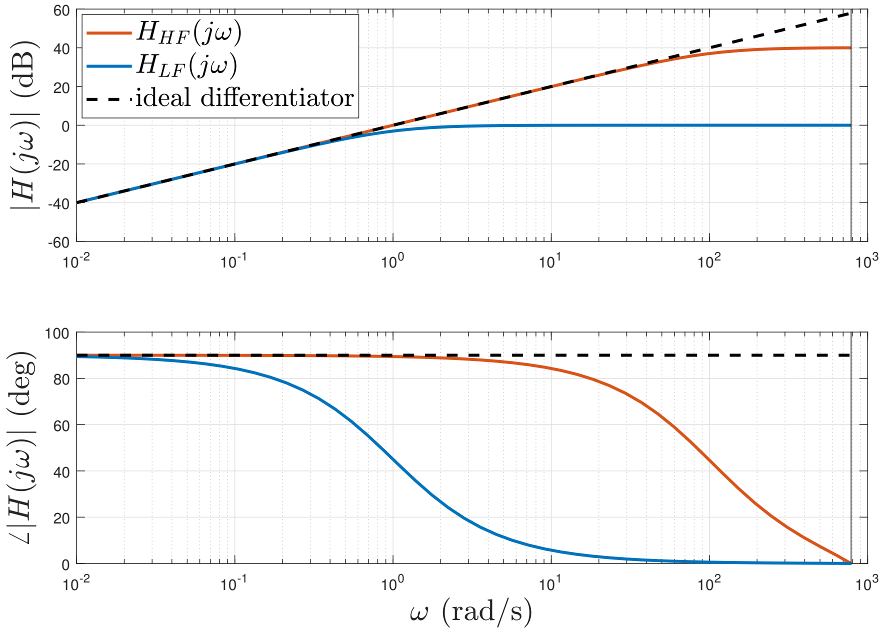


Figure A.2: Bode diagrams of $H_{HF}(z)$ and $H_{LF}(z)$ compared to the ideal differentiator frequency response

of the *Lipschitz constant*. An effective way to properly compute L which fulfils the definition A.1 consists into overestimating its value, implement the filter, check the correctness of differentiation and try to reduce it up to a reasonable limit while maintaining convergence [48]. Another method that can be employed in simulation consists in computing the second derivative of f, since L must satisfy the following inequality [8]

$$\sup_{t \geq t_0} \left| \frac{d^2 f(t)}{dt^2} \right| \leq L \tag{A.11}$$

Since $\frac{d^2 f(t)}{dt^2} = -\sin(t) - cos(10t)$ and its superior limit is equal to 2, it is sufficient to choose $L > 2$ as Lipschitz constant. For simulation comparison, two differentiators have been set up

- A first order Levant differentiator following equation (A.4).
- A second order Levant differentiator, derived from the n-th order one ((A.6))

	L	λ_0	λ_1	λ_2
1^{st} order Levant	10	1.1	1.5	
2^{nd} order Levant	10	1.1	2	5

Table A.1: Tuning parameters for the two implemented Levant differentiators

by imposing $n = 2$. Observed state is z_1;

$$\begin{aligned} \dot{z}_0 &= \nu_0, & \nu_0 &= -\lambda_0 L^{\frac{1}{3}} |z_0 - f(t)|^{\frac{2}{3}} sign(z_0 - f(t)) + z_1 \\ \dot{z}_1 &= \nu_1, & \nu_1 &= -\lambda_1 L^{\frac{1}{2}} |z_1 - \nu_0|^{\frac{1}{2}} sign(z_1 - \nu_0) + z_2 \\ \dot{z}_2 &= \nu_2, & \nu_2 &= -\lambda_2 L sign(z_2 - \nu_1) \end{aligned} \tag{A.12}$$

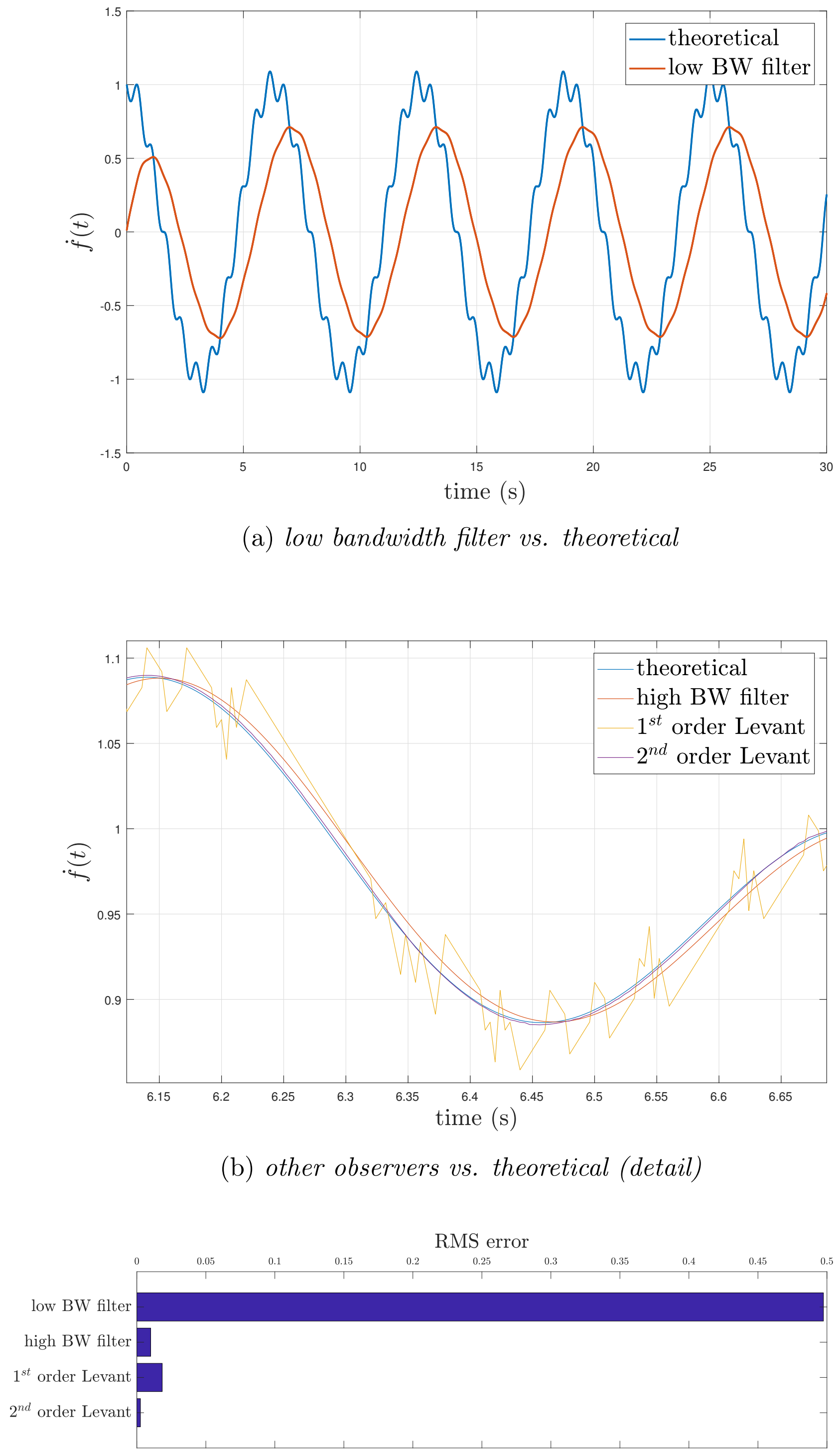
The tuning parameters are shown in table A.1.

The performed tests can be divided in the following subcategories:

1. **Differentiation of $f(t)$ without disturbances.** The input signal is given as figure A.1 and the four proposed configurations (low frequency - high frequency linear filters, 1^{st} and 2^{nd} order Levant differentiators) are compared by looking at their RMS error with respect to the ideal response;

2. **Differentiation of $f(t)$ with input disturbance.** A band-limited white noise, with flat Power Spectral Density (PSD) and noise power equal to 1×10^{-5} W, is added to the sinusoidal signal (A.7) and it is given as input to the differentiators. Same comparisons of point 1 are performed;

By looking at figures A.3 and A.4 a lot of interesting conclusions can be drawn. First of all we can clearly see the trade-off given by linear differentiation. If we decide to use a low frequency approximation of the ideal differentiator, we obtain a more stable estimation of $\dot{f}(t)$ at the price of a poor accuracy also when we have at our disposal good signal information. Opposite reasoning can be done on the high bandwidth filter, which has an accuracy degree of the same order of the Levant differentiator when input signal is clear, but it shows all its instability when noise affect measurement (even small amounts can cause totally unreliable observations).

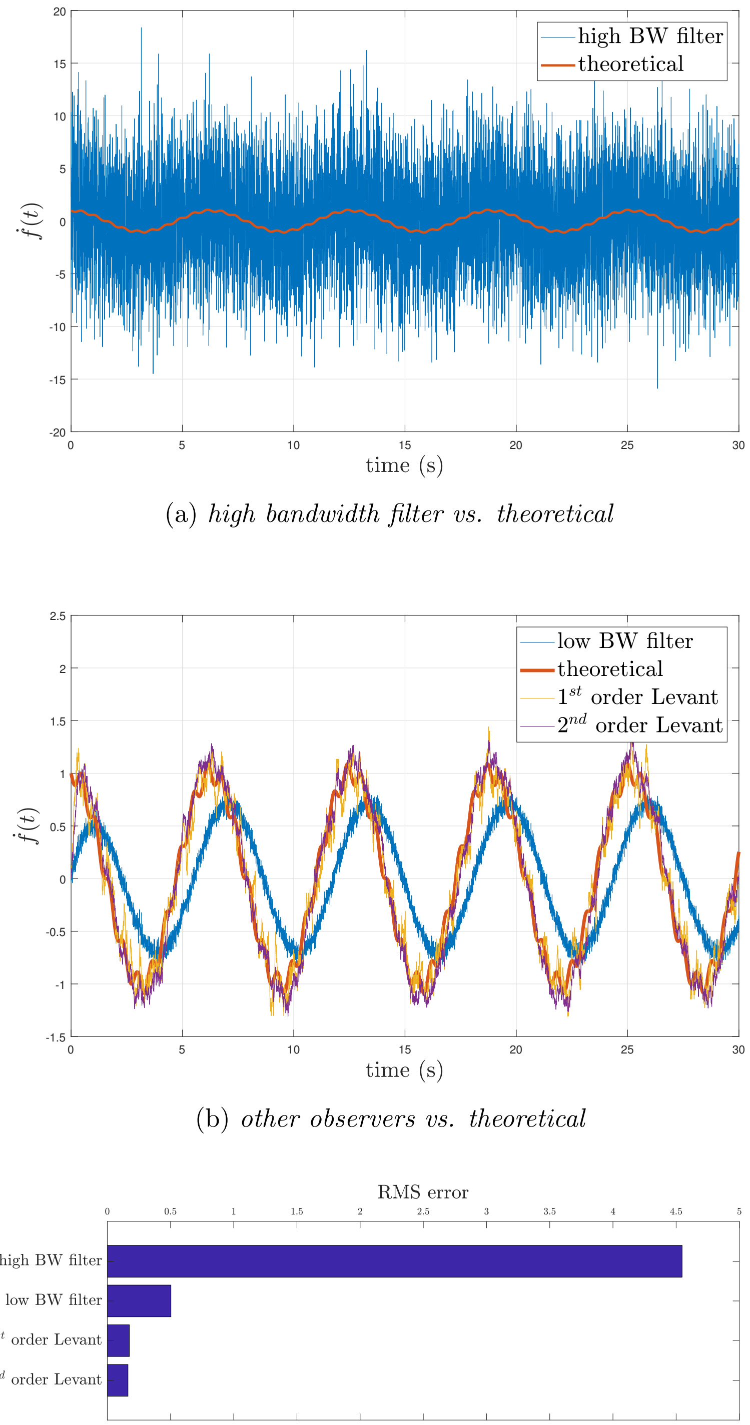


(a) *low bandwidth filter vs. theoretical*

(b) *other observers vs. theoretical (detail)*

(c) *differentiator error comparison*

Figure A.3: Simulation comparison between linear and non-linear algorithms without input noise



(a) *high bandwidth filter vs. theoretical*

(b) *other observers vs. theoretical*

(c) *differentiator error comparison*

Figure A.4: Simulation comparison between linear and non-linear algorithms with band-limited white noise on $f(t)$

RMS error	no noise	noise
Low bandwidth filter	0.4975	0.5033
High bandwidth filter	0.0100	4.5472
1^{st} order Levant diff.	0.0184	0.1745
2^{nd} order Levant diff.	0.0026	0.1635

Table A.2: Comparison of RMS errors for the four tested differentiators, with and without noise

Non-linear differentiation techniques demonstrate their high convergence rate both in the noise-free and disturbed case. In particular we can notice in figure A.3b that the estimate converges to the theoretical curve without introducing time delay. Moreover, the smoothing effect of higher order extension is clearly visible, since 1^{st} order Levant estimate suffers from chattering due to the discontinuity. Absence of phase lag is even more evident in figure A.4b, where non-linear differentiators are in phase with their reference despite uncertainty introduced by disturbances.

Finally, a numerical evaluation of RMS errors between ideal and estimated derivatives is performed.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\dot{\hat{f}}(i) - \dot{f}(i))^2} \tag{A.13}$$

Results are reported in table A.2.

Appendix B

Friction identification

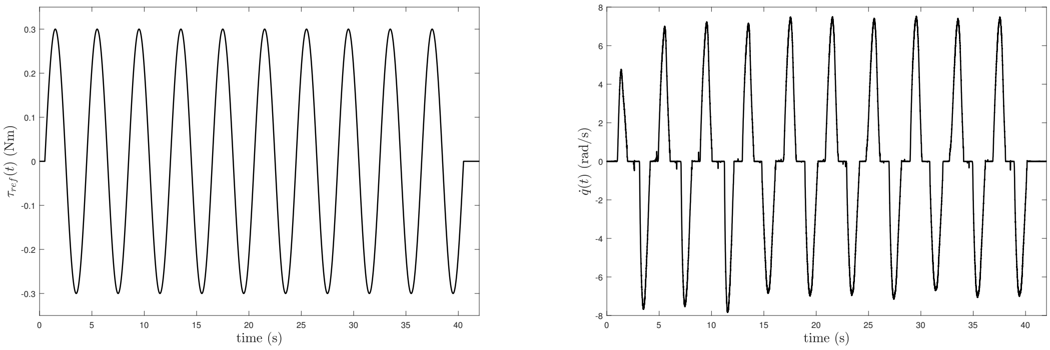
Among all the non-linearities that affect a manipulator dynamics, one of the most uncertain is surely the friction torque. Due to gear reduction stages and relatively slow velocities during common teleoperation procedures, friction may become comparable or even bigger with respect to the actuation torque that is necessary to perform the motion control.

In order to compensate, at least partially, its effect, we need to quantify joint friction through experimental tests. In this way, we can relax robustness constraints on joint controllers, reducing the overall control action required and improving the response.

The following sections report the identification and validation procedure adopted for the 7 DOFs `ABB`® `IRB 14000 YuMi` right arm. Wrist joints of this robot are the most critical from the friction point of view, because they are intended to be freely moved by the operator during end effector pose definition. Therefore, they show considerably high friction, which, in addition to small inertia, make them particularly difficult to control.

B.1 Friction - velocity map identification

Before choosing an a-priori friction model, it's better to look at the relationship that intervenes between friction torque and the joint speed, namely the *friction - velocity map*. This will give us a hint on the relevant phenomena which occur during joint



(a) $\tau_{ref}(t)$

(b) $\dot{q}(t)$

Figure B.1: Reference torque profile and measured joint speed during friction identification experiment

motion, allowing a more appropriate friction model choice.

According to [42], a proper strategy to generate the map consists in feeding the analysed joint with a low - frequency, sinusoidal torque input, while the remaining manipulator is locked. Doing so, we can reach joint *travel limits* and inspect a *wider range* of situations with respect to the classical “constant speed” experiments. Moreover, if the sinusoidal frequency is sufficiently low, Coriolis and centrifugal effects *may be neglected* (reduce uncertainties coming from robot dynamics).

After a few trials, the chosen input torque takes the following structure

$$\tau_{ref}(t) = 0.3 \sin\left(\frac{\pi}{2}t\right)$$

Using this signal we are able to span the full position range of the seventh joint ($[-\pi, \pi]$ rad), reaching also the available speed limit (≈ 8 rad/s). The cycle is iterated *ten times* in order to gather more data and check the trajectory repeatability. Experimental results are displayed in Figure B.1.

During the test, position is also logged, while the acceleration is retrieved by direct differentiation of the filtered velocity. Once all the kinematic quantities are acquired, we can straightforwardly apply the dynamic model in order to compute the friction torque τ_f.

$$\tau_f = \tau_{ref} - \boldsymbol{b_i(q)\ddot{q}} - \boldsymbol{c_i(q,\dot{q})\dot{q}} - g_i \tag{B.1}$$

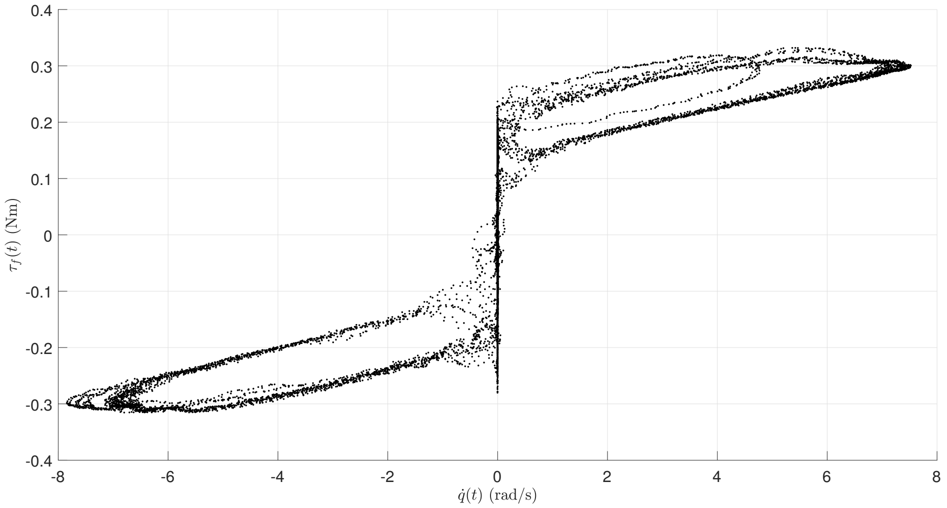


Figure B.2: Experimental friction curve obtained with the procedure described in section B.1

where subscript i indicates the i-th joint whose friction is identified, $\boldsymbol{b_i(q)}$ and $\boldsymbol{c_i(q, \dot{q})}$ are, respectively, the i-th row of the inertia matrix and the i-th row of the Coriolis - centrifugal matrix. g_i is the i-th element of the gravitational vector.

Once τ_f has been computed, we can finally plot the experimental friction - velocity map, which is visible in Figure B.2.

B.2 Friction model choice

Now that the friction profile is available, it makes sense to replicate its behavior using an appropriate friction model. According to literature [42] [69], they can be subdivided mainly into *two* categories:

- **Static models**: τ_f is represented by a non-linear curve in the speed - torque domain and it is *fixed with respect to time.* These models capture the main friction aspects and their parameters are quite easy to be identified. Their implementation is immediate (given joint speed and position, the curve's equation returns the friction value). They cannot represent time - dependent phenomena;

- **Dynamic models**: τ_f is computed based on the evolution of a *dynamic sys-*

tem's state variable. Doing so, it's possible to properly represent complex friction phenomena, such as *hysteresis* and *pre-sliding displacement.* These models require the solution of differential equations during manipulator's duty cycle. Therefore, they are computationally more expensive and their evolution can be hardly predicted. Since usually non-linear dynamic systems are chosen, also the estimator stability is difficult to be guaranteed;

If we look at Figure B.2, it's clear that the friction model we have to choose should model the following effects:

1. **Coulomb friction** τ_c: it's the *constant* amount of torque that is required to move the joint at any velocity. Its sign is determined by the speed's direction (Figure B.3a);

2. **Viscous friction** τ_v: it models the dependency between speed and friction. In our case, such relation is *linear* and can be described using a single coefficient $\tau_v = \kappa\dot{q}$ (Figure B.3b);

3. **Stiction torque** τ_s: it represents the amount of static friction that we need to overcome in order to rotate the joint, starting from *zero velocity condition* (Figure B.3c);

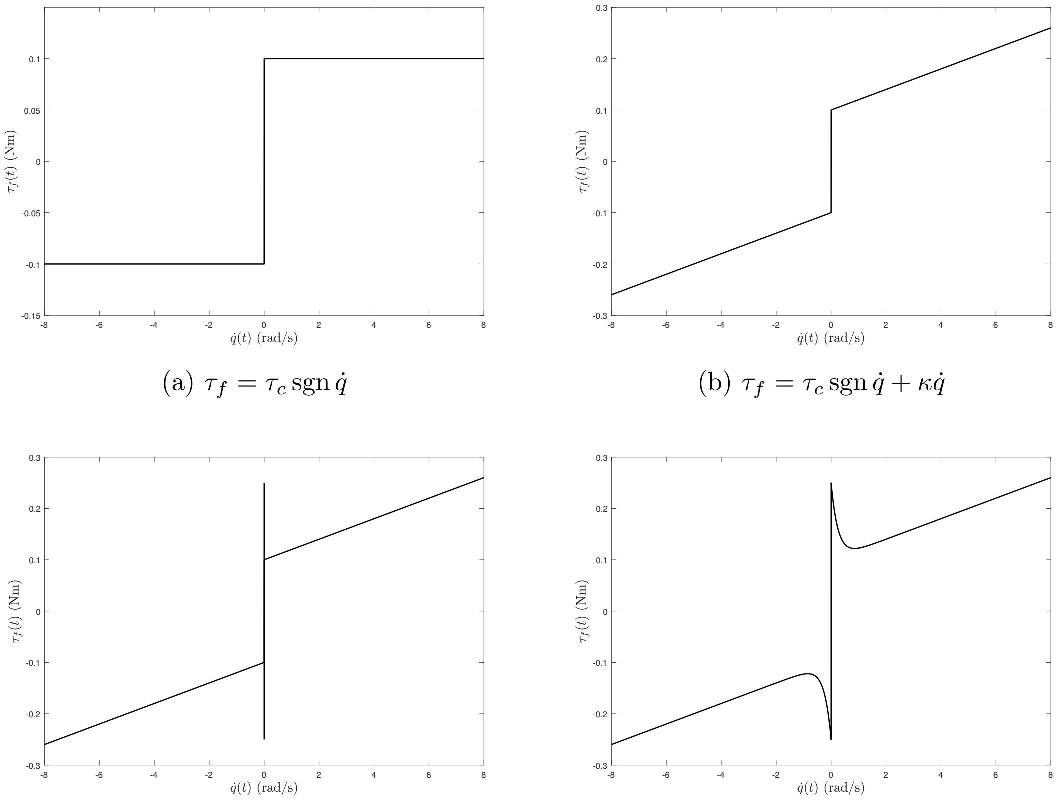
4. **Stribeck effect** α: it describes the transition phase between static and dynamic friction torque, generating a *negative viscous coefficient* at low speeds (Figure B.3d);

All the previous effects can be described using a *static model* and its curve can be directly interpolated from the experimental data. In particular, equation

$$\tau_f = [\tau_c + (\tau_s - \tau_c)e^{-\alpha|\dot{q}|}]\operatorname{sgn}\dot{q} + \kappa\dot{q} \tag{B.2}$$

which is called *Stribeck curve*, is suitable to model the majority of characteristics underlined by the friction - velocity map.

The only relevant aspect which has been neglected in Figure B.2 is the *hysteresis* shown between increasing and decreasing velocities. As we have said before, this



(a) $\tau_f = \tau_c \operatorname{sgn} \dot{q}$

(b) $\tau_f = \tau_c \operatorname{sgn} \dot{q} + \kappa \dot{q}$

(c) $\tau_f = \tau_c \operatorname{sgn} \dot{q} + \kappa \dot{q}, \dot{q} \neq 0 \wedge \pm \tau_s, \dot{q} = 0$

(d) $\tau_f = [\tau_c + (\tau_s - \tau_c) e^{-\alpha|\dot{q}|}] \operatorname{sgn} \dot{q} + \kappa \dot{q}$

Figure B.3: Static friction models that match the properties required by the experimental data

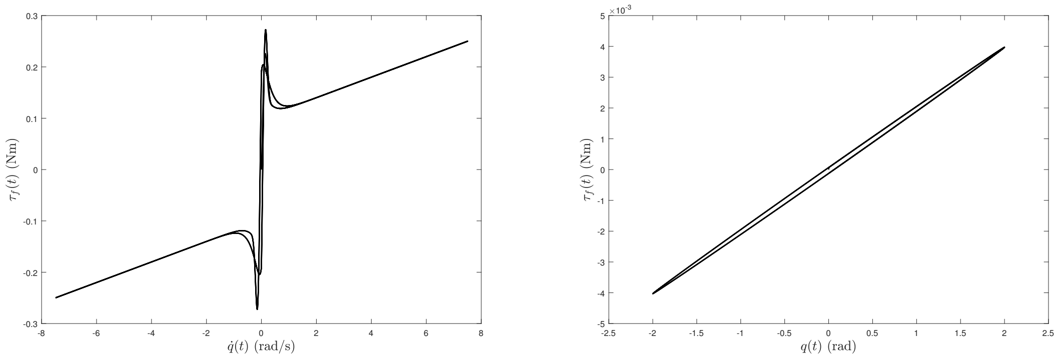
property cannot be obtained using a static model and thus should require a more complex dynamic one.

One of the most popular dynamic friction representations is the *LuGre model*. In its simpler formulation, it can be described by the following non-linear SISO system [41]:

$$\begin{aligned} \dot{z} &= \dot{q} - \frac{\sigma_0 |\dot{q}|}{\tau_c + (\tau_s - \tau_c) e^{-\alpha|\dot{q}|}} z \\ \tau_f &= \sigma_1 \dot{z} + \sigma_0 z + \kappa \dot{q} \end{aligned} \tag{B.3}$$

where z is the internal friction state, σ_0 and σ_1 describe the entity and shape of the memory effect and the remaining parameters have been already introduced in the static friction model.

To generate the estimated friction trajectory for this kind of model it's necessary



(a) *friction - velocity map*

(b) *friction - position hysteresis*

Figure B.4: LuGre model estimated friction evolution. Difference between increasing and decreasing speeds is clearly visible

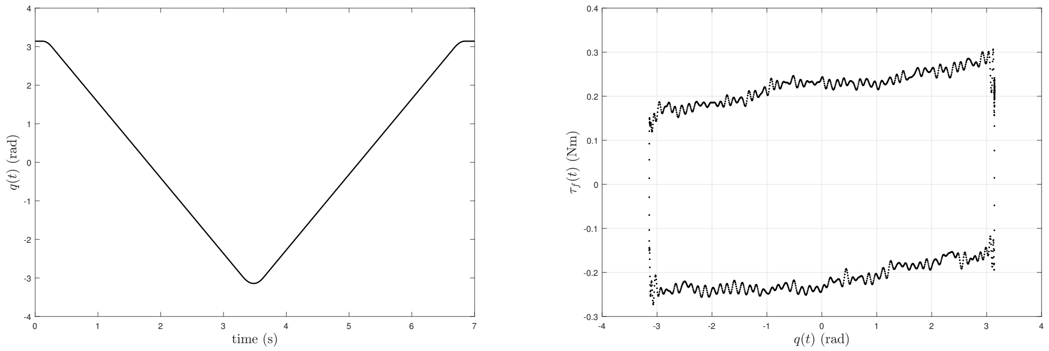
to *simulate the dynamic system.* Particular attention should be placed on the choice of σ_0 and σ_1 as well as the simulation sampling time, in order to avoid unstable behaviour. LuGre friction evolution can be inspected in Figure B.4.

As it can be seen, while the model is improved in the neighbour of zero speed, it still lacks to model the real behaviour for higher velocities. In other words, hysteresis in LuGre model is confined only to a *small portion* of the friction - speed map, which is different from the behaviour observed in our experimental data. Other drawbacks of this model are related to *manipulator implementation* (robot sampling time is slow compared to estimator convergence) as well as *difficulty in identifying parameters* (they have a weak physical meaning). For these reasons, the friction model has been implemented based on the Stribeck curve (Equation (B.2)).

B.3 Joint elasticity

Even though static friction carries a great number of advantages, it still lacks in representing the high velocity hysteresis underlined in the experiment. To investigate more deeply such phenomenon, a *new test* has been performed on the manipulator's joint.

Instead of giving a direct sinusoidal input torque as in Section B.1, the axis PID control loop has been kept active; its reference has been chosen in order to span the whole angular range at fixed speed $\dot{q}_j^0$. Doing so, we are guaranteeing that the



(a) *joint position trajectory*

(b) *friction - position map*

Figure B.5: Friction experiment performed to analyse joint elasticity

viscous friction contribution *stays constant* throughout the experiment (except for initial and final transients), leaving hysteresis as the only variable component.

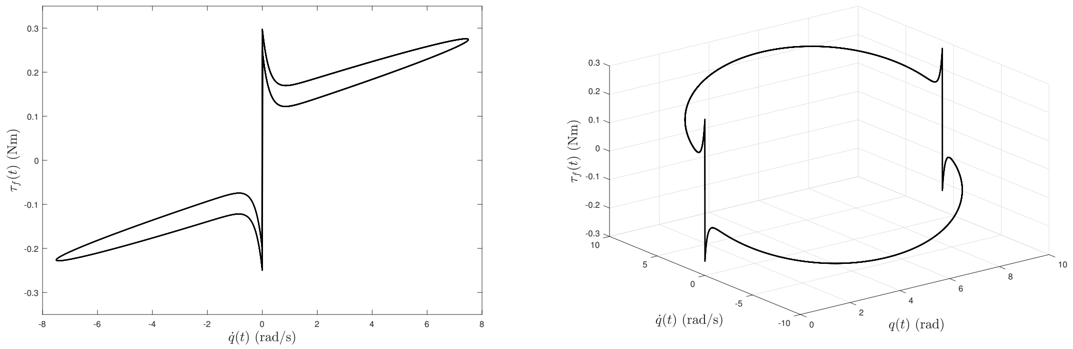
If friction had no memory, we should observe a *flat characteristic* for every encountered position. Instead, we observe a positive or negative gradient in the friction torque while the analysed joint spans different angular positions (Figure B.5). Also in this case, to compute τ_f (B.1) has been employed, substituting the sinusoidal contribute with the internal control input.

In view of the above, hysteresis effect can be explained by assuming a *spring-like behaviour* of the joint. Such position dependency is probably due to the internal cable rooting, which is put in tension or relaxed according to the different joint orientations.

This phenomenon can be easily approximated using a *linear torsional spring*, neglecting other non-linear effects such as dead zones or saturations. If we add this contribution to the static friction model (B.2), we can obtain a reasonable approximation of the experimental data acquired in Figure B.2.

$$\tau_f = [\tau_c + (\tau_s - \tau_c)e^{-\alpha|\dot{q}|}]\,\mathrm{sgn}\,\dot{q} + \kappa\dot{q} + \boxed{\eta(q + q_o)} \tag{B.4}$$

where η is the torsional spring stiffness, while q_o is a constant parameter which represents the angular position of the spring at rest. The term ηq_o generates a friction offset on τ_f, therefore it has the same effect as τ_c. Based on the test performed in Section B.1, the calculated friction offset should be equal to $\tau_c + \eta q_o$; so, τ_c



(a) *friction - velocity map*

(b) *friction - velocity - position 3D map*

Figure B.6: Graphical representation of the Stribeck curve with added torsional spring, as shown in Equation (B.4)

has to be obtained by subtracting the spring contribute and q_o takes the value of the angular position at which such offset is computed. However, since spring contribute is marginal with respect to the Coulomb effect, friction offset is $\approx \tau_c$ and $\eta(q + q_o) \approx \eta q$.

B.4 Parameter identification

Once the model has been chosen, we need to identify its parameters through the experimental data collected in sections B.1 and B.3. Starting from Equation (B.4), we can immediately notice that it *cannot* be expressed as a linear combination of the parameters, due to the presence of the exponential coefficient α. Therefore, it is convenient to rely on the *physical meaning* of each term, as proposed in [42].

First of all, we need to identify the viscous friction coefficient κ, for both positive and negative speeds. If we consider *high velocities*, we have that $e^{-\alpha|\dot{q}|} \rightarrow 0$ and the friction curve can be approximated as

$$\tau_f \approx \tau_c \operatorname{sgn} \dot{q} + \kappa \dot{q} + \eta q$$

Moreover, when $\dot{q}$ is increasing, displacement contribution ηq is *small* with respect to the speed one and the hysteresis effect does not yet take place. This is motivated by the fact that speed transient is much faster than the position one. Consequently, the spring behavior can be neglected and the expression further simplified:

$$\tau_f \approx \tau_c \operatorname{sgn} \dot{q} + \kappa \dot{q} \tag{B.5}$$

Expression (B.5) describes two straight lines, one for the positive and one for the negative speed - friction half-plane. To be more precise, both τ_c and κ can be split according to their sign

$$\tau_f \approx \begin{cases} \tau_c^+ + \kappa^+ \dot{q} & \dot{q} \geq 0 \\ \tau_c^- + \kappa^- \dot{q} & \dot{q} < 0 \end{cases} \tag{B.6}$$

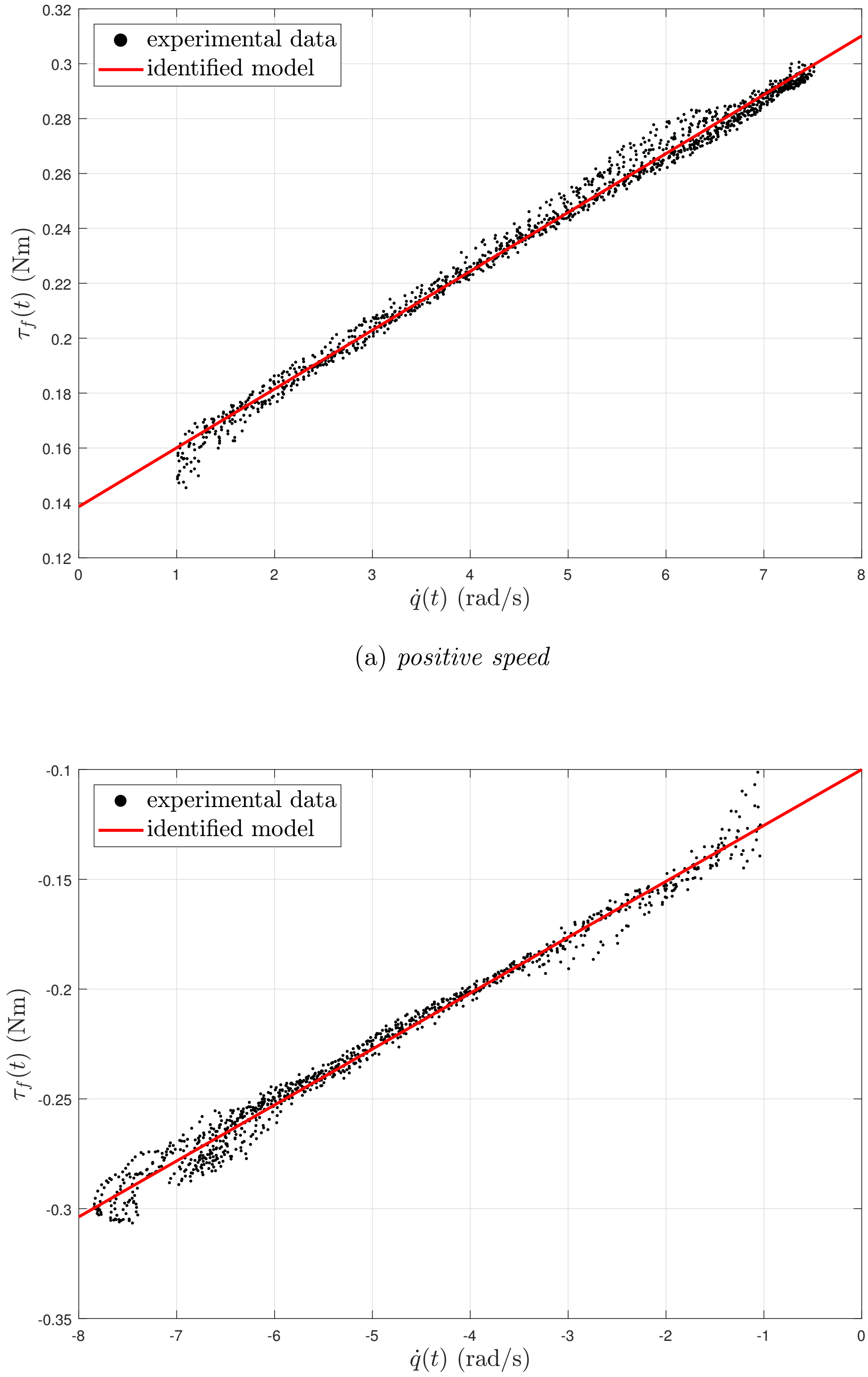
Parameters identification is now straightforward, since we just need to extrapolate, using least square method, the two linear relations from an appropriate segment of the experimental data (high velocity, small displacement). Graphical representation of this process is available in Figure B.7, while numerical values of the four identified parameters are reported in the first two rows of Table B.1.

For *stiction coefficients* τ_s^+ and τ_s^- the procedure is simpler. Since they determine the positive (negative) amplitude of the static friction peak, we can localize them by considering the data collected when speed is very close to zero. once selected, we can extract the maximum (minimum) value reached at each cycle and average them (Figure B.8).

Last static parameter left to be identified is the *exponential coefficient* α. This number is related with the inverse response of the friction curve at low joint speed. In particular, it can be approximated by the inverse of *Stribeck velocity* v_s, which is the lower point reached by the friction curve, before starting to show a linear viscous behaviour.

Knowing that, a *rough estimate* of α can be obtained by looking at the friction - velocity map. First, let's define a positive and negative threshold to split the linear viscous behaviour from the non-linear stiction one. The *inverse* of that threshold will be chosen as α. Computed values of α and τ_s are reported in Table B.1 and their graphical interpretation is explained in Figure B.8.

Now that we have concluded static friction curve estimation, we are left to evaluate *joint elasticity* to complete our process. For a correct η computation, we need



(a) *positive speed*

(b) *negative speed*

Figure B.7: Identification of viscous coefficients and Coulomb frictions

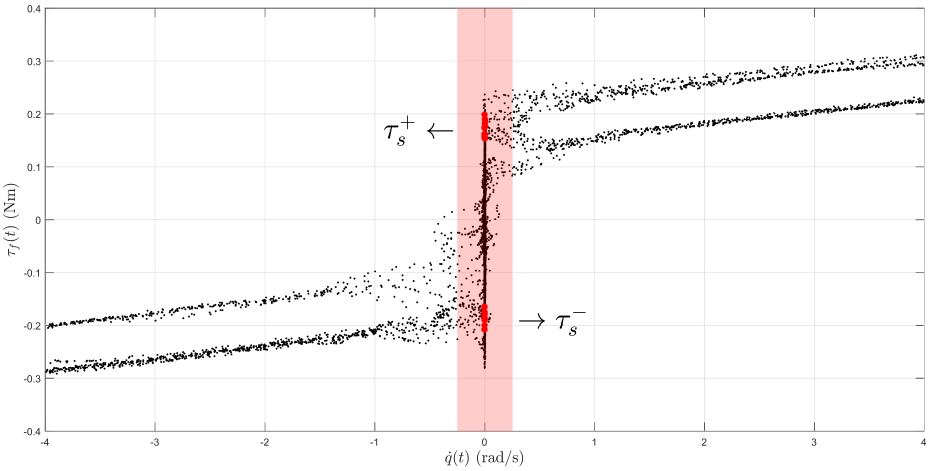


Figure B.8: Identification of stiction torque τ_s and inverse Stribeck speed α. Red dots shows the stiction for each cycle, while red area represents the inverse response phase

	Positive speed	Negative speed
κ	0.0214	0.0253
τ_c	0.1386	-0.1008
τ_s	0.1714	-0.1870
α	4	4
η	0.0183	0.0147
$\dot{q}_{th}$	0.005	-0.005

Table B.1: Experimental identified parameter values

to rely on the experiment done in section B.3, which underlines, in Figure B.5b, a linear relationship between friction torque and joint position.

As we have done for the viscous coefficient, also in this case we can proceed using least square method to retrieve the two straight lines from the empirical data. Result of this procedure is shown in Figure B.9, while coefficients are reported in the last row of Table B.1.

B.5 Implementation and validation

The conclusive section of this appendix deals with the realization of a *feed-forward friction torque compensator*, which will partially remove joint model uncertainties. In this way, the sliding mode controller is allowed to use a smaller gain, improving

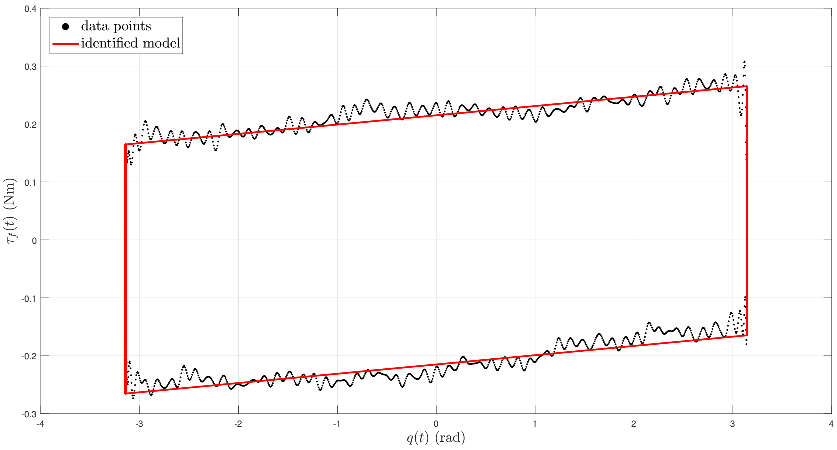


Figure B.9: Identification of joint elastic stiffness and comparison with the experimental result

stability properties as well as attenuating chattering.

Before directly inserting equation (B.4) into the manipulator's code, we need to deal properly with the *zero-speed discontinuity* shown by our model. If we choose to keep the signum function as it is, measurement noise or any other fluctuation source will cause strong vibrations at joint level, due to the high frequency switching between positive and negative feed-forward torque.

A possible solution to this problem is the implementation of *symmetric thresholds* on joint speed, deactivating friction compensation when we are inside a chosen interval. Attention should be placed on the choice of these limits. If they are too high, friction compensation *becomes useless* since where there is more need of it (high static friction) we are not supporting the controller at all. On the other hand, too tight thresholds *do not attenuate chattering.*

A good compromise between smooth and reactive response can be reached through a *linear interpolation* between the two branches of the friction curve. Doing so, we assure the continuity of the compensation while reducing solicitations at zero speed. This reasoning leads to equation (B.7). The main drawback of this approach is the progressive *loss of the Stribeck effect* when the threshold is enlarged, as shown in Figure B.10.

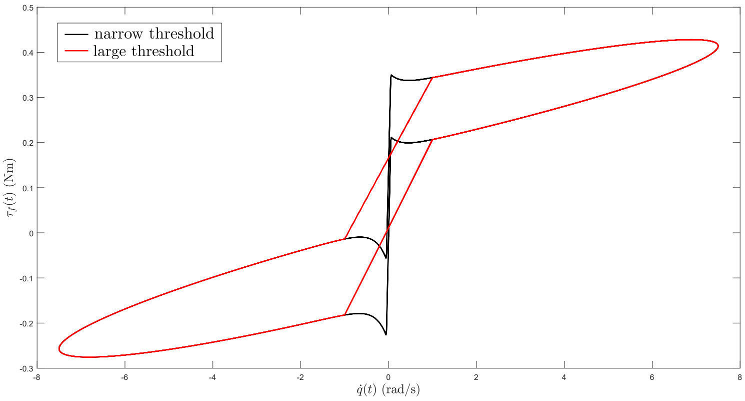


Figure B.10: Effect of the linear approximation on the original friction curve

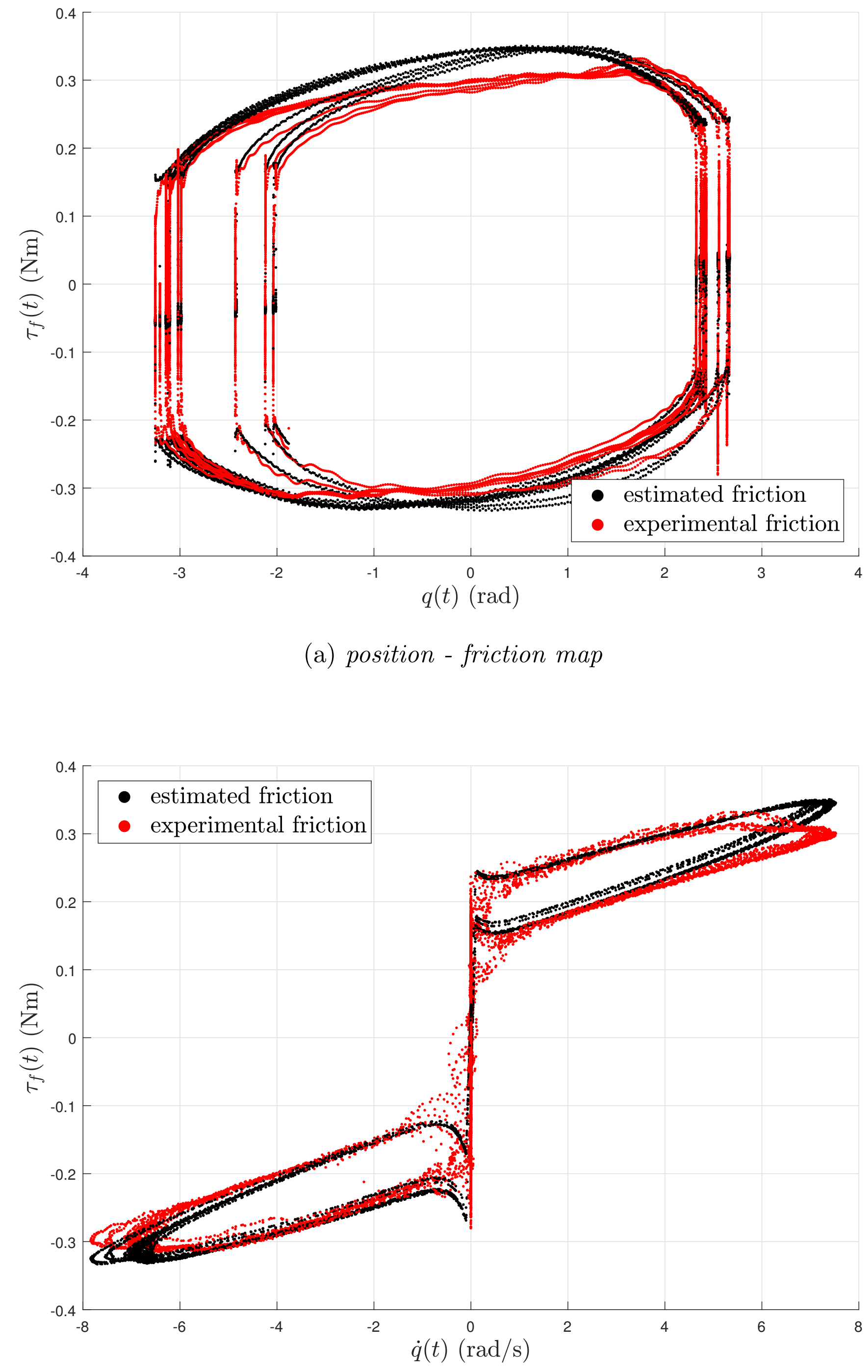
$$\tau_f = \begin{cases} [\tau_c + (\tau_s - \tau_c)e^{-\alpha|\dot{q}|}]\,\mathrm{sgn}\,\dot{q} + \kappa\dot{q} + \eta(q + q_o), & |\dot{q}| \geq \dot{q}_{th} \\ \frac{\tau_{th}^{+} - \tau_{th}^{-}}{2\dot{q}_{th}}\dot{q} + \frac{\tau_{th}^{+} + \tau_{th}^{-}}{2}, & |\dot{q}| < \dot{q}_{th} \end{cases} \tag{B.7}$$

where $\dot{q}_{th}$ is the speed threshold, while $\tau_{th}^{+} = \tau_f|_{\dot{q}_{th}}$ and $\tau_{th}^{-} = \tau_f|_{-\dot{q}_{th}}$ represent the friction curve values at positive and negative threshold speed. Numerical values adopted for $\pm\dot{q}_{th}$ are reported in Table B.1.

Once this last correction is done, we can compare the experimental results with the estimated ones (Figure B.11). As we can clearly see, *all the properties* we are interested in have been modelled quite accurately. The only remarkable mismatch is at high speeds, where experimental data show a bend with respect to the straight estimated ones. This phenomenon is probably due to the linear approximation done for the viscous friction as well as the simplified elastic joint model. This last argument is confirmed if we look at the friction - position map, where we see the strong non linearity during position variation. Another aspect that should be considered is the accuracy of the manipulator dynamic model, which directly reflects in the friction estimation accuracy.

Once we have checked the accuracy of our model with the experimental data used to fit it, we need now to perform a more thorough test, in which we validate the model over a generic joint trajectory. This test will be a true indicator of the goodness of our procedure.

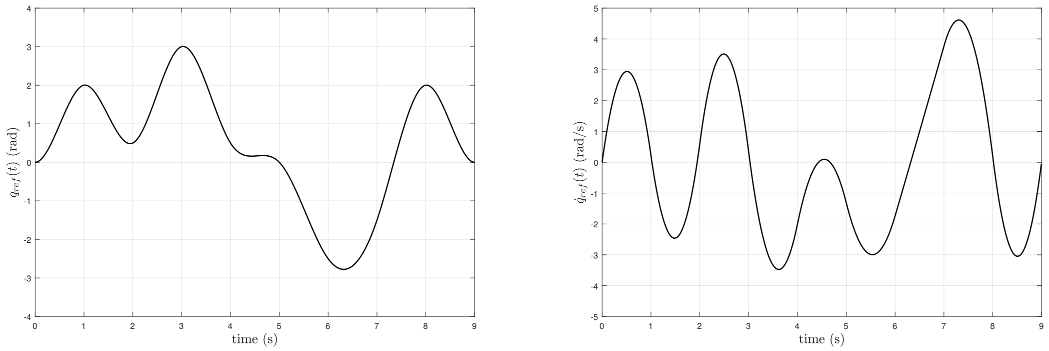
A generic spline trajectory has been set up, to span a wider range of possible



(a) *position - friction map*

(b) *velocity - friction map*

Figure B.11: Comparison between estimated and experimental friction values



(a) *position reference*

(b) *velocity reference*

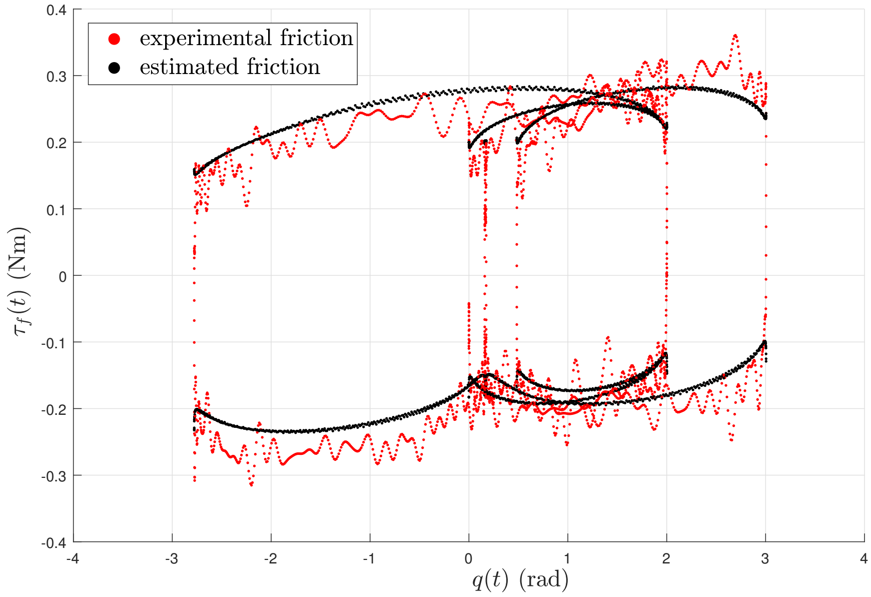
Figure B.12: Random spline trajectory used for friction model validation

positions and speeds. Then, to execute it properly, this signal has been given as a reference input for the internal manipulator control loop (Figure B.12).

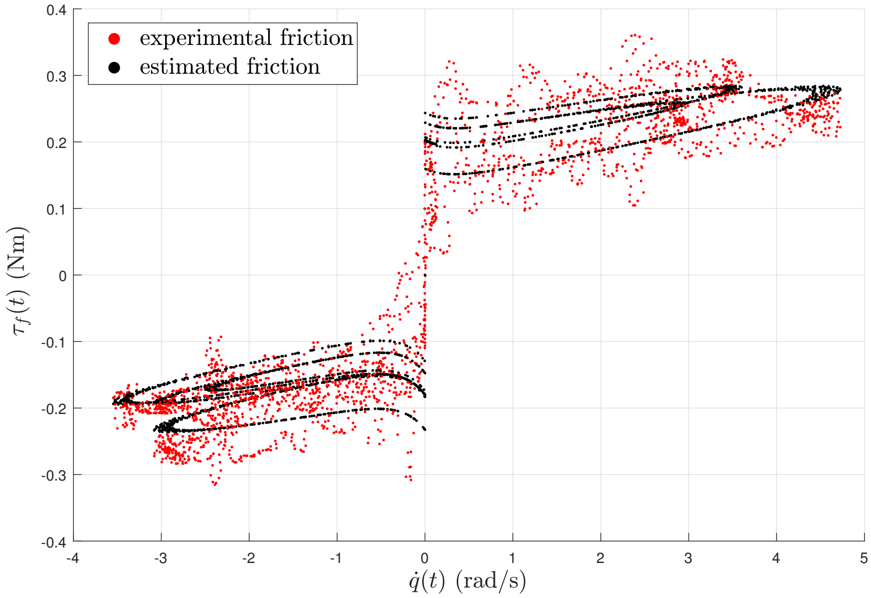
Once the joint torque has been recorded, we apply the same procedure as seen in section B.3 for joint elasticity to extract the friction component from the signal. Then, it is just a matter of comparing the validation curve with the estimated one (Figure B.13).

This time, experimentally computed friction is less predictable, since movements of the joint do not follow a standard cycle and torque measure is affected by noise (during identification test τ_{ref} was imposed). However, we can appreciate that the estimated curve still mimics the main features of the data, giving a meaningful, although rough, estimation of the friction coefficient in almost any situation.

For the sake of completeness, error evaluation between the actual and estimated friction has been performed. RMS value of the error between computed friction torque and estimated one is 0.0439 N m. The majority of such error is focused around zero. In Figure B.14, when the joint does not move, it is not possible to determine the actual friction, because the future rotation direction is unknown. However, the overall performance is quite acceptable and the friction profile tracking reflects the system real behaviour.

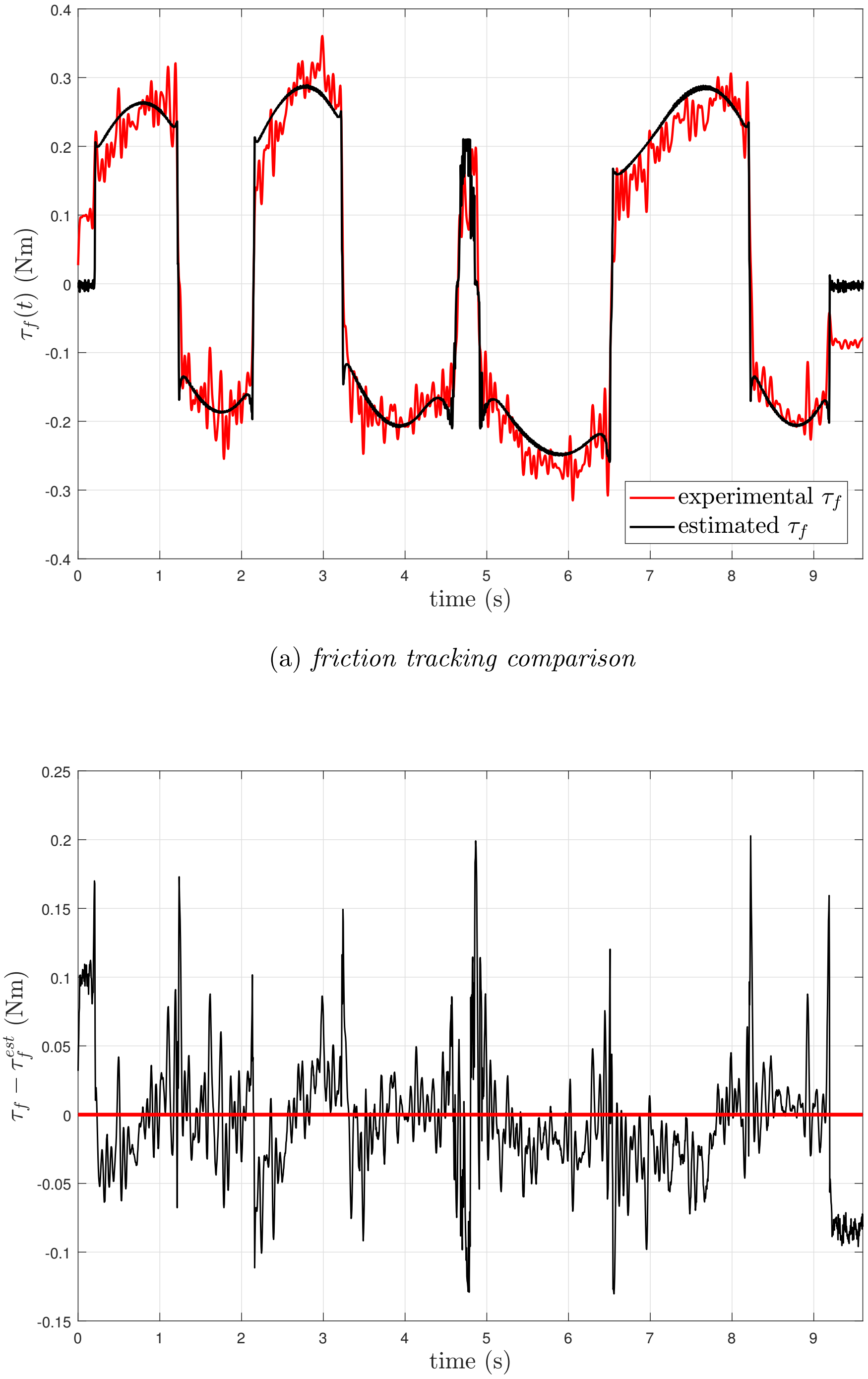


(a) *position - friction map*



(b) *velocity - friction map*

Figure B.13: Comparison between validation friction curve and estimated one



(a) *friction tracking comparison*

(b) *friction tracking error*

Figure B.14: Friction estimation error analysis

Bibliography

[1] R. J. Adams and B. Hannaford. “Control law design for haptic interfaces to virtual reality”. In: *IEEE Transactions on Control Systems Technology* 10.1 (Jan. 2002), pp. 3–13. ISSN: 1063-6536. DOI: 10.1109/87.974333.

[2] Richard J Adams and Blake Hannaford. “Stable haptic interaction with virtual environments”. In: *IEEE Transactions on robotics and Automation* 15.3 (1999), pp. 465–474.

[3] I. Aliaga, A. Rubio, and E. Sanchez. “Experimental quantitative comparison of different control architectures for master-slave teleoperation”. In: *IEEE Transactions on Control Systems Technology* 12.1 (Jan. 2004), pp. 2–11. ISSN: 1063-6536. DOI: 10.1109/TCST.2003.819586.

[4] Paolo Arcara and Claudio Melchiorri. “Control schemes for teleoperation with time delay: A comparative study”. In: *Robotics and Autonomous Systems* 38.1 (2002), pp. 49–64. ISSN: 0921-8890. DOI: https://doi.org/10.1016/S0921-8890(01)00164-6.

[5] G. Bartolini, A. Ferrara, and E. Usai. “Chattering avoidance by second-order sliding mode control”. In: *IEEE Transactions on Automatic Control* 43.2 (Feb. 1998), pp. 241–246. DOI: 10.1109/9.661074.

[6] Anders Blomdell et al. “Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers”. In: *IEEE International Conference on Robotics and Automation, 2010*. 2010, pp. 62–66.

[7] A. Calanca et al. “Improving continuous approximation of Sliding Mode Control”. In: *2013 16th International Conference on Advanced Robotics (ICAR)*. Nov. 2013, pp. 1–6. DOI: 10.1109/ICAR.2013.6766486.

[8] Vinay Chawda, Ozkan Celik, and Marcia K O’Malley. “Application of levant’s differentiator for velocity estimation and increased z-width in haptic interfaces”. In: *World Haptics Conference (WHC), 2011 IEEE*. IEEE. 2011, pp. 403–408. DOI: 10.1109/WHC.2011.5945520.

[9] Hyun Chul Cho and Jong Hyeon Park. “Impedance controller design of internet-based teleoperation using absolute stability concept”. In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 3. IEEE. 2002, pp. 2256–2261.

[10] Hyun Chul Cho and Jong Hyeon Park. “Stable bilateral teleoperation under a time delay using a robust impedance control”. In: *Mechatronics* 15.5 (2005), pp. 611–625. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2004.05.006.

[11] Göran A.V. Christiansson. “Hard Master Soft Slave Haptic Teleoperation”. PhD thesis. Delft University of Technology, 2007.

[12] J. E. Colgate and J. M. Brown. “Factors affecting the Z-Width of a haptic display”. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. May 1994, 3205–3210 vol.4. DOI: 10.1109/ROBOT.1994.351077.

[13] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*. Vol. 118. Springer, 2017.

[14] Massimo D’Agostino. “Sliding Mode Control: an output based approach”. Master thesis. Università degli studi di Padova, 2012.

[15] Alessandro De Luca. “Collision detection and robot reaction”. In: ().

[16] Qi-Wen Deng, Qing Wei, and Ze-Xiang Li. “Analysis of absolute stability for time-delay teleoperation systems”. In: *International Journal of Automation and Computing* 4.2 (2007), pp. 203–207.

[17] A. Ferrara and G. P. Incremona. “Design of an Integral Suboptimal Second-Order Sliding Mode Controller for the Robust Motion Control of Robot Manipulators”. In: *IEEE Transactions on Control Systems Technology* 23.6 (Nov. 2015), pp. 2316–2325. ISSN: 1063-6536. DOI: 10.1109/TCST.2015.2420624.

[18] A. Ferrara, G. P. Incremona, and L. Magni. “A robust MPC/ISM hierarchical multi-loop control scheme for robot manipulators”. In: *52nd IEEE Conference on Decision and Control*. Dec. 2013, pp. 3560–3565. DOI: 10.1109/CDC.2013.6760430.

[19] Antonella Ferrara. *Sliding Mode Control Handout*. course slides. Università di Pavia.

[20] H.J. Ferreau et al. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6.4 (2014), pp. 327–363.

[21] D. Feth et al. “Control-theoretic model of haptic human-human interaction in a pursuit tracking task”. In: *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*. Sept. 2009, pp. 1106–1111. DOI: 10.1109/ROMAN.2009.5326316.

[22] Thierry Floquet and Jean-Pierre Barbot. “Super twisting algorithm based step-by-step sliding mode observers for nonlinear systems with unknown inputs”. In: *International Journal of Systems Science* 38.10 (2007), pp. 803–815. DOI: 10.1080/00207720701409330.

[23] Michel Franken et al. “Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency”. In: *IEEE transactions on robotics* 27.4 (2011), pp. 741–756.

[24] W. García-Gabín and E. F. Camacho. “Sliding mode model based predictive control for non minimum phase systems”. In: *2003 European Control Conference (ECC)*. Sept. 2003, pp. 904–909.

[25] Winston Garcia-Gabin, Darine Zambrano, and Eduardo F. Camacho. "Sliding mode predictive control of a solar air conditioning plant". In: *Control Engineering Practice* 17.6 (2009), pp. 652–663. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2008.10.015. URL: http://www.sciencedirect.com/science/article/pii/S0967066108001925.

[26] Winston García-Gabín, Darine Zambrano, and E.F. Camacho. "Sliding mode predictive control for a chemical process with time delay". In: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, pp. 627–632. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20050703-6-CZ-1902.01678. URL: http://www.sciencedirect.com/science/article/pii/S147466701637690X.

[27] L. G. Garcia-Valdovinos, V. Parra-Vega, and M. A. Arteaga. "Higher-order sliding mode impedance bilateral teleoperation with robust state estimation under constant unknown time delay". In: *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* July 2005, pp. 1293–1298. DOI: 10.1109/AIM.2005.1511189.

[28] L. g. Garcia-Valdovinos, V. Parra-Vega, and M. A. Arteaga. "Observer-based Higher-Order Sliding Mode Impedance Control of Bilateral Teleoperation under Constant Unknown Time Delay". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Oct. 2006, pp. 1692–1699. DOI: 10.1109/IROS.2006.282126.

[29] Amir Haddadi. *Stability, performance, and implementation issues in bilateral teleoperation control and haptic simulation systems.* Queen's University (Canada), 2012.

[30] Mirza Tariq Hamayun, Christopher Edwards, Halim Alwi, et al. *Fault tolerant control schemes using integral sliding modes.* Springer, 2016.

[31] B. Hannaford and R. Anderson. "Experimental and simulation studies of hard contact in force reflecting teleoperation". In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation.* Apr. 1988, 584–589 vol.1. DOI: 10.1109/ROBOT.1988.12114.

[32] Blake Hannaford and Jee-Hwan Ryu. "Time-domain passivity control of haptic interfaces". In: *IEEE Transactions on Robotics and Automation* 18.1 (2002), pp. 1–10.

[33] Keyvan Hashtrudi-Zaad and SE Salcudean. "Analysis and evaluation of stability and performance robustness for teleoperation control architectures". In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*. Vol. 4. IEEE. 2000, pp. 3107–3113.

[34] Keyvan Hashtrudi-Zaad and Septimiu E. Salcudean. "Analysis of Control Architectures for Teleoperation Systems with Impedance/Admittance Master and Slave Manipulators". In: *The International Journal of Robotics Research* 20.6 (2001), pp. 419–445. DOI: 10.1177/02783640122067471.

[35] Keyvan Hashtrudi-Zaad and Septimiu E Salcudean. "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation". In: *IEEE Transactions on Robotics and Automation* 18.1 (2002), pp. 108–114.

[36] Simon S Haykin. *Active network theory*. Vol. 2680. Addison-Wesley, 1970.

[37] DJF Heck. "Delayed bilateral teleoperation: A direct force-reflecting control approach". 2015.

[38] Peter F Hokayem and Mark W Spong. "Bilateral teleoperation: An historical survey". In: *Automatica* 42.12 (2006), pp. 2035–2057. DOI: 10.1016/j.automatica.2006.06.027.

[39] G. P. Incremona, A. Ferrara, and L. Magni. "MPC for Robot Manipulators With Integral Sliding Modes Generation". In: *IEEE/ASME Transactions on Mechatronics* 22.3 (June 2017), pp. 1299–1307. ISSN: 1083-4435. DOI: 10.1109/TMECH.2017.2674701.

[40] Uri Itkis. *Control systems of variable structure*. Wiley New York, 1976.

[41] Karl Johanastrom and Carlos Canudas-De-Wit. "Revisiting the LuGre friction model". In: *IEEE control Systems* 28.6 (2008), pp. 101–114.

[42] M. R. Kermani, R. V. Patel, and M. Moallem. "Friction Identification and Compensation in Robotic Manipulators". In: *IEEE Transactions on Instrumentation and Measurement* 56.6 (Dec. 2007), pp. 2346–2353. ISSN: 0018-9456. DOI: 10.1109/TIM.2007.907957.

[43] Hassan K Khalil. "Noninear systems". In: *Prentice-Hall, New Jersey* 2.5 (1996), pp. 505–540.

[44] D. A. Lawrence. "Stability and transparency in bilateral teleoperation". In: *IEEE Transactions on Robotics and Automation* 9.5 (Oct. 1993), pp. 624–637. ISSN: 1042-296X. DOI: 10.1109/70.258054.

[45] D. Lee and M. W. Spong. "Passive Bilateral Teleoperation With Constant Time Delay". In: *IEEE Transactions on Robotics* 22.2 (Apr. 2006), pp. 269–281. ISSN: 1552-3098. DOI: 10.1109/TRO.2005.862037.

[46] F. H. F. Leung, L. K. Wong, and P. K. S. Tam. "Algorithm for eliminating chattering in sliding mode control". In: *Electronics Letters* 32.6 (Mar. 1996), pp. 599–601. DOI: 10.1049/el:19960373.

[47] A Levant. "Introduction to high-order sliding modes". In: *School of Mathematical Sciences, Israel* 58.6 (2003), p. 1.

[48] Arie Levant. "Robust Exact Differentiation via Sliding Mode Technique". In: *Automatica* 34.3 (Mar. 1998), pp. 379–384. DOI: 10.1016/S0005-1098(97)00209-4.

[49] Arie Levant. "Sliding order and sliding accuracy in sliding mode control". In: *International journal of control* 58.6 (1993), pp. 1247–1263.

[50] S Lichiardopol. "A survey on teleoperation". In: *Technische Universitat Eindhoven, DCT report* (2007).

[51] Lonnie J Love and Wayne John Book. "Contact stability analysis of virtual walls". In: Georgia Institute of Technology. 1995.

[52] Emanuele Magrini, Fabrizio Flacco, and Alessandro De Luca. “Estimation of contact forces using a virtual force sensor”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 2126–2133.

[53] D. T. McRuer and H. R. Jex. “A Review of Quasi-Linear Pilot Models”. In: *IEEE Transactions on Human Factors in Electronics* HFE-8.3 (Sept. 1967), pp. 231–249. ISSN: 0096-249X. DOI: 10.1109/THFE.1967.234304.

[54] Hunter Mk and DE Fry. “The use of Cross-Correlation and Power SDectraI Techniques for the Identification of the Hunter Mk. 12 Dynamic Response”. In: (1970), pp. 4–6.

[55] Emmanuel Nuño, Luis Basañez, and Romeo Ortega. “Passivity-based control for bilateral teleoperation: A tutorial”. In: *Automatica* 47.3 (2011), pp. 485–495.

[56] E. NuÑo et al. “A Globally Stable PD Controller for Bilateral Teleoperators”. In: *IEEE Transactions on Robotics* 24.3 (June 2008), pp. 753–758. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.921565.

[57] Roberto Oboe and Paolo Fiorini. “Issues on Internet-Based Teleoperation”. In: *IFAC Proceedings Volumes* 30.20 (1997). 5th IFAC Symposium on Robot Control 1997 (SYROCO ’97), Nantes, France, 3-5 September, pp. 591–597. ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)44322-9.

[58] Renè M. Overney. *Contact Mechanics*. course notes. University of Washington, 2010.

[59] Jong Hyeon Park and Hyun Chul Cho. “Sliding-mode controller for bilateral teleoperation with varying time delay”. In: *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)*. Sept. 1999, pp. 311–316. DOI: 10.1109/AIM.1999.803184.

[60] Carolina Passenberg, Angelika Peer, and Martin Buss. “A survey of environment, operator, and task-adapted controllers for teleoperation systems”.

English. In: *Mechatronics* 20.7 (2010), pp. 787–801. DOI: 10.1016/j.mechatronics.2010.04.005.

[61] Alessandro Pisano. "Second order sliding modes: theory and applications". In: *Universita Studi di Cagliari, Diss* (2000).

[62] Alfio Quarteroni, Fausto Saleri, and Paola Gervasio. *Scientific Computing with MATLAB and Octave*. Springer, 2014, pp. 115–117.

[63] G. J. Raju, G. C. Verghese, and T. B. Sheridan. "Design issues in 2-port network models of bilateral remote manipulation". In: *Proceedings, 1989 International Conference on Robotics and Automation*. May 1989, 1316–1321 vol.3. DOI: 10.1109/ROBOT.1989.100162.

[64] G Jagannath Raju, George C Verghese, and Thomas B Sheridan. "Design issues in 2-port network models of bilateral remote manipulation". In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE. 1989, pp. 1316–1321.

[65] Jorge Rivera et al. "Super-twisting sliding mode in motion control systems". In: *Sliding mode control*. InTech, 2011.

[66] M. Rubagotti et al. "Robust Model Predictive Control With Integral Sliding Mode in Continuous-Time Sampled-Data Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 56.3 (Mar. 2011), pp. 556–570. ISSN: 0018-9286. DOI: 10.1109/TAC.2010.2074590.

[67] Jee-Hwan Ryu, Jordi Artigas, and Carsten Preusche. "A passive bilateral control scheme for a teleoperator with time-varying communication delay". In: *Mechatronics* 20.7 (2010), pp. 812–823.

[68] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. "Stable teleoperation with time-domain passivity control". In: *IEEE Transactions on robotics and automation* 20.2 (2004), pp. 365–373.

[69] Sergio Sanchez and Ricardo Campa. "An Improvement Proposal to the Static Friction Model". In: 2013 (June 2013).

[70] T. B. Sheridan. "Space teleoperation through time delay: review and prognosis". In: *IEEE Transactions on Robotics and Automation* 9.5 (Oct. 1993), pp. 592–606. ISSN: 1042-296X. DOI: 10.1109/70.258052.

[71] Thomas B Sheridan. *Telerobotics, automation, and human supervisory control*. MIT press, 1992.

[72] Yuri Shtessel et al. "Introduction: Intuitive Theory of Sliding Mode Control". In: *Sliding Mode Control and Observation*. New York, NY: Springer New York, 2014, pp. 1–42. ISBN: 978-0-8176-4893-0. DOI: 10.1007/978-0-8176-4893-0_1. URL: https://doi.org/10.1007/978-0-8176-4893-0_1.

[73] B. Siciliano. *Robotica. Modellistica, pianificazione e controllo*. Collana di istruzione scientifica: Serie di automatica. McGraw-Hill Companies, 2008. ISBN: 9788838663222.

[74] S. Spurgeon. "Sliding mode control: a tutorial". In: *2014 European Control Conference (ECC)*. July 2014, pp. 2272–2277. DOI: 10.1109/ECC.2014.6862622.

[75] Robert J. Stone. "Haptic feedback: a brief history from telepresence to virtual reality". In: *Haptic Human-Computer Interaction*. Ed. by Stephen Brewster and Roderick Murray-Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–16. ISBN: 978-3-540-44589-0.

[76] Sezai Tokat, M Sami Fadali, and Osman Eray. "A classification and overview of sliding mode controller sliding surface design methods". In: *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*. Springer, 2015, pp. 417–439.

[77] V. Utkin. "Variable structure systems with sliding modes". In: *IEEE Transactions on Automatic Control* 22.2 (Apr. 1977), pp. 212–222. DOI: 10.1109/TAC.1977.1101446.

[78] V. Utkin and Hoon Lee. "Chattering Problem in Sliding Mode Control Systems". In: *International Workshop on Variable Structure Systems, 2006. VSS'06*. July 2006, pp. 346–350. DOI: 10.1109/VSS.2006.1644542.

[79] Vadim Utkin, Jürgen Guldner, and Jingxin Shi. *Sliding mode control in electromechanical systems*. CRC press, 2017. Chap. 7.

[80] T. A. Várkonyi et al. "Survey on the control of time delay teleoperation systems". In: *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*. July 2014, pp. 89–94. DOI: 10.1109/INES.2014.6909347.

[81] Marilena Vendittelli. *Stability and transparency in bilateral teleoperation*. course slides. Università di Roma La Sapienza, 2012.

[82] R W. Daniel and Peter Mcaree. "Fundamental Limits of Performance for Force Reflecting Teleoperation". In: 17 (Aug. 1998), pp. 811–830.

[83] David W. Weir and J. Edward Colgate. "Stability of Haptic Displays". In: 2009.

[84] D Whitney. "State space models of remote manipulation tasks". In: *IEEE Transactions on Automatic Control* 14.6 (1969), pp. 617–623.

[85] J.G.W. Wildenbeest et al. "How operator admittance affects the response of a teleoperation system to assistive forces – A model analytic study and simulation". In: *Fusion Engineering and Design* 88.9 (2013). Proceedings of the 27th Symposium On Fusion Technology (SOFT-27); Liège, Belgium, September 24-28, 2012, pp. 2001–2005. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2013.02.001.

[86] B. Willaert et al. "Transparency Trade-Offs for a 3-Channel Controller Revealed by the Bounded Environment Passivity Method". In: *2010 Third International Conference on Advances in Computer-Human Interactions*. Feb. 2010, pp. 66–72. DOI: 10.1109/ACHI.2010.22.