UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

DOTTORATO DI RICERCA IN
INDUSTRIAL INNOVATION ENGINEERING

Ciclo XXXI

ENERGY-BASED CONTROL FOR SIMULATION OF MULTI-BODY DYNAMICS USING ROBOTIC FACILITIES

MARCO DE STEFANO

Relatore: Prof. Cristian Secchi
Università degli Studi di Modena e Reggio Emilia

Correlatore: Prof. Alin Albu-Schäffer
Technical University of Munich and German Aerospace Center

Coordinatore del Corso di Dottorato: Prof. Mauro Dell'Amico
Università degli Studi di Modena e Reggio Emilia

Year 2018

PREFACE

This dissertation is based on the research undertaken at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany. The research involves the work done through several years and it would have been impossible to finalise it without the support of many people.

I would like to express my deepest gratitude to my supervisor and mentor Prof. Cristian Secchi for his guidance and inspiring discussions we had throughout the course of this work. I wish to thank Prof. Alin Albu-Schaeffer, who supported my research in DLR and Dr. Christian Ott for his support and valuable scientific input.

I would like to thank all my colleagues from the DLR, with whom I had the honour to work. Special thanks go to Dr. Jordi Artigas and Ribin Balachandran with whom I collaborated in a very productive way. They created the right atmosphere and motivation I needed to pursue this work. I thank also my team members, Lampariello Roberto, Konstantin Kondak, Wolfgang Rackl, Alessandro Giordano, Phillip Schmidt and in particular Hrishik Mishra for his motivation in triggering always interesting problems to solve. Many thanks go to Martin Stelzer, Bernhard Brunner, Alex Beyer, Wieland Bertleff and Martin Heumos for the great support in software and hardware maintenance. I would like also to thank the colleagues I met in European Union and ESA projects, whom I had the honour to work with.

I wish to thank all my friends and in particular Ribin, Karan, Christian, Hrishik, Korbinian, Gianluca, Giuseppe, Sergi, Theo, Marko, Matteo, Daichi and Gustav for their cheerfulness and all the good times spent together.

I want to express my deepest gratitude to my parents, Florinda and Francesco, for their unconditional love and support, to my sisters, Arianna and Maria Paola, who always encouraged me and to my brothers-in-law, Vincenzo and Ulderico, for the good times we spent together. Last but not least, I thank my beloved Luisa, without her patience and love, this work would have never been completed.
To Luisa and to my family I dedicate this thesis.

Munich, December 2018 *Marco De Stefano*

Abstract

The use of a spacecraft equipped with a robotic manipulator is recognised as a promising technology for on-orbit servicing missions. This can extend the lifetime of a defunct satellite or accomplish a safe de-orbiting manoeuvre, thus, mitigating the problem of space debris. Control algorithms for the spacecraft equipped with the manipulator need to be validated on ground prior to the space mission. Hence, a reliable simulator capable of reproducing the micro-gravity conditions on ground is required. To this end, a state-of-the-art robotic facility, namely OOS-Sim was developed at the German Aerospace Center for experimental validation of space robot control algorithms. The facility is composed of two admittance-controlled industrial robots equipped with force-torque sensors to simulate model-based satellite dynamics and one of the industrial robots is equipped with a light-weight robot. In such kind of robotic simulators, delays in the control loop and discretization effects of the signals can lead to an increase in energy, potentially rendering the system unstable.

In this thesis, the aforementioned factors violating the energy conservation principle are identified, isolated and addressed through control strategies to ensure an energy-consistent dynamics simulation while preserving system stability. In particular, energy-observers are designed to monitor the activity in the system and passivity-based controllers are developed to correct the velocity or the force commanded to the robot. This approach is firstly implemented to compensate the unstable effects caused by the time-delays inherent in the control loop. Secondly, an explicit and passive discrete-integrator is designed to prevent the energy drift caused by the integration process of the model-based dynamics, which the robot motion relies on. Finally, a unified framework is designed to compensate both the time-delay and discrete-time integration effects. As a result, the robotic simulator is rendered stable and energy-consistent while simulating satellite dynamics as proved by experiments. Hence, the simulator serves as a reliable platform to validate control algorithms for the space robot.

A further contribution of the thesis is the design and experimental validation of different torque-based controllers for the space manipulator. In particular, a torque-

controller for the manipulator mounted on the non-actuated satellite-base is firstly designed. Later, driven by ESA space mission requirements, which impose a low frequency control of the base actuation, a torque-controller is designed for the regulation of the manipulator and of the actuated base. In particular, stability issues due to the multi-rate controllers of the manipulator and base are analysed from an energy perspective and a passivity-based stabilising controller is proposed. Finally, the control strategy is extended to the tracking case for the manipulator considering the multi-rate effects, which is a typical scenario for a space-robot performing servicing manoeuvres. The controllers are also validated through experiments on the reliable robotic simulator.

SOMMARIO

L'impiego di un satellite equipaggiato con un manipolatore robotico è riconosciuto come una tecnologia promettente per future missioni spaziali inerenti alla manutenzione in orbita di satelliti difettosi. Questa tecnologia può estendere la vita operativa di satellite mal-funzionanti o consentire una manovra di de-orbiting, mitigando l'aumento dei detriti spaziali. Tuttavia, gli algoritmi di controllo richiedono un'accurata validazione prima della missione spaziale e un simulatore affidabile, in grado di riprodurre le condizioni di microgravità a terra, è di fondamentale importanza. In questo contesto, OOS-Sim, una struttura robotica recentemente sviluppata presso il centro aerospaziale tedesco, è impiegata per supportare la validazione sperimentale degli algoritmi di controllo. Il simulatore è composto da due robot industriali controllati in ammettenza che simulano la dinamica di un satellite sfruttando un modello matematico. Entrambi sono dotati di sensori di coppia e di forza e uno dei robot industriali è equipaggiato con un manipolatore leggero. In questo genere di simulatori, il tempo di ritardo nel loop di controllo e la discretizzazione dei segnali causano un incremento di energia rendendo il sistema potenzialmente instabile.

In questa tesi i fattori sopra menzionati, che violano il principio di conservazione dell'energia, sono identificati e gestiti attraverso strategie di controllo che assicurano una simulazione dinamica a consistenza energetica, garantendo stabilità. In particolare, osservatori di energia sono stati sviluppati per monitorare l'attività nel sistema e controllori passivi agiscono per correggere la velocità o la forza comandata al robot. Questo approccio è inizialmente implementato per compensare gli effetti instabili causati dal tempo di ritardo presente nel loop di controllo. Successivamente, un integratore passivo ed esplicito è stato sviluppato per prevenire la deriva di energia causata dall'integrazione delle dinamiche in tempo discreto, sulla quale il moto del robot si basa. Infine, una struttura di controllo unificata è stata progettata per compensare il ritardo nel loop di controllo e la deriva di energia causata dall'integratore. Quindi, la strategia adottata garantisce la stabilità del robot industriale e consente di simulare dinamiche di satelliti assicurando una consistenza energetica come dimostrato da esperimenti. Pertanto, il sistema robotico può es-

sere utilizzato come piattaforma per validare gli algoritmi di controllo anche per il satellite equipaggiato con il manipolatore spaziale.

Un ulteriore contributo della tesi è lo sviluppo e la validazione sperimentale di controllori di coppia per il manipolatore spaziale. In particolare, un primo controllo di impedenza è stato sviluppato per il manipolatore montato sulla base del satellite, la quale non è attuata. Successivamente, un secondo controllore di impedenza è stato progettato per regolare il manipolatore e la sua base considerando i requisiti imposti in progetti ESA inerenti a questo genere di missione spaziali, che impongono un controllo a bassa frequenza per l'attuazione della base. In particolare, i problemi di stabilità dovuti al basso tempo di campionamento del controllore alla base sono analizzati da un punto di vista energetico e viene quindi proposto un controllore basato sulla passività. Inoltre, quest'ultimo viene esteso al caso di inseguimento di traiettorie per il manipolatore considerando gli effetti causati dai diversi tempi di campionamento dei controllori. Questo rappresenta uno scenario tipico per un manipolatore spaziale che esegue un compito di manutenzione in orbita. A supporto della tesi, i controllori sono stati validati tramite esperimenti sul sistema robotico OOS-Sim.

CONTENTS

List of Symbols and Abbreviations

In the thesis, all scalar quantities are described by plain letters (e.g. V_s, f_e). Matrices and vectors are printed in bold (e.g. $\boldsymbol{V_s}$, $\boldsymbol{F_e}$). Total derivatives with respect to time t are abbreviated by dots, for example $\dot{\boldsymbol{x}} = \frac{d}{dt}\boldsymbol{x}$, $\ddot{\boldsymbol{x}} = \frac{d^2}{dt^2}\boldsymbol{x}$. The list contains only the quantities which appear at several places in the thesis or are of prominent importance.

List of Symbols

β	Dissipation coefficient for admittance-based passivity control
α	Dissipation coefficient for impedance-based passivity control
t	Time
E_{obs}	Energy observer
$\boldsymbol{F_e}$	External Force
$\boldsymbol{V_s}$	Velocity of the satellite
$\boldsymbol{M}$	Mass matrix of the rigid body
$\boldsymbol{I}$	Inertia matrix of the rigid body
$\boldsymbol{F_c^u}$	Unified force correction
$\boldsymbol{V_c^u}$	Unified velocity correction
$\boldsymbol{\tau}$	Vector of joint torques
$\boldsymbol{J_m}$	Jacobian matrix of the manipulator
$\boldsymbol{J_b}$	Jacobian matrix of the base
$\boldsymbol{F_m}$	Control force of the manipulator
$\boldsymbol{F_b}$	Control force of the base
$\boldsymbol{q}$	Vector of manipulator joints
$\boldsymbol{H_b}$	Inertial matrix of the whole system

$\boldsymbol{H_m}$	Inertial matrix of the manipulator
$\boldsymbol{C}$	Coriolis/centrifugal matrix
$\boldsymbol{\Delta x_m}$	Error vector for the manipulator
$\boldsymbol{\Delta x_b}$	Error vector for the base
$\boldsymbol{K_{pm}}$	Stiffness matrix of the manipulator control
$\boldsymbol{K_{pb}}$	Stiffness matrix of the base control
$\boldsymbol{K_{dm}}$	Damping matrix of the manipulator control
$\boldsymbol{K_{db}}$	Damping matrix of the base control
$\boldsymbol{\Lambda}$	Cartesian Inertia matrix

List of Abbreviations

OOS	On Orbit Servicing
HIL	Hardware in the Loop
OOS-Sim	On Orbit Servicing Simulator
FTS	Force Torque Sensor
LWR	Light Weight Robot
UDP	User Datagram Protocol
Sat. Dyn.	Satellite Dynamics
TD	Time Delay
dof	Degree of freedom
R	Robot
TDPA	Time Domain Passivity Approach
PO	Passivity Observer
PC	Passivity Controller
FO	Force Observer
min	Minimum
e. g.	*Exempli gratia* (for example)
i.e.	*Id est* (that is)
DEOS	Deutsche Orbitale Servicing Mission (German On-orbit Servicing Mission)
DLR	Deutsches Zentrum für Luft-und-Raumfahrt (German Aerospace Center)

List of Figures

List of Tables

List of Author Publications

The list includes journals and conference articles related to the research contribution of the author in the field of admittance control for multibody dynamics simulation using robotic facilities, impedance control for manipulators mounted on an actuated base, stability analysis. The articles are classified according to the year of publication in an ascending order.

2018:

1. **M. De Stefano**, H. Mishra, R. Balachandran, R. Lampariello, C. Ott, and C. Secchi. "Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy", *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1319–1326, 2019.

2. H. Mishra, **M. De Stefano**, AM. Giordano, and C. Ott. "A nonlinear observer for free-floating target motion using only position measurements", *accepted to IEEE American Control Conference (ACC),* 2019.

3. D. Henry, J. Cieslak, P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaard, H. Strauch, A. Giordano, **M. De Stefano**, C. Ott, M. J. Reiner, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, J. Gil. "Model-based fault diagnosis and tolerant control: the ESA's e.Deorbit mission", *accepted to IEEE European Control Conference (ECC),* 2019.

4. R. Lampariello, H. Mishra, N. Oumer, P. Schmidt, **M. De Stefano**, A. Albu-Schaeffer "Tracking control for the grasping of a tumbling satellite with a free-floating robot", *IEEE Robotics and Automation Letters,* 3 (4):3638-3645, Oct. 2018.

5. **M. De Stefano**, Balachandran, and C. Secchi. "A passivity-based approach for simulating satellite dynamics with robots: Time delay and discretization effects", *submitted to IEEE Transactions on Robotics (T-RO)*, July 2018.

6. S. Jaeckel, R. Lampariello, W. Rackl, **M. De Stefano**, N. Oumer, AM. Giordano, O. Porges, M. Pietras, B. Brunner, J. Ratti, Q. Muehlbauer, M. Thiel,

S. Estable, R. Biesbroek, A. Albu-Schaeffer "Design and Operational Elements of the robotics subsystem for the e.deorbit debris removal mission", *Frontieres in Robotics and AI*, 5:100, 2018.

7. P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaar, H. Strauch, AM. Giordano, **M. De Stefano**, C. Ott, M. Rainer, D. Henry, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, and J. Gil-Fernandez. "Methods and outcomes of the comrade project - design of robust combined control for robotic spacecraft and manipulator in servicing missions", *69th International Astronautical Congress,* Sept. 2018.

8. MJ. Kim, R. Balachandran, **M. De Stefano**, K. Kondak, C. Ott. "Passive compliant control for Aerial manipulation", *2018 IEEE/RSJ International Conference on Intelligent Robot and System (IROS), Madrid, Spain,* pp. 4177-4184, September 2018.

9. H. Mishra, **M. De Stefano** AM. Giordano, C. Ott. "Tracking control with robotic system for a moving traget: A vector Lyapunov approach", *IFAC-PapersOnLine, 51(22):471-478, 2018. 12th IFAC Symposium on Robot Control SYROCO 2018*, Budapest, Hungary, August 2018.

10. **M. De Stefano**, R. Balachandran, A. Giordano, C. Secchi. "An energy-based Approach for the Multi-Rate Control of a Manipulator on an Actuated Base", *2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia*, pp.1072 -1077, May 2018.

11. P.R. Perez, **M. De Stefano**, R. Lampariello. "Velocity Matching Compliant Control for a Space Robot during Capture of a Free-Floating Target", *2018 IEEE Aerospace Conference, Big Sky, Montana, USA,* pp. 1-9, March 2018.

2017:

12. **M. De Stefano**, J. Artigas, C. Secchi. "A passive Integration Strategy for Rendering Rotational Dynamics on a Robotic Simulator", *2017 IEEE/RSJ International Conference on Intelligent Robot and System (IROS), Vancouver, Canada,* pp.2806 -2812, September 2017.

13. **M. De Stefano** and J. Artigas. "The DLR On-Orbit Servicing Simulator: Reproducing free-floating dynamics with Robotic Facilites", Workshop on "Gravity offload testbed for space robotic mission simulation", *2017 IEEE/RSJ International Conference on Intelligent Robot and System (IROS), Vancouver, Canada,* September 2017.

14. J. Telaar, I. Ahrns, S. Estable, W. Rackl, **M. De Stefano**, R. Lampariello, N. Santos, P. Serra, M. Canetri, F. Ankersen, J. Gil-Fernandez. "GNC architecture for the e.Deorbit mission", *7th European Conference for Aeronautics and Space Science (EUCASS),* doi: 10.13009/EUCASS2017-317, Milan, Italy, July 2017.

15. **M. De Stefano**, R. Balachandran, J. Artigas, C. Secchi. "Reproducing Physical Dynamics with Hardware-in-the-loop Simulators: A Passive and Explicit Discrete Integrator", *2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore,* pp. 5899-5906, May 2017.

16. J. Telaar, S. Estable, **M. De Stefano**, W. Rackl, R. Lampariello, F. Ankersen, J. G. Fernandez. "Coupled control of chaser platform and robot arm for the e.deorbit mission", *GNC 2017: 10th International ESA conference on Guidance, Navigation and Control Systems, Salzburg, Austria,* May 2017.

2016:

17. A. Giordano, G. Garofalo, **M. De Stefano**, C. Ott, A. Albu-Schaeffer. "Dynamics and control of a free-floating space robot in presence of nonzero linear and angular momenta", *2016 IEEE Conference on Decision and Control (CDC), Las Vegas, USA,* pp.7527-7534, December 2016.

18. **M. De Stefano**, J. Artigas, C. Secchi. "An optimized passivity-based method for simulating satellite dynamics on a position controlled robot in presence of latencies", *2016 IEEE/RSJ International Conference on Intelligent Robot and System (IROS), Daejeon, Korea,* pp.5419-5426, October 2016.

19. J. Artigas, R. Balachandran, **M. De Stefano**, M. Panzirsch, J. Harder, J. Letschnik, R. Lampariello, A. Albu-Schaeffer, "Teleoperation for On-Orbit Servicing Missions through the ASTRA Geostationary Satellite", *2016 IEEE Aerospace Conference, Big Sky, Montana, USA,* pp. 1-12, March 2016

2015:

20. P. Schmidt, J. Artigas, **M. De Stefano**, R. Balachandran, C. Ott. "Increasing the Performance of Torque-based Visual Servoing by applying Time Domain Passivity", *IFAC-PapersOnLine, 48(19):13-18, 2015. 11th IFAC Symposium on Robot Control SYROCO 2015, Salvador, Brazil,* vol.48 pp. 13-18, August 2015.

21. J. Artigas, **M. De Stefano**, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O.Porges, A. Giordano, C. Borst, and A. Albu-Schaeffe. " The OOS-SIM: An On-ground Simulation Facility For On-Orbit Servicing Robotic Operations", *2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, USA,* pp.2854-2860, May 2015.

22. **M. De Stefano**, J. Artigas, A. Giordano, R. Lampariello and A. Albu-Schäffer. "On-ground experimental verification of a torque controlled free-floating robot", *13th Symposium on Advanced Space Technologies in Robotics and Automation 2015 (ASTRA 2015), ESA/ESTEC Noordwijk, Netherlands,* May 2015

23. **M. De Stefano**, J. Artigas, W. Rackl and A. Albu-Schäffer. "Passivity of Virtual Free-Floating Dynamics Rendered on Robotic Facilities", *2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, USA,* pp.781-788, May 2015.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Space exploration, the vanguard of humanity, has inspired researchers, artists and authors since a long time. Isaac Asimov, Robert Heinlein and Arthur C. Clarke, also known as *the Big Three of science fiction*, have stimulated the imagination of exploiting robots to support the humans in discovering new galaxies. They also conveyed the feeling of *floating* in space considering the difficulties and risks associated with the space environment, such as the collisions between orbiting platforms and satellites. Although the concept of space robotics is introduced through science fiction authors, a statement can be accepted for engineers and researchers: robots can operate in harsh environment, such as the space, where collisions between orbiting objects can generate drastic consequences.

Today, the technological progress requires the service of several satellites, which have improved our life in terms of telecommunication, localization and remote sensing. Statistical data reveal that from the launch of the first satellite, the Sputinik 1 in 1957, more than 100 satellites are launched every year in the last decade and more than 29000 objects greater than 10 centimetre are orbiting the low Earth orbit with an orbital velocity of 7 km/s [FAMPU14]. The objects in orbit include not only satellites, but also space debris, a specific type of space object which is human-made, non-functional and orbiting around the Earth. Examples of space debris are the spent upper stages of launch vehicles, inactive satellites, fragments produced by on-orbit exploration and collisions. The high density of space debris and the high velocity associated with the low orbit, can generate mutual collisions leading to a chain reaction. This cascade effect is defined in literature as *Kessler syndrome* [KCP78]. One of the major examples is the unintentional space debris collisions occurred in 2009 between the defunct Russian satellite, Kosmos 2251, and the operational satellite, Iridium 33. The collision generated more than 200000 fragments

greater than one centimetre and the number of debris around the Earth increased drastically [Lio11]. Furthermore, in recent years on-orbit failures have exceeded the launch failures leading to an increase in space debris and losses of billions of dollars [SA01]. These failures are related to thrusters systems, star sensors, wrong orbit location, problems in the apogee motors and deployment of solar arrays [EKS08]. Besides that, every satellite runs out of fuel and thus, must be decommissioned even if is still functional.

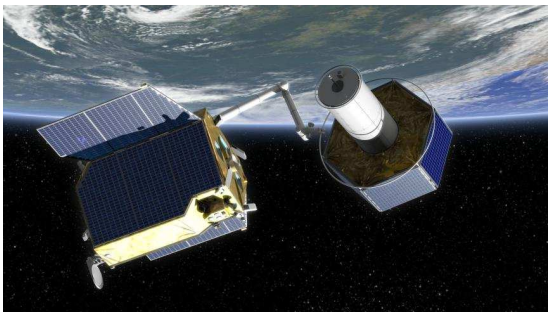
Mitigation of space debris and resolution of satellites failures have driven the main space agencies to define a recovery plan introducing the concept of on-orbit servicing [RMD82]. The term On-Orbit Servicing (OOS) refers to the maintenance in orbit, including assembly, refuelling and repair of a defective system. The goal of an OOS mission is to extend the lifetime of satellites after failures and to actively remove the space debris with a controlled re-entry in the Earth atmosphere. It is notable that such tasks have motivated the development of new technologies in space including unmanned missions. Among all, robotics emerges as a promising technology to perform servicing tasks.

The assembly and maintenance in orbit using robotic technologies began in 1981, when the space Shuttle manipulator, the *Canadarm*, was used for the first time. Several manned missions have followed for the capture and repair of malfunctioning satellites including the repair of the Hubble Telescope. Also the use of the manipulator on the International Space Station (ISS), the *Canadarm 2*, represented a milestone for the support of the ISS extra vehicular activities. In addition to the robotic servicing capabilities, that are bound to the now decommissioned Shuttle or to the Canadarm 2, unmanned mission demonstrators were flown in orbit. The most relevant missions and demonstrations are the following, ROTEX (the Robot Technology experiment) developed by the German Aerospace Center (DLR) [HBDH93], the Japanese Engineering Test Satellite VII (ETS-VII) [OKY96, Yos03], the German Robotic Component Verification experiment aboard the ISS (ROKVISS) [ASBR$^+$06] and the Demonstration of Autonomous Rendezvous Technology (DART) [HJBB04]. In particular, a milestone was the success of DARPA Orbital Express mission in 2007 [SW04]. The goal was the demonstration of autonomous rendezvous and docking operations including maintenance activities like refuelling on a *partially-cooperative* satellite. However, space debris and defective satellites are usually non-cooperative, that means no support with respect to attitude and orbit control of the defunct satellite is available. Further, the satellite does not have a docking port or retro reflectors used for vision based navigation.

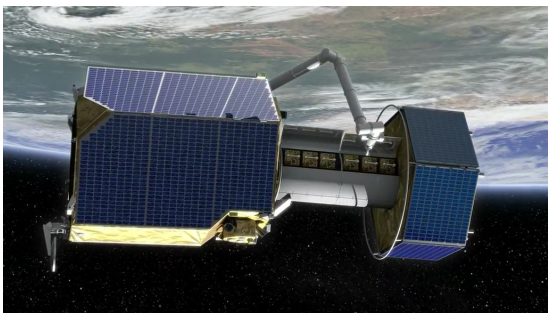
Recently, many projects focus on the capture of a non-cooperative spacecraft. This represents a challenge for a control system due to the lack of information about the states of the target. The German Orbital Servicing Mission (DEOS) [SBS$^+$10, RMNL11] aimed at grasping of a non-cooperative target by using a free-to-float space robot. Several projects followed after DEOS, such as e.Deorbit [TAE$^+$17] and the ongoing project COMRADE [CBS$^+$18], commissioned by the European Space Agency (ESA). The goal of the latter two projects is to perform OOS tasks using a manipulator arm, which operates on a controlled satellite-base. The most

challenging task considered in e.Deorbit and COMRADE, is the removal of Envisat, an eight-ton formerly Earth-monitoring satellite, that is defective and tumbling. It is recognised as the biggest debris in low Earth orbit. The DEOS, e.Deorbit and COMRADE projects consider the use of a three-meter redundant manipulator arm for operating in the multiple phases of an OOS mission, which consist of approaching the non-cooperative target satellite with the manipulation arm, grasping and stabilisation of the target. Later, the docking manoeuvre follows and OOS tasks or a de-orbiting manoeuvre can be performed on the combined structure.

Fig. 1.1 shows the scenario of the DEOS mission, where a servicer satellite, i.e. the satellite equipped with a manipulator arm, approaches a non-cooperative and tumbling satellite, the client (see Fig. 1.1a). After the grasping and stabilisation phases, a docking manoeuvre is performed and the manipulation arm is used for repairing the satellite (see Fig. 1.1b). The scenario of the e.Deorbit and COMRADE projects can be seen in Fig. 1.2a, where the manipulation arm of the servicer satellite grasps Envisat on the launch adapter ring and in Fig. 1.2b a representation of the de-orbiting manoeuvre is shown.

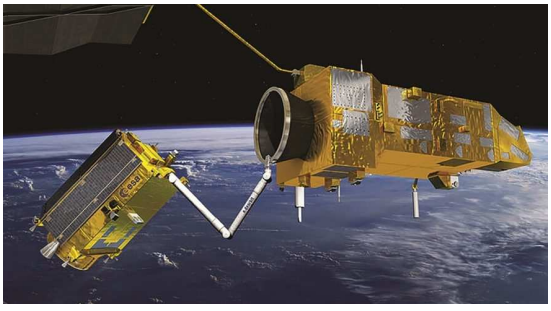


(a) Servicer satellite equipped with a manipulator during the approach phase of a target satellite ©DLR.



(b) Servicer satellite equipped with a manipulator in the docked configuration while performing OOS task ©DLR.

Figure 1.1: Scenarios for the DEOS mission.



(a) Servicer satellite equipped with a manipulator during the grasping of Envisat satellite ©ESA.

(b) Servicer satellite equipped with a manipulator during the de-orbiting of Envisat satellite ©ESA.

Figure 1.2: Scenarios for the e.Deorbit and COMRADE project.

During these phases, the manipulator arm needs to have a degree of compliance in order to react in case of accidental contacts and during the grasping phase. This factor can be achieved by using passive elements such as mechanical springs or with an active compliance. Within this field, the most popular implemented method is the impedance control, where a desired mass-spring-damper behaviour can be achieved for the arm. Commonly, this type of controller is employed in human-robot interaction, where safety and interaction with external force is required. Furthermore, the controller of the space arm needs to take into account also the floating nature of the satellite-base. This induces a challenge from the point of view of robot dynamics and impedance control design, which is different from the common fixed-base robot control. These factors motivate the need of having a platform which can reproduce microgravity conditions on ground in order to develop and validate impedance controllers for the space manipulator.

1.2 On-ground Facilities for Simulation of Microgravity Conditions

A key role for the success of an OOS space mission is played by the validation of the algorithms and controls prior to the launch. Hence, the need of having a reliable simulator, which can reproduce micro-gravity conditions on ground is of utmost importance. Within this context, several technologies can be adopted to recreate zero gravity (0-g) conditions on ground and these can be classified into: air bearing systems, neutral buoyancy, 0-g parabolic flights, cable off-loaders and hardware-in-the-loop simulators [SPH03]. The main capabilities and limitations of these technologies are presented in this section. In particular, a review on the facilities which exploit the simulation with hardware-in-the-loop is presented and details are provided for a particular hardware-in-the-loop simulator, the OOS-Sim, exploited for the validation of the controllers developed within this thesis.

Technologies based on *air bearing systems* include one or two platforms which can float on a flat floor through air bearing pads. Pressurised air passes through small holes of the platform and it establishes a thin film of compressed air with the flat floor that supports the weight of the moving section [SPH03]. This technique allows the simulation of 0-g condition in two dimensional (2-D) space, which includes one rotation and two translations. Such a system has been used for testing the control algorithms of the Japanese free-floating system [Yos03] and major companies and research groups in academia exploit this kind of test-beds [RFS07]. However the motion which can be simulated is limited to planar motion only [MBC+07] and this represents a limitation for a complete on-ground simulation of an OOS mission.

An other technology for 0-g simulation exploits a water pool to achieve *neutral buoyancy*, so that the submerged body has a tendency to float as it would be in space. The method has the advantage of having a 6-D motion within the fluid and without having time constrains [CA00]. However this technology suffers from drag forces induced by the fluid, which generate hydrodynamics effects that distort the

dynamics to be reproduced [FAMPU14].

Zero-g parabolic flights can achieve nearly zero gravity conditions. For this technique, an aircraft flying on a parabolic trajectory is exploited to achieve microgravity conditions [MBC+07]. In [SUM+04] experiments were performed in a zero-g parabolic flight for a four degree-of-freedom robot achieving $0.02g$ for 20 seconds. This method has several drawbacks, two of which are short time duration in microgravity (between 10 and 30 seconds) and limited workspace [FAMPU14].

In systems with *cable off-loaders*, the gravity force is compensated by a suspension system, usually composed of cables, which generate the same force amplitude but in opposite direction of the gravity force vector. The main drawback of this technology is the static balancing of the gravity force. Furthermore, the tensions forces might also be applied in other directions which can significantly interfere with the behaviour of the space element to be tested [FAMPU14].

The disadvantages of the aforementioned technologies can be covered when considering simulation techniques with Hardware-in-the-Loop (HIL).

1.2.1 Hardware-in-the-Loop Simulators

Simulations with hardware-in-the-loop are exploited in different fields, for example in automotive, electronics and dynamics systems [SHH+12]. As the name suggests, this technology involves both hardware and software simulation. The possibility of linking a dynamic model to a real physical system provides benefits in term of costs reduction, safety and testing the operational scenario under defined conditions with real sensors. Regarding the recreation of microgravity simulation conditions on ground, a computer-based dynamic model of the whole space system is considered and a hardware system, capable of delivering a 6-D motion, is used to reproduce the desired behaviour. Usually, the simulator which exploits this technology is a robot(s) equipped with mock-ups of a spacecraft including the necessary hardware for performing the capture or docking task. Furthermore, a force-torque sensor is used for measuring the external interaction. The force and torque signals measured by the sensor represent the input to the model-based dynamics whose output is commanded to the robot that moves in Cartesian space.

This technique of simulation has the advantage of using a software simulation of the desired dynamics, which for the satellite case is relatively simple to model. Furthermore, it benefits from an accurate measurement of the contact forces acting on the system, delivered by a force-torque sensor. Therefore real force measurements are involved without the need of having a contact dynamics model, whose accuracy is usually low [BM11]. However, HIL simulators have some limitations given mainly by the inevitable time delay from a hardware contact to the corresponding simulation driven reaction, which might lead to instability [FAMPU14].

In recent years, such kind of simulators have been employed in the aerospace field and there are several facilities exploiting this technique available in literature. The SPDM Task Verification facility developed by MD Robotics Ltd for the Canadian Space Agency (CSA) is composed of two arms with a total of 15 degrees of

freedom and it is employed for verification of robotic tasks performed on the international space station [MWSL03]. The Lockheed Martin SOCS (Space Operation Simulation Center) facility is composed of two robots with six degree of freedom each capable of simulating a full-scale spacecraft motion relative to an other object. It is used for testing the NASA's manned Orion rendezvous [MD12]. EPOS (European Proximity Operation Simulator) is composed of two industrial robots, which can be used to simulate only rendezvous and docking [BWMT10]. INVERITAS is a robotic facility used for the simulation of rendezvous and capture of satellites [PDG+]. Recently, DLR (German Aerospace Center) developed the OOS-Sim, a Hardware-in-the-loop facility which can be exploited for testing a complete on-orbit servicing task on ground. A detailed description of the facility can be found in the following subsection.

The OOS-Sim Facility

The OOS-Sim is a state-of-the-art robotic facility developed at the DLR and will be used to validate the controllers developed within this thesis. This facility supports the on-ground experimental validation of space robot control algorithms for on-orbit servicing tasks [ADSR+15].

The facility is shown in Fig. 1.3 and comprises one industrial robot equipped with a light-weight robot, namely the servicer robot, and a second industrial robot, namely the client robot. The industrial robots perform the simulation of free floating satellites dynamics while exploiting the HIL technology. The servicer robot, on the left side of Fig. 1.3, is a six- degrees-of-freedom (dof) industrial robot (KR-120), which can carry payloads up to 120Kg. The servicer end-effector mounts a mockup, which integrates a docking interface and OOS-like elements such as a refuelling interface. Furthermore, it is equipped with a force-torque sensor (FTS), which is exploited for measuring external interaction forces and a seven dof Light-Weight-Robot (LWR), which is the manipulator arm used for performing the OOS tasks. The LWR has position and torque interface at joint level and it represents an adequate robot to perform manipulation tasks. The manipulator is also equipped with a three-fingers robotic gripper, used for the grasping of the target. The servicer robot constitutes a hybrid system, where the industrial robot can be controlled only by means of position commands and the LWR can be controller in position and torque modes. A second industrial robot (KR-120) is exploited to simulate the dynamics of a target satellite (see the robot on the right side in Fig. 1.3). Similar to the servicer robot, the client robot has six dof and is equipped with a forces and torques sensor at the end-effector to measured external interaction. The computation of the model-based dynamics, later defined, are performed by a real-time computer, which runs with a period of 4 ms. The LWR manipulation arm is controlled by a second real-time operating system with a sampling time of 1 ms. Both the systems are synchronised to respect real-time determinism[1].

[1]More information about system communication and sensors of the OOS-Sim can be found in Appendix A.1.1.

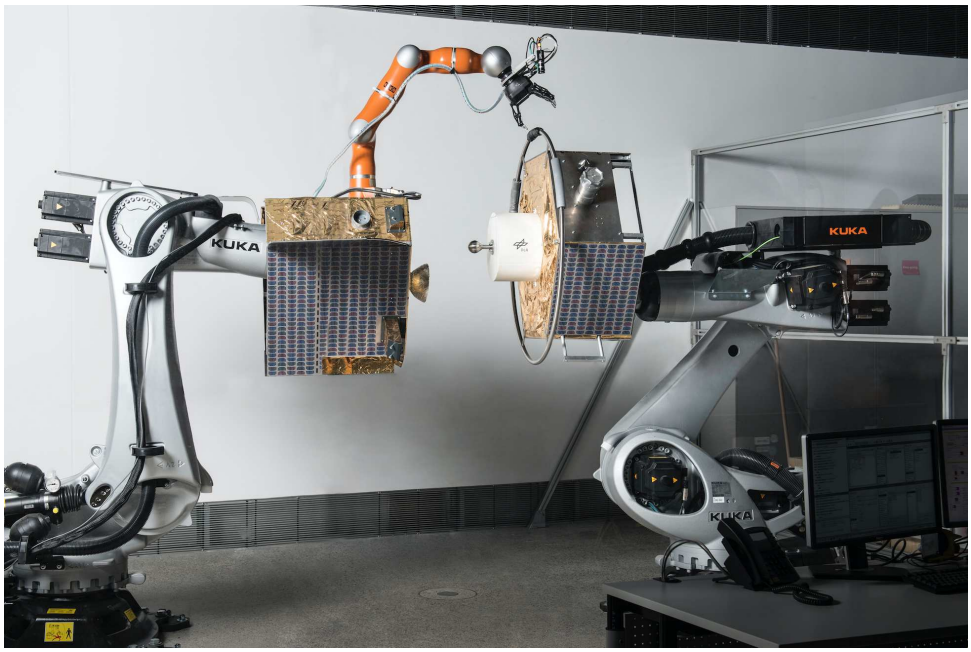


Figure 1.3: The OOS-Sim facility: Servicer robot (left) and Client robot (right).

The facility is designed to operate in a common workspace given by the intersection of the three robots. Reachability and capability maps were used to assess the OOS-Sim workspace. For this, the set-up is treated as two serial kinematic chains, i.e. a 13-dof (KR120 servicer with LWR) and 6-dof (KR120 client). The volume of the space environment which can be simulated with the OOS-Sim is approximately 23.28 m^3 (see Appendix A.1.1) and this workspace fulfils the requirements for testing the main operation modes required for an OOS mission during the approach phase.

The control modes of the OOS-Sim resemble the ones of an OOS mission and these can be divided mainly into semi-autonomy and telepresence modes. During the semi-autonomous control, the manipulator motion follows a reference trajectory provided by a motion planner [LH13]. The trajectory is computed on ground and executed on the servicer manipulator with a local controller that takes into account disturbances and prediction errors [LMO$^+$18]. The second control mode considers an operator in the loop to control the manipulator arm of the servicer satellite, namely this control mode is defined as teleoperation [ARP10]. In this mode, the operator uses a haptic device (for example a Light-Weight-robot) to control remotely the servicer manipulator. Furthermore, force-feedback and vision information are provided to the operator for increasing the feeling of being directly teleoperating on the remote environment. Telepresence for OOS application presents the main challenge of coping with communication channel characteristics, that is, to guarantee a stable system in the presence of time delays [ABDS$^+$16].

A relevant aspect which can be tested and validated on the OOS-Sim is the performance of the impedance control for free-floating robots. Indeed, compliance capability represents a key role in applications where external interaction is involved, such as the grasping of a target satellite. In contrast to classical position control,

Cartesian impedance control offers a more robust interaction. The described control strategies can be tested on the OOS-Sim facility under realistic dynamics and robot control behaviour, as well as with true sensor signals, e.g. using force measured at the industrial robot end-effector.

1.3 Related Works

For a complete simulation of an OOS scenario on ground, the dynamics generated by contact forces between satellites needs to be considered. Contact dynamics is one of the most difficult areas in multi-body systems and it is still an active research subject. The term *contact* describes the mechanical interaction between two bodies whose boundaries share the same location [GS02]. Basically, two approaches are considered for the contact dynamics simulation of a space scenario.

The first approach, exploits a full numerical simulation, where a software is used to simulate the contact and the motion of the spacecraft [Ma95], [Yas14]. This approach is flexible, however it does depend on the contact models, which can be classified into discrete and continuous. In discrete models, the contact occurs in an extremely short period of time and thus all the forces can be regarded as impulsive forces [Ray91]. This implies that the change in energy and momentum will be discontinuous [Bra98]. These models characterise the impact through one or several coefficients, typically the coefficient of restitution is used (i.e., the ratio of the velocity after and before the contact). The most widely used models in this category are Newton's, Poisson's, and Stronge's models [MWX+18]. In continuous models, the interaction forces act in a continuous manner during the contact and it is described by physical forces [Kim99]. An example is the spring-dashpot model, which considers the forces produced by a hypothetical spring and damper which represent the compression-restitution and the energy dissipation. Other models are the Hertz and non-linear damping models. In continuous contact, an accurate model of forces is required and this is typically depending on the local deformation and its derivative [GS02]. A review about the mathematical laws, which describe these models can be found in Appendix B.

Regarding contact dynamics for space applications, [WW93] assumes a discrete model for a point contact scenario. The main difficulties, however, arise when more complex geometries are involved in the contact. [Ma95] and [MBR+97] developed a continuous contact model to support the development of the Canadarm. The computational efficiency, however, resulted to be low for an accurate contact dynamic simulation [MW07]. In [GS02] and [FDDR17] different models of contact are presented and the authors conclude that each model requires the tuning of some parameters, which is difficult to obtain.

The second approach is based on the direct measurement of the contact by using a force-torque sensor. The sensor signals are feed back to the model-based dynamics whose output is sent to the robot, which moves emulating the motion of the spacecraft. However, as mentioned in the Sec. 1.2.1, HIL robotic simula-

tor equipped with force sensor are affected by the time delay in the control loop, which is known to produce instability [DSARAS15]. If the delay value is time invariant, there are several compensation approaches. Usually, the measured force is phase led [SIT+91, OKU10] in order to approximate it to the ideal force. The force compensation adds a virtual force onto the measured force, which can achieve the expected coefficient of restitution [ASJ+14a, ZBC13]. An other approach considers the feed-forward position compensation [ATA96, CCYH07], which compensates the command position of motion to the HIL simulator and makes the current position to the desired position of spacecraft. In both cases, the correction is performed based on a contact model.

Contact dynamics involved during the grasping task of the manipulator plays also an important role [SIT+91]. The manipulation task is usually performed in the operational space or known also as the Cartesian space of the robot end-effector [Kha87]. To deal with the interaction, an impedance control is usually exploited. Compared to the classical method in control, as pole placement [FW67] or backstepping [KKK+95], an impedance controller is more intuitive to parametrise because of the straight relationship of the parameters with the physical meaning. For the impedance control of the space manipulator arm, it must be considered that the manipulation motion is coupled with the floating motion of the satellite-base. A resolved motion-rate control was developed to compensate for spacecraft motion by introducing the concept of Generalized Jacobian [UY87], [UY89]. It was extended in [YTY93] for multi-arm space robots. A sensory feedback control of space robot manipulators was proposed in [MMA89]. In the direction of simultaneously controlling the base and manipulator (known also as coordinated control), the authors in [Yos94] propose a momentum-based strategy using reaction wheels and [PD91] introduces a coordinated controller scheme based on feedback linearisation. A survey of impedance control for space robot can be found in [NP17] and more details are provided in Chapter 6 and Chapter 7.

For the on-ground validation of OOS tasks using HIL robotic set-up, several facilities are available and mainly are related to test the docking or the approach phase between two satellites. For example, [SIT+91] considers a robotic system for simulating 5 dof dynamics of a spacecraft subjected to external forces. The rendezvous and capture of satellites is performed exploiting two industrial robots as in INVERITAS [PDG+] or in [BM11] where the docking between satellites is also performed. [AHLS96] proposes the use of two arms for simulating the servicer and client satellites during approach, however no motion of the satellite-base was considered. In [XLX11] two robots are also used and by combining the dynamic emulation with the kinematic equivalence, it was possible to project the simulated motion of the satellite-base onto the target under the assumption of non-contact. [DDC+95] employed a PUMA robot mounted on a Stewart platform, whose motion was simulated by a model exploiting the HIL technology. In [TIK+08] a hybrid simulation by using dual arm robot including a 6 dof parallel robot was considered. [WFH11] combines air-bearing technology with a HIL robotic simulation.

Further related works will be presented in each chapter for the respective topic.

1.4 Problem Statement

The use of an industrial robot for dynamics simulation represents a challenging task. The high-stiffness of the robot, the discretization of signals and the presence of time-delay between measured data and commands to the robot are factors which can generate instability in the system. Therefore, on one hand the stability of the robot must be guaranteed during the dynamic simulation, on the other a faithful dynamics simulation replicated by the robot needs to be achieved.

A key feature of this kind of simulator is the feed-back loop, which is closed with real sensed force during the contact. As well known, time delay in the loop can affect the stability of the system. Current state-of-works consider the modification of the loop according to a contact model or a longer contact is considered in order to mitigate the effects of the time delay in the control loop. These approaches depend mainly on the contact model parameters, which makes the accurate estimation difficult. Furthermore, existing frameworks cover only sub-domains related to the discrete nature of the signals. However, during the dynamics simulation also a passive discrete-integration technique is required in order to guarantee a stable and energy consistent motion for the robot while simulating the satellite dynamics. Therefore, a novel framework is needed, which should provide:

- a proper control action to avoid instability due to the time delay in the control loop while respecting the *energetic properties* of the dynamics to be simulated,
- *an energy consistent* and explicit integration technique in order to command to the robot a faithful set-point,
- a real-time implementation and experimental validation on an industrial robot.

Once the aforementioned points are fulfilled, control algorithms for the space robot manipulator can be tested on ground. Several publications can be found in robotics literature related to space robot control (see Sec. 1.3). However, a very limited number consider the on-ground validation of space robot. Usually these works are limited only to validate a planar motion of the space manipulator, which often lacks of torque interface. Hence, this also limits the validation of the impedance control for a space robot, which needs to operate in the complete workspace. Impedance capabilities are often required in a situation where a contact is involved due to its robustness. Furthermore, in some cases the control is designed in continuous-time disregarding the effect of the discretization. However, requirements in space projects impose a limitation on the frequency of the controller dictated by the actuation power restrictions, e.g. thrusters. This plays an important role, especially while considering the coordinated control of manipulator and satellite-base. Within this context, the following points need to be fulfilled for the space robot control:

- an on-ground validation of impedance control for space robot is required,
- a coordinated control architecture for the space robot which can deal with low frequencies control is necessary.

1.5 Contribution and Overview

The points outlined in Sec. 1.4 can be addressed using the energy-based framework developed within the thesis. The thesis contributes to both theory and experimental validation in the research field of dynamics simulation using robotic facilities and space robot impedance control. In this section, the contribution of the thesis with an overview of the chapters is presented.

Chapter 2 introduces the reader to the stability problem, which can occur while reproducing a satellite dynamics using a robotic facility. The fundamentals of the passivity-based theory are introduced. Passivity represents one of the main tools exploited in the controls developed within the thesis because it uses the concept of energy, which is fundamental in interaction control. Besides introducing the tools and the theory exploited in the thesis, the contribution of this chapter is the comparison performed with different energy-based methods which can deal with time-delay problems.

Chapter 3 explores the reasons for the increase in energy found in the control-loop of a robot. Firstly, the robotic simulator is designed in electrical domain and the time delay network is identified. Therefore, forces and energy observers are designed together with a passivity controller, which profits from the passivity criteria. A second contribution is the extension of the passivity control considering an optimal criteria to maximise the performance while guaranteeing stability. The proposed control concepts are validated on a real robot.

An other source of activity, which leads to an increase in energy during the dynamic simulation is related to the way how the dynamics is discretized. The discretized dynamics, indeed, generates the set-points in velocities, which *move* the robot in Cartesian space. Therefore, the discrete-time integration needs to operate in a passive way.
Hence, in Chapter 4 a passive and discrete-time integrator is developed, which preserves the energy and dynamic properties of a physical body rendered on a robotic simulator. The contribution is related to the identification of the energy generation that results from the integration process. These sources of energy can lead to a non-physical behaviour of the simulated dynamics, resulting in position drifts or stability issues. The proposed method dissipates this energy using a variable damper regulated by an energy observer. The passive integrator is validated with simulations and tested on a real-time robotic simulator.

In Chapter 5, an overall architecture is designed to deal with both time delay and discretization effects, which are analysed from an energetic perspective and compensated through a passivity-based control strategy. Stability is guaranteed during the dynamic simulation using position-controlled robots. The benefit of the proposed methodology is validated with simulations and experiments.

The last two chapters of the thesis are related to the design and validation of impedance controllers for the space robot. In Chapter 6, a review on the impedance controllers for a space robot is given and an on-ground validation using the common free-floating control is performed. Firstly, the space robot is considered to be free-

to-float, i.e. the satellite-base where the manipulator is connected is not controlled. The free-floating control was the baseline for the DEOS mission. The contribution is related to the extension of the classical impedance controller for the free-flying space robot, i.e. when the satellite-base is actuated by thrusters or reaction wheels. The latter control was proposed also in the e.Deorbit space project.

In Chapter 7, a novel control strategy for a light weight robot arm mounted on an actuated floating base is developed. In particular, the stability issues due to the different rates of the control units are analysed. The first contribution is the design of a non-linear regulation control for the base and the manipulator with stability proof in continuous-time. Then, the feedback loop is modified by using a passivity-based stabilising controller exploiting the time domain passivity approach. The second contribution is the design of a tracking control for a space manipulator and the regulation of its base, which is a typical scenario for an on-orbit servicing mission. Besides guaranteeing trajectory tracking for the manipulator and regulation for the base, the multi-rate nature of the controllers are analysed from an energetic perspective. Therefore, both controllers are equipped with variable damping regulated by energy observers. The effectiveness of the proposed strategies is validated on a base-manipulator multibody simulation and experimentally on the OOS-Sim facility.

Hence, the main contributions of the thesis can be summarised as follows,

- A novel passivity-based control for dealing with time delay in robotic facilities for simulation of dynamics.
- A novel discrete and passive integrator which can run in real time and therefore can be used for dynamics simulation using robotic facility.
- A unified architecture which can deal with time delay and discrete sampling in rendering dynamics.
- On-ground validation of impedance control for free-floating robot and development of a free-flying impedance robot control.
- A novel approach with stability proof for the impedance control of a free-flying space robot under multi-rate effects. In particular, the regulation and tracking are tackled.

All the aforementioned points are supported with theoretical analysis, numerical examples and experimental validations. The research finding reported in this thesis resulted in two journals [DSMB+19, DSBS18] and eight conference papers [DSBG+18, DSA17, DSAS17, DSBAS17, DSAS16, ADSR+15, DSAG+15, DSARAS15] presented in main international robotics congresses. The main publications on which the thesis is based are summarised in Table 1.1. Furthermore, a graphical overview of the chapters including the topics and the relation with the publications, can be found in Fig. 1.4.

Moreover, two journal articles [LMO+18, JLR+18] and eleven conference papers, which are related to the topic of controlling space and aerial manipulators, have been co-authored, see [SADS+15, ABDS+16, GGDS+16, TEDS+17, TAE+17, PSL18, MDSGO18, KBDS+18, CBS+18, MDSGO19, HCC+19]. However these works are

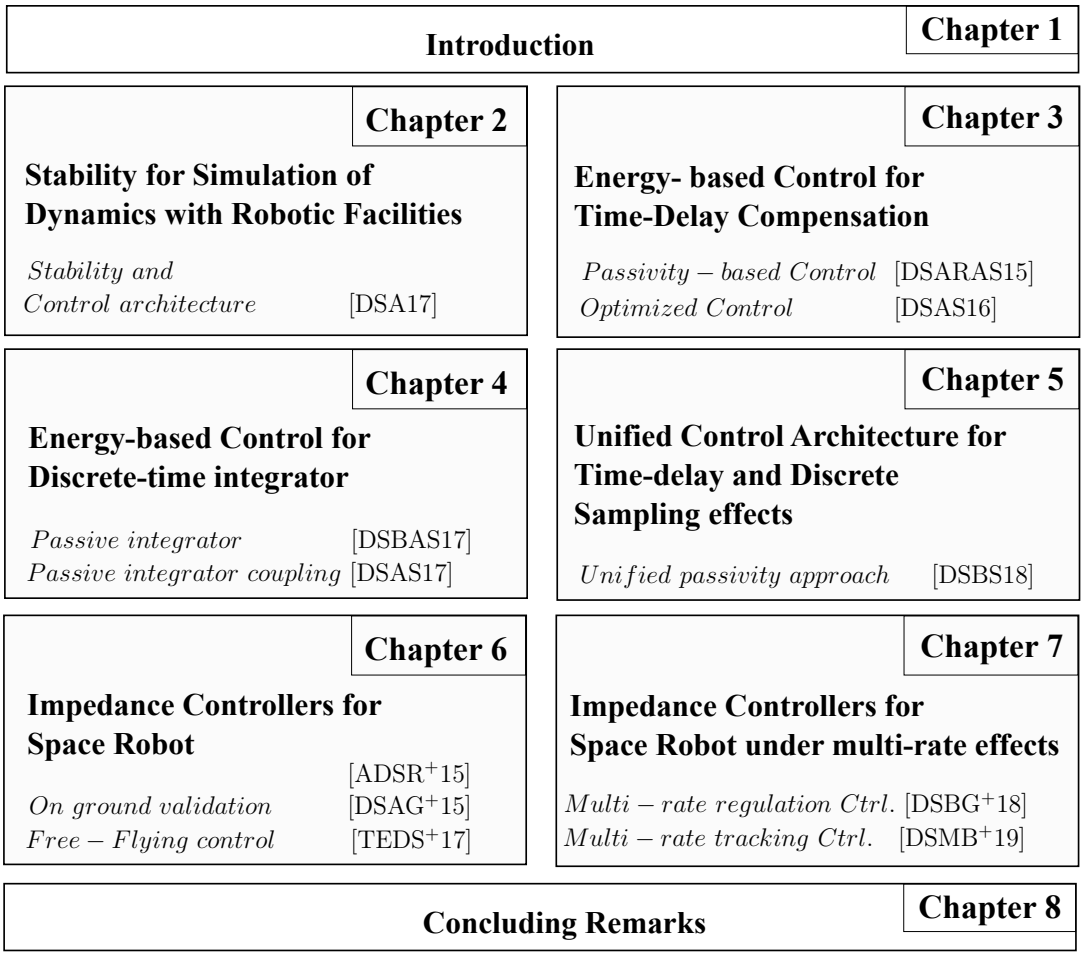


Figure 1.4: General overview of the chapters with main topics and relation to the publications.

not integrated in the thesis. The reader can refer to the *List of Author Publications*, which includes the complete research contribution of the author.

References	***Details***
Journal [DSMB+19]	**M. De Stefano**, H. Mishra, R. Balachandran, R. Lampariello, C. Ott, and C. Secchi. *Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy.* IEEE Robotics and Automation Letters (RA-L), 4(2):1319–1326, 2019.
Journal [DSBS18]	**M. De Stefano**, Balachandran, and C. Secchi. *A passivity-based approach for simulating satellite dynamics with robots: Time delay and discretization effects.* Submitted to IEEE Transactions on Robotics (T-RO), July 2018.
Conference [DSBG+18]	**M. De Stefano**, R. Balachandran, A. Giordano, C. Ott, and C. Secchi. *An energy- based approach for the multi-rate control of a manipulator on an actuated base.* In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1072–1077, May 2018.
Conference [DSAS17]	**M. De Stefano**, J. Artigas, and C. Secchi. *A passive integration strategy for rendering rotational rigid-body dynamics on a robotic simulator.* In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2806–2812, Sept. 2017.
Conference [DSA17]	**M. De Stefano**, J. Artigas. *The DLR On-Orbit Servicing Simulator: Reproducing free-floating dynamics with Robotic Facilites.* Workshop on "Gravity offload testbed for space robotic mission simulation", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2017.
Conference [DSBAS17]	**M. De Stefano**, R. Balachandran, J. Artigas, and C. Secchi. *Reproducing physical dynamics with hardware-in-the-loop simulators: A passive and explicit discrete integrator.* In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 5899–5906, May 2017.
Conference [DSAS16]	**M. De Stefano**, J. Artigas, and C. Secchi. *An optimized passivity-based method for simulating satellite dynamics on a position controlled robot in presence of latencies.* In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5419–5426, Oct. 2016.
Conference [ADSR+15]	J. Artigas, **M. De Stefano**, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O.Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer *The OOS-SIM: An On-ground Simulation Facility For On-Orbit Servicing Robotic Operations.* In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 2854-2860, 2015.
Conference [DSAG+15]	**M. De Stefano**, J. Artigas, A. Giordano, R. Lampariello, and A. Albu-Schaeffer. *On-ground experimental verification of a torque controlled free-floating robot.* In 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA), 2015.
Conference [DSARAS15]	**M. De Stefano**, J. Artigas, W. Rackl, and A. Albu-Schaeffer. *Passivity of virtual free-floating dynamics rendered on robotic facilities.* In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 781–788, 2015.

Table 1.1: Main publications on which this thesis is based. The complete author publication list can be found in *List of Author Publications*.

CHAPTER 2

STABILITY FOR SIMULATION OF DYNAMICS WITH ROBOTIC FACILITIES

2.1 Introduction

A major problem in control theory is to achieve high performance while keeping stability under all the operational conditions. There are basically two philosophies for controlling a robot having non-linear and uncertain behaviours. One is adaptive and the other is robust control [SMO99], [ADDJ91]. In adaptive control, the controller tries to *learn* the uncertain parameters of the system in order to achieve the best performance [SL87, OS88]. On the other hand, robust controllers have fixed structures that guarantee stability and performance in bounded uncertainties [ADDJ91]. Robust controllers include passivity-based control which depends on the passive nature of the rigid robot. Passivity-based controllers have better robustness properties featuring robust stability with parameters variation [RKH04].
The controllers designed within this thesis will be based on the passivity properties of a system. In this chapter, the fundamentals of the passivity properties and a background on energy-based control methods are discussed. Firstly, the chapter introduces the control architecture exploited for simulating model-based dynamics using robotic facility and presents the stability analysis for a reduced model of the simulator. Furthermore, a review on passivity theory and a comparison on the common energy-based methods is introduced. In particular, the time-domain passivity approach, the stabilising tool exploited along the thesis, will be presented in this chapter. Furthermore, the network theory, that is needed for the energy treatment tackled in this thesis, will be also introduced.

2.2 Control Architecture for Simulating Satellite Dynamics

A suitable control strategy for simulation of desired dynamics on industrial robots is the admittance control [SK16]. This strategy has been exploited in many different fields, see e.g. [FPM+15] for a surgical application, [LFS+17] for a human-robot interaction scenario and [ADSR+15] for robotic simulators for space applications. While using admittance controlled robots, several factors must be taken into account in order to guarantee stability. For example, the time delay between measured and commanded data to the robot is one of those that can jeopardise the stability of the system [DSARAS15]. In general, industrial robots can be controlled only by means of position commands. This poses an added difficulty while rendering a virtual body dynamics due to the absence of a torque interface. Therefore, this factor imposes an admittance causality on the controller architecture.

For the OOS-Sim facility, two industrial robots are exploited to simulate the dynamics of a satellite. Both robots, namely Client and Servicer presented in Sec. 1.2.1, are equipped with force-torque sensor at the end-effector. In the following sections, the model-based dynamics simulated by the robots is introduced along with the control architecture.

Client Robot

The representative dynamics of the client satellite, is based on the Newton-Euler equations [Hug04]. The equations for the translational dynamics are defined as follows:

$$\dot{\boldsymbol{v}}_{\boldsymbol{s}} = \boldsymbol{M}^{-1}\boldsymbol{f}_{\boldsymbol{e}}, \tag{2.1}$$

and the rotational dynamics is described by:

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{s}} = \boldsymbol{I}^{-1}(\boldsymbol{I}\boldsymbol{\omega}_{\boldsymbol{s}} \times \boldsymbol{\omega}_{\boldsymbol{s}} + \boldsymbol{\tau}_e) = \boldsymbol{I}^{-1}(\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega}_{\boldsymbol{s}})\boldsymbol{\omega}_{\boldsymbol{s}} + \boldsymbol{\tau}_e), \tag{2.2}$$

where the following variables are defined as:

- $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ the simulated mass matrix,
- $\boldsymbol{f_e} \in \mathbb{R}^3$ the Cartesian measured force,
- $\boldsymbol{v_s} \in \mathbb{R}^3$ the Cartesian linear velocity,
- $\dot{\boldsymbol{v}}_{\boldsymbol{s}} \in \mathbb{R}^3$ the Cartesian linear acceleration,
- $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ the inertia matrix of the body about the center of mass,
- $\boldsymbol{\omega_s} \in \mathbb{R}^3$ the angular velocity,
- $\dot{\boldsymbol{\omega}}_{\boldsymbol{s}} \in \mathbb{R}^3$ the angular acceleration,
- $\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega}_s) \in \mathbb{R}^{3\times3}$ the skew-symmetric matrix such that $\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega}_s)\boldsymbol{\omega}_s = \boldsymbol{I}\boldsymbol{\omega}_s \times \boldsymbol{\omega}_s$,
- $\boldsymbol{\tau_e} \in \mathbb{R}^3$ the Cartesian measured torques.

The total wrench and twist are defined as:

- $\boldsymbol{F_e} = (\boldsymbol{f_e}, \boldsymbol{\tau_e}) \in \mathbb{R}^6$ the wrench,

- $\boldsymbol{V_s} = (\boldsymbol{v_s}, \boldsymbol{\omega_s}) \in \mathbb{R}^6$ the twist.

The dynamics in (2.1) and (2.2) is a function of external forces and torques $\boldsymbol{F_e}$, which can be measured by the force-torque sensor (FTS) placed at the end effector of the client robot. Thus, forces and torques measured by the end effector sensor, i.e. $(\boldsymbol{f_{FS}}, \boldsymbol{\tau_{FS}})$ need to be transformed in the satellite center of mass, as:

$$\boldsymbol{f_e} = R_{EE,e}\ \boldsymbol{f_{FS}}, \tag{2.3}$$

$$\boldsymbol{\tau_e} = \boldsymbol{p}_{EE,e} \times (R_{EE,e}\ \boldsymbol{f_{FS}}) + R_{EE,e}\ \boldsymbol{\tau_{FS}}, \tag{2.4}$$

where $R_{EE,e}$ is the rotation matrix between the end-effector and the center of mass and $\boldsymbol{p}_{EE,e}$ is the position vector from the end-effector to the center of mass.

The architecture implemented for controlling the client robot is shown in Fig. 2.1. The input to the simulated dynamics ($Sat.Dyn.$) is the measured forces and torques vector $\boldsymbol{F_e}$. Thus, the satellite dynamics can be computed according to the external physical interaction. The block L represents the integral process and the transformation required to compute a desired homogeneous matrix H_d commanded to the robot through the inverse kinematics. In particular, by double integrating (2.1) a relative position can be defined and by integration (2.2) and using the Euler-Rodrigues formulation [SK08], a relative orientation matrix can be found. As a result, the robot moves in Cartesian space simulating the satellite motion.

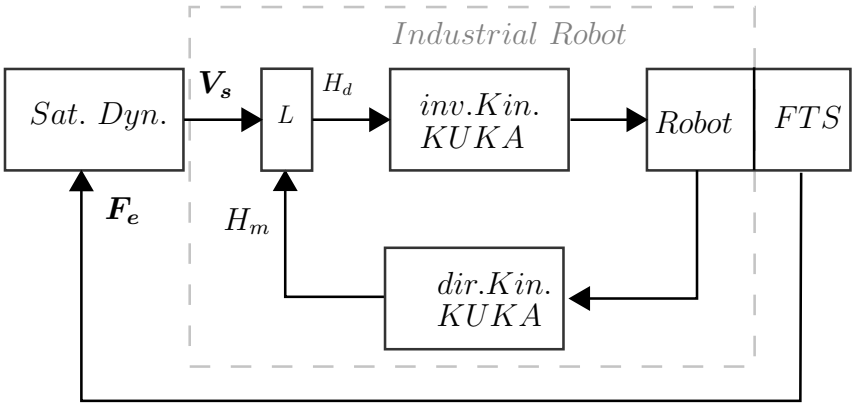


Figure 2.1: Control architecture of the industrial robot for reproducing satellite dynamics.

Servicer robot

The servicer robot is composed of an industrial robot similar to the client, but it is equipped with a 7 dof Light Weight Robot (LWR) mounted on its end-effector. The industrial robot, which simulates the servicer satellite, can be controlled only in admittance mode with the same architecture shown in Fig. 2.1 and the LWR can be controlled in torque or position mode. The industrial robot can replicate the satellite dynamics in a decoupled manner, i.e. considering Fig. 2.1 and using the

force-torque sensor at the LWR base to measure its reaction force or in a coupled manner. For the latter, the dynamic model can be expressed as follows [Fea07]:

$$\begin{bmatrix} \boldsymbol{H_b} & \boldsymbol{H_{bm}} \\ \boldsymbol{H_{bm}^T} & \boldsymbol{H_m} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C_b} & \boldsymbol{C_{bm}} \\ \boldsymbol{C_{mb}} & \boldsymbol{C_m} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_b} \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J_b^T} \\ \boldsymbol{J_m^T} \end{bmatrix} \boldsymbol{F_h}, \quad (2.5)$$

where the following are defined as:

- $\boldsymbol{H_b} \in \mathbb{R}^{6\times6}$ inertia matrices of the whole system,
- $\boldsymbol{H_m} \in \mathbb{R}^{n\times n}$ inertia matrices of manipulator,
- $\boldsymbol{H_{bm}} \in \mathbb{R}^{6\times n}$ coupling matrix between the base and the manipulator,
- $\boldsymbol{C_b} \in \mathbb{R}^{6\times6}$ non-linear Coriolis/centrifugal matrix of the base,
- $\boldsymbol{C_{bm}} \in \mathbb{R}^{6\times n}$ non-linear Coriolis/centrifugal matrix due to coupling between base and manipulator,
- $\ddot{\boldsymbol{x}}_{\boldsymbol{b}} \in \mathbb{R}^6$ acceleration vector of the base (linear and angular),
- $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ acceleration vector of the robot joints,
- $\boldsymbol{F_b} \in \mathbb{R}^6$ force-torque wrench acting on the center of mass of the base-body,
- $\boldsymbol{\tau} \in \mathbb{R}^n$ input torque vector to the manipulator,
- $\boldsymbol{F_h} \in \mathbb{R}^6$ external force-torque on the end-points,
- $\boldsymbol{J_b} \in \mathbb{R}^{6\times6}$ Jacobian of the base,
- $\boldsymbol{J_m} \in \mathbb{R}^{6\times n}$ Jacobian of the manipulator.

The motion of the servicer base can be either obtained from the integration of the upper set of (2.5) or by integration of the total momentum of the system [ADSR+15], defined as follows:

$$\dot{\boldsymbol{x}}_{\boldsymbol{b}} = -\boldsymbol{H}_b^{-1}(\boldsymbol{H_{bm}}\dot{\boldsymbol{q}} + \int_0^t \boldsymbol{F_b}dt + \int_0^t \boldsymbol{J_b^T}\boldsymbol{F_h}dt). \quad (2.6)$$

The commanded input to the industrial robot is the velocity $\dot{\boldsymbol{x}}_{\boldsymbol{b}}$ in (2.6) and the control architecture is similar to the control scheme shown Fig. 2.1, where *Sat. Dyn* receives as input the total forces, the joint state of the LWR and gives as output $\dot{\boldsymbol{x}}_{\boldsymbol{b}}$.

2.3 Stability Analysis

In this section the stability analysis is performed for a reduced model of the system shown in Fig. 2.1. In particular the main focus is on the effect of the time delay in the loop caused by the inverse kinematics computation, the actuation of the industrial robot and the transmission of signals. Therefore, in the following analysis the dashed box of Fig. 2.1, indicated as *Industrial Robot*, will be replaced with a time delay block TD. The simplification is not restrictive because the industrial robot can perfectly track the desired velocity, however the resulting velocity will be affected by some time delay. It will be analytically shown which is the critical time delay that affects the stability of the closed-loop system.

The reduced model, which represents the robotic simulator for satellite dynamics simulation, is shown in Fig. 2.2. The satellite dynamics, *Sat. Dyn*, is represented by

the integration of (2.1), which can be initialised with an initial velocity $V_s(t_0)$. The time delay block, TD, is located in the control loop and it represents the robot, which delays the velocity signal. Therefore, the velocity V_s is delayed by a quantity T_d, which is the time delay in continuous time. The block E represents the environment with which the robot can interact producing a force, F_e, as input to the *Sat. Dyn.* The environment is modelled as a spring-dashpot model, see Appendix B.

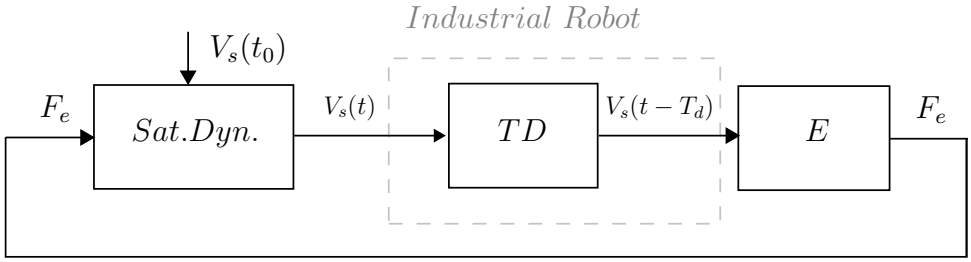


Figure 2.2: Reduced model of the robot simulator with time delay (TD) in the loop.

The corresponding transfer functions of the system shown in Fig. 2.2 are defined as follows,

$$G(s)_{sat} = \frac{1}{ms^2}, \tag{2.7}$$

$$G(s)_{TD} = e^{-sT_d}, \tag{2.8}$$

$$G(s)_E = K_w + D_w s, \tag{2.9}$$

where $G(s)_{sat}, G(s)_{TD}, G(s)_E$ are the transfer functions of the satellite dynamics, time delay and environment, respectively. The positive constants, K_w and D_w, are the stiffness and damping of the environment and m is the mass of the rigid-body. The stability boundaries can be exploited for guaranteeing stability of a system, as it has been performed in literature for a similar case (see [OKU10], [ASJ+14b], [QZG+16]). An analytical solution for the considered system was proposed in [ZBC14] and it is reported here in explicit form. Later, a first-order Pade' approximation for the time delay transfer function is considered. This allows one to define a polynomial transfer function and to exploit classical stability tools, such as root locus.

Analytical solution

The characteristic polynomial of the system, obtained from the total transfer function between the *Sat. Dyn* input to the output, results to be,

$$\Upsilon_1(s) = ms^2 + e^{-sT_d}(D_w s + K_w). \tag{2.10}$$

The stability of the system is analysed by studying the behaviour of the roots, as the time delay T_d increases from zero value. The number of roots become infinitive and some will cross the imaginary axis for a critical time-delay value, i.e T_{d_0}. The

condition for $\Upsilon_1(s)$ to have the roots on $j\omega$ is expressed considering $\Upsilon_1(j\omega) = 0$, which results in the following set of equations,

$$\left|\frac{D_w s + K_w}{ms^2}\right| = 1 \tag{2.11}$$

$$arg\Big[\frac{D_w s + K_w}{ms^2}\Big] = -\omega T_d \pm 2\pi n, \tag{2.12}$$

where $n = (0, 1, 2, ...)$. Equations (2.11) and (2.12) remind the gain-phase margins relation for a rational function [FPW98] and from these equations the following can be found,

$$m^2\omega^4 - D_w^2\omega^2 - K_w^2 = 0 \tag{2.13}$$

$$\omega T_d = arctan\Big(\frac{\omega D_w}{K_w}\Big) \pm \frac{2\pi}{\omega} n. \tag{2.14}$$

By selecting the positive root for ω, the final system of equations result in:

$$\omega = \sqrt{\frac{{D_w}^2}{2m^2} + \sqrt{\frac{D_w^4}{4m^4} + \frac{K_w^2}{m^2}}} \tag{2.15}$$

$$T_{d_n} = \frac{1}{\omega} arctan\Big(\frac{\omega D_w}{K_w}\Big) \pm \frac{2\pi}{\omega} n. \tag{2.16}$$

Equations (2.15) and (2.16) provide the analytical solution of the frequency and the time delay for which the closed-loop system becomes unstable. The critical value for the time-delay can be found by setting the first value (n=0) in (2.16).

A limit case is represented when $D_w = 0$, indeed, the critical delay calculated in (2.16) results to be zero and the associate crossing frequency results in the expected value of $\omega = \sqrt{K_w/m}$. In other words, an elastic impact with time-delay in the loop destabilises the system. Furthermore, high value of the stiffness K_w affects also the stability in presence of delay and the critical time delay $T_d \to 0$ (see (2.16)).

Solution via Pade' approximation

In order to exploit classical stability tools, such as root locus, a polynomial transfer function is required and the stability can be analysed while modifying independently the parameters, m, T_d, K_w, D_w. Therefore, a first-order Pade' approximation [FPEN09] can be considered for the transfer function of the time delay and it is reported below,

$$G(s)_{TD} = e^{-sT_d} \sim \frac{2 - sT_d}{2 + sT_d}. \tag{2.17}$$

In this case, the total transfer function between the input to the *Sat. Dyn* and the output (including the Pade' approximation) results to be,

$$H(s) = \frac{-s^2 D_w K_w + s(2D_w - T_d K_w) + 2K_w}{s^3 m T_d + s^2(2m - T_d D_w) + s(2D_w - T_d K_w) + 2K_w}. \tag{2.18}$$

To evaluate the stability, the characteristic polynomial Υ_2 from (2.18) is considered,

$$\Upsilon_2 = s^3 m T_d + s^2(2m - T_d D_w) + s(2D_w - T_d F_w) + 2K_w. \tag{2.19}$$

Therefore, the *Routh-Hurwitz* criteria [JHea92] can be applied to evaluate stability properties while exploiting the well know Def. 1.

Definition 1 *A time-invariant system is stable if all the roots of the characteristic polynomial have real part* $\Re_e < 0$.

The *Routh-Hurwitz* matrix R_H is composed of coefficients calculated from the characteristic polynomial as defined in [FPEN09]. For the considered case, the *Routh-Hurwitz* matrix results to be:

$$R_H = \begin{pmatrix} a_{\text{n}} & a_{\text{n-2}} \\ a_{\text{n-1}} & a_{\text{n-3}} \\ b_{\text{n-1}} & b_{\text{n-2}} \\ c_{\text{n-2}} & c_{\text{n-3}} \end{pmatrix}, \tag{2.20}$$

where:

$$\begin{aligned} a_{\text{n}} &= mT_d \\ a_{\text{n-2}} &= 2D_w - T_d K_w \\ a_{\text{n-1}} &= 2m - D_w T_d \\ a_{\text{n-3}} &= 2K_w \\ b_{\text{n-1}} &= -\frac{2mK_w T_d - 4mD_w + 2{D_w}^2 T_d + 2mT_d K_w - {T_d}^2 K_w D_w}{2m - T_d D_w} \\ b_{\text{n-2}} &= 0 \\ c_{\text{n-2}} &= 2D_w - T_d K_w \\ c_{\text{n-3}} &= 0 \end{aligned} \tag{2.21}$$

As well known, stability conditions can be found by imposing the positiveness of the coefficients of the first column, i.e. $(a_n,\, a_{n-1},\, b_{n-1},\, c_{n-2}) > 0$. Considering that $a_n = mT_d$, this coefficient is always greater than zero (being $m > 0$ and $T_d > 0$). Therefore, the remaining conditions can be found as follows,

$$m > \frac{D_w T_d}{2}, \tag{2.22}$$

$$D_w(4m - 2D_w T_d + T_d^2 K_w) - 4mK_w T_d > 0 \tag{2.23}$$

$$D_w > \frac{T_d K_w}{2} \tag{2.24}$$

The analysis provides, therefore, three conditions to fulfil for achieving stability and these are function of the parameters m, T_d, K_w, D_w. Equations (2.22) and (2.24) establish the conditions on the minimum mass and the minimum damping. The condition given in (2.23) imposes an active constraint which can be exploited for

finding the critical time delay. In particular, (2.23) can be rewritten as a quadratic function and results in

$$T_d^2 - 2\Big(\frac{D_w}{K_w} + \frac{2m}{D_w}\Big) + \frac{4m}{K_w} > 0 \tag{2.25}$$

Therefore, equation (2.25) can be solved to find the critical delay value, which results in

$$T_{d_c} = \Big(\frac{D_w}{K_w} + \frac{2m}{D_w}\Big) - \sqrt{\Big(\frac{D_w}{K_w} + \frac{2m}{D_w}\Big)^2 - \frac{4m}{K_w}}. \tag{2.26}$$

It is worth to note that (2.26) represents the critical delay calculated with the approximated solution and it will results similar to (2.16), as it will be shown in the following example.

Example 1 *Considering a system with the following parameters:* $m = 100kg$, $K_w = 1000\ N/m$; $D_w = 60\ Ns/m$; $T_d = 50\ ms$, *which respect the condition given in* (2.22), (2.23) *and* (2.24).

As expected, the resulting poles for the Example 1 are $p_1 = -39.3027$, $p_2 = -0.0486 + 3.1898i$, $p_3 = -0.0486 - 3.1898i$ and accordingly with Def. 1, the system results to be stable. Furthermore, the critical time delay for which the system might go unstable and the respective frequency are calculated for the analytical solution in (2.16) and the approximated solution in (2.26). The results are almost identical, as can be seen from the values reported in Table 2.1.

	Critical Time Delay [ms]	Cross Frequency [rad/s]
Analytical Solution	59.3	3.1909
Approx. Solution	59.5	3.19

Table 2.1: Critical delay comparison between analytical and approximated solution.

The stability boundaries can be exploited for guaranteeing stability of a system, however, it is clear the dependency from the contact parameters. In case of non-perfect estimation of the contact parameters (K_w and D_w) the system might behave differently. This factor can be seen in Fig. 2.3, which considers the same Example 1, but with $D_w = 0.0001Ns/m$. The analysis shows the presence of two positive poles, which cross the Real axis. This factor is in contrast with Def. 1 and instability in the close-loop system occurs.

Discussion

As it was shown, the time delay in the closed loop might cause system instability. Although, stability boundaries can be exploited for guaranteeing the stability and perform theoretical analysis, the approach requires the knowledge of the contact parameters (K_w and D_w) and the time delay T_d, which for some applications is

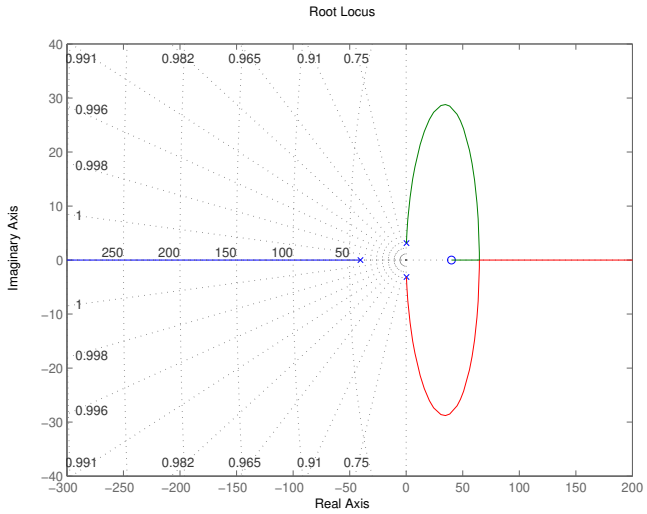


Figure 2.3: Root locus plot of Fig. 2.2 for elastic impact. Poles are $p1 = -40.4909, p_2 = 0.2455 + 3.1334i, p_3 = 0.2455 - 3.1334i \rightarrow$ Unstable system.

not constant. For a complete model-based simulation, these parameters are known directly from the implemented contact model. For a HIL simulation, these parameters depend from the environment, which is not always known. Further, no contact model is exploited in HIL because real forces are involved and measured by a sensor. The dependency of contact parameters might limit the range of the simulation which can be performed with the robotic simulator.

An other important factor for the stability of a HIL simulator, is the delay associated with the hold, typical issue in digital control [FPW98]. As it was shown in the analysis above, a delay in the system degrades the stability in continuous time. The continuous time delay, T_d can be seen equivalently in discrete time as $T_d = kT$, where k is the discrete-time step of sampling time T. For discrete systems, a generic value $u(t)$ consists in $u(kT)$ that, on average, lag by a quantity $T/2$, see [FPW98]. This quantity translates into a phase decrease of

$$\delta\phi_h - \frac{\omega_h T}{2}. \tag{2.27}$$

Thus, the loss of margin can be estimated by invoking (2.27), with ω_h being equal to the frequency where the magnitude is equal to one, which is the gain cross frequency. Discrete sampling effects were also analysed in [BC98b] and it was shown that the physical damping in the system must be sufficient to dissipate the excess of energy created by errors introduced by sampling of the discrete-time controller, commonly referred as *energy leaks*.

To summarise, the time-delay and discrete signals affect the stability margins [DNBS06]. Although the theoretical treatment discussed in this section is valid, a more general and suitable solution, which does not depend on contact parameters, must be sought. In the thesis, these stability issues will be tackled considering passivity-based approaches. Indeed, the approach is based on input/output characteristic of a system and does not rely on any contact model parameters.

2.4 Passivity Theory

This section introduces the concept of passivity which is exploited in the design of the controllers developed within this thesis.

Passivity and Stability

The concept of passivity is linked to the dissipative properties of a system [OPNSR98], which is close to the intuitive phenomena of loss or dissipation of energy. A passive system can not store more energy than the one supplied to it, with the difference being the dissipated energy [SSF07]. This factor plays a central role in dynamic system where energy with the environment is exchanged.

Considering a generic non-linear system, S, represented in Fig. 2.4 with input $\boldsymbol{u} \in \mathbb{R}^n$ and output $\boldsymbol{y} \in \mathbb{R}^n$. The system S is defined by the state space $\boldsymbol{x} \in \mathbb{R}^n$ as,

$$S : \begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \end{cases} \tag{2.28}$$

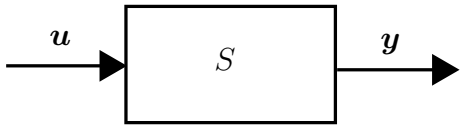


Figure 2.4: System with input $\boldsymbol{u}$ and output $\boldsymbol{y}$ variables.

Definition 2 *(**Passivity**) The system S is passive with respect to the supply rate $\boldsymbol{u}^T\boldsymbol{y}$ if and only if there exists a lower bounded function of the state (storage function) $V : \mathbb{R}^n \to \mathbb{R}$, such that at any time T,*

$$\dot{V} \leq \boldsymbol{u}^T\boldsymbol{y} \quad \Longleftrightarrow \quad V(\boldsymbol{x}(T)) \leq V(\boldsymbol{x}(0)) + \int_0^T \boldsymbol{u}^T(t)\boldsymbol{y}(t)dt. \tag{2.29}$$

The passivity property given in (2.29) relies on the energy and the the physical properties of the system. In particular, the passivity condition can be interpreted as follows:

$$\underbrace{V(\boldsymbol{x}(T))}_{current\ energy} \leq \underbrace{V(\boldsymbol{x}(0))}_{initial\ energy} + \underbrace{\int_0^T \boldsymbol{u}^T(t)\boldsymbol{y}(t)dt}_{external\ supplied\ energy}$$

To correlate the passivity with stability, it is important to introduce the following definitions.

Definition 3 *(**Output Strictly Passive (OSP)**) The system S is Output Strictly Passive if it is dissipative with supply rate such that*

$$\dot{V} \leq \boldsymbol{u}^T\boldsymbol{y} - \beta\boldsymbol{y}^T\boldsymbol{y}, \quad with \quad \beta > 0. \tag{2.30}$$

Definition 4 ***(Input Strictly Passive (ISP))*** *The system S is Input Strictly Passive if it is dissipative with supply rate such that*

$$\dot{V} \leq \boldsymbol{u^T y} - \alpha \boldsymbol{u^T u}, \quad with \quad \alpha > 0. \tag{2.31}$$

The most important feature of passivity is the close relation with the stability of dynamical system. In particular, passivity is related to $\mathcal{L}_2$-stability[1] through OSP and the following preposition holds true:

Proposition 1 ***(OSP implies $\mathcal{L}_2$-stability)*** *If $S : \boldsymbol{u} \rightarrow \boldsymbol{y}$ is Output Strictly Passive, then it is $\mathcal{L}_2$-stable. A system is defined to be $\mathcal{L}_2$-stable, if there exists a positive constant γ, such that for every initial condition $\boldsymbol{x_0}$, there exists a finite constant $\delta(x_0) > 0$, such that*

$$\|y\| \leq \gamma \|u\| + \delta. \tag{2.32}$$

The reader can find the proof of Proposition 1 in [OPNSR98, §A *Proposition A.5*].

Moreover, the benefit of using passivity is also given by the properties achieved when different passive systems are interconnected. This feature will be used in some parts of the thesis, especially when different controllers are connected to a common dynamic system. The properties are reported in the coming subsections.

Passivity Preservation for Interconnections

An important property of passivity is its preservation in term of interconnections between systems. Considering two system S_1 and S_2 represented in state space as,

$$S_1 : \begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{f}(\boldsymbol{x}_1) + \boldsymbol{g}(\boldsymbol{x}_1)\boldsymbol{u}_1 \\ \boldsymbol{y}_1 = \boldsymbol{h}_1(\boldsymbol{x}_1) \end{cases} \quad S_2 : \begin{cases} \dot{\boldsymbol{x}}_2 = \boldsymbol{f}(\boldsymbol{x}_2) + \boldsymbol{g}(\boldsymbol{x}_2)\boldsymbol{u}_2 \\ \boldsymbol{y}_2 = \boldsymbol{h}_2(\boldsymbol{x}_2) \end{cases} \tag{2.33}$$

which are passive and therefore from Def. 2 results that $\dot{V}_{S1} \leq \boldsymbol{u_1^T y_1}$ and $\dot{V}_{S2} \leq \boldsymbol{u_2^T y_2}$. The systems can be interconnected with a power preserving interconnection.

Definition 5 ***(Power Preserving interconnection)*** *An interconnection is power preserving if it guarantees that the energy is neither produced nor dissipated but only transferred between the connected systems.*

For example, the passive systems S_1 and S_2, can be interconnected through the inputs defined as

$$\begin{cases} \boldsymbol{u_1} = -\boldsymbol{y_2} \\ \boldsymbol{u_2} = \boldsymbol{y_1} \end{cases} \tag{2.34}$$

which will result in the following energy exchange:

$$\boldsymbol{u_1^T y_1} = -\boldsymbol{y_2^T u_2}. \tag{2.35}$$

[1] Sometimes this type of stability is referred as *strong* or stability with finite gain.

Therefore, the energy is transferred between the two systems and according to Def. 5 the interconnection is power preserving.

A more general feedback interconnection for the system (2.33) is represented in Fig. 2.5, where $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ are the new inputs for the interaction with the interconnected system.

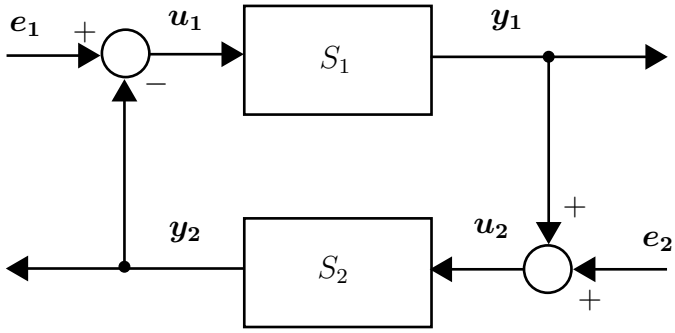


Figure 2.5: Interconnection of passive system S_1 and S_2.

Proposition 2 ***(Interconnection of passive systems)*** *Considering two passive systems* (2.33) *interconnected by a power preserving interconnection as in Fig. 2.5, with*

$$\begin{cases} \boldsymbol{u_1} = -\boldsymbol{y_2} + \boldsymbol{e_1} \\ \boldsymbol{u_2} = \boldsymbol{y_1} + \boldsymbol{e_2} \end{cases} \tag{2.36}$$

then the resulting interconnected system is passive with respect to the pair input-output:

$$\Big(\underbrace{\begin{pmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_2} \end{pmatrix}}_{input}, \underbrace{\begin{pmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \end{pmatrix}}_{output} \Big) \tag{2.37}$$

and the total storage function is $V = V_{S1} + V_{S2}$.

Proof *Considering the passivity of the two systems, the total variation of energy is given as* $\dot{V} = \dot{V}_{S1} + \dot{V}_{S2} \leq \boldsymbol{u_1^T y_1} + \boldsymbol{u_2^T y_2}$. *For the given interconnection in* (2.36)*, it will result,*

$$\dot{V} \leq \boldsymbol{u_1^T y_1} + \boldsymbol{u_2^T y_2} = -\boldsymbol{y_2^T y_1} + \boldsymbol{e_1^T y_1} + \boldsymbol{y_2^T y_1} + \boldsymbol{e_2^T y_2}. \tag{2.38}$$

Then, it follows $\dot{V} \leq \boldsymbol{e_1^T y_1} + \boldsymbol{e_2^T y_2}$ *and according to Def. 2 it results to be a passive system with respect to the new input* $\boldsymbol{e_1, e_2}$ *and output* $\boldsymbol{y_1, y_2}$.

To summarise this section, the passivity property represents the energetic behaviour of a dynamical system and it is related to stability ($\mathcal{L}_2$). This factor is important especially in interaction control, where a physical behaviour depends on the exchange of internal and external energy, as for the case of a robot simulating a satellite dynamics. Passivity can be also easily linked to interconnection of passive

systems, which result in a passive system. Therefore, the use of a passivity-based framework is useful, when energy considerations are taken into account for a complex system.

These fundamental properties motivate the choice of passivity as the basic concept on which the controllers are developed within this thesis.

2.5 Power Port and Network Modelling

The interconnection of passive systems represents a key role for the controllers developed within the thesis. In this section the basic tools to represent a mechanical system are explained in order to analyse its energetic behaviour. The tools are borrowed from the electrical domain.

In each physical domain there is a pair of dual variables, called power conjugate variables, whose duality product is power. These variables are abstractly defined as flow and effort.

Definition 6 ***(Power Port)*** *A power-port is represented by a pair of effort and flow variables, e.g.* $(\boldsymbol{u}, \boldsymbol{y})$*, whose product* $\boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{y}$ *is the power traversing the port.*

Power ports is a practical representation for interconnecting complex systems and together with the passivity theory, they form a powerful tool [ARPH11]. The power port for the system in (2.28) is shown in Fig. 2.6 (left) with input and output variable $\boldsymbol{u}, \boldsymbol{y}$. The power ports for the system interconnected as in (2.33) are shown in Fig. 2.6 (right) with input and output variables $\boldsymbol{e_1}, \boldsymbol{e_2}$ and $\boldsymbol{y_1}$,$\boldsymbol{y_2}$, respectively.

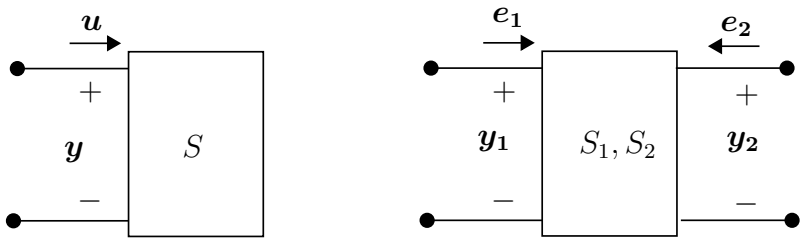


Figure 2.6: One-port with power variables $(\boldsymbol{u}, \boldsymbol{y})$ (left) and two-port with power variables $(\boldsymbol{e_1}, \boldsymbol{y_1})$ and $(\boldsymbol{e_2}, \boldsymbol{y_2})$ (right).

In particular, the system on the left of Fig. 2.6 is defined as *one-port* and the one on the right as *two-ports* and the product of the variables at each port results to be the power. The power ports can be interconnected between each other which will result in a port-network.

Definition 7 ***(Power port-network)*** *A network is composed of power ports which can interact with external connectivity through one or more power-ports.*

When a complex system is considered, it might be not easy to identify the respective power ports [Art14]. A suitable approach is to exploit the electrical and

mechanical analogy in order to build a network representation of the system. This facilitates the identification of the power ports and the energy analysis.

The conventional mechanical-electrical analogy, namely velocity-current analogy, maps forces into voltages and velocities into currents. A mechanical model can be represented in electrical domain according to the analogy reported in Table 2.2.

Mechanical	**Electrical**
$Force \quad [N]$	$Voltage \quad [V]$
$Velocity \quad [m/s]$	$Current \quad [A]$
$Friction \quad [Ns/m]$	$Resistance \quad [ohms]$
$Stiffness \ [N/m]$	$1/Capacitance \ [farad^{-1}]$
$Mass \quad [Kg]$	$Inductance \quad [H]$
$Power \quad [W]$	$Power \quad [W]$

Table 2.2: Mechanical and electrical analogies.

The representation in electrical domain is an intermediate step between the common control block scheme used in control theory and the port-network representation. It is worth to point out that even while using the different domains, the product of the power port variables is a power.

Here, a basic example is presented in order to introduce to the reader the electrical-mechanical analogy and to build the relative port-network. For this, a mass-spring-damper system is considered and it is shown in Fig. 2.7. The mechanical model in Fig. 2.7 (left) is converted in the electrical domain (right) as per the analogy described in Table 2.2. The force applied by the spring, kx results to be proportional to the displacement x of the mass, m. The viscous damping force, ie. $-b\dot{x}$ is proportional to the velocity of the mass, $\dot{x}$ and F is an external force. The total force F_N acting on the mass-spring-damper system is given by:

$$F_N = F(t) - (b\dot{x} + kx) = m\ddot{x} \tag{2.39}$$

The mechanical model can be represented in the electrical domain where the effort V correspond to the force F, the charge (integral of current i) corresponds to displacement, x, the inductance L corresponds to mass m, the resistance R corresponds to

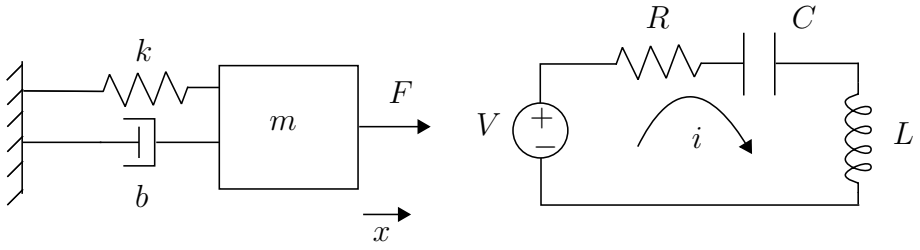


Figure 2.7: Mass-spring-damper system and relative electrical circuit.

viscous damping b, and the inverse capacitance C corresponds to the spring stiffness, k. Therefore the circuit on the right side of Fig. 2.7 is a single loop and the current flowing is the same throughout the circuit at any given time, $i(t)$. Therefore, from Kirchhoff's voltage law,

$$V(t) - Ri - \frac{1}{C}\int idt = L\frac{di}{dt}. \tag{2.40}$$

Note that the equation in the electrical domain has the same form of the mass-spring-damper system. In particular, they are both second-order systems and the state equation of the mechanical system can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F(t), \tag{2.41}$$

and for the corresponding RLC circuit as:

$$\begin{bmatrix} i \\ \frac{di}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} q \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V(t), \tag{2.42}$$

where q is the charge given by $q = \int idt$. These analogies and others like them turn out to be quite useful conceptually in understanding the behaviour of dynamical systems.

Finally, the network representation can be derived from the electrical circuit and it is shown in Fig. 2.8. The network is composed of three power ports:

- N_1 with power correlated variable $(F, \dot{x})$, which is a one-port network
- N_2 with power correlated variable $(F, \dot{x})$ and $(F_N, \dot{x})$, which is a two-port network
- N_3 containing the inductance L, with power $(F_N, \dot{x})$, which is one-port network

Therefore, once the power ports are defined, energy consideration related to the passivity treatment discussed in Sec. 2.4 can be derived. In this thesis, the passivity theory and the power-port network are exploited for applying methods based on energy observation.

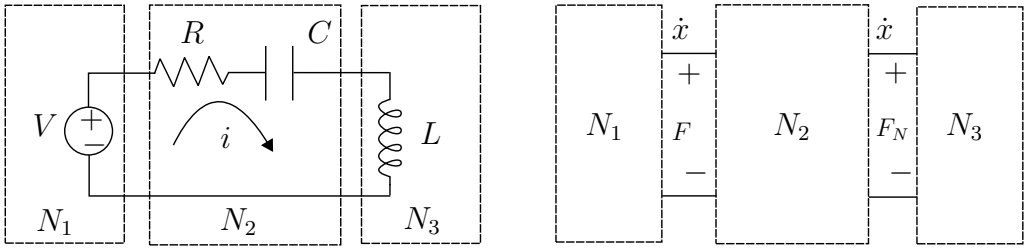


Figure 2.8: Network representation of the relative electrical circuit for the mass-spring-damper system.

2.6 Background on Energy-based Methods

In Sec. 2.3, it was shown that the stability of a robotic simulator can be affected by the time delay between measured and commanded data. The passivity theory (Sec. 2.4) provides information on the global energetic behaviour of a system, in particular in relation to the power variables which can be derived by using the treatment presented in Sec. 2.5.

The field of teleoperation has exploited approaches based on passivity to deal with instabilities due to time delay in communication channels [HS06]. To deal with that, the most common approaches implemented are [BAMR16]: wave variables, port-Hamiltonian and energy tanks, energy bounding and the time domain passivity approach. In this section the aforementioned approaches are summarised and a numerical comparison is proposed for the reduced model of the robotic simulator.

Scattering Parameters and Wave Variables

Scattering and wave variables were discussed in [Nie04], [AS89] and extended for teleoperation in [NS98]. Scattering parameters are used in a system affected by time delay and they are a linear combination of voltage and current displayed at the network ports whose resulting product is power. [AS89] shows that the passivity of a network can be checked using the quadratic norm of its scattering operator S with the condition $|S|^2 \leq 1$. It was proved that, if a network contains a delay, no matter how small, the passivity condition can not be fulfilled [AS89] and a suitable control action is therefore required.

The wave variables transformation contains information related to the energy exchange that occurs on two sides of a system. In particular, the method is based on a simplified transformation of the standard variables (V_s, F_e) to the wave variable (u,v), where u is the forward wave, from master to environment, and v the returning wave, from the environment to master [Nie96].

For the considered scenario, it is possible to augment the model shown in Fig. 2.2 with the wave variable transformation and it is reported in Fig. 2.9. Note that a low-pass filter has been introduced in Fig. 2.9. Indeed, high-frequency noise characterises the wave variable domain, as described in [NS97] and a common approach is given by using a filter to improve the performance. The phase lag that the filter might

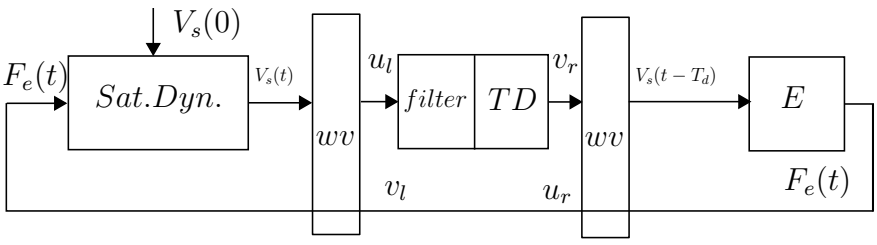


Figure 2.9: Reduced model of the robot simulator with time delay (TD) in the loop and wave variable transformation.

cause is considered in the wave variable transformation layer and therefore does not influence the stability. The wave variable layers (wv) introduced in the scheme compute the wave characteristics defined as follows:

$$\begin{aligned} u_{\mathrm{l}} &= \frac{bV_s(t) + F_l(t)}{\sqrt{2b}} \\ v_{\mathrm{l}} &= \frac{F_l(t) - bV_s(t)}{\sqrt{2b}} \\ u_{\mathrm{r}} &= \frac{F_e(t) - bV_s(t - T_d)}{\sqrt{2b}} \\ v_{\mathrm{r}} &= \frac{bV_s(t - T_d) + F_e(t)}{\sqrt{2b}} \end{aligned} \tag{2.43}$$

where u_l and u_r are the input waves of the respective port and v_l and v_r are the output waves. The parameter $b \in \mathbb{R}^+$ in (2.43) is the characteristic impedance, which is always positive and represents a critical role in determining the system response because it represents a trade off between the velocity and the force.

Note that the set of equation given in (2.43) is commonly used in wave variable formulation. However, for the analysed scenario the time delay is only in the forward wave and not in the backwards. Therefore it is possible to set $F_l(t) = F_e(t)$ in (2.43).

Although the passivity of a system endowed with wave variable can be proved to be stable for a constant time delay [Nie96], this technique introduces a characteristic impedance which might deteriorate the desired behaviour [SFF16]. Drawbacks in applying this method are due to non-idealities in the communication channel like time-varying delays and package loss which might lead to position and force mismatch, see [SAF+09]. Furthermore, wave variables are affected by energy reflection which needs to be matched resulting in an additional damping in the system.

Port-Hamiltonian and Energy Tanks

The port-Hamiltonian framework is a generalisation of the standard Hamiltonian mechanics, where the energy exchange between subsystems are clearly identified. All physical systems can be represented using the port-Hamiltonian formalism [FPM+15]. In interaction control, this framework allows the use of an energy tank (see [SSF06] and [FSM+11]) to implement a desired control action. The tank is an element designed for storing the dissipated energy and it allows one to utilise the (virtual) energy circulating in the controlled system in a flexible way while preserving passivity. A generic port-Hamiltonian system endowed with an energy tank can be described by the following set of equations:

$$\begin{cases} \dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + g(x)u \\ \dot{x_{\mathrm{t}}} = \frac{\sigma D(x)}{x_{\mathrm{t}}} + \frac{(\sigma P_{\mathrm{in}} - P_{\mathrm{out}})}{x_{\mathrm{t}}} + u_{\mathrm{t}} \\ y_1 = (y, y_{\mathrm{t}})^{\mathrm{T}} \end{cases} \tag{2.44}$$

where $x \in \mathbb{R}^n$ is the state vector and $H(x) : \mathbb{R}^n \longrightarrow \in \mathbb{R}$ is the lower bounded Hamiltonian function representing the amount of energy stored in the system. $J(x) =$

$-J(x)^T$ and $R(x) \geq 0$ are the internal energetic interconnections and the dissipation of the port-Hamiltonian system, respectively and $g(x)$ is the input matrix [FPM+15]. The input u and the output y are power correlated variables whose product is the power. The values P_{in} and P_{out} are the incoming and outgoing power flows of the tank, respectively. The value $x_\mathrm{t} \in \mathbb{R}$ is the state variable of the tank $T(x_\mathrm{t})$ defined as follows

$$T(x_\mathrm{t}) = \frac{1}{2}{x_\mathrm{t}}^2. \tag{2.45}$$

The benefit of this design is that energy can be injected or extracted from the tank via a power port $(u_\mathrm{t}, y_\mathrm{t})$. However, some quota of energy must always be present in the tank to avoid singularities in the solutions. Therefore a threshold $\varepsilon > 0$ should be posed. It represents the minimum amount of energy that needs to be always stored. Then, a parameter $\sigma \in \{0, 1\}$ is used for bounding the amount of energy that can be stored in the tank and it is set as follows:

$$\begin{cases} \sigma = 1 \;\; if \;\; T(x_\mathrm{t}) \leq \bar{T} \\ \sigma = 0 \;\; otherwise, \end{cases}$$

where $\bar{T}$ is an upper bound energy value that can be stored in the tank. The energy stored can be used for implementing a desired input $w \in \mathbb{R}^n$ to the port-Hamiltonian system. This can be done through the following power preserving port interconnection:

$$\begin{cases} u = w\frac{y_\mathrm{t}}{x_\mathrm{t}} = w\frac{x_\mathrm{t}}{x_\mathrm{t}} = w \\ u_\mathrm{t} = -w^\mathrm{T}\frac{y}{x_\mathrm{t}}. \end{cases}$$

Note that the energy extracted from the port-Hamiltonian system is exactly equal to the energy supplied to the tank and vice-versa. This means that no energy is generated and passivity is preserved as long as some energy is stored in the tank.

A generic system as the robotic simulator with time delay in the loop can be represented using the port-Hamiltonian framework endowed with energy tank. Therefore, it is possible to augment the dynamic model (given in (2.1)) with an energy storing element, the tank, whose role is to store the energy dissipated by the controlled system. Formally, the augmented dynamics is given by the following,

$$\begin{cases} M\dot{v}_s - wx_\mathrm{t} = F_\mathrm{e} \\ \dot{x}_t = -w^T v_s \\ y = ({v_s}^T, x_\mathrm{t})^T \end{cases} \tag{2.46}$$

The term x_t is the state associated with the tank and the corresponding stored energy is given in (2.45). The method based on energy tank has the flexibility of choosing any control input [SSF07]. The vector w is the control input through which it is possible to control the exchange of energy between the main admittance model (the satellite dynamics) and the tank. The control input w for the system (2.46) is chosen to have a mass-spring-damper behaviour and it is defined as,

$$w(t) = \begin{cases} -\frac{1}{x_t}(K\tilde{x} + D\dot{\tilde{x}}) & T(x_t) > \varepsilon \\ 0 & else, \end{cases} \tag{2.47}$$

where $K \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{n\times n}$ are the stiffness and the damping matrix, respectively. The vector $\dot{\tilde{x}}$ is the variation of velocity between the non-delayed values and the delayed one expressed as, $\dot{\tilde{x}} = v_s(t) - v_s(t - T_d)$. Therefore, the vector $\tilde{x}$ is the corresponding error in position. The value $\varepsilon \in \mathbb{R}^+$ in (2.47) represents the minimum value of the energy in the tank needed to implement the control input and to avoid singularities in the solution. The tank needs to be initialised in a way that $T(x_\mathrm{t}(0)) \geq \varepsilon$ and the energy extraction is not allowed if $T(x_\mathrm{t}) \leq \varepsilon$. Thus, it is necessary to set an upper bound on the amount of energy that can be stored in the tank.

Note that the exchange of energy between the tank and the mechanical system is power preserving, i.e. the energy injected (or extracted) in the tank is the same as the one extracted (or injected) by the system, see proof in [FPM+15].

Energy Bounding Algorithms

The Energy Bounding Algorithm (EBA) was originally presented in [KR04] and extended to the telemanipulation case in [SKK+08], [SKKR09]. The method is based on a viscous friction model, which is exploited to dissipate the virtual energy due to the communication channel. However, the stability of the system is affected by the deviation of the friction model with respect to the real friction in the device. Thus a lower bound of the friction parameters needs to be set and a physical coefficient must be estimated to guarantee passivity.

Time Domain Passivity Approach

A different solution to deal with the instability due to time delay was proposed in [HR01, HR02] where the Time Domain Passivity Control (TDPA) was introduced. The approach is based on the energetic characteristic of the system and a passivity observer is designed in order to monitor the energy at the power-port input. When the passivity condition (reported in Def. 2) is violated, it means that some energy is introduced into the system and a passivity controller is triggered to dissipate it. The passivity controller is a variable damping, which can act in admittance or impedance causality in order to ensure passivity of the system. It can be defined as

$$\beta(k) = -\frac{V_o(k)}{u(k)^T u(k) T}, \tag{2.48}$$

where V_o is the energy observed, u is the input (e.g. the force F_e) and T the sampling time. More details about the passivity observer and passivity controller will be provided in Sec. 2.7.

A potential limitation of TDPA, can be related to the over dissipation which might saturate the actuators. However, a limit on the damping injection can be set and the resulting excess of energy will be stored in the energy observer, which will be dissipated in the next time-step.

2.6.1 Comparison Between the Energy-based Methods

A numerical comparison between the energy-based methods is proposed for the specific case of the reduced model of the robotic simulator shown in Fig. 2.2. The aim is to evaluate the performance of each method in term of energy. In the analysis, the energy bounding algorithms is not considered because of the dependency of a friction model which is not easy to identify. In literature, solutions consider to have a local damping [SKKR09] in case of inaccurate model. However, for the considered case, this will affect the simulation of the free-floating dynamics rendered by the robot. Therefore the analysis is restricted to TDPA, wave variable and energy tank.

A number of 50 simulations is considered for a simulated mass of $100Kg$ which is subjected to elastic forces during its motion. The time delay in the loop changes randomly between 5 and 50 ms in order to prove the robustness of the methods against time delay variations. The comparison allows one to understand how each method operates in comparison to the ideal case (no time delay in the loop). In order to evaluate the performance between the methods, the mean value for the velocity $\dot{\bar{x}}$ and the energy $\bar{E}$ is calculated for each method with respect to the ideal velocity and energy. Fig. 2.10 shows the mean value of energy calculated for all the methods during 60 s of simulation. The ideal case represents the condition without time delay in the control loop and its respective mean values results: $E_{ideal} = 0.435\ J$. The time delay in the loop leads to an increase in energy with respect to the ideal case, which results to be $E_{time\ delay} = 0.576\ \pm 0.0543\ J$. The three energy based methods are applied and the respective mean values of energy are as follows: $E_{TDPA} = 0.435\ J$, $E_{energy\ tank} = 0.433\ J$, $E_{wave\ variable} = 0.441\ J$. The corresponding values of standard deviation for the velocity and energy are reported in Table 2.3.

For the considered application, the wave variable method results less robust against the time delay variation, as shown in Fig. 2.10 where the energy value increases with respect to the ideal case. This factor is due to the fixed value of the impedance characteristic, b, whose tuning is not robust with respect the variation of time delay.

Better performance can be achieved by using the energy tank or the TDPA approach. However, for the energy tank, a suitable control action needs to be chosen, for example in this case an impedance action composed of stiffness and damping.

	Robustness			
	Mean Value		Standard deviation	
	$\dot{\bar{x}}\ [m/s]$	$\bar{E}\ [J]$	$\sigma_{\dot{x}}$	σ_E
Ideal	-0.0071	0.435	0	0
Tdpa	-0.0071	0.435	0.00001	0.0002
Energy Tank	-0.0070	0.433	0.00004	0.0003
Wave Variable	-0.0079	0.441	0.0029	0.0076

Table 2.3: Robustness analysis for the considered energy-based methods.

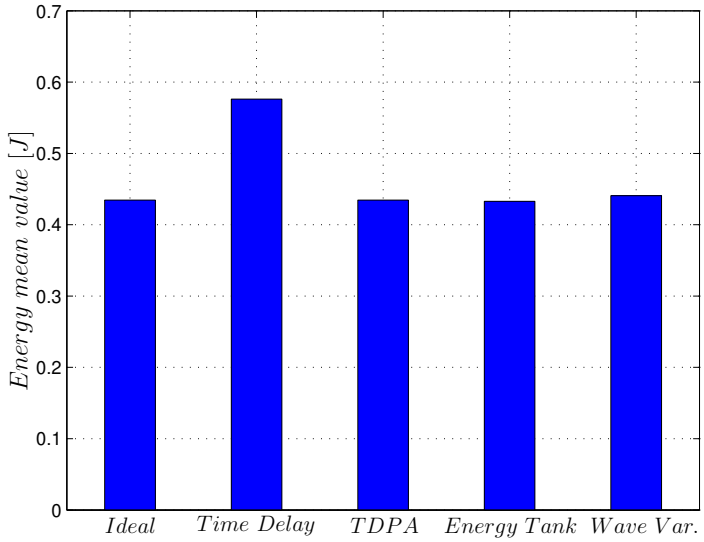


Figure 2.10: Numerical comparison between the energy-based methods for 50 simulation runs with random time delay between 5ms and 50ms: $E_{ideal} = 0.435\ J$, $E_{time\ delay} = 0.576\ J$, $E_{TDPA} = 0.435\ J$, $E_{energy\ tank} = 0.433\ J$, $E_{wave\ variable} = 0.441\ J$.

Furthermore, the minimum and maximum value of the energy stored in the tank, needs also to be set. Then the method relies on some tuning of the controller and can not be generalised to all the operative conditions of the robot simulator.

The TDPA method has the advantage that no tuning of parameters is required because the passivity controller dissipates, through a variable damping, the exact amount of active energy monitored by the passivity observer. This could be an important advantage when the simulation parameters variate, e.g. time delay and mass. Table 2.4 summarises the parameters needed for each of the method and the corresponding control action.

	TDPA	**Energy Tank**	**Wave Variable**
Parameters set.	none	K, D, x_t, ϵ	b, filter characteristic
Control Action	damping	any	wave variable transformation

Table 2.4: Difference between the passive control methods.

The independence from parameters setting and the presence of an energy observer, which can be easily linked with the passivity theory, makes the TDPA a flexible method. This method can be exploited for solving problems related to the time-delay in discrete sampled system, which is one of the goal of this thesis. Therefore, the TDPA-based method will be exploited in the thesis as a tool to enforce passivity and more details are provided in the following section.

2.7 Enforcing Passivity Through Time Domain Passivity Approach

The Time Domain Passivity Approach is an effective control strategy for enforcing passivity in one-port or two-port networks affected by *energy leaks*. The method proposes the use of a Passivity Observer (PO) to monitor the energy of a port network and a Passivity Controller (PC), which is added to the network to compensate for the extra energy. The approach exploits the network representation, which was introduced in Sec. 2.5 and the passivity theory discussed in Sec. 2.4. In this section it will be shown how the method can be applied to one-port and two-port networks, which will be considered in the thesis. The conjugate variables that define the power ports are discrete-time values and the method assumes that no change in values occurs between the discrete sampling time T [RKH04].

Considering the systems in Fig. 2.11, where the one-port network has power correlated variables force and velocity expressed as $(\boldsymbol{f_1(k)}, \boldsymbol{v_1(k)})$ and the two-port network has power correlated variables $(\boldsymbol{f_1(k)}, \boldsymbol{v_1(k)})$ and $(\boldsymbol{f_2(k)}, \boldsymbol{v_2(k)})$, respectively to each port.

The passivity condition for power ports is given as follows [HR02, RHPH03],

Definition 8 ***(Passivity condition for one-port)*** *The one port-network endowed with a power port* $(\boldsymbol{f_1(k)}, \boldsymbol{v_1(k)}) \in \mathbb{R}^n \times \mathbb{R}^n$ *by means of which it can interact (or be interconnected) with external systems is passive if there is a lower bounded energy function* $E(m)$ *such that:*

$$E(m) = E(0) + \sum_{k=0}^{m} \boldsymbol{f_1^T(k)} \boldsymbol{v_1(k)} T \quad \geq \quad 0, \forall m > 0 \tag{2.49}$$

where $E(0)$ *represents the initial energy stored in the system.*

Definition 9 ***(Passivity condition for two-port)*** *The two-port network endowed with a two-power ports* $(\boldsymbol{f_1(k)}, \boldsymbol{v_1(k)}) \in \mathbb{R}^n \times \mathbb{R}^n$ $(\boldsymbol{f_2(k)}, \boldsymbol{v_2(k)}) \in \mathbb{R}^n \times \mathbb{R}^n$ *is passive if there is a lower bounded energy function* $E(m)$ *such that:*

$$E(m) = E(0) + \sum_{k=0}^{m} \boldsymbol{f_1^T(k)} \boldsymbol{v_1(k)} T + \sum_{k=0}^{m} \boldsymbol{f_2^T(k)} \boldsymbol{v_2(k)} T \quad \geq \quad 0, \forall m > 0 \tag{2.50}$$

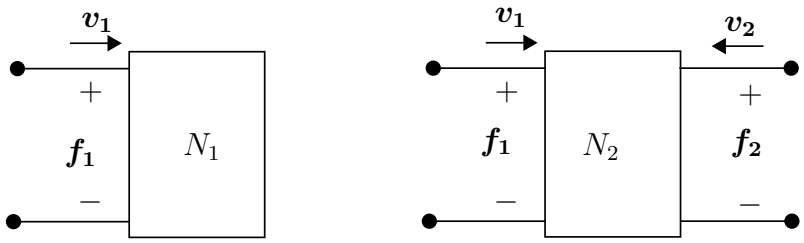


Figure 2.11: One-port network (left) with power variables $(\boldsymbol{f_1}, \boldsymbol{v_1})$ and two-port network with power variables $(\boldsymbol{f_1}, \boldsymbol{v_1})$,$(\boldsymbol{f_2}, \boldsymbol{v_2})$.

where $E(0)$ represents the initial energy stored in the system.

Loosely speaking, (2.49) (or the similar (2.50)) simply states that the energy supplied to a passive network must be greater than negative $E(0)$ for all the time [HR02], [AH99]. To check the passivity properties, a *monitoring block* can be included in the system, which is formally defined as Passivity Observer (PO) [HR01]. For one-port network it is defined as:

$$E_{obsN1} = E(0) + \sum_{k=0}^{m} \boldsymbol{f_1^T(k) v_1(k)} T \tag{2.51}$$

and similarly for the two-port, as

$$E_{obsN2} = E(0) + \sum_{k=0}^{m} \boldsymbol{f_1^T(k) v_1(k)} T + \sum_{k=0}^{m} \boldsymbol{f_2^T(k) v_2(k)} T. \tag{2.52}$$

Independently of the ports configuration (one or two ports), the PO monitors the energy flow. When $E_{obsNi}(k) > 0$ for $i = 1$ *or* 2, at each time step k, it means that the system dissipates energy. On the other hand, when $E_{obsNi}(k) < 0$, this means that the system generates energy and the amount of generated energy is $-E_{obsNi}(k)$, which is defined to be *active energy* [Art14] and it may contribute to instability.

Moreover, the exact amount of active energy is known and a time-varying element can be designed to dissipate it. This element is called Passivity Control (PC). The PC takes the form of a dissipative element in a series or parallel configuration and it will restore the energy balance guaranteeing the passivity of the port. Therefore, if this port is connected with other passive ports, the final system results to be an interconnection of passive system, which is passive (as discussed in Sec. 2.4).

2.7.1 Passivity Controller

The configuration of the PC depends on the input/output causality of the model underlining the port. The action of the PC can be considered in admittance or impedance causality. The description is provided first for one-port network and later for the case of two-ports.

Admittance causality

In admittance causality the velocity through the port is modified in order to provide a dissipative action and the force across the port is conserved. In the admittance causality, a time-varying damping, namely β is placed in parallel to the power-port as shown in Fig. 2.12 (left). The equivalent electrical element for the varying damping is a variable resistor represented in Fig. 2.12 (right).

From the schematic it is easy to see that $\boldsymbol{v_1'}$ is the velocity correction given to the network N. Therefore, considering the initial energy to be zero, the Passivity Observer (PO) is designed as:

$$E_{obsv}(m) = \sum_{k=0}^{m} \boldsymbol{f_1^T(k) v_1'(k)} T + \sum_{k=0}^{m-1} \beta(k) \boldsymbol{f_1(k)^T f_1(k)}. \tag{2.53}$$

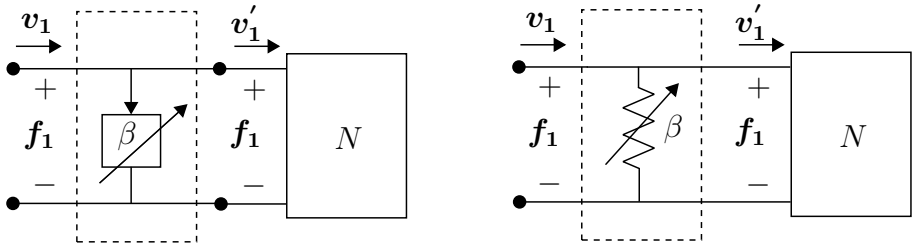


Figure 2.12: One-port network representation with admittance causality PC. Network representation on the left and electrical domain representation on the right.

The exact amount of energy to be dissipated is given when the PO has a negative value, i.e. $E_{obsv} < 0$, therefore the time-varying β is activated when the passivity of the port is violated and it is expressed as follows:

$$\beta(k) = \begin{cases} -\frac{E_{obsv}(k)}{\boldsymbol{f_1(k)^T f_1(k)} T} & E_{obsv}(k) < 0 \\ 0 & E_{obsv}(k) \geq 0 \end{cases} \tag{2.54}$$

The correction can finally be applied in admittance causality as,

$$\boldsymbol{v_1'(k)} = \boldsymbol{v_1(k)} - \beta(k)\boldsymbol{f_1(k)}. \tag{2.55}$$

The active energy is dissipated with the PC in admittance causality and the system results to be passive. The control action will preserve the passivity as can be demonstrated considering that:

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T = \sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1'(k)} T + \sum_{k=0}^{m} \beta \boldsymbol{f_1(k)^T f_1(k)} T = \sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1'(k)} T + \sum_{k=0}^{m-1} \beta \boldsymbol{f_1(k)^T f_1(k)} T + \beta \boldsymbol{f_1(k)^T f_1(k)} T, \tag{2.56}$$

and from (2.53), it will result to be:

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T = E_{obsv} + \beta \boldsymbol{f_1(k)^T f_1(k)} T. \tag{2.57}$$

Substituting β from (2.54) in (2.57), it results

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T \geq 0, \tag{2.58}$$

which is the passivity condition given in Def. 8.

Impedance causality

In impedance causality, the force across the network is modified in order to produce the dissipation and the velocity through the port is conserved. In this case, the

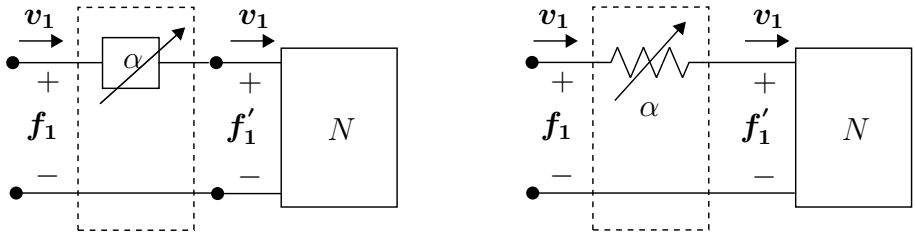


Figure 2.13: One-port network representation with impedance causality PC. Network representation on the left and electrical domain representation on the right.

varying damping α is placed in series with respect to the port as it is shown in Fig. 2.13 (left). The equivalent electrical circuit is also shown on the right and the the time-varying damping is represented with a variable resistor.

The variable $\boldsymbol{f_1'}$ is the force correction given to the network. Similar to the previous case, the passivity observer at the network is

$$E_{obsv}(m) = \sum_{k=0}^{m} \boldsymbol{f_1^T(k) v_1(k)} T + \sum_{k=0}^{m-1} \alpha(k) \boldsymbol{v_1(k)^T v_1(k)}. \tag{2.59}$$

The observer can monitor the exact amount of energy to be dissipated ($E_{obsv} < 0$), therefore the time-varying α is activated when the passivity of the port is violated and it is expressed as follows:

$$\alpha(k) = \begin{cases} -\frac{E_{obsv}(k)}{\boldsymbol{v_1(k)^T v_1(k)} T} & E_{obsv}(k) < 0 \\ 0 & E_{obsv}(k) \geq 0 \end{cases} \tag{2.60}$$

The correction can finally be applied with admittance causality through the port as,

$$\boldsymbol{f_1'(k) = f_1(k)} - \alpha(k) \boldsymbol{v_1(k)}. \tag{2.61}$$

The action of the PC in impedance causality expressed in (2.61) will restore the passivity of the system. In particular, passivity can be demonstrated considering that:

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T = \sum_{k=0}^{m} \boldsymbol{f_1'(k)^T v_1(k)} T + \sum_{k=0}^{m} \alpha \boldsymbol{v_1(k)^T v_1(k)} T =$$
$$\sum_{k=0}^{m} \boldsymbol{f_1'(k)^T v_1(k)} T + \sum_{k=0}^{m-1} \alpha \boldsymbol{v_1(k)^T v_1(k)} T + \alpha \boldsymbol{v_1(k)^T v_1(k)} T, \tag{2.62}$$

and from (2.59), it will results to be:

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T = E_{obsv}(m) + \alpha \boldsymbol{v_1(k)^T v_1(k)^T} T, \tag{2.63}$$

substituting α from (2.60), it results to be,

$$\sum_{k=0}^{m} \boldsymbol{f_1(k)^T v_1(k)} T \geq 0, \tag{2.64}$$

which confirms the passivity condition given in Def. 8.

As discussed before, passivity is a sufficient condition for stability and, in particular, if a system is passive then it is also stable [vdS00]. Hence, enforcing passivity with TDPA allows to enforce a stable behaviour of the network.

Passivity control for two-ports

The main difference for the two-ports network is given by the energy observer because the port has two gateways through which the energy can flow. However, the architecture and the control action performed by the PC is similar to the one-port case. It is proved that a single PC can be added to the network to regulate the energy production [HR02]. As for the one-port, the causality of correction (admittance or impedance), is determined by the system architecture of the port. This implies that the time-varying damper can be placed in series or parallel to the power-port.

An example of the PC applied to the two-ports network is shown in Fig. 2.14, with admittance and impedance causalities. As stated before, placing the PC on the first or second port is independent because any action will be seen by the energy observer at any time k.

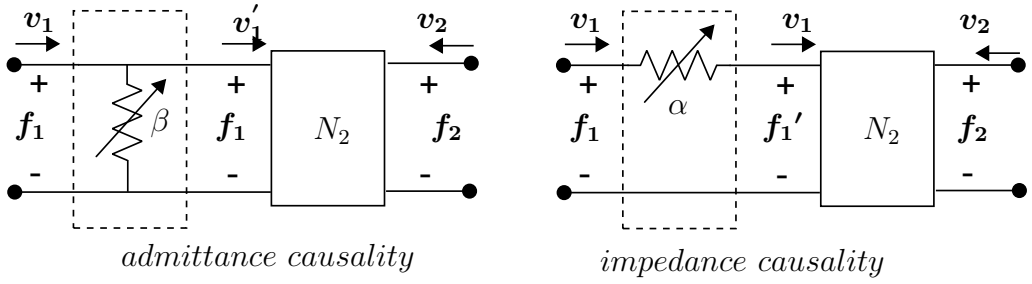


Figure 2.14: Two port network equipped with PC in admittance causality (right) and impedance causality (left).

Although the passivity control follows the approach described for the one-port, for completeness the steps are reported below in a compact form.
For the admittance causality:

- The Passivity Observer at the port is

$$E_{obsv}(m) = \sum_{k=0}^{m} \boldsymbol{f_1^T(k) v_1'(k)} T + \sum_{k=0}^{m} \boldsymbol{f_2^T(k) v_2(k)} T + \sum_{k=0}^{m-1} \beta(k) \boldsymbol{f_1(k)^T f_1(k)}.$$

- β is activated when the passivity of the port is violated and it is expressed as follows:

$$\beta(k) = \begin{cases} -\frac{E_{obsv}(k)}{\boldsymbol{f_1(k)^T f_1(k)} T} & E_{obsv}(k) < 0 \\ 0 & E_{obsv}(k) \geq 0 \end{cases} \tag{2.65}$$

- The correction can finally be applied with admittance causality through the port as, $\boldsymbol{v_1'(k) = v_1(k) - \beta(k) f_1(k)}$.

For the impedance causality

- The Passivity Observer is defined as

$$E_{obsv}(m) = \sum_{k=0}^{m} \boldsymbol{f_1^T(k)v_1(k)}T + \sum_{k=0}^{m} \boldsymbol{f_2^T(k)v_2(k)}T + \sum_{k=0}^{m-1} \alpha(k)\boldsymbol{v_1(k)^T v_1(k)}.$$

- The time-varying damping is activated as follows,

$$\alpha(k) = \begin{cases} -\frac{E_{obsv}(k)}{\boldsymbol{v_1(k)^T v_1(k)}T} & E_{obsv}(k) < 0 \\ 0 & E_{obsv}(k) \geq 0 \end{cases} \tag{2.66}$$

- The passivity control correction is applied with impedance causality through the port as,

$$\boldsymbol{f_1^{'}(k)} = \boldsymbol{f_1(k)} - \alpha(k)\boldsymbol{v_1(k)}. \tag{2.67}$$

Following the steps already described for the one-port case, passivity can be proved for the PC acting in admittance (2.65) and impedance causalities (2.66) and the following condition of passivity holds,

$$\sum_{k=0}^{m} \boldsymbol{f_1^T(k)v_1(k)}T + \sum_{k=0}^{m} \boldsymbol{f_2^T(k)v_2(k)}T \geq 0. \tag{2.68}$$

As stated before, this ensures the passivity of the two-port network according to Def. 9.

2.8 Discussion

The stability of the robotic simulator needs to be guaranteed under all the operating conditions while simulating the satellite dynamics. In this chapter, the stability issue caused by the time delay in the control loop has been analysed for a reduced model of the simulator. Further, the passivity theory has been introduced together with the network representation. Passivity uses the energy concept which can be exploited in energy-based methods. A background on these methods was presented and a comparison of the methods for a reduced model of the robot simulator was performed. The results show that TDPA, compared to wave variables and energy tank approaches, has better performance in dealing with time delay intrinsic in the control loop.

The TDPA-based methods, the passivity concept and the power-port networks will be exploited for the design and analysis of the controllers developed in this thesis.

CHAPTER 3

ENERGY-BASED CONTROL FOR TIME DELAY COMPENSATION

3.1 Introduction

The control architecture of a complex system might have several algorithms running on different CPUs. This factor can affect the low-level control, which results sensitive to time delays caused by the transmission of the signals. As shown in Chapter 2, the time delay in the control loop of a robotic simulator causes system instability while reproducing the satellite dynamics.

In this chapter, the effects of the time delay are analysed from an energy perspective and a passivity-based control is developed to cope with its destabilising effects. Firstly, the robotic system which simulates the satellite dynamics is designed in electrical domain. This representation will reveal the network of the system and the corresponding power-ports. The network analysis is used to identify and isolate the *energy leaks*, that is, undesired sources of energy that can lead the system to become unstable [ARPH11]. Once the network is identified, the extra energy can be dissipated in order to counteract the damaging effects on the system stability. Therefore, a designed passivity controller will act in an admittance causality and it will exploit the time-domain passivity approach. Secondly, an optimal control is developed in order to guarantee performance in reproducing the dynamics with a robot while keeping the stability properties. The methods are validated with simulations and experiments.

Related works

Controller designs to compensate time delay in the robotic simulator have been implemented over the past years [QZG+16], [ZLBC12]. These methods rely on the modification of the contact parameters in order to achieve stable simulation. In

[OKU10] and [ASJ+14b] a first order compensation model was introduced based on the contact frequency and time delay knowledge. The drawback of these approaches is the *a priori* knowledge of the contact parameters, which might be inaccurate and therefore can limit the range of reproducibility of the dynamics simulated by the robot.

On the other hand, approaches based on passivity have emerged as intuitive and effective strategies for achieving stability independent of time delay. In fact, ensuring the passivity of the overall system is a sufficient condition for having a stable behaviour [SSF07]. The field of haptics and teleoperation has thoroughly exploited approaches based on passivity to deal with instabilities due to time delay (see [HS06] for a survey). In [DNBS06], the time delay and zero-order-hold are shown to be energy generating, resulting in an active system causing potential instability. In [BC98b] a minimum virtual mass passivity value is found for a haptic device in an impedance like configuration. The wave variables approach ([Nie04, SSvdSF05]) have been implemented for making a bilateral communication channel passive, however this technique introduces a characteristic impedance which might deteriorate the desired behaviour [SFF16]. In [FSM+11] and [FPM+15] the concept of energy tank has been exploited for dealing with the delay, but a suitable dissipative action needs to be chosen. To compensate the time delay effects, the time domain passivity approach (TDPA, [HR01]) is also exploited. Within this context, [PHRH03] proposed a geometric solution for haptic devices. In [HHK+10], delay has been compensated using a time-varying damping weighted with the inertia matrix for a haptic device that, unlike industrial robots, is backdrivable. In [OAP11], a technique for dissipating the active energy in the null space of a redundant robot is suggested. This solution is not applicable in the considered context since most of the facilities for rendering satellite dynamics do not have redundancy (e.g., [BWMT10] and [ADSR+15]).

The TDPA-based method was exploited for reliably simulating the satellite dynamics on a robot simulator, independently of the time delay. This approach will be presented in this chapter, which refers to the author contribution published in [DSARAS15] and [DSAS16].

3.2 Problem Statement

In Sec. 2.3, the stability analysis of the robotic simulator with time delay in the close-loop was performed and the results have shown that the time delay might cause system instability, especially when an elastic contact is involved. The time delay introduces energy in the system and it causes damaging effects in reproducing the dynamics with the robot. In particular, it results in an increase in velocity for the simulated dynamics, which represents a non-physical behaviour.

The increase in energy and velocity represent the problem statement, which will be better explained in this section. The model of the satellite dynamics was already

presented in Sec. 2.2 and it is reported below for completeness,

$$\begin{aligned} \dot{\boldsymbol{v}}_{\boldsymbol{s}} &= \boldsymbol{M}^{-1}\boldsymbol{f}_{\boldsymbol{e}}, \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{s}} &= \boldsymbol{I}^{-1}(\boldsymbol{I}\boldsymbol{\omega}_{\boldsymbol{s}} \times \boldsymbol{\omega}_{\boldsymbol{s}} + \boldsymbol{\tau}_e). \end{aligned} \tag{3.1}$$

The velocity $\boldsymbol{V}_{\boldsymbol{s}} = (\boldsymbol{v}_{\boldsymbol{s}}, \boldsymbol{\omega}_{\boldsymbol{s}}) \in \mathbb{R}^6$ is the input to the robot, which can be obtained by the integration of (3.1) in discrete-time.

It is worth pointing out that the energy associated with the simulated dynamics in discrete-time will be affected by the time delay, in particular, for the discrete system this can be defined as:

$$E(n) = E(0) + \sum_{k=0}^{n} \boldsymbol{F}_{\boldsymbol{e}}(\boldsymbol{k})^T \boldsymbol{V}_{\boldsymbol{s}}(\boldsymbol{k} - \boldsymbol{\mu})T, \tag{3.2}$$

where μ is the number of discrete-time steps of sampling time T and its analogy with a continuous-time delay is that $T_d = \mu T$. The vector $\boldsymbol{F}_{\boldsymbol{e}} = (\boldsymbol{f}_{\boldsymbol{e}}, \boldsymbol{\tau}_{\boldsymbol{e}}) \in \mathbb{R}^6$ is the total wrench.

The problem statement is introduced with a simple example, which considers a rigid body with a defined virtual inertia, I. The body moves in an environment as shown in Fig. 3.1 and it receives elastic forces during the contact with the environment.

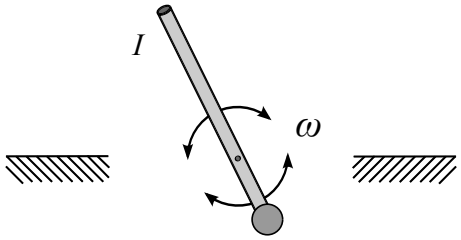


Figure 3.1: A rigid-body rotating and colliding against walls.

The energy of this system is calculated for the ideal case (without time delay) and when time delay is involved in the control loop. Fig. 3.2 shows a comparison of the calculated energy. E_i is the ideal energy calculated in continuous-time and without time delay, which represents the ground truth of the motion. E_{TD} is the energy calculated in discrete-time with time delay, which represents the deviation from the ideal behaviour. As can be seen, the time delay introduces negative energy in the system, which violates the passivity condition given in (2.49).

Further, when an elastic contact occurs between the body and the environment, the body should have the same velocity after the contact (or less in case of inelastic contact). However, a divergence of the angular velocity, ω_{sTD}, with respect to the ideal velocity, ω_{si}, is found in the system, as it can be seen in the bottom plot of Fig. 3.2. The system has an active behaviour, which is caused by the time delay in the loop. The velocity will be transmitted to the robot for simulating the dynamics and the corresponding motion results to be unstable.

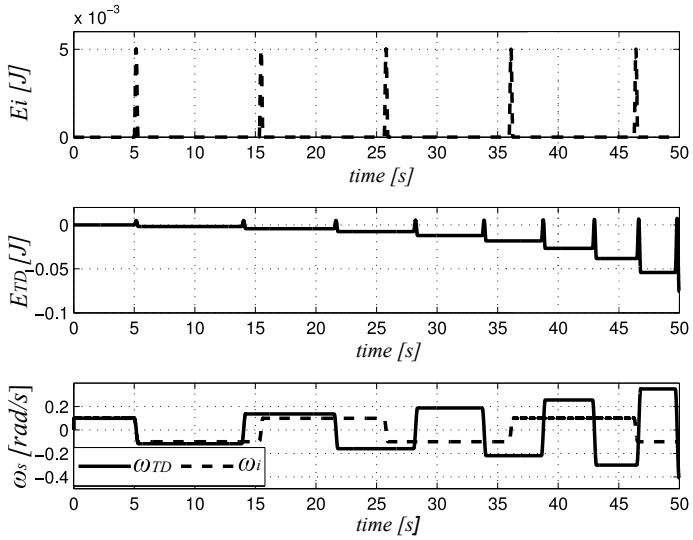


Figure 3.2: Energy for the ideal case without time delay (top), energy with time delay in the loop (middle), comparison of the velocities (bottom).

In the next section, a network representation and a more detailed analysis is proposed in order to identify the energy leaks. Furthermore, the TDPA will be exploited to overcome this unstable phenomenon.

3.3 Network Representation

The modelling process is divided into two steps: first, the system is represented as a block diagram. The block diagram is a straightforward representation of the system as an interconnection of transfer functions and feedbacks. However, when energy considerations are taken into account, this representation might not be well suited. To that end, a second step which uses the mechanical-electrical analogies is exploited and the corresponding networks can be found. As it will be seen, this facilitates the passivity analysis and helps to identify the energy leaks, that is, undesired sources of energy that appear due to the delay in the system.

Fig. 3.3 shows the block diagram of the complete robotic system which simulates the full dynamics of a satellite. The satellite dynamics is represented by a single block, whose output is the velocity vector $\boldsymbol{V_s(k)}$ and inputs are the forces and torques $\boldsymbol{F_e}$, measured by the sensor. The robot is represented in the gray area and it is composed of an admittance R (that generates a velocity $\boldsymbol{V_r}$) and C, which is the controller of the robot and it is assumed to be known. The block TD represents the time delay in the loop and E is the environment.

The second step is to design the electrical scheme of the block diagram using the analogies described in Sec. 2.5. The corresponding electrical scheme for the ideal case (no time delay in the loop) is shown in Fig. 3.4. The forces and torques coming from the sensor are modelled as an ideal force generator $\boldsymbol{F_e}$ that acts on the impedance Z_m, which represents the virtual mass and its analogy is an inductance.

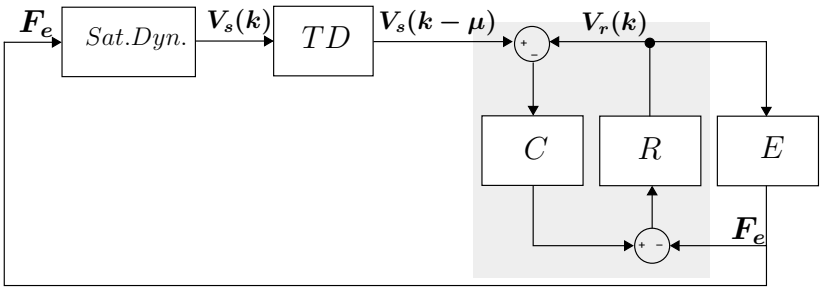


Figure 3.3: Block diagram of the satellite dynamics (*Sat.Dyn.*) simulated by the Robot (R). TD is the time delay in the loop, C, the robot controller and E, the environment.

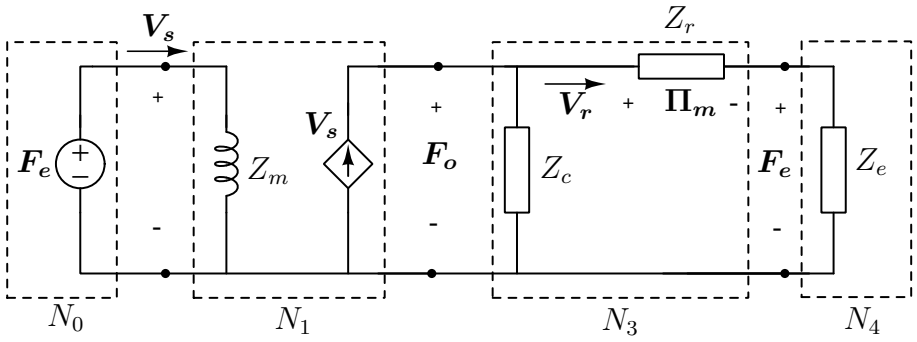


Figure 3.4: Electrical analogy of the system in ideal case.

Through a dependent current source, the velocity *through* the mass is fed to the controller Z_c, which, in turn, moves the robot, represented by a general impedance, Z_r with voltage $\boldsymbol{\Pi_m}$. The vector $\boldsymbol{F_o}$ is the force of the controller which actuates the robot. The environment is represented by an impedance Z_e[1]. Fig. 3.4 also defines the ports of the system, where N_0 is a one-port containing the sensor and N_1 is a two-port transmission. N_3 is a network containing the robot and the controller.

The proposed electrical scheme with the time delay network, N_T, is shown in Fig. 3.5. The time delay is located between the networks N_1 and N_3 and it produces a delayed velocity $\boldsymbol{V_s(k-\mu)}$. N_T represents an active network, with input $\boldsymbol{V_s(k)}$, output $\boldsymbol{V_s(k-\mu)}$ and voltage $\boldsymbol{F_o}$. The two-ports network, N_T, is responsible for the leaks of energy. The passivity can be re-established by introducing a control port, namely N_2, where a time-varying damping will be activated by the passivity controller, Z_{pc}, later defined. This element will ensure the stability of the system by dissipating the active energy due to the time delay, as it will be shown in the following section.

[1]Note that the electrical analogy with the force across the controller (and the robot) given as, $\boldsymbol{F_o} = \boldsymbol{V_s}\frac{Z_c Z_{re}}{Z_c + Z_{re}}$, where $Z_{re} = Z_r + Z_r$ is the equivalent serial impedance of the robot and the environment.

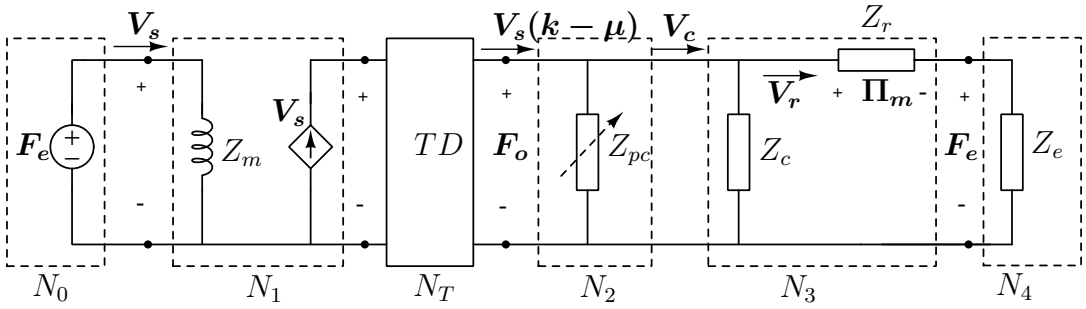


Figure 3.5: Electrical analogy of the system with time delay and PC.

3.4 Passivity-based Controller

The passivity-based control will provide a correction in velocity $\boldsymbol{V_c}$ as input to the robot network in Fig. 3.5. The controller is characterised by three main elements defined as:

- **Force Observer**, which monitors the total forces in the system.
- **Energy Observer**, which monitors the energy in the system due to the time delay.
- **Passivity Controller** in admittance causality, which will enforce the passivity by correcting the velocity of the robot.

Forces Observer

A forces observer can monitor the forces acting on the system and it can be used when the dynamic of the robot is available. The forces observer is based on the dynamics model of the robot and it is described in Cartesian space as follows:

$$\boldsymbol{\Lambda_m}(\boldsymbol{V_r})\ddot{\boldsymbol{V}}_{\boldsymbol{r}} + \boldsymbol{\eta}(\boldsymbol{V_r}, \dot{\boldsymbol{V}}_{\boldsymbol{r}})\dot{\boldsymbol{V}}_{\boldsymbol{r}} + \boldsymbol{g}(\boldsymbol{V_r}) = \boldsymbol{\Pi_m}, \tag{3.3}$$

where $\boldsymbol{\Lambda_m} \in \mathbb{R}^{6\times6}$ is the Cartesian mass matrix, $\boldsymbol{\eta} \in \mathbb{R}^{6\times6}$ is the non-linear Coriolis/centrifugal Cartesian matrix, and $\boldsymbol{g} \in \mathbb{R}^6$ is the gravity vector. The Cartesian mass matrix, Coriolis matrix and gravity vector are expressed as follows:

$$\boldsymbol{\Lambda_m} = \boldsymbol{J^{-T}} \boldsymbol{H_m} \boldsymbol{J^{-1}}, \tag{3.4}$$

$$\boldsymbol{\eta}\dot{\boldsymbol{V}}_{\boldsymbol{r}} = \boldsymbol{J^{-T}} \boldsymbol{C_m} \dot{\boldsymbol{q}} - \boldsymbol{\Lambda_m} \dot{\boldsymbol{J}} \dot{\boldsymbol{q}}, \tag{3.5}$$

$$\boldsymbol{g} = \boldsymbol{J^{-T}} \boldsymbol{G}. \tag{3.6}$$

where $\boldsymbol{H_m}$ and $\boldsymbol{C_m} \in \mathbb{R}^{6\times6}$ are the inertia and the Coriolis matrix available in joints space. $\boldsymbol{G} \in \mathbb{R}^6$ is the gravity vector expressed in joints space and $\dot{\boldsymbol{q}} \in \mathbb{R}^6$ is the velocity of the joints [SK08]. It is assumed that the robot operates in a free-of-singularity workspace, i.e. the Jacobian $\boldsymbol{J} \in \mathbb{R}^{6\times6}$ has full row rank.

The vector $\boldsymbol{\Pi_m} \in \mathbb{R}^6$ in (3.3) represents the contribution of the end-effector forces due the the robot motion. When external forces are applied at the robot end-effector, these can be measured by the force-torque sensor which provides measured

values represented by the vector $\boldsymbol{F_e} \in \mathbb{R}^6$. Therefore, the equation of the forces observer $\boldsymbol{F_o}$ is given by:

$$\boldsymbol{F_o} = \boldsymbol{F_e} + \boldsymbol{\Pi_m}. \tag{3.7}$$

The analogy with the electrical system in Fig. 3.5 is given by the voltage $\boldsymbol{F_o}$ across the network N_3.

Passivity Observer (PO)

The network N_T shown in Fig. 3.5 is the active network to be passivated. N_T is a two-port network where the net energy, which is the difference between the input and output energies, can be calculated as:

$$E_{obs}(k) = \sum_{k=0}^{m} \boldsymbol{F_o}^T(k)(\boldsymbol{V_s(k)} - \boldsymbol{V_s(k-\mu)})T. \tag{3.8}$$

where the time delay is taken into account. To observe the energy, passivity is handled for each degree of freedom, i, independently. The balance of energy holds also if the components of the vectors are considered, i.e. each degree of freedom, i, independently. In particular, (3.8) results in the following,

$$E_{obs}(k) = \sum_{k=0}^{m}\sum_{i=1}^{n} F_{o,i}(k)(V_{s,i}(k) - V_{s,i}(k-\mu))T = \sum_{i=i}^{n} E_{obs_i}, \tag{3.9}$$

where $V_{s,i}(k)$ is the component of the velocity vector of the body, $F_{o,i}(k)$ is the component of the controller forces and torques and $V_{s,i}(k-\mu)$ is the delayed velocity. In the design of the passivity observer, the corresponding energy introduced by the passivity controller must be also considered. Therefore, the final equation for the observer is given as follows,

$$E_{obs_i}(k) = E_{obs_i}(k-1) + \underbrace{F_{o,i}(k)(V_{s,i}(k) - V_{s,i}(k-\mu))T}_{E_{N_T}} + \underbrace{F_{o,i}^2(k-1)\beta(k-1),}_{E_{pc}} \tag{3.10}$$

where the second term on the right side is the energy due to the time delay of the network N_T and the last term, the energy due to the passivity controller, later defined.

Passivity Controller (PC)

The Passivity Controller is based on an admittance causality and it acts through a time-varying damping factor β, which is function of the observed energy flow (3.10). In the admittance configuration, the velocity is modified to produce the dissipation. Therefore, the PC will correct the velocity commanded to the robot as follows,

$$V_{c,i}(k) = V_{s,i}(k-\mu) - \beta_i(k-1)F_{o,i}(k)$$

where $\beta_i(k-1)$ is the time varying damping factor which corrects the velocity by a quantity $V_{pci}(k)$, given by:

$$V_{pci}(k) = -\beta_i(k-1)F_{o,i}(k). \tag{3.11}$$

The passivity condition is taken into account in the design of the damping coefficient, defined as,

$$\beta_i(k) = \begin{cases} -\frac{E_{obs_i}(k)}{F_{o,i}^2(k)T} & E_{obs_i}(k) < 0 \\ 0 & E_{obs_i}(k) \geq 0. \end{cases} \tag{3.12}$$

Therefore, the final velocity sent to the robot, $V_{c,i}$, will be modified according to the time-varying damping in (3.12) when the passivity condition is violated, i. e. $E_{obs_i}(k) < 0$. Thus, the network created by N_T along with N_2, i.e., the passivity controller, is rendered to be passive.

Note that the coefficient β_i is free of mathematical singularities because there is always a value of force ($F_{o,i}$) available from (3.7). Indeed, the forces observer guarantees the monitoring of the forces in free-motion (through the dynamics of the robot described by the vector $\Pi_{m,i}$) and in presence of external forces (through the vector F_e). The only case when $F_{o,i} = 0$ is when the robot does not move and in this case no loss of passivity is detected and no correction is required [DSAS16].

3.4.1 Results: Simulations and Experiments

In the first simulation, an initial angular velocity is given to the satellite in order to simulate a spin around its body axis. The body moves in a workspace and external forces are modelled using a spring-dashpot model in order to simulate the FTS sensor (see Fig. 3.1). The contact model is therefore described by the following transfer function:

$$H(z) = K_w + D_w \frac{z-1}{\Delta T z}, \tag{3.13}$$

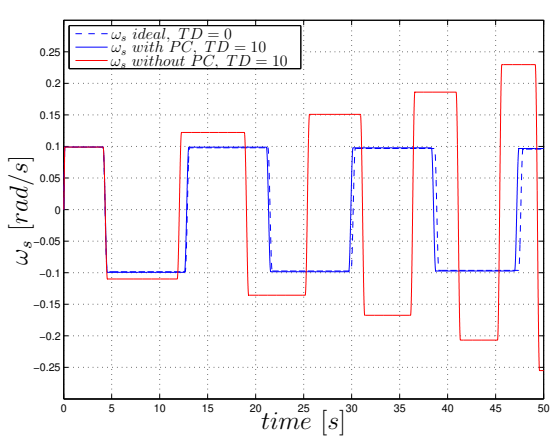
where K_w is a stiffness gain and D_w is a damping gain. In the case of an ideal impact, the rotational velocity before and after should be maintained according to the principle of conservation of energy for an elastic impact.

The angular velocity shown in Fig. 3.6a is simulated for a body with $diag(I) = 0.9\ Kgm^2$ with initial angular velocity $0.1\ rad/s$ on the $z-axis$ and it is compared for the case of:

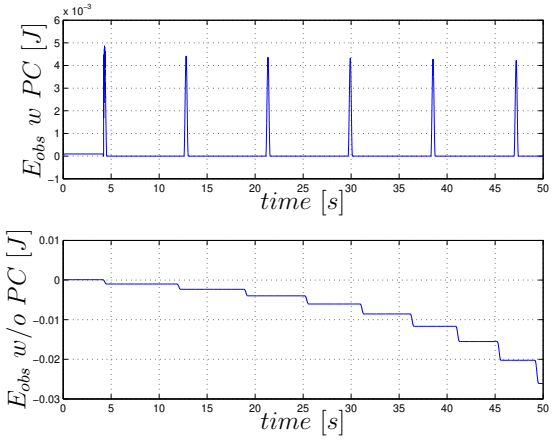
- **ideal scenario:** no time delay in the loop,
- **real scenario:** time delay in the loop and without PC,
- **real scenario with proposed control:** time delay in the loop with PC.

The dashed line displays the motion of the body in the ideal case (without time delay) while rotating and colliding against the environment. If a time delay of 10 ms is located in the loop without PC, the angular velocity increases after each impact

(see the red line), resulting in an active behaviour in the system. This factor was already presented as problem statement and it was visualised in Fig. 3.2. Now, the proposed method is applied, the solid blue line in Fig. 3.6a shows the angular velocity corrected with the PC. As it can be seen, it is maintained very close to the ideal case. Therefore, the problem described in section Sec. 3.2 is solved using the described approach. In particular, the energy behaviour can be seen in Fig. 3.6b. The bottom plot of Fig. 3.6b, clearly shows the active behaviour of the network without PC, leading to an increase of the angular velocity. The growing negative

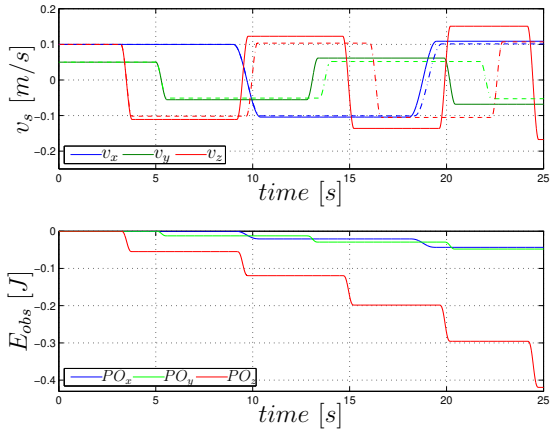


(a) Angular velocity without time delay for the ideal case (dashed line), with time delay (red line) and with time delay and PC (solid blue line).

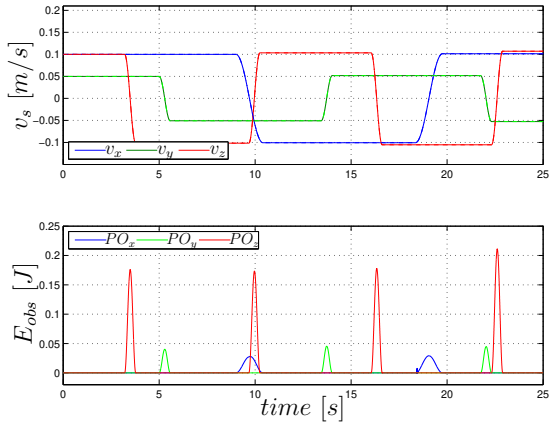


(b) Energy observer with PC (top) and energy observer without PC. Activity is detected by the negativeness of the energy without the PC. Stable behaviour with PC.

Figure 3.6: Angular velocity comparison and energy observer with PC and without PC.



(a) Top: Velocity comparison between ideal case (dashed line) and with time delay (solid line), Bottom: Energy observer.



(b) Velocity comparison: ideal case (dashed line) and with time delay and PC(solid line), Bottom: Energy observer with PC.

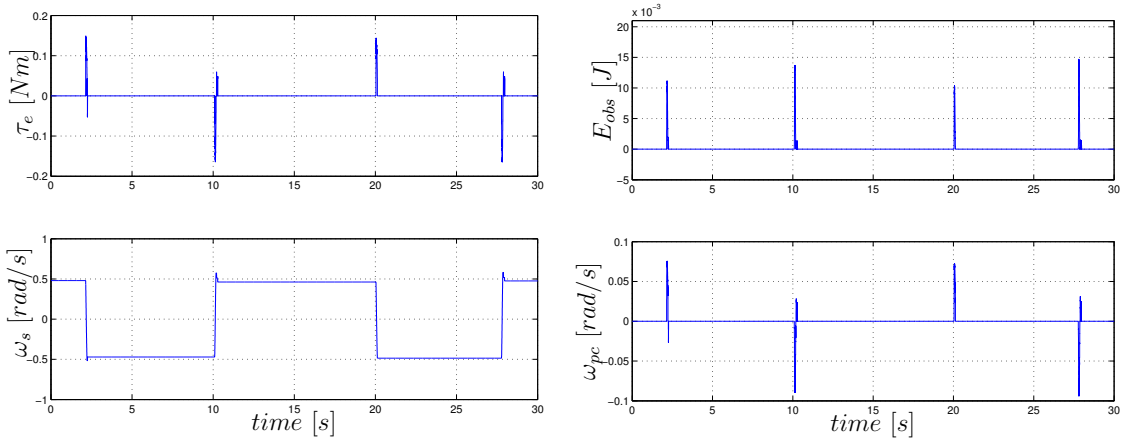
Figure 3.7: Velocity comparison and energy observer.

value in the energy shows that energy is injected into the system after each contact. Fig. 3.6b (top) shows the passivity observer with the action of the PC. The positive definiteness of the energy respects the passivity condition and the network results to be passive.

The second simulation shows the dynamics interaction of a simulated mass of $50Kg$ with an initial velocity $v_1 = [0.1, 0.05, 0.1]m/s$ and a time delay of 10 ms. The body collides against virtual walls that have been modelled with high stiffness springs. The velocity in Fig. 3.7a shows the unstable behaviour given by the time delay. The dashed line is the ideal motion of the satellite obtained in the ideal case (without time delay). At each impact, the velocity increases along all the components. Further, active energy is introduced as the Passivity Observer in Fig. 3.7b bottom shows. By applying the method, the velocity of the simulated mass follows the one given by the ideal motion (dashed line). The network is then passive as Fig. 3.7a shows. As it can be seen, the simulations prove the validity of the proposed approach.

Experiments

Preliminary experiments are carried out on a 1 dof SensorDrive set-up equipped with a torque sensor and a rigid bar attached to it (see Sec. A.2 for more details about this set-up). This experiment is required to prove the robustness of the controller with sensor characteristics such as noise. The bar is a mechanical interface which makes contacts with the rigid walls on both sides. It can be seen as a 1 dof spinning satellite where the inertial matrix is a scalar, the vectors V_s and F_o are composed of only one component. The internal controller was parametrised for a maximum performance assuming an ideal case, i.e. with high stiffness and nearly-null damping. The initial angular velocity of the system is $\omega_{init} = 0.473\ rad/s$ and the time delay is 10 ms.



(a) Measured torque signals and angular velocity with PC.

(b) Energy observer and correction of the PC.

Figure 3.8: Experiment results with the passivity control.

The virtual inertia to be simulated is $I = 0.009Kgm^2$. The choice of a small value for inertia is made to minimise the contact time and it allows one to see the influence of the time delay. Fig. 3.8a shows the measured torque and the velocity of the simulated body during the experiment with the PC enabled. It can be seen that through out the duration of the experiment, the magnitude of the angular velocity remains constant after each impact. The generated energy due to the time delay is damped making the network (and therefore the system) passive. The experimental proof of the concept can be seen in Fig. 3.8b in which the top plot shows the energy behaviour of the network and the bottom plot shows the velocity correction made by the PC. As can be seen, the resulting energy of the network is always positive due to the action of the PC. Furthermore, instability features are not detected proving the validity of the proposed method considering also sensor noise.

3.5 Optimised Passivity-based Controller

As shown in the previous section, the passivity control can guarantee the stability of the system. However, a possible limitation of the controller might be given by the high frequency velocity modifications of the TDPA damping injection which can create over-dissipation. This factor can affect the performance in simulating the satellite dynamics.

In this section an optimised passivity-based control is developed to provide higher accuracy for simulating the satellite dynamics on a robot. The method considers a unique energy observer which monitors the energy due to the delay and an optimised PC for ensuring passivity while maximising the performance.

Limitation of the passivity control

The strategy described in the preceding section enforces passivity of the overall system, however the control action might be over conservative and it can lead to a decrease in the performance. In fact, when the energy observer $E_{obs}(m) \geq 0$ the overall system is passive, however the passivity control might dissipate more than the required energy. An optimal condition will be to dissipate exactly the amount of energy required in order to have an overall balance of extra energy with $E_{obs}(m) = 0$. The following example clarifies the problem.

Example 2 *Considers a system with a time delay of* 10 *ms acting on the linear and angular velocities commanded from the simulated dynamics to the robot. The simulated mass of the satellite is* 60 Kg *and the inertia parameters are* $I_{xx} =$ 18 $Kgm^2, I_{yy} = 20\ Kgm^2$ *and* $I_{zz} = 22\ Kgm^2$ *and the initial conditions are defined as follows:* $v_{init} = [0.1\ 0.05\ 0.1]\ m/s$, $\omega_{init} = [2\ \ -3\ 3]\ deg/s$.

Fig. 3.9 (top) shows the passivity observer when the passivity control is applied without an optimal criteria. This results in a stable system ($E_{obs} \geq 0$), however the performance in tracking the simulated velocity is affected. This can be seen at the

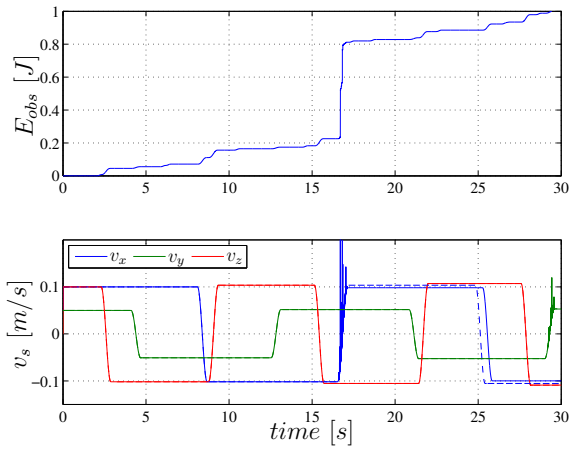


Figure 3.9: Ideal velocity (dashed line) and actual linear velocity of the satellite without the optimized passivity control (solid line). Over dissipation affects the performance.

bottom of Fig. 3.9, which shows the comparison of the velocity with the ideal one (dashed line). The high frequency velocity modifications affect the performance. In order to avoid over-dissipation, an optimal time-varying damping can be designed while respecting the energy constraints.

Optimised Passivity Controller

In Sec. 3.4 the energy flow was observed using different POs applied component-wise and the produced energy was dissipated by having independent PCs. Here a multi-dimensional PC is considered as a time-varying dissipative matrix. The energy exchange corresponding to the active network is monitored through the PO as described in (3.8) and the velocity correction provided to the robot is given as,

$$\boldsymbol{V_c(k)} = \boldsymbol{V_s(k-\mu)} - \boldsymbol{\beta(k) F_o(k)}. \tag{3.14}$$

The matrix $\boldsymbol{\beta(k)}$ represents now a time-varying diagonal matrix. The assumption to choose a diagonal structure for the damping coefficient $\boldsymbol{\beta(k)} = diag(\beta_1(k) \dots \beta_n(k))$ is not restrictive because by properly tuning the elements of $\boldsymbol{\beta(k)}$ it is possible to act on all the Cartesian directions. In order to make the robotic simulator passive while simultaneously maximising the tracking performance, $\boldsymbol{\beta(k)}$ is chosen in such a way that the following conditions hold:

- **Energy condition:** If $E_{obs}(k) < 0$, then $\boldsymbol{F_o^T(k) \beta(k) F_o(k)} T = -E_{obs}(k)$.
- **Performance condition**: The compensated velocity $\boldsymbol{V_c(k)}$ in (3.14) has to be as close as possible to the target satellite velocity $\boldsymbol{V_s(k)}$.

The first condition guarantees the passivity of the system while the second one ensures optimal performance. In order to meet these conditions, a minimization

problem will be designed for tuning the PC. Fig. 3.10 shows the scheme of the elements involved in the design of the optimal passivity control. The environmental forces and torques (E) measured with the sensor are sent to the satellite dynamics (3.1) which provides a vector $\boldsymbol{V_s(k)}$. TD is the time delay located in the loop that is responsible for sending the delayed velocity $\boldsymbol{V_s(k-\mu)}$ to the robot. This velocity is corrected with the PC in (3.14) and it represents the input to the robot (R). $\boldsymbol{F_o}$ is the dynamics forces estimator defined in (3.7). The PO is the energy observer in (3.8) that provides a scalar, E_{obs}, to the minimization problem (MIN) in order to compute the optimal coefficients β. Therefore, the PC commands the corrected velocity, $\boldsymbol{V_c}$ in (3.14).

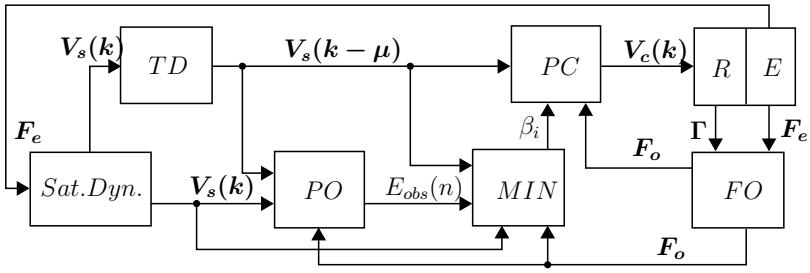


Figure 3.10: Scheme of the control elements

The coefficients of the damping matrix exploit the results of an optimization problem, which is formulated as follows,

$$\min_{\beta(k)} \|\boldsymbol{V_s(k-\mu)} - \boldsymbol{\beta(k)F_o(k)} - \boldsymbol{V_s(k)}\|^2, \tag{3.15}$$

where the following equality constraint needs to be satisfied:

$$\textstyle\sum_{k=0}^{m} \boldsymbol{F_o(k)}^T \boldsymbol{\beta(k)F_o(k)}T = \bar{E}_{obs}(m), \tag{3.16}$$

where

$$\bar{E}_{obs}(m) = \begin{cases} 0 & \text{if } E_{obs}(m) \geq 0 \\ -E_{obs}(m) & \text{if } E_{obs}(m) < 0. \end{cases} \tag{3.17}$$

The function to be minimised $f(\boldsymbol{\beta(k)}) = \|\boldsymbol{V_s(k-\mu)} - \boldsymbol{\beta(k)F_o(k)} - \boldsymbol{V_s(k)}\|^2$ is convex with respect to $\beta_i(k)$ (the variables to be optimised) and the Hessian of $f(\boldsymbol{\beta})$ is positive semi-definite. Thus (3.15) is a convex optimization problem and, therefore, it is suitable to be solved in real-time. The minimization problem forces a choice of $\beta(k)$ such that the energy produced by the delay is dissipated and the velocity transmitted to the robot is as close as possible to the ideal target velocity.

Notice that no constraint has been set on the sign of the elements on the diagonal of $\boldsymbol{\beta(k)}$ because it will be a positive definite matrix (see the equality constraint). Furthermore, the absence of positive constraint on the β_i terms, provides a greater flexibility in the choice of the multidimensional damper and, in principle, to a better solution for the minimization problem. It is possible to reformulate (3.15) as a

quadratic problem. For ease of notation, the velocity terms are grouped and the time dependency is omitted by setting:

$$\begin{aligned} \boldsymbol{V} &:= \boldsymbol{V_s}(\boldsymbol{k}-\boldsymbol{\mu}) - \boldsymbol{V_s}(\boldsymbol{k}), \\ \boldsymbol{\beta} = \operatorname{diag}(\beta_1, \ldots, \beta_n) &:= \boldsymbol{\beta}(\boldsymbol{k}), \\ \boldsymbol{F} &:= \boldsymbol{F_o}(\boldsymbol{k}). \end{aligned} \tag{3.18}$$

Using the proposed notation, (3.15) can be written as:

$$(\boldsymbol{V}-\boldsymbol{\beta F})^T(\boldsymbol{V}-\boldsymbol{\beta F}) = \boldsymbol{V}^T\boldsymbol{V} - 2\boldsymbol{V}^T\boldsymbol{\beta F} + \boldsymbol{F}^T\boldsymbol{\beta}^T\boldsymbol{\beta F}, \tag{3.19}$$

where the corresponding terms are:

$$\begin{aligned} &2\boldsymbol{V}^T\boldsymbol{\beta F} = 2V_{s1}\beta_1 F_1 + \cdots + 2V_{sn}\beta_n F_n = \\ &\begin{pmatrix} 2V_{s1}F_1 & \cdots & 2V_{sn}F_n \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}, \\ &\boldsymbol{F}^T\boldsymbol{\beta}^T\boldsymbol{\beta F} = F_1^2\beta_1^2 + \cdots + F_n^2\beta_n^2 = \\ &\begin{pmatrix} \beta_1 & \cdots & \beta_n \end{pmatrix} \begin{pmatrix} F_1^2 & 0 & \cdots & 0 \\ 0 & F_2^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F_n^2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}. \end{aligned} \tag{3.20}$$

The constraint in (3.16) can be rewritten as:

$$\begin{pmatrix} F_1^2T & \cdots & F_n^2T \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} = \bar{E}_{obs}(m). \tag{3.21}$$

Thus, by setting:

$$\boldsymbol{x} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}, \qquad \boldsymbol{P} = 2\begin{pmatrix} F_1^2 & 0 & \cdots & 0 \\ 0 & F_2^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & F_n^2 \end{pmatrix},$$

$$\boldsymbol{q}^T = \begin{pmatrix} 2v_1F_1 & \cdots & 2v_nF_n \end{pmatrix}, \quad r = \boldsymbol{V}^T\boldsymbol{V}, \tag{3.22}$$

$$\boldsymbol{A}^T = \begin{pmatrix} F_1^2T & \cdots & F_n^2T \end{pmatrix}, \qquad b = \bar{E}_{obs}(m),$$

the equation in (3.15) can be rewritten as a standard quadratic optimization problem as follows,

$$\begin{aligned} &\min_x \tfrac{1}{2}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x} + r, \\ &\boldsymbol{A}^T\boldsymbol{x} = b. \end{aligned} \tag{3.23}$$

For (3.23), the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient [BV04] and they require that:

$$\begin{pmatrix} \boldsymbol{P} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^\star \\ \lambda^\star \end{pmatrix} = \begin{pmatrix} -\boldsymbol{q} \\ b \end{pmatrix}, \tag{3.24}$$

where $\boldsymbol{x}^\star$ are the primal solutions, that is, the optimal solutions for the damping coefficients required and $\lambda^\star$ is the dual solutions of (3.23). If $\boldsymbol{A}$ has full rank and $\boldsymbol{P}$ is symmetric and positive definite, then (3.24) admits only one solution [BV04] given by:

$$\begin{pmatrix} \boldsymbol{x}^\star \\ \lambda^\star \end{pmatrix} = \begin{pmatrix} \boldsymbol{P} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} -\boldsymbol{q} \\ b \end{pmatrix}. \tag{3.25}$$

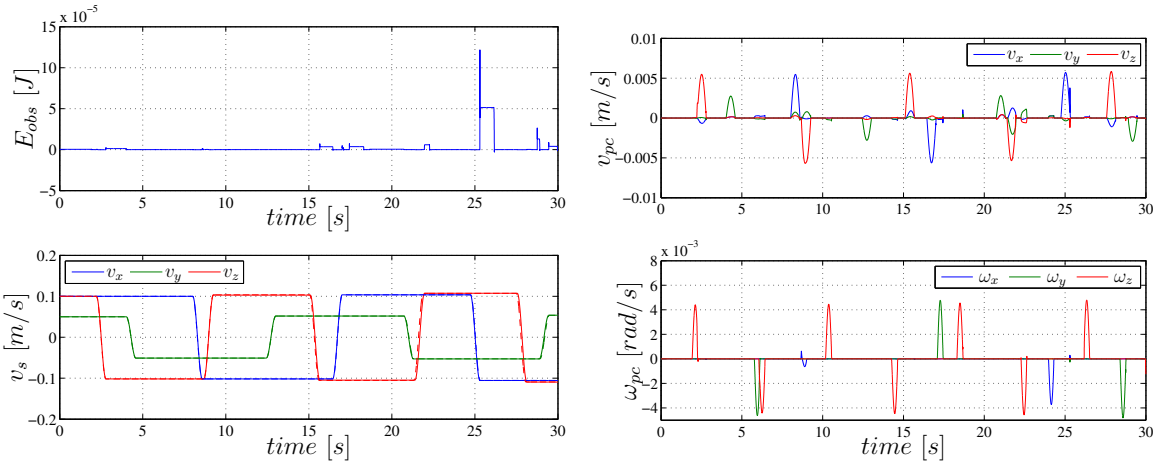
This property is very appealing from a computational point of view but it is necessary to verify if the required assumptions are always satisfied for the given case. From (3.22), it can be seen that $\boldsymbol{A}^T$ is a row matrix. The only case in which $\boldsymbol{A}$ has not full rank is when all the components of the force $\boldsymbol{F_o}$ are zero. This can happen only when the robot does not move since $\boldsymbol{F_o}$ considers also the dynamics of the robot. In this case no energy can be produced and no damping is necessary. From (3.22) we can see that $\boldsymbol{P}$ is a diagonal matrix. If all the components of the control force are different from 0, then $\boldsymbol{P} > 0$. In case $F_i = 0$ for some $i = 1, \ldots, n$, then the corresponding elements on the diagonal of $\boldsymbol{P}$ are zero and, consequently, $\boldsymbol{P}$ becomes positive semi-definite. Nevertheless, if $F_i = 0$ no dissipation occurs along the i^{th} component and the corresponding term on the function to minimise is zero. Thus, it is possible to safely set $\beta_i = 0$. In order to determine the other components of β a reduced minimization problem, where only the non-zero components of $\boldsymbol{F}$ are considered, can be built out of (3.23). It is easy to see that the $\boldsymbol{A}$ and $\boldsymbol{P}$ matrices of the reduced problem satisfy the assumptions required for having a unique solution.

To sum up, the vector $\boldsymbol{x^*}$ represents the optimal solution for the minimization problem and the values are the coefficients of the matrix $\boldsymbol{\beta}$ in (3.14). Simulations and experiments are performed also with this approach to validate the method.

3.5.1 Results: Simulations and Experiments

Simulations

Considers the Example 2 where the passivity control without optimal damping created over-dissipation as shown in Fig. 3.9. For the same example, the optimal passivity control is applied in order to avoid over-damping while keeping performance. The optimal damping value of the PC acting on velocity, results in a velocity correction shown in Fig. 3.11b, where $\boldsymbol{V_{pc}} = \boldsymbol{\beta F_o}$. This leads the velocity of the simulated satellite closer to the ideal one as imposed by the minimization problem and shown in Fig. 3.11a (bottom). In particular, the passivity proof which guarantees stability with the proposed approach (i.e. the sum of the observed energy and the energy due to the passivity control) is shown in Fig. 3.11a (top). It is worth to note that the magnitude of energy shown in the passivity proof is in the order of 10^{-4} J respecting



(a) Ideal velocity (dashed line) and actual linear velocity of the satellite (solid line).

(b) Velocity correction of the optimized passivity control, $V_{pc} = (v_{pc}, \omega_{pc})$

Figure 3.11: Passivity control with optimal damper avoids over-damping action.

the equality constraint imposed in the minimization problem. This means that the active behaviour is now solved by virtue of the optimal damping that modulates the PC in an optimal manner. It is worth also to compare the effect of using the optimized passivity control in Fig. 3.11a with the passivity control without optimization (see Fig. 3.9).

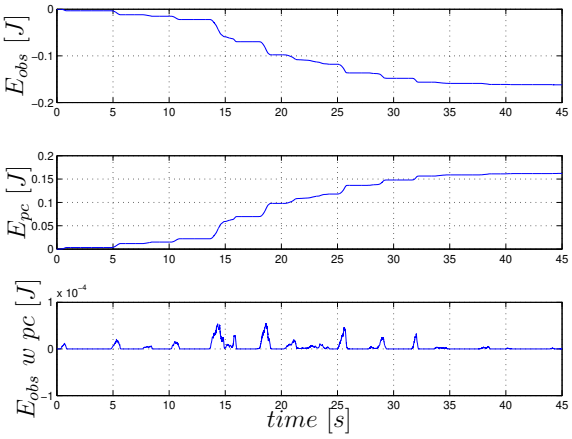
Experiments

In the experiments, a higher time delay of 20 ms is considered in order to prove the effectiveness of the proposed method. Fig. 3.12a (top) shows the energy observed in the network during the experiment which results to be negative. This energy is corrected by the optimal PC providing the exact correction in energy (E_{pc}). It leads to a stable behaviour as the passivity proof shows at the bottom of Fig. 3.12a. Fig. 3.12b shows the linear velocity correction due to the optimal damping and the forces due to the forces estimator. Similarly, Fig. 3.12c shows the angular velocity correction and the torque due to the forces estimator.

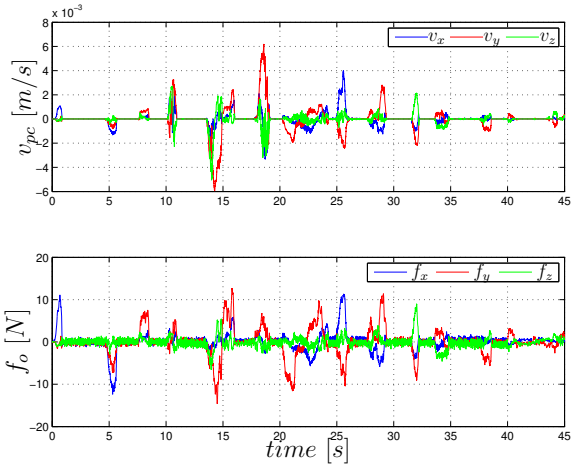
For completeness the velocity of robot, which simulates the satellite is reported in Fig. 3.12d. The results prove the effectiveness of the proposed method in rendering satellite dynamics on a position controlled robot with the proposed performance oriented method. It can be also concluded that the method can deal with sensor noise as shown by the experiments.

3.6 Discussion

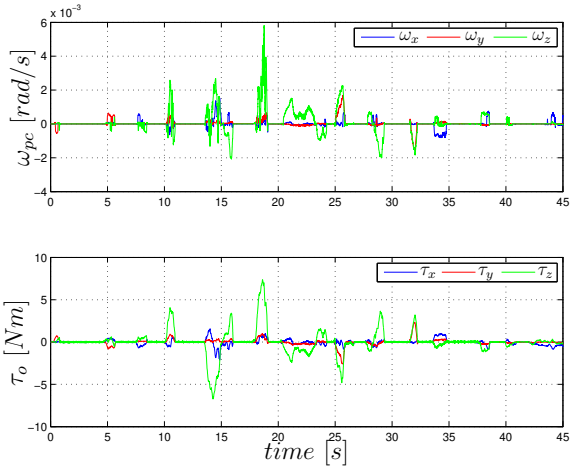
In Chapter 3, the control strategy to achieve high fidelity dynamics simulation rendered on admittance-controlled robotic facilities has been proposed. In particular, stability issues and performance degradation due to intrinsic latencies found in robot



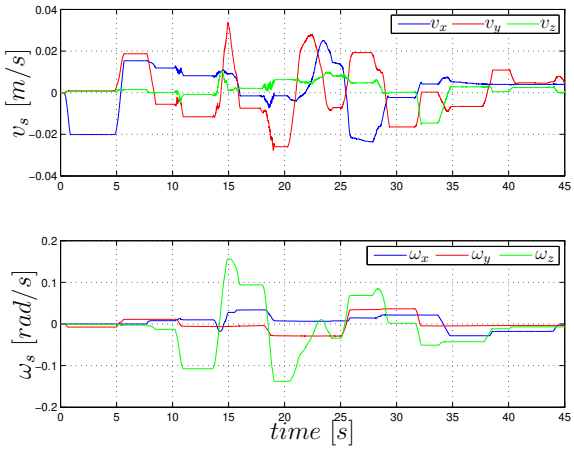
(a) Observed energy, energy due to the optimized PC and passivity proof.



(b) Optimal velocity correction and forces estimated.



(c) Optimal angular velocity correction and torque estimated.



(d) Linear and angular velocity of the satellite.

Figure 3.12: Experiment results: Energy, velocity correction, forces estimator $F_0 = (f_0, \tau_o)$ and motion of the satellite.

controllers have been analysed. The proposed method identifies the sources of intrinsic instability and counteracts the destabilising effects using the passivity criteria. Furthermore, in order to implement the dynamics accurately, the damping process is formulated as an optimization problem. The optimized approach guarantees stability through passivity and preserves the performance through the use of an optimal damping. Hence, over-dissipation can be avoided and the system becomes less conservative. Performance and effectiveness of the method are shown in simulation and verified experimentally on different robotic facilities equipped with force-torque sensor.

CHAPTER 4

ENERGY-BASED CONTROL FOR DISCRETE-TIME INTEGRATION

4.1 Introduction

A further issue tackled in the thesis is the methodology by which the dynamics model is discretized. The states of motion are achieved by integrating a dynamics model, which commands the set-points to the industrial robot during the satellite motion simulation. The integration methods proposed in literature are not suitable for simulation with hardware-in-the-loop because they are implicit. Thus, they require an iterative solution, which might be prohibitive for the real-time implementation required by the robot. Therefore, explicit integration techniques are exploited (e.g. Euler integration) since these are simple, fast and suitable for a real-time implementation. Nevertheless, it is known (see e.g. [SSF07]) that standard integration techniques for implementing a dynamics subject to external forces, lead to a generation of energy and, therefore, to a non-physical behaviour causing position drifts. This undesired behaviour becomes more evident for large sampling time, which is common in industrial robots, where the control frequency is usually low.

In this chapter, the energy drift due to the Euler integration method is analysed for a rigid body dynamics and an energy-based control is developed in order to restore the passivity. The energy drift represents the deviation from the physical behaviour and it is due to the discrete-time integration. Firstly, the energy which causes the drift is identified and an analysis with its continuous-time counterpart is performed. Secondly, an explicit and passive integrator is developed for meeting the passivity constraint. The effectiveness of the proposed integrator is validated with simulations and experiments.

Related Works

Standard discretization techniques destroy the geometric and energetic properties, which characterise the physics of a body [BC98a]. The field of geometric integration tackles the development of numerical integration methods that preserve the energetic properties and/or other geometric structures (e.g. symplectic form) of the system to be integrated. Several methods have been developed over the years (see e.g. [HWL06, BC16, OB08]). Energy-preserving integrators based on momentum conservation [BU07] are also available in the literature. However, geometric integrators concern mainly isolated systems (e.g. astronomical systems) or systems with some dissipation [MS11] and the interaction with the environment is not considered. In [SSvdSF05], an implicit integrator based on port-Hamiltonian modelling, and considering external interaction, has been proposed. In [LH08] a fast but implicit and variable rate integration strategy for implementing a mass-spring-damper systems is illustrated. The integration methods proposed in literature are implicit. Thus, they require an iterative solution, which is prohibitive for the real-time implementation. [BC98a] shows that standard and explicit integrators do not ensure passivity, therefore, a more complex and harder to implement integration strategy has to be sought.

Within this context, a fast and explicit passive integrator for simulating the dynamics of a rigid body has been proposed by the author of this thesis and it is presented in this chapter. The contribution refers to the author publications [DSBAS17] and [DSAS17].

4.2 Problem Statement

The problem related to the discrete-time integration is analysed in this section. In order to isolate the effects generated by the discrete-time integrator, further sources of activity (e.g. time-delay in the control loop) are not considered. However, they will be included in the unified architecture presented in the next chapter.

Fig. 4.1 shows the architecture of a generic robotic simulator with the model-based satellite dynamics and it was presented already in Sec. 2.3. However, the focus here is on the discrete time integration (dashed box in Fig. 4.1), which includes the block $T\sum$ by means of discrete integration with sampling time T.

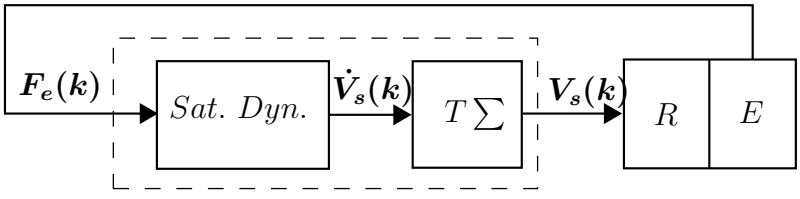


Figure 4.1: Admittance architecture with the desired dynamic. R is the robot, E is the environment, *Sat. Dyn.* is the force-acceleration model of the dynamics to implement, $T\Sigma$ is the discrete integrator with time step T.

The total wrench $\boldsymbol{F_e} = (\boldsymbol{f_e}, \boldsymbol{\tau_e}) \in \mathbb{R}^6$ is the input to the desired dynamics (in the dashed box) where the acceleration $\dot{\boldsymbol{V}}_{\boldsymbol{s}}$ is computed and discretely integrated at sampling time T. Thus, $\boldsymbol{V_s} = (\boldsymbol{v_s}, \boldsymbol{\omega_s}) \in \mathbb{R}^6$ is the final twist velocity vector of the model-based dynamics sent to the robot. The model-based dynamics was presented in (2.1), (2.2) and it is reported also here for completeness, as follows:

$$\dot{\boldsymbol{v}}_{\boldsymbol{s}} = \boldsymbol{M}^{-1}\boldsymbol{f}_e, \tag{4.1}$$

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{s}} = \boldsymbol{I}^{-1}(\boldsymbol{I}\boldsymbol{\omega}_{\boldsymbol{s}} \times \boldsymbol{\omega}_{\boldsymbol{s}} + \boldsymbol{\tau}_e) = \boldsymbol{I}^{-1}(\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega}_{\boldsymbol{s}})\boldsymbol{\omega}_{\boldsymbol{s}} + \boldsymbol{\tau}_e). \tag{4.2}$$

The following analysis is performed considering the Euler integrator because this method is usually exploited in applications with industrial robots. The reason is given by the fact that the method is fast, explicit and it can be implemented on a real-time operating system. The analysis is based on the energy characteristics of the system, therefore, the total energy $H(\boldsymbol{v_s}, \boldsymbol{\omega_s})$ of the rigid body dynamics is introduced and defined as follows:

$$H(\boldsymbol{v_s}, \boldsymbol{\omega_s}) = H_t(\boldsymbol{v_s}) + H_r(\boldsymbol{\omega_s}), \tag{4.3}$$

where

$$H_t = \frac{1}{2}\boldsymbol{v}_{\boldsymbol{s}}^T\boldsymbol{M}\boldsymbol{v_s}, \tag{4.4}$$

$$H_r = \frac{1}{2}\boldsymbol{\omega}_{\boldsymbol{s}}^T\boldsymbol{I}\boldsymbol{\omega_s}, \tag{4.5}$$

being H_t the translational kinetic energy and H_r the rotational energy.

In order to compute the discrete velocity set-point to be sent to the robot, the dynamics needs to be integrated in discrete time using, for example, the standard Euler method. Therefore, for the translational dynamics in (4.1) the following discrete system is obtained:

$$\boldsymbol{v_s(k)} = \boldsymbol{v_s(k-1)} + T\boldsymbol{M}^{-1}\boldsymbol{f}_e\boldsymbol{(k-1)}. \tag{4.6}$$

An angular velocity set-point is also required to the robot in order to simulate a complete spacecraft dynamics. Therefore, similarly to the translational dynamics, (4.2) needs to be integrated and according to the Euler method will result:

$$\boldsymbol{I}(\boldsymbol{\omega_s(k)} - \boldsymbol{\omega_s(k-1)})T^{-1} = \boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})\boldsymbol{\omega_s(k-1)} + \boldsymbol{\tau_e(k-1)}, \tag{4.7}$$

consequently, the angular velocity set-point to be commanded to the robot is:

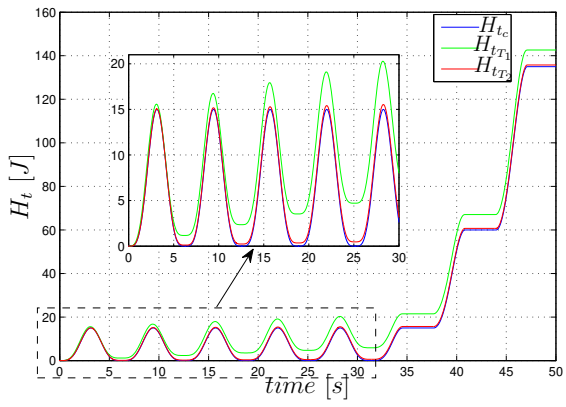
$$\boldsymbol{\omega_s(k)} = \boldsymbol{\omega_s(k-1)} + \boldsymbol{I}^{-1}T\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})\boldsymbol{\omega_s(k-1)} + T\boldsymbol{I}^{-1}\boldsymbol{\tau_e(k-1)}. \tag{4.8}$$

The energy variation for the translational and rotational dynamics is due to the power exchanged with the environment and in case of free motion (i.e. $\boldsymbol{f_e} = 0$ and $\boldsymbol{\tau_e} = 0$), the velocity of the system is constant over time. Thus, in this simple case, standard Euler integration is energetically well posed since it allows the discretized dynamics to behave in a physically consistent way independent of the sampling time. However, this condition does not hold anymore in case of interaction with the environment, as can be seen in the following example and later analytically analysed.

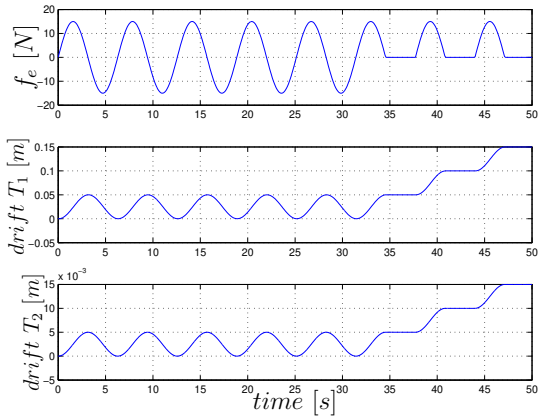
Example 3 *A rigid body with mass $M = 30\ kg$ and inertia $I = diag(18, 20, 22) Kgm^2$ is subjected to external forces and torques. The force profile is shown at the top of Fig. 4.2b and the torque profile is shown in Fig. 4.3b (top). Considers the integration of the rigid body dynamics in continuous time and discrete time with the Euler integrator for sampling times: $T_1 = 0.1\ s$ and $T_2 = 0.01\ s$.*

Due to external interaction, the total energy of the system increases while using the Euler integration method. In particular, the increase in energy with respect to the continuous time integrator is shown in Fig. 4.2a, where H_{t_c} is the translational energy calculated in continuous time (4.4), $H_{t_{T_1}}$ and $H_{t_{T_2}}$ are the same translational energies but calculated at sampling time T_1 and T_2, respectively.

As it can be seen, the increase in energy depends on the sampling time and this leads to position drifts. The drift due to the integration with T_1 reaches $0.05\ m$ (when the force profile acts between $0\ s$ and $32\ s$) and $0.15\ m$ between $32\ s$ and $50\ s$, (see Fig. 4.2b middle). Also for the case with T_2, the drift appears. Since the sampling time is smaller, it results in a drift 10 times lower, as shown in Fig. 4.2b bottom. This drift shows that the interaction causes inconsistency in reproducing the desired dynamics.



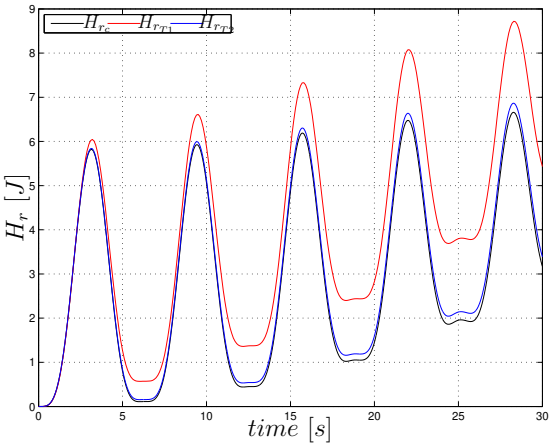
(a) Energy ($H_{t_{T_1}}$) calculated with sampling time T_1 and ($H_{t_{T_2}}$) calculated with sampling time T_2. H_{t_c} is the continuous-time energy.



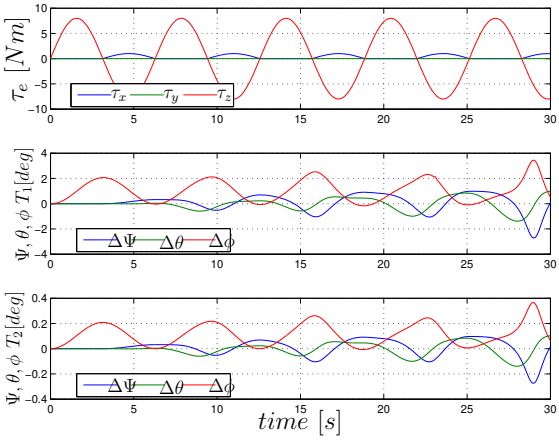
(b) Force profile, drift in position due to the discrete integration with sampling times T_1 and T_2.

Figure 4.2: Discrete integration causes drift in energy for the translational dynamics.

Unlike the translational dynamics, where the evolution is due only to the external force, the integration of the rotational dynamics is function also of non-linear coupling as evident from (4.8). This makes the numerical integration a more challenging problem. Considering Example 3, a comparison of the angular energy calculated in continuous time, (i.e. H_{r_c} in (4.5)) and the respective energy calculated in discrete times ($H_{r_{T_1}}, H_{r_{T_2}}$) is shown in Fig. 4.3a. As can be seen, the energy in discrete time increases and this extra energy generates drift in orientation (see Fig. 4.3b) as previously stated for the translational dynamics.



(a) Comparison of energy calculated in continuous time (H_{r_c}) and discrete sampling times T_1 ($H_{r_{T_1}}$) and T_2 ($H_{r_{T_2}}$).



(b) Applied torques (top) and drift in roll, pitch and yaw considering sampling time T_1 (middle) and T_2 (bottom).

Figure 4.3: Discrete integration causes drift in energy for the rotational dynamics.

Orientation and position drifts may lead the robot to interact with unforeseen objects that produce further (drifted) behaviours leading to a deteriorated performance of the dynamic simulation and even instabilities. Furthermore, energy drifts will produce an energy inconsistent behaviour. In the next section, the analytical value which causes this energy drift will be identified and later exploited for implementing a passive control action.

4.3 Energy Generated with the Discrete Integrator

The energetic properties of the simulated dynamics can be restored if the active energy is identified. In this section, the active energy causing the drift is analytically found. Firstly for the case of the translational dynamics and later for the rotational dynamics, which is more challenging from an integration point of view.

Translational Dynamics

The kinetic energy can be calculated in discrete-time for the translational dynamics discretized with the Euler method in (4.6). Therefore, substituting (4.6) in (4.4), it will result in the following expression,

$$H_t(k) = \frac{1}{2}\boldsymbol{v_s(k)}^T \boldsymbol{M v_s(k)} = \tag{4.9}$$
$$\frac{1}{2}[\boldsymbol{v_s(k-1)} + T\boldsymbol{M^{-1}f_e(k-1)}]^T \boldsymbol{M}[\boldsymbol{v_s(k-1)} + T\boldsymbol{M^{-1}f_e(k-1)}].$$

Considering that the inertia matrix $\boldsymbol{M}$ is symmetric and positive definite, i.e. $(\boldsymbol{M^{-1}})^T = (\boldsymbol{M^T})^{-1} = \boldsymbol{M^{-1}}$, equation (4.9) can be rewritten as:

$$H_t(k) = H_t(k-1) + T\boldsymbol{v_s(k-1)}^T\boldsymbol{f_e(k-1)} + \underbrace{\frac{1}{2}T^2\boldsymbol{f_e(k-1)}^T\boldsymbol{M}^{-1}\boldsymbol{f_e(k-1)}}_{\Delta H_t(k)}. \tag{4.10}$$

Therefore, the discrete energy in (4.10) is composed of three terms, the first is the integral part, the second is the power provided to the port $(\boldsymbol{f_e}, \boldsymbol{v_s})$ and the third one (ΔH_t) is the extra energy term generated by the Euler integration process, which does not represent a physical and passive behaviour. Indeed, the variation of energy should be due only to the power provided through the port, i.e. $\boldsymbol{v_s(k-1)^T f_e(k-1)}$ and the extra energy term defined as

$$\Delta H_t(k) = \frac{1}{2}T^2\boldsymbol{f_e(k-1)}^T\boldsymbol{M}^{-1}\boldsymbol{f_e(k-1)} \tag{4.11}$$

is due to Euler integration process.

This extra energy causes energy-inconsistency and drift that make the reproduced dynamics diverge from the ideal behaviours as discussed in Sec. 4.2. Further, as evident from (4.10), the discrete dynamics is not passive and, therefore, it may happen that the system becomes unstable during interactions with the environment [SSF07].

Rotational dynamics

The coupling due to the rotational dynamics renders the integration process more challenging. This can be seen in the following expression where the discrete rotational dynamics (4.8) is substituted in $H_r(k)$ given in (4.5), which results in

$$\begin{aligned} H_r(k) = \frac{1}{2}\{&\boldsymbol{\omega_s(k-1)}^T\boldsymbol{I}\boldsymbol{\omega_s(k-1)} + T\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega(k-1)})\boldsymbol{\omega_s(k-1)} \\ &+ T\boldsymbol{\omega_s(k-1)}^T\boldsymbol{\tau_e(k-1)} + T\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})^T\boldsymbol{\omega_s(k-1)} \\ &+ T^2\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})^T\boldsymbol{I}^{-1}\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})\boldsymbol{\omega_s(k-1)} \\ &+ T^2\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})^T\boldsymbol{I}^{-1}\boldsymbol{\tau_e(k-1)} \\ &+ T\boldsymbol{\tau_e(k-1)}^T\boldsymbol{\omega_s(k-1)} + T^2\boldsymbol{\tau_e(k-1)}^T\boldsymbol{I}^{-1}\boldsymbol{S}(\boldsymbol{I}\boldsymbol{\omega_s(k-1)})\boldsymbol{\omega_s(k-1)} \\ &+ T^2\boldsymbol{\tau_e(k-1)}^T\boldsymbol{I}^{-1}\boldsymbol{\tau_e(k-1)}\}, \end{aligned} \tag{4.12}$$

where it has been considered that $(\boldsymbol{I}^{-1})^T = (\boldsymbol{I}^T)^{-1} \equiv \boldsymbol{I}^{-1}$. Furthermore, exploiting the properties of the skew-symmetry of matrix $\boldsymbol{S}(\cdot)$, (4.12) can be simplified as[1],

$$H_r(k) = H_r(k-1) + T\boldsymbol{\omega_s(k-1)}^T\boldsymbol{\tau_e(k-1)} + \underbrace{\frac{1}{2}T^2\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S(k)}^T\boldsymbol{I}^{-1}\boldsymbol{S(k)}\boldsymbol{\omega_s(k-1)}}_{\Delta H_{r1}(k)} + \underbrace{\frac{1}{2}T^2\boldsymbol{\tau_e(k-1)}^T\boldsymbol{I}^{-1}\boldsymbol{\tau_e(k-1)}}_{\Delta H_{r2}(k)}. \tag{4.13}$$

As for the translational case, the discretized rotational dynamics leads to a non-physical behaviour. Indeed, extra energy terms are identified in (4.13), which are,

$$\Delta H_r(k) = \Delta H_{r1}(k) + \Delta H_{r2}(k) = \frac{1}{2}T^2\boldsymbol{\omega_s(k-1)}^T\boldsymbol{S(k)}^T\boldsymbol{I}^{-1}\boldsymbol{S(k)}\boldsymbol{\omega_s(k-1)} + \frac{1}{2}T^2\boldsymbol{\tau_e(k-1)}^T\boldsymbol{I}^{-1}\boldsymbol{\tau_e(k-1)}. \tag{4.14}$$

In particular, the energy variation should be due only to the energy provided through the power port, i.e. $\boldsymbol{\omega_s(k-1)}^T\boldsymbol{\tau_e(k-1)}$. However, unlike the translational case, two quota of energy are found. ΔH_{r1} which results from the integration of the rotational coupled dynamics, and ΔH_{r2} which results from the integration of the external torque $\boldsymbol{\tau_e}$ (similar to ΔH_t for the translational case).

The extra energy terms found for the translational and rotational dynamics, namely $\Delta H_t(k)$ in (4.11) and $\Delta H_r(k)$ in (4.14), are responsible for the drifts discussed in Sec. 4.2. These terms can be computed in real time and can be used for adjusting the output velocity of the Euler integrator.

In the following subsection, an analysis will reveal how close to the continuous time energies are these identified values in discrete time.

4.3.1 Relation Between Continuous and Discrete Dynamics

The extra energy terms ΔH_t and ΔH_r represent the energy produced when a continuous-time dynamics is discretized, namely it is the difference between the discrete and the continuous energy. Notice that ΔH_t and ΔH_r have a similar structure, i.e. are functions of a force or torque. In this section, a numerical and graphical analysis is performed considering ΔH_t in discrete and continuous time. However, similar results can be derived for ΔH_r.

Firstly, a numerical analysis is performed for the system described in Example 3. The body velocity is derived using both continuous and discrete time integration, where E_c is the energy in continuous time and E_d in discrete time. The difference between energy increments of the dynamic system in one sampling cycle between the two integration methods is $\Delta E = E_d - E_c$. The analytical value of the additional energy due to discrete integration ΔH_t is also calculated in the numerical analysis. Fig. 4.4 shows the difference between ΔE and ΔH_t for different sampling rates. It can be seen that the difference is small and it tends to zero as the sampling time

[1]for the sake of brevity, the dependency of $\boldsymbol{I\omega_s(k-1)}$ in $\boldsymbol{S}$ will be omitted.

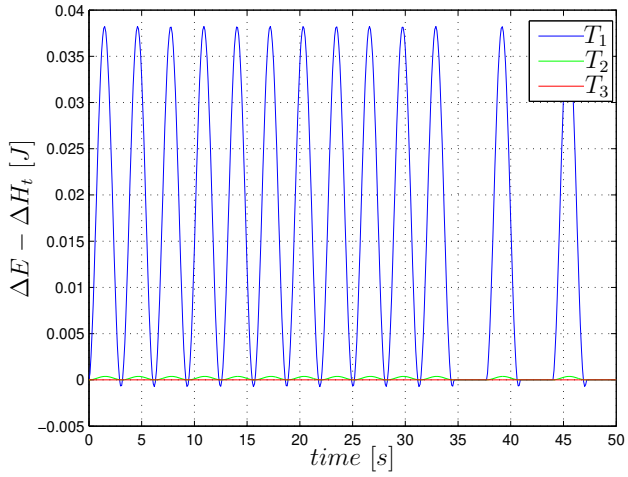


Figure 4.4: Difference between analytical (ΔE) and identified values of additional energy (ΔH_t). Sampling times are $T_1 = 0.1s$, $T_2 = 0.01s$, $T_3 = 0.001s$.

tends to zero. The difference in energy is due to the loss of information during the discretization process and it can not be avoided.

The reason for the difference between ΔE and ΔH_t is analysed in a graphical representation proposed in Fig. 4.5 where the power plots for both continuous (dashed curve) and discrete (samples) dynamic systems are shown. This analysis can help to analytically quantify the difference between ΔE and ΔH_t. The difference between the energy increase per sampling cycle (in continuous and discrete) is analysed in the lower part of Fig. 4.5 (positive, increasing power in A and negative, decreasing power in B). In part A, the power of the continuous system varies linearly between the samples $k-1$ and k from $P(t-T)$ to $P(t)$ since the input force $f_e(k-1)$ is constant during this time[2]. The area of the shaded regions (quadrilateral *acde*) is the extra energy ΔE in the sampling time T produced by the discrete system with respect to the continuous one. This area is the sum of the areas of the rectangle *abde* and the triangle *bcd*. If $Area(*)$ function is defined as the area of the polygon $(*)$, ΔE is given by:

$$\begin{aligned}
\Delta E &= Area(acde) = Area(abde) + Area(bcd) \\
&= [P(k-1) - P(t)]T + \frac{1}{2}T[P(t) - P(t-T)] \\
&= f_e(k-1)T[v_s(k-1) - v_s(t)] + \frac{1}{2}Tf_e(k-1)[v_s(t) - v_s(t-T)],
\end{aligned}$$

where $[v_s(t) - v_s(t-T)] = \frac{f_e(k-1)}{m}T$. Then, it is possible to write:

[2]the bold notation will be omitted in the analysis because it considers the component of the vector.

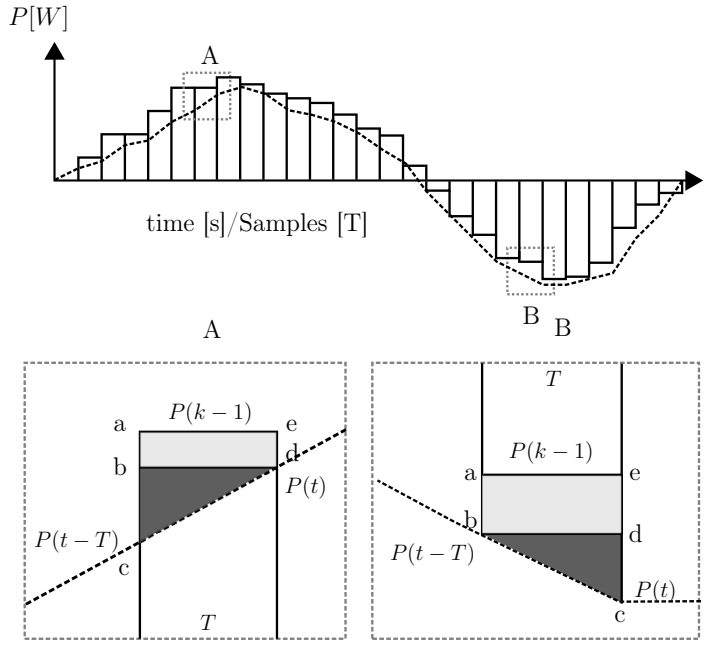


Figure 4.5: Power difference between discrete and continuous time domains.

$$\begin{aligned}\Delta E &= f_e(k-1)T[v_s(k-1) - v_s(t)] + \frac{1}{2}Tf_e(k-1)\frac{f_e(k-1)}{m}T \\ &= f_e(k-1)T[v_s(k-1) - v_s(t)] + \underbrace{\frac{T^2 f_e(k-1)^2}{2m}}_{\Delta H_{t,i}} \\ &= f_e(k-1)T[v_s(k-1) - v_s(t)] + \Delta H_{t,i},\end{aligned} \tag{4.15}$$

which results in a difference:

$$\Delta E - \Delta H_{t,i} = f_e(k-1)T[v_s(k-1) - v_s(t)]. \tag{4.16}$$

Similarly, in part B of Fig. 4.5, ΔE is given by the sum of the areas of rectangle *abde* and triangle *bcd*, which results in:

$$\Delta E - \Delta H_{t,i} = f_e(k-1)T[v_s(k-1) - v_s(t-T)]. \tag{4.17}$$

Equations (4.16) and (4.17) represent analytically the error between ΔE and ΔH_t (shown in Fig. 4.4). The equations clearly show that as the sampling rate increases, ΔE gets closer to ΔH_t since $v_s(k-1)$ gets closer to $v_s(t)$ and $v_s(t-T)$. In the graphical analysis, this turns to be a smaller area of rectangle *abde*.

The analysis shows that there is always a difference between what it is possible to estimate in real time, namely ΔH_t or the equivalent ΔH_r, and the real difference between the energetic behaviour in the discrete case and in the real case. The error is described by (4.16) and (4.17) and it is due to the loss of information related

to the discretization process, which can not be avoided. Such a difference gets smaller as the sample time gets lower. However, the output velocity can be adjusted for recovering the passivity of the discrete model and it brings several advantages. First, a physical behaviour of the discrete dynamics is ensured. The evolution will be close to the ideal one in the limits reported in (4.16). Second, a stable interactive behaviour is achieved thanks to the passivity of the discrete dynamics.

4.4 Passivity-based Integration Method

The passive integrator is designed to re-establish the energy behaviour by dissipating the extra energy terms, namely ΔH_t and ΔH_r, which have been identified. The problem associate to the discrete-time integrator can be represented in a two-ports network analogy proposed in Fig. 4.6, where E_c is the energy in continuous time which, due to the discretization, will assume the value of E_d. The energy associated to this two-port network can be monitored by introducing the following energy observer,

$$E_{obs}(k) \;=\; E_{obs}(k-1) + E_c(k) - E_d(k) = E_{obs}(k-1) - \Delta E(k). \qquad (4.18)$$

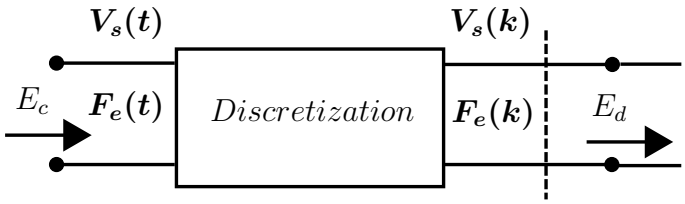


Figure 4.6: Two-port system with continuous energy (E_c) and discrete energy (E_d).

The energy observer, as defined above, relies on the knowledge of E_c (continuous time energy), which is not available in a real-time operating system. Therefore, the power-correlated variable at this port are not available to perform energy considerations. However, the energy between continuous and discrete time (namely ΔH_t, ΔH_r) has been analytically identified with the limits described in (4.16), (4.17). Hence, the two-ports network system can be seen as a one-port network, which produces the extra energy (ΔH_t and ΔH_r) as shown in Fig. 4.7. The active energy will

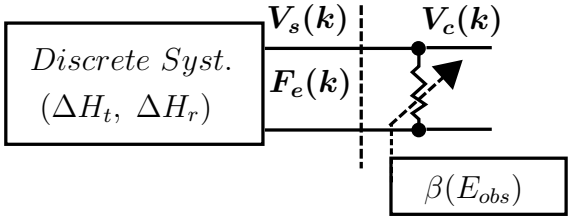


Figure 4.7: One-port analogue discrete system of Fig. 4.6.

be dissipated by the time-varying damping, β, later defined and it is represented as a variable resistor.

Passive integration scheme for translational dynamics

The extra energy term produced by the integration and reported in (4.11) is always positive and, therefore, it always corresponds to a production of energy. It is possible to consider this term in a passivity observer in order to detect a loss of passivity. Thus, it is possible to rewrite (4.18) as:

$$E_{obs_2}(k) = E_{obs_2}(k-1) - \Delta H_t(k) + \boldsymbol{f_e(k-1)}^T \boldsymbol{\beta_1(k-1)} \boldsymbol{f_e(k-1)} T, \tag{4.19}$$

where $\Delta E(k)$ has been approximated with $\Delta H_t(k)$, which represents the energy to dissipate. The last term in (4.19) is the energy dissipated by the passivity controller, which will act with a time-varying damping matrix $\boldsymbol{\beta_1}$, later defined and the relative energy is monitored by the energy observer. The energy observer will measure active energy as soon as there is an external force (which causes ΔH_t). Being the output of the integrator a velocity, the passivity controller (PC) will act in admittance causality as follows:

$$\boldsymbol{v_{pc}(k)} = \boldsymbol{\beta_1(k)} \boldsymbol{f_e(k)}. \tag{4.20}$$

Therefore, a correction to the output velocity of the admittance dynamics, i.e. the velocity commanded to the robot, is given as follows:

$$\boldsymbol{v_c(k)} = \boldsymbol{v_s(k)} - \boldsymbol{v_{pc}(k)}, \tag{4.21}$$

where $\boldsymbol{v_s(k)}$ is achieved from (4.6). The switching action of the passivity control is given when the passivity condition ($E_{obs_2}(k) < 0$) is violated and it is considered in the matrix $\boldsymbol{\beta_1(k)} \in \mathbb{R}^{3\times 3}$. This matrix is chosen to have a diagonal form such that $\boldsymbol{\beta_1} = diag(\beta_{1,1}, \beta_{1,2}, \beta_{1,3})$. Thus, the coefficient of the damping matrix are defined as:

$$\beta_{1,i}(k) = \begin{cases} -\frac{E_{obs_{2,i}}(k)}{f_{e,i}^2(k)T} & E_{obs_{2,i}}(k) < 0 \\ 0 & else, \end{cases} \tag{4.22}$$

where $i = 1, 2, 3$ is the $i-th$ component. The time-varying damping will provide the exact amount of damping required to restore the energy behaviour of the system. It is worth to note that (4.22) is free of singularity because, when the damping injection is activate, i.e. $E_{obs_{2,i}} < 0$, there is always a force available $f_{e,i}$ which causes the generation of $\Delta H_{t,i}$.

The value of energy $E_{obs_{2,i}}$ in (4.22), can be calculated from (4.19) and it results in:

$$E_{obs_{2,i}}(k) = E_{obs_{2,i}}(k-1) - \Delta H_{t,i}(k) + \beta_{1,i}(k-1) f_{e,i}^2(k-1) T, \tag{4.23}$$

where $\Delta H_{t,i}(k) = \frac{T^2 f_{e,i}^2}{2M_{i,i}}$. It can be proved that the energy balance of the energy observer defined in (4.19) still holds. Indeed,

$$E_{obs_2}(k) = \sum_{i=i}^{3} E_{obs_{2,i}}(k) \tag{4.24}$$

where,

$$E_{obs_2}(k) = -\Delta H_{t,i}(k) + \sum_{m=1}^{k}\sum_{i=1}^{3} \beta_{1,i}(m-1) f_{e,i}^2(m-1)T. \tag{4.25}$$

Passive integration scheme for rotational dynamics

The coupling of the rotational dynamics renders the design of the passive integrator more complex. The power port to be considered is given by the pair $(\boldsymbol{\tau_e(k)}, \boldsymbol{\omega_s(k)})$. Similarly to (4.20), it would be possible to implement a dissipative action on this port (i.e. $\boldsymbol{\beta_2(k)\tau_e(k)}$) with a passivity controller (PC), as shown in Fig. 4.8a. However, only when the robot interacts with the environment (i.e. $\boldsymbol{\tau_e(k)} \neq 0$), it will be possible to dissipate the desired amount of energy (ΔH_{r2} and ΔH_{r1}). But, in case of free motion, i.e. $\boldsymbol{\tau_e(k)} = 0$, the dissipation is not straightforward. In particular, the value ΔH_{r2} will result to be zero but some energy can be generated by the coupling which generates ΔH_{r1} and it can not be dissipated because $\boldsymbol{\tau_e(k)} = 0$.

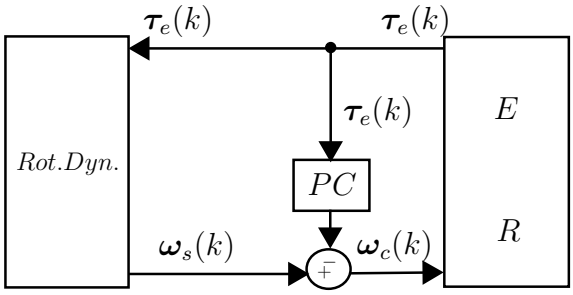
Therefore, the natural port $(\boldsymbol{\tau_e}, \boldsymbol{\omega_s})$ is not sufficient for dissipating all the energy produced by the discrete integration for the rotational case. To overcome this problem, a fictitious port where to apply the PC can be designed in order to re-established an energetic behaviour both during contacts and in free motion. The fictitious port is represented in Fig. 4.8b and it defined as $(\boldsymbol{\chi}, \boldsymbol{\omega}_s)$, where

$$\boldsymbol{\chi(k)} = \boldsymbol{\tau_e(k)} + \underbrace{\frac{1}{2} T^2 \boldsymbol{\omega_s(k)}^T \boldsymbol{S}(\boldsymbol{I\omega_s(k)})^T \boldsymbol{I}^{-1} \boldsymbol{S}(\boldsymbol{I\omega_s(k)})}_{\boldsymbol{\tau_f}}. \tag{4.26}$$

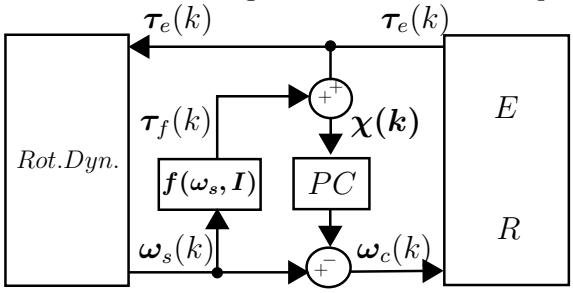
The torque $\boldsymbol{\chi} \in \mathbb{R}^3$ is composed of the external torque $\boldsymbol{\tau_e}$ and a fictitious torque $\boldsymbol{\tau_f} = f(\boldsymbol{\omega_s}, \boldsymbol{I})$ which explicitly considers the contribution of ΔH_{r1} at the port level. Hence, $\boldsymbol{\chi}$ can observe the real and fictitious torques and it can be used for performing a passive control action. In particular, the term $\boldsymbol{\tau_e}$ takes into account the presence of ΔH_{r2} while the term $\boldsymbol{\tau_f}$ takes into account the presence of ΔH_{r1}. Therefore, the PO/PC approach can be exploited for dissipating the energy produced by the Euler integrator even during free motion. The energy observer monitors the port $(\boldsymbol{\chi}, \boldsymbol{\omega}_s)$ and it is defined as:

$$\begin{aligned} E_{obs3}(k) = {} & E_{obs3}(k-1) - (\Delta H_{r1}(k) + \Delta H_{r2}(k)) \\ & + T\boldsymbol{\chi(k-1)}^T \beta_2(k-1)\boldsymbol{\chi(k-1)}, \end{aligned} \tag{4.27}$$

which includes the energy to be dissipated $(\Delta H_{r1}(k) + \Delta H_{r2}(k))$ and the energy of the passivity control expressed in the last term of (4.27).



(a) Representation of the integrator with PC on the port $(\boldsymbol{\tau_e}, \boldsymbol{\omega_s})$.



(b) Representation of the integrator with PC and fictitious port $(\boldsymbol{\chi}, \boldsymbol{\omega_s})$.

Figure 4.8: Representation of the integrator for the rotational dynamics, E is the environment and R the robot.

Unlike the translational case where the dynamics is independent along its components, the rotational dynamics is coupled. Therefore for the dissipative action a single damping coefficient (β_2) is designed in order to act along all the directions of the angular velocity. Otherwise, compensating in different directions will lead to a distortion of the reproduced dynamics. Then, when the active energy is detected, i. e. $E_{obs3} < 0$, the time-varying damper $\beta_2(k)$ acts as follows:

$$\beta_2(k) = \begin{cases} -\frac{E_{obs3}(k)}{T\|\chi(k)\|^2} & if\ E_{obs3} < 0 \\ 0 & else, \end{cases} \tag{4.28}$$

where $\|\chi(k)\|^2 = \boldsymbol{\chi(k)}^T\boldsymbol{\chi(k)}$. The use of $\boldsymbol{\chi(k)}$ guarantees a singularity-free damping action for the time-varying damping in (4.28). This is given by the fact that $\boldsymbol{\chi(k)}$ is always different than zero even in free-motion, i.e. $\boldsymbol{\tau_e} = 0$.

To modify the output of the integrator, i.e. the velocity $\boldsymbol{\omega_s}$, a velocity modification is provided by the passivity controller as follows:

$$\boldsymbol{\omega_{pc}}(k) = \beta_2(k)\boldsymbol{\chi(k)}, \tag{4.29}$$

which is used for correcting the output of the Euler integrator as:

$$\boldsymbol{\omega_c(k)} = \boldsymbol{\omega_s(k)} - \boldsymbol{\omega_{pc}(k)}, \tag{4.30}$$

where $\boldsymbol{\omega_c(k)}$ represents the velocity commanded to the robot.

A schematic of the passive integrator is shown in Fig. 4.9. The corrected velocity sent to the robot is given by $\boldsymbol{V_c} = (\boldsymbol{v_c}, \boldsymbol{\omega_c})$, where $\boldsymbol{v_c}$ is obtained from (4.21) and

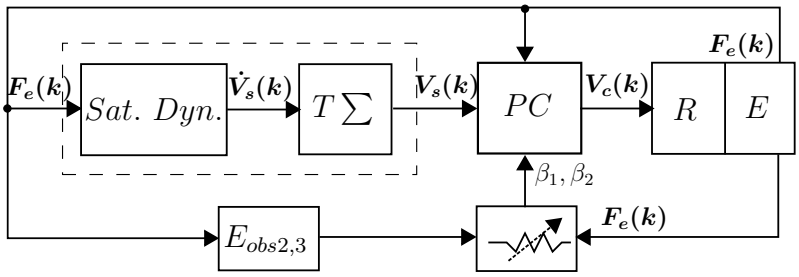


Figure 4.9: The passivity-based integration scheme.

$\boldsymbol{\omega_c}$ from (4.30). The time-varying damping coefficients, β_1, β_2 act accordingly to the activity measured by the energy observers and the applied force (torque). Then, the PC provides the corrected velocity to the robot.

4.5 Results: Simulations and Experiments

The method is firstly verified in simulation where the proposed integration scheme is applied. Further, experiments are carried out on the client robot of the OOS-Sim facility, where the force sensor at its end-effector is exploited to measure the external interaction.

Simulations

The simulations consider the dynamics of a rigid body with the same conditions introduced in Example 3. The passive discrete integrator is applied considering sampling time $T_1 = 0.1s$ and $T_2 = 0.01s$. The simulation results for the translational dynamics using sampling time T_1 and T_2 are shown in Fig. 4.10a and Fig. 4.10b, respectively. The plots at the top show the active energy observed in the system without the action of the PC. The activity is compensated by the PC (see middle plot) by introducing the correction velocity v_{pc}. Thus, the passivity proof is provided at the bottom of Fig. 4.10a and Fig. 4.10b which show that the observed energy has always a non-negative value according to the passivity condition.

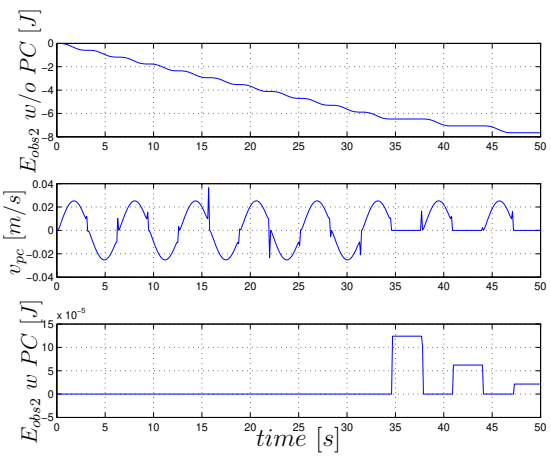
Also for the rotational dynamics the energy observer measures activity, as shown in Fig. 4.11a and Fig. 4.11b considering the integration with sampling time T_1 and T_2. The passivity is restore by the passivity control, which provides a correction (ω_{pc}). The passivity proof is given by E_{obs} w PC shown at the bottom of Fig. 4.11a and Fig. 4.11b.

Energy properties

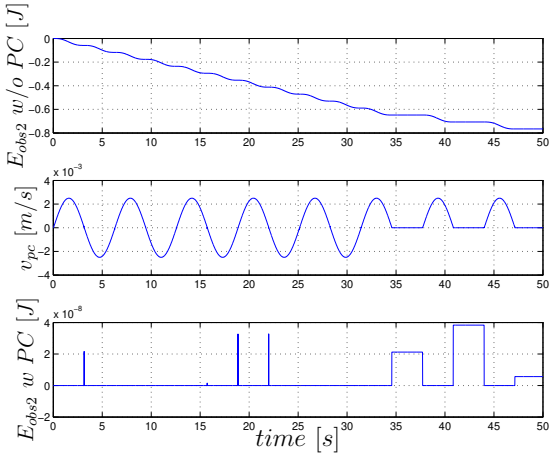
In order to verify that the energy properties are preserved, a comparison with the continuous time integrator is performed and it is shown in Fig. 4.12 and Fig. 4.13 for the translational and rotational dynamics, respectively. H_{tc} is the translational energy and H_{rc} the rotational energy calculated in continuous time. H_{tT1}, H_{tT2} is

the energy calculated as results of the correction with the proposed method with sampling time T_1 and T_2, respectively. The energy drift discussed in Sec. 4.2 are here corrected as shown in Fig. 4.12 for the translational case and Fig. 4.13 for the rotational case. The energy compensated with the proposed passive integrator is close to the continuous-time energy. It is worth to compare Fig. 4.2a with Fig. 4.12 and Fig. 4.3a with Fig. 4.13 to understand the benefit of the method.

The results prove that the discrete dynamics behaves passively in discrete time with the proposed integration method and it preserves the energy properties of the simulated rigid body.

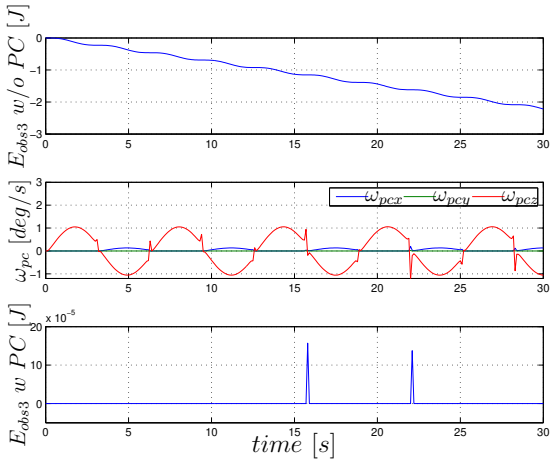


(a) Sim. 1 - E_{obs} without (w/o) PC, velocity corrected by the PC and E_{obs} with (w) PC for $T_1 = 0.1s$.

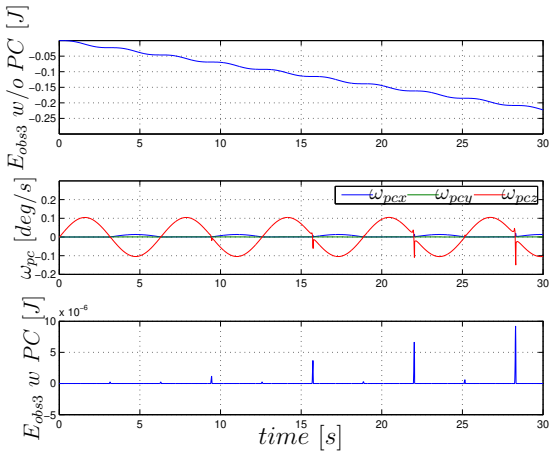


(b) Sim. 2 -E_{obs} without (w/o) PC, velocity corrected by the PC and E_{obs} with (w) PC for $T_2 = 0.01s$.

Figure 4.10: Simulation results for the translational dynamics integrated with T_1 and T_2.



(a) Sim. 1 with T_1 - Energy observed, angular velocity correction and passivity proof.



(b) Sim. 2 with T_2 - Energy observed, angular velocity correction and passivity proof.

Figure 4.11: Simulation results for the rotational dynamics integrated with T_1 and T_2.

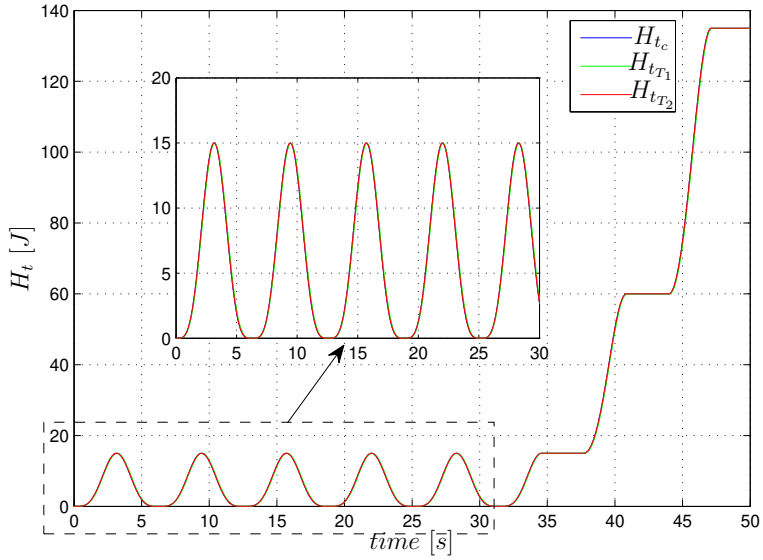


Figure 4.12: Translational energy corrected with the proposed integrator considering sampling time (T_1 and T_2) and comparison with the continuous case (H_{t_c}).

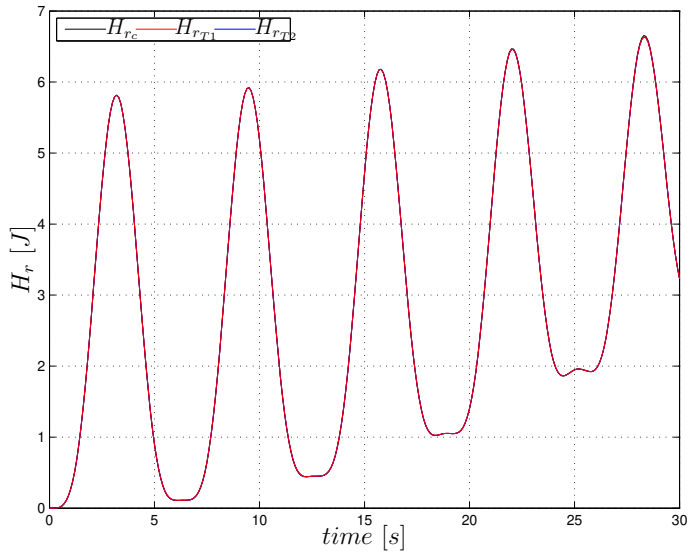


Figure 4.13: Rotational energy corrected with the proposed integrator in continuous time (H_{r_c}) and discrete time ($H_{r_{T1}}$, $H_{r_{T2}}$).

Experiments

The experiments are carried out on the OOS-Sim facility. Plant and torque-force sensor run on a real-time system with a frequency of 250 Hz, thus the considered sampling time of the integrator is 4 ms. The simulated mass considered for the experiments is 250 kg with inertia $diag(18, 20, 22)$ Kgm^2, typical value of a satellite.

Fig. 4.14, Fig. 4.15, Fig. 4.16, show the data along the components (x, y, z) measured during the experiment. Each figure shows the energy observed without

(w/o) PC, the energy with (w) PC, the damping coefficient β_1, the velocity corrected by the PC, the measured force f_e and the relative position of the robot, respectively.

As it can be seen, the E_{obs} w/o the PC results in a negative energy which can produce an active behaviour. However, this active energy is corrected by the PC which commands a velocity correction v_{pc} as a function of the damping coefficient β_1 to preserve the passivity condition. Indeed, the passivity proof of the method is described by the energy observed with the PC in each figure which results to be positive, thus, passive. The difference in the observed energy (and consequently the v_{pc}) for each direction is due to the different magnitudes of applied forces, e.g. along the z the force reaches 60 N, lower values for x and y, therefore, the extra energy term ΔH_t will be different.

For the rotational dynamics, active energy is also measured by the observer as shown at the top of Fig. 4.17. However, this energy is corrected by PC (see Fig. 4.18) which provides the required correction in velocity according to β_2, the time-varying damping. Therefore, the implemented dynamics results to be passive and the body preserves its energy consistency. As it can be seen, system passivity is evidenced by the energy signal which results to be greater than or equal to zero (see Fig. 4.17 bottom). For reason of completeness, Fig. 4.19 shows the measured torques, the velocity commanded to the robot and the measured and commanded roll, pitch and yaw angles.

4.6 Discussion

An explicit and passive integrator has been designed for simulating the rigid body dynamics of a satellite rendered by a hardware-in-the-loop simulator. Starting from the standard Euler integrator, the energy generation that results from the integration process has been analytically identified and compared with its continuous time counterpart. This active energy makes the discrete-time dynamics deviate from the ideal one, resulting in position drifts or stability issues, as it was analysed in this chapter. The proposed integration scheme modifies the velocity output of the discrete-time integrator in order to preserve the energetic properties of the simulated dynamics. This allows to have precise simulation of virtual bodies on robotic facilities.

The effectiveness of the passive integrator is proved through simulations and it has been validated experimentally on an industrial robot equipped with a force-torque sensor. The experiment results prove that the integration method can deal also with sensor noise and model uncertainties, typical issues intrinsic in the hardware. The proposed passive discrete integrator does not depend on the robot dynamics model and therefore, it can be applied also in other domains.

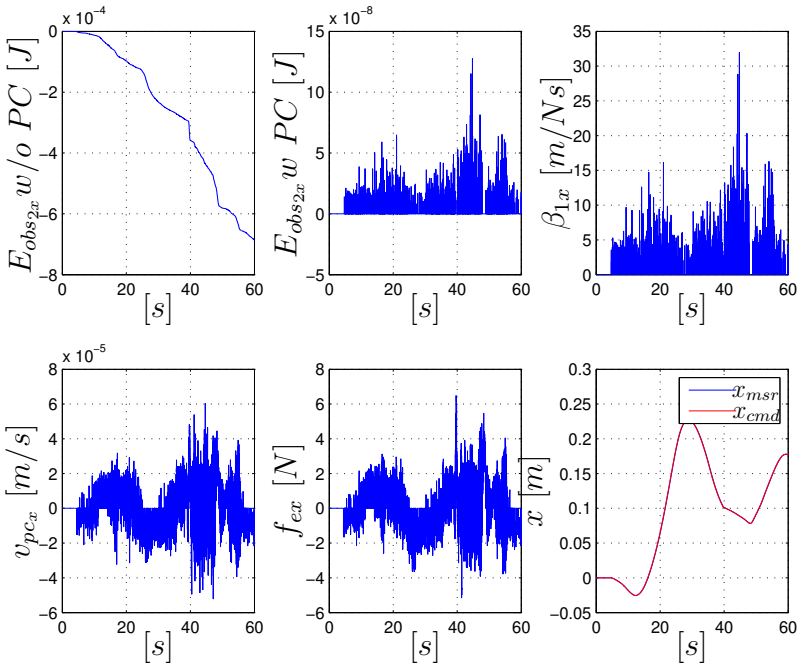


Figure 4.14: Experiment results - Energy observed (without and with PC), damping factor β_{1x}, velocity corrected by the PC, forces measured f_{ex} and motion of the robot in x.

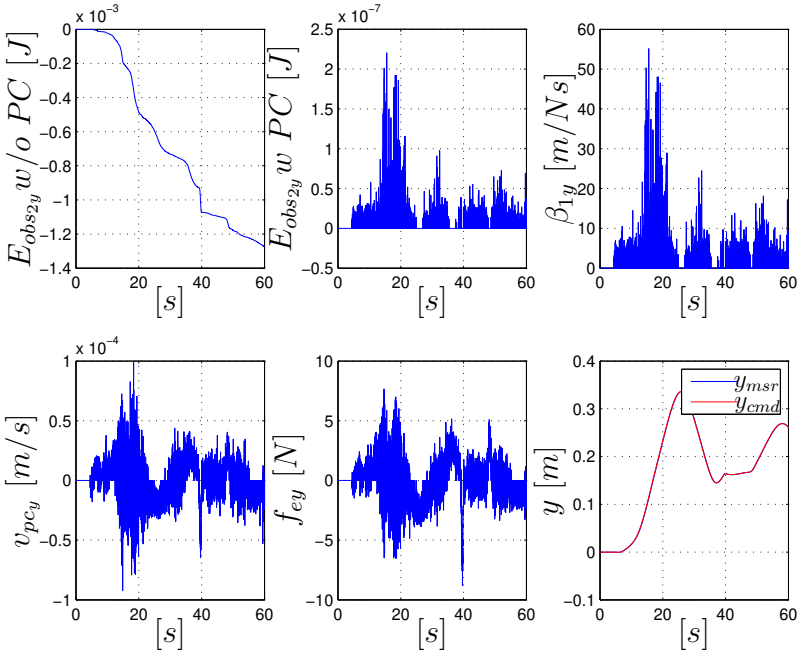


Figure 4.15: Experiment results - Energy observed (without and with PC), damping factor β_{1y}, velocity corrected by the PC, forces measured f_{ey} and motion of the robot in y.

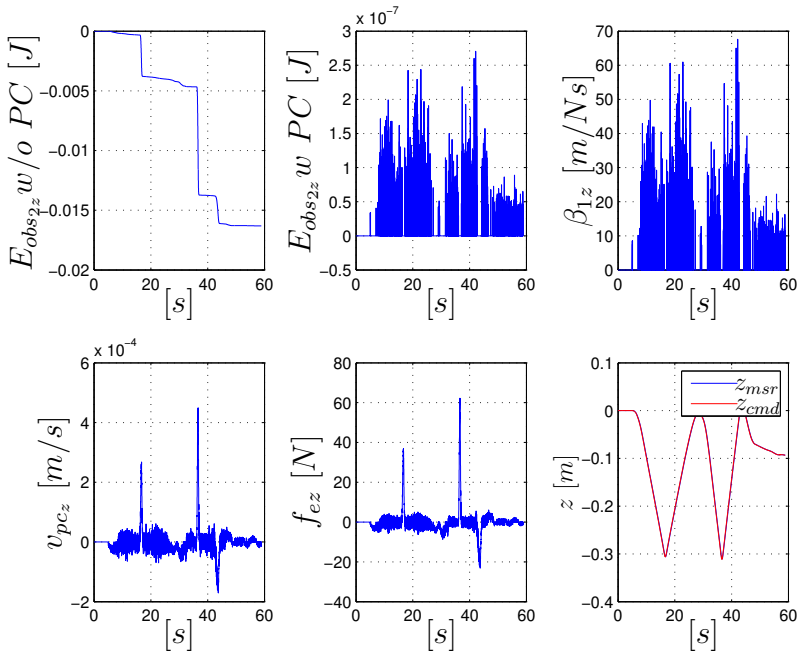


Figure 4.16: Experiment results - Energy observed (without and with PC), damping factor β_{1z}, velocity corrected by the PC, forces measured f_{ez} and motion of the robot in z.

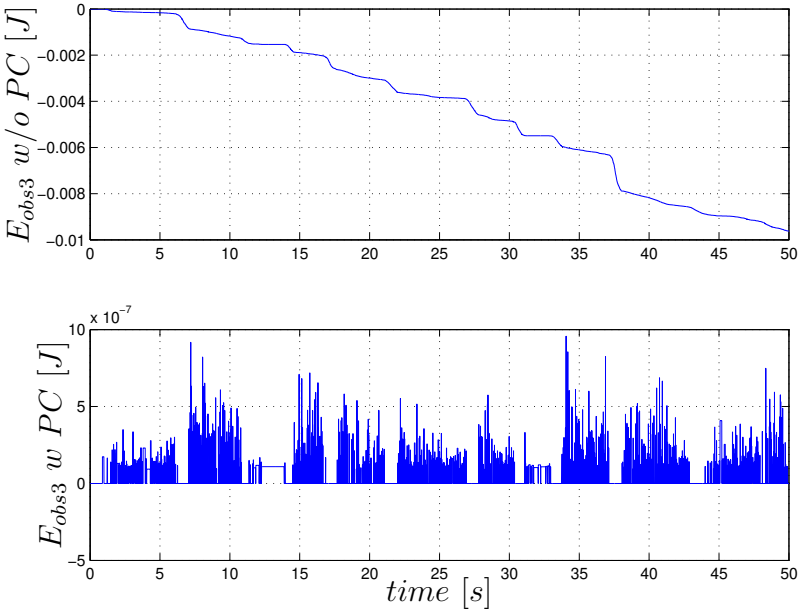


Figure 4.17: Experiment results - energy without the PC and with the PC (passivity proof).

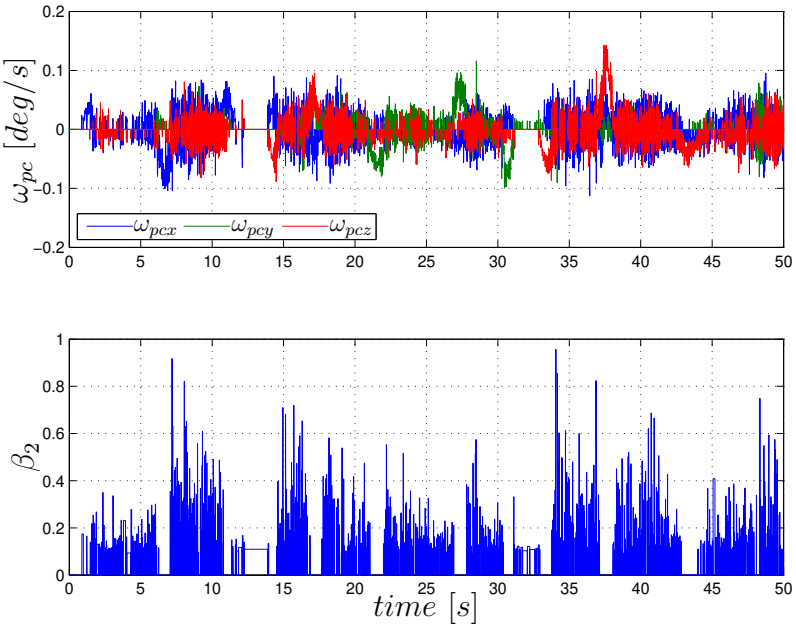


Figure 4.18: Experiment results - velocity correction (ω_{pc}) and damping coefficient (β_2).

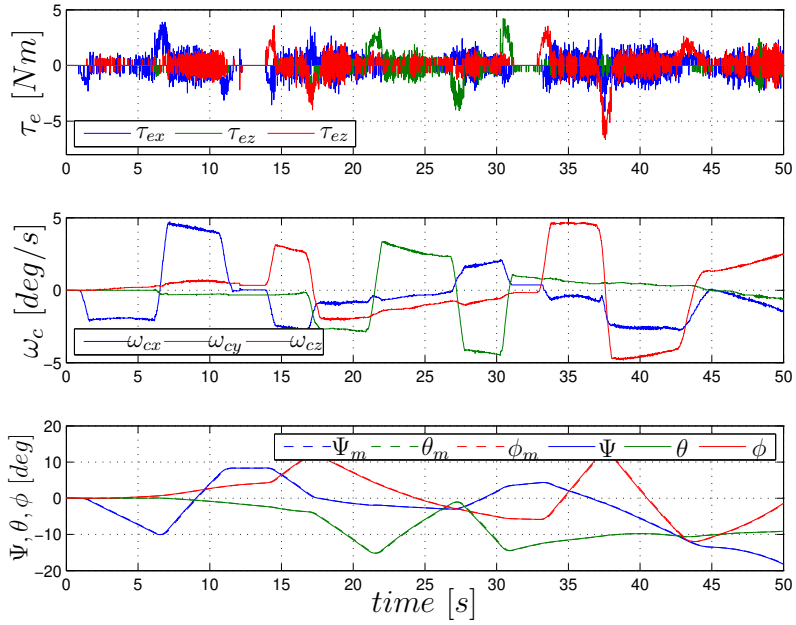


Figure 4.19: Experiment results - measured torque, corrected angular velocity (ω_c) and roll, pitch and yaw [Ψ_m θ ϕ] angles (measured: dashed line, commanded: solid line).

CHAPTER 5

UNIFIED CONTROL ARCHITECTURE FOR TIME DELAY AND DISCRETE SAMPLING EFFECTS

5.1 Introduction

In Chapter 3, a passivity-based control was designed to deal with the time delay intrinsic in the control loop of the robotic simulator. Furthermore, in Chapter 4 a passive integrator was proposed in order to compensate the energy drift caused by the Euler integration method. The aim of this chapter is to have a unified passivity-based control for dealing with both, the time delay and discrete integration effects.

Firstly, the delay compensation strategy presented in Chapter 3 is extended in order to reduce the dependency from the robot dynamics. Indeed, when the dynamic model of a robot or the access to the low-level control is not available, a force estimator as the one described in Sec. 3.4 can not be implemented. This is the case of some industrial robots, where the dynamics model or the inertia parameters are not provided. Moreover, the lack of a torque interface at the joints of the industrial robot reduces also the possibility of a dynamics identification. Therefore, the method described in Chapter 3 needs to be adapted and generalised for all the applicable cases.

Secondly, the work performed in Chapter 4 is merged in a unified passive integrator for computing the dynamics of a satellite. Then, a unified control architecture is obtained, which is able to deal both with the effects of time delay and of real-time discrete integration. The discrete-time integration effects are therefore compensated using a passive control in admittance causality and the time delay is compensated with a second passive control, which acts in impedance causality. Finally, the designed control structure is validated on the robotic simulator, the OOS-Sim. This chapter refers to the author publication [DSBS18].

5.2 Network Modelling

The benefit of using industrial robots for simulation of dynamics is given by the high tracking capability of a given velocity or position set-point. Furthermore, the possibility of connecting a forces-torques sensor with the plant improves the range of simulations in case of external interaction. The resulting unstable effects, caused by the time-delays inherent in the control loop, can be compensated with the designed passivity-based control.

Although the industrial robot can precisely track the desired set-point, when this is used for simulating satellite dynamics, its computation might require several sampling steps. Mainly, this is due to the inverse kinematics calculation, internal control of the robot and the transfer of measured data (e.g. force-torque sensor signals). Therefore the desired set-point which the robot will track is affected by delay. This aspect was already tackled in Sec. 2.3, however only for a reduced model of the simulator and without considering the discrete time integration effects. The discrete integration of the simulated dynamics (4.1) and (4.2) can lead to erroneous desired velocities while using standard integration techniques as discussed in Chapter 4. As a result, the accuracy of the simulation can drastically decrease causing position drifts and energy inconsistencies.

In this section the passivity controllers are unified in order to obtain stability during the simulation of the satellite dynamics using industrial robot and an energy consistent behaviour. Therefore, two networks are considered. One network is related to the discrete integrator and a second one to the time delay in the loop. A preliminary block diagram is shown in Fig. 5.1 where now the location of the discrete integrator and the time delay are both explicit in the loop. Notice that the robot has been represented with a time-delay block TD. This is due to the assumption discussed above, which considers the latency between the computed desired point and the measured one though the industrial robot can precisely track the set-points. Therefore the robot will move with a delayed velocity $\boldsymbol{V_s(k-\mu)}$.

A multi-dimensional network of the system can be defined exploiting the conventional mechanical-electrical analogy that maps forces into voltages and velocities into currents (see Chapter 2 for more details). The resulting network is shown in Fig. 5.2 where the robot is represented by the impedance $\boldsymbol{Z_r}$ and the passive environment with impedance $\boldsymbol{Z_e}$. The interaction with the environment pro-

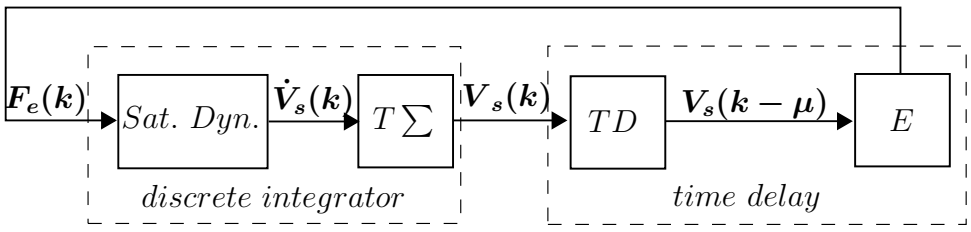


Figure 5.1: Admittance architecture where $T\Sigma$ is the discrete integrator with time step T and TD the time delay location.

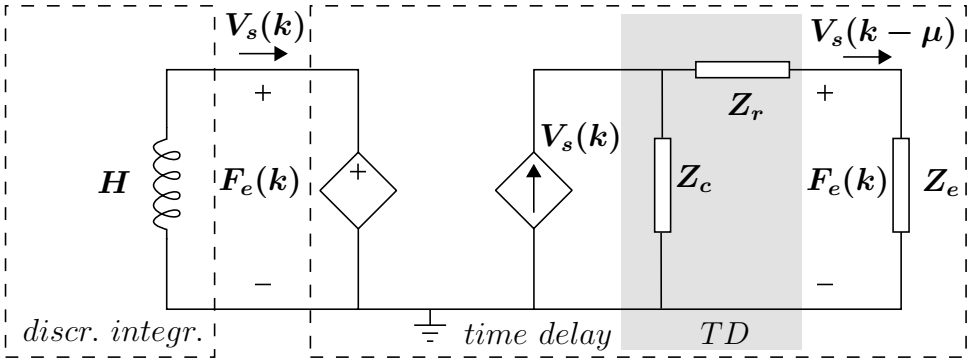


Figure 5.2: Modelling in electrical domain and networks.

duces a force $\boldsymbol{F_e}$ which is measured by the force-torque sensor placed on the robot. This is transmitted with an ideal voltage source, $\boldsymbol{F_e}$, that acts on the inductance $\boldsymbol{H} = (\boldsymbol{M}, \boldsymbol{0_{3\times3}}; \boldsymbol{0_{3\times3}}, \boldsymbol{I}) \in \mathbb{R}^{6\times6}$, which represents the total mass matrix (i.e. the admittance dynamics).

Through the dependent current source $\boldsymbol{V_s}$, the velocity of the mass is commanded to the internal controller of the robot (modelled as impedance $\boldsymbol{Z_c}$) which will finally move the robot. As stated before, if access to the internal controller of the industrial is limited, it will lead to have a missing variable. Specifically, the control force vector that is the voltage across $\boldsymbol{Z_c}$. Nevertheless, this force and the force across the robot impedance, $\boldsymbol{Z_r}$, can be associated, as represented in Fig. 5.2 by the gray box. However, it must be considered that the velocity tracked by the robot is affected by some time delay. This factor can be represented in the network representation using a time delay port, TD. The time delay port will replace the gray box of Fig. 5.2 in the following analysis.

Therefore, two one-port networks can be identified in the electrical diagram and these are represented by the two dashed boxes in Fig. 5.2. One network outputs the discrete dynamics integration $\boldsymbol{V_s}$ and the second network considers the time-delay and a passive environment. Hence, energy considerations of the system can be performed from the interaction port, which power conjugated variables are $\boldsymbol{V_s}$ and $\boldsymbol{F_e}$.

5.3 Passivity-based Control in Impedance Causality

In this section the passivity controller, which is used to compensate the time delay in the loop, is extended with an impedance causality. This allows to render passive the sub-network containing the robot and time delay without the knowledge of the robot dynamics. The main different with the method proposed in Chapter 3 is given by the impedance causality, which is required because of two main factors. Firstly, to remove the dependency from the robot dynamics and secondly to act together with the passive integrator which operates already in admittance causality.

Fig. 5.3 shows the one-port network with power-correlated variables $(\boldsymbol{F_e}, \boldsymbol{V_s})$

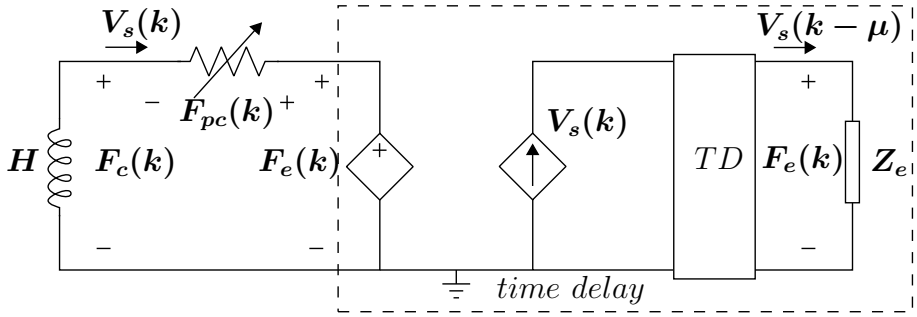


Figure 5.3: Modelling in electrical domain for the time delay: the output is a variable force.

represented in the dashed box, which includes the time-delay and a passive environment. The impedance causality of the PC will modify the output force $\boldsymbol{F_e(k)}$ by a quantity $\boldsymbol{F_{pc}(k)}$ with a time-varying damping. Therefore the wrench on the inductance $\boldsymbol{H}$ is the corrected force $\boldsymbol{F_c(k)}$, later defined.

The effect of the time delay is included in the force signals as can be seen in the following analysis. For the ideal case the energy at one-port results to be,

$$\sum_{k=0}^{m} \boldsymbol{F_e^T(k) V_s(k)} T = \sum_{k=0}^{m} \boldsymbol{V_s^T(k) Z_e V_s(k)} T. \tag{5.1}$$

However, when the delay is considered, the previous balance does not hold because the robot during the interaction with the environment has a delayed velocity, i.e. $\boldsymbol{V_s(k-\mu)}$. Therefore it will produce a force given by $\boldsymbol{V_s^T(k-\mu) Z_e}$ and the balance of energy at port level will be different, more specifically it results,

$$\sum_{k=0}^{m} \boldsymbol{F_e^T(k) V_s(k)} T = \sum_{k=0}^{m} \underbrace{\boldsymbol{V_s^T(k-\mu) Z_e}}_{\boldsymbol{F_e^T(k)}} \boldsymbol{V_s(k)} T. \tag{5.2}$$

This energy difference causes a discrepancy in velocity during the contact with the environment and the measured force will be function of it.

However, the energy at the port $\boldsymbol{(F_e, V_s)}$ can be monitored at each time-step with an energy observer to ensure passivity. If the passivity condition given in (2.49) is violated, a time varying damping will be introduced into the system in an impedance causality. Therefore, the energy observer can be written as:

$$E_{obs_1}(k) = E_{obs_1}(k-1) - \boldsymbol{F_e^T(k) V_s(k)} T + \boldsymbol{V_s^T(k-1) \alpha(k-1) V_s(k-1)} T, \tag{5.3}$$

where $\boldsymbol{\alpha(k)} \in \mathbb{R}^{6\times 6}$ is a positive definite matrix which represents the time-varying damping matrix. The variable damping matrix will regulate the action of the passivity controller, which can be defined in impedance causality as:

$$\boldsymbol{F_{pc}(k) = \alpha(k) V_s(k)}. \tag{5.4}$$

The energy observer monitors also the energy associated to the passive control action and it is expressed by the last term in (5.3). Therefore, the force correction provided to the admittance model in Fig. 5.3 is given by $\boldsymbol{F_c}$,

$$\boldsymbol{F_c(k)} = \boldsymbol{F_e(k)} - \boldsymbol{F_{pc}(k)}. \tag{5.5}$$

The time-varying damping matrix $\boldsymbol{\alpha(k)} = diag(\alpha_1, .., \alpha_6)$ can be choose to have a diagonal structure and the following energy exchange from (5.3) holds:

$$E_{obs_1}(k) = -\sum_{m=0}^{k}\sum_{i=1}^{6} F_{e,i}(m)V_{s,i}(m)T + \sum_{m=1}^{k}\sum_{i=1}^{6} \alpha_i(m-1)V_{s,i}^2(m-1)T, \tag{5.6}$$

where i is the $i-th$ component of the vector. This leads to have that:

$$E_{obs_1}(k) = \sum_{i=i}^{6} E_{obs_{1,i}}(k), \tag{5.7}$$

where

$$E_{obs_{1,i}}(k) = E_{obs_{1,i}}(k-1) - F_{e,i}(k)V_{s,i}(k)T + \alpha_i(k-1)V_{s,i}^2(k-1)T. \tag{5.8}$$

The control term in (5.4), which dissipates at each time step the active energy, is function of the variable damping matrix, whose components are:

$$\alpha_i(k) = \begin{cases} -\frac{E_{obs_{1,i}}(k)}{V_{s,i}^2(k)T} & if \quad E_{obs_{1,i}} < 0 \\ 0 & else. \end{cases} \tag{5.9}$$

In order to avoid singularities in (5.9), a small threshold should be considered if the velocity $V_{s,i}$ crosses the zero value.

Note that the optimal solution for the time-varying damping matrix presented in Sec. 3.5 can not be applied in case of impedance causality. Indeed, the designed minimization problem in (3.15) was function of a velocity difference, namely the velocity $\boldsymbol{V_s(k)}$ and $\boldsymbol{V_s(k-\mu)}$, both available data. Then the resulting solution of the minimization problem ($\min_{\beta(k)} \|\boldsymbol{V_s(k-\mu)} - \boldsymbol{\beta(k)F_o(k)} - \boldsymbol{V_s(k)}\|^2$) was the optimal time-varying damping $\boldsymbol{\beta}$.

Similarly, the minimization problem can be written in impedance causality as follows,

$$\min_{\alpha(k)} \|\boldsymbol{F_e(k-\mu)} - \boldsymbol{\alpha(k)V_s(k)} - \boldsymbol{F_e(k)}\|^2 \tag{5.10}$$

and in general it is possible to find an optimal solution for $\boldsymbol{\alpha}$ following the steps in Sec. 3.5. However, for the considered case the difference in force is not available, due to the lack of knowledge in the delayed force vector, $\boldsymbol{F_e(k-\mu)}$.

Nevertheless, the energy observer takes into account the update of energy of the passivity controller (see the last term in (5.8)), therefore, the conservatism of the damping injection is reduced and passivity can be achieved. The only possible

limitation might be the high frequency force modifications given by the PC. This effect can be minimised by using passive filters in the force signal, as shown in [KH01].

Fig. 5.4 shows the new schematic with the PC in impedance causality, where the corrected force $\boldsymbol{F_c(k)}$ is achieved by (5.5) and it is the input to the desired dynamics. Thus, if the one-port system behaves passively, the force correction is zero, i.e. $\boldsymbol{F_{pc}} = 0$ and the measured force $\boldsymbol{F_e}$ is sent as an input to the admittance dynamics without modification. If some energy is produced, the damping factor is set such that the extra energy is dissipated and passivity is restored. The damping coefficients, α_i, will make the $E_{obs_{1,i}} > 0, \forall i$. Therefore the overall energy in (5.7) will be $E_{obs_1}(k) > 0$ achieving passivity of the system.

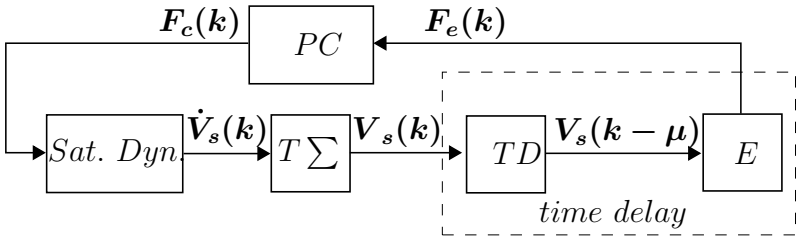


Figure 5.4: Scheme with impedance causality PC for time-delay compensation.

5.4 Passivity-based Control in Admittance Causality

This section summarises the algorithm of the passive integrator, which will be used together with the passive control acting in impedance causality for compensating the time delay in the loop. The network representation in Fig. 5.5 shows the location of the integrator (dashed box) which acts on the inductance $\boldsymbol{H}$ and after the integration process it provides a velocity $\boldsymbol{V_s}$. This network has been endowed with a variable resistor which provide the dissipation of the extra energy by using a passivity control. The admittance causality of the passive integrator will modify the output velocity of the integrator with a quantity $\boldsymbol{V_{pc}(k) = (v_{pc}, \omega_{pc})}$ (i.e. the variable resistor in Fig. 5.5). Therefore, the velocity sent to the robot is a corrected velocity $\boldsymbol{V_c(k) = (v_c, \omega_c)}$. As discussed in Sec. 4.4, the integration process will generate some extra active energy, namely ΔH_t and ΔH_r, which can be dissipated. The dissipation is handled by the time-varying damping $\beta_1(k)$ defined in (4.22) and $\beta_2(k)$ defined in (4.28). The algorithmic for the passive and explicit discrete integrator is summarised in Alg. 1 and it uses the approach presented in Sec. 4.4.

The algorithm computes the extra energy produced during the integration process and it builds the fictitious variable $\boldsymbol{\chi}$. Once everything is set-up, it observes if there is a violation of passivity and then it computes the time-varying damping gains of the passivity controllers (that will be zero if passivity is not violated). Finally the dissipative actions of the PCs are exploited for updating the velocities commanded

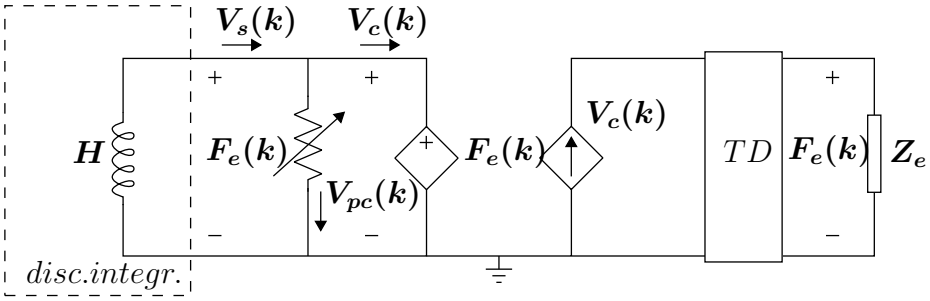


Figure 5.5: Modeling in electrical domain for discretization: the output is a variable velocity.

Algorithm 1 Passive Integrator for satellite dynamics

Input: $\boldsymbol{v_s(k-1)}, \boldsymbol{\omega_s(k-1)}, \boldsymbol{f_e(k)}, \boldsymbol{\tau_e(k)}, \boldsymbol{M}, \boldsymbol{I}, T$
Output: $\boldsymbol{v_c(k)}$, $\boldsymbol{\omega_c(k)}$

Compute ΔH_t using (4.11)
Compute ΔH_r using (4.14)
Build the $\boldsymbol{\chi(k)}$ variable using (4.26)
Compute $E_{obs2}(k)$ using (4.23)
Compute $E_{obs3}(k)$ using (4.27)
Set the damping $\beta_1(k)$ using (4.22)
Set the damping $\beta_2(k)$ using (4.28)
Compute $\boldsymbol{v_c(k)} = \boldsymbol{v_s(k)} - \beta_1(k)\boldsymbol{f_e(k)}$
Compute $\boldsymbol{\omega_c(k)} = \boldsymbol{\omega_s(k)} - \beta_2(k)\boldsymbol{\chi(k)}$
Output: $\boldsymbol{V_c(k)} = (\boldsymbol{v_c(k)}, \boldsymbol{\omega_c(k)})$

to the robot. The presented pseudo-code can be easily executed in real-time on industrial robots.

5.5 The Overall Architecture

In this section, the overall architecture which combines both passivity controllers is presented. As shown in Fig. 5.6, the networks endowed with the passivity controllers developed in the previous sections have been connected. The network N_1 deals with the discrete integrator, where the variable resistor has resulting current $\boldsymbol{V_{pc}^u}$ and N_2 deals with the passivity control for the time delay, where the voltage drop is $\boldsymbol{F_{pc}^u}$[1]. The networks N_1 and N_2 are rendered passive by the passivity controllers.

As discussed in Sec. 2.4, a system resulting from two passive systems which are connected through a power preserving interconnection, is also passive (see Proposition 2). Therefore, the flows and efforts in the new system variate accordingly to the new input-output causality when the networks N_1 and N_2 are connected.

In particular, in the network N_1, the corrected wrench $\boldsymbol{F_c^u}$ is obtained by (5.5), where $\boldsymbol{F_{pc}}$ in (5.4), $E_{obs1,i}$ in (5.8) and α_i in (5.9) are recalculated considering the

[1] The index u stands for unified velocity or force, later defined.

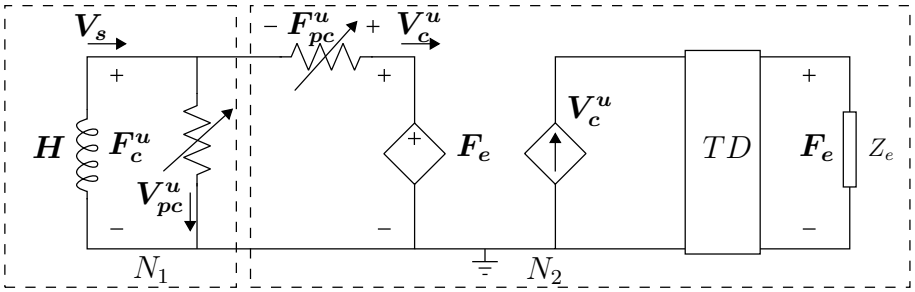


Figure 5.6: Interconnection of passive networks. N_1 is the network of the passive integrator and N_2 the network of the passive controller for the time delay compensation.

corrected velocity $\boldsymbol{V_c^u}$ (later defined) instead of $\boldsymbol{V_s}$. Similarly, the corrected twist in network N_2, namely, $\boldsymbol{V_c^u} = (\boldsymbol{v_c^u}, \boldsymbol{\omega_c^u})$ is obtained by the corrected velocity (4.21) and the corrected angular velocity (4.30), where $\boldsymbol{v_{pc}}$ in (4.20), $E_{obs2,i}$ in (4.23), $\beta_{1,i}$ in (4.22), and ΔH_t in (4.11) are recalculated considering the corrected force $\boldsymbol{F_c^u} = \boldsymbol{(f_c^u, \tau_c^u)}$ instead of $\boldsymbol{F_e} = \boldsymbol{(f_e, \tau_e)}$. Similarly E_{obs3}, $\Delta H_{r1}(k)$, $\Delta H_{r2}(k)$, $\boldsymbol{\chi(k)}$ are recalculated from (4.14), (4.26) with the corrected values $\boldsymbol{\tau_c^u}$ and $\boldsymbol{\omega_c^u}$ to recompute $\boldsymbol{\omega_{pc}}$ from (4.29). The unified force and velocity correction is summarised in the following subsections.

Unified force correction

The unified force correction $\boldsymbol{F_c^u} = (\boldsymbol{f_c^u, \tau_c^u})$ is the input to the discrete dynamics and by defining F_c^u its generic component, it results

$$F_c^u(k) = F_e(k) - F_{pc}^u(k), \tag{5.11}$$

where the term $F_{pc}^u(k) = \alpha^u(k) V_c^u(k)$ is the force modification due to the passivity control. The time-varying damping factor $\alpha^u(k)$ is defined as:

$$\alpha^u(k) = \begin{cases} -\frac{E_{obs_1^u}(k)}{V_c^u(k)^2 \Delta T} & if\ E_{obs_1^u} < 0 \\ 0 & else \end{cases} \tag{5.12}$$

where the energy observer is:

$$E_{obs_1^u}(k) = E_{obs_1^u}(k-1) + F_c^u(k) V_c^u(k) T + E_{pc_1^u}(k-1),$$

being $E_{pc_1^u}(k) = V_c^{u2}(k) \alpha^u(k) T$, the energy dissipated by the PC. The previous equations are functions of the corrected velocity V_c^u which is the output of the passive integrator, defined in the following subsection.

Unified velocity correction

The corrected velocity sent to the robot is the vector $\boldsymbol{V_c^u}$ composed of $\boldsymbol{V_c^u} = (\boldsymbol{v_c^u}, \boldsymbol{\omega_c^u})$ where $\boldsymbol{v_c^u}$ is the corrected velocity for the translational dynamics and $\boldsymbol{\omega_c^u}$ the corrected velocity for the rotational dynamics.

For the translational dynamics, it is possible to consider the individual components of the vectors $\boldsymbol{v_c^u}$ and $\boldsymbol{f_c^u}$ (because the dynamics is decoupled), therefore:

$$v_c^u(k) = v_s(k) - \beta_1^u(k) f_c^u(k). \tag{5.13}$$

The velocity correction provided by the PC is $v_{pc}^u(k) = \beta_1^u(k) f_c^u(k)$ and the time-varying damping is:

$$\beta_1^u(k) = \begin{cases} -\frac{E_{obs2}^u(k)}{f_c^u(k)^2 T} & if\ E_{obs2}^u(k) < 0 \\ 0 & else. \end{cases} \tag{5.14}$$

The energy observer is rewritten as:

$$E_{obs_2}^u(k) = E_{obs_2}^u(k-1) - \Delta H_t^u(k) + \beta_1^u(k-1) f_c^u(k-1)^2 T, \tag{5.15}$$

where $\Delta H_t^u(k)$ is recalculated from (4.10) with the corrected values of the force f_c^u.

For the angular velocity commanded to the robot will result:

$$\boldsymbol{\omega_c^u(k) = \omega_s(k) - \omega_{pc}^u(k)}, \tag{5.16}$$

where $\boldsymbol{\omega_{pc}^u(k)} = \beta_2^u(k) \boldsymbol{\tau_c^u(k)}$ is the correction provided by the passivity controller where the variable damper is:

$$\beta_2^u(k) = \begin{cases} -\frac{E_{obs3}^u(k)}{T \|\chi^u(k)\|^2} & if\ E_{obs3}^u < 0 \\ 0 & else \end{cases} \tag{5.17}$$

and the energy observer is written as:

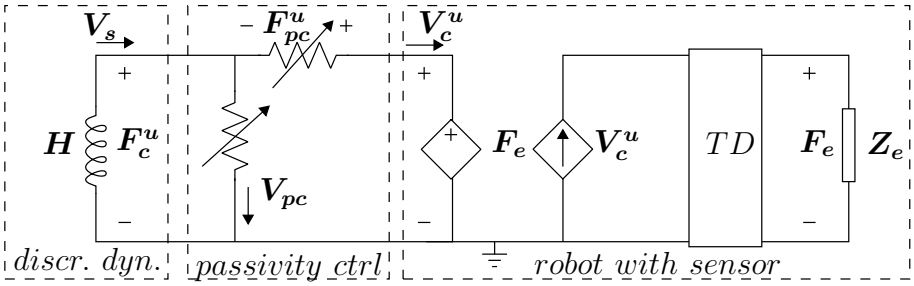
$$\begin{aligned} E_{obs3}^u(k) = E_{obs3}^u(k-1) &- (\Delta H_{r1}^u(k) + \Delta H_{r2}^u(k)) \\ &+ T \boldsymbol{\chi^u(k-1)}^T \beta_2^u(k-1) \boldsymbol{\chi^u(k-1)}, \end{aligned} \tag{5.18}$$

where $\Delta H_{r1}^u(k)$, $\Delta H_{r2}^u(k)$, $\boldsymbol{\chi^u(k)}$ are recalculated from (4.13), (4.26) with the corrected values $\boldsymbol{\tau_c^u}$ and $\boldsymbol{\omega_c^u}$.

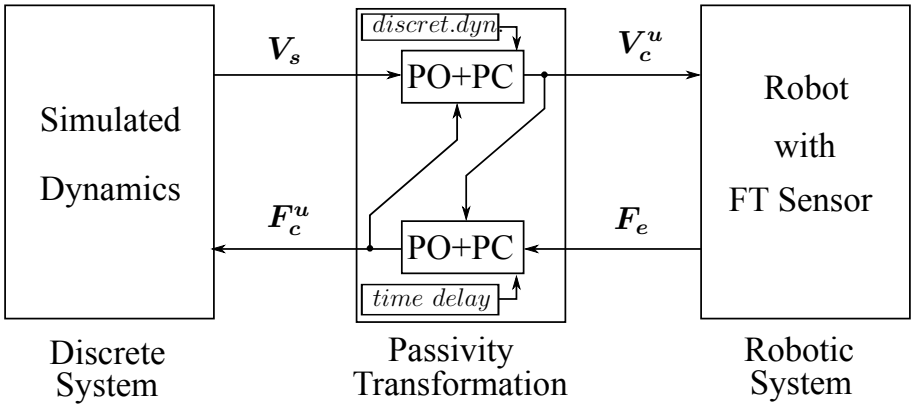
Overall passivity

The variables of the controllers, $\boldsymbol{V_{pc}^u} = (\boldsymbol{v_{pc}^u}, \boldsymbol{\omega_{pc}^u})$ and $\boldsymbol{F_{pc}^u}$, will render the respective networks passive. Hence, the overall system is composed of an interconnection of two passive networks (N_1, N_2) which lead to an overall passive system that has no energy production [Kha02b]. Being the overall system passive, a 2-ports network can be isolated from Fig. 5.6 and it is possible to combine the two passivity control variables, namely $\boldsymbol{F_{pc}^u}$ and $\boldsymbol{V_{pc}^u}$. As can be seen in Fig. 5.7a, the 2-ports network, named *passivity ctrl*, acts as a passivity transformation or layer from the simulated *discrete dynamics* network to the *robot with sensor* network. The passivity transformation is summarised in Fig. 5.7b considering the input-output variables.

The transformation considers the passivity controllers acting in admittance and impedance causality. Therefore, the robot receives a modified twist $\boldsymbol{V_c^u}$ to correct the effects of the discrete integration and the simulated dynamics receives as input a modified wrench $\boldsymbol{F_c^u}$ to correct the effects of the time delay.



(a) The passivity control architecture represented in electrical domain.



(b) Network representation of the overall architecture with the passivity transformation layer.

Figure 5.7: The overall architecture for simulating satellite dynamics on a robot in presence of time delay and discrete integration effects.

5.6 Results: Simulations and Experiments

Simulations

The overall architecture has been tested firstly in a simulation study. The simulated mass of the satellite is $M = 280\ Kg$ with inertia $\boldsymbol{I} = diag(18, 20, 22) Kgm^2$ (i.e. client satellite for the DEOS mission [DS]). The considered sampling time is 4 ms and the time delay in the loop 40 ms. Forces-torques are simulated with a spring-dashpot model and the contacts occur a defined positions. The behaviour of the system affected by the time delay can be seen in Fig. 5.11a where the velocity with the time delay in the loop (solid line) is compared with the ideal case velocity (dashed line) without delay. As it can be seen, the velocity of the simulated rigid body increases after each collision and the system becomes unstable.

The proposed method is applied and the results related to the translational dynamics are shown in Fig. 5.8. The first column shows the passivity control for time delay compensation where the correction is provided in impedance mode with the forces $\boldsymbol{f^u_{pc}}$. The second column shows the results for the passivity control which compensate the effects of discrete integration by providing an admittance correction $\boldsymbol{v^u_{pc}}$. The corrections provided by the passivity controller, ($\boldsymbol{f^u_{pc}}$ and $\boldsymbol{v^u_{pc}}$) act in

order to remove the activities measured by the energy observer which is shown in the second row of Fig. 5.8. As it can be seen, the energies observed without the passivity controllers, (E^u_{obs1} w/o PC and E^u_{obs2} w/o PC) become negative indicating activity in the system. The passivity proof is shown in the third row (E^u_{obs1} w PC and E^u_{obs2} w PC), where the positive semi-definiteness of these energies indicate the passivity of the system for translational dynamics.

Similarly, the first column in Fig. 5.9 shows the corresponding results for the rotational dynamics where $\boldsymbol{\tau^u_{pc}}$ is the torque correction related to the passivity control for the time delay. The second column is related to the passivity control which provides an angular velocity correction $\boldsymbol{\omega^u_{pc}}$ to avoid the discretization effects. Also for the rotation dynamics, the energies with the passivity control (E^u_{obs1} w PC and E^u_{obs3} w PC) shown in the last row of Fig. 5.9 are positive semi-definite and therefore, all the activity has been dissipated. As a result, the diverging behaviour is resolved and the motion of the rigid body behaves stable and energetically consistent. It can be seen in Fig. 5.11b where the translational and rotational velocities of the rigid body are shown and compared with the velocity in the ideal case.

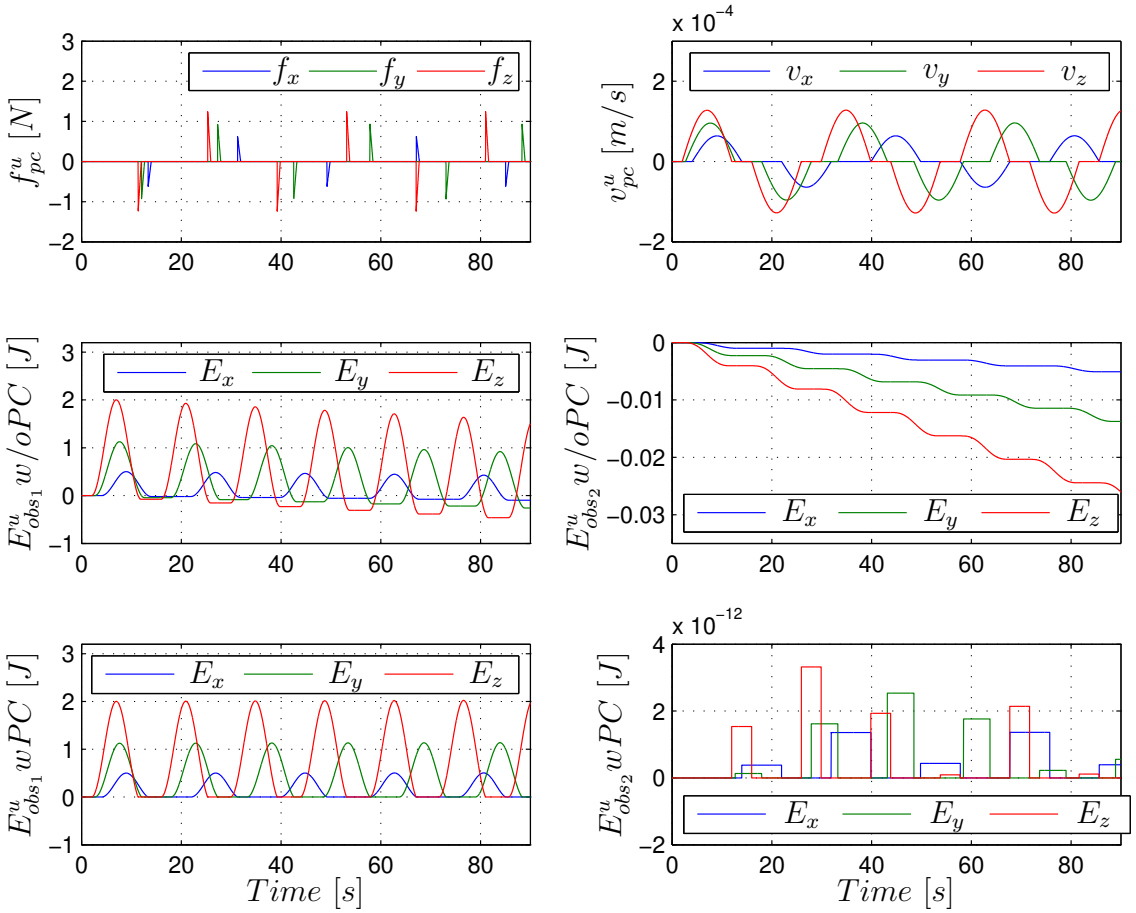


Figure 5.8: Translational dynamics: force correction $\boldsymbol{f^u_{pc}}$ and velocity correction $\boldsymbol{v^u_{pc}}$, energy observers without and with PC. Last row (E^u_{obs} w PC) indicates the passivity of the system.

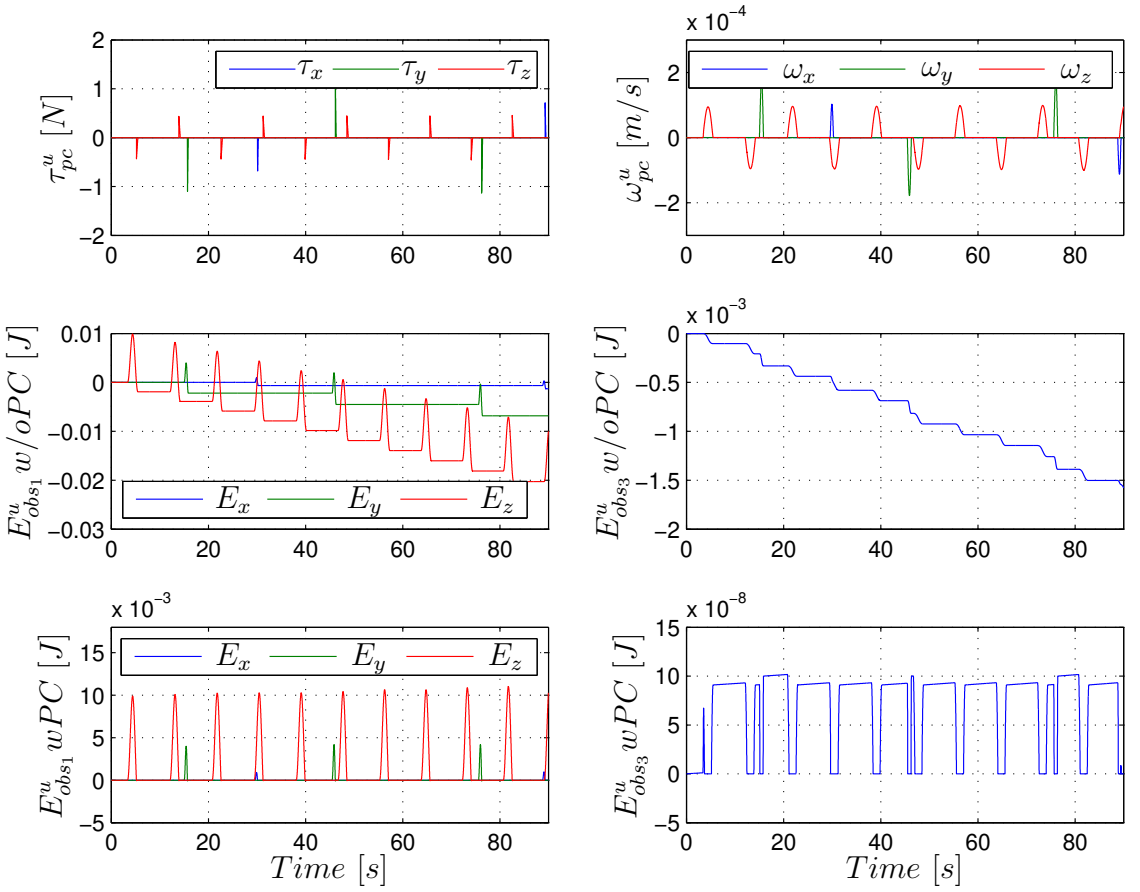


Figure 5.9: Rotational dynamics: torque correction $\boldsymbol{\tau^u_{pc}}$, angular velocity correction $\boldsymbol{\omega^u_{pc}}$ energy observers without and with PC. Last row (E^u_{obs} w PC) indicates the passivity of the system.

Experiments

The experiment is conducted on the client robot of the OOS-Sim facility shown in Fig. 1.3. The considered rigid body has a mass of 700 Kg and inertia $I = diag(116, 160, 160)$ Kgm^2 (i.e. chaser satellite in the DEOS mission [DS]). The intrinsic time-delay of the facility is considered during the experiment. The motion of the satellite is initialised with an initial linear velocity of $[0, 0, 0.02]$ m/s and interacts with a passive environment in both the direction of the motion (z-axis). The robot motion and the corresponding energies in reproducing the dynamics are analysed in the following subsections.

Robot Motion During the Experiment

When the proposed architecture is not active, the velocity of the satellite, simulated with the robot during its interaction with the environment increases, as shown in Fig. 5.10. It can be seen that the velocity diverges after each contact leading to an unstable behaviour. Now the proposed passivity-based approach is applied for the

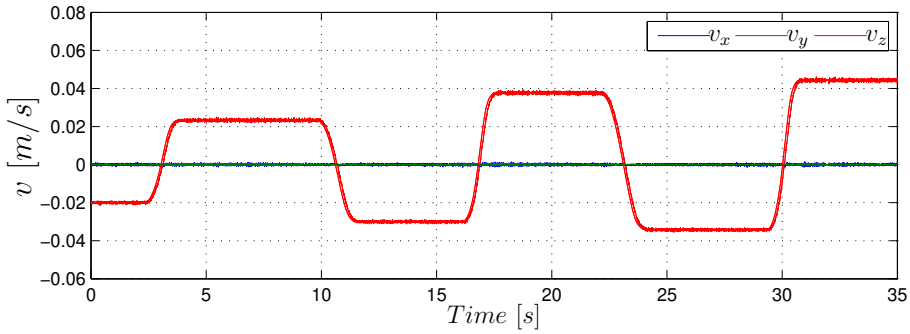


Figure 5.10: Experiment: increase in the robot velocity due to the time delay and discretization without passivity control.

same mass and the same initial conditions. The forces and torques are measured by the sensor during the contact with the environment as shown in Fig. 5.12a and the active energy is observed and dissipated during the experiment. Therefore the velocity of the robot does not increase as shown in Fig. 5.12b, resulting in a stable behaviour. The following subsection analyses the results recorded during the experiment related to the passivity controllers. The passivity of the system is proved for the translational and rotational dynamics, see Fig. 5.13 and Fig. 5.14.

Passivity of the Dynamics Rendered by the Robot

The data in Fig. 5.13a show the impedance correction, $\boldsymbol{f}_{\boldsymbol{pc}}^{\boldsymbol{u}}$, to avoid the activity due to the time delay which is observed with the E_{obs1}^{u} w/o pc. The activity due to the discretization is shown in Fig. 5.13b (middle) which is corrected with the passivity controller in order to provide a correction $\boldsymbol{v}_{\boldsymbol{pc}}^{\boldsymbol{u}}$ (see Fig. 5.13b top) to restore the passivity. Passivity proof is given by the positive semi-definiteness of E_{obs}^{u} w pc in Fig. 5.13a and Fig. 5.13b. The action of the passivity control dealing with the time delay for the rotational dynamics is shown in Fig. 5.14a. In particular, $\boldsymbol{\tau}_{pc}^{u}$ is the passivity correction due to the energy generated by the time delay (measured in Fig. 5.14a middle). Also for the rotational dynamics, the activity due to the discretization is corrected by the admittance passivity control which provides a velocity correction $\boldsymbol{\omega}_{pc}^{u}$ shown in Fig. 5.14b top. This allows to restore the passivity properties of the system as proved in Fig. 5.14a and Fig. 5.14b bottom. The positive semi-definiteness of the energy indicates the passivity of the system.

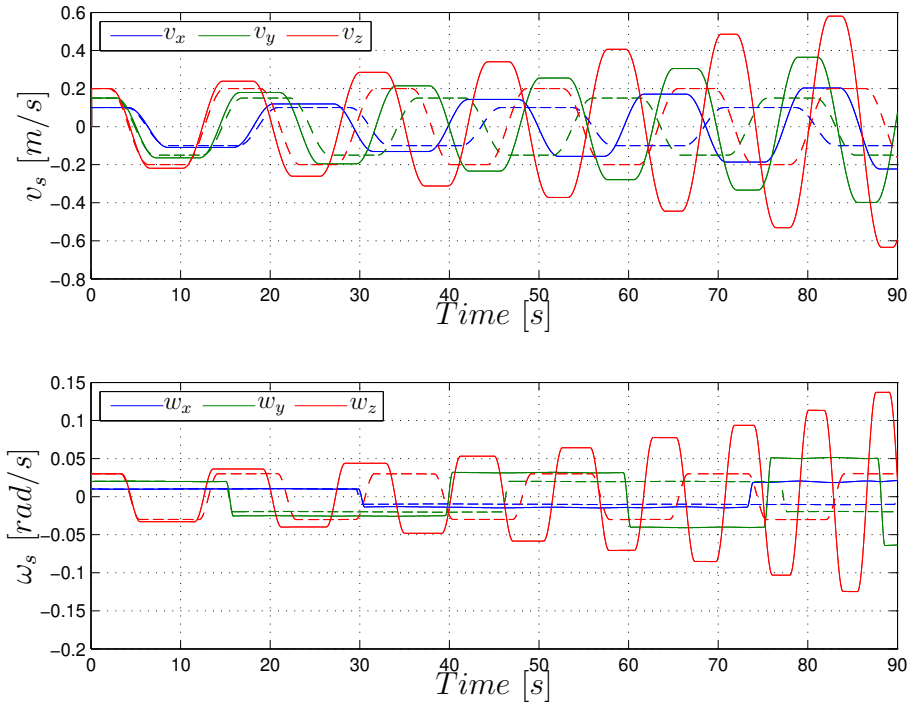
5.7 Discussion

In this chapter, a unified architecture is proposed to faithfully reproduce rigid-body dynamics with an industrial robot. The architecture is based on two passivity controllers. On one side, the force is modulated as input to the simulated dynamics. On the other side, the velocity is modulated as output to the discrete integrator. Therefore, the robot will receive a corrected velocity in order to preserve the en-

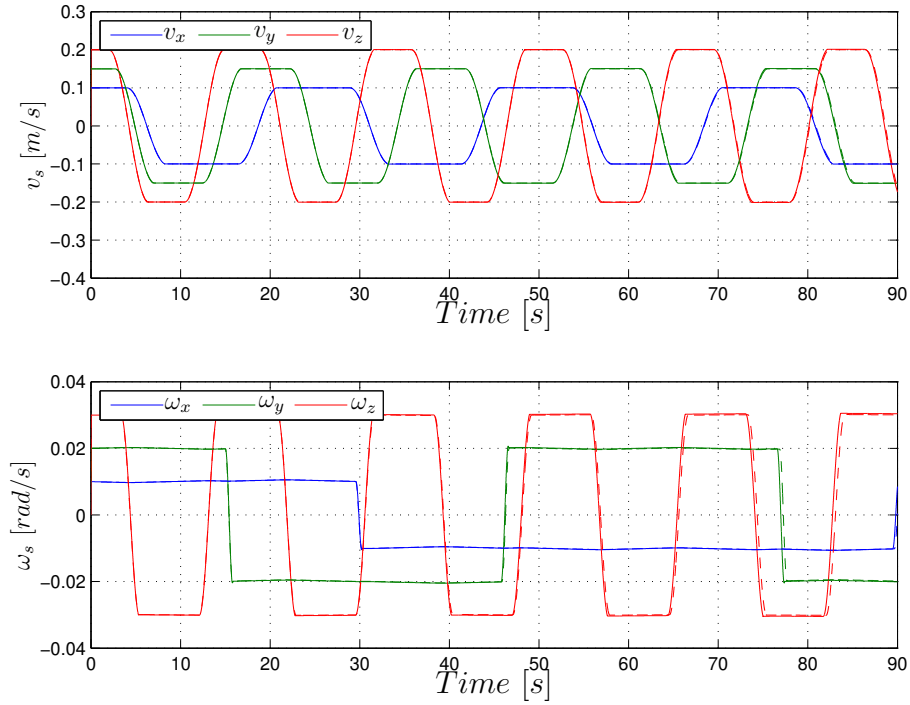
ergetic properties of the dynamics during the simulation. The obtained results are promising in both, simulations and experiments. The experiments proved that the developed method can run on a real-time robot and can deal also with sensor noise, which is intrinsic in measured data.

The reader should note that the extra energies introduced by the Euler integration as reported in (4.11) and (4.14) are calculated in discrete-time. These values might be slightly different with respect to the real energetic disparity between discrete and continuous-time cases as it was extensively discussed in Sec. 4.3.1. This error is given by the loss of information due to discretization and can not be avoided. However, this difference becomes smaller as the sampling time decreases. In the case of the experiment the sampling time is 0.004 s and therefore the relative difference in energy can be neglected.

Regarding the action of the passivity control, it is worth to notice that high frequency force and velocity modifications can represent a limitation of the TDPA damping injection [KH01]. The high frequency damping generates an effect which might modify the natural dynamic behaviour of the system. However, this effect can be minimised by using passive filters in the force and velocity signals to remove the chattering effects.

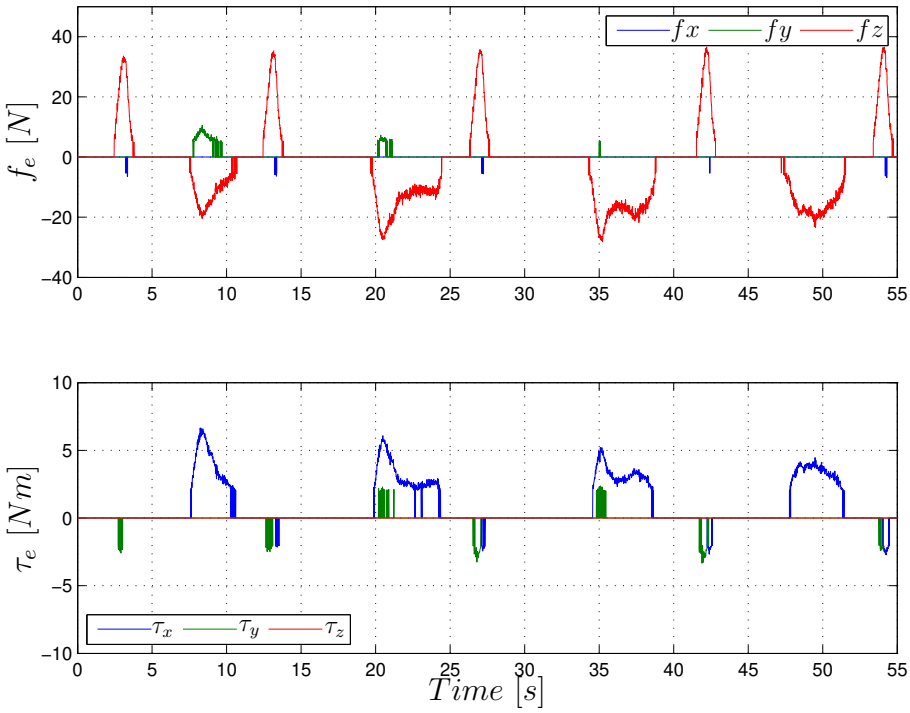


(a) Unstable system: comparison of velocity with time delay in the loop (solid line) and ideal velocity without time delay (dashed line).

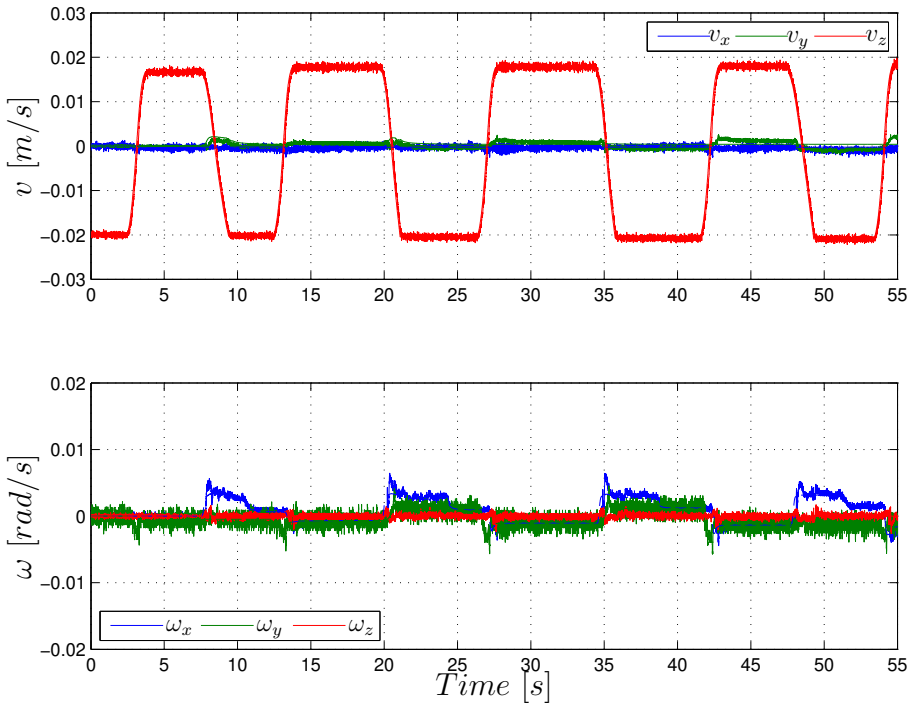


(b) Stable system: comparison of velocity with time delay in the loop and the overall passivity control architecture (solid line) and ideal velocity without time delay (dashed line).

Figure 5.11: Time delay causes system instability (top). The passivity control restores the passivity of the system (bottom).

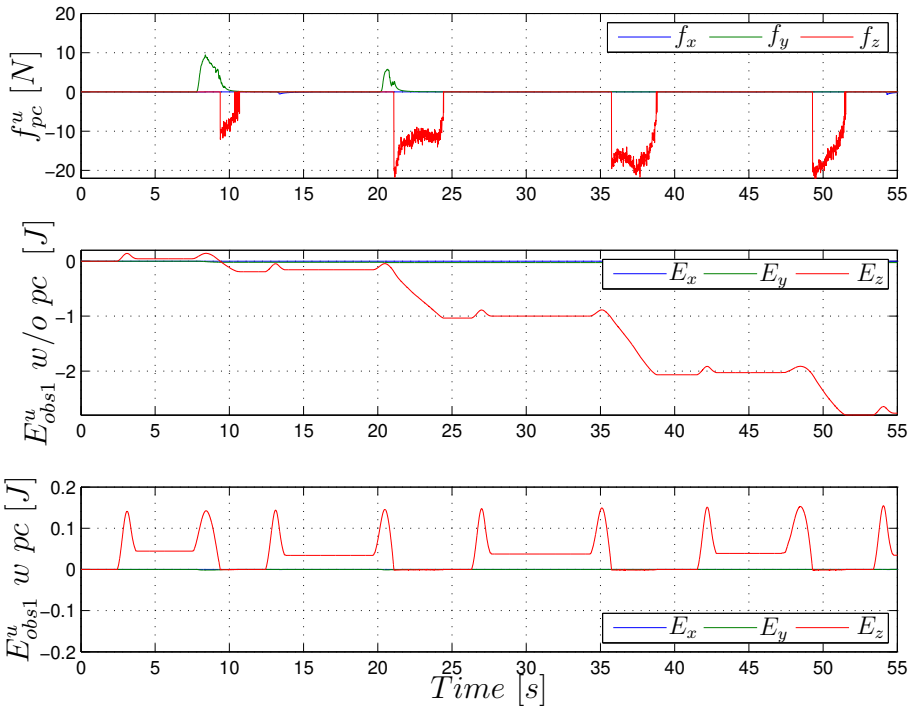


(a) Forces and torques measured by the sensor.

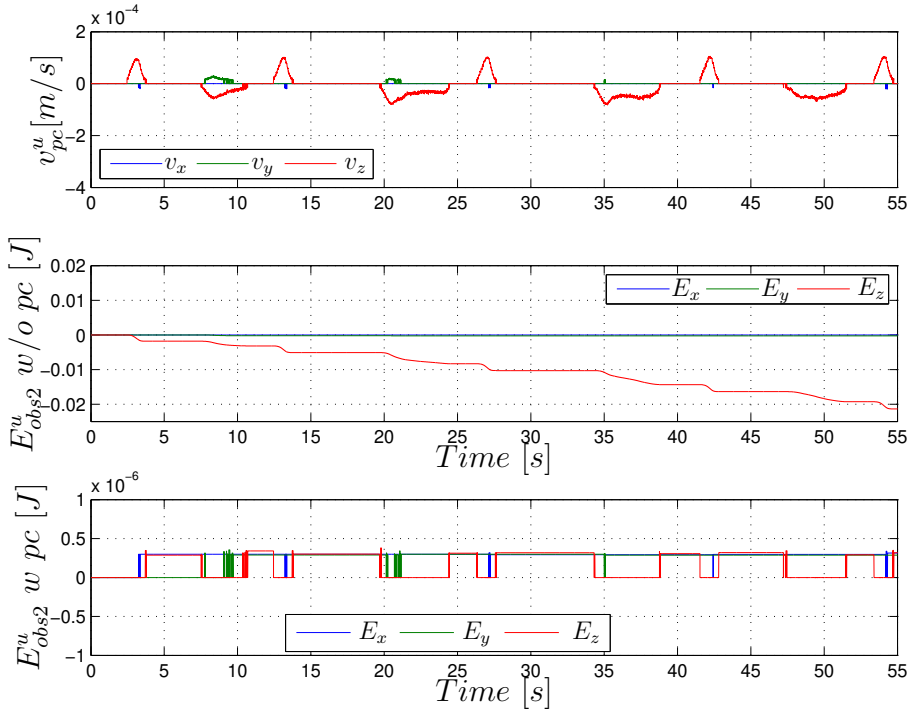


(b) Linear and angular velocity of the robot with the proposed approach.

Figure 5.12: Experiment: force and torque measured and velocity of the robot.

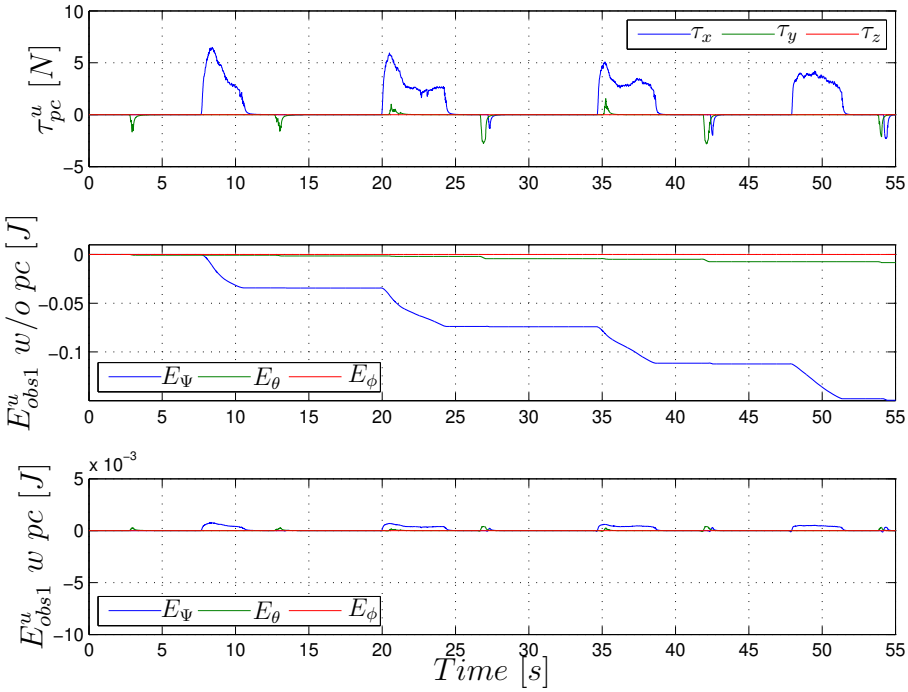


(a) Force corrected by the PC, energy observer without the PC and energy with PC.

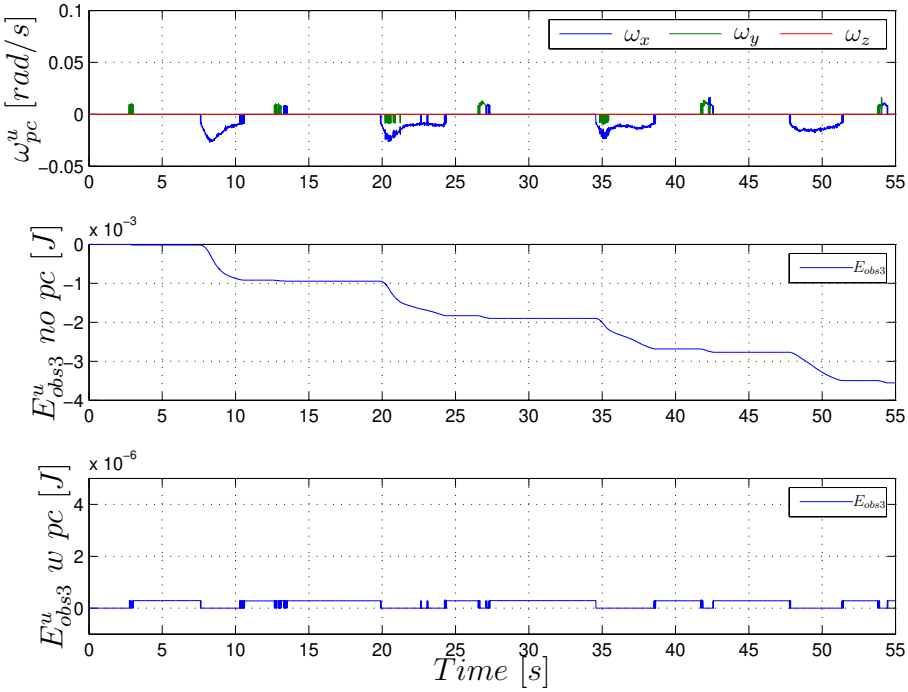


(b) Linear velocity corrected by the PC, energy observer without PC and energy with PC.

Figure 5.13: Experiment: Passivity controller for compensating time delay (a) and discretization effects (b) for the translational dynamics. Last rows in (a) and (b) indicate the passivity of the system.



(a) Torque corrected with the PC, energy observer without the PC and energy with PC.



(b) Angular velocity corrected by the PC, energy observer without PC and energy with PC.

Figure 5.14: Experiment: Passivity controller for compensating time delay (a) and discretization effects (b) for the rotational dynamics. Last rows in (a) and (b) indicate the passivity of the system.

CHAPTER 6

IMPEDANCE CONTROLLERS FOR SPACE ROBOTS

6.1 Introduction

The previous chapters have tackled the stability issues which occur while simulating a dynamics with a robotic facility and controllers were develop to achieve a stable and faithful simulation during the robot motion. The aim, therefore, was to obtain a reliable platform for testing the control algorithms for OOS missions. Such kind of missions consider the use of a satellite equipped with a robotic manipulator, which performs the servicing tasks on a defective target satellite.

In general, there are two established modes to control the manipulator while performing these tasks, which can be divided in position or torque mode. In position control mode, the manipulator maintains a strict deterministic position and in doing so uses all available torques. This behaviour is mostly useful in industrial applications, where the environment is known and high position accuracy is required. However, the position control lacks of compliance that is required in uncertain environment. Using impedance control, the torque itself is controlled and can be shaped to exhibit both stiff and compliance behaviour. Impedance capabilities represent an important factor for a space scenario, especially during the stabilisation and the approach phase where the manipulator might interact with a floating target satellite pushing it away and compromising the mission [ABDS+16]. Impedance control for the manipulator and the dynamic coupling with the floating base, (i.e the satellite where the manipulator is connected), induce many challenges for the dynamic, planning and control. Mainly two approaches are considered in literature for the control of the manipulator arm connected to a satellite-base. In *free-floating robot control*, the actuation of the base is not considered. Therefore, the base reacts accordingly to the motion of the controlled manipulator. When the control of the satellite base is required, then the base needs to be actuated and, in this case, the approach is defined as *free-flying robot control* [NP17].

In this chapter, impedance controllers for a space robot are developed and validated on ground with the OOS-Sim robotic facility. Firstly, torque controllers for the free-floating manipulator are proposed. In particular, two approaches are described, a passivity-based controller which exploits the so-called generalized Jacobian, later defined, and a second one based on the computed torque using feedback linearisation. The controller based on the generalized Jacobian was used as the baseline for the DEOS mission [DSAG+15]. Furthermore, it was also exploited for performing teleoperation experiment, as described in [ABDS+16] and the main features are also presented in this chapter.

Secondly, an other impedance controller is designed for the free-flying space robot, i.e. when the satellite base is actuated by thrusters or reaction wheels. The free-flying control presented here was proposed by the author in e.Deorbit ESA space project [TEDS+17, JLR+18]. Furthermore, an impedance control, which can be used during the stabilisation phase is also presented and analysed through Motecarlo simulations.

Related Works

The presence of a floating base for a space manipulator induces many challenges from the point of view of robot dynamics, planning and control. Indeed, the manipulator motion is coupled with the floating motion of the satellite, inducing translations and rotations of the base together with interferences on the desired motion of the end effector. One important consequence of such coupling is the presence of dynamic singularities [PD93, DP93] that are path-dependent, unlike for the fixed-base case. The presence of dynamic singularities further complicates the task of the trajectory planning algorithms [LH13]. In [UY87] the kinematic problem for a free floating robot was addressed and the generalized Jacobian was presented. This relates the end effector velocities to the joint velocities, taking into account the conservation of momentum, which characterises the motion of the free-floating robot in absence of external forces. Based on this concept, a kinematics-based control was then presented in [UY89]. The dynamics problem was treated in [MMA89], where a generalized transposed Jacobian approach was presented for the end effector regulation control and the stability was proved. In [PD90] the generalized Jacobian was shown to be the natural extension of the classic Jacobian for free-floating systems, allowing to address most of the control problems using classical approaches for fixed-base manipulators. The effectiveness of the transpose Jacobian control was further investigated for free-flying systems in [DP93]. The problem of the impedance control of a free-floating robot was also treated in [NP17] and the control for a free-flying robot was developed in [NY06]. Both authors consider an approach based on feedback linearisation.

The author contribution to the research field is the design and validation of non linear impedance controls for a space robot. The reader can refer to [DSAG+15], for on-ground validation, [ABDS+16] for the teleoperation application, [TEDS+17] for the free-flying robot control and [CBS+18] for the stabilisation control.

6.2 Dynamics Model

This section introduces the dynamics model of the space robot, which is considered for the design of the controllers and for the validation on the OOS-Sim. The general equations of motion for a satellite equipped with a manipulator with $n-joints$, can be defined as follows [Fea07]:

$$\begin{bmatrix} \boldsymbol{H_b} & \boldsymbol{H_{bm}} \\ \boldsymbol{H_{bm}^T} & \boldsymbol{H_m} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c_b} \\ \boldsymbol{c_m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_b} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{6.1}$$

where $\boldsymbol{H_b} \in \mathbb{R}^{6\times6}$, $\boldsymbol{H_m} \in \mathbb{R}^{n\times n}$, $\boldsymbol{H_{bm}} \in \mathbb{R}^{6\times n}$ are the inertia matrices of the base and manipulator and the coupling inertia matrix between the base and the manipulator, respectively. The vectors $\ddot{\boldsymbol{x}}_{\boldsymbol{b}} \in \mathbb{R}^6$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the acceleration of the base and the acceleration of the robot joints; $\boldsymbol{c_b} \in \mathbb{R}^6$ and $\boldsymbol{c_m} \in \mathbb{R}^6$ are the non-linear velocity dependent terms on the base and on the manipulator, respectively. $\boldsymbol{F_b} \in \mathbb{R}^6$ is the force torque wrenches acting on the center of mass of the base-body and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the internal torque vector of the manipulator.

The reader should note that in (6.1), $\boldsymbol{F_b}$ refers to the actuation force of the base. Usually two control approaches are available in the literature, the free-floating robot control, which considers no actuation at the base and the free-flying controller where actuation at the base is available. The controllers developed within this thesis will deal with both cases. Therefore, the reader can refer to the following

Remark 1 *Difference between the space robot controllers*

- ***Control for a free-floating robot**: The satellite-base is not actuated, i.e $\boldsymbol{F_b} = \boldsymbol{0}$, therefore only the manipulator arm is controlled.*
- ***Control for a free-flying robot**: The satellite-base is actuated, i.e $\boldsymbol{F_b} \neq \boldsymbol{0}$ and also the manipulator arm is controlled.*

Generalized space robot dynamics

A common way in literature to deal with the control problem for a free-floating robot is to exploit the generalized equation of the multi-body dynamics [UY89], [SK08]. This allows to express (6.1) in a fixed-base like dynamic function, which includes however the nature coupling of the floating base. Therefore, from the upper and lower sets of equation (6.1), the dependency from $\ddot{\boldsymbol{x}}_{\boldsymbol{b}}$ can be eliminated to obtain the following expression:

$$\boldsymbol{H}^*\ddot{\boldsymbol{q}} + \boldsymbol{C}^* = \boldsymbol{\tau} + \boldsymbol{J}^{*T}\boldsymbol{F_e}, \tag{6.2}$$

where $\boldsymbol{H}^* \in \mathbb{R}^{6\times6}$, $\boldsymbol{C}^* \in \mathbb{R}^6$, $\boldsymbol{J}^* \in \mathbb{R}^{6\times n}$ are the so-called generalized inertia matrix, generalized Coriolis/centrifugal forces and generalized Jacobian of a space robot, respectively [UY87]. Equation (6.2) resembles the dynamic equation of a fixed-based robot, however the dynamic interaction of the base is included in the generalized

matrix. In particular the matrix $\boldsymbol{H}^*$ represents the inertia properties of the system in the joint space and it is defined as follows:

$$\boldsymbol{H}^* = \boldsymbol{H_m} - \boldsymbol{H_{bm}^T} \boldsymbol{H_b^{-1}} \boldsymbol{H_{bm}}. \tag{6.3}$$

This can be mapped onto the Cartesian space using the generalized Jacobian matrix defined as:

$$\boldsymbol{J}^* = \boldsymbol{J_m} - \boldsymbol{J_b} \boldsymbol{H_b^{-1}} \boldsymbol{H_{bm}}, \tag{6.4}$$

where $\boldsymbol{J_b} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{J_m} \in \mathbb{R}^{6\times n}$ are the Jacobian matrices of the base and manipulator, respectively. The generalized Jacobian is an useful concept because contains kinematics information of the base and it can be exploited to solve the control problem for the free-floating robot control [SK08].

6.2.1 Kinematics Model and Error Definition

The goal of the controllers is to minimise the error between the current pose to the target pose, or otherwise, to follow a reference trajectory. Fig. 6.1 shows the main frames which are considered in the design of the controllers. In particular, E represents the end-effector frame of the space robot, T the target frame and I the inertial frame. In the simulation case, the inertia frame is assumed to be the same as the base frame at the initial instant ($t_0 = 0$). The controllers will be validated on the OOS-Sim simulator, therefore the corresponding frames are reported as in Fig. 6.2. The main difference between Fig. 6.1 and Fig. 6.2 is given by the inertial frame I, located at the base of the industrial robot for the OOS-Sim. However, this will not affect the design of the controllers.

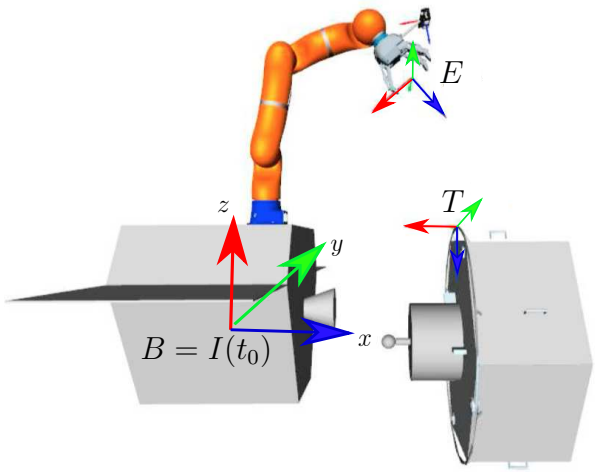


Figure 6.1: Frames for the free-flying space robot. B is the base frame and E is the end-effector frame, the inertial frame I is the same as B at the time $t_0 = 0$.

The velocity of the manipulator end-effector, $\dot{\boldsymbol{x}}_{\boldsymbol{m}} \in \mathbb{R}^6$, relates the kinematics between the operational space and the joint space and it is give as follows,

$$\dot{\boldsymbol{x}}_{\boldsymbol{m}} = \boldsymbol{J_b} \dot{\boldsymbol{x}}_{\boldsymbol{b}} + \boldsymbol{J_m} \dot{\boldsymbol{q}}. \tag{6.5}$$

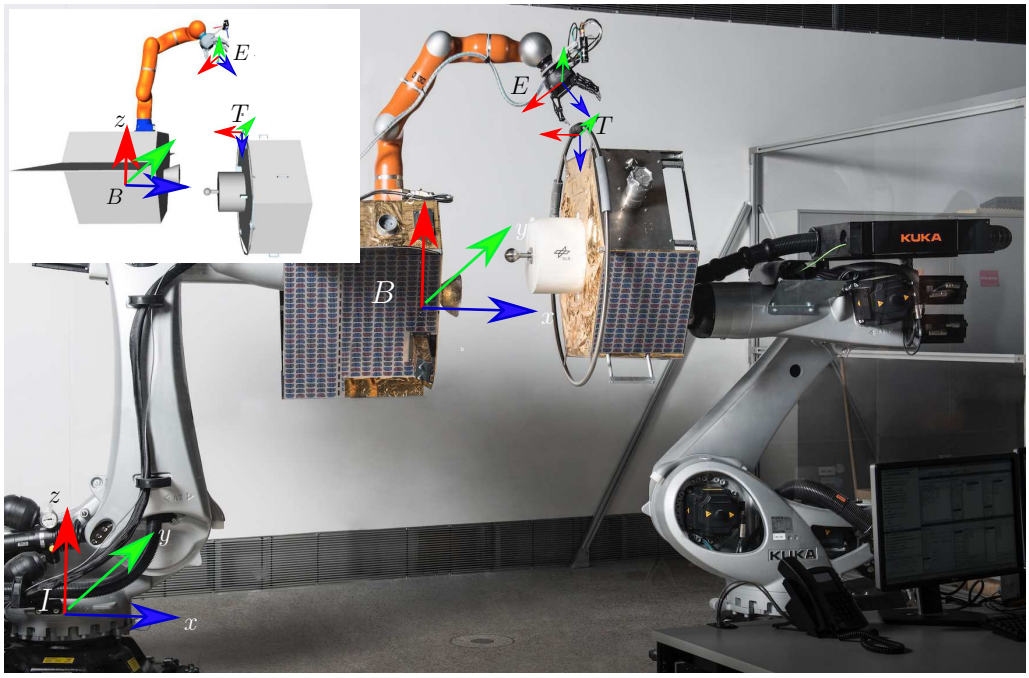


Figure 6.2: Frames in the OOS-Sim facility and in free-flying space robot. I is the inertial frame, B the base frame and E is the end-effector frame.

The pose-error between two frames can be written as a trivial concatenation of the position and orientation error. The position error between the end-effector and target position is expressed in the inertial frame as follows,

$$\Delta \boldsymbol{p_m} = \boldsymbol{p_t} - \boldsymbol{p_m}, \tag{6.6}$$

where $\boldsymbol{p_m}$ is the position of the end effector and $\boldsymbol{p_t}$ is the target position, both expressed in the inertial frame. The orientation error exploits the quaternions representation calculated from a relative error matrix. In particular, considering $\boldsymbol{R}_{e,i} \in \mathbb{R}^{3\times3}$ the rotational matrix of the end effector with respect to the inertial frame and $\boldsymbol{R}_{t,i} \in \mathbb{R}^{3\times3}$ the desired rotational matrix, expressed in the same frame. Then, the error matrix can be defined as $\mathbf{R}_\phi = \boldsymbol{R}_{t,i}\boldsymbol{R}_{e,i}^T$. Therefore, by exploiting the quaternion representation, a scalar η and a vector $\hat{\boldsymbol{\epsilon}} \in \mathbb{R}^3$ can be found, see [SK08]. Considering the matrix $\boldsymbol{E} = I_3\eta - \tilde{\boldsymbol{\epsilon}} \in \mathbb{R}^{3\times3}$, where $\tilde{\boldsymbol{\epsilon}}$ is the skew-symmetric matrix of quaternion vector $\hat{\boldsymbol{\epsilon}}$, the orientation error $\Delta\boldsymbol{\phi} \in \mathbb{R}^3$ is defined as,

$$\Delta \boldsymbol{\phi_m} = 2\boldsymbol{E}^{\boldsymbol{T}}\hat{\boldsymbol{\epsilon}}. \tag{6.7}$$

The total vector of the error, $\Delta\boldsymbol{x_m} \in \mathbb{R}^6$, concatenates the position and orientation of the manipulator end-effector and it is given as,

$$\Delta \boldsymbol{x_m} = [\Delta \boldsymbol{p_m}; \Delta \boldsymbol{\phi_m}]. \tag{6.8}$$

The vector (6.8) will be used for the design of the Cartesian impedance controller, later defined.

For the free-flying robot control, see Remark 1, the control of the actuators at the base requires also a definition of the error. This can be derived analogously to the concatenation of the error in position and orientation presented in (6.6) and (6.7). However, the respective position and orientation of the base must be considered. Therefore the definition of the error for the base results to be,

$$\Delta \boldsymbol{x_b} = [\Delta \boldsymbol{p_b}; \Delta \boldsymbol{\phi_b}], \tag{6.9}$$

where $\Delta \boldsymbol{p_b} = \boldsymbol{p_{tb}} - \boldsymbol{p_b}$ is the position error between the desired base position ($\boldsymbol{p_{tb}}$) and the measured one ($\boldsymbol{p_b}$). The orientation error for the base, $\Delta \boldsymbol{\phi_b}$, can be calculated from (6.7) with $\hat{\boldsymbol{\epsilon}}$ built from the matrix error $\boldsymbol{R_{\phi b}} = \boldsymbol{R}_{tb,i}\boldsymbol{R}^T_{b,i}$, being $\boldsymbol{R}_{tb,i}$ the desired base orientation matrix and $\boldsymbol{R}_{b,i}$ the measured one.

The representation of the Cartesian errors reported in (6.8) and (6.9) is used in the design of the impedance controllers (next section) and also in the next chapter.

6.3 Impedance Control for Free-floating Space Robot

One of the most important phases during the capture of a satellite by a space robot is the approach phase towards a target satellite, afterwards the grasp and stabilisation phases follow. During these phases the forces interaction between manipulator and target is unavoidable. Therefore, an impedance control can be considered in order to obtain compliance capabilities between end-effector and target.

In this section, impedance control available in the literature are implemented and validated for the free-floating case. In particular, two torque-based control algorithms acting in Cartesian space are designed for a free-floating robot. The first is a non-linear control, which uses the generalized Jacobian transpose, while the second is based on a complete feedback linearisation of the free floating robot dynamics. The analysed task is the tracking of a desired trajectory (e.g. provided by a motion planner) and the controllers must satisfy predefined compliance and impedance conditions for the manipulator end-effector. Both controllers are evaluated firstly in simulation. Furthermore, the control based on the generalized Jacobian matrix, which exploits the passivity of a classical PD control, is validated on the hardware. The latter, was used as baseline controller for the DEOS mission for its robustness. Moreover, it was also successfully utilised in teleoperation for the grasping of the target satellite using the OOS-Sim facility [ABDS+16].

6.3.1 Torque Controller Using the Generalized Transposed Jacobian

A simple controller which guarantees compliance behaviour between the end-effector and the target point is given by using the generalized Jacobian transposed (6.4), in this way, the end effector position is not disturbed by the reactive motion of the base. Thus, the manipulator can operate under a resolved motion control. If a redundant robot is considered as in this case, the motion in the null space of the manipulator is

enabled and it must be taken into account in the control law. Therefore, the torque control law which guarantees regulation towards the grasping point and control of the null space can be defined as follows:

$$\boldsymbol{\tau} = \underbrace{\boldsymbol{J^{*T}F}}_{\boldsymbol{\tau_1}} + \underbrace{(I - \boldsymbol{J}^{*T}\overline{\boldsymbol{J}}^{*T})\boldsymbol{\tau_n}}_{\boldsymbol{\tau_2}}. \tag{6.10}$$

where $\boldsymbol{\tau} \in \mathbb{R}^7$ is the input torques to the multibody dynamics (6.1). The control law in (6.10) is composed of two torque-vector terms. In particular,

- $\boldsymbol{\tau_1}$ is the torque contribution which controls the Cartesian pose of the end-effector. This is function of a Cartesian virtual force $\boldsymbol{F}$, later defined, which allows compliance between the end effector and the grasping point.
- $\boldsymbol{\tau_2}$ is the torque contribution to control the null space motion where a generalized joint torque vector $\boldsymbol{\tau_n}$ is considered.

The torque vector $\boldsymbol{\tau_1}$ is function of $\boldsymbol{J^{*T}}$, the transposed of the generalized Jacobian matrix in (6.4) and $\boldsymbol{F} \in \mathbb{R}^{6\times 1}$, the virtual control forces vector applied at the end-effector. The compliance during the approaching phase is provided by the virtual Cartesian forces vector $\boldsymbol{F}$ at the end-effector and it is modelled like a PD (Proportional Derivative) behaviour. Therefore, $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F} = \boldsymbol{K_{pm}}\Delta\boldsymbol{x_m} + \boldsymbol{K_{dm}}\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}. \tag{6.11}$$

The matrices $\boldsymbol{K_{pm}}$ and $\boldsymbol{K_{dm}} \in \mathbb{R}^{6\times 6}$ are positive definite and they represent the stiffness and damping gains of the controller. The vectors $\Delta\boldsymbol{x_m}$ is the position error defined in (6.8) and $\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}$ is the velocity error.

For the redundant manipulator (such as the LWR) the null space motion requires particular attention. In (6.10), the torque $\boldsymbol{\tau_2}$ is function of a $\overline{\boldsymbol{J}}^{\boldsymbol{*}} \in \mathbb{R}^{7\times 6}$, being the dynamically consistent generalized inverse matrix [RK92]. Indeed, the generalized Jacobian does not have a square structure and its inverse is not unique. Using dynamically consistent generalized inverse provides decoupling of Cartesian and null-space motion [Kha95]. The Jacobian $\overline{\boldsymbol{J}}^{\boldsymbol{*}}$ is therefore exploited and it is defined as:

$$\overline{\boldsymbol{J}}^{*} = \boldsymbol{H}^{*-1}\boldsymbol{J}^{*T}\boldsymbol{\Lambda}, \tag{6.12}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{6\times 6}$ is the inertia matrix in the Cartesian space defined as in [UY87] and reported below,

$$\boldsymbol{\Lambda} = (\boldsymbol{J^{*}H^{*-1}J^{*T}})^{-1}. \tag{6.13}$$

Therefore, a generalized torque can be applied in (6.10) to control the null space motion and it is defined as $\boldsymbol{\tau_n} = -\boldsymbol{D_n}\dot{\boldsymbol{q}}$, where $\boldsymbol{D_n} \in \mathbb{R}^{7\times 7}$ is a damping matrix. It is worth to note that $\boldsymbol{\tau_n}$ acts in the null space of the robot and it will not interfere with the end effector Cartesian motion.

The benefit of using the controller in (6.10) is that does not require a priory the knowledge of the system dynamic. The stability of this controller has been presented in [XS91] and it exploits the generalized dynamics of the system introduced in the previous session.

6.3.2 Torque Controller Using Inverse-Dynamics and Feedback Linearisation

A second approach is the torque controller based on computed torque using feedback linearisation. This approach leads to an exact linearisation of system dynamics obtained by means of a non-linear state feedback. The authors in [ALH06] define the equations of motion of a space robot in the operational space, as a function of the end effector acceleration and joint torques. Then, a controller based on feedback linearisation of the manipulator dynamics is employed.

The equation of motion for the system dynamics expressed in (6.2), can be also derived in Cartesian space. Then, the Cartesian end effector acceleration $\ddot{\boldsymbol{x}}_{\boldsymbol{m}}$ can be found as shown in [ALH06] and reported as follows,

$$\ddot{\boldsymbol{x}}_{\boldsymbol{m}} = \boldsymbol{J}^{*}\boldsymbol{H}^{*-1}\boldsymbol{\tau} + (\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda}_{\boldsymbol{b}}^{-1})\boldsymbol{F}_{\boldsymbol{e}} + \boldsymbol{\eta}, \tag{6.14}$$

where $\boldsymbol{\eta} \in \mathbb{R}^6$ is the Coriolis and centrifugal forces vector in Cartesian space and $\boldsymbol{\Lambda}_{\boldsymbol{b}} = (\boldsymbol{J}_{\boldsymbol{b}}^{*}\boldsymbol{H}_{\boldsymbol{b}}^{*-\mathbf{1}}\boldsymbol{J}_{\boldsymbol{b}}^{*\boldsymbol{T}})^{-1} \in \mathbb{R}^{6\times 6}$ is the inertia matrix of the base [UY89].

The approach considers to find a control vector $\boldsymbol{u}$, as a function of the system state, which realises an input/output relationship of linear type such that $\ddot{\boldsymbol{x}}_{\boldsymbol{m}} = \boldsymbol{u}$, where $\boldsymbol{u}$ is a desired control input behaviour. Therefore the torque $\boldsymbol{\tau}$ which linearises the system in (6.14) results to be,

$$\boldsymbol{\tau} = \boldsymbol{H}^{*}\overline{\boldsymbol{J}}^{*}\boldsymbol{u} + \boldsymbol{H}^{*}\overline{\boldsymbol{J}}^{*}[-\boldsymbol{\eta} - (\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda}_{\boldsymbol{b}}^{-1})\boldsymbol{F}_{\boldsymbol{e}}]. \tag{6.15}$$

Considering the case where the external force are $\boldsymbol{F}_{\boldsymbol{e}} = \mathbf{0}$, the torque controller in (6.15) can be substituted in (6.14) and considering (6.12), it is easy to see that the controller will linearise the dynamic. The output will result in the end effector acceleration expressed as:

$$\ddot{\boldsymbol{x}}_{\boldsymbol{m}} = \boldsymbol{u}. \tag{6.16}$$

The system under control (6.15) is linear and decoupled with respect to the new input $\boldsymbol{u}$. The control $\boldsymbol{u}$ can be chosen to have an impedance behaviour as,

$$\boldsymbol{u} = \ddot{\boldsymbol{x}}_{\boldsymbol{md}} - \boldsymbol{K}_{\boldsymbol{dm}}\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}} - \boldsymbol{K}_{\boldsymbol{pm}}\Delta\boldsymbol{x}_{\boldsymbol{m}}, \tag{6.17}$$

where $\ddot{\boldsymbol{x}}_{\boldsymbol{md}}$ is the desired acceleration, $\boldsymbol{K}_{\boldsymbol{pm}}$ and $\boldsymbol{K}_{\boldsymbol{dm}} \in \mathbb{R}^{6\times 6}$ are positive-definite diagonal matrices representing the stiffness and damping gains.

From the above consideration, asymptotic stability is guaranteed and the error dynamics will result:

$$\Delta\ddot{\boldsymbol{x}} + \boldsymbol{K}_{\boldsymbol{dm}}\Delta\dot{\boldsymbol{x}} + \boldsymbol{K}_{\boldsymbol{pm}}\Delta\boldsymbol{x} = 0. \tag{6.18}$$

Note that the implementation of this controller is based on the computation of the inverse dynamic and therefore it requires an accurate model of the system dynamics. This condition might be difficult to verify in practice and it might be also prohibitive to compute the inverse dynamics in real-time.

6.3.3 Torque Controller for Free-Floating Robot During Stabilisation

The controllers described above are generally used for the approach phase of the manipulator in Cartesian space towards the client. Once the end-effector of the manipulator reaches the grasping point within a given tolerance, the capture of the client can be performed and the stabilisation phase follows.

This section covers the design of an impedance control, which can be used during the stabilisation phase. The goal of this phase is to bring the joints velocity of the manipulator to zero [CBS+18]. In this way, the manipulator has a quasi-rigid connection between the servicer and the client. The design of the torque-based control, which can be used during the stabilisation phase, needs to fulfil two points listed as follows,

- the joint velocities have to decrease to zero in a stable manner,
- the commanded torque signals should not exceed the actuators maximum torque.

The first condition can be achieved by using a proportional derivative control architecture, similar to the PD structure for the approach phase, but in joint space. The second condition is dictated by the fact that during the stabilisation phase the manipulator is coupled with the client and extreme torques can be required by the joint actuators, which, however, have power supply limitation. Therefore, the controller during the stabilisation phase needs to take into account the maximum torque available for the actuators.

A saturation on the commanded torques could be applied. However, the use of a linear saturation to limit the torques does not ensure globally the achievement of the regulation objective and it might lead to undesirable effect in the closed-loop system [ZRS06a]. Then, a generalized saturation function, later defined, can be exploited to ensure stability as discussed in [ZRS06b].

The designed control law for the stabilisation phase has the following expression,

$$\boldsymbol{\tau} = -s(\boldsymbol{K_{dq}}\dot{\boldsymbol{q}} + \boldsymbol{K_{pq}}\Delta\boldsymbol{q}) = s(\boldsymbol{\tau}) \tag{6.19}$$

where $\boldsymbol{K_{pq}}, \boldsymbol{K_{dq}} \in \mathbb{R}^{7\times7}$ are positive definite diagonal matrix representing stiffness and damping. The vector $\Delta\boldsymbol{q} = \boldsymbol{q} - \boldsymbol{q_d} \in \mathbb{R}^7$ is the error between the measured joints position, $\boldsymbol{q}$ and the desired one, $\boldsymbol{q_d}$. The terms in parenthesis in (6.19) represent the classical PD control at joints level. Furthermore, the function $s(\boldsymbol{\tau})$ is a saturation function, which takes into account the torque limits of the joint actuators.

Stability of PD control with bounded input saturation has been proved in [ZRS06b]. The bounded input considers a special form of the natural saturation and the stability is proved under the condition that the saturation function for the PD torque controller must be *a strictly increasing linear saturation function* with bounded values[1].

[1]The reader can refer to Appendix C.1 for definition of generalized function and linear saturation.

Therefore, considering τ_i the i-th component of the torque vector expressed in (6.19), the strictly increasing linear saturation function is defined as,

$$s(\tau_i) = \begin{cases} -G_i + (Z_i - G_i)\ tanh\ \frac{\tau_i + G_i}{Z_i - G_i} & if\ \tau_i < -G_i \\ \tau_i & if\ |\tau_i| \leq G_i \\ G_i + (Z_i - G_i)\ tanh\ \frac{\tau_i - G_i}{Z_i - G_i} & if\ \tau_i > -G_i \end{cases} \tag{6.20}$$

where Z_i is a positive constant and it represents the joint torque limit that the i-th actuator can supply and G_i is a positive constant, where $G_i \leq Z_i$.

Therefore, the closed-loop dynamics with the controller (6.19) and the strictly increasing linear saturation function (6.20) results to be stable as proved in [ZRS06b] for PD control with input saturation.

6.3.4 Results: Simulations and Experiments

The Cartesian impedance controllers defined in Sec. 6.3.1 and Sec. 6.3.2 for the approach phase are firstly compared in a simulation study. Later, the non-linear controller will be validated on the hardware. The mass and inertia parameters of the simulated servicer satellite can be found in Table 6.1. The considered manipulation arm for the servicer satellite is a 7 dof light-weight-robot, whose mass and inertia parameter were identified and are reported in Table 6.2.

For the analysis, the error in orientation (shown in the following plots) is represented with the yaw, pitch and roll angles $[\psi, \theta, \phi]$. This representation is more intuitive than the quaternions representation exploited by the respective controller.

$M_b\ [kg]$	$I_x\ [kgm^2]$	$I_y\ [kgm^2]$	$I_z\ [kgm^2]$
150	38	20	23

Table 6.1: Mass and inertia properties of the servicer satellite (base of the LWR).

$M_{link}\ [kg]$	$I_x\ [kgm^2]$	$I_y\ [kgm^2]$	$I_z\ [kgm^2]$
2.71	0.023	0.023	0.005
2.71	0.024	0.005	0.024
2.54	0.013	0.013	0.005
2.50	0.023	0.005	0.002
1.30	0.023	0.022	0.003
1.57	0.003	0.003	0.003
4.1	0.024	0.002	0.024

Table 6.2: Mass and inertia properties of the LWR servicer arm

Simulations

The task is to approach the target while considering a Cartesian trajectory for the end effector, which is shown in Fig. 6.3. Usually, in the autonomous mode, the trajectory can be provided by a motion planner in order to guarantee feasibility with respect to motion constraints, such as singularities, collision avoidance and end-effector camera field of view [LH13].

The considered impedance is described by the stiffness and damping gains. For the first Jabobian-based controller, a stiffness and damping of $\boldsymbol{K_{pm}} = diag[600, 600, 600, 80, 80, 80]$, $\boldsymbol{K_{dm}} = diag[25, 25, 25, 4, 4, 4]$ has been implemented. For the second linearisation-based controller, optimal gains were derived from the case of a unitary mass with a damping ratio $\xi = 0.7$ and frequency $f = 10Hz$, resulting in $\boldsymbol{K_{pm}} = diag[40, 40, 40, 40, 40, 40]$ and damping $\boldsymbol{K_{dm}} = diag[13, 13, 13, 13, 13, 13]$. Both simulations were performed for the same reference trajectory.

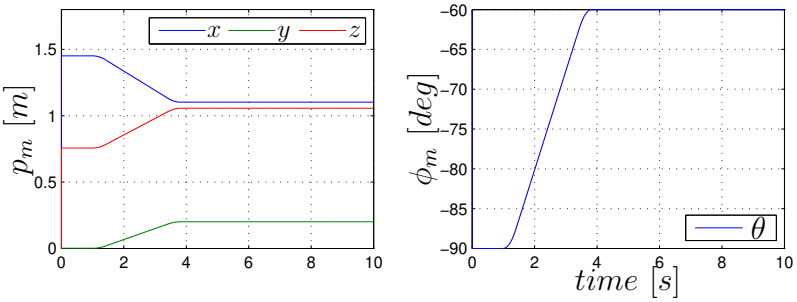


Figure 6.3: Input end effector trajectory referred to the inertial frame. p_m is the position of the manipulator and ϕ_m the orientation.

Torque controller using the generalized transposed Jacobian

The behaviour of the torque control law given in (6.10) is here analysed. The error detected during the tracking is shown at the top of Fig. 6.4. The maximum error detected is 0.007 m in position and 0.01 deg in orientation during the tracking phase. The plot at the bottom of Fig. 6.4, is the measured state of the base in position and orientation, due to the reaction motion of the LWR. Despite the controller solves only a regulation problem, the tracking performance are satisfactory for the given trajectory.

Torque controller using the inverse dynamics linearisation

The simulation considers the torque input as reported in (6.15) for the same servicer manipulator and trajectory. Since the manipulator is a redundant robot, the null space motion is damped by adding the generalized torque $\boldsymbol{\tau_2}$ reported in (6.10), which will not affect the motion in Cartesian space. The error detected during the tracking is shown in Fig. 6.5 (top). During the tracking phase the maximum error is found to be 0.004 m for the position and 0.003 deg for the orientation. In the steady

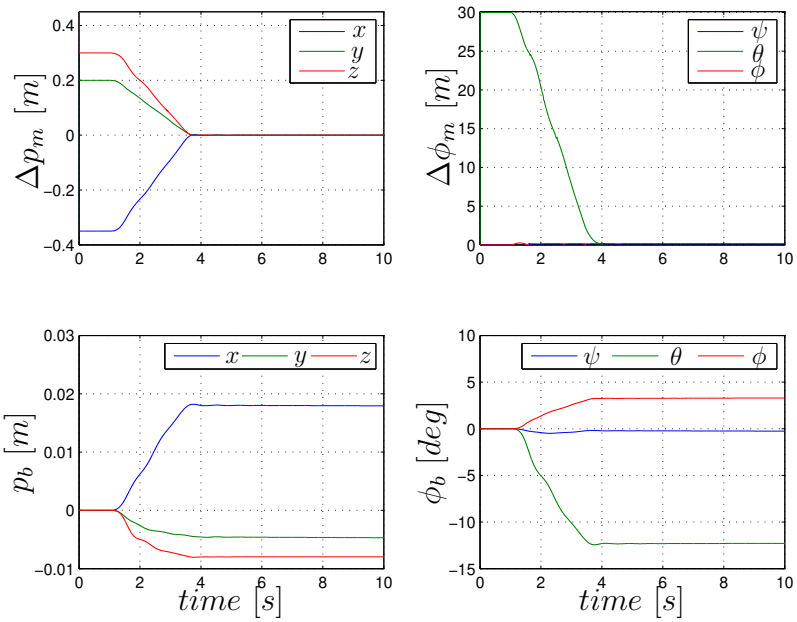


Figure 6.4: End effector error with the torque control using the transpose of the generalized Jacobian (top) and motion of the base (bottom).

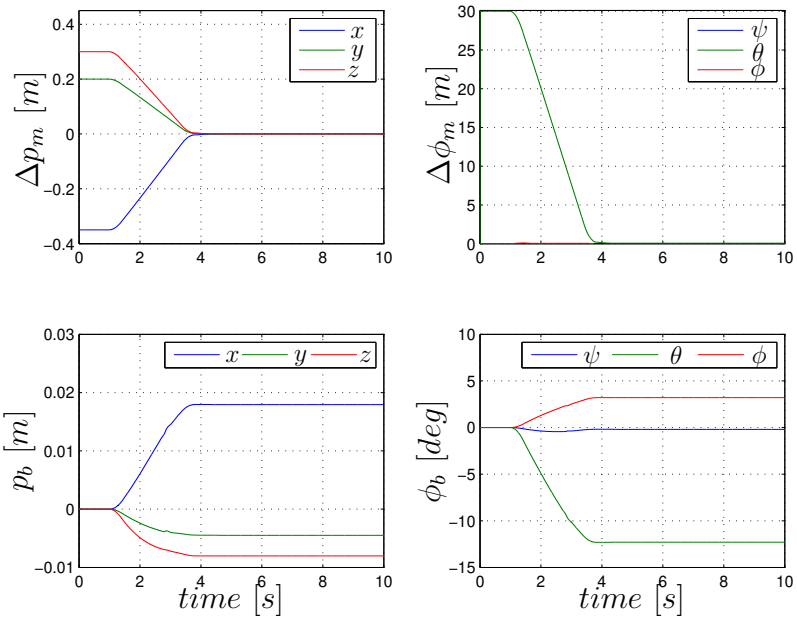


Figure 6.5: Error of the manipulator with the torque control base on the inverse dynamics linearisation (top) and relative motion of the base (bottom).

state it is 10^{-6} m in position and 0.001 deg in orientation. The state of the base due to the LWR motion are shown in Fig. 6.5 (bottom). The simulation shows that the controller based on inverse dynamics has good performance (even using lower gains) in case of a rigid joints robot. However, if unmodeled dynamics are not taken into account, the linearisation will not be perfect and the controller might behaves

differently, see [ASOH07] for a complete treatment.

Furthermore the control law based on feedback linearisation results to be computationally heavy. For this reason, in the experiment only the torque controller based on the generalized Jacobian transpose is considered. Indeed, this control law is not function of dynamic model and therefore it can provide higher robustness to the unmodelled dynamics [ASOH07].

Experiment

The experiment is performed on the servicer robot of the OOS-Sim facility with the torque control law described in (6.10). The torque gravity vector $\boldsymbol{g}$ has been added to compensate the gravity effects on ground. The gains $\boldsymbol{K_{pm}}$ and $\boldsymbol{K_{dm}}$ are the same as those used in the simulation.

The initial error is of the manipulator is $[-0.05\ 0.10\ 0.2]$ m in position and $[0\ 20\ 0]$ deg in orientation and a linear trajectory is provided. The end effector follows the defined trajectory and it motions is shown in Fig. 6.6. The data show a maximum error of 0.008 m in position and 0.8 deg in orientation during the tracking phase. Furthermore, after the manoeuvre, the maximum error found experimentally is 0.004 m in position and 0.4 deg in orientation at steady state. The difference with the analysed case in simulation is given by the presence of joints friction and not perfect gravity compensation which is intrinsic in the hardware. Moreover, the measured motion of the base due to the reaction of the LWR motion, is shown in Fig. 6.6 (bottom).

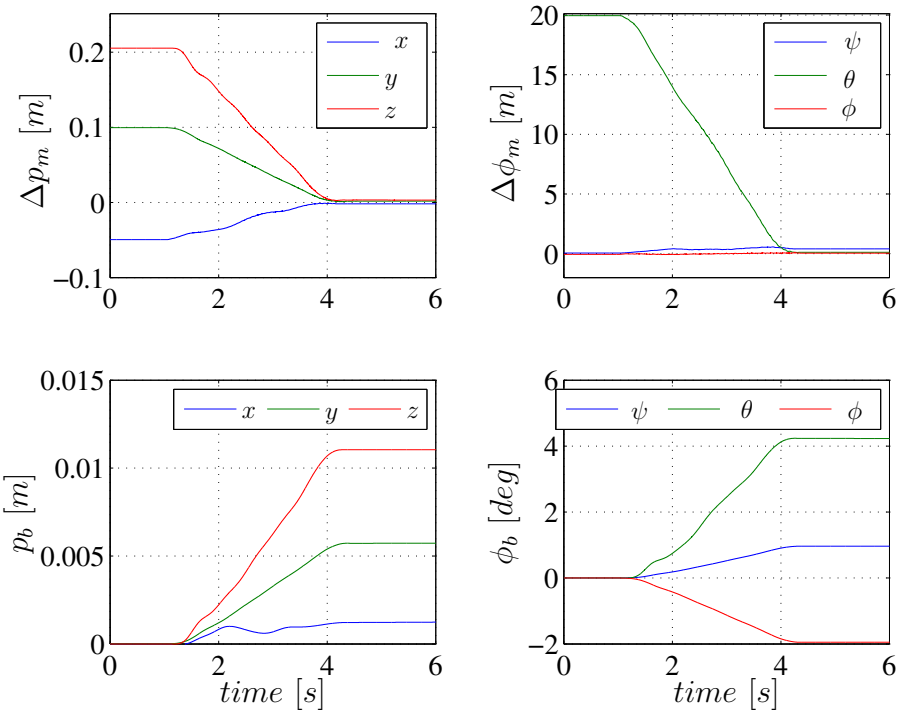


Figure 6.6: Experimental results: Tracking error in position and orientation

A second experiment has been performed and snapshots are shown in Fig. 6.7. The first snapshot shown the initial position of the end-effector and the second snapshot shows the final position. The reader can clearly see the dynamic nature of the free-floating control by seen the motion of the base, which results as dynamic reaction of the light-weight-robot motion.

The results shown above represent the experimental proof that satisfactory performance can be achieved using the free-floating robot control with the generalized Jacobian matrix. Since the robot is controlled in torque mode, uncertainties given by the gravity compensation and by the joints friction are intrinsic in the system. Nevertheless, the controller is able to deal with them adequately. Therefore, this controller was considered also for performing experiment in teleoperation, as it will be presented in the following subsection.

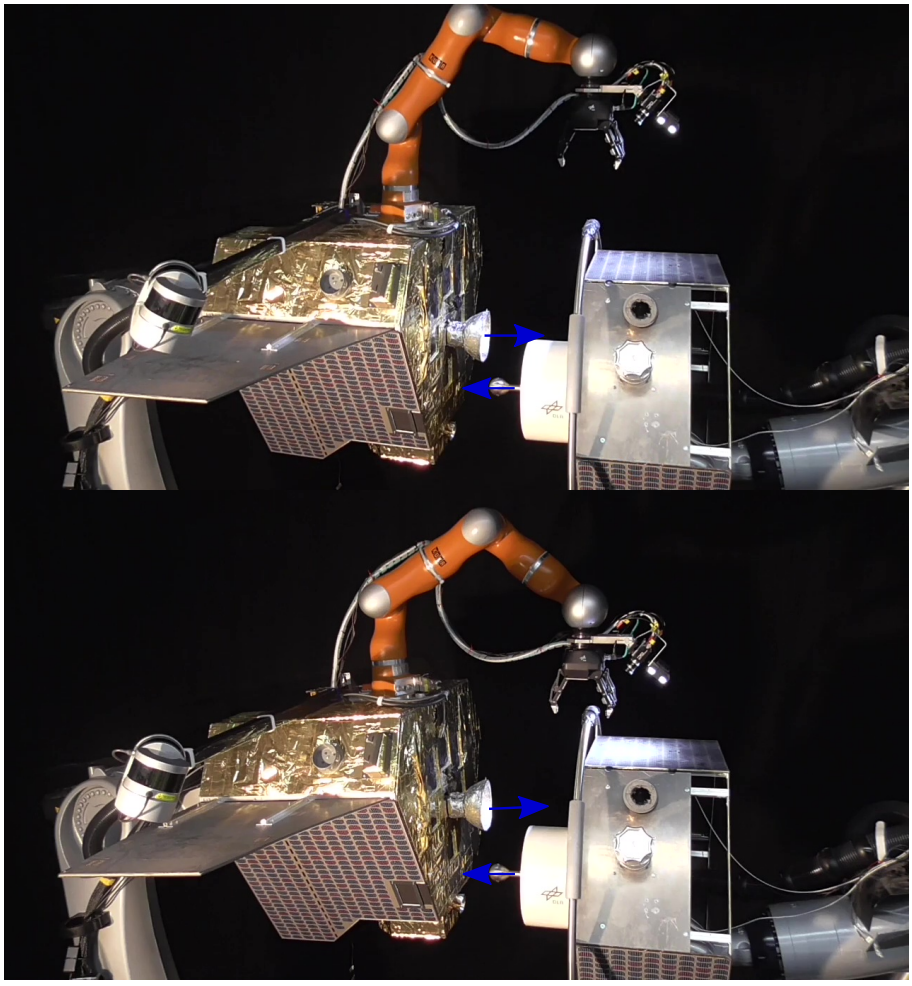


Figure 6.7: Snapshots of the experiment for the free-floating control using the Generalized Jacobian transpose. The base of the servicer satellite (left) moves accordingly to the LWR motion.

6.3.5 Experiment with the ASTRA Geostationary Satellite

The free-floating impedance control using the generalized Jacobian was validated also in a complete teleoperation scenario. Teleoperation represents indeed an other mode to control the space manipulator, namely the slave, while using a haptic interface, namely the master.

For an on-orbit servicing mission the target satellite is considered to be in a Low Earth Orbit (LEO) and the master interface is located on ground. An user interface (master) is placed at the on ground station and it is composed of the following hardware components:

1. Master device: a Light-Weight Robot (LWR) is employed as the master device. This choice is motivated by the fact that the LWR can offer large workspaces and high levels of force interactions, therefore, the human operator can intuitively manipulate the slave robot. A six dof forces and torques sensor is also mounted on its end effector in order to capture the human forces applied during the telemanipulation.

2. Data-Glove: In order to control the gripper attached to the end-effector of the slave robot, the operator wears a data-glove, which provides local measurement of the finger positions. These measurements are mapped into the gripper kinematics for remote control.

3. Visual display/Mission Control panel: In order to provide visual feedback from the slave environment to the operator, a display station is posed at the user interface.

The teleoperation experiment is performed with a real communication link between master and slave. Common communication links between Low Earth Orbit satellite and on-ground station are based on a Geostationary Earth Orbit (GEO) relay infrastructure. With this structure the data are sent from ground to a LEO satellite through a GEO satellite. As a results, the total communication interval with LEO increases, but it comes at the cost of more time delay.

During the mission, the operator commands through the master a position to the slave using a communication link accordingly to the GEO relay structure, as shown in Fig. 6.9a. The operator receives visual and force feedback from the slave.

The slave considered in the experiment is the space robot of the OOS-Sim facility, which was physically located in DLR Oberpfaffenhofen (Germay), but its signals are linked with the ASTRA satellite. The links with the ASTRA satellite is provided by LRT (Garching, Germany) and a standard UDP segment connects DLR and the LRT. In the experiment the master device is located on-ground in DLR and it is connected to the OOS-Sim facility through the satellite communication link described above, the final implemented scenario is summarised in Fig. 6.9b. The measured mean value which characterises the delay in the signals was 269.72 ms with a standard deviation of 3.11 ms for each channel in forward direction. More details about the the communication link are reported in Table 6.3.

Delay Mean [ms]	*jitter[ms]*	*Pack.lost*	*Packet out of order*
269.72	3.11	2.12	24.54

Table 6.3: Communication link properties of the ASTRA satellite

To deal with the time delay due to the communication, approaches based on time domain passivity are employed. In particular energy observer and passivity controller can be designed exploiting the methods presented in Chapter 3. The teleoperation control used for the experiment was composed of a 4-Channel architecture. It is defined by two forward channels carrying master position and force signals, and two feedback channels, carrying slave computed and measured force signals. The reason for choosing this architecture instead of the symmetric 4-Channels, with position and force in both directions, is to reduce the spring-like characteristic caused by the delayed position closed loop, which can be disturbing, specially for substantial or varying delays (more details are provided in [ABDS+16]).

Therefore the control law in (6.10) augmented with the passivity control was implemented as input to the slave robot. Results of the experiment are shown in Fig. 6.8. The dense points visualise the position of the desired final state (almost zero relative motion of the satellites). The phase during which there is a mismatch between the positions of slave and target corresponds to the time that the fingers of the gripper approach the target satellite. It can be observed that grasping and stabilisation of a free-floating target using a free-floating manipulator was feasible using the controller based on the generalized Jacobian.

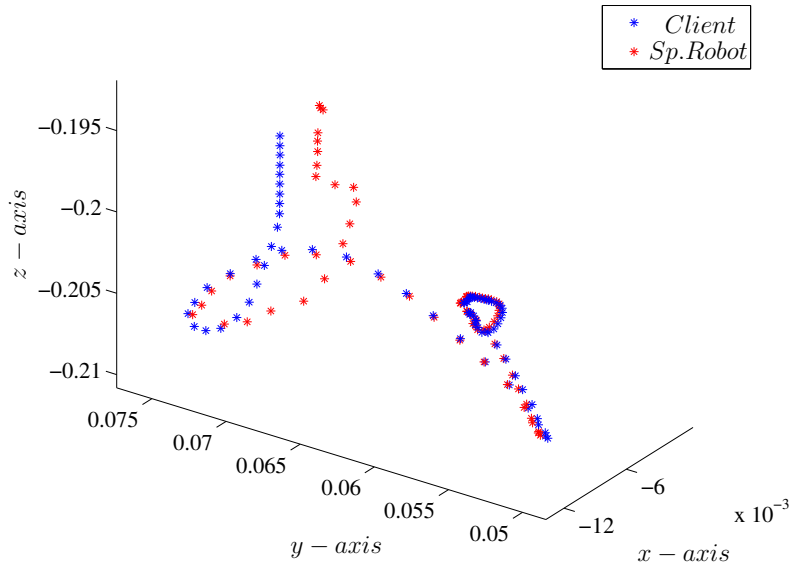
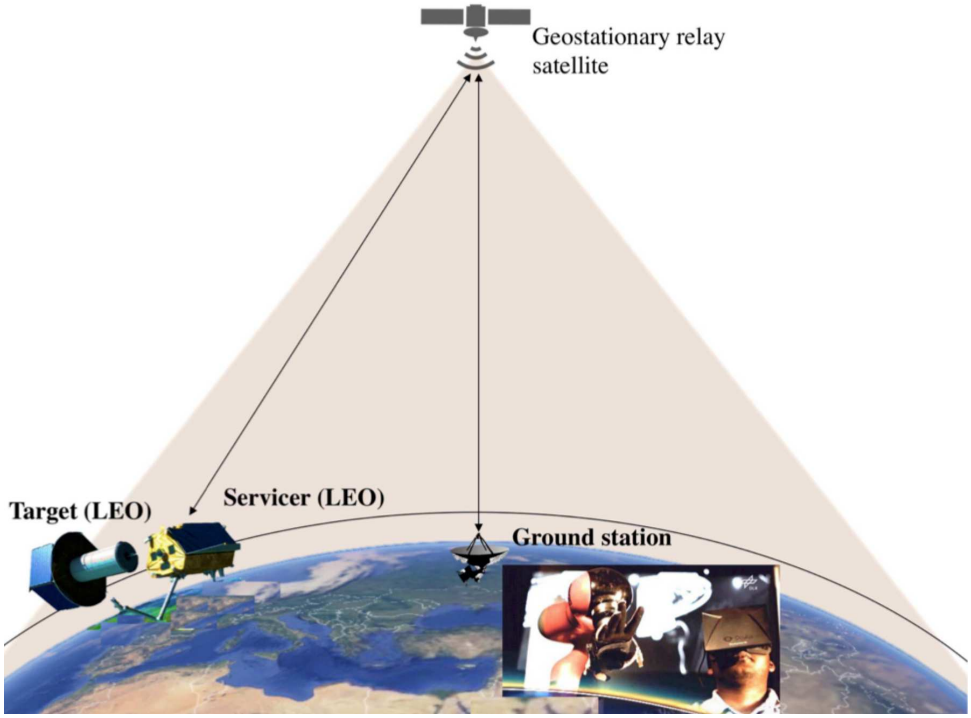
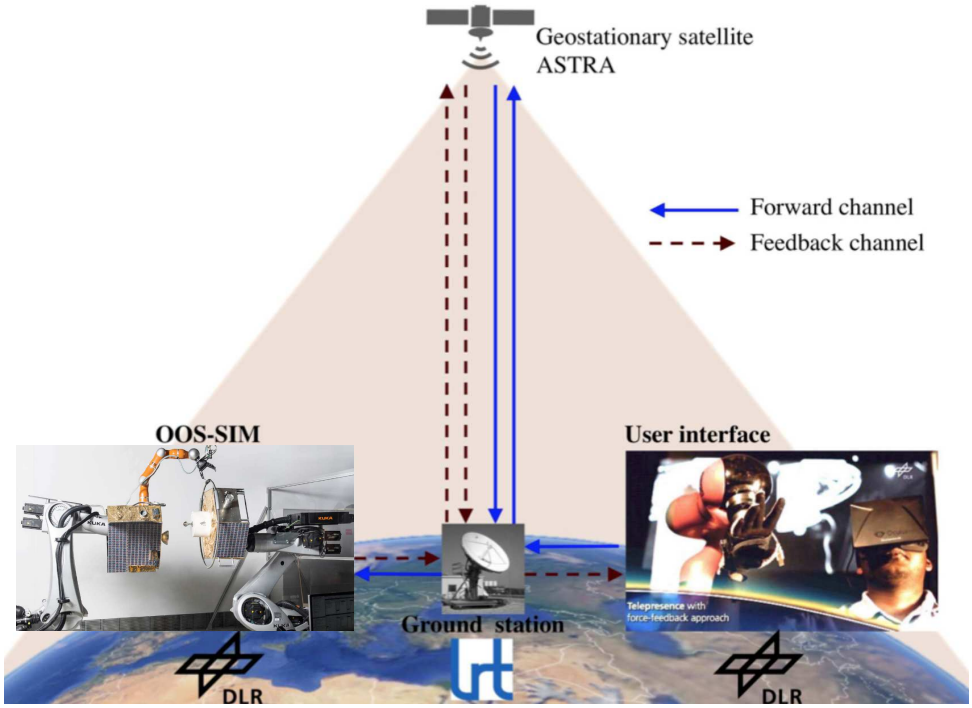


Figure 6.8: Experiment: 3D Plot of Grasping and Stabilization of the target with the OOS-Sim facility.



(a) Targeted scenario in space: The space robot controlled from the on-ground station.



(b) Implemented scenario: The OOS-Sim (space robot) controlled from the on-ground station with the user interface and ASTRA satellite link.

Figure 6.9: Teleoperation experiment using the ASTRA satellite as GEO-relay.

6.4 Impedance Control for Free-flying Space Robot

In free-flying control mode, the base of the space robot is actuated and controlled together with the manipulator arm. In this section the torque control based on the generalized Jacobian is extended and validated in a simulation study. In particular, the considered phases are the approach and stabilisation of a defective satellite, Envisat, as a part of the e.Deorbit and COMRADE projects [TAE+17], [CBS+18]. The aim of the approach phase is to control the manipulator end-effector to the grasping point, which is located at the Target's launch adapter ring (LAR) while controlling the base. During the approach phase of the manipulator to the grasping point, the arm movement introduces disturbance forces and torques on the base.

In the design of the controller for the approach phase, a coupled control strategy has been considered as interface of two systems, i.e. the base and the manipulator. The manipulator controller regulates the pose of the end-effector and it runs with a frequency of 1 KHz. This controller uses the impedance control law reported in (6.10) and experimentally validated. The control at the base controls the attitude of the chaser and it operates relative to the target. Both controllers will exchange data. Position and orientation data between the chaser and the target are provided by the chaser to the manipulator controller. On the other hand, the robot controller will exchange data with the base by means of forces and torque computed at the robot base, calculated as,

$$\boldsymbol{F_{rb}} = -\boldsymbol{H_{bm}}\ddot{\boldsymbol{q}} - \boldsymbol{c_b}, \tag{6.21}$$

where $\boldsymbol{F_{rb}} \in \mathbb{R}^6$ is the reaction force of the manipulator to the base. The actuation of the base exploits a PD (Proportional-Derivative) behaviour and it receives as input the disturbance force due to the manipulator motion, $\boldsymbol{F_{rb}}$ in (6.21). The base controller provides a control force $\boldsymbol{F_b} \in \mathbb{R}^6$ as input to the system dynamics defined in (6.1).

The controller acting at the base is therefore defined as follows,

$$\boldsymbol{F_b} = -\boldsymbol{F_{rb}} + \boldsymbol{K_{pb}}\Delta\boldsymbol{x_b} + \boldsymbol{K_{db}}\Delta\dot{\boldsymbol{x}}_{\boldsymbol{b}}, \tag{6.22}$$

the matrices $\boldsymbol{K_{pb}}$ and $\boldsymbol{K_{db}} \in \mathbb{R}^{6\times6}$ are positive definite and they represent the stiffness and damping gains of the controller at the base. $\Delta\boldsymbol{x_b}$ is the error at the base defined in (6.9) and $\Delta\dot{\boldsymbol{x}}_{\boldsymbol{b}}$ the velocity error. The controller at the base usually runs between $1Hz$ and $10Hz$, which is standard in such kind of scenario [TEDS+17].

The controller of the manipulator exploits the torque law $\boldsymbol{\tau}$ presented in (6.10), however a saturation on the virtual Cartesian force $\boldsymbol{F}$ was imposed to control the end-effector acceleration. The force $\boldsymbol{F_b}$ and the torque $\boldsymbol{\tau}$ are the control input to the space robot dynamics in (6.1). The robustness of this controller strategy has been validated in a Montecarlo simulation study considering the approach and grasping/stabilisation of the target satellite Envisat.

Approach phase

The control for the approach phase has been validated in simulation where the base controller operates at $10Hz$ and the manipulator control runs at $1kHz$. The motion of the target, Envisat satellite, is initialised considering the worst case scenario described in the e.Deorbit project, i.e. a spinning of of 5 deg/s along its inertial x-axis. The aim is to analyse the accuracy of the gripper positioning with respect to the grasping point, while the base controller is actively stabilising the relative pose of the spacecraft. Further details can also be found in [TEDS+17] and in [TAE+17]. The gains of the manipulator controller are set as $\boldsymbol{K_{pm}} = diag(350, 350, 350, 50, 50, 50)$, $\boldsymbol{K_{dm}} = diag(25, 25, 25, 15, 15, 15)$ and for the null space motion the damping matrix is set as $\boldsymbol{D_n} = diag(2, 2, 2, 2, 2, 2, 2)$.

Fig. 6.10a (top) shows the error of the manipulator with respect to the LAR in position and orientation, respectively. Fig. 6.10a (bottom) shows the error in position and orientation of the controller acting at the base. Fig. 6.10b (top) shows the commanded robot joint torques of the controller in (6.10). The actuation forces of the controller at the base are shown in Fig. 6.10b (bottom). The robustness of this controller was proved with more than 100 Montecarlo simulations considering variation in mass parameters, initial conditions and variation of the Envisat motion. The final error in position was bounded to 0.005 m and 0.5 deg, see Fig. 6.11b. The final position was reached within the requirements of 0.01 m and 2 deg, as it was reported for the e.Deorbit project in [TEDS+17, JLR+18].

Stabilisation phase

After the approach phase, the stabilisation phase begins. During this phase the thrusters of the servicer are turned-off and the system is in a free-floating mode. Therefore, the control of the manipulator developed in Sec. 6.3.3, (see the control law (6.19)) for the free-floating case can be applied during this phase. The goal of the stabilisation control is to reduce the joint velocities towards zero.

A Montecarlo analysis has been performed considering variations in the inertial parameters of Envisat mass, center of mass, inertia and including also sloshing effects and flexibility. The results of the statistical analysis are reported in Fig. 6.12a where the mean value of the residual joint velocity results to be 0.08 deg/s, which might results from the tumbling of the combined system. During the stabilisation phase, the maximum torque value (measured along the joints actuation axis) was 53.08 N/m in the 4th axis. The results of all the joint maximum torques are reported in Fig. 6.12b. It can be concluded that the stabilisation control respects the torque limits of the actuators, which were considered in the design of the controller.

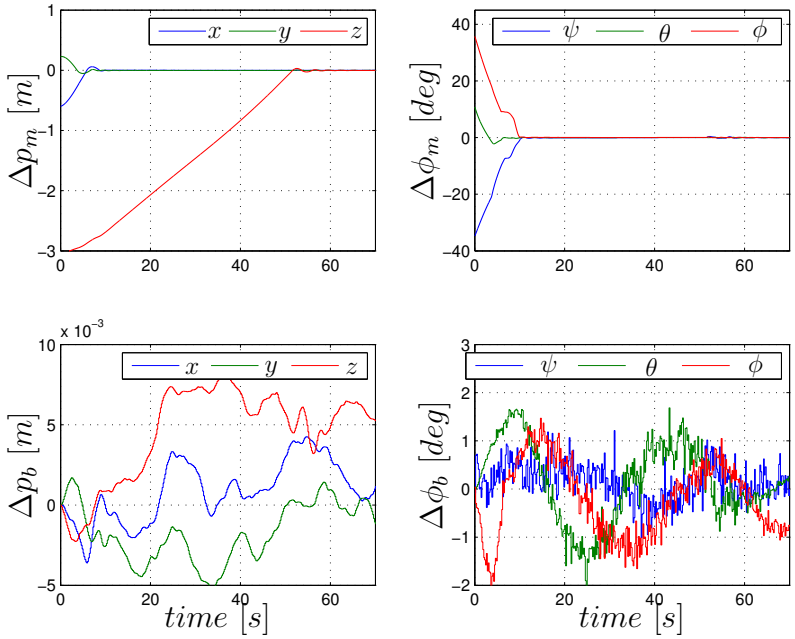
Snapshots of the manoeuvre simulated in this section are shown in Fig. 6.11a. The servicer manipulator follows the Envisat motion using the developed free-flying controller and it reaches the launch adapter ring, after that the grasping is performed and the stabilisation control acts.

6.5 Discussion

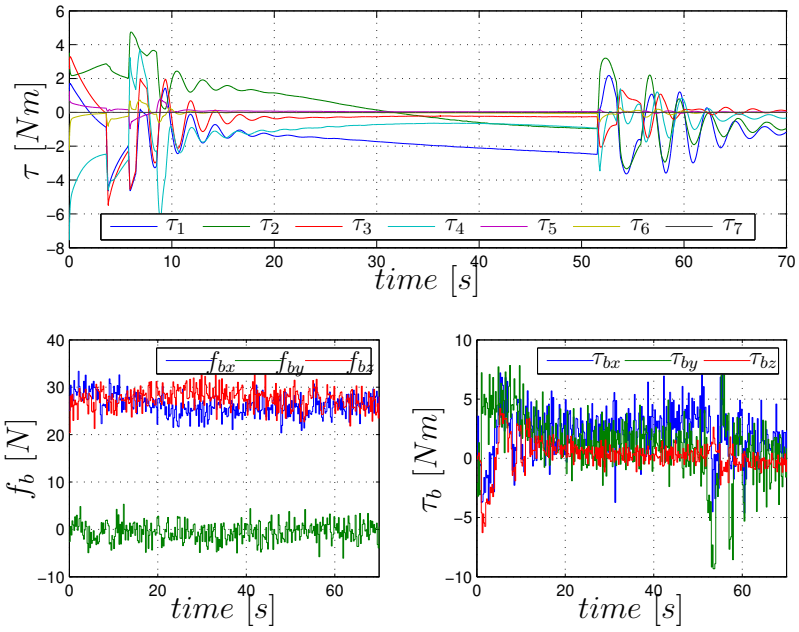
Impedance controllers offer a compliance behaviour needed during the grasping and stabilisation of a target satellite. In the first part of this chapter, a comparison between two torque-based control algorithms for a free-floating robot has been performed. The first controller exploits the generalized Jacobian transpose, while the second is based on a complete feedback linearization of the free floating robot dynamics. The analysed task was to follow a desired trajectory and the controllers need to satisfy predefined compliance and impedance conditions for the manipulator end effector. The torque controller based on the generalized Jacobian transpose results less sensitive to model uncertainties and it has been tested on the OOS-Sim facility. Further, the same controller resulted effective also for a teleoperation scenario, where the OOS-Sim servicer robot was controlled with a master device through the ASTRA satellite communication link.

In the last part of the chapter, the impedance controller was extended to the free-flying case. Manipulator and base were controlled in a coupled manner to perform the approach of a client satellite, Envisat. The validation was performed within the e.Deorbit and COMRADE space projects and the end-effector accuracy adhered to the imposed requirements. Furthermore, the impedance control developed for the stabilisation phase of the client was also validated and it respected the torque limits imposed by the actuators. The robustness of the impedance control for the free-flying case was proved by Montecarlo simulations.

In the next chapter, the theoretical treatment with stability proofs for the free-flying robot control will be introduced and validate with experiments. In particular, the potential stability issues of the base and manipulator controllers will be tackled.

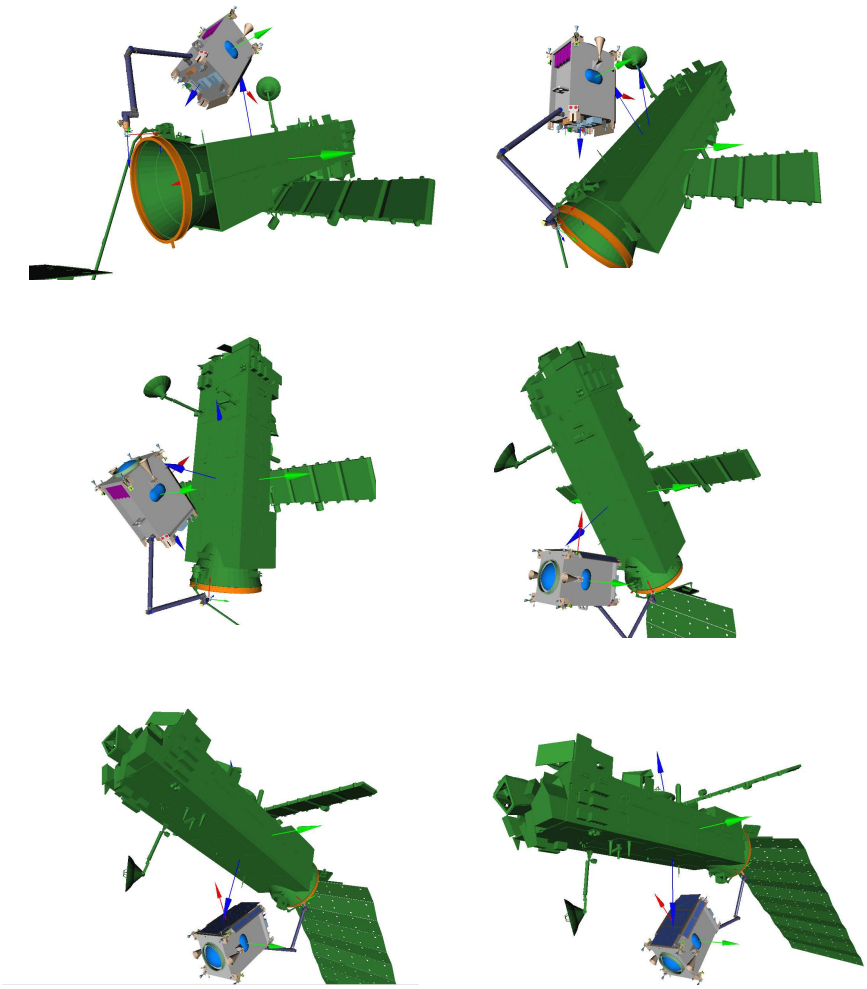


(a) Error of the manipulator (top) and of the base (bottom).

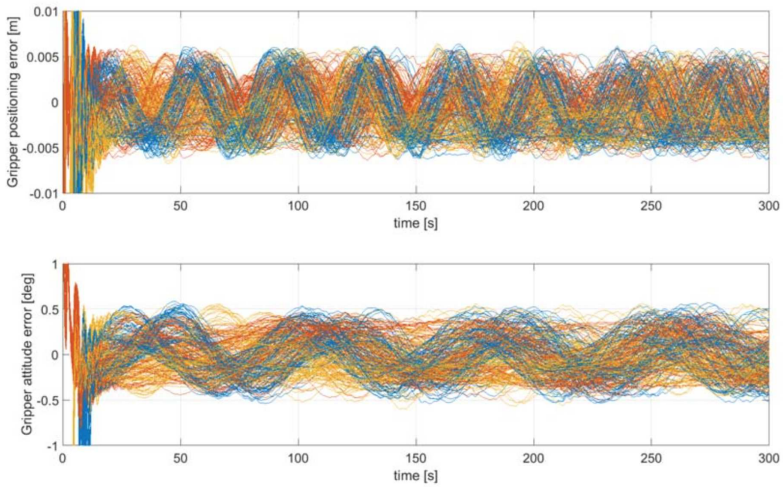


(b) Torque of the manipulator (top) and force at the base $F_b = (f_b, \tau_b)$ (bottom).

Figure 6.10: Approach control during the Envisat motion. Error of the manipulator and the base with corresponding joint torques and forces at the base.

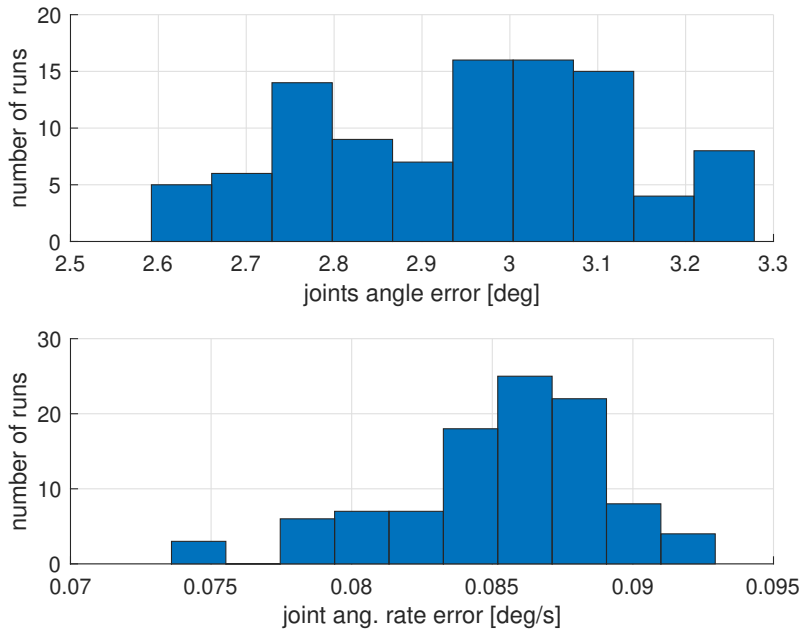


(a) Snapshots of the simulation environment: free-flying control during the approach and stabilisation of Envisat.

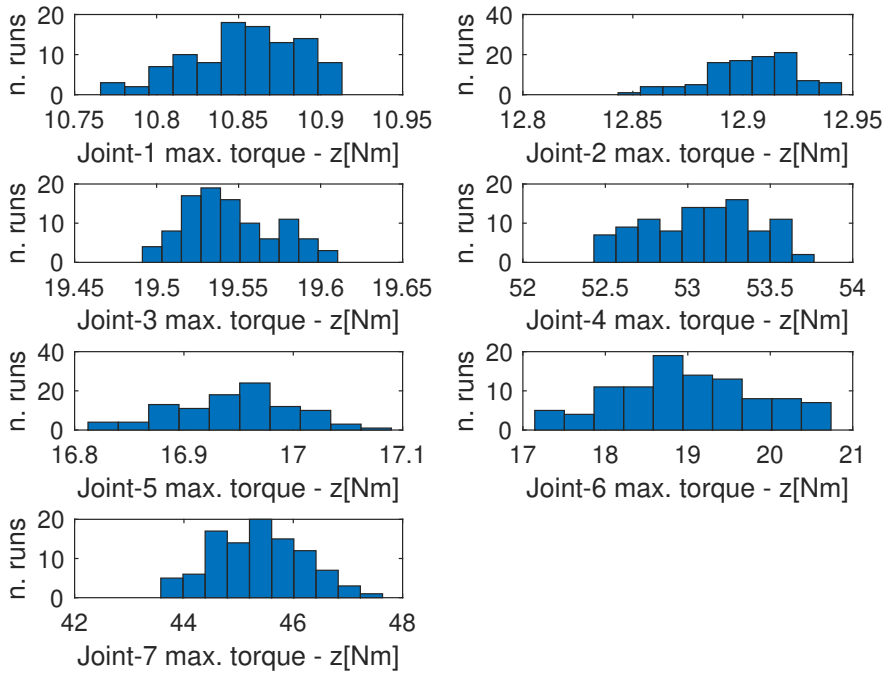


(b) Montecarlo analysis for the developed impedance control. Position error (top) and orientation error (bottom).

Figure 6.11: Viewer of the manoeuvre (top) and Montecarlo analysis for the final stage of the approach control (bottom).



(a) Maximum value of position and velocity of joints during the stabilization phase.



(b) Measured torques along the actuation z-axis of the joints.

Figure 6.12: Statistical analysis for the joint velocities and torques during the stabilisation phase.

CHAPTER 7

IMPEDANCE CONTROLLERS FOR SPACE ROBOTS UNDER MULTI-RATE EFFECTS

7.1 Introduction

The controller of a robot is usually designed to operate on a single control unit with high-enough rate in order to disregard the effects caused by discrete signals. In this case, the controller can be designed in continuous-time and the discrete-time effects can be neglected. In some applications, a robotic system can be composed of multiple sub-systems, which are physically coupled and controlled. Each sub-system might be controlled with different control units, which might run at different rate. For example, some humanoid robots belong to this group [ON08]. Therefore, the continuous-time control assumption considered in the design of the controller is not valid anymore.

As shown in the previous chapter, the control architecture of a free-flying robot is characterized by two systems. One is the controller at the base of the manipulator, which can be used to enable coarse positioning and the second one is the control of the manipulator, which enables fine dexterous control for complex tasks (e.g. satellites recovery). In particular, the control system of the base, namely the GNC (Guide Navigation and Control), runs at a low rate, between 1 Hz and 10 Hz, while the controller of the manipulator runs at a much higher rate (usually 1000 Hz), see [TEDS+17] and [TAE+17] for details about GNC and control requirements. The different rates of the controllers, possible delays in the communication between the control units and the presence of zero order holds can jeopardise the stability of the overall system, if not properly taken into account.

In this chapter, the stability issues due to the different rates of the two controllers are analysed from an energetic perspective. First, an impedance control which resolves a regulation problem for the base and the manipulator is proposed. The designed controller is firstly proved to be stable in continuous-time. Later, the

feedback in the control loop of the base is modified to maintain stability properties which are broken when the controller at the base runs at low rate.

Secondly, the impedance controller is extended to the tracking case for the manipulator which is a typical scenario in space while approaching a moving target. In particular, besides guaranteeing trajectory tracking for the manipulator and regulation for the base, two passivity controllers are designed. One acts on the base and the second acts on the manipulator in order to maintain the passivity of the overall system, independently of the sampling time. Rigorous stability proofs are provided for the designed controllers, which are also experimentally validate on the OOS-Sim servicer robot.

Related Work

The interaction between satellite-base control and manipulator control has been firstly studied in the context of the Shuttle Remote Manipulator [LLZ87]. A centralised control strategy for base attitude, base position and arm end-tip control has been studied in [SA90], resolving the redundancy of the whole system at velocity level.

For the free-floating control, a strategy was proposed in [MMA89] to reduce fuel consumption. The approach was based on a transposed Jacobian algorithm, where no contacts and zero momentum was assumed. The problem of the impedance control of a free-floating robot was treated in [NY06] and [NP17] using feedback linearisation. In [UY89], a control based on a generalized Jacobian matrix is proposed and [GGDS$^+$16] extends this approach to the nonzero momentum case. However in these works the base of the manipulator was not controlled.

In the direction of simultaneously control the base and manipulator (known also as coordinated control), [Yos94] proposes a momentum-based strategy using reaction wheels and [PD91] introduces a coordinated controller scheme based on feedback linearisation. For the tracking control, [PP88] proposes a passivity-based tracking controller, however only for a fixed-base robot and the tracking control was performed in joint space. A trajectory tracking control for a space robot was proposed in [LMO$^+$18], but considering an unactuated base.

Although the regulation and tracking control of a free-flying space robot has been tackled previously in literature (see [NP17], [GGDS$^+$16], [Yos94], [PD91], [SA90], [LLZ87]), the specific issue related to the multi-rate control was not addressed. From a mission perspective point of view, the problem associated with the different rates controller between the base and robotic arm (for the approach and capture phase) has been underlined as an important factor for a space mission, see for example [ESA15, §3.2.2] and [TEDS$^+$17]. Indeed, due to the GNC system constraints, the control of the base needs to run at low rate and the controller of the manipulator at higher rate [TEDS$^+$17]. Therefore, the discrete nature of the controllers and the multi-rate effects must be taken into account during the design of the controllers because they can jeopardise the stability of the system [DNBS06].

Within this context, a first approach which deals with the multi-rate effect for a

space robot is presented in this chapter. Firstly, an impedance control is proposed for the manipulator and base which resolves the regulation problem. In particular the effects of the multi-rate are analysed from an energetic perspective and passive control are proposed. The reader can refers to the work published by the author in [DSBG+18]. Secondly, the impedance controller is extended to the trajectory-tracking for the manipulator while regulating the base, which is needed in the approach phase of an OOS mission. The reader can refer to the work published by the author in [DSMB+19]. The controllers, which will presented in this chapter, exploit the time-domain passivity approach extensively presented in Chapter 3 and Chapter 4.

7.2 Multi-rate Regulation Control for a Free Flying robot

In this section an impedance controller is proposed for the regulation of the base and the manipulator of a space robot. Firstly, the design is performed in continuous-time and stability of the controller is proved. Second, the effects of the multi-rate are discussed and presented in the problem statement section. The controllers are designed for the free-flying robot dynamics system of the manipulator mounted on the satellite base, which is reported in (6.1).

The Cartesian motion of the manipulator is controlled in torque mode with a simple transposed Jacobian law, which is defined as,

$$\boldsymbol{\tau} = \boldsymbol{J}_{\boldsymbol{m}}^{\boldsymbol{T}} \boldsymbol{F}_{\boldsymbol{m}}. \tag{7.1}$$

The vector $\boldsymbol{F_m} \in \mathbb{R}^6$ is the virtual control force applied at the end-effector, which can be modelled as a PD (Proportional Derivative) behaviour, defined as,

$$\boldsymbol{F_m} = \boldsymbol{K_{pm}} \Delta \boldsymbol{x_m} - \boldsymbol{K_{dm}} \dot{\boldsymbol{x}}_{\boldsymbol{m}}, \tag{7.2}$$

where $\boldsymbol{K_{pm}}$ and $\boldsymbol{K_{dm}} \in \mathbb{R}^{6\times6}$ are positive definite matrices, representing stiffness and damping gains of the manipulator controller. The vector $\Delta \boldsymbol{x_m}$ is the Cartesian error of the end-effector defined in (6.8) and $\dot{\boldsymbol{x}}_{\boldsymbol{m}}$ is the velocity of the manipulator end-effector expressed in (6.5).

The controller for the satellite-base is described by the following control law,

$$\boldsymbol{F_b} = \boldsymbol{K_{pb}} \Delta \boldsymbol{x_b} - \boldsymbol{K_{db}} \dot{\boldsymbol{x}}_{\boldsymbol{b}} + \boldsymbol{J}_{\boldsymbol{b}}^{\boldsymbol{T}} \boldsymbol{F_m}, \tag{7.3}$$

where $\boldsymbol{K_{pb}}$ and $\boldsymbol{K_{db}} \in \mathbb{R}^{6\times6}$ are positive definite matrices, representing stiffness and damping gains of the base controller. The vector $\Delta \boldsymbol{x_b}$ is the Cartesian error of the base defined in (6.9) and $\dot{\boldsymbol{x}}_{\boldsymbol{b}}$ is the velocity vector of the base. The term $\boldsymbol{J}_{\boldsymbol{b}}^{\boldsymbol{T}} \boldsymbol{F_m}$ in (7.3) is a coupling term between manipulator and base, which is needed to the controller for keeping stability properties, as will be proved in the coming proposition.

In case of a non-redundant manipulator which operates in a workspace free of singularity (i.e. non-singular $\boldsymbol{J_m}$), it can be proven that the velocity error of the base and manipulator and the respective position errors converge to zero. Therefore, the following vector is an equilibrium point:

$$[\dot{\boldsymbol{x}}_{\boldsymbol{b}}, \dot{\boldsymbol{x}}_{\boldsymbol{m}}, \Delta \boldsymbol{x}_{\boldsymbol{b}}, \Delta \boldsymbol{x}_{\boldsymbol{m}}] = \boldsymbol{0} \tag{7.4}$$

and the system is asymptotically stable using the following energetic argument.

Proposition 3 *The equilibrium point* (7.4) *for the free-flying robot dynamics expressed in* (6.1) *is asymptotically stable using the control laws given in* (7.2) *and* (7.3).

Proof *Considering the following candidate Lyapunov function V, as the total positive definite energy of the system,*

$$V = \tfrac{1}{2} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T & \dot{\boldsymbol{q}}^T \end{bmatrix} \underbrace{\begin{bmatrix} \boldsymbol{H_b} & \boldsymbol{H_{bm}} \\ \boldsymbol{H_{bm}^T} & \boldsymbol{H_m} \end{bmatrix}}_{\boldsymbol{H}} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{q}} \end{bmatrix} + \tfrac{1}{2} \Delta \boldsymbol{x_m}^T \boldsymbol{K_{pm}} \Delta \boldsymbol{x_m} + \tfrac{1}{2} \Delta \boldsymbol{x_b}^T \boldsymbol{K_{pb}} \Delta \boldsymbol{x_b}, \tag{7.5}$$

when computing the derivative $\dot{V}$, the well-known passivity property of the Euler Lagrange systems expressed as follows is considered,

$$\frac{1}{2} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T & \dot{\boldsymbol{q}}^T \end{bmatrix} \dot{\boldsymbol{H}} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{q}} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T & \dot{\boldsymbol{q}}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{c_b} \\ \boldsymbol{c_m} \end{bmatrix} = 0. \tag{7.6}$$

Therefore, it is possible to obtain:

$$\dot{V} = \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{F_b} + \dot{\boldsymbol{q}}^T \boldsymbol{\tau} - \dot{\boldsymbol{x}}_{\boldsymbol{m}}^T \boldsymbol{K_{pm}} \Delta \boldsymbol{x_m} - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{K_{pb}} \Delta \boldsymbol{x_b}. \tag{7.7}$$

Considering, the transpose of the end-effector velocity vector expressed as follows,

$$\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T = \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{J_b}^T + \dot{\boldsymbol{q}}^T \boldsymbol{J_m}^T \tag{7.8}$$

and substituting the control law for the base (i.e. $\boldsymbol{F_b}$ from (7.3)*) and for the manipulator (i.e. $\boldsymbol{\tau}$ from* (7.1)-(7.2)*) in* (7.7)*, the following balance results:*

$$\dot{V} = -\dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{K_{db}} \dot{\boldsymbol{x}}_{\boldsymbol{b}} + \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{J_b}^T \boldsymbol{F_m} - \dot{\boldsymbol{q}}^T \boldsymbol{J_m}^T \boldsymbol{K_{dm}} \dot{\boldsymbol{x}}_{\boldsymbol{m}} - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{J_b}^T \boldsymbol{K_{pm}} \Delta \boldsymbol{x_m}. \tag{7.9}$$

In (7.9) *the value $\dot{\boldsymbol{q}}^T \boldsymbol{J_m}^T$ is equal to $\dot{\boldsymbol{q}}^T \boldsymbol{J_m}^T = \dot{\boldsymbol{x}}_{\boldsymbol{m}}^T - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{J_b}^T$ and by substitution, the derivative of Lyapunov function results:*

$$\dot{V} = -\dot{\boldsymbol{x}}_{\boldsymbol{b}}^T \boldsymbol{K_{db}} \dot{\boldsymbol{x}}_{\boldsymbol{b}} - \dot{\boldsymbol{x}}_{\boldsymbol{m}}^T \boldsymbol{K_{dm}} \dot{\boldsymbol{x}}_{\boldsymbol{m}} \leq 0. \tag{7.10}$$

Using standard LaSalle arguments (see Lemma C.2.1), the statement of the proposition is proven.

Therefore, in continuous-time the system behaves in a dissipative way and stability can be achieved by using the control laws (7.2) and (7.3). In particular, the term $\boldsymbol{J}_{\boldsymbol{b}}^{T}\boldsymbol{F}_{\boldsymbol{m}}$ in (7.3) is required to the base control to achieve (7.10).

A simulation example for the controllers running in continuous-time is performed considering the following.

Example 4 *Considers a satellite-base with a mass of 150 kg and inertia* $H_{bx} = 38kgm^2$, $H_{by} = 20kgm^2$, $H_{bz} = 23kgm^2$. *Considers the manipulator whose mass and inertia parameters are reported in Table 6.2. The manipulator has an initial error of (0, 0, -0.1) m in position and (0, 20, 0) deg in orientation and the base an initial error of (0 0 0.1) m and (20, 0, 0) deg.*

The regulation error dynamics for the manipulator end-effector and the satellite is shown in Fig. 7.1. As can be seen in the plots the error for the base (top) and the manipulator (bottom) converge in continuous time. However, when the difference between the rate of the controller is high, the stability is not guaranteed as will be discussed in the coming subsection.

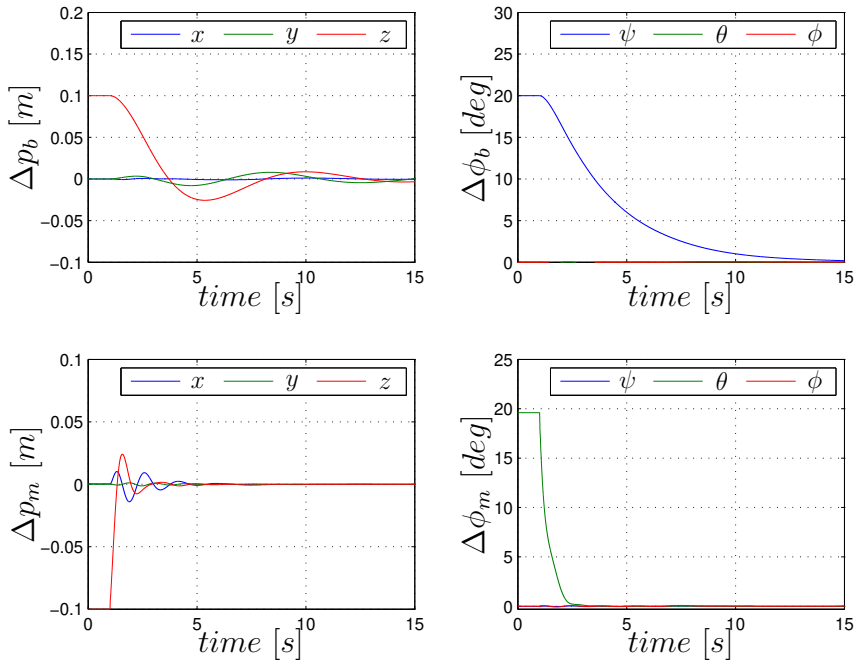


Figure 7.1: Stable system with continuous-time controllers: Error of the base in position and orientation (top) and error of the manipulator in position and orientation (bottom). Orientation error is represented by ψ, θ, ϕ (yaw, pitch and roll) angles.

7.2.1 Problem Statement: Regulation with Multi-rate

This section shows how the stability properties are broken when the control laws for the base and the manipulator are computed in discrete-time and run at different sampling rates. The sampling time for the manipulator control and the base will be

indicated with T_m and T_b, respectively. The sampling time of the slow rate controller can be expressed as an integer multiple, n, of the rate of the fast controller, i.e. $T_b = nT_m$. Therefore, it will results that a discrete time $k = k_m T_m = k_b T_b$, where k_m and k_b are the discrete steps in each of the controllers. Then, the control law expressed in (7.2) and (7.3) can be rewritten in a discrete form considering the respective sampling time, as follows:

$$\boldsymbol{F_m}(\boldsymbol{k_m}) = \boldsymbol{K_{pm}}\Delta\boldsymbol{x_m}(\boldsymbol{k_m}) - \boldsymbol{K_{dm}}\dot{\boldsymbol{x}}_{\boldsymbol{m}}(\boldsymbol{k_m}), \tag{7.11}$$

$$\boldsymbol{F_b}(\boldsymbol{k_b}) = \boldsymbol{K_{pb}}\Delta\boldsymbol{x_b}(\boldsymbol{k_b}) - \boldsymbol{K_{db}}\dot{\boldsymbol{x}}_{\boldsymbol{b}}(\boldsymbol{k_b}) + \boldsymbol{J_b^T}(\boldsymbol{k_b})\boldsymbol{F_m}(\boldsymbol{k_b}). \tag{7.12}$$

The discrete nature of the controllers leads to creation of virtual energy, which might lead to instability. From an analytical point of view, it can be easily seen when (7.11) and (7.12) are substituted in the total Lyapunov function in (7.7), the energy balance will not hold anymore. This is due to the fact that the discrete values computed at k_b or k_m are different with respect to the continuous-time values. In particular, these discrepancies result to be large when the sampling time is larger.

The following numerical example clarifies this aspect. The simulation considers the discrete laws (7.11) and (7.12), with the same conditions of Example 4 simulated for the continuous case. The controllers run with a sampling time of $T_m = 0.001s$ and $T_b = 0.5s$, which are common control frequencies used in the free-flying scenario, see [TEDS+17]. The results in Fig. 7.2 shown that the error in position Δp and in orientation $\Delta\phi$ of both controllers diverge. It is worth comparing Fig. 7.1

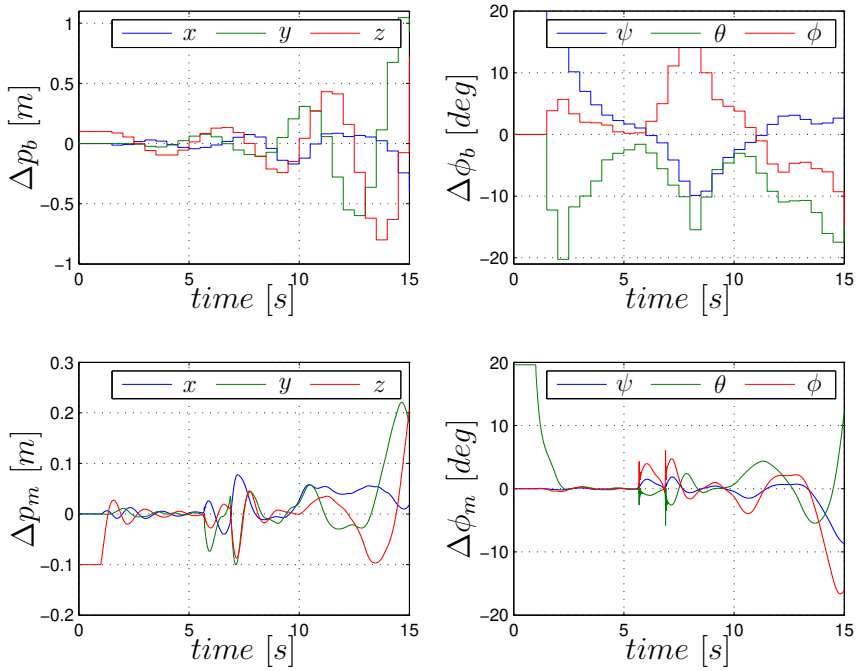


Figure 7.2: Unstable system with different-rate controllers: Error of the base in position and orientation (top) and error of the end-effector in position and orientation (bottom).

(continuous-time control) and Fig. 7.2 (different rate control) to see the differences while using controllers at different rates.

The presence of different-rate controllers destroys the dissipative effect discussed in *Proposition 3* and, therefore, some energy is produced. This implies that the passivity of the system is lost. Indeed, the interaction between two controllers running at different rates is equivalent of having delays and packet losses between the systems. Delays and packet losses in the communication channel introduce energy in the system and make the system active [AS89]. In the next section, an energy observer will be designed in order to monitor the produced energy, which is dissipated using a passivity controller.

7.3 Energy-based Coordinated Control for Regulation

This section introduces the network representation for the regulation control of the free-flying space robot in order to perform an energy analysis. In particular, the energy produced by the interaction of the controllers will be identified and it is passivated using a passivity controller.

The methods exploits the time domain passivity approach because it provides flexibility and it does not depend on system dynamics modeling [HR01]. The reader can refer to Sec. 2.7 where this approach was extensively discussed. For completeness the general passivity condition is reported below

$$\sum_{k=0}^{m}(\boldsymbol{F}(k)^T\boldsymbol{v}(k)T) + E(0) \geq 0, \tag{7.13}$$

where $(\boldsymbol{F}, \boldsymbol{v})$ and $E(0)$ are the power correlated variable and the initial energy storage of the system respectively. If condition (7.13) holds, the system is passive. The extra energy generated in the port that violates the passivity condition is dissipated with a time-varying damper, the Passivity Controller (PC). This is done in order to ensure that the system is an interconnection of passive ports. This condition will be used for the design of the controller.

7.3.1 Passive Regulation Control

The network representation of the system is shown in Fig. 7.3, where S denotes the dynamic system represented by the left hand side of the free-flying dynamics equation expressed in (6.1), C_b is the controller at the base reported in (7.12) and C_m is the controller of the manipulator in (7.11).

The controller C_b is represented by electrical elements with impedance $\boldsymbol{Z_{cb}}$ and a dependent voltage source which represents the coupling term of the manipulator, i.e. $\boldsymbol{J_b^T F_m}$. The control rate of the manipulator C_m is usually assumed to be high (1000 Hz) and, therefore, its behaviour is similar to that of an equivalent system controlled by a spring-damper controller (7.2) in continuous time. Therefore, the

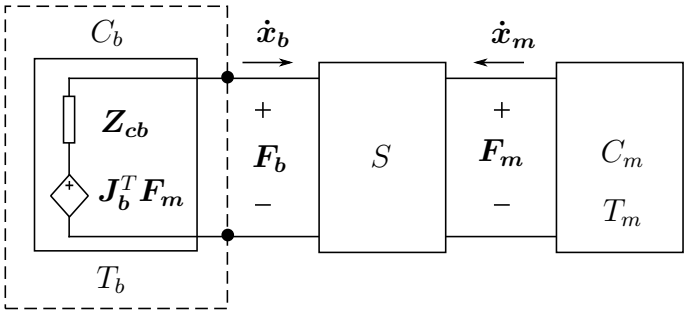


Figure 7.3: Regulation case: network representation for the satellite-base control and manipulator.

small amount of energy which could introduce, can be neglected. However, the main source of energy is given by the low rate control which acts on the base (dashed box in Fig. 7.3). The possible energy injections is observed through the port $(\boldsymbol{F_b}, \dot{\boldsymbol{x}}_{\boldsymbol{b}})$.

An energy observer (PO) is designed to monitor the energy flowing in and out of this port using (7.13) with power variables $(\boldsymbol{F_b}, \dot{\boldsymbol{x}}_{\boldsymbol{b}})$. The observer runs with the faster rate, T_m, in order to obtain a greater accuracy. If the controller at the base is rendered to be passive at this port, then the overall controlled system will be an interconnection of passive systems and, consequently, will be passive [SSF07]. Then all the regenerative and destabilising effects would be compensated.

The PO-PC implementation is applied for each dof separately and it can be mathematically shown that if passivity can be guaranteed for all the dof separately, the overall multi-dof system is also passive. Therefore, for a n-dof system with initial energy storage $E(0) = 0$, the passivity condition for each dof separately leads to:

$$\sum_{k=0}^{m} (\boldsymbol{F}(k)^T \boldsymbol{v(k)} T) = \sum_{k=0}^{m} \sum_{i=1}^{n} F_i(k) v_i(k) T, \tag{7.14}$$

which proves that (7.13) can be split into a sum of the n-dof components. Therefore, if the passivity condition holds for each component, then the overall system would be passive. The bold notation will be then omitted in the coming analysis because the components of the vectors are considered.

The flow of energy is monitored with an energy observer, defined as follows

$$E_{obsb}(k_m) = E_{obsb}(k_m - 1) + F_b(k_m)\dot{x}_b(k_m)T_m + \beta(k_m - 1)\dot{x}_b(k_m - 1)^2 T_m, \tag{7.15}$$

where the satellite-base velocity $\dot{x}_b$ needs to be available at sampling rate T_m (eg., using a high-rate position sensor or a velocity observer as developed in [MDSGO19]).

Although the controller of the base runs at low rate T_b, the energy observer is implemented considering the high sampling rate T_m, therefore, the energy flowing through the port is updated at each T_m. Then, it is considered that between two values of F_b, the observer holds the last received value. Thus, the second term in (7.15) computes the energy flow of the slow-rate controller and the last term

considers the effect of the PC which is function of a variable damper β defined as:

$$\beta(k_m) = \begin{cases} -\frac{E_{obsb}(k_m)}{\dot{x}_b(k_m)^2 T_m} & E_{obsb}(k_m) < -E(0) \\ 0 & else. \end{cases} \tag{7.16}$$

The impedance correction due to the PC is given by the following quantity:

$$F_{pc}(k_m) = -\beta(k_m)\dot{x}_b(k_m). \tag{7.17}$$

It has to be noted that although the calculation for the PO and PC are implemented at fast rate with a sampling period T_m, the calculated values of the passivity controller $F_{pc}(k_m)$ are sent to the base controller which runs at T_b and it will modify the output force $F_b(k_b)$. Hence, when passivity condition is violated (e.g. $E_{obsb}(k_m) < -E(0)$), the force commanded to the base will be corrected as follows:

$$F_b'(k_b) = F_b(k_b) + F_{pc}(k_b). \tag{7.18}$$

As a result, the energy will be restored and the observer will be $E_{obs}(k) \geq -E(0)$ making the network passive. The network representation modified with the passivity controller can be seen in Fig. 7.4. The benefit of the method is that the control forces depend only on the correlated variables at the port $(\boldsymbol{F_b}, \dot{\boldsymbol{x}}_{\boldsymbol{b}})$.

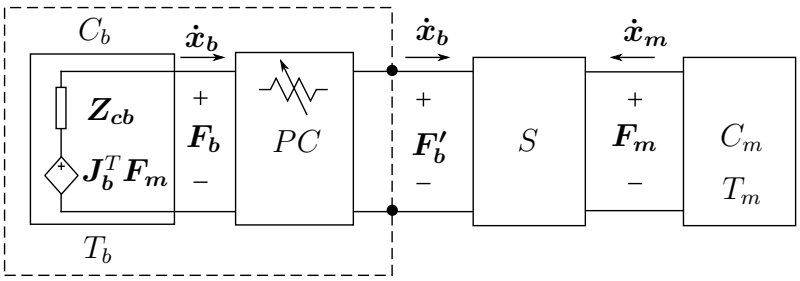


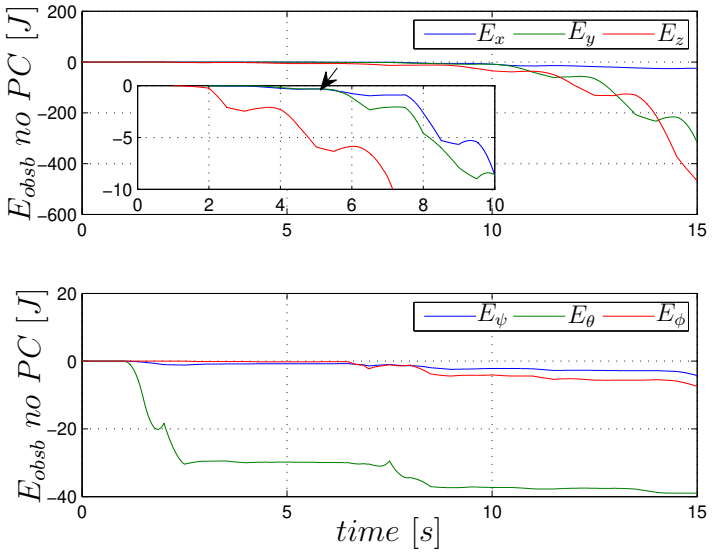
Figure 7.4: Regulation case: network representation scheme modified with PC.

7.3.2 Results

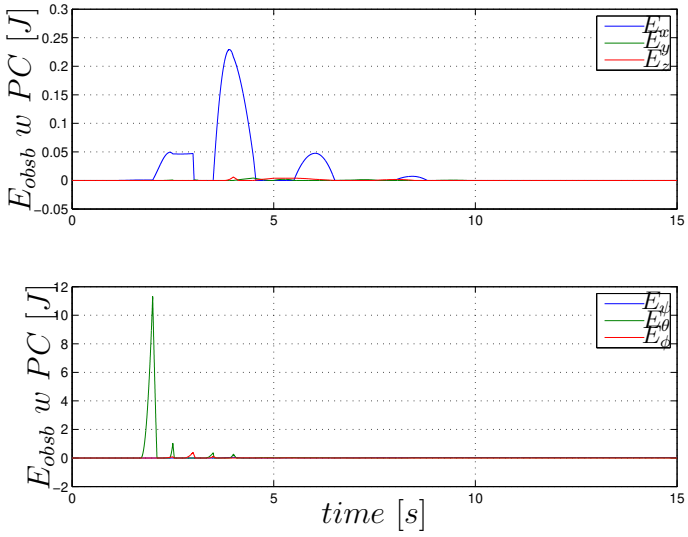
This section shows simulation results performed with the passive coordinated control for the base and the manipulator. The simulation considers the system described in Example 4 where the sampling time for the manipulator is $T_m = 0.001\ s$ and for the base $T_b = 0.5\ s$. The problem described in Sec. 7.2.1 is here resolved by applying the proposed method. Indeed, the destabilising effects described in Fig. 7.2 are due to the active observed energy which is shown in Fig. 7.5a for the linear and angular components. The negative trend of energy shows the activity of the system as the violation of (7.13) confirms.

The passivity control is then applied and it will provide the variable force-torque as shown in Fig. 7.6a. This will lead to a passive system as shown by the energy plots in Fig. 7.5b. The positive trend indicates that no active energy is pushed into the

system. Therefore the error for both systems, i.e. manipulator and base, converges (see Fig. 7.6b) and the system results to be passive. It is worth comparing Fig. 7.2 (before applying the method) with Fig. 7.6b (proposed method). The results prove the validity of the method.

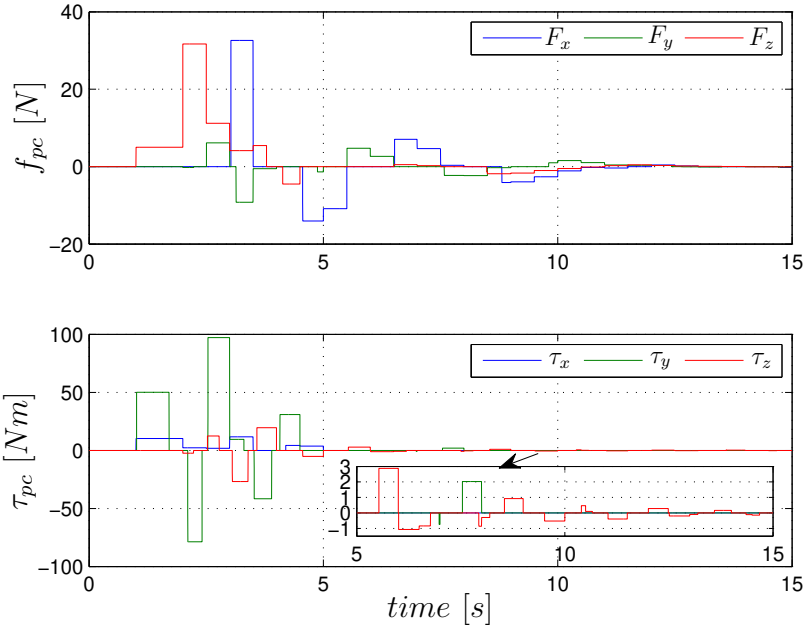


(a) Active energy is detected by the negative trend. Energy without PC for translation (top) and rotation (bottom).

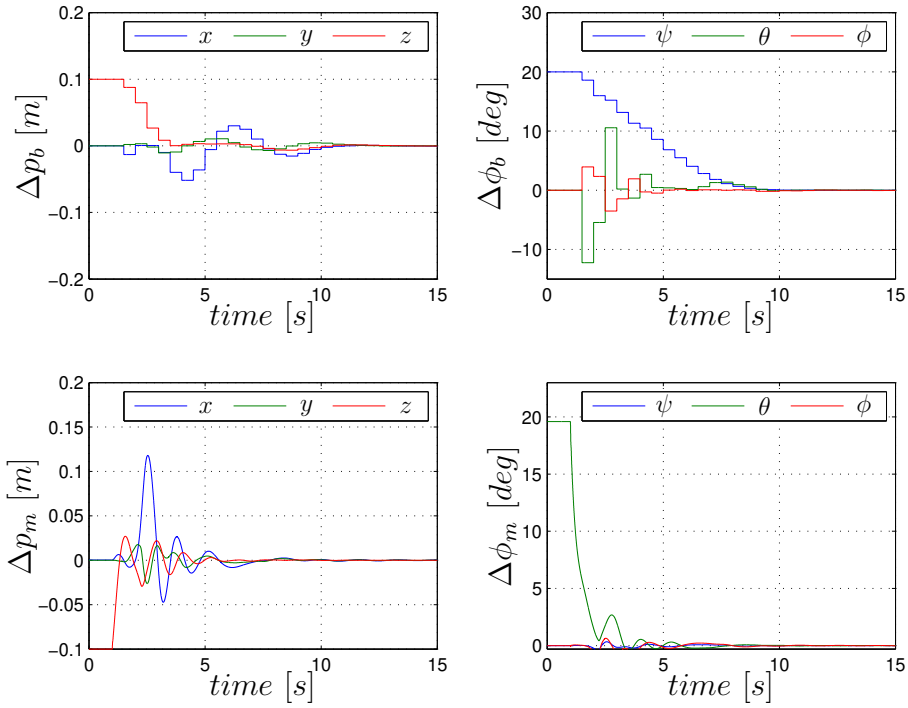


(b) Positive energy indicates the passivity of the system. Energy with PC for traslation (top) and rotation (bottom).

Figure 7.5: Energy observer without PC (top) and with PC (bottom).



(a) Force and torque of the passivity controller $F_{pc} = (f_{pc}, \tau_{pc})$.



(b) Stable system with the proposed method. Top row: base error in position and orientation, Bottom row: end-effector error in position and orientation.

Figure 7.6: Force and torque of the passivity controller and error of the manipulator and base.

7.4 Multi-rate Tracking Control for a Free Flying Robot

In Sec. 7.2 the regulation control problem for a free-flying space robot was resolved for both the base and end-effector. The effects due to the difference of the sampling rate have shown to produce activity and a passivity controller was designed to restore the overall energy. However, when the end-effector of the manipulator has to perform a tracking task, a simple regulation control (as the one presented in Sec. 7.2) might not be sufficient. Trajectory tracking is particularly required for a space robot in order to avoid singularities and/or minimize a designed cost function. Therefore, the tracking of a trajectory provided by a motion planner is a common approach required in space missions [LMO+18].

In this section the tracking problem for the end-effector of a free-flying robot and the regulation of its base will be tackled. To achieve stability of the controllers, the controller at the base will be function of the end-effector desired trajectory. This is due to the coupled dynamics between base and end-effector motions and must be taken into account in the design of the control laws. The controllers are presented firstly in continuous-time with stability proofs, later, the effects of the controllers running at different sampling rates are discussed.

In order to facilitate the analysis, the free-flying robot dynamics expressed in (6.1) will be transformed in Cartesian space with a proper change of coordinates from $\boldsymbol{\xi_q}$ to $\boldsymbol{\xi_m}$, where:

$$\boldsymbol{\xi_q} = \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{q}} \end{bmatrix}, \boldsymbol{\xi_m} = \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{m}} \end{bmatrix}. \tag{7.19}$$

The relationship between the new coordinate $\boldsymbol{\xi_m}$ with $\boldsymbol{\xi_q}$ are given by the Jacobian $\tilde{\boldsymbol{J}}$ defined as follows:

$$\boldsymbol{\xi_m} = \underbrace{\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{J_b} & \boldsymbol{J_m} \end{bmatrix}}_{\tilde{\boldsymbol{J}}} \boldsymbol{\xi_q}. \tag{7.20}$$

In case of a not redundant manipulator operating in a singularity-free configurations e.g. achieved by a motion planner [LMO+18], the Jacobian $\tilde{\boldsymbol{J}}$ can be inverted as follows:

$$\tilde{\boldsymbol{J}}^{-\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{J_b^T J_m^{-T}} \\ \boldsymbol{0} & \boldsymbol{J_m^{-T}} \end{bmatrix}. \tag{7.21}$$

Therefore, using the relationship in (7.20), the general dynamics equation (6.1) can be re-written in Cartesian space as:

$$\underbrace{\begin{bmatrix} \boldsymbol{M_b} & \boldsymbol{M_{bm}} \\ \boldsymbol{M_{bm}^T} & \boldsymbol{M_m} \end{bmatrix}}_{\boldsymbol{M}} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \ddot{\boldsymbol{x}}_{\boldsymbol{m}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{\Gamma_b} & \boldsymbol{\Gamma_{bm}} \\ \boldsymbol{\Gamma_{mb}} & \boldsymbol{\Gamma_m} \end{bmatrix}}_{\boldsymbol{\Gamma}} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{b}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{m}} \end{bmatrix} = \tilde{\boldsymbol{J}}^{-\boldsymbol{T}} \begin{bmatrix} \boldsymbol{F_b} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_b} - \boldsymbol{J_b^T F_m} \\ \boldsymbol{F_m} \end{bmatrix}, \tag{7.22}$$

where $\boldsymbol{M} \in \mathbb{R}^{12\times12}$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{12\times12}$ are the inertia and Coriolis/centrifugal matrices in the new coordinate system $\boldsymbol{\xi_m}$. $\boldsymbol{F}_m \in \mathbb{R}^6$ is the Cartesian wrench at the end-effector defined as $\boldsymbol{F_m} = \boldsymbol{J_m^{-T}\tau}$. The system in (7.22) results to be a fully actuated Euler-Lagrange system, see e.g. [SL91] and therefore it is passive with respect to the generalized force-velocity pair, which are $((\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}}, \dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}})^T, ((\boldsymbol{F_b} - \boldsymbol{J_b^T F_m})^T, \boldsymbol{F}_{\boldsymbol{m}}^T)^T)$.

Coordinated tracking control in continuous time

The tracking control for the end-effector is designed for the system (7.22), firstly in continuous-time. For a given trajectory composed of a desired acceleration $\ddot{\boldsymbol{x}}_{\boldsymbol{md}}(\boldsymbol{t}) \in \mathbb{R}^6$, a desired velocity $\dot{\boldsymbol{x}}_{\boldsymbol{md}}(\boldsymbol{t}) \in \mathbb{R}^6$ and a desired position $\boldsymbol{x_{md}}(\boldsymbol{t}) \in \mathbb{R}^6$ expressed in the inertial frame, the tracking law for the manipulator can be defined as follows:

$$\boldsymbol{F_m} = \boldsymbol{M_m}\ddot{\boldsymbol{x}}_{\boldsymbol{md}} + \boldsymbol{\Gamma_m}\dot{\boldsymbol{x}}_{\boldsymbol{md}} - \boldsymbol{K_{pm}}\Delta\boldsymbol{x_m} - \boldsymbol{K_{dm}}\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}, \tag{7.23}$$

where $\boldsymbol{K_{pm}}, \boldsymbol{K_{dm}} \in \mathbb{R}^{6\times6}$ are stiffness and damping matrices, respectively. The vector $\Delta\boldsymbol{x_m} = (\boldsymbol{x_m} - \boldsymbol{x_{md}}) \in \mathbb{R}^6$ is the error between the measured and desired poses (position and orientation) of the manipulator end-effector. Similarly, the vector $\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}} = (\dot{\boldsymbol{x}}_{\boldsymbol{m}} - \dot{\boldsymbol{x}}_{\boldsymbol{md}})$ is the error between the measured and desired end-effector velocities. The controller in (7.23) can be projected into the joints space to have a torque command to the manipulator, which is $\boldsymbol{\tau} = \boldsymbol{J}_{\boldsymbol{m}}^T\boldsymbol{F_m}$.

The regulation control at the base is designed as follows:

$$\boldsymbol{F_b} = \boldsymbol{M_{bm}}\ddot{\boldsymbol{x}}_{\boldsymbol{md}} + \boldsymbol{\Gamma_{bm}}\dot{\boldsymbol{x}}_{\boldsymbol{md}} - \boldsymbol{K_{pb}}\Delta\boldsymbol{x_b} - \boldsymbol{K_{db}}\dot{\boldsymbol{x}}_{\boldsymbol{b}} + \boldsymbol{J_b^T F_m}. \tag{7.24}$$

The matrix $\boldsymbol{K_{pb}}, \boldsymbol{K_{db}} \in \mathbb{R}^{6\times6}$ are stiffness and damping matrices of the regulation control of the base, respectively. The vector $\Delta\boldsymbol{x_b} = (\boldsymbol{x_b} - \boldsymbol{x_{bd}}) \in \mathbb{R}^6$ is the error between the measured and the desired pose of the base.

As can be seen from (7.24), although the controller of the base resolves a regulation problem, the control law is a function of the desired tracking end-effector velocity and acceleration. This is due to the inertial coupling between the base and end-effector tracking motions. Therefore, the regulation control law proposed in (7.3) for the base controller is not suitable when the end-effector performs a tracking task. Otherwise the system might results to be unstable during the tracking.

In the following, the closed-loop stability of the tracking controller and regulation of the base is proved. The closed-loop behaviour of the system resembles the one of a PD+ controller and its asymptotic stability can be proved by using a strict Lyapunov function, see for example [SK97, *Proposition 4*]. However, the strict Lyapunov function (as reported in [SK97]) includes extra cross terms bilinear in position and velocity, which affect the meaning of total energy. The connection with the passivity control action, which is part of the next section, will be based on mechanical energy observed at the port. Therefore, an alternative Lyapunov function which resemble the total energy for the system in (7.22) with the controller (7.23) and (7.24) must be chosen to prove stability.

Then, defining

$$[\dot{\boldsymbol{x}}_{\boldsymbol{b}}, \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}, \Delta\boldsymbol{x_b}, \Delta\boldsymbol{x_m}] = \boldsymbol{0} \tag{7.25}$$

an equilibrium point of (7.22) using (7.23) and (7.24), the following proposition holds true.

Proposition 4 *Considers the system* (7.22)*, with the tracking control law* (7.23) *for the end-effector and the regulation control law* (7.24) *for the base. Then, the equilibrium point in* (7.25) *is asymptotically stable.*

Proof *Considering the following candidate Lyapunov function V, as the total positive definite energy of the system,*

$$V = \tfrac{1}{2}\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}\end{bmatrix}\boldsymbol{M}\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}\end{bmatrix} + \tfrac{1}{2}\begin{bmatrix}\Delta\boldsymbol{x}_{\boldsymbol{b}}^{T} & \Delta\boldsymbol{x}_{\boldsymbol{m}}^{T}\end{bmatrix}\underbrace{\begin{bmatrix}\boldsymbol{K_{pb}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K_{Pm}}\end{bmatrix}}_{\boldsymbol{K}}\underbrace{\begin{bmatrix}\Delta\boldsymbol{x}_{\boldsymbol{b}}\\ \Delta\boldsymbol{x}_{\boldsymbol{m}}\end{bmatrix}}_{\Delta\boldsymbol{x}}, \tag{7.26}$$

Since $\boldsymbol{M}$ and $\boldsymbol{K}$ are positive definite matrices, the following bounds given holds,

$$\begin{aligned} V &\geq \frac{1}{2}\Big(\underline{\sigma}(\boldsymbol{M})||\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{T} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{T}\end{bmatrix}^{T}||^2 + \underline{\sigma}(\boldsymbol{K})||\Delta\boldsymbol{x}||^2\Big)\\ V &\leq \frac{1}{2}\Big(\overline{\sigma}(\boldsymbol{M})||\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{T} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{T}\end{bmatrix}^{T}||^2 + \overline{\sigma}(\boldsymbol{K})||\Delta\boldsymbol{x}||^2\Big)\end{aligned} \tag{7.27}$$

where $\underline{\sigma}(*)$ and $\overline{\sigma}(*)$ are the lowest and highest values of the symmetric matrices $(*)$, respectively. Computing the time-derivative of (7.26)*, it is possible to obtain the following balance:*

$$\dot{V} = \begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}\end{bmatrix}\left[\boldsymbol{M}\begin{bmatrix}\ddot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \ddot{\boldsymbol{x}}_{\boldsymbol{m}} - \ddot{\boldsymbol{x}}_{\boldsymbol{md}}\end{bmatrix} + \tfrac{1}{2}\dot{\boldsymbol{M}}\underbrace{\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}\end{bmatrix}}_{\Delta\dot{\boldsymbol{x}}} + \boldsymbol{K}\begin{bmatrix}\Delta\boldsymbol{x}_{\boldsymbol{b}}\\ \Delta\boldsymbol{x}_{\boldsymbol{m}}\end{bmatrix}\right]. \tag{7.28}$$

Substituting the values $\boldsymbol{M}[\ddot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}}\ \ddot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}]^T$ from (7.22)*, it is possible to get:*

$$\begin{aligned}\dot{V} = \begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}\end{bmatrix}\Bigg[-\boldsymbol{\Gamma}\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \dot{\boldsymbol{x}}_{\boldsymbol{m}}\end{bmatrix} + \begin{bmatrix}\boldsymbol{F_b} - \boldsymbol{J_m^T F_m}\\ \boldsymbol{F_m}\end{bmatrix} - \boldsymbol{M}\begin{bmatrix}\boldsymbol{0}\\ \ddot{\boldsymbol{x}}_{\boldsymbol{md}}\end{bmatrix} + \frac{1}{2}\dot{\boldsymbol{M}}\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}\end{bmatrix} +\\ \boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{0}\\ \dot{\boldsymbol{x}}_{\boldsymbol{md}}\end{bmatrix} - \boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{0}\\ \dot{\boldsymbol{x}}_{\boldsymbol{md}}\end{bmatrix} + \boldsymbol{K}\begin{bmatrix}\Delta\boldsymbol{x}_{\boldsymbol{b}}\\ \Delta\boldsymbol{x}_{\boldsymbol{m}}\end{bmatrix}\Bigg]\end{aligned} \tag{7.29}$$

and considering the skew-symmetric property of the Euler-Lagrange system, where:

$$\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}} & \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}\end{bmatrix}\frac{1}{2}(\dot{\boldsymbol{M}} - 2\boldsymbol{\Gamma})\begin{bmatrix}\dot{\boldsymbol{x}}_{\boldsymbol{b}}\\ \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}\end{bmatrix} = 0, \tag{7.30}$$

equation (7.29) *can be simplified and rewritten as:*

$$\begin{aligned}\dot{V} = \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{F_b} - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{J_b^T F_m} + \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T\boldsymbol{F_m} - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{M_{bm}}\ddot{\boldsymbol{x}}_{\boldsymbol{md}} - \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T\boldsymbol{M_m}\ddot{\boldsymbol{x}}_{\boldsymbol{md}} - \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{\Gamma_{bm}}\dot{\boldsymbol{x}}_{\boldsymbol{md}}\\ - \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T\boldsymbol{\Gamma_m}\dot{\boldsymbol{x}}_{\boldsymbol{md}} + \dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{K_{Pb}}\Delta\boldsymbol{x_b} + \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T\boldsymbol{K_{Pm}}\Delta\boldsymbol{x_m}.\end{aligned} \tag{7.31}$$

Substituting the control laws (7.23)*-*(7.24) *in* (7.31)*, the time-derivative of Lyapunov function results to be:*

$$\dot{V} = -\dot{\boldsymbol{x}}_{\boldsymbol{b}}^T\boldsymbol{K_{db}}\dot{\boldsymbol{x}}_{\boldsymbol{b}} - \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}}^T\boldsymbol{K_{dm}}\Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}} \leq 0. \tag{7.32}$$

As a result of negative semi-definiteness of the time-derivative Lyapunov function in (7.32)*, stability is proved. From* (7.32)*, it is standard to invoke Barbalat's Lemma (C.2.2) for non-autonomous systems to conclude that* $\dot{\boldsymbol{x}}_{\boldsymbol{b}}, \Delta\dot{\boldsymbol{x}}_{\boldsymbol{m}} \to 0$ *for a trajectory which is bounded in* $\ddot{\boldsymbol{x}}_{\boldsymbol{md}}, \dot{\boldsymbol{x}}_{\boldsymbol{md}}, \boldsymbol{x}_{\boldsymbol{md}}$*, see [LPPT05]. Furthermore, for establishing asymptotic stability, the Matrosov theorem, which has been exploited in literature for tracking problem can be used, see [PP88]. Usually an auxiliary function of the states,* W*, is chosen such that* $\dot{W} \neq 0$ *when* $\dot{V} = 0$*. In particular, the auxiliary function can be chosen as follows:*

$$W = \Delta\boldsymbol{x}^T \boldsymbol{M} \Delta\dot{\boldsymbol{x}} \tag{7.33}$$

Therefore in the considered case, by computing $\dot{W}$ *and substituting the values* $\boldsymbol{M}[\ddot{\boldsymbol{x}}_{\boldsymbol{b}}^{\boldsymbol{T}} \ \ddot{\boldsymbol{x}}_{\boldsymbol{m}}^{\boldsymbol{T}}]^{\boldsymbol{T}}$ *from* (7.22) *with the control laws* (7.23) *and* (7.24)*, it results*

$$\dot{W} = \Delta\dot{\boldsymbol{x}}^T \boldsymbol{M} \Delta\dot{\boldsymbol{x}} + \Delta\boldsymbol{x}^T \Big(- \boldsymbol{K}\Delta\boldsymbol{x} - \boldsymbol{K_D}\Delta\dot{\boldsymbol{x}} + (\dot{\boldsymbol{M}} - \boldsymbol{\Gamma})\Delta\dot{\boldsymbol{x}} \Big) \tag{7.34}$$

where $\boldsymbol{K_D}$ *is the total damping matrix of the base and manipulator expressed in a compact form as:*

$$\boldsymbol{K_D} = \begin{bmatrix} \boldsymbol{K_{db}} & \boldsymbol{0}_{6\times6} \\ \boldsymbol{0}_{6\times6} & \boldsymbol{K_{dm}} \end{bmatrix}. \tag{7.35}$$

Considering that the trajectory is bounded (for example, enforced by the motion planner) and the states are bounded (from Barbalat's lemma applied in (7.32)*), it can be concluded that* $\dot{W}$ *is bounded. In particular, it can be seen that when* $\dot{V} \to 0$*, it means that* $\Delta\dot{\boldsymbol{x}} \to 0$*, and therefore,* $\dot{W}$ *results to be:*

$$\dot{W} = -\Delta\boldsymbol{x}^T \boldsymbol{K} \Delta\boldsymbol{x}, \tag{7.36}$$

which is sign-definite (negative) and non-zero for non-zero error $||\Delta\boldsymbol{x}||$*. Therefore, with this result, the conditions of Matrosov's theorem (C.2.3) are satisfied to establish asymptotic stability of the state in* (7.25)*.*

From a physical point of view, the sign-definiteness of $\dot{W}$ found in the proof and the fact that $W = 0$, on the set of $\dot{V} = 0$, implies that the system trajectory can not remain trapped in the set ($\Delta\dot{\boldsymbol{x}} = 0$, $\Delta\boldsymbol{x} \in \mathbb{R}^{12}$) unless they go to zero. As validation, a simulation with the control laws in (7.23) and (7.24) is performed in continuous time considering the following,

Example 5 *Considers a satellite with a mass of 150 kg and inertia Inertia* $H_{bx} = 38kgm^2$*,* $H_{by} = 20kgm^2$*,* $H_{bz} = 23kgm^2$*. Considers the manipulator whose mass and inertia parameters are reported in Table 6.2. Considers a desired trajectory commanded to the end-effector as shown in Fig. 7.7 for a position* x *and orientation* θ*.*

The tracking error of the manipulator for the considered Example 5 is shown at the top of Fig. 7.8 and it is expressed in position and orientation as: $\Delta\boldsymbol{x}_{\boldsymbol{m}} = (\Delta\boldsymbol{p}_{\boldsymbol{m}}, \Delta\boldsymbol{\Phi}_{\boldsymbol{m}})$. In the second row of Fig. 7.8, the error $\Delta\boldsymbol{x}_{\boldsymbol{b}} = (\Delta\boldsymbol{p}_{\boldsymbol{b}}, \Delta\boldsymbol{\Phi}_{\boldsymbol{b}})$ in position and orientation of the base is shown. As it can be seen, both the errors converge in continuous time. The tracking error is in the order of 10^{-4} m for the end-effector and the base regulates about the given set-point.

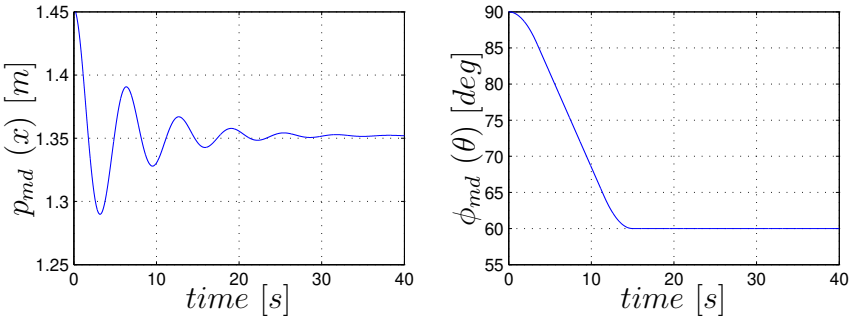


Figure 7.7: Desired trajectory for the end-effector, $x_{md} = [p_{md}^T \quad \Phi_{md}^T]^T$.

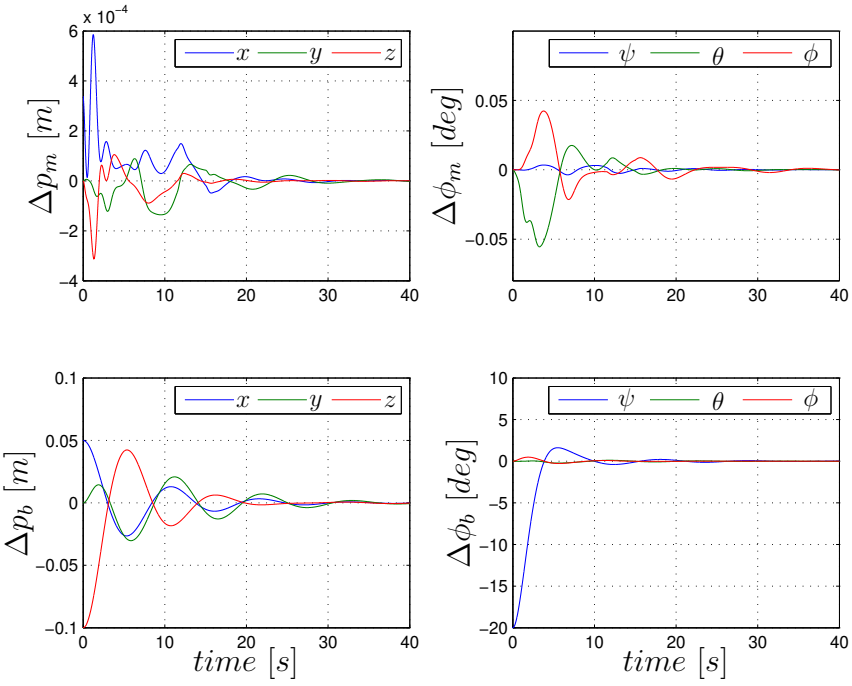


Figure 7.8: Stable system with continuous-time controllers. Tracking error of the end-effector (top) and error of the base (bottom).

7.4.1 Problem Statement: Tracking with Multi-rate

As discussed already for the regulation case, in real space applications the control at the base is usually actuated with a low rate controller with respect to the manipulator control rate [TEDS+17]. This section summarises the control in discrete time which generates instability and this aspect will be recovered in the coming section.

Considering T_m the sampling time of the controller for the manipulator and T_b the sampling time of the controller for the base , where $T_b = nT_m$.
The control law (7.23) and (7.24) can be rewritten accordingly to the sampling time for the discrete case as follows:

$$\begin{aligned} \boldsymbol{F_m(k_m)} = \boldsymbol{M_m(k_m)\ddot{x}_{md}(k_m)} + \boldsymbol{\Gamma_m(k_m)\dot{x}_{md}(k_m)} \\ - \boldsymbol{K_{pm}\Delta x_m(k_m)} - \boldsymbol{K_{dm}\Delta \dot{x}_m(k_m)}, \end{aligned} \tag{7.37}$$

$$\begin{aligned} \boldsymbol{F_b}(\boldsymbol{k_b}) = \boldsymbol{M_{bm}}(\boldsymbol{k_b})\ddot{\boldsymbol{x}}_{\boldsymbol{md}}(\boldsymbol{k_b}) + \boldsymbol{\Gamma_{bm}}(\boldsymbol{k_b})\dot{\boldsymbol{x}}_{\boldsymbol{md}}(\boldsymbol{k_b}) \\ - \boldsymbol{K_{pb}}\Delta\boldsymbol{x_b}(\boldsymbol{k_b}) - \boldsymbol{K_{db}}\dot{\boldsymbol{x}}_{\boldsymbol{b}}(\boldsymbol{k_b}) + \boldsymbol{J_b^T}(\boldsymbol{k_b})\boldsymbol{F_m}(\boldsymbol{k_b}). \end{aligned} \tag{7.38}$$

During inter-sampling period (k_m and k_b), the discrete control law might not cancel the respective power terms as can be seen by substituting $\boldsymbol{F_m}(\boldsymbol{k_m})$ from (7.37) and $\boldsymbol{F_b}(\boldsymbol{k_b})$ from (7.38) into (7.32). Therefore, the discrete nature of the controllers leads to creation of virtual energy which destroys the passivity of the system and it might lead to instability [SSvdSF05].

In particular, this effect can be seen in the following simulation considering the Example 5, but using the discrete controllers (7.37) and (7.38) and assuming sampling time $T_m = 0.001s$ and $T_b = 0.3s$ (typical sampling-times for controllers in space scenarios [TEDS+17]). The behaviour of the system is shown in Fig. 7.9. Both the tracking error of the end-effector (top) as well as the error at the base increase (see $\Delta\phi_b$) and the system results to be unstable.

In the following section, it will be shown how to remove the destabilising effects of the multi-rate controllers.

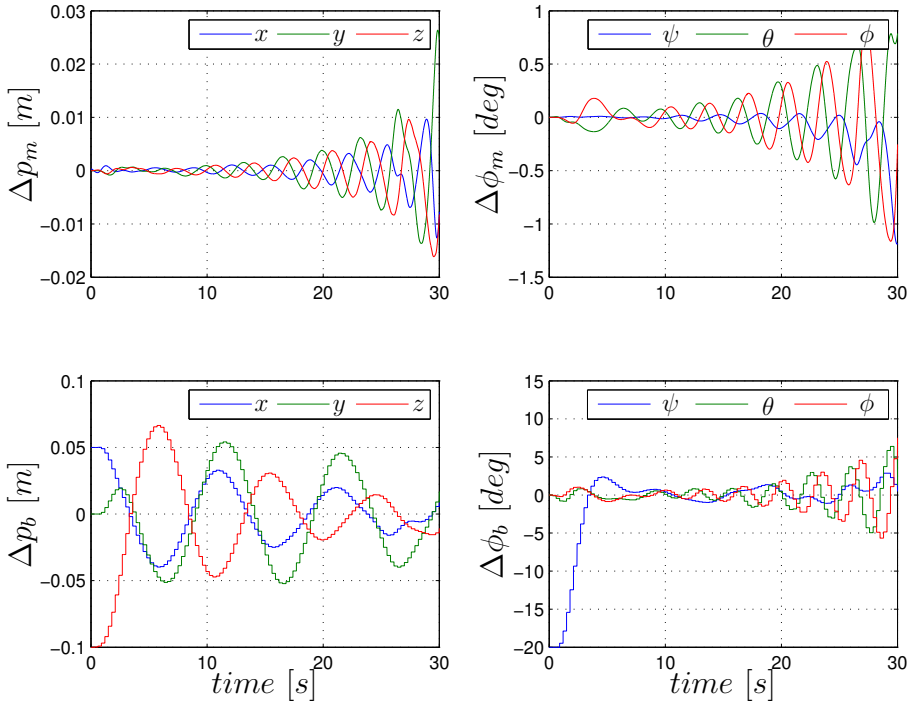


Figure 7.9: Unstable system with discrete-time controllers. Tracking error (top) and error of the base (bottom).

7.5 Energy-based Coordinated Control for Tracking

A passive control action similar to the one discussed for the regulation case in Sec. 7.3 is needed to cope with these destabilising effects. However, in Sec. 7.3, the

passivity controller was applied only on the controller at the base. In this section, the use of two passivity controllers (one each for manipulator and base controllers) is considered. Indeed, computational constrains might impose also a low rate for the manipulator control and it can introduce some activity in the system. Therefore, it is advisable to render passive the entire system. The energy condition introduced in (7.13) will be exploited to prove the passivity, which is a sufficient condition for stability [vdS00].

Network representation for the manipulator and base controllers

In order to perform the energy analysis, the network representation presented for the case of the regulation in Sec. 7.3 needs to be augmented with the tracking terms. Therefore, the network of the system with the proposed controllers is shown in Fig. 7.10. The block S represents the dynamics of the manipulator on the actuated base as per equation (7.22) and it receives, from the left side, the controller force of the base $\boldsymbol{F_B(k_b)}$, i.e. $\boldsymbol{F_B = F_b - J_b^T F_m}$ and from the right side the controller forces of the manipulator $\boldsymbol{F_m(k_m)}$ as in (7.37).

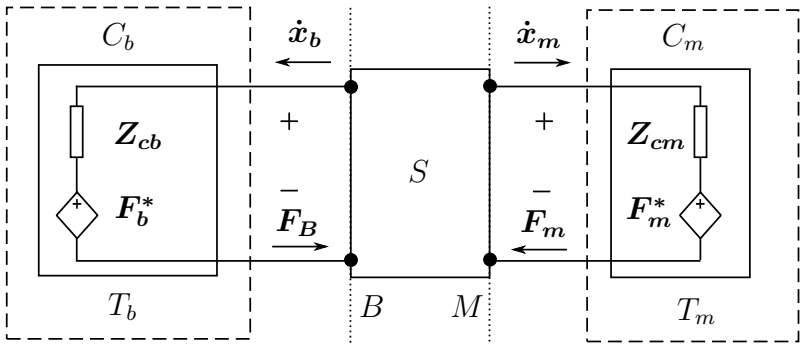


Figure 7.10: Tracking case: network representation of the satellite-manipulator system (S) with tracking controller C_m for the manipulator and regulation controller for the base C_b.

The network of the controller at the base is represented with C_b, which contains the dependent terms $\boldsymbol{F_b^* = M_{bm}\ddot{x}_{md} + \Gamma_{bm}\dot{x}_{md}}$ and the impedance of the stiffness and damping terms, Z_{cb}. Similarly, the network for the manipulator controller is represented with C_m, which contains the dependent term $\boldsymbol{F_m^* = M_m\ddot{x}_{md} + \Gamma_m\dot{x}_{md}}$ and the impedance of the stiffness and damping terms, Z_{cm}.

In this case, both controllers are represented within dashed boxes in Fig. 7.10 because they run in discrete-time. In particular, C_b runs at a low rate T_b and C_m at a higher rate T_m. This difference in rates while connecting the discrete-controllers to the system dynamics through the ports B and M leads to an unstable system.

In the design of the controller in continuous-time, (7.28) represents the energy function of the controlled system (interpreting the proportional and the derivative actions of the controllers as springs and dampers). The control actions compensate the coupling between the base and the manipulator which can achieve tracking by dissipating all the energy of the controlled system, as evident from (7.31) and

(7.32) which prove the stability of the system. However, since the controllers run at different rates, the compensation of the coupling is not perfect and, therefore it creates virtual energy which destroys the energy balance of the system.

Since the system S is passive, the passivity of the controllers can be enforced and the overall controlled system in Fig. 7.10, given by the interconnection of passive systems, can be rendered passive. Therefore, the fact that the overall internal energy does not increase, it will enforce stability through passivity [vdS00]. This can be achieved by monitoring the energy flows at the ports B and M and, then using passivity controllers it is possible to ensure the passivity of the overall system. In particular, the power correlated variables at the ports B and M of Fig. 7.10 need to be identified. These can be expressed as *(input,output)* variables as: $(\boldsymbol{F_B}, \dot{\boldsymbol{x}}_{\boldsymbol{b}})$ and $(\boldsymbol{F_m}, \dot{\boldsymbol{x}}_{\boldsymbol{m}})$, respectively. Then, passivity analysis of these ports will be performed in the following subsections by exploiting the TDPA approach.

7.5.1 Passive Tracking Control

The passivity controllers for the base and manipulator are characterised by an energy observer and a controller which acts in impedance causality. The passivity observers are designed to monitor the energy of the ports using (7.13) and will be applied to each degree-of-freedom (dof) independently. Enforcing passivity component-wise on the variables composing a port is sufficient for enforcing the passivity of the overall port. In fact, considering (7.13) and neglecting $E(0)$, the energy can be split to the n components as follows:

$$\sum_{k=0}^{m}(\boldsymbol{F}(k)^T\boldsymbol{v}(k)T) = \sum_{k=0}^{m}\sum_{i=1}^{n}F_i(k)v_i(k)T, \tag{7.39}$$

where it follows to say that if $\sum_{i=1}^{n}F_i(k)v_i(k)T > 0$ for each $i = 1, \ldots, n$ then $\sum_{k=0}^{m}(\boldsymbol{F}(k)^T\boldsymbol{v}(k)T) > 0$.

PO-PC for the manipulator controller

A first PO-PC architecture is implemented on the port $(\boldsymbol{F_m}, \dot{\boldsymbol{x}}_{\boldsymbol{m}})$ to monitor the activity of the manipulator on the port M. Therefore, the passivity observer is defined as:

$$E_{obs_m}(k_m) = E_{obs_m}(k_m - 1) + F_m(k_m)\dot{x}_m(k_m)T_m + \beta_m(k_m - 1)\dot{x}_m(k_m - 1)^2T_m, \tag{7.40}$$

where β_m is the variable damper calculated as:

$$\beta_m(k_m) = \begin{cases} -\frac{E_{obs_m}(k_m)+E_m(0)}{\dot{x}_m(k_m)^2T_m} & E_{obs}(k_m) < -E_m(0) \\ 0 & else \end{cases} \tag{7.41}$$

and $E_m(0)$ is the initial energy of the manipulator. The second term on the right side of (7.40) is the input energy at the port M and the last term is the update of

energy which is dissipated by the passivity controller whose force is given by:

$$F_{pc_m}(k_m) = -\beta_m(k_m)\dot{x}_m(k_m). \tag{7.42}$$

Therefore, if passivity condition is violated, the correction in force will be provided to the manipulator as follows:

$$F_m^{'}(k_m) = F_m(k_m) + F_{pc_m}(k_m). \tag{7.43}$$

PO-PC for the base controller

A second PO-PC monitors the port $(\boldsymbol{F_B}, \dot{\boldsymbol{x}}_{\boldsymbol{b}})$ at the base. In order to have higher accuracy, the passivity observer for this port is implemented in the system running at faster rate (T_m) and it is defined as:

$$E_{obs_b}(k_m) = E_{obs_b}(k_m - 1) + F_B(k_m)\dot{x}_b(k_m)T_m + \beta_b(k_m - 1)\dot{x}_b(k_m - 1)^2 T_m, \tag{7.44}$$

where the velocity of the base $\dot{x}_b$ can be measured at sampling rate T_m. The total energy $E_{obs_b}(k_m)$ is updated at each sampling time T_m and the values of F_B change at each time step T_b and between two values, the observer holds the previously received value. Similar to the energy observer for the manipulator, the second term on the right side of (7.44) is the energy of the port B and the last one is the update of the passivity controller. The variable damper β_b is derived as:

$$\beta_b(k_m) = \begin{cases} -\frac{E_{obs_b}(k_m) + E_b(0)}{\dot{x}_b(k_m)^2 T_m} & E_{obs_b}(k_m) < -E_b(0) \\ 0 & else, \end{cases} \tag{7.45}$$

where $E_b(0)$ is the initial energy of the base. Then the force of the passivity controller at the base is calculated as:

$$F_{pc_b}(k_m) = -\beta_m(k_m)\dot{x}_b(k_m). \tag{7.46}$$

Although the passivity controller is calculated at high rate, its force correction is provided at the sampling step T_b. Therefore, when passivity condition is violated, the force provided to the system through the base controller is:

$$F_B^{'}(k_b) = F_b(k_b) + F_{pc_b}(k_b). \tag{7.47}$$

The network with the PCs placed on the manipulator and base controller is shown in Fig. 7.11. As can be seen, $F_B^{'}$ is the force vectors modified by the PO/PC architecture for the base controller and $F_m^{'}$ for the manipulator controller. Therefore, both the controllers in the dashed-boxes endowed with the PCs are rendered passive. Consequently, the overall system is passive since it is an interconnection of passive networks.

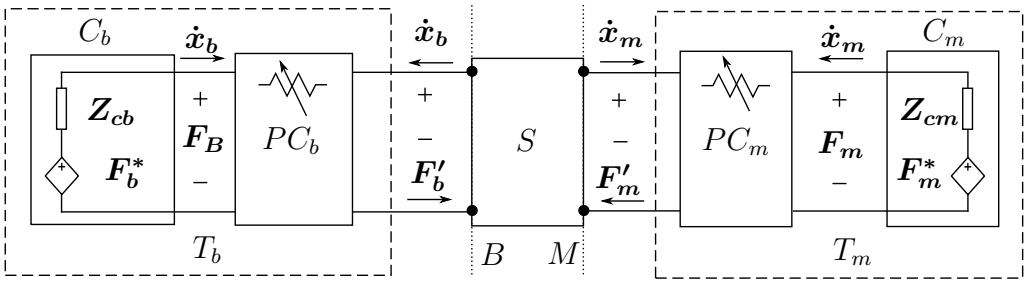


Figure 7.11: Tracking case: network representation scheme with tracking control C_m and controller at the base C_b endowed with the two PCs.

7.5.2 Results: Simulations and Experiments

The first part of this section shows the simulation results performed with the proposed controllers. The manipulator control runs at $T_m = 0.001s$ and the base control runs at $T_b = 0.3s$ and the same condition described in Example 5 (including the trajectory) are considered. Fig. 7.12 shows the error of the manipulator during the tracking and the error of the base for the regulation, respectively. As can be seen the error converges. In particular it gets really close to the continuous case scenario. To understand the benefit of the controller, it is worth to compare the error shown in Fig. 7.9, where the passivity controller was not applied with Fig. 7.12 which uses the proposed controller.

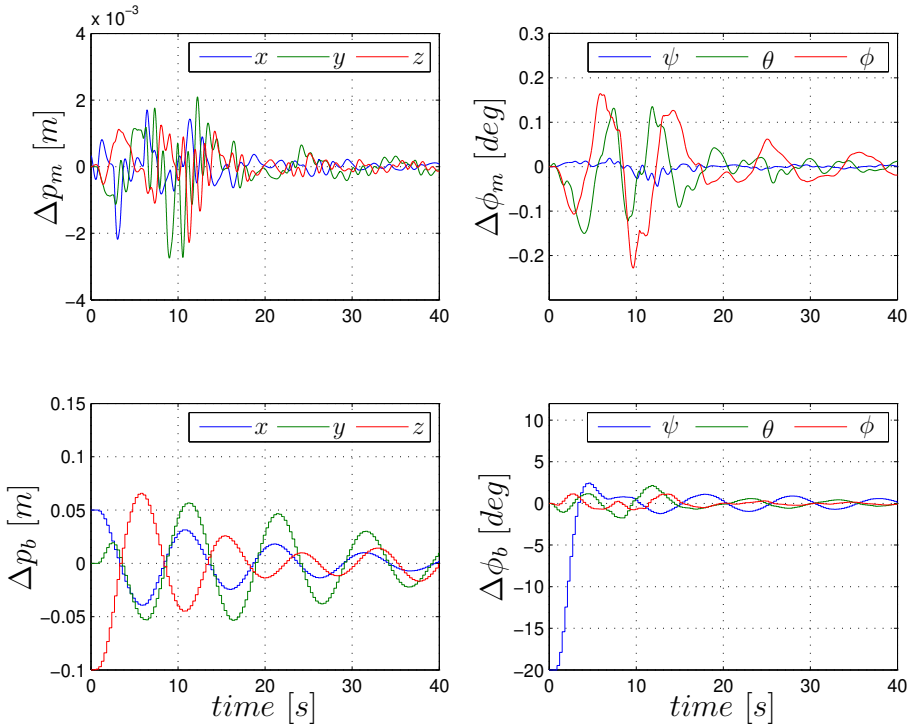


Figure 7.12: Tracking error of manipulator (top) and regulation error of the base (bottom).

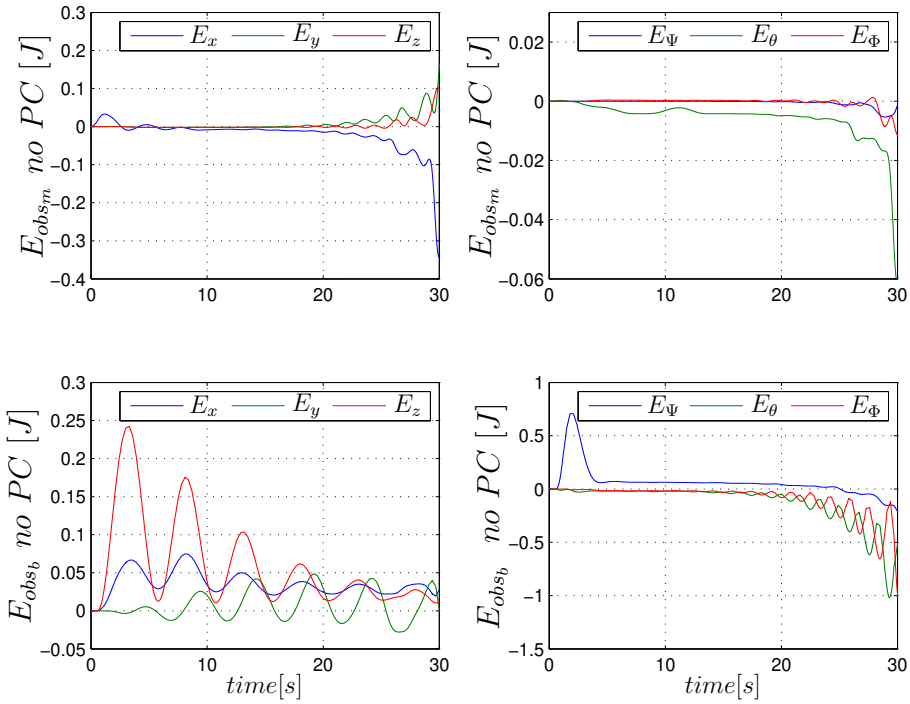
Regarding the passivity, Fig. 7.13a shows the energy observed in the system for the base (top) and manipulator (bottom) without the action of the passivity control. The negative trend indicates activity in the system and it caused the instability showed in Fig. 7.9. The energy observers with the actions of the PCs are shown in Fig. 7.13b where the positive semi-definiteness indicates the passivity of the system. It can be seen that all the activity has been removed and the errors converge.

Experiments

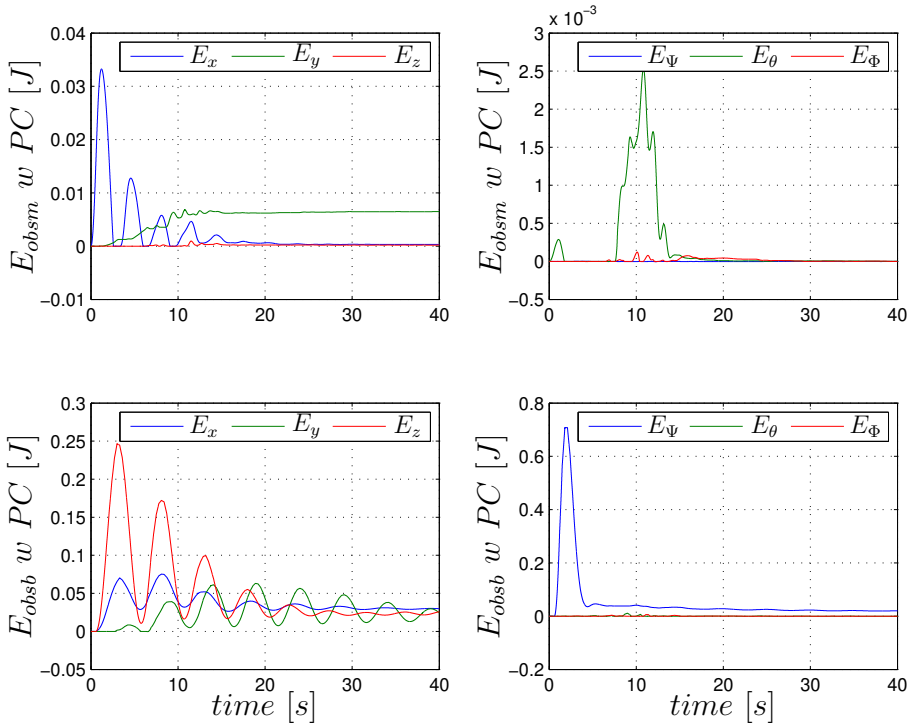
The passive tracking control architecture has been validated on the servicer robot of the OOS-Sim facility, where the control of the base runs at $250Hz$ and manipulator at 1000 Hz. The mass and inertial parameters of the space robot are the same as reported in Example 5. A linear trajectory of 20 cm is commanded to the Light-Weight-Robot arm along the $-z$ axis in the inertial frame (see Fig. 6.2) for 10 seconds and the base is regulated about a set-point in a relative position of $[0.04, 0, 0.08]$ m and $[10, 0, 0]$ deg in orientation with respect to the inertial frame. Fig. 7.14a (top) shows the tracking error of the manipulator in position and orientation and the applied forces ($\boldsymbol{F_m}$) are shown at the bottom, respectively. As it can be seen, during the tracking (first $10s$) the maximum error is 0.008 m in position and 1.1 deg in orientation.

The regulation error for the base is shown in Fig. 7.14b (top) and the respective applied force are shown at the bottom of Fig. 7.14b. The residual error in position and orientation for the base is 0.012 m in position and -2 deg in orientation. The main reason is given by the residual forces $\boldsymbol{F_m}$ applied by the manipulator to overcome its own disturbances and it is transmitted to the base as defined in (7.38). During the experiment, the energy observers measured activity given by the multi-rate controllers as shown in Fig. 7.15a. The extra energy is compensated with the passivity controllers to render the system passive. The energy observers with the action of the passivity controllers are shown in Fig. 7.15b. As can be seen, the energy is greater than zero respecting the passivity condition.

A second experiment has been performed. A trajectory was commanded to the manipulator in order to reach the grasping point located on the client and the control at the base regulates its motion about its initial pose. The snapshots of the phases of the experiment are shown in Fig. 7.16.

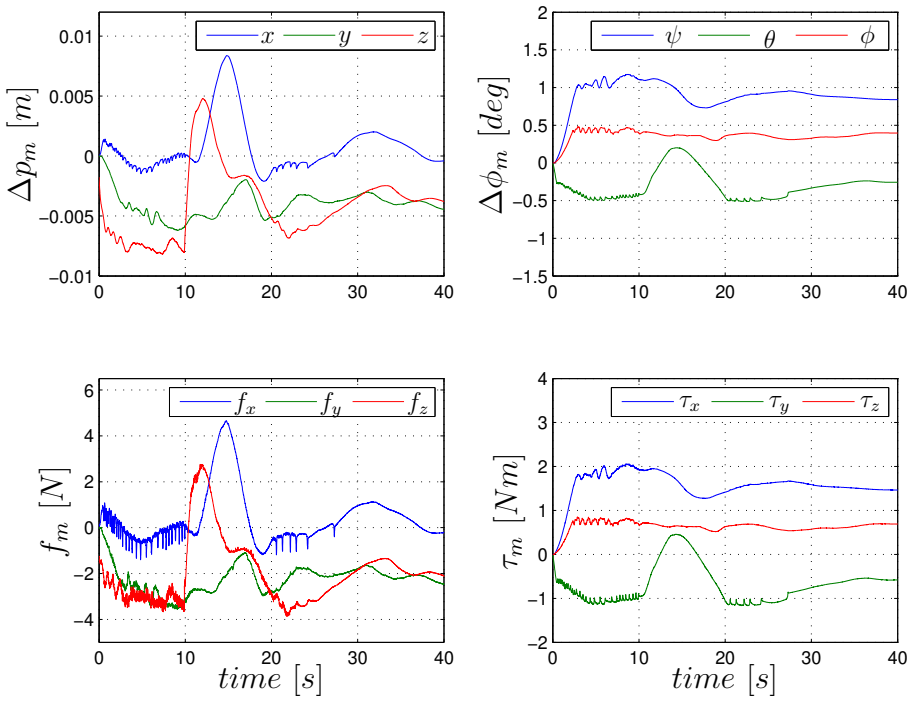


(a) Energy observers for manipulator (top) and base (bottom) without PC which creates instability shown in Fig. 7.9

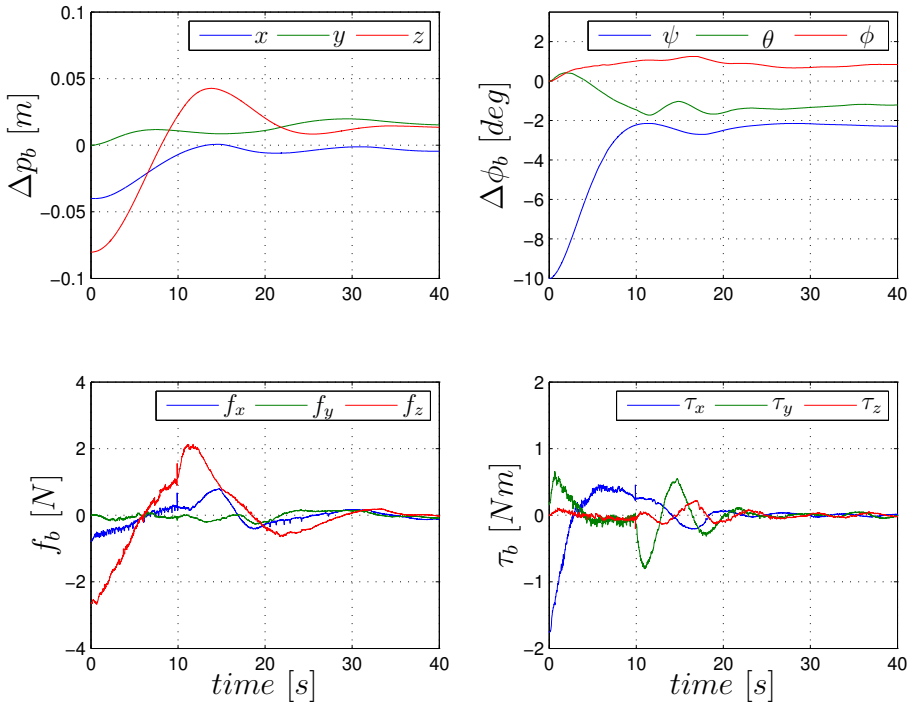


(b) Energy observers for manipulator (top) and base (bottom) with PC.

Figure 7.13: Energy observer without and with PC during tracking.

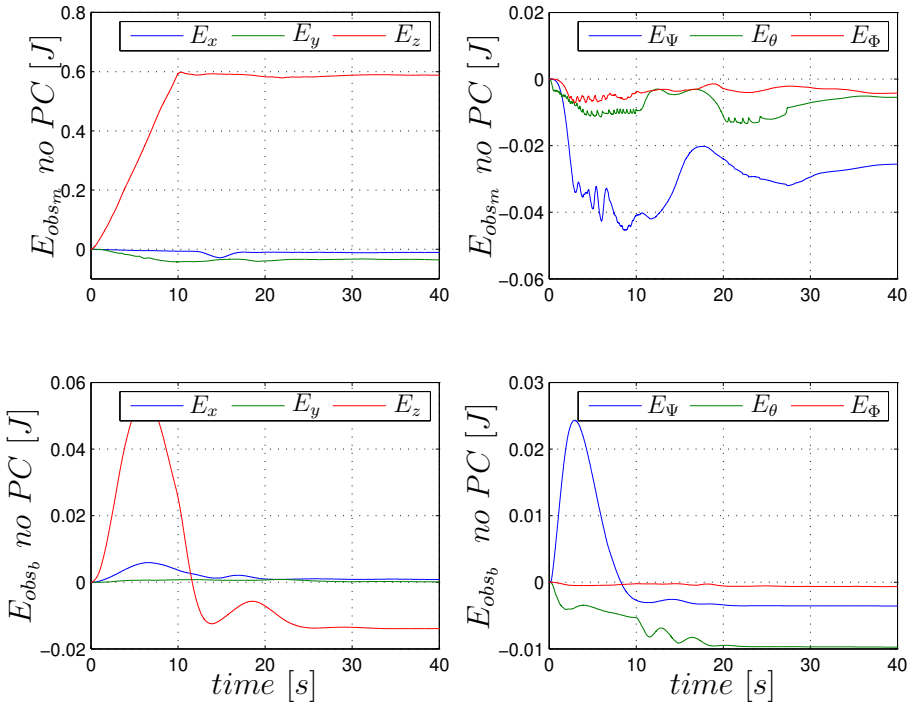


(a) Experiment results: Tracking error in position and orientation (top) and Force $F_m = (f_m, \tau_m)$ (bottom) of the manipulator.

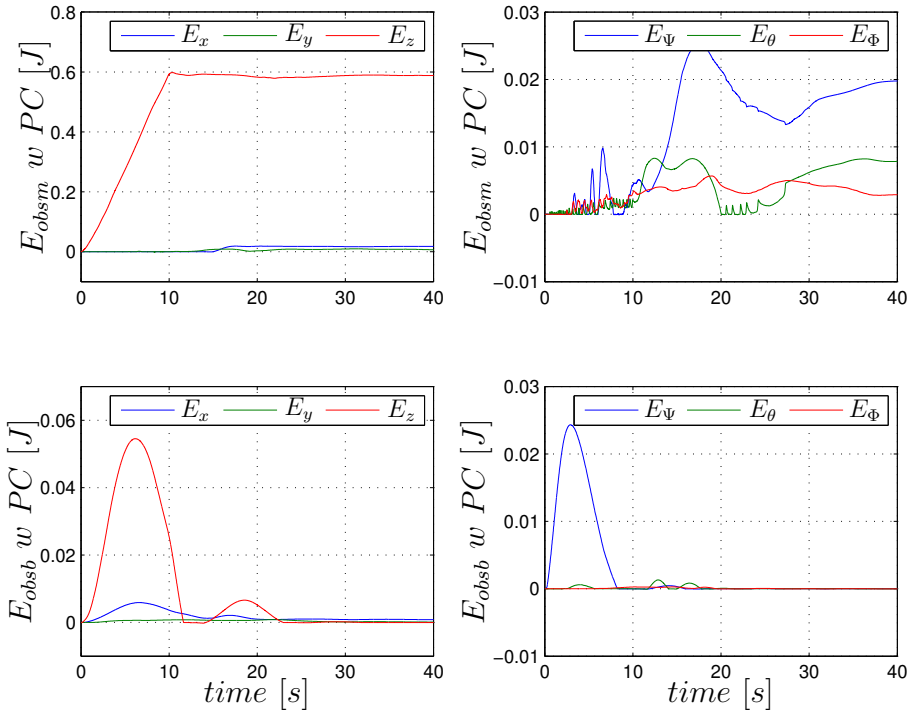


(b) Experiment results: regulation error of the base (top) and force at the base $F_b = (f_b, \tau_b)$ (bottom).

Figure 7.14: Experiment results: Error and forces of the manipulator during tracking Fig. 7.14a and error and forces of the base during regulation Fig. 7.14b.



(a) Experiment results: Energy observers without PC for the manipulator (top) and base (bottom). Negative trend indicates activity.



(b) Experiment results: Energy observers with PC for the manipulator (top) and base (bottom). Positive trend indicates passivity.

Figure 7.15: Experiment results: Energy observers for manipulator and base.

7.6 Discussion

Space missions requirements impose hard constraints on the on-board power and the relative frequencies that can be used for the control. This factor can affect the stability of a designed controller.

In this chapter, it has been shown how different sampling rates between two controllers operating on the free-flying robot can deteriorate the performances and lead to instability. Therefore, two impedance controller have been designed firstly in continuous time with rigorous stability analysis and then adapted for the multi-rate case. The first controller resolves a regulation problem for the base and the manipulator, the second one deals with a more challenging problem, which is the tracking of a trajectory with the manipulator and the regulation of its base. It has been shown that the controller architecture endowed with passivity observers and passivity controllers can fulfil the tasks of regulation and tracking a desired end-effector trajectory while ensuring stability. The passivity observers have been designed to monitor the activities of the base and manipulator and passivity-based controllers have been used for removing the activities. As a result, the overall system is stable. The proposed controllers have been validated in simulations and verified experimentally on the OOS-Sim facility.

Fig. 7.17 shows the snapshots of an experiment, which resembles the phases of an on-orbit servicing mission. These phases include the tracking control for the space manipulator with the impedance control developed in Sec. 7.4, the grasping of the target satellite and the relative stabilisation with the controller developed in Sec. 6.3.3. All these phases, including the stability of the simulator, profit from the passivity-based controllers developed in this thesis.

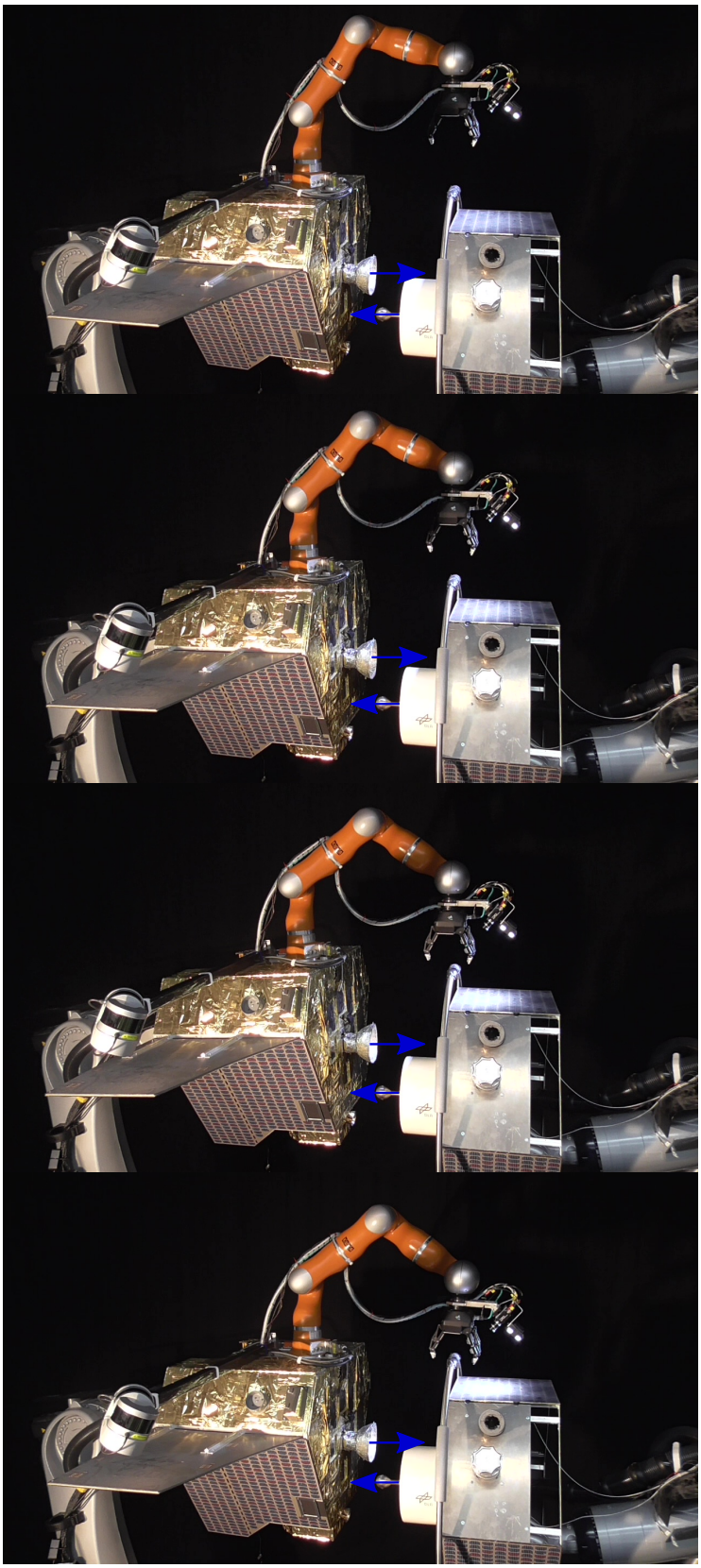
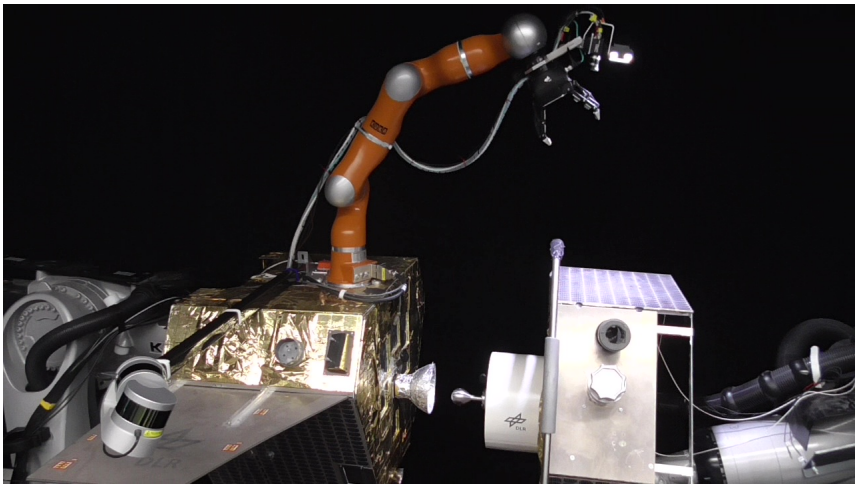
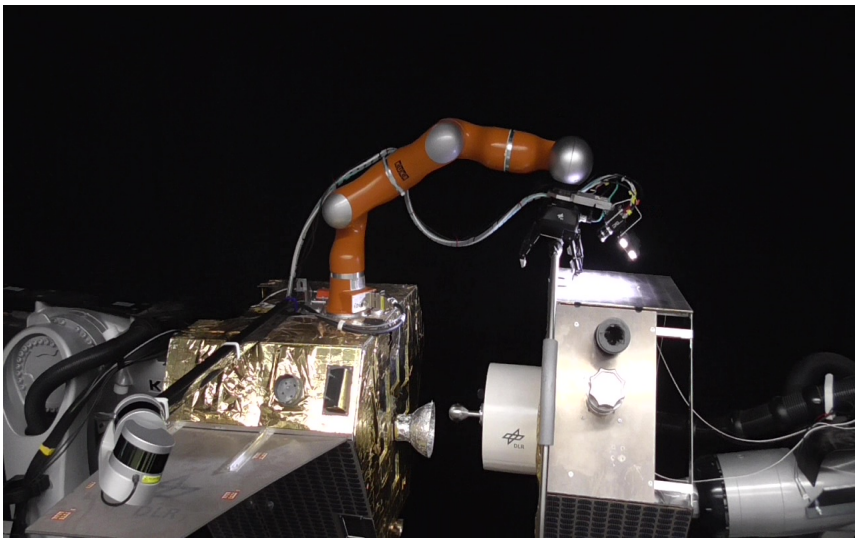


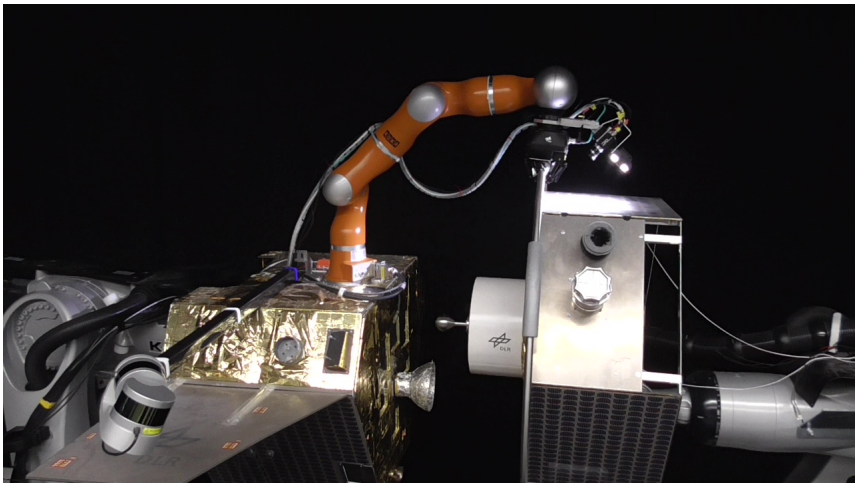
Figure 7.16: Snapshots of the experiment during tracking and base regulation. The control at the base holds the initial pose and the manipulator tracks a trajectory to reach the grasping point on the client.



(a) Impedance tracking control of a moving target.



(b) Approach to the grasping point of the moving target.



(c) Grasping and stabilisation of the moving target.

Figure 7.17: Snapshots of the experiment for an OOS-like manoeuvre: tracking, grasping and stabilisation of a moving target. All these phases, including the stability of the simulator, profit from the passivity controllers developed in this thesis.

CHAPTER 8

CONCLUDING REMARKS

The persistent progress witnessed in the field of robotics has pervaded into several domains resulting in development of advanced control algorithms. A robotic system can be complementary in tasks which are considered hazardous for human e.g., in extra vehicular activities in space. Consequently, the research community has identified this possibility and therefore an urgent need has arisen for faithful simulation of space scenarios on ground. Robotic facilities provide a solution to simulate micro-gravity conditions and to test the control algorithms for space applications.

The goal of this thesis was to design controllers for efficiently simulating, on ground, a model-based dynamics, e.g. satellite using admittance-controlled robots and for controlling a space robot in impedance mode. The control strategies and the methods proposed in the thesis focused on the stability and passivity of a robotic system where energy characteristics are exploited.

To address the problem of time-delay between the measured forces and the data commanded to the robot, a novel passivity-based control strategy was proposed and optimized while taking performance into account. The optimized approach guarantees stability through passivity and preserves the performance through the use of an optimal damping. Energy drifts caused by the discrete integration process were tackled with a novel discrete and passive integration method, which is able to run in real time and, therefore, can be easily implemented on a real robot. Further, the proposed unified architecture developed in the thesis can deal with time-delay, and discrete-time integration for rendering dynamics while using a robotic platform.
This architecture contributes also to the state of the art in various fields, such as control design and stability theory. Apart from the theoretical contributions in the aforementioned topics, the proposed methods were also proved to be effective through validation on a robotic facility, namely the OOS-Sim. In particular, experiments were performed in Chapter 3 for the the passivity-based control, which deals with the time-delay problem, in Chapter 4 for the passive integration and in

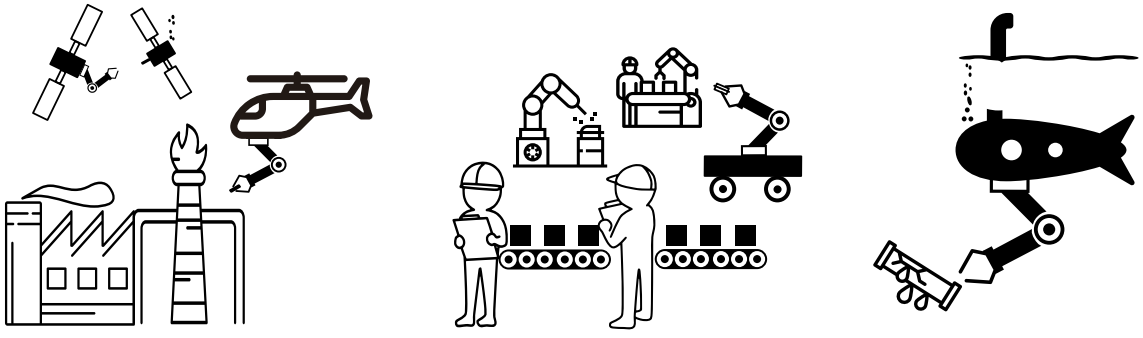
Chapter 5, for the unified approach.

A further contribution of the thesis was the development of impedance controllers for the space manipulator arm. The advantage of using impedance control is the compliance which can be achieved and it allows robust task execution and a safe interaction with an uncertain environment. This factor is important in different fields such as human-robot interaction, where safety of the operator needs to be guaranteed or in on-orbit servicing, where impact forces generated by the arm during the grasping of a target satellite needs to be controlled.

In particular, an on-ground validation of the impedance control for a free-floating robot arm was presented in Chapter 6, with applications in the area of teleoperation and space robotic projects. In Chapter 7, a novel approach with stability proofs for the impedance control of a manipulator mounted on an actuated base was designed for resolving the regulation and tracking problems. The theoretical contribution was related to the energy analysis performed while considering the multi-rate controllers operating on a multi-body systems. Indeed, when hard constraints for the controllers' frequencies are imposed, the passivity properties are lost thereby negatively affecting the closed-loop stability. The proposed approach modifies the feedback loop exploiting the passivity criteria, which guarantees a stable behaviour. The impedance controllers were successfully validated on the on-ground robotic facility for a space mission scenario, where the space manipulator arm approaches a target satellite.

The controllers developed within this thesis, can be generalised for a wide range of applications in industrial, household and even underwater scenarios. In particular, mounting and controlling light-weight manipulators on mobile platforms improves the mobility and flexibility during the operations. These properties make the manipulator especially useful in hazardous environment and area with limited accessibility e.g., nuclear power plants, disaster-stricken sites or for underwater maintenance operations. A sketch of the general domains of applications is shown in Fig. 8.1.

Oil and gas refineries are also classified as dangerous sites. Usually, specialised technicians operate on pipelines located at high level from ground to perform point-



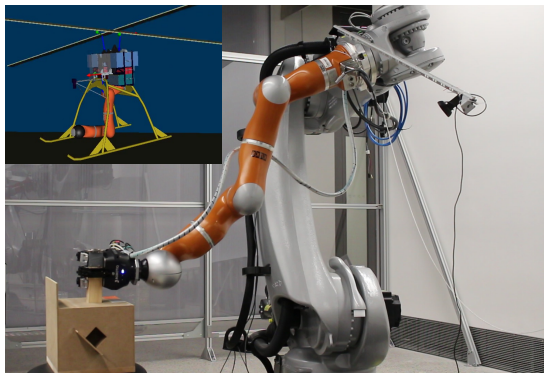
Space and aerial Industry Underwater

Figure 8.1: General domains of applications: space and aerial, industrial and underwater scenario.

contact measurements, maintenance and sensors installation.
Within this context, aerial platforms equipped with robotic arm(s) are recognised by industries and researchers as a promising technology because they can increase the safety and reduce the costs, see e.g. the EU AEROARMS project [OHF+18]. An aerial manipulation system, indeed, can operate in such dangerous sites and can deploy a magnetic-wheeled robotic crawlers on pipelines in order to perform point-contact measurements. Recently, DLR has developed a helicopter equipped with a manipulator to support inspection operations and deployment of the magnetic-wheels robotic crawler on the pipelines, see Fig. 8.2a [LHK16]. The release of the crawler on the pipe involves force interaction and a passive impedance control for the manipulator is required also in this case. An impedance controller, similar to the one presented in Chapter 7, was also applied for the aerial scenario (see [KBDS+18]) and it was possible to achieve a stable interaction with the environment.

Furthermore, in order to allow repeatability of the flight experiments under desired conditions, a solution based on hardware-in-the-loop simulation can be adopted. This solution follows the one proposed in the thesis for the space scenario. Therefore, the industrial robot equipped with a force-torque sensor can be exploited to simulate the helicopter dynamics, as shown in Fig. 8.2b. The light-weight robot arm is identical to the one mounted on the helicopter and it is used for manipulation tasks. Fig. 8.2b shows a hardware-in-the-loop simulation where the manipulator arm performs a peg-in-hole task and interactions forces are generated. Effects of time-delay in the control loop, during the simulation of dynamics using robotic facilities, has been thoroughly investigated in the thesis. Currently, the stability of this robotic simulator is achieved also with the passivity controls developed within the thesis to mitigate the effects of the time-delay and discretization.

(a) DLR Helicopter equipped with a light-weight-robot during flying experiment.



(b) Industrial robot equipped with a light-weight-robot simulating an aerial task.

Figure 8.2: Aerial manipulation scenario. Helicopter equipped with a light-weight-robot (left) and on-ground test of an aerial task (right). The industrial robot simulates the helicopter dynamics and the light-weight-robot performs the task of approaching, grasping and lifting an object. The energy framework developed within this thesis is considered also for the aerial scenario within the AEROARMS project.

At the beginning of this thesis, the concept of exploiting robots to support the humans in space was abstractly introduced through science fiction authors. With the current development in the field of robotics in all the aforementioned domains, it is a matter of time before robots will complement actual orbital missions. During the course of this thesis, this abstract concept was de-constructed into concrete mathematical and physical formulations addressing practical issues like time delay, physical consistency of on-ground simulators and requirement of impedance control, all of which are required before proving mission readiness for such a venture.
Moreover, the topics developed in this thesis are relevant to all the scenarios mentioned above and therefore this thesis provides an overarching analysis of the challenges and solutions.

APPENDIX A

TECHNICAL DATA

A.1 OOS-Sim Facility: Technical Data

A.1.1 Communication and Sensors

The main hardware elements of the OOS-Sim facility are summarized in Table A.1 with the respective interface, sampling time and input-output (I/O) characteristics. The OOS-Sim facility allows also to validate the interaction between these elements. For instance, visual servoing can be tested along with impedance control in a free-floating setup. The sensors involved in the facility are summarized in Table A.2 with the respective interface and functional description.

A schematic of the communication between hardware and sensor is show in Fig. A.1. A real-time computer (RT-C1) performs the computation of the model-based dynamics for the servicer and client. This computer runs at a sampling of 4 ms. A second real-time computer (RT-C2) runs at 1 ms and controls the manipulation arm (LWR) with a Sercos interface which allows real time communication. The two real-time computer communicate through udp-protocol and consequentially

	Robot	**DoF**	**Interface, T**	**Input/Output**
Servicer	KR120	6	Ethernet, 4ms	X/X
Manipulator	KR4+	7	Sercos, 1ms	$X, F/X, F$
Client	KR120	6	Ethernet, 4ms	X/X
Gripper	Robotiq 3f	4+8	Ethercat, 1ms	$X, I/X, I$
Haptic M.	KR4+	7	Sercos, 1ms	X/X

Table A.1: Hardware components of the OOS-Sim facility. X=position; F=force; I=current; T=sampling time.

	Sensor	Interface, T	Function
Client FTS	FTS160	Ethernet, 4ms	Physical interaction
Servicer FTS	FTS160	Sercos, 1ms	Physical interaction
Haptic FTS	FTS78	Ethernet, 4ms	Dynamics compens.
Cameras	GigE	Ethercat, 1ms	Visual servoing
IMU	Xsense MTI	Sercos, 1ms	Star field simulation

Table A.2: Sensors and interface of the OOS-Sim facility. FTS=Force-Torque Sensor; IMU=Inertial Measurement Unit.

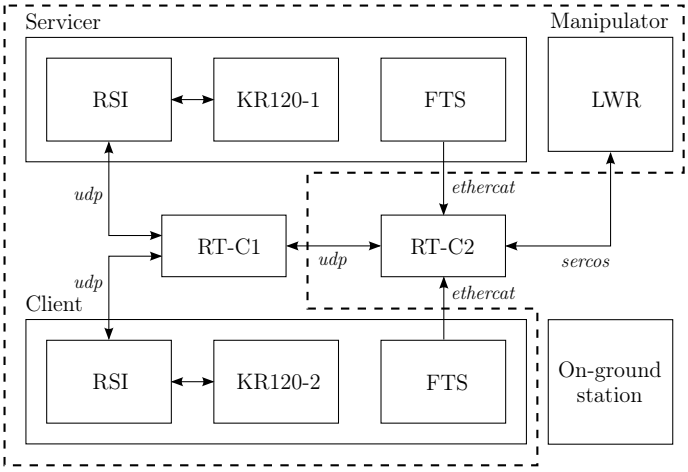


Figure A.1: Overview of the system communication. The dashed line divides simulated and mission-like elements

with the Robot Sensor Interface (RSI), which communicate with the robot KR120-1, the servicer and KR120-2, the client. Furthermore, the RSI interface is synchronized to RT-C1 in order to preserve real time determinism. Thus, RT-C2 delivers both desired position and orientation commands to the RT-C1. The RT-C1 delivers current position and orientation signals to the RT-C2 to allow controller and dynamics computations. The sensor data from both FTS are fed to the RT-C2 using the real time clock of the RT-C2. The schematic highlights the simulated elements which are those corresponding to the space environment with respect to those that do not need to be simulated and can be taken as in the real mission. Thus, on-ground station and main control CPU are regarded as mission-like hardware.

A.1.2 Workspace

For the workspace evaluation, the facility is treated as a two serial kinematic chains, i.e. a 13-dof (KR120 servicer with LWR) and 6-dof (KR120 client). The capability maps are shown in Fig. A.2, where reachable discretized directions are indicated by the voxel ranging from red (close to 0%) to blue (close to 100%). The underlying voxel grids can be merged into a new map to obtain the common workspace volume

including the direction information (more information about the methods for generating such maps are described in [PSCR13]). The resulting volume of the space environment which can be simulated with the OOS-Sim is approximately 23.28 m^3.

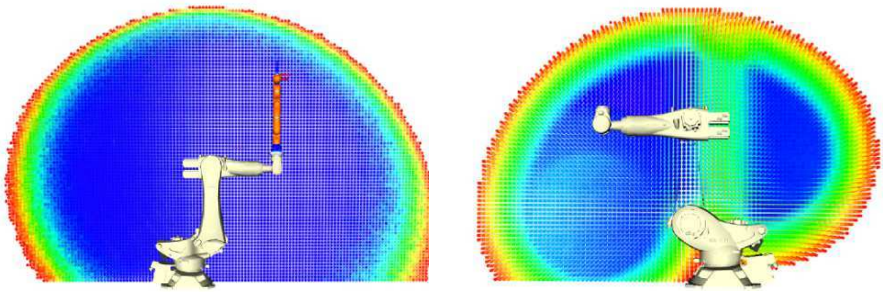


Figure A.2: Workspace of OOS-Sim: servicer robot(left) and Client robot (right).

A.2 Further Experimental Set-up

The methods presented in this thesis are sustained by a set of experiments, which include the OOS-Sim facility, LWR and a 1 dof set-up. The 1 dof set-up is shown in Fig. A.3, which is composed of motor-gear-unit developed by Sensor-drive GmbH and it is equipped with a torque sensor. The motor is based on brushless technology with a nominal torque of 0.7 Nm and torque peak of 2 Nm. The system runs on a *QNX* real-time operating system at 1 *ms* sampling rate. Technical data are summarized in Table A.3.

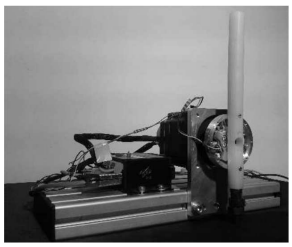


Figure A.3: 1 dof Sensor drive set-up equipped with a torque sensor.

Operational Frequency	1 kHz
Bus Interface	CAN or ethercat
Intrinsic Delay	2 ms
Nominal torque	0.7 Nm
Pick torque	2 Nm
Motor type	Brushless DC motor

Table A.3: Technical data of the 1 dof system.

APPENDIX B

CONTACT MODELING

The term *contact* indicates a mechanical interaction between two bodies which boundaries share the same location. An *impact* on the other hand, is typically regarded as a contact involving large accelerations and changes in the energy of the two bodies. It is important to note that while a contact can be continuous, an impact is typically of very short duration. This section aims to shortly review the most widely used contact models, as well as the conditions for their use. These models can be typically classified in two types, *impulse-momentum* or *discrete* models, and *force based* or *continuous* models [GS02].

B.1 Discrete models

Discrete contact models assume that the contact happens during an extremely short period of time and thus all the forces can be regarded as impulsive. This implies that changes in energy and momentum will be discontinuous, and that in this context contact and impact are synonyms. These models characterize the impact through one or several coefficients. The most widely used models in this category are *Newton's*, *Poisson's*, and *Stronge's* models. Given an initial velocity and angular momentum, $\boldsymbol{v}_{i0}$ and $\boldsymbol{h}_{i0}$, the models aim to determine the final velocities $\boldsymbol{v}_i$ and the linear and angular impulses $\boldsymbol{P}$ and $\boldsymbol{M}$, respectively. Therefore, given the masses of the bodies m_i (with i = 1,2) and the distance between the impact point and the center of mass of each body $\boldsymbol{d}_i$, it results,

$$\begin{aligned} m_1(\boldsymbol{v}_1 - \boldsymbol{v}_{10}) &= \boldsymbol{P}, \\ m_2(\boldsymbol{v}_2 - \boldsymbol{v}_{20}) &= -\boldsymbol{P}, \\ \boldsymbol{h}_1 - \boldsymbol{h}_{10} &= \boldsymbol{d}_1 \times \boldsymbol{P} + \boldsymbol{M}, \\ \boldsymbol{h}_2 - \boldsymbol{h}_{20} &= \boldsymbol{d}_2 \times \boldsymbol{P} - \boldsymbol{M}. \end{aligned} \tag{B.1}$$

The three aforementioned models aim to impose conditions upon the system

from equation (B.1) so that they can be solved. Usually, the expression use the *coefficient of restitution*. Such coefficient of restitution is defined as the ratio of the relative tangential velocity before and after the contact. This definition is quite extensively used throughout the literature, see [NUY10, ZLBC12, UNNY12].

The discrete models do have a series of disadvantages though, since they produce inconsistent results under certain conditions and they disagree on the amount of energy that has been dissipated [FDDR17]. Also the use of Coulomb's law of friction give raise to inconsistencies as well, producing none or many solutions for the problem in certain cases.

B.2 Continuous models

Continuous models analyse the contact at every point of time by considering the forces and interactions also during the contact. This requires an accurate modelling of such forces, which typically are considered to depend on the local deformation δ and its derivative, $\dot{\delta}$. For a normal component of the contact force F_n, the general formulation will be

$$F_n = F_n(\delta, \dot{\delta}) = F_\delta(\delta) + F_{\dot{\delta}}(\dot{\delta}) \tag{B.2}$$

The models used more often are, the *spring-dashpot model*, the *Hertz's model* and the *nonlinear damping model*, which is a combination of the two previous ones.

Spring-dashpot model: This model considers the forces produced by a hypothetical spring and damper which represent the compression-restitution and the energy dissipation, respectively. This model is widely used due to its simplicity, despite it has some weaknesses regarding its realism. These weaknesses are the presence of discontinuities in the contact force, the appearance of a negative contact force toward the end of the contact, and a coefficient of restitution which does not depend on impact velocity [GS02]. The model is expressed by the following equation,

$$F_n = k\delta + d\dot{\delta}, \tag{B.3}$$

where k and d is the stiffens and damping of the contact.

Hertz's model: This model considers a nonlinear spring between the impacting bodies, which can be characterised using elastostatic theory. The Hertz's model does not consider energy dissipation, which makes it adequate only for low speed contacts of hard surfaces, although for this case it is a more realistic model than the spring-dashpot model. The model is usually expressed as,

$$F_n = k\delta^n, \tag{B.4}$$

where k and n are the parameters to determine.

Non-linear damping: This model, also known as Hunt-Crossley model, aims to combine the two aforementioned ones, retaining the advantages of Hertz's model thus providing a more realistic description than the spring-dashpot model. The force model has a non-linear damping term and it is expressed as,

$$F_n = d\delta^p \dot{\delta}^q + k\delta^n \tag{B.5}$$

where k and d are stiffness an damping coefficient and n, p and q are positive real number describing the non-linearity and hysteresis behaviour.

Empirical model: This model is employed when geometry of contact are not axis-symmetric and when the value of k or n are difficult to find. The empirical model is function of the penetration depth and area, A. It is described as,

$$F_n = k(A)d \tag{B.6}$$

where $k(A)$ is a stiffness coefficient that is function of A, the contact parameters and geometry.

Methods related to finite-element are also available in literature, such as the Lagrange multipliers. However, finite-element methods are more complex and computationally expensive when multi-body dynamics are involved [ZLL14].

APPENDIX C

LEMMAS AND THEOREMS

C.1 Definitions

Definition 10 ***Generalised saturation***
Given a positive constant Z, a function $s : \mathbb{R} \to \mathbb{R} : \tau \to s(\tau)$ is said to be a generalized saturation function with bound Z, if it is locally Lipschitz, non-decreasing and satisfies the following:

$$\tau s(\tau) > 0 \quad \forall \ \ \tau \neq 0$$
$$|s(\tau_i)| \leq Z \quad \forall\ \tau \in \mathbb{R}^n$$

Definition 11 ***Linear saturation***
Given a positive constant G and Z with $G \leq Z$, a function $s : \mathbb{R} \to \mathbb{R} : \tau \to s(\tau)$ is said to be a linear saturation for (G,Z) if it is locally Lipschitz, non-decreasing and satisfies the following

$$s(\tau) = \tau \quad when \quad |\tau_i| \leq G$$
$$|s(\tau_i)| < Z \quad \forall \qquad \tau \in \mathbb{R}^n$$

C.2 Lemmas and Theorems

- **LaSalle's invariance principle**. The theorem is reported from [Kha02a] in a compact form.

  Theorem C.2.1 ***(LaSalle's invariance principle)*** *If there exists a continuously differential, positive definite and radially unbounded function $V(x) : \mathbb{R}^n \to \mathbb{R}$ such that $\dot{V} \leq 0$, $\forall x \in \mathbb{R}^n$, then every solution of the system $\dot{x} = f(x)$ converges to the last invariant set M contained in $Z = \{x \in \mathbb{R}^n, \dot{V}(x) = 0\}$. Furthermore, if $M = 0$, the origin of the system $\dot{x} = f(x)$ is globally asymptotically stable.*

- **Barbalat's Lemma**. The Lemma is reported from [SL91] in a form of corollary

  Lemma C.2.2 *(**Barbalat's Lemma**) If a differentiable function $f(t)$ has a finite limit as $t \to \infty$ and its second derivative $\ddot{f}$ exists and it is bounded, then $\dot{f}(t) \to 0$ as $t \to 0$.*

- **Matrosov's Theorem**. The theorem is reported from [LN97] in a compact form.

  Theorem C.2.3 *(**Matrosov's Theorem**) Consider the system $\dot{x} = f(t, x)$ with $f(t, 0) = 0, \forall t \leq 0$. Let $\Omega \in \mathbb{R}^n$ be an open connected region in $\mathbb{R}^n$ containing the origin. if there exist two $\mathcal{C}^1$ functions $V \in \mathbb{R} : [0, \infty) \times \Omega \to \mathbb{R}, W \in \mathbb{R} : [0, \infty) \times \Omega \to \mathbb{R}$. A $\mathcal{C}^0$ function $V^* : \Omega \to \mathbb{R}$; three function a,b,c such that such that for every $(x, t) \in \Omega \times [0, \infty)$:*

  1. $a(\|x\|) \leq V(t, x) \leq b(\|x\|)$
  2. $\dot{V}(t, x) \leq V^*(x) \leq 0$
  3. $|W(t, x)|$ *is bounded*
  4. $|\dot{W}(t, x)| \geq c\|x\|$
  5. $\|f(t, x)\|$ *is bounded*

  then when $t \to \infty$, uniformly $x(t) \to 0$ and the origin is uniformly asymptotically stable.

BIBLIOGRAPHY

[ABDS+16] J. Artigas, R. Balachandran, M. De Stefano, M. Panzirsch, R. Lampariello, A. Albu-Schaeffer, J. Harder, and J. Letschnik. Teleoperation for on-orbit servicing missions through the astra geostationary satellite. In *2016 IEEE Aerospace Conference*, pages 1–12, March 2016.

[ADDJ91] Chaouki Abdallah, Darren M Dawson, Peter Dorato, and Mohammad Jamshidi. Survey of robust control for rigid robots. *IEEE Control Systems*, 11(2):24–30, 1991.

[ADSR+15] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer. The OOS-SIM: An on-ground simulation facility for on-orbit servicing robotic operations. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 2854–2860, May 2015.

[AH99] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, June 1999.

[AHLS96] Sunil Kumar Agrawal, Gerd Hirzinger, Klaus Landzettel, and Richard Schwertassek. A new laboratory simulator for study of motion of free-floating robots relative to space targets. *IEEE Transactions on Robotics and Automation*, 12(4):627–633, 1996.

[ALH06] S. Abiko, R. Lampariello, and G. Hirzinger. Impedance control for a free-floating robot in the grasping of a tumbling target with parameter uncertainty. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1020–1025, Oct 2006.

[ARP10] Jordi Artigas, Jee-Hwan Ryu, and Carsten Preusche. Time domain passivity control for position-position teleoperation architectures. *Presence: Teleoperators and Virtual Environments*, 19(5):482–497, 2010.

[ARPH11] Jordi Artigas, Jee-Hwan Ryu, C. Preusche, and G. Hirzinger. Network representation and passivity of delayed teleoperation systems. In *Intelligent*

Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 177–183, Sept 2011.

[Art14] J. Artigas. Time domain passivity control for delayed teleoperation. In *Ph.D. dissertation, Universidad Politecnica de Madrid*, 2014.

[AS89] R. Anderson and M.W. Spong. Bilateral control of teleoperators with time delay. *Automatic Control, IEEE Transactions on*, 34(5):494–501, May 1989.

[ASBR+06] A. Albu-Schaeffer, W. Bertleff, B. Rebele, B. Schafer, K. Landzettel, and G. Hirzinger. Rokviss - robotics component verification on iss current experimental results on parameter identification. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3879 –3885, may 2006.

[ASJ+14a] Satoko Abiko, Yoshikazu Satake, Xin Jiang, Teppei Tsujita, and Masaru Uchiyama. Delay time compensation based on coefficient of restitution for collision hybrid motion simulator. *Advanced Robotics*, 28(17):1177–1188, 2014.

[ASJ+14b] Satoko Abiko, Yoshikazu Satake, Xin Jiang, Teppei Tsujita, and Masaru Uchiyama. Delay time compensation based on coefficient of restitution for collision hybrid motion simulator. *Advanced Robotics*, 28(17):1177–1188, 2014.

[ASOH07] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Rob. Res.*, 26(1):23–39, January 2007.

[ATA96] S Ananthakrishnan, Rebecca Teders, and Ken Alder. Role of estimation in real-time contact dynamics enhancement of space station engineering facility. *IEEE Robotics & Automation Magazine*, 3(3):20–28, 1996.

[BAMR16] R. Balachandran, J. Artigas, U. Mehmood, and J. Ryu. Performance comparison of wave variable transformation and time domain passivity approaches for time-delayed teleoperation: Preliminary results. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 410–417, Oct 2016.

[BC98a] J.M. Brown and J.E. Colgate. Minimum mass for haptic display simulations. In *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, pages 249–256, 1998.

[BC98b] J.Michael Brown and J. Edward Colgate. Minimum mass for haptic display simulation. In *Dynamic System and Control Division ASME*, pages 249–256, 1998.

[BC16] S. Blanes and F. Casas. *A Concise Introduction to Geometric Numerical Integration*. CRC Press, 2016.

[BM11] T. Boge and O. Ma. Using advanced industrial robotics for spacecraft rendezvous and docking simulation. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, May 2011.

[Bra98] Raymond M Brach. Formulation of rigid body impact problems using generalized coefficients. *International journal of engineering science*, 36(1):61–71, 1998.

[BU07] Peter Betsch and Stefan Uhlar. Energy-momentum conserving integration of multibody dynamics. *Multibody System Dynamics*, 17(4):243–289, 2007.

[BV04] Stephen Boyd and Lieven Vandenberghe, editors. *Convex Optimization*. Cambrige University Press, 2004.

[BWMT10] Toralf Boge, Tilman Wimmer, Ou Ma, and Tristan Tzschichholz. EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions. *Guidance, Navigation, and Control and Co-located Conferences*, aug 2010.

[CA00] C. R. Carignan and D. L. Akin. The reaction stabilization of on-orbit robots. *IEEE Control Systems Magazine*, 20(6):19–33, Dec 2000.

[CBS+18] P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaar, H. Strauch, AM. Giordano, M. De Stefano, C. Ott, M. Rainer, D. Henry, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, and J. Gil-Fernandez. Methods and outcomes of the comrade project - design of robust combined control for robotic spacecraft and manipulator in servicing missions: Comparison between hinf and nonlinear lyapunov-based approaches. In *69th International Astronautical Congress*, Sept 2018.

[CCYH07] Tongli Chang, Dacheng Cong, Zhengmao Ye, and Junwei Han. Time problems in hil simulation for on-orbit docking and compensation. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 841–846. IEEE, 2007.

[DDC+95] Steven Dubowsky, William Durfee, Thomas Corrigan, Andrew Kuklinski, and Uwe Müller. -a laboratory test bed for space robotics: The ves ii. In *Intelligent Robots and Systems*, pages 463–475. Elsevier, 1995.

[DNBS06] N. Diolaiti, G. Niemeyer, F. Barbagli, and J.K. Salisbury. Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects. *Robotics, IEEE Transactions on*, 22(2):256–268, April 2006.

[DP93] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *Robotics and Automation, IEEE Transactions on*, 9(5):531 –543, oct 1993.

[DS] DLR and SpaceTech. Deos functional specification. January 2009 https://spacetech-i.com/products/missions-satellites/deos.

[DSA17] M. De Stefano and J. Artigas. The dlr on-orbit servicing simulator: Reproducing free-floating dynamics with robotic facilities. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Gravity Offload Testbed for Space Robotic Mission Simulation*, Sept 2017.

[DSAG+15] M. De Stefano, J. Artigas, A. Giordano, R. Lampariello, and A. Albu-Schaeffer. On- ground experimental verification of a torque controlled free-floating robot. In *13th Symposium on Advanced Space Technologies in Robotics and Automation 2015 (ASTRA)*, May 2015.

[DSARAS15] M. De Stefano, J. Artigas, W. Rackl, and A. Albu-Schaeffer. Passivity of virtual free-floating dynamics rendered on robotic facilities. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 781–788, May 2015.

[DSAS16] M. De Stefano, J. Artigas, and C. Secchi. An optimized passivity-based method for simulating satellite dynamics on a position controlled robot in presence of latencies. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5419–5426, Oct 2016.

[DSAS17] M. De Stefano, J. Artigas, and C. Secchi. A passive integration strategy for rendering rotational rigid-body dynamics on a robotic simulator. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2806–2812, Sept 2017.

[DSBAS17] M. De Stefano, R. Balachandran, J. Artigas, and C. Secchi. Reproducing physical dynamics with hardware-in-the-loop simulators: A passive and explicit discrete integrator. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5899–5906, May 2017.

[DSBG+18] M. De Stefano, R. Balachandran, A. Giordano, C. Ott, and C. Secchi. An energy-based approach for the multi-rate control of a manipulator on an actuated base. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, pages 1072–1077, May 2018.

[DSBS18] M. De Stefano, Balachandran, and C. Secchi. A passivity-based approach for simulating satellite dynamics with robots: Time delay and discretization effects. In *submitted to IEEE Transactions on Robotics (T-RO)*, 2018.

[DSMB+19] M. De Stefano, H. Mishra, R. Balachandran, R. Lampariello, C. Ott, and C. Secchi. Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy. *IEEE Robotics and Automation Letters*, 4(2):1319–1326, April 2019.

[EKS08] Alex Ellery, Joerg Kreisel, and Bernd Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5-6):632–648, 2008.

[ESA15] e.deorbit implementation plan. ESA – Report ESA-TEC-SC-TN-2015-007, Clean Space, European Space Research and Technology Centre, Dec. 2015.

[FAMPU14] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, 68:1 – 26, 2014.

[FDDR17] N. Fazeli, E. Donlon, E. Drumwright, and A. Rodriguez. Empirical evaluation of common contact models for planar impact. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3418–3425, May 2017.

[Fea07] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[FPEN09] Gene Franklin, J. D. Powell, and Abbas Emami-Naeini, editors. *Feedback Control of Dynamic Systems*. Prentice Hall, 2009.

[FPM+15] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi. An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31(5):1073–1088, Oct 2015.

[FPW98] G. F. Franklin, J. D. Powell, and M. Workman, editors. *Digital Control of Dynamic Systems*, chapter Effect of Sampling, pages 57–66. Addison Wesley Longman, Inc., 1998.

[FSM+11] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli. Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE Transactions on Robotics*, 27(4):741–756, Aug 2011.

[FW67] Peter L Falb and William A Wolovich. Decoupling in the design and synthesis of multivariable control systems. 1967.

[GGDS+16] A. M. Giordano, G. Garofalo, M. De Stefano, C. Ott, and A. Albu-Schäffer. Dynamics and control of a free-floating space robot in presence of nonzero linear and angular momenta. In *IEEE 55th Conference on Decision and Control*, pages 7527–7534, Dec 2016.

[GS02] G Gilardi and I Sharf. Literature survey of contact dynamics modelling. *Mechanism and machine theory*, 37(10):1213–1239, 2002.

[HBDH93] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-based space robotics-rotex and its telerobotic features. *IEEE Transactions on Robotics and Automation*, 9(5):649–663, Oct 1993.

[HCC+19] D. Henry, J. Cieslak, P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaard, H. Strauch, A. Giordano, M. De Stefano, C. Ott, M. J. Reiner, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, and J. Gil. Model-based fault diagnosis and tolerant control: the esa's e.deorbit mission. In *Accepted to 2019 European Control Conference (ECC)*, 2019.

[HHK+10] K. Hertkorn, T. Hulin, P. Kremer, C. Preusche, and G. Hirzinger. Time domain passivity control for multi-degree of freedom haptic devices with time delay. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1313–1319, May 2010.

[HJBB04] Richard T Howard, Albert S Johnston, Thomas C Bryan, and Michael L Book. Advanced video guidance sensor (avgs) development testing. In *Spaceborne Sensors*, volume 5418, pages 50–61. International Society for Optics and Photonics, 2004.

[HR01] B. Hannaford and Jee-Hwan Ryu. Time domain passivity control of haptic interfaces. In *Robotics and Automation ICRA, 2001 IEEE International Conference on*, volume 2, pages 1863–1869, 2001.

[HR02] B. Hannaford and J.-H. Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, Feb. 2002.

[HS06] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: an historical survey. In *Automatica, 2006*, volume 42, pages 2035 – 2057, 2006.

[Hug04] Peter C. Hughes, editor. *Spacecraft Attitude Dynamics*. Dover Publications, Inc., Mineola, New York, 2004.

[HWL06] E. Hairer, G. Wanner, and C. Lubich. *Geometric Numerical Integration*. Computational Mathematics. Springer, 2006.

[JHea92] Marshall JE, Goerecki H, and Korytowski A et al, editors. *Time-delay systems stability and performance criteria with application.* 1st ed. London: Ellis Horwood, 1992.

[JLR+18] Steffen Jaekel, Roberto Lampariello, Wolfgang Rackl, Marco De Stefano, Nassir Oumer, Alessandro M. Giordano, Oliver Porges, Markus Pietras, Bernhard Brunner, John Ratti, Quirin Muehlbauer, Markus Thiel, Stephane Estable, Robin Biesbroek, and Alin Albu-Schaeffer. Design and operational elements of the robotic subsystem for the e.deorbit debris removal mission. *Frontiers in Robotics and AI*, 5:100, 2018.

[KBDS+18] M.J. Kim, R. Balachandran, M. De Stefano, K. Kondak, and C. Ott. Passive compliance control of aerial manipulators. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4177–4184, Oct 2018.

[KCP78] Donald J Kessler and Burton G Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978.

[KH01] Yoon Sang Kim and B. Hannaford. Some practical issues in time domain passivity control of haptic interfaces. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. (Cat. No.01CH37180)*, volume 3, pages 1744–1750 vol.3, 2001.

[Kha87] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[Kha95] Oussama Khatib. Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, 14(1):19–36, 1995.

[Kha02a] Hassan K. Khalil, editor. *Nonlinear system*. Upper Saddler River, (N.J.) Prentice Hall, 2002.

[Kha02b] H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.

[Kim99] Sun-Wook Kim. *Contact dynamics and force control of flexible multi-body systems*. PhD thesis, McGill University Libraries, 1999.

[KKK+95] Miroslav Krstic, Ioannis Kanellakopoulos, Petar V Kokotovic, et al. *Nonlinear and adaptive control design*, volume 222. Wiley New York, 1995.

[KR04] Jong-Phil Kim and Jeha Ryu. Energy bounding algorithm based on passivity theorem for stable haptic interaction control. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.*, pages 351–357, March 2004.

[LFS+17] C. Talignani Landi, F. Ferraguti, L. Sabattini, C. Secchi, M. Bonfè, and C. Fantuzzi. Variable admittance control preventing undesired oscillating behaviors in physical human-robot interaction. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3611–3616, Sept 2017.

[LH08] D. Lee and K. Huang. On passive non-iterative varying-step numerical integration of mechanical systems for haptic rendering. In *Proceedings of the ASME Dynamic Systems and Control Conference*, Ann Arbor (MI), USA, Oct. 2008.

[LH13] R. Lampariello and G. Hirzinger. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5652–5659, Nov 2013.

[LHK16] M. Laiacker, F. Huber, and K. Kondak. High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1631–1636, Oct 2016.

[Lio11] J.-C. Liou. An active debris removal parametric study for leo environment remediation. *Advances in Space Research*, 47(11):1865 – 1876, 2011.

[LLZ87] Richard W. Longman, Robert E. Lindbergt, and Michael F. Zedd. Satellite-mounted robot manipulators — new kinematics and reaction moment compensation. *The International Journal of Robotics Research*, 6(3):87–103, 1987.

[LMO+18] R. Lampariello, H. Mishra, N. Oumer, P. Schmidt, M. De Stefano, and A. Albu-Schäffer. Tracking control for the grasping of a tumbling satellite with a free-floating robot. *IEEE Robotics and Automation Letters*, 3(4):3638–3645, Oct 2018.

[LN97] A. A. J. Lefeber and H. Nijmeijer. Globally bounded tracking controllers for rigid robot systems. In *1997 European Control Conference (ECC)*, pages 1714–1719, July 1997.

[LPPT05] A. Loria, E. Panteley, D. Popovic, and A. R. Teel. A nested matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems. *IEEE Transactions on Automatic Control*, 50(2):183–198, Feb 2005.

[Ma95] O. Ma. Contact dynamics modelling for the simulation of the space station manipulators handling payloads. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1252–1258 vol.2, May 1995.

[MBC+07] Carlo Menon, S. Busolo, S. Cocuzza, A. Aboudan, A. Bulgarelli, C. Bettanini, M. Marchesi, and F. Angrilli. Issues and solutions for testing free-flying robots. *Acta Astronautica*, 60(12):957 – 965, 2007.

[MBR+97] Ou Ma, Kerman Buhariwala, Neil Roger, John MacLean, and Robert Carr. Mdsf—a generic development and simulation facility for flexible, complex robotic systems. *Robotica*, 15(1):49–62, 1997.

[MD12] Zoran Milenkovic and Christopher D'Souza. *The Space Operations Simulation Center (SOSC) and Closed-loop Hardware Testing for Orion Rendezvous System Design*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2012. doi:10.2514/6.2012-5034.

[MDSGO18] H. Mishra, M. De Stefano, A. Giordano, and C. Ott. Tracking control with robotic systems for a moving target: A vector lyapunov function approach. *IFAC-PapersOnLine*, 51(22):471 – 478, 2018. 12th IFAC Symposium on Robot Control SYROCO 2018.

[MDSGO19] H. Mishra, M. De Stefano, AM. Giordano, and C. Ott. A nonlinear observer for free-floating target motion using only position measurements. *accepted to the American Control Conference (ACC)*, 2019.

[MMA89] Y. Masutani, F. Miyazaki, and S. Arimoto. Sensory feedback control for space manipulators. In *Robotics and Automation, IEEE International Conference on*, pages 1346–1351 vol.3, May 1989.

[MS11] K. Modin and G. Söderlind. Geometric integration of hamiltonian systems perturbed by rayleigh damping. *Bit Numer Math*, 2011.

[MW07] Ou Ma and Jiegao Wang. Model order reduction for impact-contact dynamics simulations of flexible manipulators. *Robotica*, 25(4):397–407, 2007.

[MWSL03] Ou Ma, Jiegao Wang, Misra Sarthak, and Michael Liu. Validation plan for spdm task verification facility. In *i-SAIRAS symposium on Artificial Intelligence, Robotics and Automation in Space*, Japan 2003.

[MWX+18] Fangli Mou, Shuang Wu, Xuan Xiao, Tao Zhang, and Ou Ma. Control of a space manipulator capturing a rotating object in the three-dimensional space. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 763–768. IEEE, 2018.

[Nie96] G. Niemeyer. Using wave variables in time delayed force reflecting teleoperation. In *Ph.D. dissertation, Dept. Aeronaut. Astronaut., MIT, Cambridge, MA*, 1996.

[Nie04] J.-J. Niemeyer, G. ans Slotine. Telemanipulation with Time Delays. *International Journal of Robotics Research*, 23(9):873–890, September 2004.

[NP17] Kostas Nanos and Evangelos G. Papadopoulos. On the dynamics and control of free-floating space manipulator systems in the presence of angular momentum. *Frontiers in Robotics and AI*, 4:26, 2017.

[NS97] G. Niemeyer and J. . E. Slotine. Using wave variables for system analysis and robot control. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1619–1625 vol.2, April 1997.

[NS98] G. Niemeyer and J. . E. Slotine. Towards force-reflecting teleoperation over the internet. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 3, pages 1909–1915 vol.3, May 1998.

[NUY10] H. Nakanishi, N. Uyama, and K. Yoshida. Virtual mass of impedance system for free-flying target capture. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4101–4106, Oct 2010.

[NY06] Hiroki Nakanishi and Kazuya Yoshida. Impedance Control for Free-flying Space Robots -Basic Equations and Applications-. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, Oct 2006.

[OAP11] C. Ott, Jordi Artigas, and C. Preusche. Subspace-oriented energy distribution for the time domain passivity approach. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 665–671, Sept 2011.

[OB08] S. Ober-Bloebaum. Discrete mechanics and optimal control. *Ph.D. thesis, University of Paderborn, Germany*, 2008.

[OHF+18] Anibal Ollero, Guillermo Heredia, Antonio Franchi, Gianluca Antonelli, Konstantin Kondak, Alberto Sanfeliu, Antidio Viguria, J Ramiro Martinez-De Dios, Francesco Pierri, Juan Cortés, et al. The aeroarms project: Aerial robots with advanced manipulation capabilities for inspection and maintenance. *IEEE Robotics and Automation Magazine*, 2018.

[OKU10] Kohei Osaki, Atsushi Konno, and Masaru Uchiyama. Delay time compensation for a hybrid simulator. *Advanced Robotics*, 24(8-9):1081–1098, 2010.

[OKY96] M. Oda, K. Kibe, and F. Yamagata. Ets-vii, space robot in-orbit experiment satellite. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 739–744 vol.1, April 1996.

[ON08] C. Ott and Yoshihiko Nakamura. Employing wave variables for coordinated control of robots with distributed control architecture. In *2008 IEEE International Conference on Robotics and Automation*, pages 575–582, May 2008.

[OPNSR98] R. Ortega, J.A.L. Perez, P.J. Nicklasson, and H. Sira-Ramirez. *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*. Communications and Control Engineering. Springer London, 1998.

[OS88] R. Ortega and M. W. Spong. Adaptive motion control of rigid robots: a tutorial. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 1575–1584 vol.2, Dec 1988.

[PD90] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1102–1108 vol.2, May 1990.

[PD91] E. Papadopoulos and S. Dubowsky. Coordinated manipulator/spacecraft motion control for space robotic systems. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1696–1701 vol.2, Apr 1991.

[PD93] E. Papadopoulos and D. Dubowsky. Dynamic singularities in free-floating space manipulators. In *Journal of Dynamic Systems, Measurement, and Control*, pages 115(1):44–52., 1993.

[PDG+] J. Paul, A. Dettmann, B. Girault, J. Hilljegerdes, F. Kirchner, I. Ahrns, and J. Sommer. INVERITAS: a facility for hardware-in-the-loop long distance movement simulation for rendezvous and capture of satellites and other autonomous objects. *Acta Astronautica*, 116:1 – 24.

[PHRH03] C. Preusche, G. Hirzinger, J.-H. Ryu, and B. Hannaford. Time domain passivity control for 6 degrees of freedom haptic displays. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2944–2949 vol.3, Oct 2003.

[PP88] Brad Paden and Ravi Panja. Globally asymptotically stable 'pd+' controller for robot manipulators. *International Journal of Control*, 47(6):1697–1712, 1988.

[PSCR13] O. Porges, T. Stouraitis, C. Corst, and MA. Roa. Reachability and capability analysis for manipulation tasks. In *ROBOT2013: First Iberian Robotics Conference*, pages 703–718, 2013.

[PSL18] P. R. Pérez, M. De Stefano, and R. Lampariello. Velocity matching compliant control for a space robot during capture of a free-floating target. In *2018 IEEE Aerospace Conference*, pages 1–9, March 2018.

[QZG+16] Chenkun Qi, Xianchao Zhao, Feng Gao, Anye Ren, and Qiao Sun. Contact stiffness and damping identification for hardware-in-the-loop contact simulator with measurement delay compensation. *Acta Astronautica*, 123(C):171–180, 2016.

[Ray91] M Brach Raymond. Mechanical impact dynamics-rigid body collisions. *A wiley-Interscience publication, Wiley, Cop*, 1991.

[RFS07] Marcello Romano, David A Friedman, and Tracy J Shay. Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target. *Journal of Spacecraft and Rockets*, 44(1):164–173, 2007.

[RHPH03] Jee-Hwan Ryu, Blake Hannaford, Carsten Preusche, and Gerd Hirzinger. Time domain passivity control with reference energy behavior. In *IROS*, pages 2932–2937. IEEE, 2003.

[RK92] Jeff Russkow and Oussama Khatib. A new control structure for free-flying space robots. In *i-SAIRAS symposium on Artificial Intelligence, Robotics and Automation in Space*, Toulouse 1992.

[RKH04] Jee-Hwan Ryu, Dong-Soo Kwon, and Blake Hannaford. Stability guaranteed control: Time domain passivity approach. *IEEE Transactions on Control Systems Technology*, 12(6):860–868, 2004.

[RMD82] Miller R., Minsky M., and Smith D. Space applications of automation, robotics and machine intelligence systems (aramis), volume 1: executive summary report nasa-cr 162079. 1982.

[RMNL11] P Rank, Q Mühlbauer, W Naumann, and K Landzettel. The deos automation and robotics payload. In *Symp. on Advanced Space Technologies in Robotics and Automation, ASTRA, the Netherlands*, 2011.

[SA90] John R Spofford and David L Akin. Redundancy control of a free-flying telerobot. *Journal of Guidance, Control, and Dynamics*, 13(3):515–523, 1990.

[SA01] Brook R Sullivan and David L Akin. A survey of serviceable spacecraft failures. In *AIAA Space 2001- Conference and Exposition, Albuquerque, NM*, 2001.

[SADS+15] P. Schmidt, J. Artigas, M. De Stefano, R. Balachandran, and C. Ott. Increasing the performance of torque-based visual servoing by applying time domain passivity. *IFAC-PapersOnLine*, 48(19):13 – 18, 2015. 11th IFAC Symposium on Robot Control SYROCO 2015.

[SAF+09] M. Satler, C. A. Avizzano, A. Frisoli, P. Tripicchio, and M. Bergamasco. Bilateral teleoperation under time-varying delay using wave variables. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4596–4602, Oct 2009.

[SBS+10] Florian Sellmaier, Toralf Boge, Jörn Spurmann, Sylvain Gully, Thomas Rupp, and Felix Huber. On-orbit servicing missions: Challenges and solutions for spacecraft operations. In *American Institute of Aeronautics and Astronautics, Inc.. SpaceOps 2010 Conference, 25,Huntsville, Alabama, USA*, 2010.

[SFF16] C. Secchi, F. Ferraguti, and C. Fantuzzi. Catching the wave: A transparency oriented wave based teleoperation architecture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2422–2427, May 2016.

[SHH+12] K Sharma, S Haddadin, J Heindl, T Bellmann, S Parusel, T Rokahr, S Minning, and G Hirzinger. Serial kinematics based motion simulator-evaluation of safety of the passenger. In *The International Conference on the Safety of Industrial Automated Systems (SIAS)*, 2012.

[SIT+91] Haruhiko Shimoji, Masao Inoue, Kazuo Tsuchiya, Keiken Niomiya, Ichiro Nakatani, and JUN'ICHIRO Kawaguchi. Simulation system for a space robot using six-axis servos. *Advanced robotics*, 6(2):179–196, 1991.

[SK97] Victor Santibáñez and Rafael Kelly. Strict lyapunov functions for control of robot manipulators. *Automatica*, 33(4):675 – 682, 1997.

[SK08] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

[SK16] B. Siciliano and O. Khatib, editors. *Handbook of Robotics*, chapter Force Control, pages 195–220. Springer, 2016.

[SKK+08] Changhoon Seo, Jaeha Kim, Jong-Phil Kim, Joo Hong Yoon, and Jeha Ryu. Stable bilateral teleoperation using the energy-bounding algorithm: Basic idea and feasibility tests. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 335–340, July 2008.

[SKKR09] C. Seo, J. Kim, J. Kim, and J. Ryu. Energy-bounding algorithm for stable haptic interaction and bilateral teleoperation. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 617–618, March 2009.

[SL87] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987.

[SL91] Jean-Jacques Slotine and Weiping Li, editors. *Applied nonlinear control*. Englewood Cliffs (N.J.) Prentice Hall, 1991.

[SMO99] H. G. Sage, M. F. De Mathelin, and E. Ostertag. Robust control of robot manipulators: A survey. *International Journal of Control*, 72(16):1498–1522, 1999.

[SPH03] Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall. Historical review of air-bearing spacecraft simulators. In *Journal of Guidance, Control and Dynamics*, pages AAS 03–125, 2003.

[SSF06] C. Secchi, S. Stramigioli, and C. Fantuzzi. Position drift compensation in port-hamiltonian based telemanipulation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4211–4216, Oct 2006.

[SSF07] C. Secchi, S. Stramigioli, and C. Fantuzzi. *Control of Interactive Robotic Interfaces: a port-Hamiltonian Approach*. Springer Tracts in Advanced Robotics. Springer, 2007.

[SSvdSF05] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Transactions on Robotics*, 21(4):574–587, Aug. 2005.

[SUM+04] Hirokata Sawada, Kyoichi Ui, Makoto Mori, Hiroshi Yamamoto, Ryoichi Hayashi, Saburo Matunaga, and Yoshiaki Ohkami. Micro-gravity experiment of a space robotic arm using parabolic flight. *Advanced Robotics*, 18(3):247–267, 2004.

[SW04] James Shoemaker and Melissa Wright. Orbital express space operations architecture program. In *Spacecraft Platforms and Infrastructure*, volume 5419, pages 57–66. International Society for Optics and Photonics, 2004.

[TAE+17] J. Telaar, I. Ahrns, S. Estable, W. Rackl, M. De Stefano, R. Lampariello, N. Santos, P. Serra, Canetri M., Ankersen F., and J. Gil Fernandez. Gnc architecture for the e.deorbit mission. *7th Europena Conference for Aeronautics and Space Science (EUCASS) , July 2017*, 2017.

[TEDS+17] J. Telaar, S. Estable, M. De Stefano, W. Rackl, R. Lampariello, F. Ankersen, and J. Gil Fernandez. Coupled control of chaser platform and robot arm for the e.deorbit mission. *10th Int. ESA conference on Guidance Navigation and Control Systems (GNC), May 2017*, 2017.

[TIK+08] Ryohei Takahashi, Hiroto Ise, Atsushi Konno, Masaru Uchiyama, and Daisuke Sato. Hybrid simulation of a dual-arm space robot colliding with a floating object. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1201–1206. IEEE, 2008.

[UNNY12] N. Uyama, H. Nakanishi, K. Nagaoka, and K. Yoshida. Impedance-based contact control of a free-flying space robot with a compliant wrist for non-cooperative satellite capture. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4477–4482, Oct 2012.

[UY87] Yoji Umetani and Kazuya Yoshida. Continuous path control of space manipulators mounted on {OMV}. *Acta Astronautica*, 15(12):981 – 986, 1987.

[UY89] Y. Umetani and K. Yoshida. Resolved motion rate control of space manipulators with generalized jacobian matrix. *Robotics and Automation, IEEE Transactions on*, 5(3):303–314, Jun 1989.

[vdS00] A.J. van der Schaft. *L_2-Gain and Passivity Techniques in Nonlinear Control*. Communication and Control Engineering. Springer Verlag, 2000.

[WFH11] X. Wei, S. Fuchun, and L. Huaping. Design and development of a ground experiment system with free-flying space robot. In *2011 6th IEEE Conference on Industrial Electronics and Applications*, pages 2101–2106, June 2011.

[WW93] Liang-Boon Wee and Michael W Walker. On the dynamics of contact between space robots and configuration control for impact minimization. *IEEE Transactions on Robotics and Automation*, 9(5):581–591, 1993.

[XLX11] Wenfu Xu, Bin Liang, and Yangsheng Xu. Survey of modeling, planning, and ground verification of space robotic systems. *Acta Astronautica*, 68(11-12):1629–1649, 2011.

[XS91] Y. Xu and H.Y. Shum. Dynamic control of a space robot with no thrust jets controlled base. *No. CMU-RI-TR-33. Pittsburgh: The Robotics Institutem Carnegie Mellon University*, 1991.

[Yas14] Andrey Yaskevich. Real time math simulation of contact interaction during spacecraft docking and berthing. *J. Mech. Eng. Autom.*, 4:1–15, 2014.

[Yos94] K. Yoshida. Practical coordination control between satellite attitude and manipulator reaction dynamics based on computed momentum concept. In *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94.*, volume 3, pages 1578 –1585, sep 1994.

[Yos03] Kazuya Yoshida. Engineering test satellite vii flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit. *The International Journal of Robotics Research*, 22(5):321–335, 2003.

[YTY93] Yasuyoshi Yokokohji, Takeshi Toyoshima, and Tsuneo Yoshikawa. Efficient computational algorithms for trajectory control of free-flying space robots with multiple arms. *IEEE Transactions on Robotics and Automation*, 9(5):571–580, 1993.

[ZBC13] Melak M Zebenay, Toralf Boge, and Daniel Choukroun. Modeling and stability analysis of a hybrid docking simulator. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4520, 2013.

[ZBC14] M. Zebenaya, Toralf Boge, and D. Choukroun. Modeling, stability analysis, and testing of a hybrid docking simulator. *CoRR*, abs/1409.0562, 2014.

[ZLBC12] M. Zebenay, R. Lampariello, T. Boge, and D. Choukroun. A new contact dynamics model tool for hardware-in-the-loop docking simulation. In *11th i-SAIRAS 2012, Turin, Italy*, 2012.

[ZLL14] Lanhao Zhao, Zhi Liu, and Tongchun Li. Mixed finite element method for static and dynamic contact problems with friction and initial gaps. *Mathematical Problems in Engineering*, 2014, 2014.

[ZRS06a] A. Zavala-Rio and V. Santibanez. Simple extensions of the pd-with-gravity-compensation control law for robot manipulators with bounded inputs. *IEEE Transactions on Control Systems Technology*, 14(5):958–965, Sep. 2006.

[ZRS06b] A. Zavala-Río and V. Santibáñez. A natural extension of the pd-with-desired-gravity-compensation control law for robot manipulators with input saturations. *IFAC Proceedings Volumes*, 39(15):7 – 12, 2006. 8th IFAC Symposium on Robot Control.