POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Robust attitude control for NASA Astrobee robots operating in the ISS

Supervisors

Dr. Elisa CAPELLO

Dr. Hyeongjun PARK

Candidate

Dario RUGGIERO

July 2020

Abstract

Free-flying robots have been recently developed to operate on-board the International Space Station (ISS) as semi-autonomous robotic assistants. Free-flyer robots can be designed as modular base for integration of a wide range of hardware and software as monitoring and maintenance tools of ISS systems. Thus, the free-flyers are the ideal platform for manual observation of the ISS by ground control, autonomous sensor readings and surveying of ISS conditions, and human-robot interaction during long duration human missions. The NASA Astrobee project aims to develop a highly capable robot that can operate for long periods of time without crew supervision or operation.

The goal of this research is to realize a model and a robust attitude control system for the NASA Astrobee, which is a free-flyer equipped with a 3 DoF manipulator. Two subsystems, the Astrobee's main body and the manipulator, have been modelled and controlled independently, considering the torques and the disturbances due to the manipulator motion.

For the attitude control and stabilization of the main body, the twisting sliding mode controller (TW-SMC) and the back-stepping controller are proposed, while for the manipulator motion a first order sliding mode controller is proposed. Robustness and performance are analyzed to show the effectiveness of the proposed control system. First, the TW-SMC is designed to achieve a trade-off among control law flexibility, robustness and precision attitude control. Among robust control strategies, SMC are characterizes as low complexity, low computational and low cost control methods. Including the chattering attenuation introduced with the second order SMC and the hyperbolic tangent in the reaching law, the TW-SMC is a suitable approach for the Astrobee's main body attitude control.

The TW-SMC is compared with a back-stepping approach, which consist in an adaptive controller based on Lyapunov functions, that use an iterative algorithm for the control design. The controller outputs are torque taking into account attitude error, the arm torque and motion, and the inertia variation, leading the Astrobee to track the desired attitude.

Controllers' performance is evaluated and analyzed both in MATLAB/Simulink enviroment and with the NASA's ROS/Gazebo Astrobee simulator. Simulink-ROS combined simulations have been run to test the arm motion stabilization, attitude changing and their combination. To show the robustness of the proposed control methodologies, simulations with variable body mass, links masses and end-effector mass have been run to test the controllers in off-design conditions.

Summary

Questo progetto si colloca nel recente scenario di ricerca riguardante la progettazione di robot autonomi capaci di assistere gli astronauti all'interno della Stazione Spaziale Internazionale (ISS). Il progetto Astrobee, condotto da NASA, asprira a progettare un robot dotato di braccio meccanico, capace di operare autonomamente all'interno della stazione spaziale per lunghi periodi di tempo.
L'obiettivo di questa ricerca è la realizzazione del modello matematico dell'Astrobee e la progettazione di un controllore d'assetto robusto, capace di stabilizzare il volo durante l'utilizzo del braccio robotico, cosiderando le coppie e i disturbi introdotti dal suo movimento.
I due sistemi, corpo principale e braccio robotico, sono stati modellati e controllati separatamente. Il *twisting sliding mode controller* (TW-SMC) e il *backstepping controller* vengono proposti come controllori d'assetto, mentre per il controllo del braccio robotico è stato utilizzato uno *sliding mode controller* (SMC) del primo ordine.
Prima il TW-SMC è stato progettato raggiungendo un compromesso tra flessibilità, robustezza e precisione del controllo. Generalmente i SMC sono caratterizzati da bassa complessità, basso costo computazionale ed elevata robustezza, rendendo il TW-SMC un possibile approccio per il controllo d'assetto dell'Astrobee.
Il TW-SMC è stato confrontato con il *backstepping controller*, che consiste in un controllore adattivo, che sfrutta un algoritmo iterativo per la definizione della legge di controllo. Questo controllore richiede un costo computazionale più alto, ma il suo output sarà una coppia che considera l'errore d'assetto, la variazione d'inerzia introdotta dal movimento del braccio e la coppia di disturbo, permettendo all'Astrobee di mantenere l'assetto desiderato.
Le prestazioni dei controllori sono state valutate ed analizzate in ambiente MATLAB/Simulink e con il simulatore NASA dedicato all'Astrobee, progettato in abiente ROS/Gazebo. Sono state quindi eseguite simulatzioni combinate Simulink-ROS per valutare il controllo d'assetto e la stabilizazzione del braccio robotico. Per mostrare la robustezza delle strategie di controllo proposte, simulazioni con massa variabile di corpo e braccio robotico sono state effettuate per valutare i controllori in condizioni fuori progetto.

Acknowledgements

Reaching this milestone, I feel compelled to thank some people who contributed to the achievement of this important goal. First of all, I really thank my supervisors Dr.Capello and Dr.Park for their help in the technical conduction of my thesis, and most especially for their constant presence and friendly support during my period abroad, in a so far place and during this difficult times.

I want to thank my family for their emotional and economical support, especially my grand-mother Luigina, who did everything to make me feel constantly her presence and to keep me safe.

Friends are special, and I want to thank the ones who never left me, even at more than 8000 km of distance. I would like to say lots of thing to you, Andrea, but it's easier to say that you are more like a brother than a friend, in fact, more than ten years ago I met you, and I have no doubt about you to be one of my best friends right now. I am going to betray the first rule, but the *PANZclub* deserves to be mentioned, not just for the funny name, but more for the importance of what the two of you, Emanuele and Valeria, means to me: not simple friends but *"panz-partners"* for life.

Silvia, I should thank you for too many things that I could write a book about it, but moreover than your support, presence and constantly making feel me important and happy, I want to thank for making me feel conscious of who I really am.

In the last, I want to thank all the people, colleagues, friends, and relatives who take part in this great adventure.

To mom †

Table of Contents

List of Figures VIII

1 Introduction 1

1.1 Overview 3

2 Astrobee system description and mathematical model 4

2.1 Astrobee's dynamics model 6

3 Control strategies 10

3.1 Sliding mode approach 10

3.2 Backstepping approach 13

4 Astrobee's controllers design 16

4.1 Manipulator motion control design 17

4.2 Attitude control design 17

5 Simulation results 22

5.1 Manipulator control 23

5.2 Torque-free simulation 27

5.3 Deploying/stowing manipulator stabilization 28

5.4 Full manipulator motion stabilization 32

5.5 Controller robustness test 37

6 NASA's Astrobee simulator 47

6.1 Simulink implementation 47

6.2 Simulations results 49

7 Conclusions and future work 61

List of Figures

1.1 IVA Freeflyers [7] 2

2.1 NASA Astrobee design and component breakdown [7] 4
2.2 Manipulator model 7

3.1 Sliding mode convergence [14] 11
3.2 Second order sliding mode trajectory [17] 13
3.3 Generic dynamic system 14

4.1 Astrobee's Simulink implementation 16
4.2 Second-order sliding mode parameters' bounds 19

5.1 MATLAB/Simulink Astrobee 3D model 22
5.2 Manipulator motion references 24
5.3 Opening/closing manipulator motion 25
5.4 Opening/rotating/closing manipulator motion 26
5.5 Torque-free Euler angles 27
5.6 Euler angles in deploying/stowing stabilization 29
5.7 Angular velocity in deploying/stowing stabilization 30
5.8 Command torque in deploying/stowing stabilization 31
5.9 Euler angles in full motion stabilization 33
5.10 Steady-state error 34
5.11 Angular velocity in full motion stabilization 35
5.12 Command torque in full motion stabilization 36
5.13 Attitude changing with stowed arm 38
5.14 Attitude changing with deployed arm 39
5.15 Attitude changing with gripped object 40
5.16 Variable main body mass and inertia 42
5.17 Variable manipulator links mass 44
5.18 Variable end-effector mass 46

6.1 Euler angles - Attitude changing 50

6.2 Angular velocity - Attitude changing 51
6.3 Command torque - Attitude changing 52
6.4 Euler angles - Manipulator motion stabilization 54
6.5 Angular velocity - Manipulator motion stabilization 55
6.6 Command torque - Manipulator motion stabilization 56
6.7 Euler angles - Combined motion 58
6.8 Angular velocity - Combined motion 59
6.9 Command torque - Combined motion 60

Chapter 1

Introduction

The goal of this thesis is to design the attitude controller for a free-flyer robot operating in the International Space Station (ISS). In particular, the NASA Astrobee project has been considered, which aim to develop a highly capable robotic system, equipped with a robotic arm, able to operate for long periods of time without crew supervision. The Astrobee navigation and control system is based on trajectory execution that are generated by ground station control or by the Astrobee's flight software. A Proportional-Integral-Derivative (PID) controller calculate forces/-torques commands designed to overcome the error between the actual state and the desired trajectory [1]. This control strategy leads to some challenges: first, since the propulsion system has been projected as fan-based, it is non-trivial to calibrate the propulsion physics model needed by the force allocator; second, the Astrobee configuration changes, such as moving arm and adding or removing payloads, will induce inertial property changes and new disturbance; third, during activities like perching and docking, the controller will need to accommodate contact forces.

The coordinated control of attitude and manipulator is a topic discussed in several researches about aerial and space manipulators, where the disturbances introduced by the manipulator usage are reported, and robust control strategies are proposed to stabilize the system [2, 3]. As well, the Astrobee is subjected to variable disturbances induced by the manipulator motion, and it is why the realization of an attitude controller able to stabilize the disturbances induced by the Astrobee configuration changing and the manipulator motion is needed. Since both the main body and the manipulator are non linear systems in terms of dynamics and kinematics, the attitude controller should ensure high accuracy and robustness against external disturbances and parameters variation [4]. The twisting sliding mode controller (TW-SMC) has been proposed for precise attitude control. Among robust control strategies, sliding mode controllers (SMC) are characterizes as low complexity, low computational and low cost control methods, making the TW-SMC

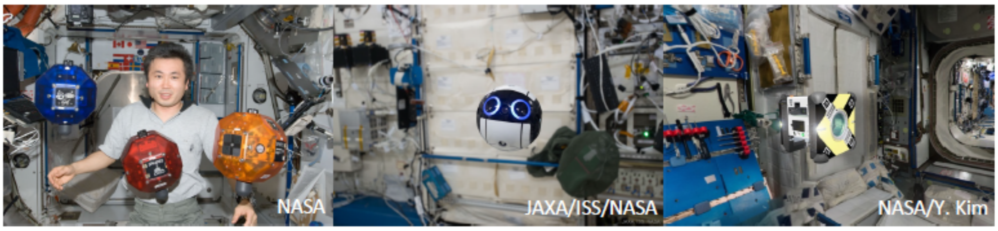


Figure 1.1: IVA Freeflyers [7]

a suitable approach for the Astrobee's main body attitude control [5].
The TW-SMC has been compared with a backstepping approach, which consist in an adaptive controller based on Lyapunov functions. The controller outputs are torques taking into account attitude error, disturbance torque and inertia variation induced by the arm motion, despite an higher computational cost than SMC [6].

Free-flying robots have been recently developed to operate on board the ISS as semi autonomous robotic assistants, offering the opportunity to complement the astronauts for monitoring and maintenance works.
Recently several free-flyer projects have been developed to perform intravehicular-activity (IVA) on the ISS. The *Synchronized Position Hold Engage and Reorient Experimental Satellites* (SPHERES) project has been really successful as micro-gravity research platform. Astrobee project is growing up on the legacy of SPHERES, with the objective of improving the characteristics that limit its usefulness for ISS operations. In fact, SPHERES use CO_2 for propulsion, so the crew is needed to refill CO_2 tanks and change batteries, and for safety consideration SPHERES are not allowed to work without crew supervision [8].
Int-Ball and CIMON are further examples of free-flyers already employed in the international space station. Int-Ball, developed by *Japan Aerospace Exploration Agency* (JAXA), is a free flying video-camera equipped with micro-fans and reaction wheels to freely move in the ISS environment. It localize itself by using markers, ultrasonic distance sensors and an on-board camera, so it is more autonomous than SPHERES, but it still needs crew support for batteries recharging [7]. CIMON is multi-camera free-flier realized by Airbus, able to assist astronauts by displaying experiment instructions, maintenance procedures, and other documents and media [7].

The NASA Astrobee project aim to develop a highly capable robotic system to help astronauts during routine duties, and letting them focus on operations that only humans can do. It consist in a cubed-shaped robot, equipped with a 3 degrees of freedom (DoF) manipulator, able to perform IVA. Thus, it is the ideal platform for manual observation of the ISS by ground control, autonomous sensor reading

and surveying of ISS conditions, and human-robot interaction during long duration human missions.
The Astrobee system has been primary projected to be a platform for guest science, to provide remotely operated mobile camera and to perform as mobile sensor inside the ISS. Its technologies, including propulsion, navigation, co-located human interaction and a payload interface, make the Astrobee the base to study concepts of operations for future deep space missions, including robotic caretaking of manned human spacecraft [7].

1.1 Overview

The goal of this research is to realize a model and a robust attitude control system for NASA Astrobee robots. Specifically, TW-SMC and backstepping controller have been implemented, analyzed, and compared for precise attitude control and manipulator motion stabilization.

- In chapter 2, the Astrobee system is described in all its technologies and features. Thus, all the mathematical equations to describe the system dynamics are shown and explained in detail. The kinematics is studied with quaternion formulation, while the Astrobee dynamics equation is derived from the angular momentum conservation. Moreover, it's explained the manipulator dynamics and how its motion have an effect in the attitude dynamics.

- In chapter 3, the theoretical references of sliding mode controllers and backstepping controllers are introduced, and their features and characteristics are presented.

- In chapter 4, the controllers are analyzed in detail, and the control problem of main body attitude and manipulator is discussed. First, the first order sliding mode controller for the manipulator motion is designed. Then, TW-SMC and back-stepping controller are designed for the attitude control, underlining their differences in control methodologies and implementation.

- In chapter 5, the simulations' results of TW-SMC and back-stepping controller tests are showed and analyzed. Moreover, to show the robustness of the proposed control methodologies, simulations with variable Astrobee parameters have been run to test the controllers in different condition of body, manipulator's links and end-effector masses.

- In chapter 6, the NASA's ROS/Gazebo Astrobee simulator is presented and ROS-Simulink combined simulations results are showed.

- In chapter 7, conclusions and future work are briefly discussed.

Chapter 2

Astrobee system description and mathematical model

The Astrobee system is a 32cm-on-a-side cube shaped robot able to move in each direction and around any axis. To achieve the long time operation uses without crew supervision or intervention, the Astrobee has been projected with a fan-based propulsion system, that frees it from needing any consumables except for battery charge, but it will able to replenish its charge by autonomously connecting to its docking station.

The overall design and component breakdown is showed in Figure 2.1. The propulsion system consists in two propulsion modules, which include a centrifugal

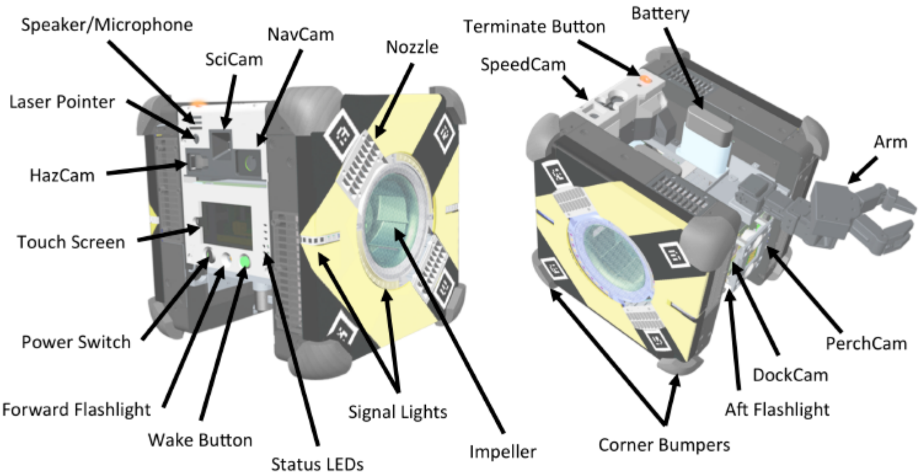


Figure 2.1: NASA Astrobee design and component breakdown [7]

fan, to pressurize the module, and nozzles on each axis to allow 6 Degrees-of-Freedom (DoF) holonomic control. Moreover, Astrobee will be equipped with a perching arm able to grip to handrails to maintain the position without wasting energy [1].
The power subsystem consists of Lithium Ion rechargeable batteries that the Astrobee will be able to autonomously recharge when docked to its dock station [9].

The guidance, navigation and control (GNC) subsystem is essentially based on visual-based navigation and IMU sensors to detect position and speed. The forward face includes the *NavCam*, a fixed focus color camera, which send images to the on-board visual navigation system, that provide situation awareness. The forward face also include *HazCam* to detect obstacles with LIDAR, and *SciCam* to stream live high-definition video. The back face is equipped with *DockCam*, to trace the docking station position during autonomous docking maneuvers, and the *PerchCam*, to detect ISS handrails for autonomously perching maneuvers [7, 1]. Moreover, to minimize collision risk, *SpeedCam*, that provides an optical estimation of velocity, infrared ranging and IMU sensors, work together to cut off over-speed levels.
To estimate its position the Astrobee combines the comparison between images obtained trough the *NavCam* with an on-board map of the ISS interior and the IMU inertial measurements [9].

The Astrobee can communicate with ground control or crew toward the ISS WiFi, accepting commands and sending telemetry data. To exchange large files, an Ethernet wired connection can be established in the dock station. Thus, to avoid data losses, the Astrobee can keep on-board logs and the user can downlink a complete copy after the activity [7].

The Astrobee is equipped with a pearching arm in its top-aft bay that allow it to grasp ISS handrails and objects [10]. The manipulator has three DoF and it is composed of two joints and a gripper. When not in use, the manipulator is stowed in the top payload bay, while the joints allow the arm deploying and motion for grasping. Its grip is not strong enough to injure crew, that can manually backdrive the gripper and arm to perch the robot on an handrail when no needed. When the arm is perched to an handrail, the Astrobee powers down the propulsion system and the joints work for the *SciCam* pointing [7, 9].

The Astrobee is designed to interact with ISS crew and flight controllers. Thus, it can be also used for human-robot interaction research. It is equipped with touch screen, speaker and microphone, signal lights, and a laser pointer to help the crew understand its state and intentions and as telepresence for flight controllers [7, 1].

The Astrobee system has been analyzed considering its uncertainties and disturbance, that's why robust attitude controllers have been considered.

2.1 Astrobee's dynamics model

The Astrobee is a cube shaped robot, equipped with a manipulator, that will influence the attitude dynamics of the main body. In fact, the manipulator motion leads to a disturbance torque and a gyroscopic torque, as well as, to the overall inertia changing. Moreover, its motion generate the shifting of the centre of mass (CoM) [11].

Considering the body reference frame centered in the CoM of the main body parallel to the Astrobee edges and oriented with the positive values of the x-axis along the arm deployed position, z-axis along the arm vertical position and y-axis to complete the Cartesian triad, as shown in Figure 2.2. For hypothesis, the CoM of the combined structure, main body and manipulator, is assumed to be fixed in the position of the CoM of the main body. Since the main body is much heavier than the manipulator, the fixed CoM assumption leads to negligible errors. In this way, the Astrobee inertia matrix can be obtained from:

$$J = J_{body} + J_{arm,0} \tag{2.1}$$

where $J_{arm,0}$ is the inertia matrix of the manipulator referred to the CoM, that is variable with the arm position, and J_{body} is the inertia matrix of the main body that is approximately constant value.

Attitude dynamics

The Astrobee's dynamics can be derived from the the angular momentum conservation [12], that can be expressed in the inertial reference frame by

$$T_i = \dot{H}_i \tag{2.2}$$

that can be rewritten in the body reference frame as

$$\dot{H}_b = T_b - \omega_{bi} \times H_b \tag{2.3}$$

The angular momentum of the combined system in the body reference-frame is determined from both main body rotation and manipulator motion.

$$H_B = J\omega + J_{arm}\omega_{arm} \tag{2.4}$$

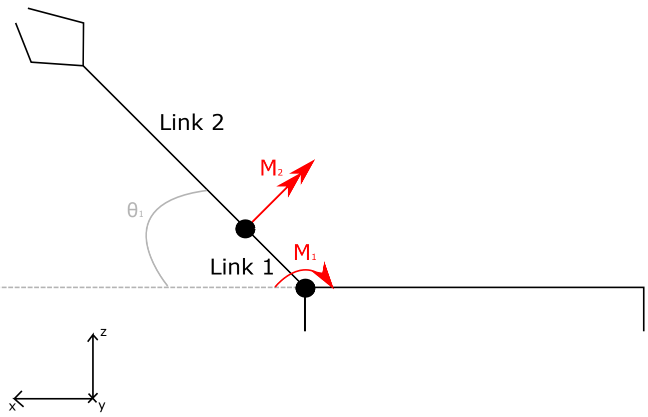


Figure 2.2: Manipulator model

where the term $J_{arm}\omega_{arm}$ represents the angular momentum of the manipulator rotation part around each joint expressed in the body reference frame. Thus, by exploiting the angular momentum in Equation (2.3), as expressed in Equation (2.4), the attitude dynamic equation can be determined

$$\dot{J}\omega + J\dot{\omega} + \dot{J}_{arm}\omega_{arm} + J_{arm}\dot{\omega}_{arm} = T - \omega \times (J\omega + J_{arm}\omega_{arm}) \tag{2.5}$$

$$\dot{\omega} = J^{-1}(T_{cmd} - T_{arm}) - J^{-1}[\omega \times (J\omega + J_{arm}\omega_{arm}) + (\dot{J}\omega + \dot{J}_{arm}\omega_{arm})] \tag{2.6}$$

where each term linked to the manipulator motion can be determined from the manipulator dynamics as expressed in the next section.
The quaternion formulation can be used to describe spatial rotations thanks to their compactness, low computational cost and their lack of singularity. Determined the angular velocity from the dynamics, the attitude can be computed with the kinematics quaternian formulation:

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & -\omega_z & \omega_y \\ \omega_y & \omega_z & 0 & -\omega_x \\ \omega_z & -\omega_y & \omega_x & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{2.7}$$

Manipulator dynamics

As shown in Figure 2.2, the Astrobee's manipulator can be modelled as a 2-link robotic arm, where the first joint provides to open/close motion and the second one to side motion. Let's consider the first joint placed in the middle of the cube edge

$$x_1 = \begin{Bmatrix} l/2 \\ 0 \\ l/2 \end{Bmatrix} \tag{2.8}$$

Defining θ_1 as the first joint angle, and assuming as $\theta_1 = 0$ for the configuration with deployed manipulator, the position of the second joint can be determined with

$$x_2 = L_1 \begin{Bmatrix} cos(\theta_1) \\ 0 \\ sin(\theta_1) \end{Bmatrix} + x_1 \tag{2.9}$$

Defining θ_2 as the second joint angle and assuming as $\theta_2 = 0$ for the aligned configuration of the two links, the position of the end-effector can be determined with

$$x_{EE} = L_2 \begin{Bmatrix} cos(\theta_1)cos(\theta_2) \\ sin(\theta_2) \\ sin(\theta_1)cos(\theta_2) \end{Bmatrix} + x_2 \tag{2.10}$$

The joint angles θ_1 and θ_2 identify the arm state and can be determined from the arm dynamics, expressed by one equation for each joint

$$\begin{cases} \ddot{\theta}_1 = \frac{M_1}{J_1} \\ \ddot{\theta}_2 = \frac{M_2}{J_2} \end{cases} \tag{2.11}$$

where M_i is the applied torque in the i-th-joint, and J_i identify the inertia of the manipulator rotating part around each joint. Considering the overall torque applied for the manipulator motion, it can be written in the body reference frame as

$$T_{arm} = \begin{Bmatrix} -M_2cos(\theta_1) \\ -M_1 \\ M_2sin(\theta_1) \end{Bmatrix} \tag{2.12}$$

where the vector T_{arm} represents the product $J_{arm}\dot{\omega}_{arm}$ expressed in the body reference frame. From T_{arm}, the disturbance torque that acts in the main body dynamics can be defined as

$$T_{dis} = -T_{arm} = \begin{Bmatrix} M_2cos(\theta_1) \\ M_1 \\ -M_2sin(\theta_1) \end{Bmatrix} \tag{2.13}$$

In the same way, the manipulator angular momentum can be expressed in the body reference frame as

$$h_{arm} = J_{arm}\omega_{arm} = \begin{Bmatrix} J_2\omega_2cos(\theta_1) \\ J_1\omega_1 \\ -J_2\omega_2sin(\theta_1) \end{Bmatrix} \tag{2.14}$$

The manipulator motion provides the overall inertia changing, which is determined from the manipulator position. The overall inertia is computed in reference to the

CoM of the main body, and it can be expressed by the sum of the body inertia matrix, approximately constant, and the manipulator contribution as expressed in the equation (2.1). Thus, the inertia variation can be expressed by the following equation

$$\dot{J} = \dot{J}_{body} + \dot{J}_{arm,0} \tag{2.15}$$

Since the main body's inertia matrix is constant, the inertia variation is completely due to the manipulator term that be computed with

$$\dot{J} = \dot{J}_{arm,0} = \frac{\partial J_{arm,0}}{\partial \theta}\frac{\partial \theta}{\partial t} = \frac{\partial J_{arm,0}}{\partial \theta}\omega_{arm} \tag{2.16}$$

The inertia variation can be computed according to the manipulator state variables. The matrix formulation output is a vector containing each inertia variation terms that leads to the inertia variation matrix.

In this way, it is possible to define each terms in the main body attitude dynamics due to the manipulator motion. If the manipulator is not moving, the attitude dynamic can be traced back to the Euler equations, where the inertia is determined by the main body inertia and the manipulator contribution, that will be a constant value refered to the manipulator position. Thus, when the manipulator is moving, each terms, introduced before, can be considered as a disturbance due to its motion, and the overall inertia will change according to the manipulator position.

Chapter 3

Control strategies

The attitude and manipulator dynamics are both described by nonlinear systems, and subjected to variable disturbances and inertia properties changing. Thus, the attitude controller should assure high accuracy and robustness against disturbances and parameters variation.
In this chapter the sliding mode and the backstepping approaches are introduced and described.

3.1 Sliding mode approach

SMC is widely used to control nonlinear systems. It shows some noise-rejection properties as a robust method. It is characterized by a solid mathematical base, and its most important feature is the robustness against imprecise knowledge of the controlled system and disturbances [5, 13].
The controller goal is to satisfy the tracking control problem, which means that the plant output signal $y(t)$ needs to reach in finite time the reference signal y_r. This can be mathematically expressed by

$$y(t) \rightarrow y_r(t) \quad t > t_r \tag{3.1}$$

The controller sends an input signal to the plant that is a function of the plant state $x(t)$ and output $y(t)$.
SMC consists in the definition of a sliding surface σ, which has to be reached by the system. Thus, the control forces lead the system on σ, and let it slide on the surface, as shown in Figure 3.1.

The sliding surface is defined by a linear combination of the tracking error derivatives

$$\sigma = k_\gamma e^{(\gamma)} + k_{\gamma-1} e^{(\gamma-1)} + ... + k_1 e \tag{3.2}$$

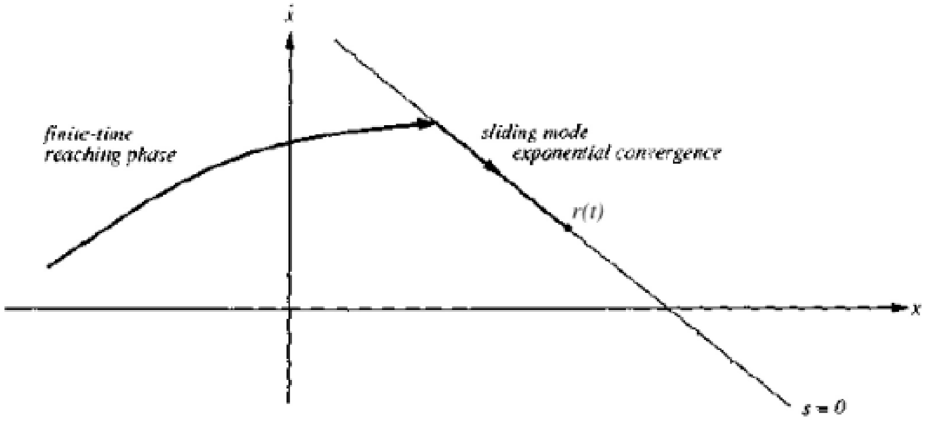


Figure 3.1: Sliding mode convergence [14]

where the terms k_i have to lead to negative real part roots in the polynomial $P(x)$, defined as

$$P(x) = k_\gamma x^\gamma + k_{\gamma-1} x^{\gamma-1} + ... + k_1 x \tag{3.3}$$

When the trajectory is confined on the sliding surface, the tracking error will reach 0 in exponential way, according to the roots of $P(x)$.

First-order sliding mode controller

Let's consider the following *single-input-single-output* (SISO) dynamic system

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \tag{3.4}$$

where $x(t) \in \mathbb{R}^n$ is the system state vector, $y(t) \in \mathbb{R}$ is the system output, and $u(t) \in \mathbb{R}$ is the control input. The tracking error can be expressed by

$$\tilde{y} = y - y_r \tag{3.5}$$

where $y_r(t)$ is the desired output. Lets consider the sliding surface, defined by the following equation:

$$S = \{x(t) \in \mathbb{R}^3 : \ \sigma(t) = 0\} \tag{3.6}$$

where $\sigma(t)$ is a function of the tracking error derivatives, as shown in Equation (3.2).

The control law has to be computed so that the sliding surface is both invariant and attractive. So, a discontinuous term is needed in the u definition.

The ideal control law is expressed by using the *sign* function

$$u = -k\ sign(\sigma) \tag{3.7}$$

To improve these properties the equivalent control can be introduced in the control law definition. It consists in exploiting $\dot{\sigma} = 0$ to find a relation between u and the system dynamics. Applying this control to the system leads to the natural attraction to the surface. The equivalent control is not considered in this study to keep the low complexity, low computational, and low cost characteristics of the SMC methodology. However, the discontinuous term brings high frequency oscillation around the sliding surface. To overcome this problem, defined *chattering effect*, the *sign* function in the control law can be approximated with a sigmoid function, as in Equation (3.8), partially losing robustness, or by using an higher order sliding mode [15].

$$sign(\sigma) \approx tanh(\eta\sigma) \tag{3.8}$$

Twisting sliding mode controller

The twisting controller is a SMC that takes advantage of the first-order sliding surface derivative to improve the controller performance and to reduce the chattering effect [16]

$$u = -r_1 sign(\sigma) - r_2 sign(\dot{\sigma}) \tag{3.9}$$

where $k_1 \in \mathbb{R}^{\nVdash}$ and $k_2 \in \mathbb{R}^{\nVdash}$ are constant parameters.
As Equation (3.9) shows, the control law is expressed by $u = u(\sigma, \dot{\sigma})$, thus, the twisting controller belongs to the second-order SMC family. The generally r-th order sliding modes is determined from

$$\sigma = \dot{\sigma} = \ddot{\sigma} = ... = \sigma^{(r-1)} = 0 \tag{3.10}$$

Considering a general second-order derivative of the sliding surface, it is given by [17]

$$\ddot{\sigma} = h(t,x) + g(t,x)u \quad g(t,x) \neq 0 \tag{3.11}$$

where the functions h and g are defined as

$$h(t,x) = |\ddot{\sigma}_{u=0}| \quad g(t,x) = \frac{\partial \ddot{\sigma}}{\partial u} \tag{3.12}$$

Analyzing the system dynamics, the bounds of the functions h and g can be computed as

$$|\ddot{\sigma}_{u=0}| < C \qquad 0 < K_m < \frac{\partial \ddot{\sigma}}{\partial u} < K_M \tag{3.13}$$

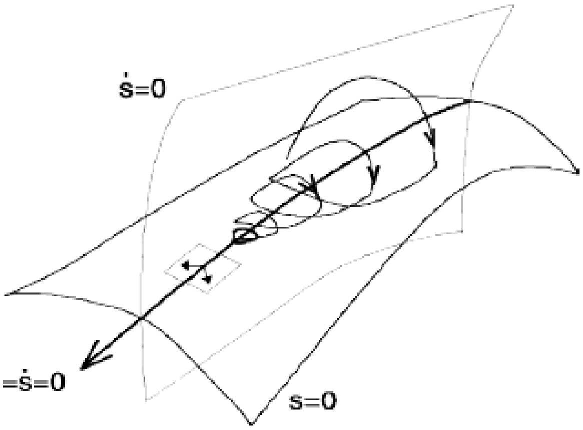


Figure 3.2: Second order sliding mode trajectory [17]

In order to achieve overall stability of the controlled system and the requested accuracy, the controller gains r_1 and r_2 have to satisfy

$$\begin{cases} K_m(r_1+r_2)-C > K_M(r_1-r_2)+C \\ K_m(r_1-r_2) > C \end{cases} \tag{3.14}$$

In practice the parameters are not assigned considering this inequalities, because usually the real system is not exactly known or the model is not really adequate, so C, K_m and K_M parameters estimation is much larger than the actual values [17].

3.2 Backstepping approach

The backstepping procedure consists in an iterative algorithm for the controller design based on Lyapunov functions [18]. In control theory, a Lyapunov function is a function $V(x)$ that is continuously differentiable, positive definite and such that

$$\forall x \neq 0, \exists u \qquad \dot{V}(x,u) = \nabla V(x) f(x,u) < 0 \tag{3.15}$$

where the last condition stands that for each state x, a control u able to drive $V(x)$ to zero can be found. Thus, it means that the controller is able to bring the system to an equilibrium stable state.
The controller will be designed to satisfy the tracking problem. At every iteration, the back-stepping controller builds a Lyapunov function that certifies the stability of the tracked state and the needed control law to reach it.

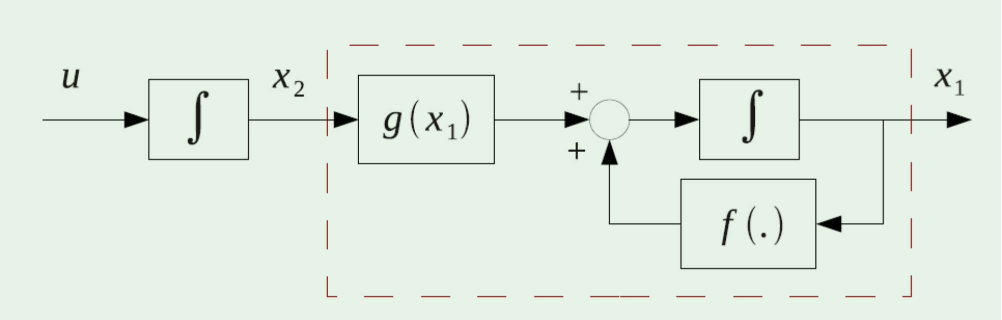


Figure 3.3: Generic dynamic system

A generic dynamic system is given by:

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1)x_2 \\ \dot{x}_2 = u \end{cases} \tag{3.16}$$

As Figure 3.3 shows, the dynamic system can be divided into two subsystems: the first equation can be represented by the inner closed-loop, while the second one by the integration of the controller output.
Let's assume that for the closed-loop system stands

$$\dot{x}_1 = f(x_1) + g(x_1)\phi_1$$

where $\phi_1 \in C^1$, such that $\bar{x}_1 = 0$ is an equilibrium state for the closed-loop system and let's assume that the Lyapunov function $V_1(x)$ that certifies stability of $\bar{x}_1 = 0$ is known, then the dynamics of the error of the control variable can be defined and computed by

$$\begin{cases} \dot{x_1} = f(x_1) + g(x_1)(x_2 - \phi_1(x_1)) \\ \dot{x_2} = u \end{cases} \tag{3.17}$$

Equation (3.17) can be rewritten applying the following coordinate change as

$$\begin{cases} \eta = x_2 - \phi_1(x_1) \\ v = u - \dot{\phi_1} \end{cases} \tag{3.18}$$

In this way ϕ_1 is back-stepped before the integrator and v is such that for $\bar{v} = 0$, the state $\bar{x}_1 = \bar{\eta} = 0$ is an equilibrium state. Equation (3.17) can be so rewritten as

$$\begin{cases} \dot{x_1} = f(x_1) + g(x_1)\phi_1(x_1) + g(x_1)\eta \\ \dot{\eta} = v \end{cases} \tag{3.19}$$

A second candidate Lyapunov function is needed to define the control law. Let's consider the following Lyapunov function $V_2(x_1, \eta)$

$$V_2(x_1, \eta) = V_1(x_1) + \frac{\eta^2}{2} \tag{3.20}$$

and its derivative

$$\dot{V_2} = D_{x_1}[V_1]\dot{x_1} + \eta\dot{\eta} \tag{3.21}$$

Replacing $\dot{x_1}$ and $\dot{\eta}$ as expressed in Equation (3.19), $\dot{V_2}$ can be rewritten as

$$\dot{V_2} = D_{x_1}[V_1](f(x_1) + g(x_1)\phi_1(x_1)) + D_{x_1}[V_1]g(x_1)\eta + \eta v \tag{3.22}$$

Choosing

$$v = -D_{x_1}[V_1]g(x_1) - \tilde{k}\eta \quad \tilde{k} > 0 \tag{3.23}$$

and replacing its value in Equation (3.22)

$$\dot{V_2} = \dot{V_1} - \tilde{k}\eta^2 \tag{3.24}$$

It finds out that the Lyapunov function V_2 is negative defined, so the system is asymptotically stable. Going back in the original coordinates, the control u can be defined as

$$u = v + \dot{\phi_1} = D_{x_1}[\phi_1](f(x_1) + g(x_1)x_2) - D_{x_1}[V_1]g(x_1) - \tilde{k}(x_2 - \phi_1(x_1)) \tag{3.25}$$

The control law is a function of the system dynamics, so the control input is adapted to the actual state of system by using the estimation provided from the system dynamics mathematical model. In this way it is possible to design a control input that takes into account all the disturbances introduced by the manipulator motion, by accepting an higher computational cost than the SMCs [6, 19].

Chapter 4

Astrobee's controllers design

The Astrobee model has been realized considering the two subsystems, the main body and manipulator, coupled with the relations defined in the previous chapter. This leads to design two different controllers, first one to execute the desired arm motion and the second one to handle attitude stabilization and control [20].

Figure 4.1 shows the overall scheme for the Astrobee's model implementation. The arm controller has been implemented to track the desired manipulator motion, while the attitude controller is needed to keep the system stable during the arm movements and to handle precise attitude control. In this way, the Astrobee is able to perform any autonomous task just modifying the reference signals. The manipulator controller has been implemented using the sliding mode approach, while for the attitude controller the second-order sliding mode and the backstepping approaches are proposed as suitable options.

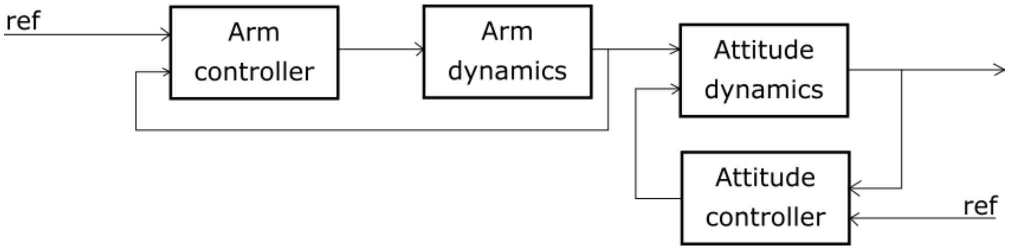


Figure 4.1: Astrobee's Simulink implementation

4.1 Manipulator motion control design

The manipulator controller has been implemented as a first-order SMC, which has been chosen thanks to its low computational and low complexity features. It is based on the definition of a sliding surface where the system is forced to move. For the manipulator motion control, the sliding surface has been defined by the manipulator state variable θ and ω, respectively the joints angles and rotational speed as

$$\sigma = \lambda\theta_e + \omega_e \tag{4.1}$$

where θ_e and ω_e are the tracking errors, defined as the differences between the desired and the actual joint angle and angular velocity

$$\begin{cases} \theta_e = \theta - \theta_{des} \\ \omega_e = \omega - \omega_{des} \end{cases} \tag{4.2}$$

To make the sliding surface attractive, the control law has been defined as discontinuous function

$$u = -ksign(\sigma) \tag{4.3}$$

where $k \in \mathbb{R}$ is a constant parameter. In this way the controller will lead the system to the sliding surface. To avoid the chattering introduced by the *sign* function, the controller has been modified introducing the hyperbolic tangent formulation

$$u = -ktanh(\eta\sigma) \tag{4.4}$$

4.2 Attitude control design

The attitude controller has been implemented in two different ways to evaluate two different approaches. First, the TW-SMC has been designed to achieve a trade-off among control law flexibility, robustness, and precision attitude control. Thus, the TW-SMC has been compared with a back-stepping approach, which consists in an adaptive controller based on Lyapunov functions. Its output takes into account attitude error, the arm torque and motion, and the inertia variation, leading the Astrobee to track the desired attitude.

Twisting sliding mode controller

The TW-SMC is a second-order SMC, which enjoys all the feature of the sliding mode approach and improves performance and chattering attenuation introducing the first-order sliding surface derivative in the control law definition. The TW-SMC is defined by the following control law

$$u = -k_1 sign(\sigma) - k_2 sign(\dot{\sigma}) \tag{4.5}$$

where $k_1 \in \mathbb{R}^{3\times3}$ and $k_2 \in \mathbb{R}^{3\times3}$ are diagonal matrices, which need to satisfy the following relations to guarantee the system convergence and stability

$$\begin{cases} K_m(k_1 + k_2) - C > K_M(k_1 - k_2) + C \\ K_m(k_1 - k_2) > C \end{cases} \tag{4.6}$$

where C, K_m and K_M are constant values defined by

$$|\ddot{\sigma}_{u=0}| < C \qquad 0 < K_m < \frac{\partial \ddot{\sigma}}{\partial u} < K_M \tag{4.7}$$

The sliding surface has been defined by the quaternion and angular velocity error

$$\sigma = q_e \lambda + \omega_e \tag{4.8}$$

where q_e is the quaternion error, which can be computed with

$$q_e = q \otimes q_{des}^{-1} \tag{4.9}$$

and ω_e is the angular velocity error. To define the control law parameter's bounds an easier formulation of the dynamics has been considered

$$\dot{\omega} = J^{-1}(u + T_{dis} - \omega \times (J\omega) \tag{4.10}$$

where T_{dis} is a constant value, introduced to approximate all the disturbances due to the arm motion.
Since the parameter bounds are defined by Equation (4.7), the sliding surface derivatives have to be computed. Considering σ defined by Equation (4.8), the first derivative can be computed as

$$\dot{\sigma} = \dot{q}\lambda + \dot{\omega} = \frac{1}{2}\lambda(q_0 I + [q\times])\omega + J^{-1}(u + T_{dis} - \omega \times (J\omega) \tag{4.11}$$

From $\dot{\sigma} = 0$ the equivalent control can be determined

$$u_e = \omega \times (J\omega) - \frac{1}{2}\lambda J((q_0 I + [q\times])\omega \tag{4.12}$$

The equivalent control takes into account the system dynamics to define the control input to lead the system to the desired configuration through the sliding surface.
From Equation (4.11) $\ddot{\sigma}$ can be computed as

$$\ddot{\sigma} = \frac{1}{2}\lambda(\dot{q}_0\omega + q_0\dot{\omega} + \dot{q}_v \times \omega + q_v \times \dot{\omega}) + J^{-1}(\dot{u} - \dot{\omega} \times (J\omega) - \omega \times (J\dot{\omega})) \tag{4.13}$$

that can be rewritten by using the Taylor envelop as

$$\begin{cases} \ddot{\sigma}_{u=0} = \frac{1}{2}\lambda(\dot{q}_0\omega + q_0\dot{\omega} + \dot{q}_v \times \omega + q_v \times \dot{\omega}) - J^{-1}(\dot{\omega} \times (J\omega) + \omega \times (J\dot{\omega})) \\ \frac{\partial \ddot{\sigma}}{\partial t}_{u=0} = \frac{1}{2}\lambda(q_0 \frac{\partial \dot{\omega}}{\partial t}_{u=0} + q_v \times \frac{\partial \dot{\omega}}{\partial t}_{u=0}) - J^{-1}(\frac{\partial \dot{\omega}}{\partial t}_{u=0} \times (J\omega) + \omega \times (J\frac{\partial \dot{\omega}}{\partial t}_{u=0})) \end{cases}$$

The bounds can be determined by running several simulations with random values of initial attitude, angular velocity and external disturbances, and the equivalent control is considered as control output. Figure 4.2 shows the simulations results of $\ddot{\sigma}_{u=0}$ and $\frac{\partial \ddot{\sigma}}{\partial t}_{u=0}$. The upper and lower plots' limits identify C, K_m, and K_M parameters. Generally, the computed parameters are not used to assign k_1 and k_2,

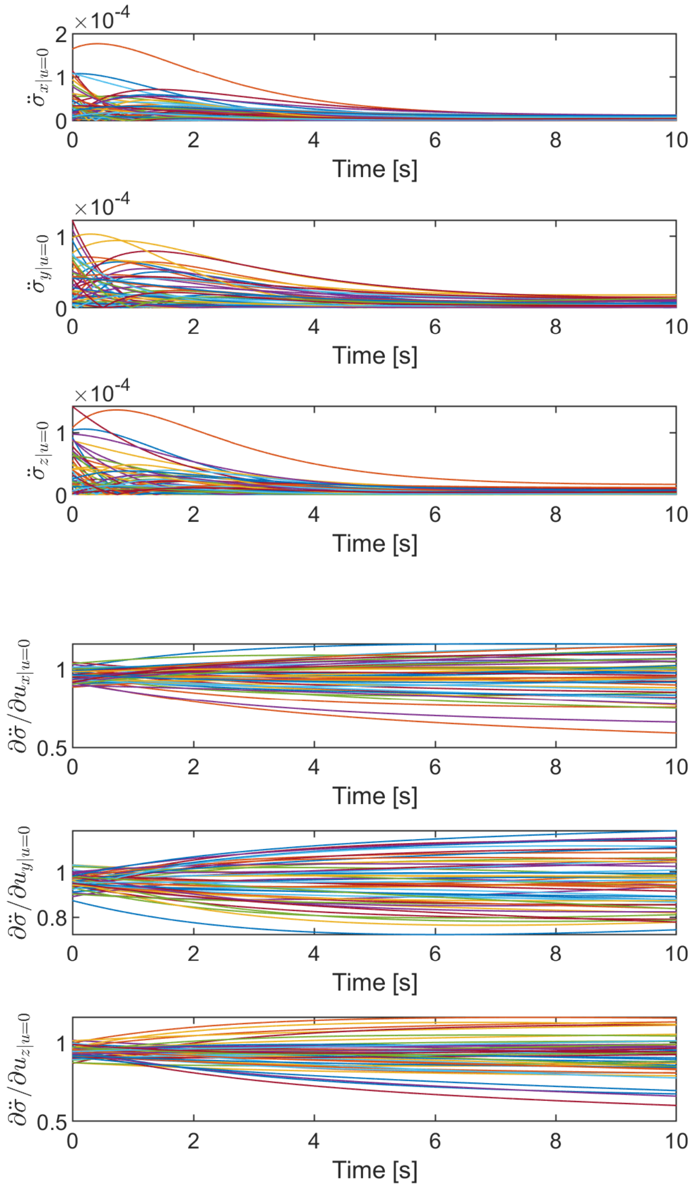


Figure 4.2: Second-order sliding mode parameters' bounds

because they are usually much larger than the actual values, but they have been anyway computed to verify that the selected k_1 and k_2 values are contained into the bounds.
The control law parameters k_1 and k_2 have been designed to achieve a trade-off among control law robustness and precision attitude control. Thus, the control law input is computed with the following equation

$$u = -k_1 sign(\sigma) - k_2 sign(\dot{\sigma}) \tag{4.14}$$

where $\dot{\sigma}$ is numerically determinated. To further improve the chattering attenuation, the hyperbolic tangent formulation has been implemented

$$u = -k_1 tanh(\eta\sigma) - k_2 tanh(\eta\dot{\sigma})$$

Backstepping controller

The backstepping controller is an adaptive controller based on Lyapunov functions. The backstepping approach consists in the definition of a virtual control variable to ensure the tracking of the desired attitude q_d [21].
Let's considering the tracking error, defined by the quaternion error, expressed as

$$q_e = \begin{Bmatrix} \tilde{q}_0 \\ \tilde{q}_v \end{Bmatrix} \tag{4.15}$$

Which derivative can be determined by

$$\dot{q}_e = \begin{Bmatrix} \tilde{q}_v^T \tilde{\omega} \\ 1/2(q_0 I + [q_v \times])\tilde{\omega} \end{Bmatrix} \tag{4.16}$$

Considering the backstepping variable $\tilde{q}_v$, its derivative can be determined by:

$$\dot{\tilde{q}}_v = 1/2(q_0 I + [q_v \times])(\omega_c + \tilde{\omega} - R(q_e)\omega_d) \tag{4.17}$$

where $R(q)$ is the rotational matrix and ω_c is the back-stepping virtual control variable. Let's define the first Lyapunov function candidate as:

$$V_1 = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 \tag{4.18}$$

The derivative of V_1 can be determined using the Equation (4.16):

$$\dot{V}_1 = 2\tilde{q}_v^T \dot{\tilde{q}}_v - 2(1 - \tilde{q}_0)\dot{\tilde{q}}_0 \tag{4.19}$$

Defining ω_c as

$$\omega_c = -K_1 \tilde{q}_v + R(q_e)\omega_d \tag{4.20}$$

The Equation (4.19) can be rewritten as:

$$\dot{V}_1 = -\tilde{q}_v^T K_1 \tilde{q}_v + \tilde{q}_v^T \tilde{\omega} \tag{4.21}$$

When the angular velocity error is equal to the desired value, $\tilde{\omega} = 0$, $\dot{V}_1$ is negative definite. This means that the system is able to reach and stop into the desired configuration.
To define the control law, the second Lyapunov function candidate is needed to be defined. Let's consider the following Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2}\tilde{\omega}^T J \tilde{\omega} \tag{4.22}$$

which has derivative

$$\dot{V}_2 = -\tilde{q}_v^T K_1 \tilde{q}_v + \tilde{q}_v^T \tilde{\omega} + \tilde{\omega}^T J \dot{\tilde{\omega}} + \frac{1}{2}\tilde{\omega}^T \dot{J} \tilde{\omega} \tag{4.23}$$

Let's consider the following equality

$$J\dot{\tilde{\omega}} + \dot{J}\tilde{\omega} = -(\tilde{\omega} + \omega_c) \times J(\tilde{\omega} + \omega_c) + T - J\dot{\omega}_c - \dot{J}\omega_c \tag{4.24}$$

Combining Equations (4.23) and (4.24), $\dot{V}_2$ can be rewritten as

$$\dot{V}_2 = -\tilde{q}_v^T K_1 \tilde{q}_v + \tilde{q}_v^T \tilde{\omega} + \tilde{\omega}^T (\Sigma\tilde{\omega} - [\omega_c \times] J\tilde{\omega} + \gamma + T - \frac{1}{2}\dot{J}\tilde{\omega}) \tag{4.25}$$

where Σ and γ are defined as

$$\begin{cases} \Sigma = [J\tilde{\omega}\times] + [J\omega_c\times] \\ \gamma = [J\omega_c\times]\omega_c - J\dot{\omega}_c - \dot{J}\omega_c \end{cases} \tag{4.26}$$

Choosing the control law as

$$u = -K_2\tilde{\omega} - \tilde{q_v} - T_{dis} - \gamma + [\omega_c \times] J\tilde{\omega} + \frac{1}{2}\dot{J}\tilde{\omega} \tag{4.27}$$

then Equation (4.25) can be rewritten as

$$\dot{V}_2 = -\tilde{q_v}^T K_1 \tilde{q_v} - \tilde{\omega}^T K_2 \tilde{\omega} \tag{4.28}$$

which is a negative definite function. Thus, both V_1 and V_2 have been verified as Lyapunov functions, and the proposed control law, expressed by Equation (4.27), will request a torque taking into account the attitude error, the arm torque and motion, and the inertia variation, leading the Astrobee to track the desired attitude.

Chapter 5

Simulation results

The Astrobee model has been implemented in MATLAB/Simulink environment. The manipulator motion has been considered as disturbance for the attitude dynamics, and the controllers have been designed to achieve a trade-off between manipulator stabilization and precise attitude control.

The manipulator motion is regulated by the first-order SMC, which defines the sliding surface using joints' angles and angular speed to compute the needed torque

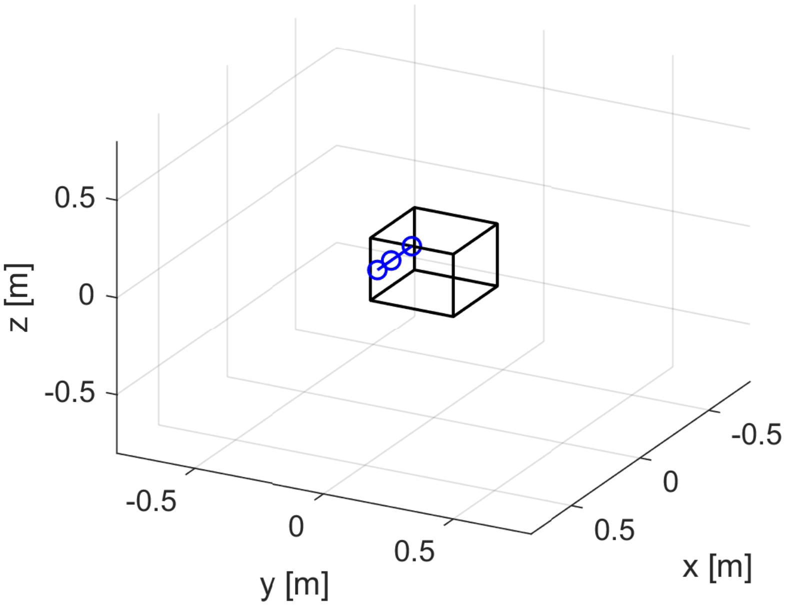


Figure 5.1: MATLAB/Simulink Astrobee 3D model

with the control law

$$u_{arm} = -k_{arm} tanh(\sigma_{arm}) \tag{5.1}$$

For the attitude controller, the TW-SMC and the backstepping controller have been proposed. The twisting controller defines the sliding surface using the quaternions and the angular velocity errors. Thus, by using a simple formulation of the control law, the control input can be determined by

$$u = -k_1 tanh(\sigma) - k_2 tanh(\dot{\sigma}) \tag{5.2}$$

where $\dot{\sigma}$ is numerically determined. On the other hand, the backstepping controller contains the mathematical model of the Astrobee, needed to compute all the control law parameters in a function of the joints' angles and rotational speed

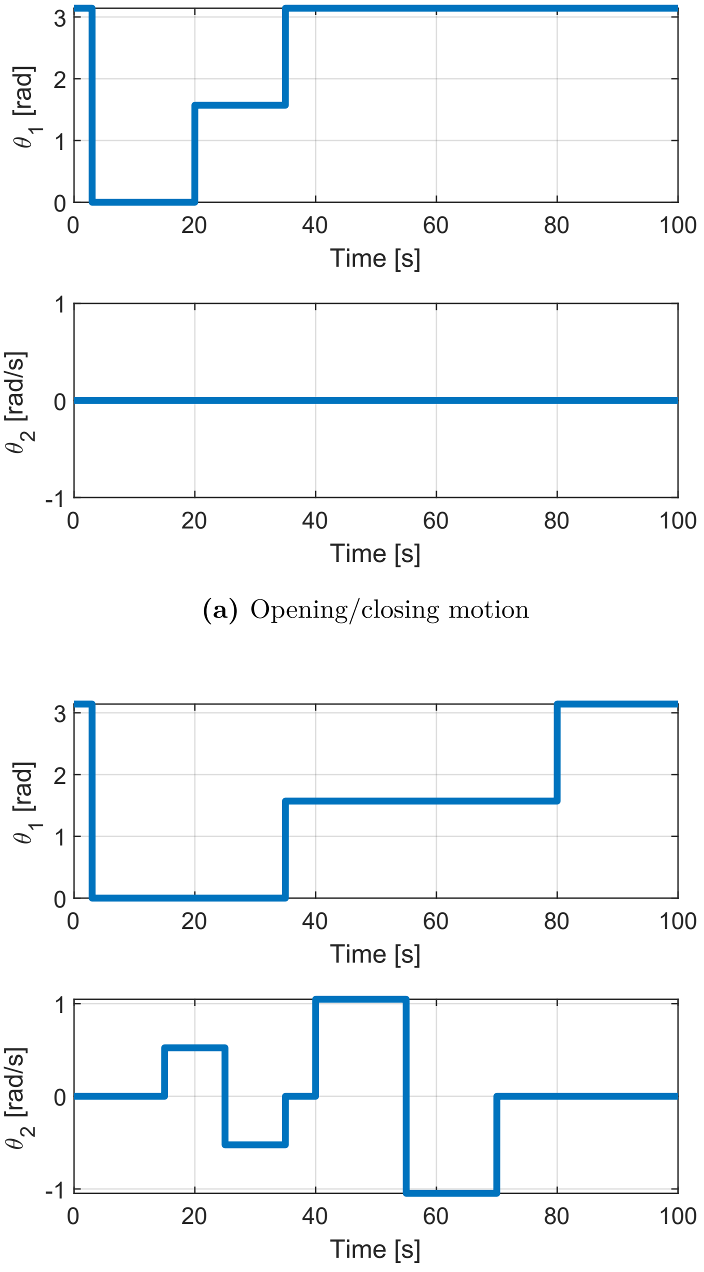
$$u = -K_2\tilde{\omega} - \tilde{q_v} - T_{dis} - \gamma + [\omega_c \times] J\tilde{\omega} + \frac{1}{2}\dot{J}\tilde{\omega} \tag{5.3}$$

In this chapter, controllers' performance is evaluated and analyzed in MATLAB/Simulink, and to show the robustness of the proposed control methods, simulations with variable body mass, links masses, and end-effector mass have been run to test the controllers in off-design conditions.

5.1 Manipulator control

Two different manipulator motions have been considered to evaluate the disturbances generated during the manipulator deploying, stowing, and moving.
First, the manipulator deploying and stowing have been considered to test the attitude dynamics and the stabilization to the rotation around y-axis. Figure 5.2a shows the desired manipulator motion, while Figure 5.3a shows the motion provided from the manipulator controller, and 5.3b shows the disturbance torque expressed in the body reference frame, defined in Equation (2.13). Then, a more complicated manipulator motion has been considered, adding the second joint rotation request during the deployed and vertical manipulator configuration. In Figure 5.2b, the requested motion is showed, and in Figures 5.4a and 5.4b the manipulator controller results are showed.

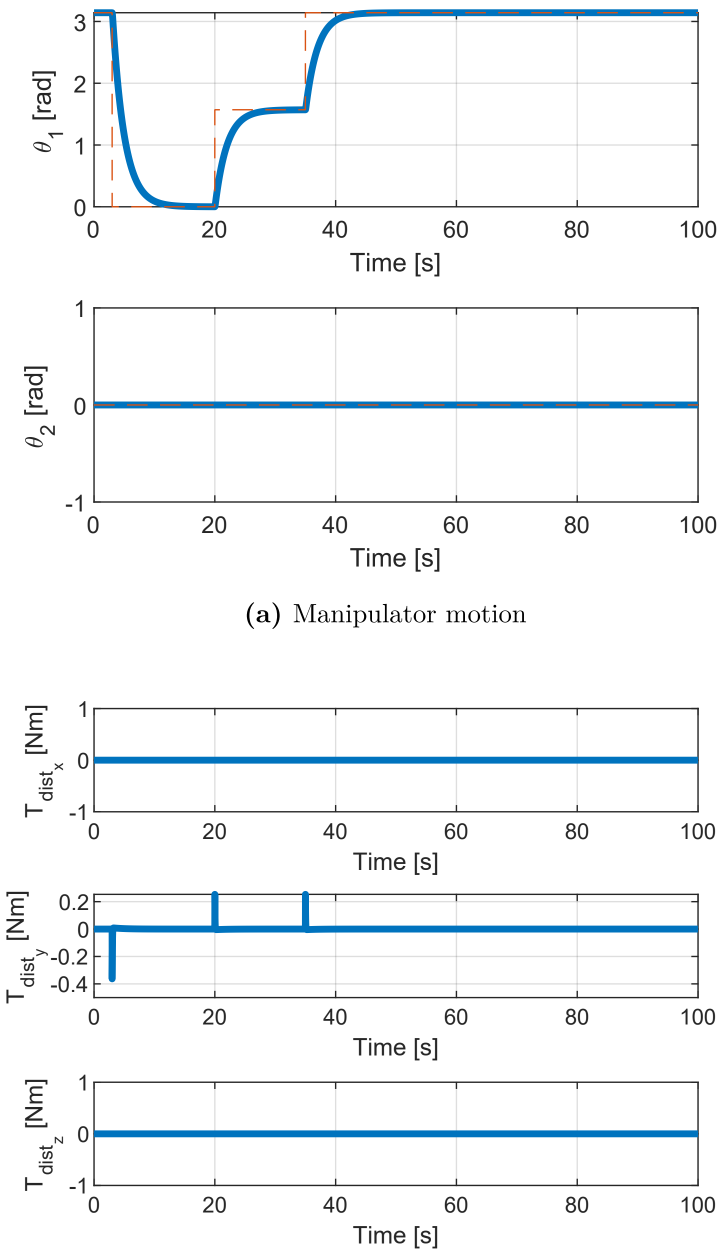
With both the manipulator motion references, when the desired joints' angles change, the manipulator controller request torque is proportional to the differences between the desired and actual joints' angles. Thus, with the angles error reducing, the angular velocity error will prevail and the controller will work to slow down the manipulator to reach the desired configuration.



(a) Opening/closing motion

(b) Opening/rotating/closing motion

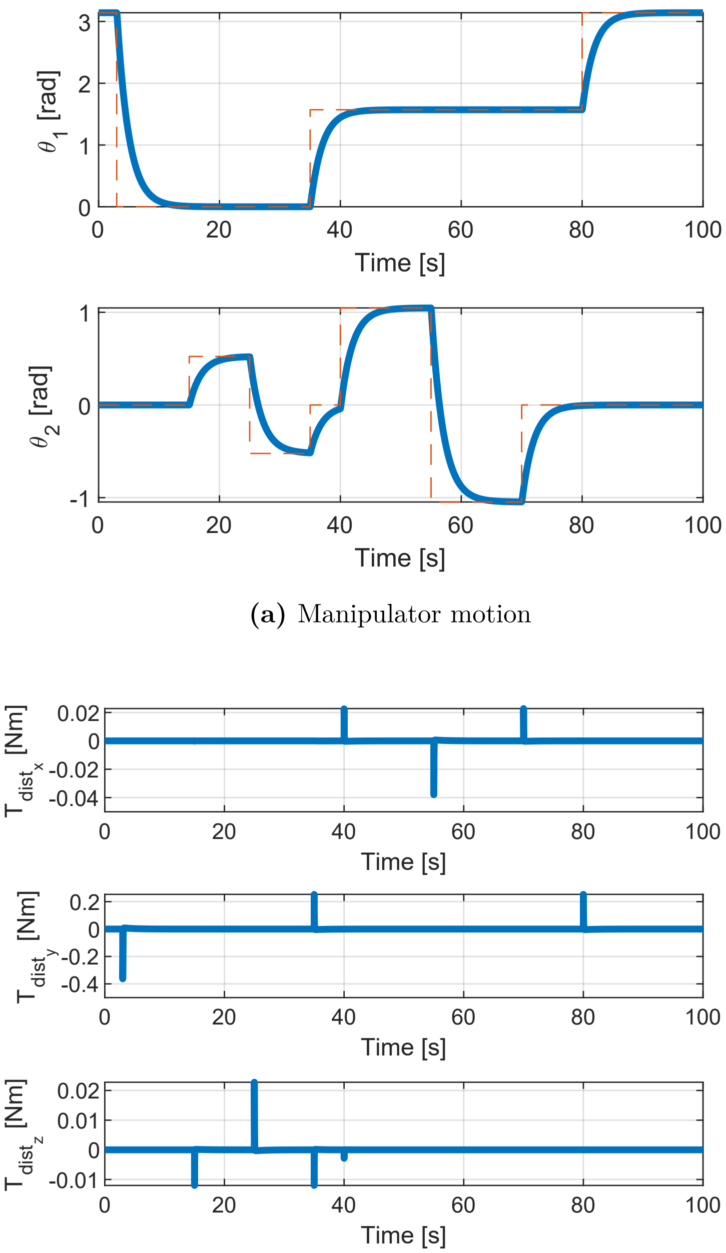
Figure 5.2: Manipulator motion references



(a) Manipulator motion

(b) Disturbance torque

Figure 5.3: Opening/closing manipulator motion



(a) Manipulator motion

(b) Disturbance torque

Figure 5.4: Opening/rotating/closing manipulator motion

5.2 Torque-free simulation

The torque-free simulation has been run to test the attitude dynamics. The manipulator deploying and stowing has been considered for this simulation and its motion is shown in Figure 5.3a.

Figure 5.5 reports the simulation results. Without applying a control torque, the manipulator movement is associated to the main body rotation. This effect is due to the angular momentum conservation, in fact, considering the combined system (the manipulator and main body), since the the application of a torque in the joints is internal to the system the overall angular momentum will remain constant. Thus, when the manipulator is rotating around the y-axis, the main body will rotate in the opposite direction according to the angular momentum conservation.

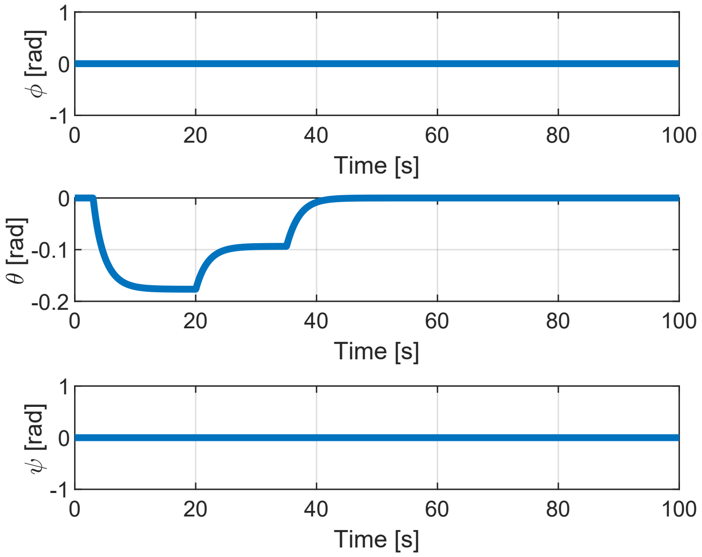


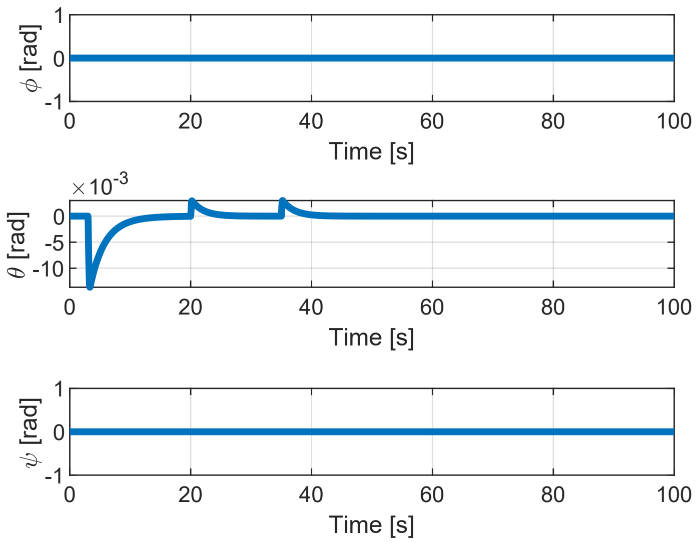
Figure 5.5: Torque-free Euler angles

5.3 Deploying/stowing manipulator stabilization

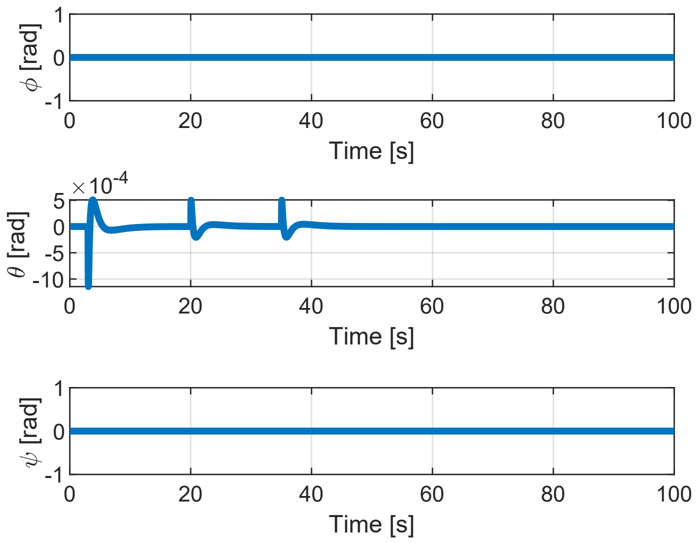
This simulation runs the manipulator motion shown in Figure 5.3a, which consists in deploying and stowing task. Since the main body is already oriented in the desired attitude, both the controllers have been tested for the stabilization of the disturbances introduced by the manipulator motion, in terms of disturbance torque and inertia variation.

Both the TW-SMC and the backstepping controller are able to stabilize the manipulator deploying and stowing motion. In Figure 5.6 the Euler angles are showed and the performance of the controllers can be evaluated. The TW-SMC, thanks to its attractive features, reacts to the disturbance introduced by the manipulator motion whenever the system leaves the sliding surface. The backstepping controller reacts faster to the disturbances, despite a small oscillation around the desired configuration.

In Figure 5.8 is showed the controls input, to be provided to the system. The backstepping controller can adapt its output to the disturbance, in fact, it is able to react with an higher torque request to compensate the disturbances as fast as possible. On the other hand, the TW-SMC has a limited control output due to the constant parameters that define the control law.

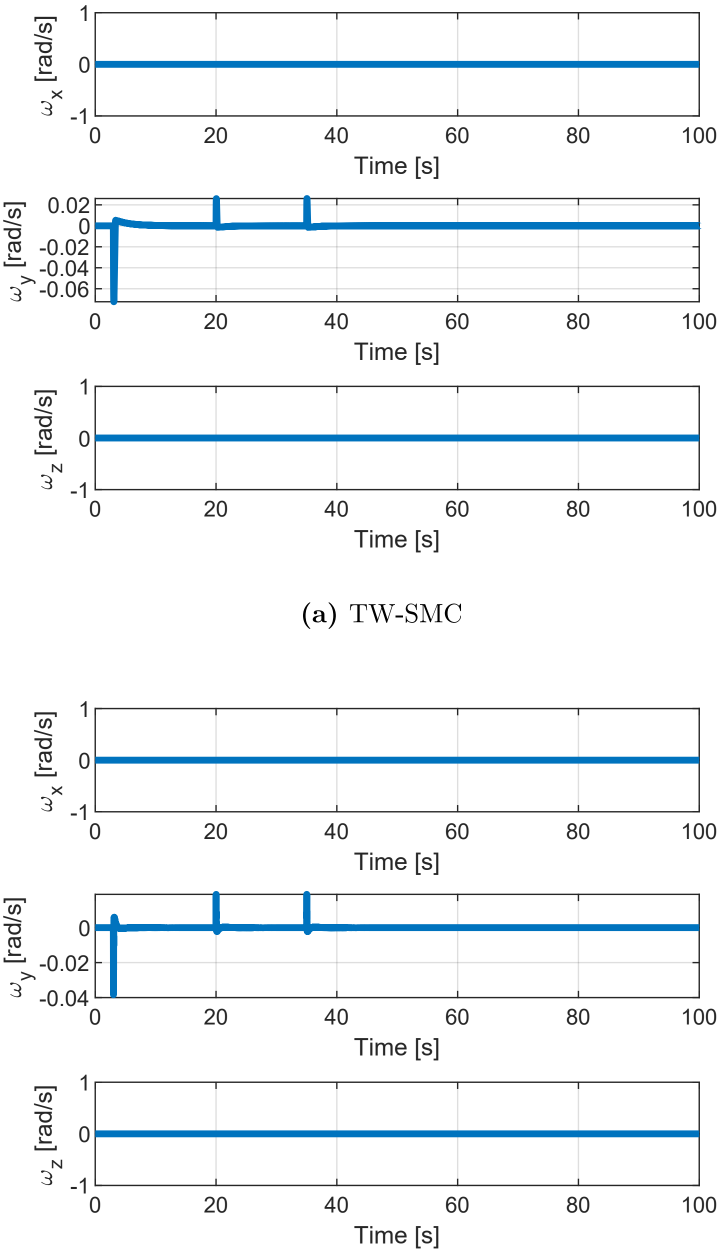


(a) TW-SMC



(b) Backstepping controller

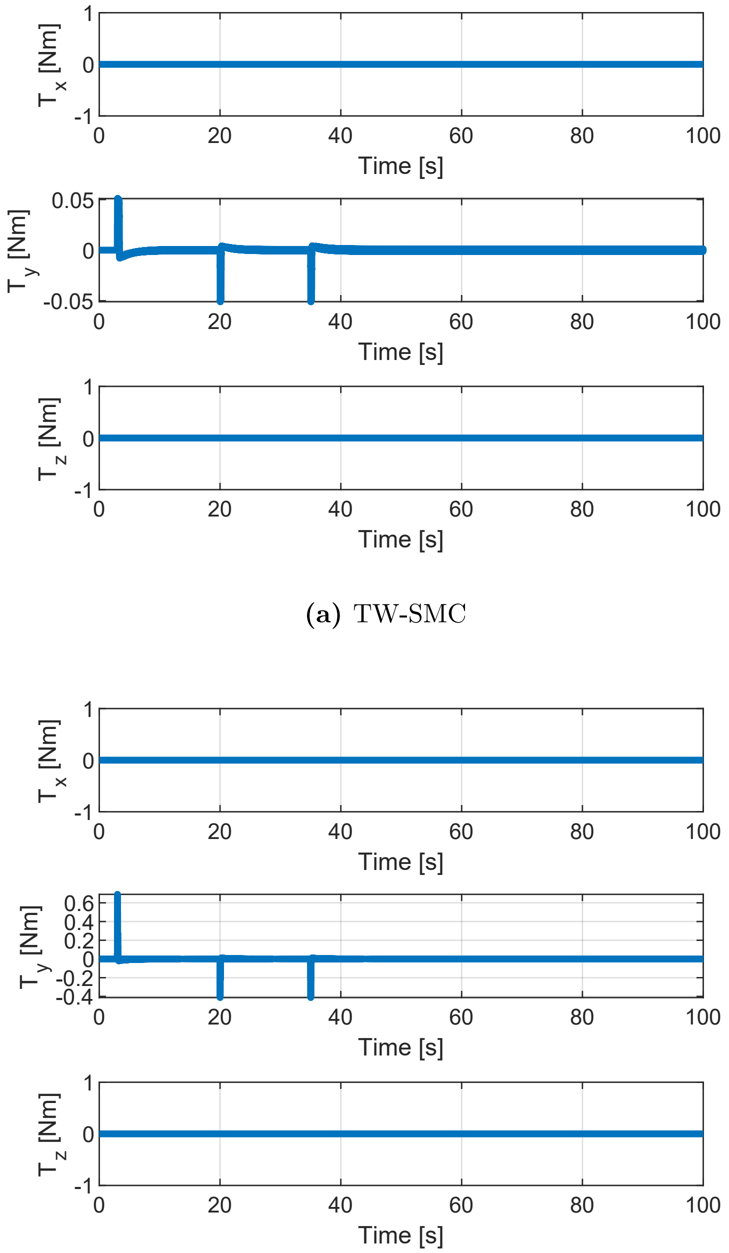
Figure 5.6: Euler angles in deploying/stowing stabilization



(a) TW-SMC

(b) Backstepping controller

Figure 5.7: Angular velocity in deploying/stowing stabilization



(a) TW-SMC

(b) Backstepping controller

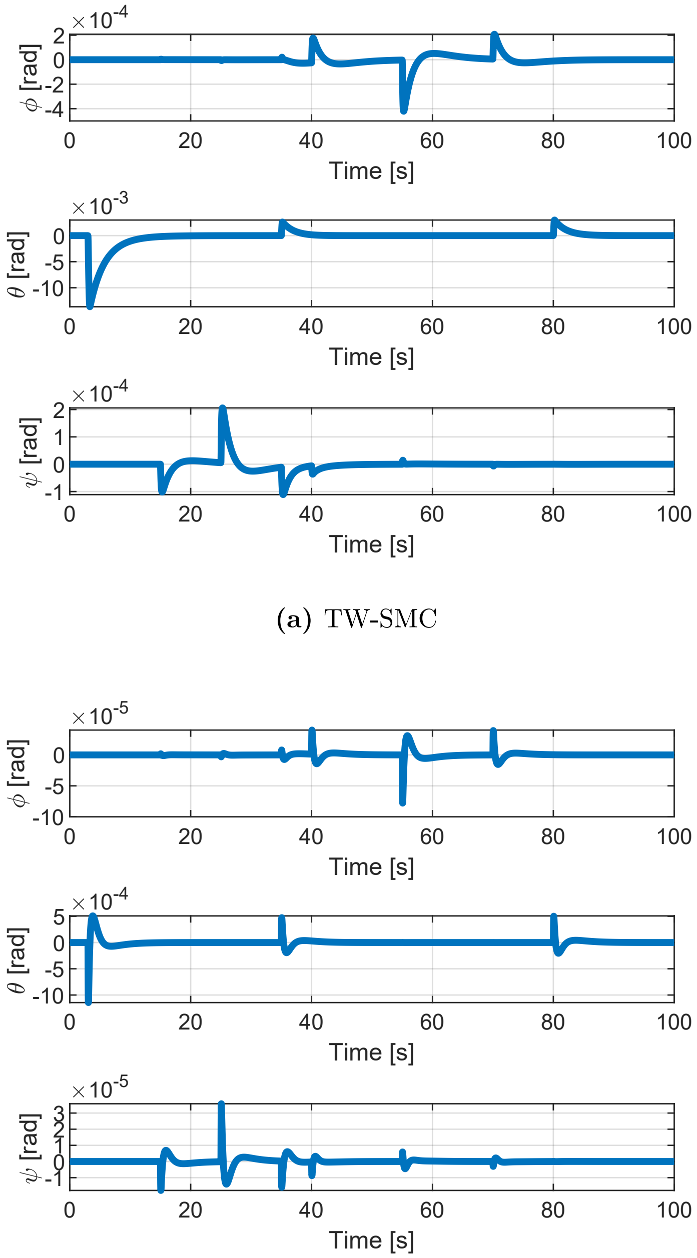
Figure 5.8: Command torque in deploying/stowing stabilization

5.4 Full manipulator motion stabilization

A secondary manipulator motion has been considered. It is presented in Figure 5.4a and consists in introducing the second joint rotation during the deployed and vertical manipulator configuration. A difference of the rotation around the y-axis, the manipulator pan produces negligent inertia variation, so the disturbances introduced are more controllable.

Figure 5.9 shows the simulation results in term of Euler angles. As noticed for the manipulator deploying and stowing stabilization, the backstepping controller reacts faster to the disturbances introduced, despite a small oscillation, and the TW-SMC is able to restore the desired configuration with acceptable time.

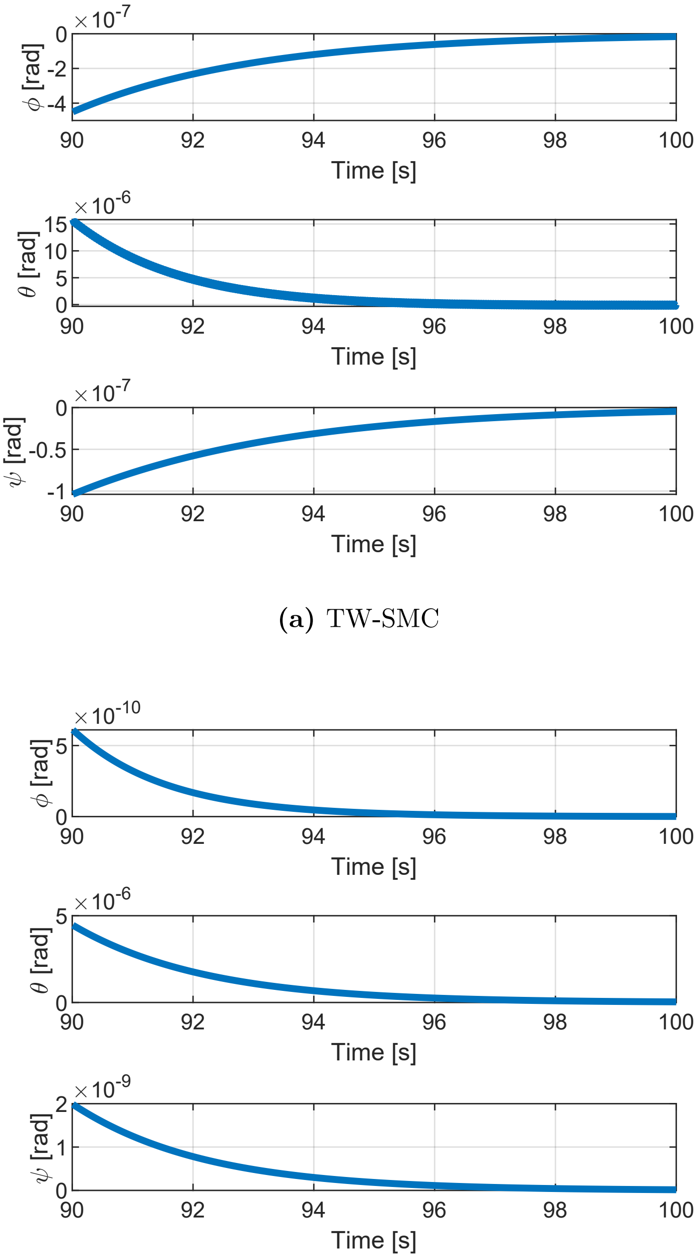
Analysing the results of the simulation, it is evident how the backstepping controller is well suited for the manipulator stabilization, since the system is subject to variable disturbances.



(a) TW-SMC

(b) Backstepping controller

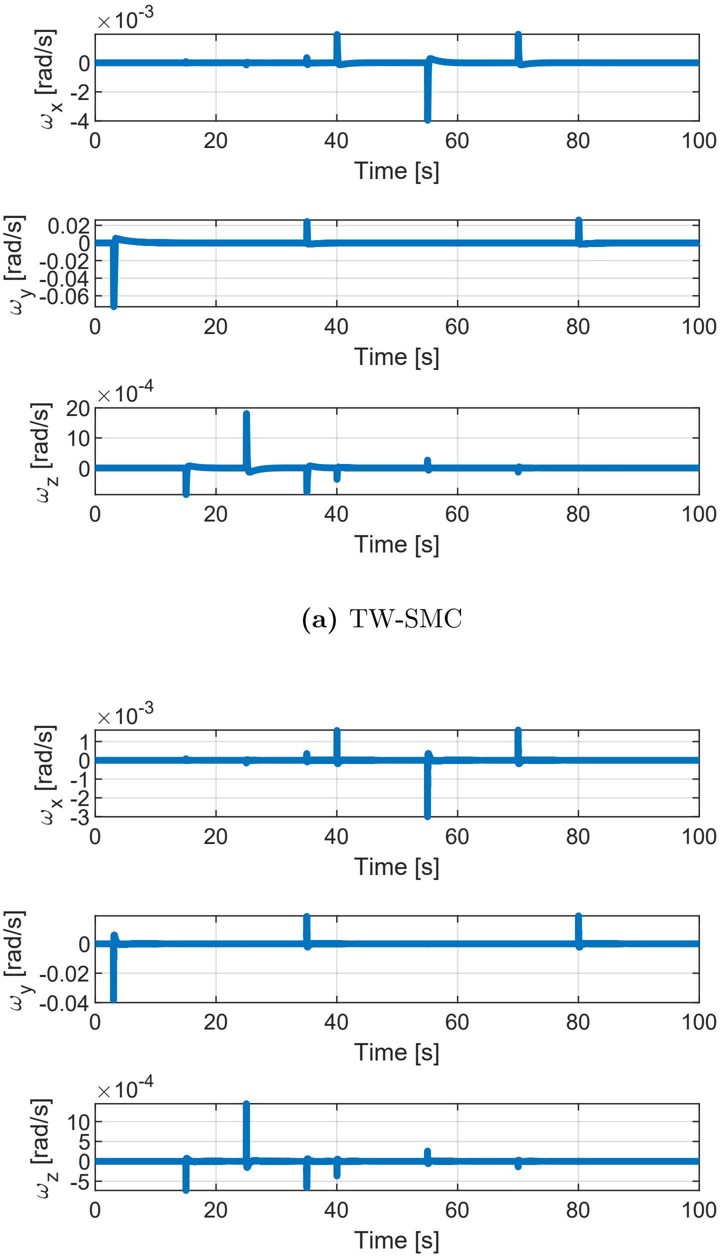
Figure 5.9: Euler angles in full motion stabilization



(a) TW-SMC

(b) Backstepping controller

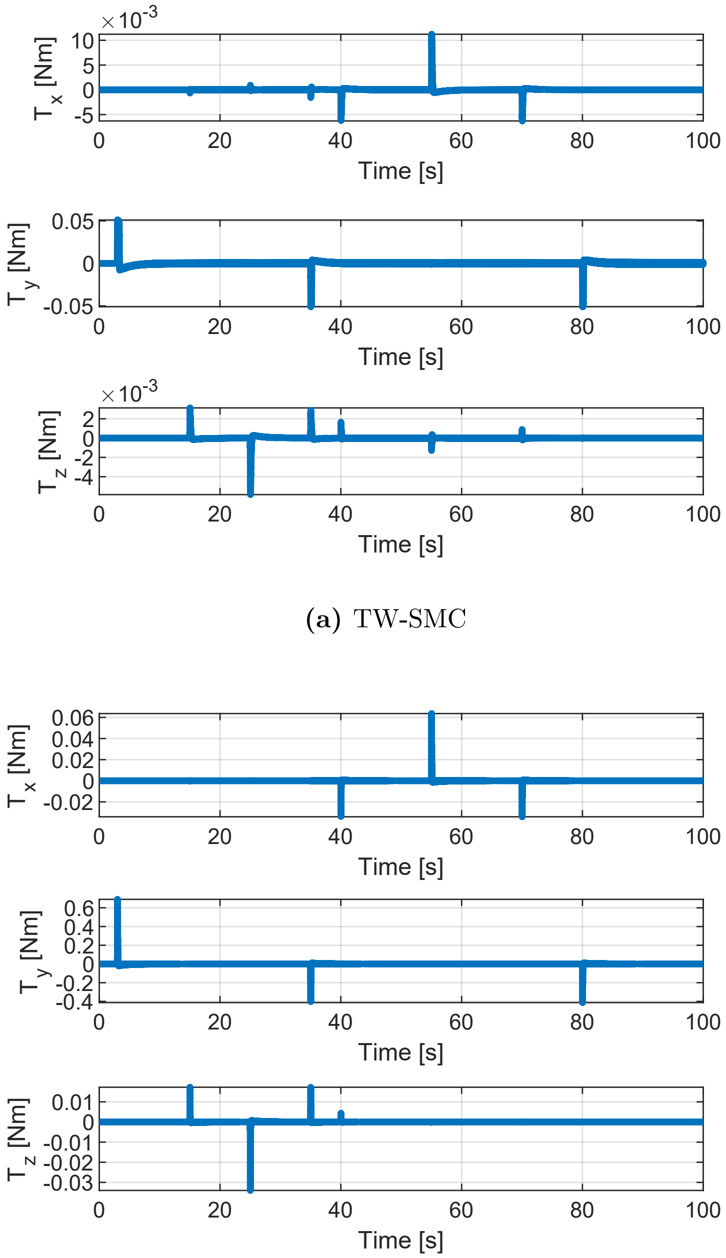
Figure 5.10: Steady-state error



(a) TW-SMC

(b) Backstepping controller

Figure 5.11: Angular velocity in full motion stabilization



(a) TW-SMC

(b) Backstepping controller

Figure 5.12: Command torque in full motion stabilization

5.5 Controller robustness test

TW-SMC and backstepping controller have been both proposed for their robustness feature. To evaluate the robustness of these control methods, further simulations have been run to test the controllers performance in off-design conditions. Thus, the following tests have been considered:

- Attitude changing task;
- Variation of the main body weight and inertia;
- Variation of the manipulator links weight;
- Variation of the end-effector weight, to simulate the manipulator stabilization while an object is grabbed.

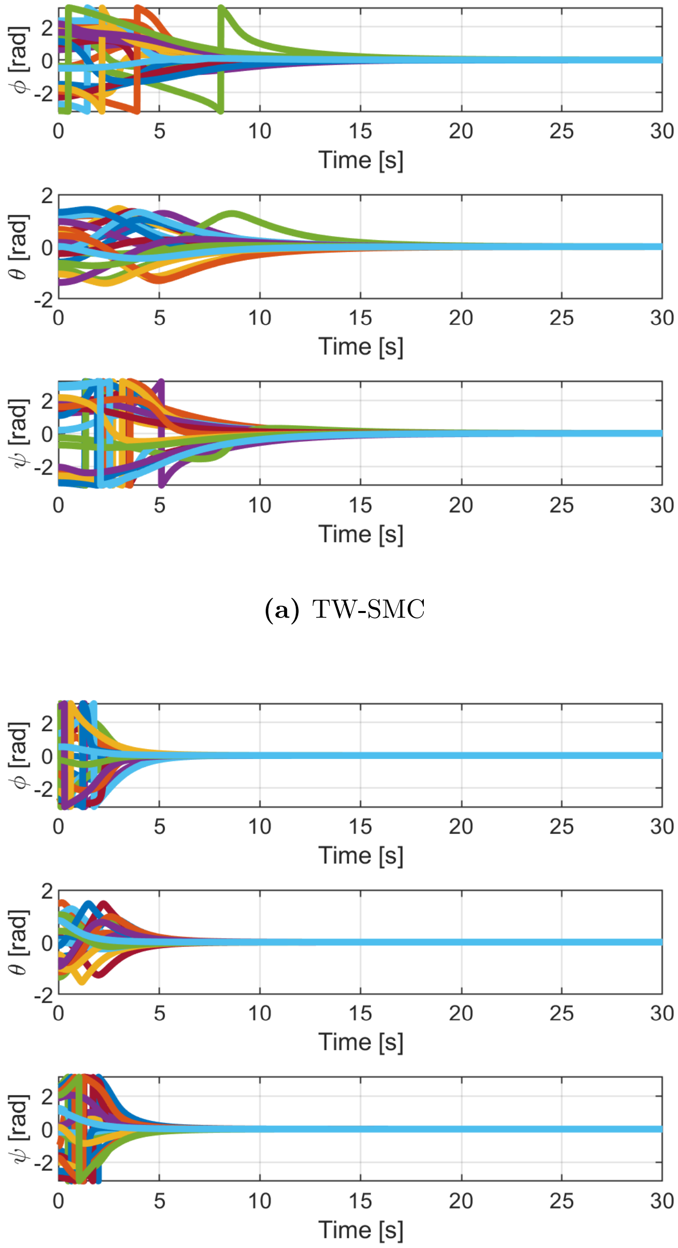
All the robustness tests, except the attitude changing, are considering variation of structural parameters to evaluate their influence in the manipulator motion stabilization.

Attitude changing task

Precision attitude control is the main goal of attitude controllers. After designing the controllers for the manipulator stabilization, several simulations with random initial attitude have been executed to test the controllers goal to restore the desired orientation.

Figures 5.13, 5.14 and 5.15 show the simulation results respectively with stowed and deployed manipulator and with a gripped object in the end-effctor position. The simulations results show the variation of Euler angles during time. Both the controllers are able to complete the task in acceptable time, but the backstepping controller reaches faster the desired attitude thanks to its adaptive feature, while the twisting sliding mode controller needs few more seconds, because it has been designed to achieve a trade-off between precision attitude control and manipulator stabilization.

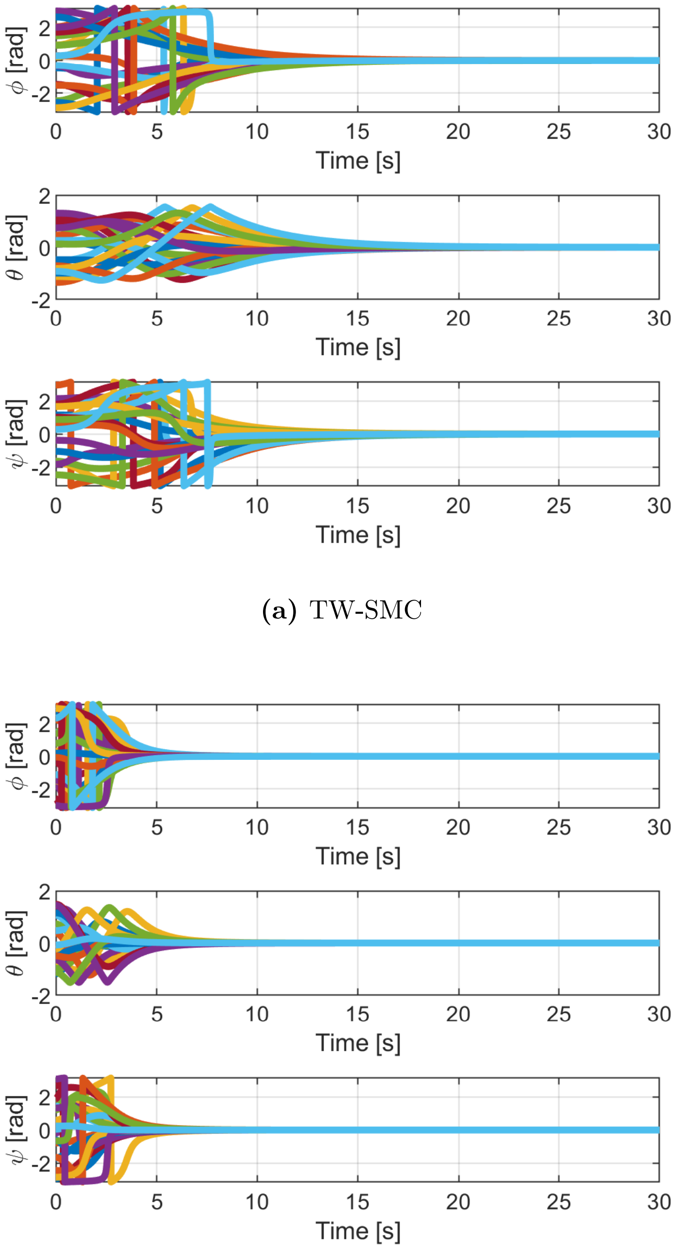
The gripped object test uses random value of object masses in the end-effector position to evaluate its effect for precision attitude control. Both the controllers are able to overcome the inertia changes due to the presence of the object achieving the desired results in short simulation times.



(a) TW-SMC

(b) Backstepping controller

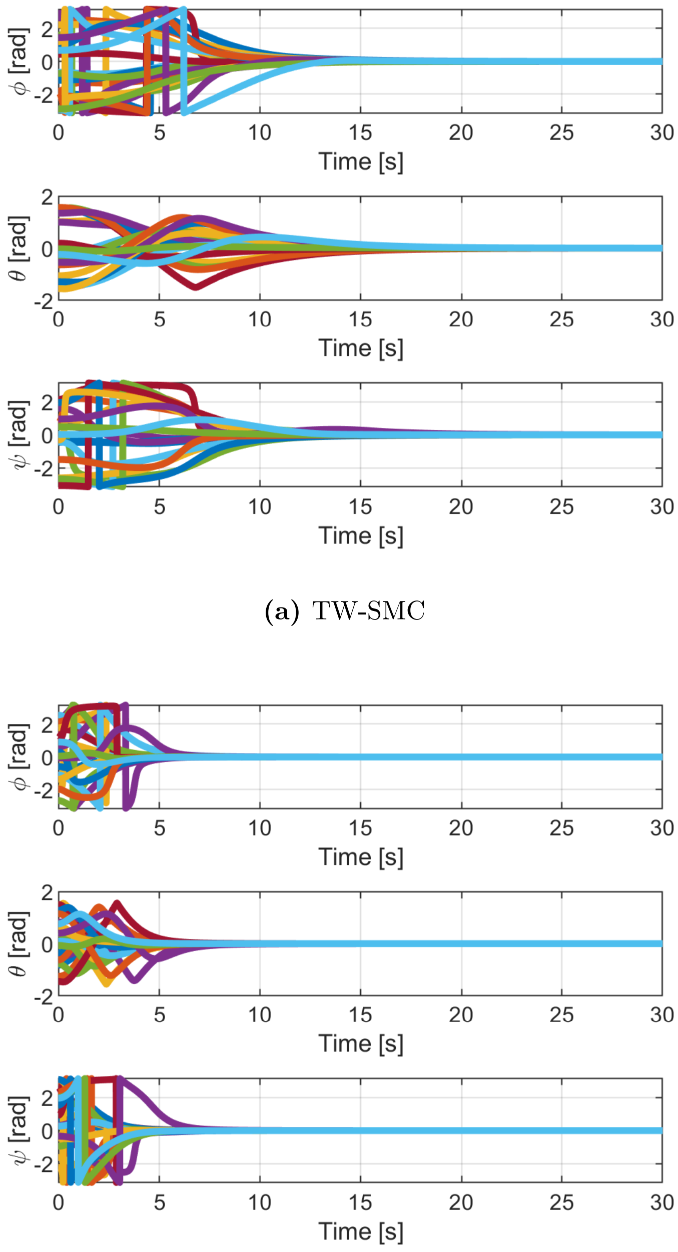
Figure 5.13: Attitude changing with stowed arm



(a) TW-SMC

(b) Backstepping controller

Figure 5.14: Attitude changing with deployed arm



(a) TW-SMC

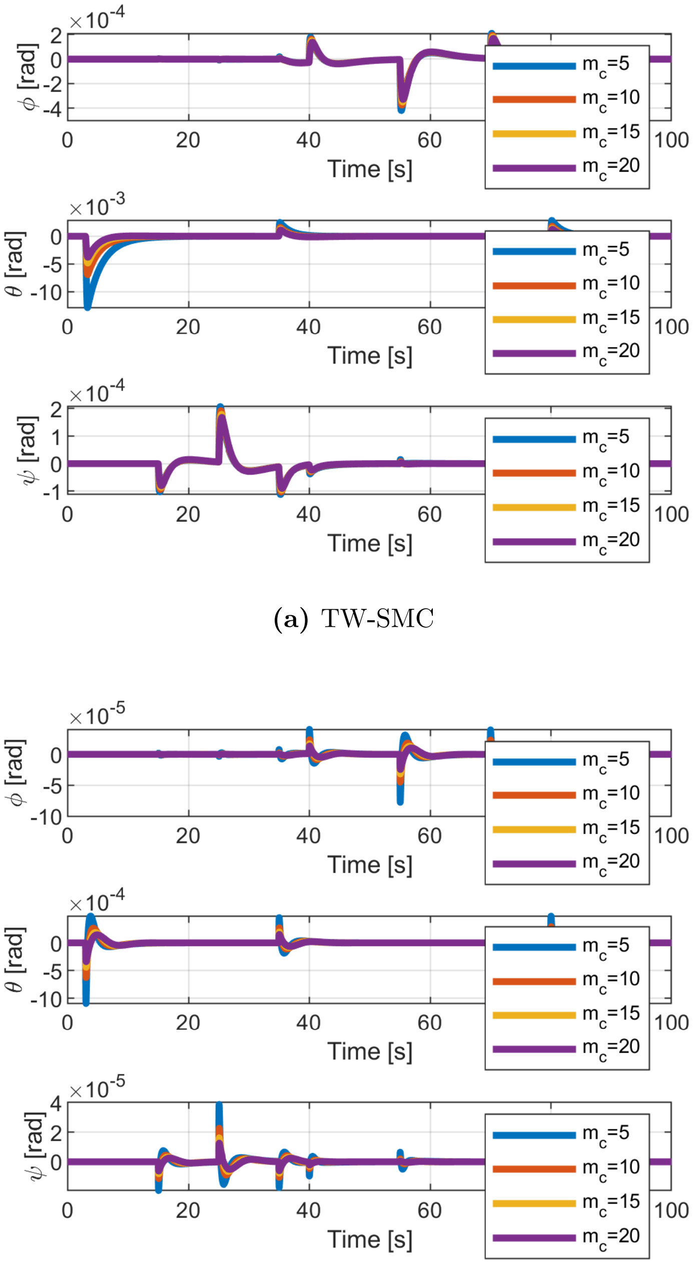
(b) Backstepping controller

Figure 5.15: Attitude changing with gripped object

Variable core mass

The main body structural mass and inertia have been changed to evaluate their influence for the controllers performance. Four different main body masses have been considered.
At the start of the simulation, the Astrobee is already placed in the desired attitude, and the full manipulator motion has been considered, as shown in Figure 5.4a.

Figure 5.16 shows the simulation results as the variation of the Euler angles during time. Both the controllers show an improvement of performance with the increasing of the main body mass. This is due to the improvement of the natural stabilization of the system, in fact the higher main body inertia reduces the effects of the inertia variation introduced by the manipulator motion, so the overall system is more stable.



(a) TW-SMC

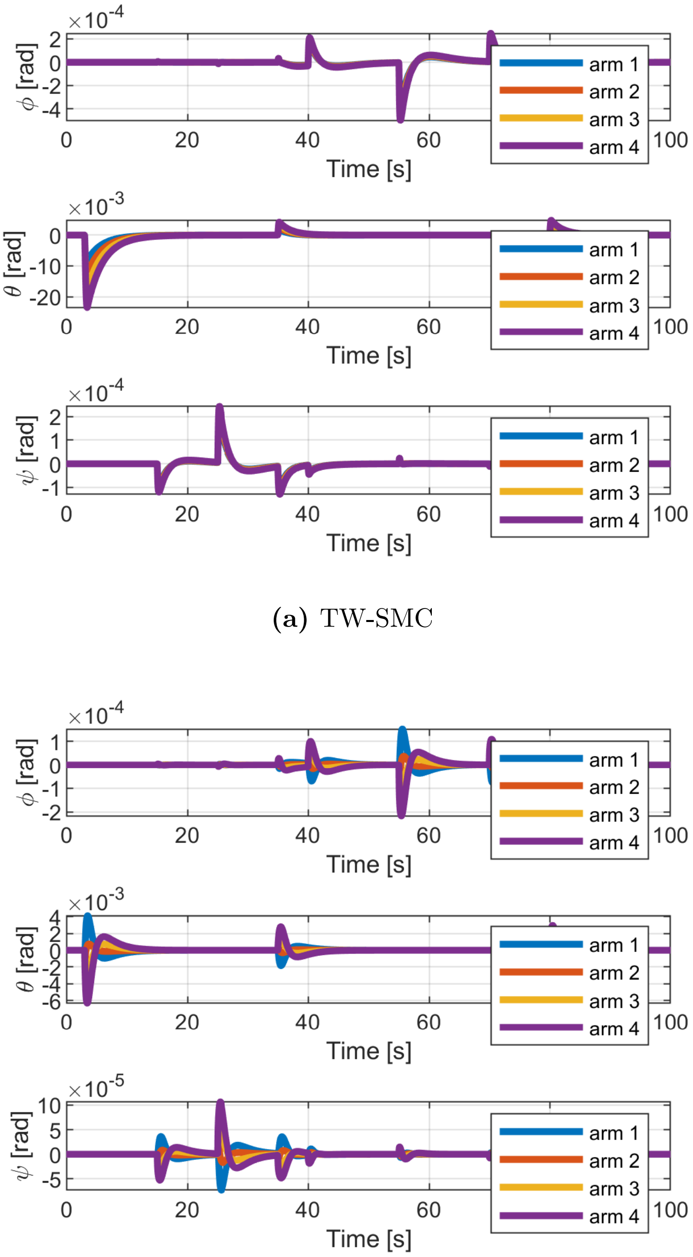
(b) Backstepping controller

Figure 5.16: Variable main body mass and inertia

Variable arm links mass

The manipulator links masses have been changed to evaluate its effect in the manipulator motion stabilization. Three configurations of heavier manipulators have been considered, and full motion shown in Figure 5.4a has been executed. At the start of the simulation the Astrobee is placed in the desired attitude, so the controllers' goal is to stabilize the manipulator motion.

Figure 5.17 shows the simulation results in term of Euler angles. The variation of the links masses doesn't produce significant changes in controllers' performance. Since the manipulator motion is handled by the first-order SMC, the maximum value of the disturbance torque doesn't change, so the main changes in the attitude dynamics are due to the inertia variation and gyroscopic torque introduced by the arm motion.
The TW-SMC performance reduces with the links masses increase, due to the higher disturbances. The backstepping controller estimates the disturbance torque cosidering the manipulator's real velocity and the inertia referred to the mathematical model of the Astrobee realized with the design parameters of the structure. Since the initial joint torque is the same, increasing the links masses leads to a slower manipulator motion. Thus, the backstepping controller estimates a slower disturbance torque for the computation of its output. This leads to an initial improvement of the performance, for links masses little heavier than the design value, and to a consecutive reducing.



(a) TW-SMC

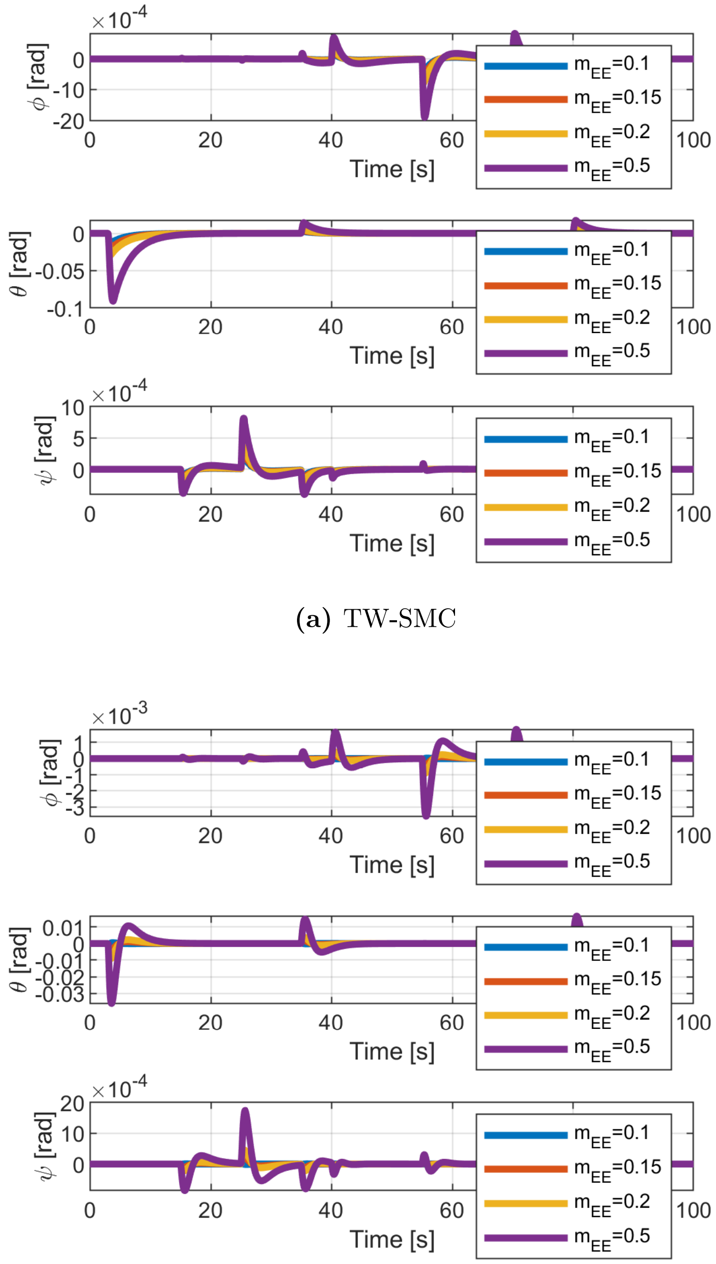
(b) Backstepping controller

Figure 5.17: Variable manipulator links mass

Variable end-effector mass

The end-effector mass has been changed to simulate the manipulator motion while an object is grabbed. Three heavier masses in the end-effector have been considered, and full motion shown in Figure 5.4a has been execute. At the start of the simulation the Astrobee is placed in the desired attitude, so the controllers' goal is to stabilize the manipulator motion.

Figure 5.18 shows the simulation results in term of Euler angles. Both the controllers are able to restore the desired attitude, but adding an object in the end-effector position will lead to higher errors during the manipulator motion. The TW-SMC suffers more the inertia variation and the gyroscopic torque introduced by the manipulator motion. The backstepping controller thanks to the adaptive feature is able to handle better the disturbances introduced by the object mass increasing.



(a) TW-SMC

(b) Backstepping controller

Figure 5.18: Variable end-effector mass

Chapter 6

NASA's Astrobee simulator

The NASA's Astrobee simulator has been used to validate MATLAB/Simulink simulation results. It is realized in ROS-Gazebo enviroment and uses the Astrobee CAD to simulate efficiently the physics in the micro-gravity environment of the ISS [22].
To perform the attitude controller tests, ROS-Simulink combined simulation has been realized. In this way, it is possible to get the information needed as controller input from Gazebo, and to applying the requested torque directly on the Astrobee's main body.

6.1 Simulink implementation

The MATLAB *ROS toolbox* is needed to allow the communication between the simulator and Simulink. The following blocks have been used to achieve the access to the model data and to send the desired commands to the simulator:

- The *Subscriber* block, in combination with the *bus selector*, allows to access to the output of the ROS topics;
- The *Publisher* block allows to send a message to a ROS topic.
- The *Call Service* block allows to execute a ROS service request.
- The *Blank Message* block, in combination with the *bus assignment*, allows to prepare the correct message or service request needed to run a ROS topic or service.

The *Subscriber* block has been used in the Simulink model to get the attitude and the angular velocity from the topics */loc/truth/pose* and */loc/truth/twist*. The two topics respectively send to Simulink the orientation of the main body, with the

manipulator pointing backward, and the angular velocity expressed in the body refernce frame.
The *Call Service* block has been used to call the ROS serivce */gazebo/ApplyBodyWrench* to apply the requested control torque on the Astrobee's body. The service request for this ROS service need the torque expressed in the inertial reference frame, so the computed control torque has been converted by using the rotational matrix

$$T_i = R(q)T_b \tag{6.1}$$

The ROS Service */gazebo/GetJointState* has been also used to get access to the manipulator's joints state.
Moreover, the *Real-time pacer* block has been added to slow down the Simulink simulation to synchronize Simulink and Gazebo's simulation time. Practically, the communication between Simulink and ROS slow down Simulink's speed, so the Gazebo simulation's speed is needed to be also set to low, but by running the ROS/Simulink co-simulation is impossible to perfectly synchronize the two simulations, so some errors of time delay can be produced.

Twisting sliding mode controller

TW-SMC implementation is strictly related to the sliding surface computation, which has been defined by using the quaternions and angular velocity errors.
The sliding surface can be computed reading the messages from the ROS topics */loc/truth/pose* and */loc/truth/twist*, and the command torque can be obtained from the control law:

$$u = -k_1 tanh(\sigma) - k_2 tanh(\dot{\sigma}) \tag{6.2}$$

Thus, it can be applied on the main body using the ROS serivce */gazebo/ApplyBodyWrench.*

Backstepping controller

The backstepping controller implementation requires further information rather than attitude and angular velocity to define the control law.
The mathematical model of the Astrobee can be used to define each parameter of the control law formulation in function of the joints' angles. Using the ROS service */gazebo/GetJointState* to get the arm joints data, it is possible to define each term of the control law, so that the command torque can be computed by:

$$u = -K_2\tilde{\omega} - \tilde{q_v} - T_{dis} - \gamma + [\omega_c \times]J\tilde{\omega} + \frac{1}{2}\dot{J}\tilde{\omega} \tag{6.3}$$

Thus, it can be applied to the Astrobee using the ROS serivce */gazebo/ApplyBodyWrench.*

6.2 Simulations results

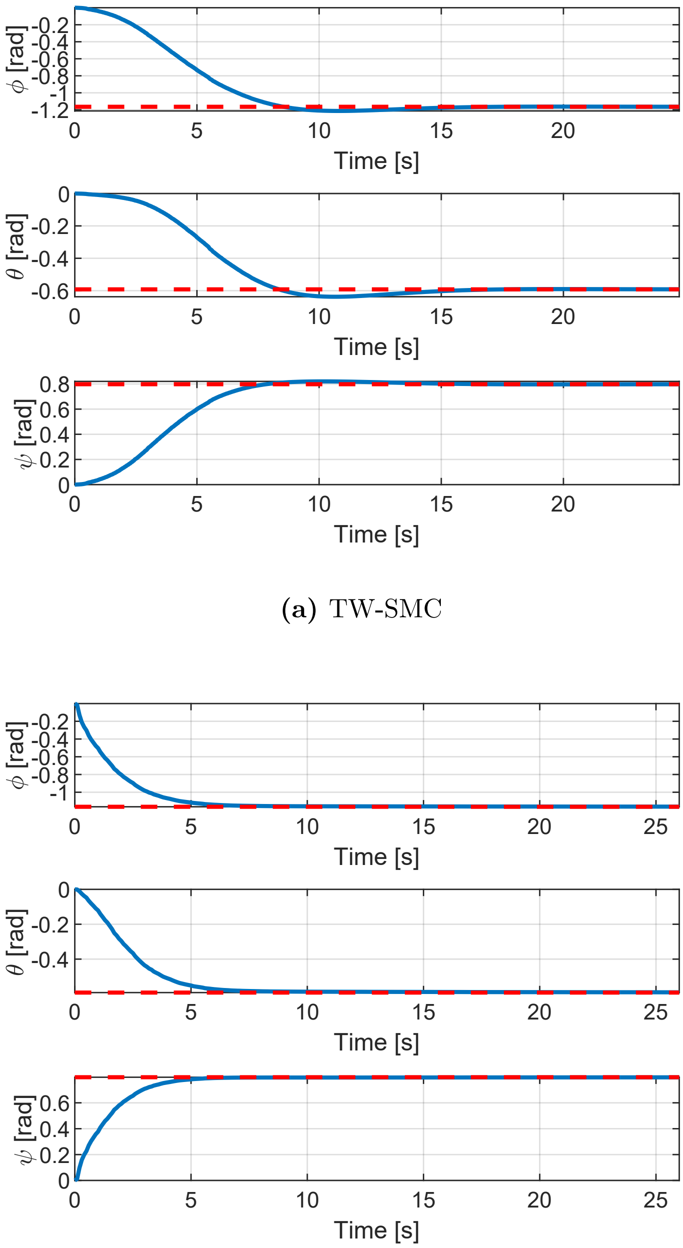
The NASA's Astrobee ROS/Gazebo simulator has been used in combination with MATLAB/Simulink to perform attitude control and stabilization simulations to evaluete the controllers performance in more accurate physical environment.
All the commands that interact directly with the simulator have been executed as command line instructions, while Simulink has been used exclusively as interface between the controller and the ROS/Gazebo simulator.

The Astrobee has been spawned inside the ISS module and oriented according to the inertial reference frame

$$q = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.4}$$

Attitude change

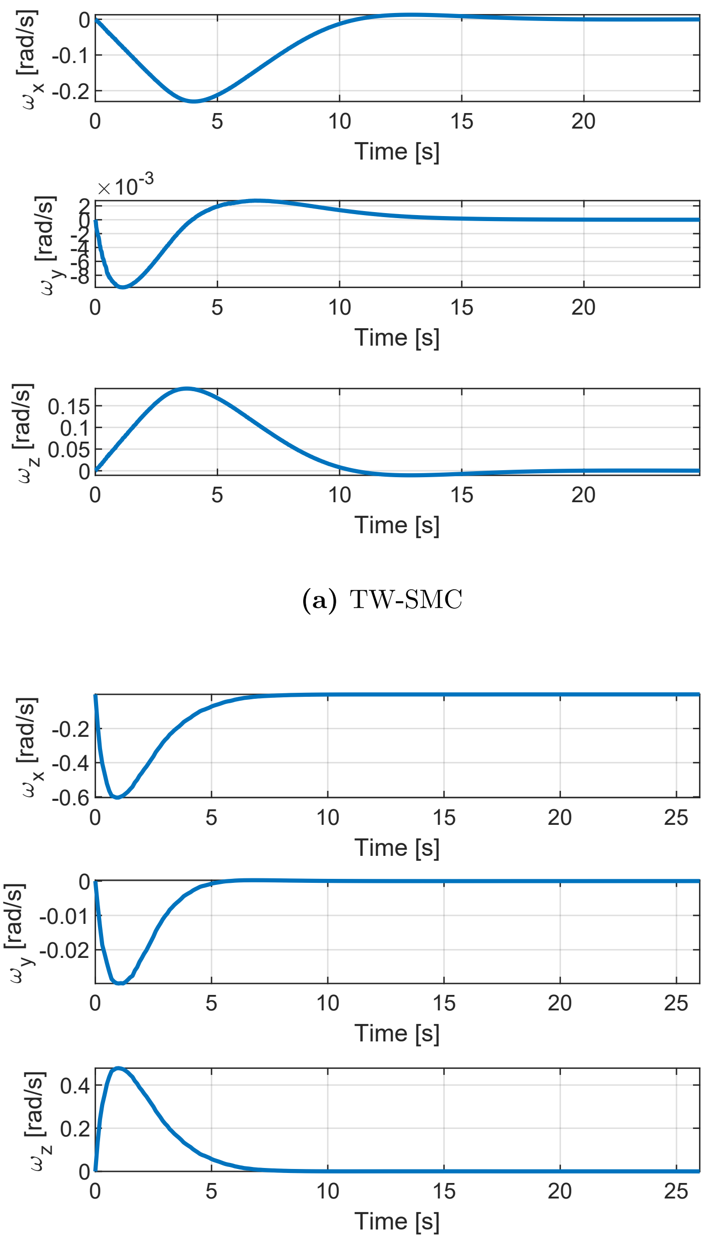
The attitude changing simulation has been computed requesting a random orientation to the controllers. Figure 6.1 shows the attitude changing results in terms of Euler angles. The TW-SMC and backstepping controller run two different simulations, but the results are well-matched with the expectation. The backstepping controller reacts faster to the attitute changing request reaching directly the desired orientation, while the TW-SMC results show some oscillations before completing the request.



(a) TW-SMC

(b) Backstepping controller

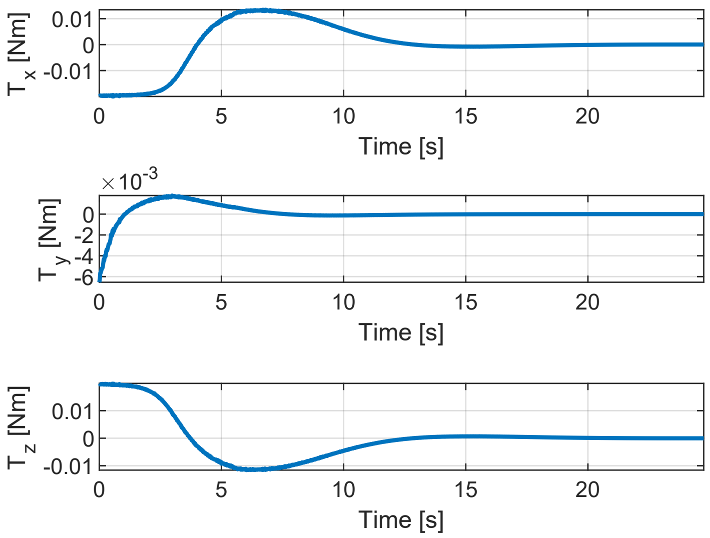
Figure 6.1: Euler angles - Attitude changing



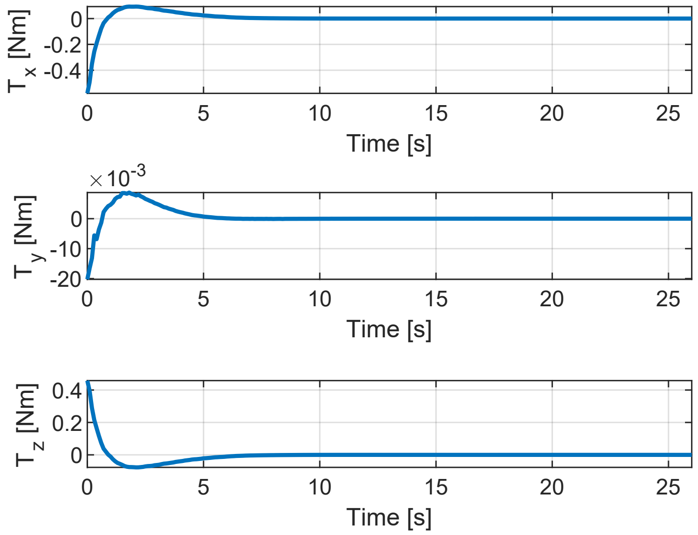
(a) TW-SMC

(b) Backstepping controller

Figure 6.2: Angular velocity - Attitude changing



(a) TW-SMC



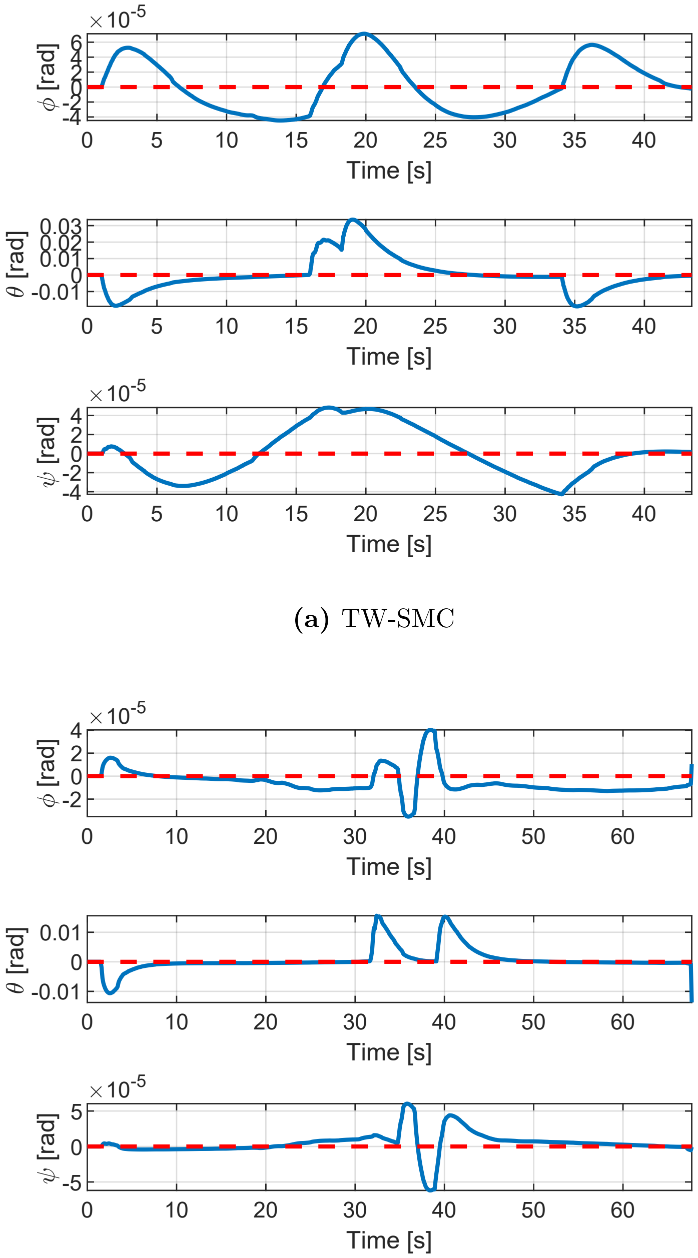
(b) Backstepping controller

Figure 6.3: Command torque - Attitude changing

Manipulator motion stabilization

The manipulator motion can be executed with the simulator instructions related to the *arm_tool.* The manipulator deploying and stowing have been considered, but since the motion is requested by a manual instruction the simulations have different execution times.

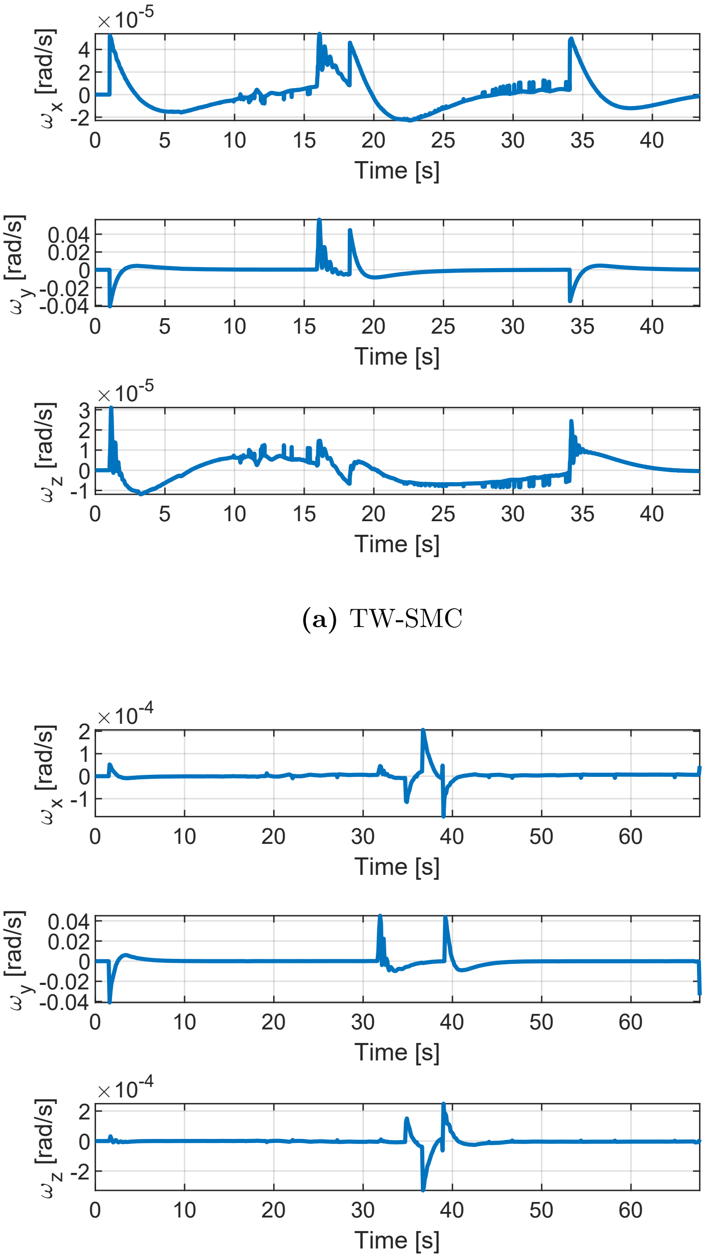
Since the high computational cost of the combined simulation, there are some delays in the arm deployment, especially for the backstepping simulation. In Figure 6.6 the effect of the delay of Simulink simulation respect to ROS/Gazebo one is much evident with TW-SMC, where there is the presence of the chattering effect. The results in terms of Euler angles, shown in Figure 6.4, have about the same magnitude of Simulink results, except for the backstepping controller that reports an error increasing probably due to the time delay error.



(a) TW-SMC

(b) Backstepping controller

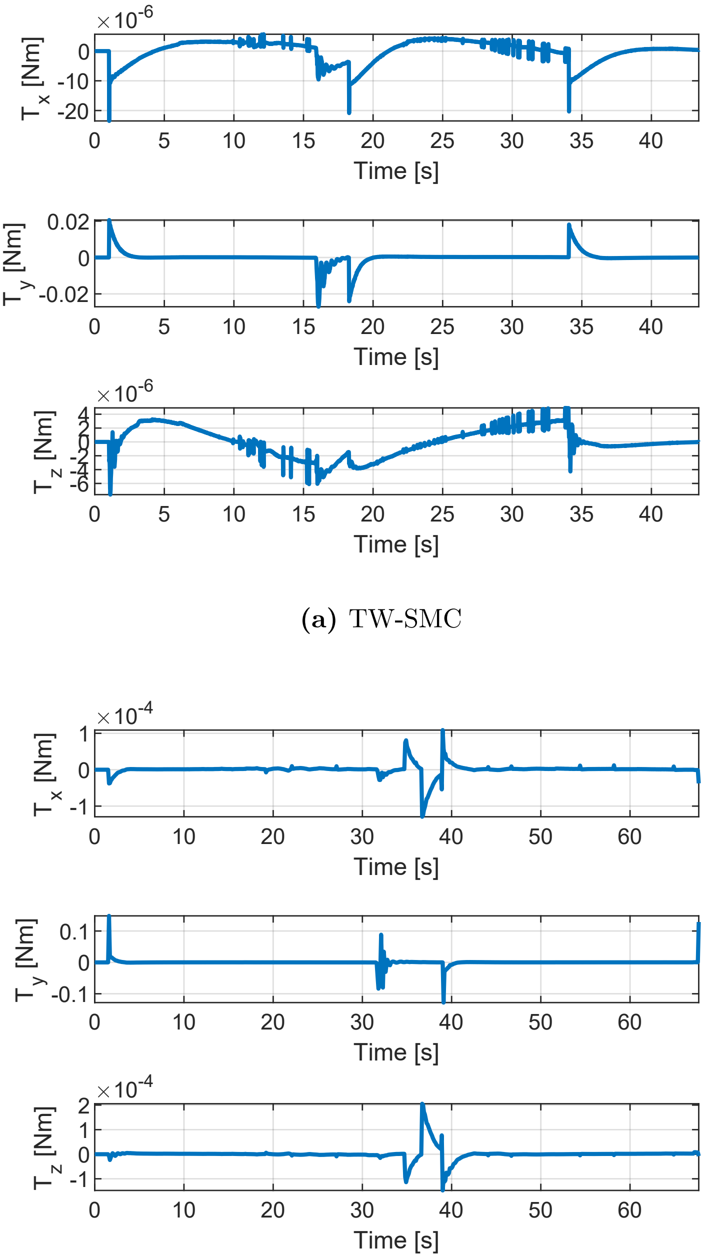
Figure 6.4: Euler angles - Manipulator motion stabilization



(a) TW-SMC

(b) Backstepping controller

Figure 6.5: Angular velocity - Manipulator motion stabilization



(a) TW-SMC

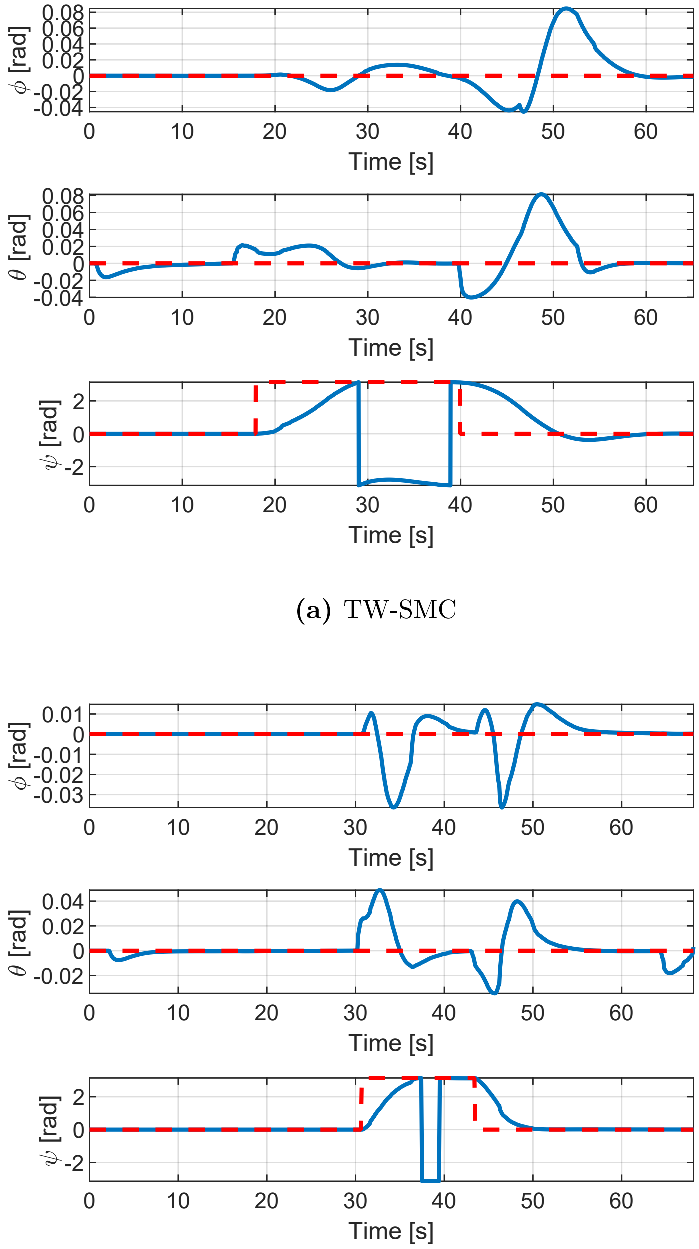
(b) Backstepping controller

Figure 6.6: Command torque - Manipulator motion stabilization

Combined motion

The combination of manipulator motion and attitude changing has been considered. The simulation consists in the manipulator deployment, rotation of the Astrobee of 180° around z-axis, and the restoration of the initial attitude while the manipulator is being stowed.

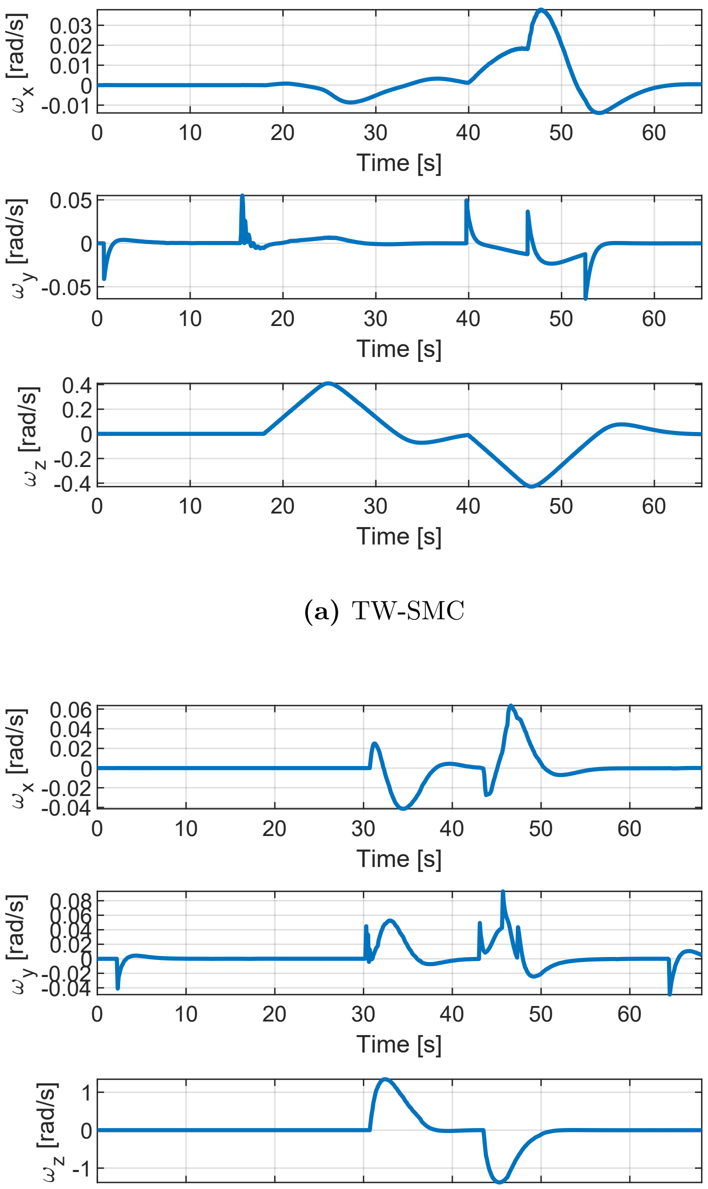
The high computational cost reports some errors due to the time delay between the two simulations, especially when the manipulator is going to stop its motion. During the rotation around the z-axis some oscillations are produced around y and x axis. Figure 6.9 shows the controllers output. In this simulation, the chattering effect introduced by the time delay is less evident, but, especially for the backstepping controller, it can be noticed the presence of unjustified peaks, due to the controller delays.
In Figure 6.7 the Euler angle are shown. As expected, the backstepping controller reacts faster than TW-SMC reducing the errors introduced by the manipulator motion.



(a) TW-SMC

(b) Backstepping controller

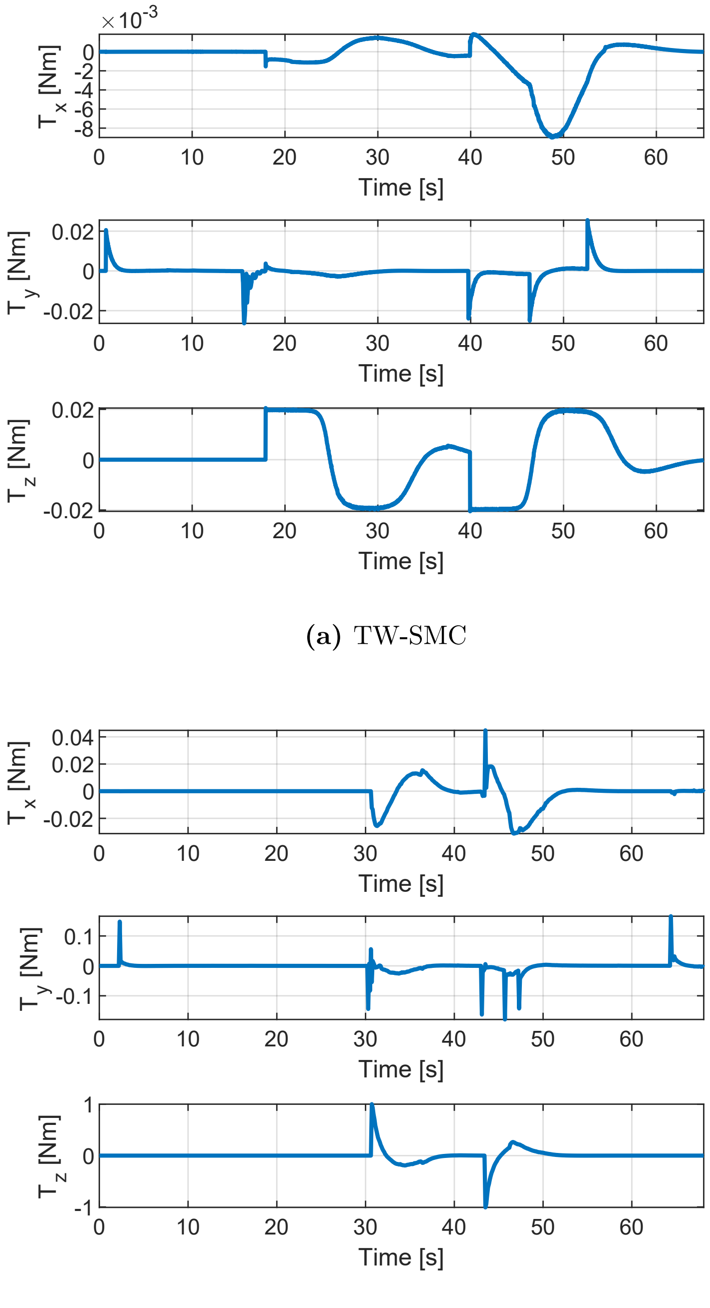
Figure 6.7: Euler angles - Combined motion



(a) TW-SMC

(b) Backstepping controller

Figure 6.8: Angular velocity - Combined motion



(a) TW-SMC

(b) Backstepping controller

Figure 6.9: Command torque - Combined motion

Chapter 7

Conclusions and future work

The objective of this research is to propose an attitude control strategy to exploit the full functionality of the Astrobee's manipulator. The actual robotic arm has been designed for handrail gripping and camera pointing, since its motion during free-flying leads to important inertial changes and disturbances. Designing a robust attitude controller able to overcome the disturbances introduced by the manipulator moving during free-flight makes the Astrobee able to perform more challenging tasks like object gripping and carrying.

An accurate model of the Astrobee system has been obtained neglecting the centre of mass variation during the manipulator motion. The application of TW-SMC or the backstepping controller is a suitable solution for the manipulator motion stabilization and precision attitude control. From the results obtained with MATLAB/Simulink simulations, the backstepping approach seems to be more flexible and precise for the attitude changing request, but both the controllers are able to reach the desired orientation in acceptable times, also when an object is gripped. The Astrobee's attitude is very well preserved against the manipulator's reactions, except with a gripped object, in fact, it is more challenging task to stabilize the disturbances, since the relevant ones induced during a heavier object motion.
The NASA Astrobee simulator has been used in collaboration with Simulink to run ROS-Simulink combined simulations. In this way, the results obtained for attitude changing and arm stabilization have been verified, despite synchronization errors induced by the combined simulation.

However, the Astrobee model is considering the controller ideal torque request without the implementation of the actuator model. Thus, the first step for a possible future work is to realize the actuators model to evaluate the controllers performance in the real scenarios. At this point the combination between the attitude and position control can be executed to evaluate the overall performance of the Astrobee system in more challenging tasks, like object carrying, or by building a more accurate model considering the possible disturbances introduced from actuators failures and transitory effects in position and attitude control.

Bibliography

[1] Trey Smith, Jonathan Barlow, Maria Bualat, Terrence Fong, Christopher Provencher, Hugo Sanchez, Ernest Smith, and The Team. «Astrobee: A New Platform for Free-Flying Robotics Research on the International Space Station». In: June 2016 (cit. on pp. 1, 5).

[2] I. N. Ibrahim, M. A. Al Akkad, and I. V. Abramov. «Attitude and altitude stabilization of a microcopter equipped with a robotic arm». In: *2017 International Siberian Conference on Control and Communications (SIBCON)*. 2017, pp. 1–8 (cit. on p. 1).

[3] M. Oda. «Coordinated control of spacecraft attitude and its manipulator». In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 1. 1996, 732–738 vol.1 (cit. on p. 1).

[4] L. Shi, J. Katupitiya, and N. Kinkaid. «A robust attitude controller for a spacecraft equipped with a robotic manipulator». In: *2016 American Control Conference (ACC)*. 2016, pp. 4966–4971 (cit. on p. 1).

[5] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding mode control and observation*. Springer, 2014 (cit. on pp. 2, 10).

[6] MA SI Mohammed, H Boussadia, A Bellar, and A Adnane. «Adaptive backstepping control for three axis microsatellite attitude pointing under actuator faults». In: *Journal of Physics: Conference Series*. Vol. 783. 1. 2017, p. 012020 (cit. on pp. 2, 15).

[7] Maria Bualat, Trey Smith, Ernest Smith, Terrence Fong, and Dw Wheeler. «Astrobee: A New Tool for ISS Operations». In: May 2018. DOI: 10.2514/6.2018-2517 (cit. on pp. 2–5).

[8] Alvar Saenz-Otero and David W. Miller. «SPHERES: a platform for formation-flight research». In: *UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts II*. Ed. by Howard A. MacEwen. Vol. 5899. International Society for Optics and Photonics. SPIE, 2005, pp. 230 –240. DOI: 10.1117/12.615966. URL: https://doi.org/10.1117/12.615966 (cit. on p. 2).

[9] Maria Bualat, Jonathan Barlow, Terrence Fong, Chris Provencher, and Trey Smith. «Astrobee: Developing a Free-flying Robot for the International Space Station». In: Aug. 2015. DOI: 10.2514/6.2015-4643 (cit. on p. 5).

[10] In-Won Park, Trey Smith, Hugo Sanchez, Sze Wun Wong, Pedro Piacenza, and Matei Ciocarlie. «Developing a 3-DOF compliant perching arm for a free-flying robot on the International Space Station». In: *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2017, pp. 1135–1141 (cit. on p. 5).

[11] Gourab Nandy, Basukinath Chatterjee, and Amartya Mukherjee. «Dynamic analysis of two-link robot manipulator for control design». In: *Advances in Communication, Devices and Networking*. Springer, 2018, pp. 767–775 (cit. on p. 6).

[12] L. Crassidis F. Landis MarkleyJohn. «Fundamentals of Spacecraft Attitude Determination and Control». In: Springer, 2014 (cit. on p. 6).

[13] Vadim I Utkin. *Sliding modes in optimization and control problems*. Springer Verlag, New York, 1992 (cit. on p. 10).

[14] C.Novara. «Non linear control and aerospace application class slides». In: Tech. rep. Politecnico di Torino,Torino,Italy. 2020 (cit. on p. 11).

[15] A. Bartoszewicz. «A new reaching law for sliding mode control of continuous time systems with constraints». In: *Transactions of the Institute of Measurement and Control* 37.4 (2015), pp. 515–521 (cit. on p. 12).

[16] A. Levant. «Sliding order and sliding accuracy in sliding mode control». In: *International journal of control* 58.6 (1993), pp. 1247–1263 (cit. on p. 12).

[17] A. Levant. «Introduction to high-order sliding modes». In: School of Mathematical Sciences,Israel. 2003 (cit. on pp. 12, 13).

[18] G. Ferrari Trecate. «Nonlinear systems: Design of nonlinear controllers class slide». In: Università degli Studi di Pavia,Italy (cit. on p. 13).

[19] Ola Härkegård and S Torkel Glad. «Flight control design using backstepping». In: *IFAC Proceedings Volumes* 34.6 (2001), pp. 283–288 (cit. on p. 15).

[20] Filippo Caldera. *Sliding mode attitude control for a small satellite with a manipulator*. Master Thesis, Politecnico di Torino, 2019 (cit. on p. 16).

[21] Stefano Di Lucia, Gian Diego Tipaldi, and Wolfram Burgard. «Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach». In: *2015 European Conference on Mobile Robots (ECMR)*. IEEE. 2015, pp. 1–6 (cit. on p. 20).

[22] «NASA Astrobee Robot Software». In: ROS-based open source software. URL: https://github.com/nasa/astrobee (cit. on p. 47).