



Università degli Studi di Trento
Department of Industrial Engineering

Master Degree in
Mechatronics Engineering

FINAL DISSERTATION

Dynamics of a Space Robot Subjected to
an Impact

Supervisor:
Daniele Bortoluzzi

Co-supervisor:
Edoardo Dalla Ricca

Student:
Simone Manfredi 239337

Academic Year 2024/2025

Abstract

This thesis aims to describe, control and simulate a planar revolute-revolute (RR) manipulator in absence of gravity (i.e. located in space) subjected to an impact due to the docking of a tumbling satellite.

The actual outputs of this work are methods to retrieve the satellite mass and initial velocities assuming them unknown in advance. Therefore, an initial guessed mass is implemented in the control law initially.

This work is organized as follows: the first chapter exposes an overview of the actual state-of-art of modern space manipulators.

The second chapter introduces the basics of robotic kinematics and the manipulator jacobian matrix is found.

In the third chapter the system equation of motions are calculated with lagrangian-based approaches.

The impact analysis is carried on in the fourth chapter, where the arms are modeled initially as rigid bodies and afterwards as Euler-Bernoulli beams.

Satellite's mass and initial velocity retrieval methods are proposed in the fifth chapter, along with the found results.

Methods' comparison and considerations, together with possible further developments are listed in the sixth and last chapter.

Contents

Contents	i
Introduction	iii
1 Space Manipulators: an Overview	1
1.1 Historical background	3
1.2 Main characteristics	5
1.3 State-of-art	8
1.3.1 SSRMS	9
1.3.2 SPDM	10
1.3.3 JEMRMS	10
1.3.4 ERA	12
1.3.5 RMSs Comparison	12
2 Kinematics	14
2.1 Fundamental of Kinematics	16
2.1.1 Roto-Translation Matrix	16
2.1.2 Velocity Matrix	19
2.1.3 Danavit-Hartenberg Method	21
2.1.4 Jacobian Matrix	22
2.2 Planar Vehicle-Manipulator System	23
2.2.1 Object's Kinematics	27
3 Dynamics	29
3.1 Classic Approach	31
3.1.1 Object's Dynamics	35
3.2 Homogeneous Matrices Approach	36
3.2.1 Object's Dynamics	39
4 Impact Analysis	41
4.1 Rigid Bodies	43
4.1.1 Free Motion	45

4.1.2	Control Based Motion	50
4.2	Elastic Bodies	55
4.2.1	Free Motion	62
4.3	Control Based Motion	66
5	Inertial and Kinetics Properties Retrieval	71
5.1	Mass Extraction: Joints' Decoupling	73
5.1.1	Direct Solution	75
5.1.1.1	Position Roots	75
5.1.1.2	Mass Fit	75
5.1.2	Closed Form Solution	77
5.1.2.1	Derivative Roots	77
5.1.2.2	Damping Coefficient and Natural Frequency Fit . .	78
5.2	Mass Extraction: Coupled Solution	79
5.2.1	Continuous Domain	79
5.2.2	Discrete Domain	80
5.3	Velocity Extraction	81
6	Results and Conclusion	82
6.1	Mass Retrieval	83
6.2	Velocity Retrieval	92
6.3	Results' Analysis	95
6.4	Final Considerations	96
6.4.1	Further Developments	97
6.4.2	Conclusion	99
Appendices		100
A	Matrices	101
B	Code	103
C	Plots	106

Introduction

We live in a society overwhelmed by such technology, autonomous systems, industrial manipulators, AI and, very soon, services' humanoid robots, that we forget to look up in the sky. Most of the major progress in robotics and the scientific field is done in space, where human curiosity is pushing our abilities beyond any imaginable scenario.

Bilionaires such as Elon Musk are pushing the technology even further with Mars colonization as a goal.

However, this great chapter of human history is still quite hidden. When someone says that he is studying Law, Economics or Literature, nobody bats an eye; but when I tell people that I study Mechatronics Engineering, I always have to add some more information, key words like "mechanics", "robotics" or "programming". In my opinion, this makes clear that one the most revolutionary fields (even though I aknowledge that it is a very broad term) is not really interiorized by the people, like others faculties and studies are.

In their defence, I must say that when I first enrolled in the university, aiming to the Mechatronics Master's Degree, I was more pushed by the unknown around that term than by real interest or passion related to it. But with the new available technology, such as the aforementioned AI, directly in our homes (and soon maybe also humanoid robots) it will become a common topic and an everyday experience. Yuval Noah, in his book "Sapiens: a Brief History of Humankind" [26] claims that three were the great human revolutions that completely changed our way to live and to act as social creatures: the Cognitive Revolution (around 70.000 years ago), which allowed us to think of nonexistent concepts, such as states, moral, legends, myths, tools; the Agricultural Revolution (about 12.000 years ago) when we started to settle and grow our own food; the Industrial Revolution (200 y.a.), which is fairly known to everybody. It is not difficult to think that the next human revolution, if there is ever going to be one, will have to do with space exploration.

This is to say that it is a really promising and exiting field, which will (and it is already doing) play an important role in our society.

Far from aiming to the moon, this thesis was born due to my curiosity in the field of robotics merged with the need to face space and modern problems, such as the

berthing and docking of external objects by a manipulator or the tumbling of a spinning satellite.

This dissertation aims to analyse the dynamic of a space manipulator after the impact with an external satellite to retrieve its kinetic and inertial properties before the impact.

The workflow was the following: a first phase committed to gather information about space manipulators and to study papers regarding the useful impact equations to model the docking of the satellite.

Secondly, a 2D model of the manipulator has been implemented in Wolfram 14.2 and equation of motion for the pre-impact and post-impact phases have been calculated. Finally, the results have been listed and considerations have been made.

The machine on which the calculation have been made is a MacBook Air M3 with 16 GB of RAM.

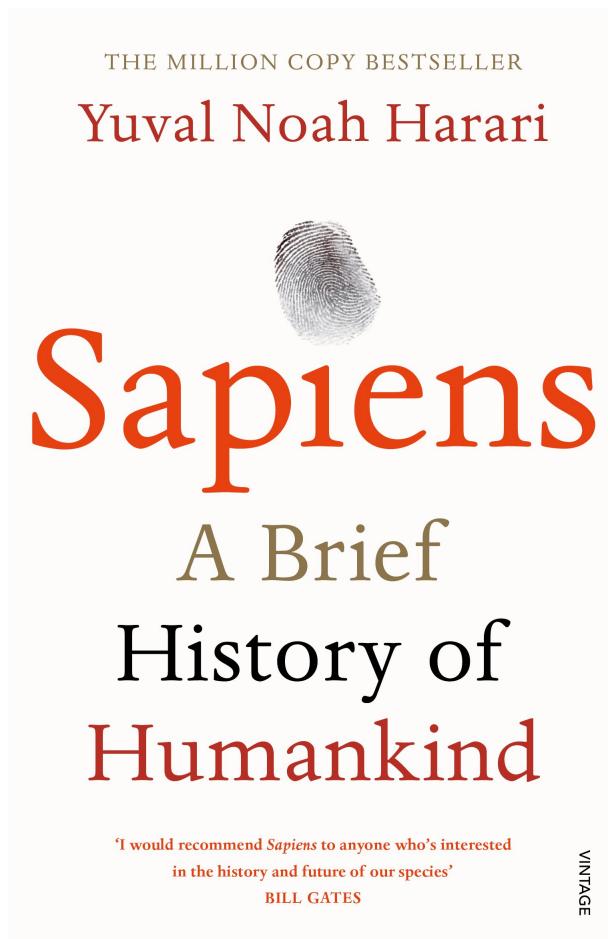


Figure 1: Yuval Noah's book cover.

Chapter 1

Space Manipulators: an Overview

Space Robotics is important to human's overall ability to explore or operate in space. Autonomous systems can reduce the cognitive load on humans and improve human and systems' safety.

Space robots can be split into two main categories: orbital robots and planetary robots. The formers can be used for repairing satellites, assembling large space telescopes, capturing and returning asteroids, etc; the latters play a key role in surveying, observations, extraction, working on planetary surfaces for subsequential human arrival, etc.

Furthermore, spacecraft are classified as space robots when two attributes are satisfied: locomotion and autonomy. Depending on its application, a space robot is designed to possess locomotion to manipulate, grip, rove, drill and/or sample; at the same time, it is expected to possess varying levels of autonomy, ranging from teleoperation by humans to fully autonomous operation by the robot itself [3].

Another classification of robotic devices in space could be represented by mobile robots, flying robots, and manipulators. Manipulators are used on board spacecraft (Space Shuttle) or space stations [5]. According to [1], a very loose definition calls every unmanned space probe a robotic spacecraft, referring to the challenges of largely autonomous operation in a complex mission. But even discarding this and focussing on space robotics elements in a more narrow sense (systems involving arms for manipulation or some kind of locomotion device for mobility and having the flexibility to perform varying tasks), there is a wide array of uses in the scope of space missions, giving rise to challenging problems and ingenious solution.

Robots and manipulators in space are designed to perform the following operations:

- capture
- manevring
- berthing/unberthing

- support of Extra-Vehicular-Activities (EVA)
- positioning and release

One of the most important features of all robots deployed in space is their flexibility. Robots and manipulators used in space are built out of light materials, and very often, their links deflect.

There are quite a number of on-orbit applications requiring advanced robotics capabilities, which are envisaged to take place in the 2025-2035 timeframe. The operators for these missions may range from space administrations to national governments to businesses. The following mission foci are envisaged: space debris removal, rescue mission, planned orbit raising, inspection/support to deployment, deployment/assembly aid, repair, refuelling and orbit maintenance, mission evolution/adaptation, lifetime extension, and re/de-orbiting. The International Space Station (ISS) continues to represent an excellent opportunity for scientific experiments to be conducted in space amid the unique characteristics, constraints and pressures that the environment brings [3].

But what is so special about a space robot? In many ways, robot systems for space applications are very different from the more familiar terrestrial robots, be they industrial robots in production automation or the newer kind of "service robots". One of the main and simplest differences is the high development and manufacturing costs and the missing "economy of scale" due to the experimental prototype nature of these developments.

This thesis will investigate space manipulators' dynamics, whose tasks involve berthing and docking satellites or managing tumbling meteorites; thus, a historical summary of their evolution and the actual state-of-art of this field will be briefly exposed.

1.1 Historical background

The first robot used in space was the Remote Manipulator System (RMS), also called *Canadarm*, developed by the Canadian Space Agency and mounted on the Space Shuttle. This robot was used for the first time on 12th November 1981 in the Columbia Shuttle and it was retired in July 2011. It was 15 meters long and had three hinged joints for pitch and three others for yaw and roll, which makes a total of six degrees of freedom (DOF).

There were numerous problems related to the use of the RMS in space. One of the most important was related to its positioning: since it is built out of light materials to minimise the launch cost, its links deflect substantially. When the arm is accelerated and stopped, large vibrations occur.

There are two main methods of improving the positioning accuracy of the RMS: passive and active. The passive methods focus on redesigning the manipulator and applying different materials. The active ones refer to a preshape input and to the usage of position, velocity or force feedback.

Eventually, some lessons have been learnt during the RMS' lifetime: the speed of manipulation should be drastically improved; the positioning accuracy is one of the most important issues in day-to-day operations; the new manipulators should be designed to include passive vibration control capabilities; active vibration control should be added in order to improve the positioning accuracy and shorten the settling time; the force loop could add significantly to the capabilities of the manipulator [5].

It is interesting to note that the actuators are not powerful enough to lift even the weight of the arm itself when on Earth. The Canadarm has been used for over 100 operations and is normally used for repairing, retrieving, and deploying satellites; assisting humans during extravehicular activities; and for remote inspection tasks [6].

Another early mission including a robotic arm in space was the MIR space station. This Soviet Union/Russian space station was the first modular space station and was in operation from 1986 to 2001. MIR was a microgravity research laboratory which conducted research in biology, astronomy, meteorology, and physics, to name a few. Four of the modules were equipped with the Lyappa robotic arm used for assembling the modules of the space station.

In 1997 the National Space Development Agency of Japan (NASDA) launched the Engineering Test Satellite No. 7 (ETS-VII), the first ever satellite to be equipped with a robotic arm. The ETS-VII performed several successful docking operations using the manipulator arm. The main objectives of the project were performance evaluation of a satellite-mounted robotic system; coordinated control of the satellite attitude and robot arm; teleoperation of the robot arm; demonstration of in-orbit

satellite servicing.

The project demonstrated the successful execution of experiments which gained insight into operations such as docking, fuel transfer, assembling, and berthing. Several successful docking operations were performed with the chaser and target satellite.

The Japanese also developed a robotic arm which was mounted on the Japanese Experiment Module at the International Space Station (ISS), the JEMRMS, in March 2008. The Small Fine Arm (SFA) was added in July 2009 (see Section 1.3). A larger and more advanced version of the SRMS arm is the MSS arm (Mobile Servicing System), which is mounted on the International Space Station. The MSS arm consists of a mobile base, a more advanced version of the Canadarm, called the *Canadarm2*, or SSRMS, and dexterous manipulator hand [6]. It has seven DOF, unlike the RMS, making it a redundant manipulator to avoid singularities in some specific but important positions. Its properties will be discussed in the next Section.

[5]

In March 2008 the last part of the Mobile Servicing System was added to the International Space Station. The Canada Hand, or the Special Purpose Dexterous Manipulator (SPDM), is a two-armed robot but is often referred to as a hand because it can be attached to the Canadarm2 robotic arm and taken to any location on the space station [6].

The last manipulator in space was launched in July 2021: the European Robotic Arm (ERA) serves as main manipulator on the Russian part of the Space Station. Clearly, the usage of manipulator in space involves recent activities, hence a lot of research and experiments are still in development. This research field offers numerous starting points in analysis and technologic progress.

1.2 Main characteristics

Orbiting robotic systems consist of a free-floating base, typically a satellite or a space station, with a robotic manipulator attached to it.

The main difference between space manipulators and Earth-based robots is the lack of a fixed base on which the robot is mounted.

Depending on whether or not actuation is utilized to control the spacecraft position and orientation, we can divide these systems into *free-flying* and *free-floating* space manipulators.

In a free-flying robotic system the position and orientation of the base, in this case the spacecraft, is actively controlled by the spacecraft's actuators. This allows to completely control both the base configuration and the manipulator arm.

Due to the dynamic coupling between the manipulator and the spacecraft, the motion of the manipulator arm will constantly affect the motion of the base. The main concern with these systems is therefore the excessive fuel consumption required to compensate for the dynamic coupling between the manipulator and the base while maintaining a constant base attitude [6].

This mode is employed during the final approach of a manipulator to its target, so that the target is within the manipulator workspace [2].

A free-floating robotic system, on the other hand, does not use spacecraft actuation to compensate for the manipulator motion. The spacecraft motion is therefore not controlled directly, but arises as a result of the dynamic coupling between the manipulator and the base. In this case one can either choose not to control the spacecraft motion at all (as it has been done in this work), or to use the manipulator arm to obtain the required motion also for the spacecraft. In many cases it is necessary to generate a manipulator motion which guarantees that the spacecraft orientation remains almost constant so that antennas and other instruments point in the right direction [6].

In this mode, the spacecraft attitude is controlled actively with momentum control devices (MCD), such as reaction wheels or momentum gyros, while the system CoM does not translate. The free-floating or the partial free-floating modes are preferred during grasping, since they eliminate sudden motions due to thrusters, and conserve propellant and power [2].

Even though space exploration is challenging compared to Earth-based systems, space manipulator design also benefits from the free-fall environment in some areas. It is, for example, possible to construct robots with extremely high redundancy and with several joints. This type of robots, are able to support their own weight in space due to the small gravitational forces. On Earth, however, such robots would collapse due to their own weight when the number of joints becomes too large. This allows for more redundant robots in space than on Earth, and also

more fault tolerant robots because the robots can continue operation even after several joint failures. Similarly, the free-fall environment allows for effective control with very small actuators. In fact, most space manipulators are able to handle very high payloads in space, while they cannot even withstand the weight of the manipulator arm itself if they were to be placed in the Earth’s gravity field [6].

To execute on-orbit tasks being inaccessible to, or too dangerous for humans, robotic on-orbit servicing (OOS) can be employed, with tasks to be performed by space manipulator systems (SMSs), also called chasers or servicers in the literature. An SMS consists of a satellite base equipped with one or more robotic manipulators (arms) with grappling devices on them and driven by a vision system which allows them to capture a target (client) satellite, or another object. An SMS also can be a large servicing manipulator mounted on a space facility.

Targets for capture may be cooperative, i.e., a stable and safe target due to its operational Attitude and Orbit Control Subsystem (AOCS), or non-cooperative i.e., an unknown or tumbling target with a varying axis of rotation. They can also be collaborative, i.e., designed for capture or servicing, equipped with visual markers and grapple fixtures, or non-collaborative, as most of today’s satellites. In many cases in the literature, the term cooperative stands for collaborative, too. As often revealed by ground observations, many on-orbit objects are tumbling in an uncontrolled way (non-cooperative targets), making the robotic capture a very challenging task. This is the case analysed in this thesis.

Clearly, only after a manipulator has successfully captured and stabilized a tumbling target, can a service operation be started. Therefore, a common robotic capture task for on-orbit servicing consists of four operational phases:

1. observation and planning phase;
2. final approach phase;
3. impact and grasping/capture phase;
4. post-capture stabilization phase.

With a target locally stationary, the approach phase can be achieved by point-to-point planning and depending on the actuation mode, by simple on-off thruster control, and attitude fine-tuning using momentum exchange devices. Capturing a tumbling, non-cooperative target is more challenging, as here velocity matching between the SMS end-effector and the capture point is required.

In fact, to avoid impacts during the grasping phase, the difference between the velocities of the target and the end-effector should be zero. In practise, this never happens, thus small impacts and vibrations have to be taken into account. Research

on this topic is still going on, some focusing on bio-inspired isolation systems ([2018], [2020]).

In deploying a SMS for target capture, a manipulator trajectory is needed to achieve the goal. Several secondary optimization goals, such as obstacle and singularities avoidance, fuel consumption, and base disturbance minimization [1999] can be sought, too. E.g. [2009] focuses on the optimal time needed to reach the target [2]. For this thesis' purpose, only impact and captured phase along with post-captured stabilization phase have been considered.

1.3 State-of-art

The Low Earth Orbit (LEO), a whole band of orbits between 300 and 700 km altitude, is the main of today's manned space missions. This class belongs to the US Skylab, the Space Shuttle, the Russian Space Station Mir, and the International Space Station (ISS). The orbits are just high enough to be practically free of destabilising dynamic disturbances but as low as possible to minimise launch costs. Apart from human physiology interest (to investigate the effects of weightlessness on astronauts), the main application field has always been a microgravity research. The term "microgravity" refers to the typical level of "weightlessness" on such manned missions: due to various disturbances (remaining atmospheric drag, moving machinery, but most of all man motion), some $10e-6$ g acceleration will continue to act on every mass [1].

Nowadays, as already mentioned, on the ISS there are three main SMSs:

1. SSRMS: Space Station Remote Manipulator System
2. SPDM: Special Purpose Dexterous Manipulator
3. JEMRMS: Japanese Experiment Module Remote Manipulator System
4. ERA: European Robotic Arm

The ISS assembly, maintenance and scientific research tasks depend on the three robotic manipulators carrying out their intended operations. Each manipulator has a specific purpose and each has been designed to successfully complete its assigned mission.

All three ISS robotic manipulators are designed to resemble the human arm. They are mechanically and electrically connected to the ISS at the shoulder, which is comprised of a set of revolute mechanical joints. A boom segment and then an elbow joint follow the shoulder. After the elbow there is another boom segment followed by a set of wrist joints and an end effector through which the operational grappling is carried out.[2]

1.3.1 SSRMS

The SSRMS, or Canadarm2, is primarily used for ISS assembly. This involves acquiring an assembly payload, either directly from the Shuttle Payload Bay or handed off from the Shuttle Remote Manipulator System (SRMS). The SSRMS is also used to manoeuvre astronauts and whatever they are holding from point to point as they complete their tasks during the EVA.

The 16.9m SSRMS has 7 offset joints and is symmetric about the elbow with identical 3DOF shoulder and wrist clusters (Figure 1.1). The shoulder/wrist clusters contain three mechanical joints with perpendicular rotation axes: pitch, yaw and roll. One joint cluster acts as the shoulder while the other acts as the wrist of the manipulator. All of the SSRMS joints are functionally identical with the same performance capabilities and angular range of ± 270 degrees.

The three pitch and two roll joints are identical and interchangeable. The two yaw joints are identical and interchangeable (the difference between the yaw and pitch/roll joints being that the yaw joints have a 5.25 inch extension non one end). Each of the joints is an ORU and each contains two identical joint electronics units and motor modules.

Attached to the roll joint on each cluster is a Latching End Effector (LEE) which provides a mechanical and electrical connection to a payload, SPDM or the ISS when grappling an SSRMS grapple fixture. In this way, it is possible for the robotic arm to move around the station's exterior. Each LEE also includes a Force Moment Sensor (FMS) and two identical LEE Electronics Units (LEUs). This symmetric configuration allows either end of the SSRMS to be attached to a Power and Data Grapple Fixture (PDGF) and to function as the base of the arm.

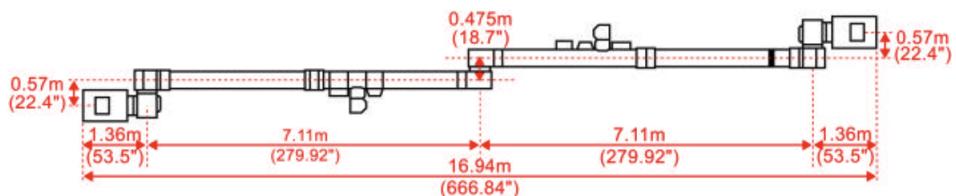


Figure 1.1: SSRMS Dimensions [4].

1.3.2 SPDM

The SPDM is a highly dexterous dual-armed robot, with two shoulders that provide support for the arms.

Each arm has seven joints, terminating with an Orbit Replacement Unit/Tool Changeout Mechanism (OTCM), the mechanism through which the SPDM arm interfaces with payloads and tools. The SPDM is also equipped with lights, video equipment, a tool platform and four tool holders.

The SPDM's primary role is to carry out delicate maintenance and servicing tasks on the International Space Station. Tasks include installing and removing small payloads such as batteries, power supplies and computers; operate robotic tools such as specialized wrenches and socket extensions; provide power and data connectivity to payloads; manipulate, install, remove and inspect scientific payloads.

It works near the Canadarm2 and can be grappled by it to be moved around.

1.3.3 JEMRMS

The JEMRMS system is used primarily for experiment payload handling. It consists of three major subsystems, the Console, Main Arm (MA), and Small Fine Arm (SFA).

The Main Arm (MA) is a 10m, six-degree-of-freedom robot arm. It consists of three booms, six joints, an end effector on the tip, and two vision systems (one on the shoulder, the other on the tip).

The Small Fine Arm (SFA) is also a six-degree-of-freedom robot arm. It can perform more dexterous tasks than the MA. During operation, the SFA is grappled by the MA and receives power and data from the MA. The 2m SFA consists of two booms, six joints, and an end effector on the tip called the “tool,” and a camera on the tool. The tool has a force and torque sensor inside it for compliance operation [8].

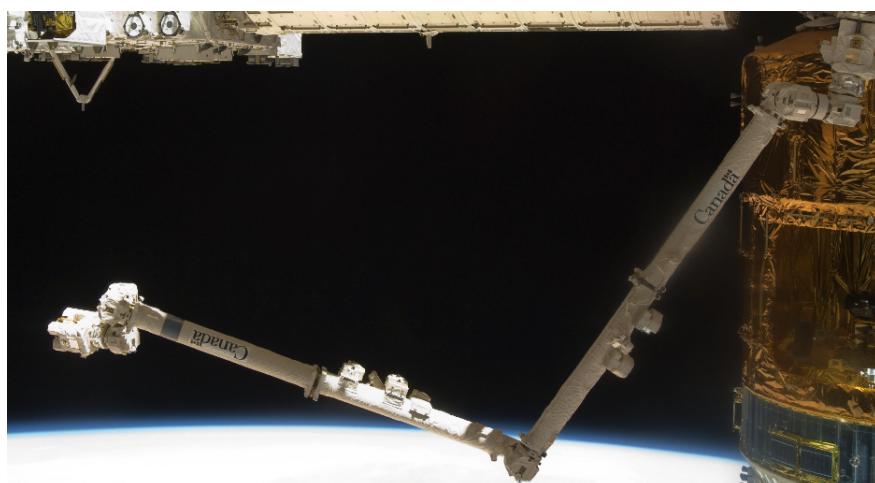


Figure 1.2: The SSRMS on orbit [7].



Figure 1.3: The SPDM on orbit.

The JEM RMS end effector was designed based on the Shuttle RMS (SRMS)

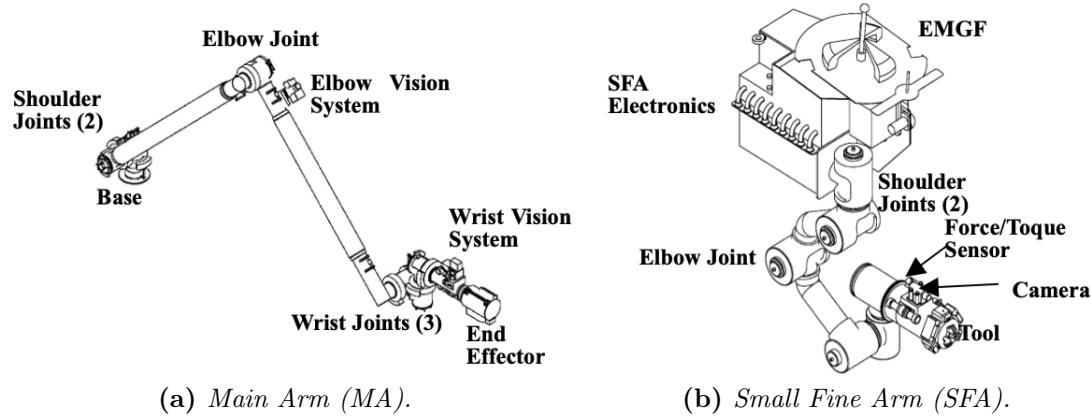


Figure 1.4: The JEMRMS components. [8]

end effector. The grapple fixtures compatible with the JEMRMS MA can also be grappled by the SSRMS [4].

1.3.4 ERA

The ERA is used on the Russian segment (RS) of the ISS. Specific tasks for the ERA include: installing, removing or replacing experiment payloads and large Station elements; transferring small payloads in and out of the Station through the Russian airlock; transporting crew members from one external working site to another; freeing cosmonauts to do other work during spacewalks; using its cameras to inspect the outside of the Space Station. The ERA manipulator arm spans 11.3m and has 7 inline joints in a symmetric configuration with respect to the elbow joint (Figure 2). It has a shoulder with 3 DOFs, an elbow with 1 DOF and a wrist with 3 DOFs. Both the shoulder and wrist contain three functionally identical joints with mutually perpendicular rotation axes.

The ERA has two end effectors (EE) which can act as a base, enabling either end of the ERA to be used as the shoulder. Thus the wrist and shoulder are operationally interchangeable, just as in the SSRMS design. The identical EE's are able to grapple and ungrapple an ERA grapple fixture, transferring power or data to an external device/load. The EE also has an Integrated Service Tool (IST) which is a built-in motorized "screwdriver" that can provide mechanical torque actuation.

The ERA is equipped with four Camera and Lighting Units (CLU) one on each EE and one on either side of the elbow. The CLU will be used to provide video images of operational tasks and, if necessary, to illuminate the scene and to act as an optical sensor to enable proximity control of a target during approach of a base point or grapple fixture in closed loop. The proximity control function enables ERA to approach grapple fixtures automatically and smoothly [4].

1.3.5 RMSs Comparison

To conclude, The 7DOF design of the European and Canadian arms provide an extra degree of freedom that will enable both manipulators to access tight workspaces and avoid ISS structure during operations. This is particularly important when working in areas that have exposed station structure. The 7th DOF also allows for Arm Pitch Plane Change manoeuvres to avoid structure without changing the end effector position.

The maximum handling payloads the manipulators can capacity also dictates the type of grasp. The SSRMS is capable of handling payloads that have an equivalent mass to the Orbiter. The ERA is designed for slightly smaller payloads such as solar panels.

The ERA and the JEMRMS provide much finer position accuracy than the SSRMS: smaller payloads require more precise alignment. Also, the force moment sensor is always used for ERA and JEM operations and is optional for SSRMS operations.

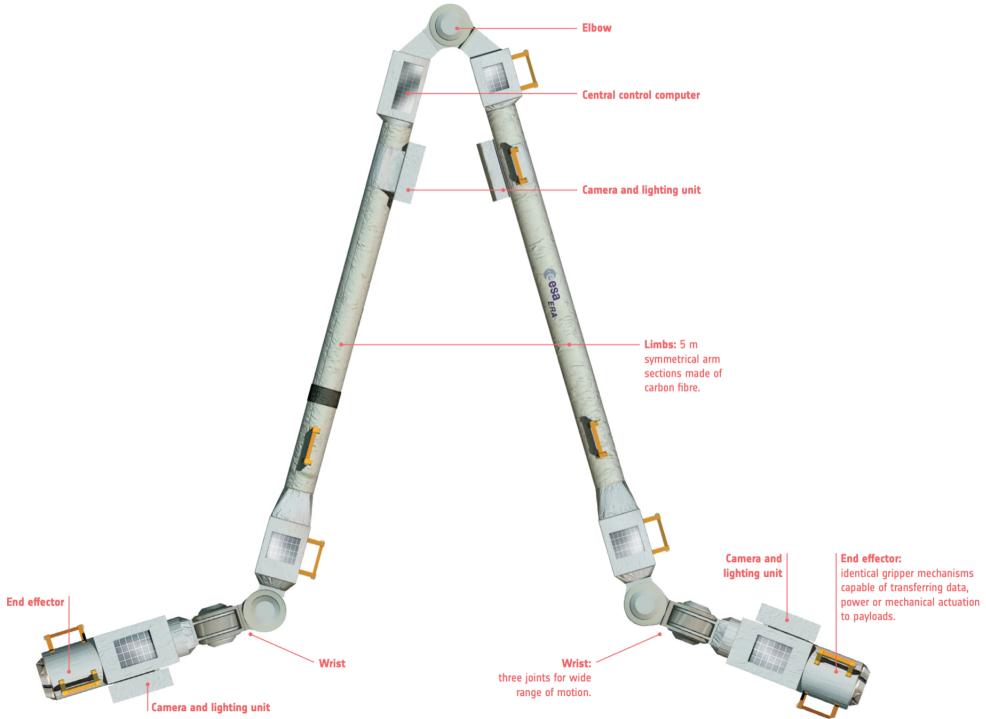


Figure 1.5: ERA’s components.

This FMS capability facilitates “peg in the hole” type of operations. In the Table 1.1 the specifications of the SSRMS, ERA and JEMRMS are reported [4].

Table 1.1: SSRMS, ERA and JEMRMS specifications [4].

	SSRMS	ERA	JEMRMS (MA)
Span	14.22 m	11.3 m	9.91 m
Boom Span	7.11 m	7.77 m	3.93 m
DOFs	7	7	6
Joints	Offset	Inline	Offset
Base	Relocatable	Relocatable	Fixed
Positioning Accuracy	65 mm, 0.71°	40 mm, 1°	50 mm, 1.8°
Mass	1336 kg	630 kg	757 kg
Max Handling Capacity	116 Mg	8 Mg	7 Mg
Power Consumption	1360 W (average)	>800 W	2.3 kW

Chapter 2

Kinematics

What is a robot, then? In the previous chapter we have seen an overview of the actual reality of the space robotics field, but a proper definition of what a robot is has not been given. A robot is a re-programmable machine for manipulation, positioning, orientation, machining of bodies able to interact with the environment, with a certain level of autonomy.

A robot is not just an automated machine, it is something more: takes decisions autonomously, can do different things at different times, is flexible.

According to [11] there are two main families of robots:

- serial robots: a set of bodies rigidly connected by joints in an open kinematic chain.
- parallel robots: a set of bodies connected in a closed kinematic chain. They can withstand/produce a larger force.

A kinematic chain is therefore the concatenation of single degree of freedom (i.e. the minimum amount of independent coordinates that define the configuration of the robot) links. The joints can be revolute, prismatic or spherical, depending on the movement allowed by them (plane rotational, translational or 3D rotatational respectively).

Grubler equation for planar mechanisms gives the overall DoFs of a multibody system:

$$n = 3(m - 1) - 2c_1 - c_2 \quad (2.1)$$

where m is the number of rigid bodies attached to the base, c_1 and c_2 are class 1 and class 2 joints respectively, depending on how many DoFs do they allow (one or two). Revolute joints allows one DoF, i.e. rotation around their axes.

Whenever $n < n_J$, where n_J is the number of joints, a redundancy occurs, meaning that the system has more joints than the dimension of the task space [11].

A very important definition for serial robots is the end defector (EE), i.e the point

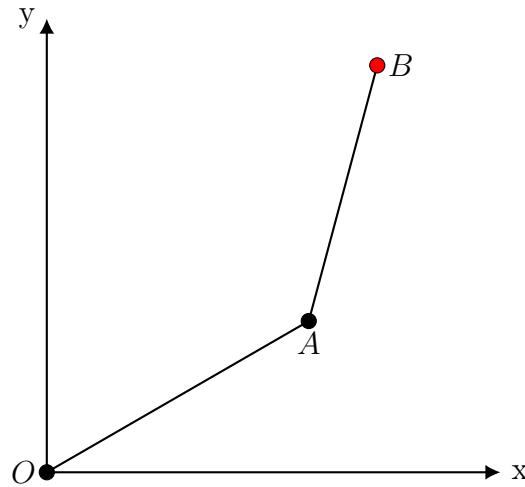


Figure 2.1: 2D RR (revolute-revolute) manipulator.

of the chain dedicated to the programmed task, as shown in Figure 2.1 in red. Studying the kinematics of a mechanical system (such as a robot) refers to the calculation of the position and orientation of its end-effector frame from its joint coordinates [11].

Usually we speak of:

- Forward or Direct Kinematics when, given the joints' angles we want to find the EE position, used to define the relation between joints and EE and determine the system jacobian (see Section 2.1.4);
- Inverse Kinematics when we want to find the joints' angles that realize a desired position at the EE, used to define the joint's time evolution to move the EE to follow a desired trajectory or to reach a desired final position (see Chapter 4).

We will now analyze the forward kinematics of a manipulator mounted on top of a mobile base, to form a vehicle-manipulator system (VMS) that can be controlled as a whole.

Starting from a simple model of the VMS, it will be possible to analyze its behaviour before and after the impact with an external object (i.e. meteorite or satellite).

2.1 Fundamental of Kinematics

In this thesis, the kinematics chain have been evaluated by means of the "Homogeneous Matrix Approach" shown in [9,10]. For such approach, a briefly summary of notation is here presented.

2.1.1 Roto-Translation Matrix

Given two vectors in an xy plane, only rotated without distortion, as in Figure 2.2, the relationship between them can be found in the following way:

$$v_2 = \begin{bmatrix} v_{x_2} \\ v_{y_2} \end{bmatrix} = \begin{bmatrix} v_{x_1} \cos \theta - v_{y_1} \sin \theta \\ v_{x_1} \sin \theta + v_{y_1} \cos \theta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_{x_1} \\ v_{y_1} \end{bmatrix} = R(\theta)v_1 \quad (2.2)$$

where R_θ is the rotation matrix.

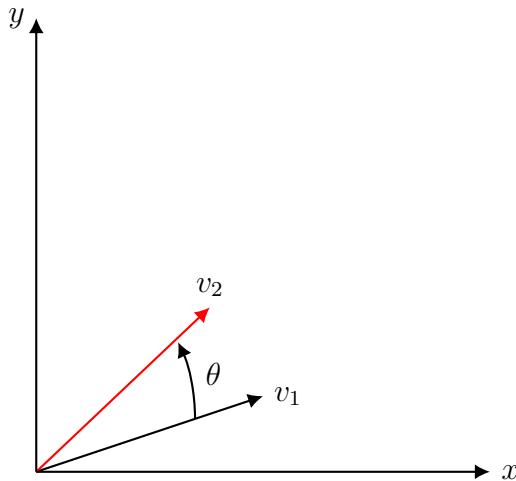


Figure 2.2: Vecors v_2 is rotate by θ withh respect to v_1 .

We can use the same matrix to describe another transformation, i.e. to find the coordinated of a point initially defined with respect to a mobile frame, with respect to another frame, such as a fixed one (Figure 2.3):

$${}^f P = {}^m R(\theta) {}^m P \quad (2.3)$$

The notation used here is the following: the left superscript indicates the reference frame in which the point or the vector is referred to: ${}^m P$ refers to the point P with coordinate written with respect to the mobile frame. The matrix ${}^m R(\theta)$ "translates" a point described in the mobile frame to the fixed frame.

An important property of the rotation matrix is that it is orthogonal, hence its inverse is equal to its transpose.

We can concatenate the rotational matrices such that we can describe a point in a

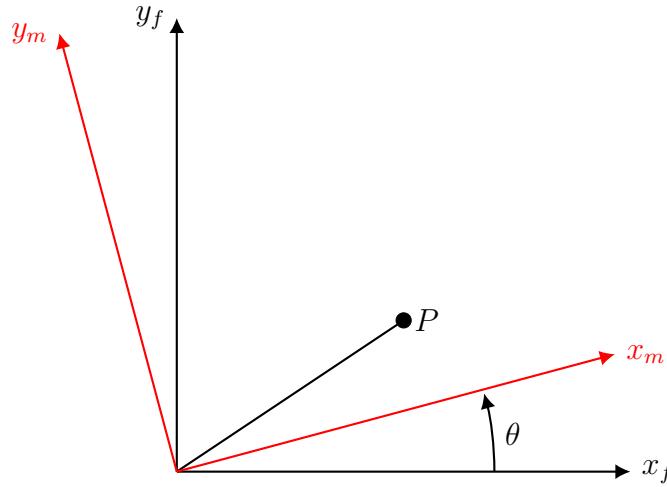


Figure 2.3: The same point can be described with respect to both a fixed and a rotated mobile frame (in red).

given reference frame to another one if we know all the relative rotational matrices of the frames between the two we are interested in:

$${}^0P = {}^0R(\theta) {}^1R(\theta) {}^2P \quad (2.4)$$

This concept can be expanded easily in the 3D case, noticing that the columns of $R(\theta)$ in (2.2) are the unit vectors of the moving (new) frame written in the fixed (old) frame. In this way, rotation matrices along the x, y and z axis are found [6]:

$$\begin{aligned} R_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} & R_y(\psi) &= \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \\ R_z(\theta) &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.5)$$

When a translation occurs, as shown in Figure 2.4, the transformation becomes:

$${}^f\{O_fP\} = {}^f\{O_fO_m\} + {}^f\{O_mP\} = {}^f\{O_fO_m\} + {}_fR {}^m\{O_mP\} \quad (2.6)$$

which can be written as follows:

$$\begin{bmatrix} {}^fO_fP_x \\ {}^fO_fP_y \\ {}^fO_fP_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^f\{O_fO_mx\} \\ {}^f\{O_fO_my\} \\ {}^f\{O_fO_mz\} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} {}^m\{O_mP_x\} \\ {}^m\{O_mP_y\} \\ {}^m\{O_mP_z\} \\ 1 \end{bmatrix} = {}_fM \begin{bmatrix} {}^m\{O_mP_x\} \\ {}^m\{O_mP_y\} \\ {}^m\{O_mP_z\} \\ 1 \end{bmatrix} \quad (2.7)$$

where $\mathbf{0} \in \mathbb{R}^{3 \times 1}$ and ${}^m_f M \in \mathbb{R}^{4 \times 4}$ is called the *Roto-Translational Matrix* or *Homogeneous Transformation Matrix*, for which the same properties seen in equation (2.4) hold. A fourth component has been appended at the end of the vectors, which is 1, since the Homogeneous Matrix would have had three rows but four columns otherwise. In this way it is squared and invertible, and ${}^0_1 M^{-1} = {}^1_0 M$.

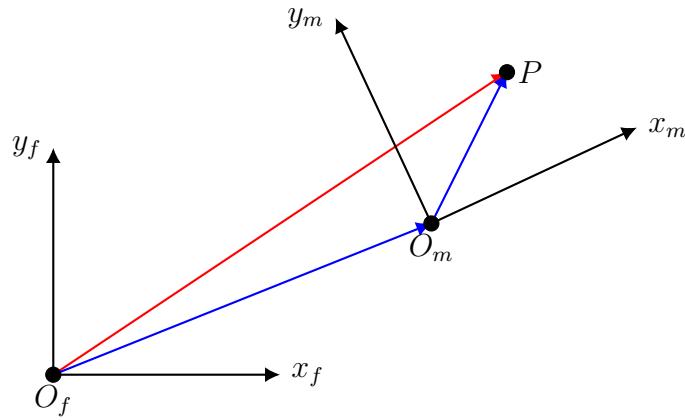


Figure 2.4: Definition of a point in a rotated and translated frame with respect to the fixed one. The position of P with respect to the fixed frame (red arrow) can be seen as the sum of the position of the moving frame O_m with respect of O_f and the relative position of P in the moving frame, both in blue.

2.1.2 Velocity Matrix

Given the same point in Figure 2.4 it is possible to find the velocity of the point with respect to the fixed frame. By derivation of equation (2.7) one can get:

$${}^f\{\dot{O_f}P\} = {}_m^f\dot{M}{}^m\{O_mP\} + {}_m^fM{}^m\{\dot{O_m}P\} \quad (2.8)$$

and since ${}^m\{\dot{O_m}P\} = 0$, assuming the point fixed in the mobile frame, and ${}^m\{O_mP\} = {}_f^mM{}^f\{O_fP\}$:

$${}^f\{\dot{O_f}P\} = {}_m^f\dot{M}{}^m{}_f^mM{}^f\{O_fP\} = {}^fW{}^f\{O_fP\} \quad (2.9)$$

with fW the *Velocity Matrix*.

The Velocity Matrix can be generalized as follows [9]:

$$W = \begin{bmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.10)$$

where ω indicates the angular velocity of the body and v is the velocity of the point, considered belonging to the body that in a considered instant is passing through the origin of the reference frame.

The composition of the Velocity Matrix follows the Rival's Theorem (see Figure 2.5):

$${}^nW_{02} = {}^nW_{01} + {}^nW_{12} \quad (2.11)$$

which is true only when all the matrices are projected in the same frame n .

The notation used in equation (4.9) has to be read in the following way: ${}^nW_{02}$ is the Velocity Matrix of frame two with respect to frame zero, projected in frame n . It is also possible to change the frame in which the matrix is projected:

$${}^mW = {}_n^mM{}^nW{}^n{}_m^mM \quad (2.12)$$

Some attention must be given also to the matrix representing a "boolean" interpretation of the rotation: the L matrix. For the aim of this thesis, and as it is introduced in [10], this matrix can be easily obtained from its Velocity Matrix dividing it by the module of its angular (or linear, when there is translation) velocity:

$$L = \frac{W}{|\omega|} \quad (2.13)$$

where $|\omega| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$.

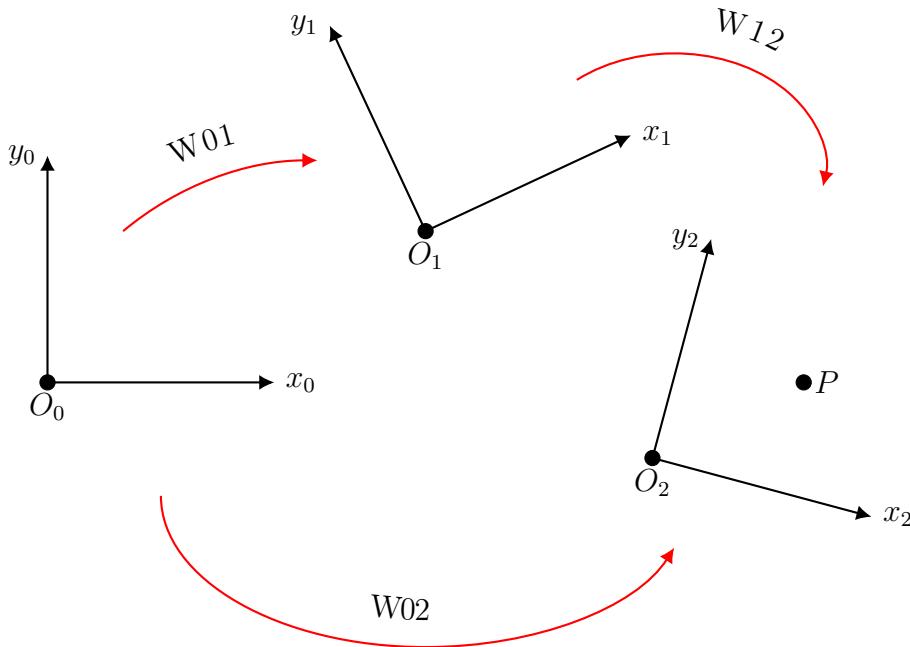


Figure 2.5: Rival's Theorem for velocities composition. The three transformation W_{01} , W_{12} and W_{02} are also called *Drag Motion*, *Relative Motion* and *Absolute Motion* respectively.

Given the robotic focus of this work, these calculations will be quite straightforward once defined the axis of rotation of a link. For a revolute joint, for example, the Velocity Matrix with respect of the joint frame will be:

$$W = \begin{bmatrix} 0 & -\dot{\alpha} & 0 & 0 \\ \dot{\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.14)$$

with $\dot{\alpha} = \omega_z$, and the "logic" of the rotation is thus given by L :

$$L = \frac{W}{\dot{\alpha}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.15)$$

2.1.3 Danavit-Hartenberg Method

To define the reference frames fixed to the links of a serial manipulator, one can follow the Danavit-Hartenberg algorithm, which guarantees a coherent and functional method through all the calculations. It has been thought for robots with revolute joints.

It follows this ordered procedures:

1. z_i axis: axis of the revolute joint which connects the link to the following.
2. x_i axis: line of minimum distance between z_{i-1} and z_i , oriented from z_{i-1} to z_i .
3. y_i axis: obtained by the vectorial product of the other two axes.

In Figure 2.6, a DH convention for a planar serial manipulator has been used for configuration shown in Figure 2.1.

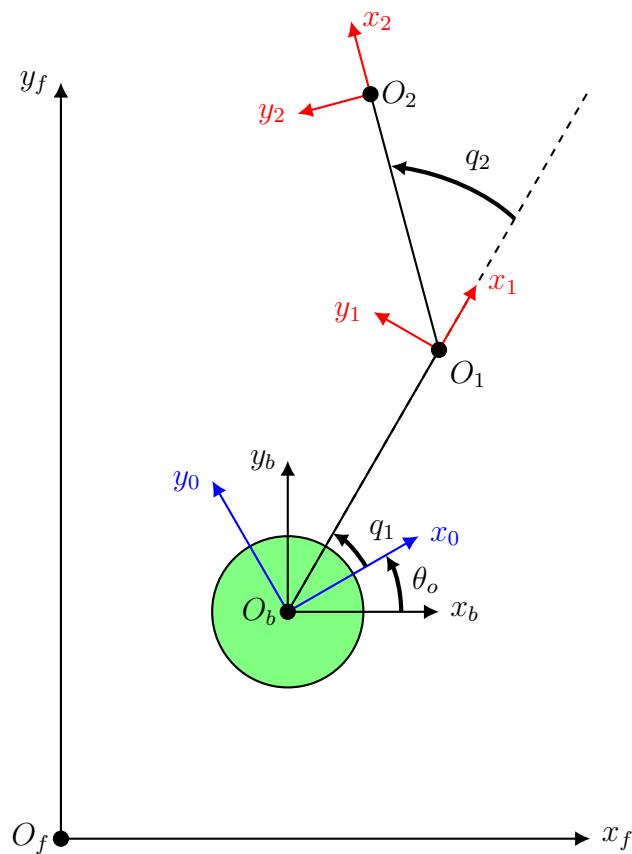


Figure 2.6: The Danavit-Hartenberg rule has been used to place the local joints' frame (in red) in a planar RR manipulator, with arms length l_1 and l_2 , mounted on top of a disk-shaped base (in green). The rotated frame of the base of the angle θ_o has been plotted in blue. Following the hand rule, the z axis is orthogonal to the plane.

2.1.4 Jacobian Matrix

Another important matrix in the analysis if a manipulator is the so called *Jacobian Matrix*.

To study the direct kinematics of a manipulator, as already said, means to find the relationship between the end-effector position and orientation and its joints. When writing the position of the EE (denoted by S), we obtain a function of the robot's joints:

$$S = f(q_1, q_2, \dots, q_n) \quad (2.16)$$

where q_i are the manipulators DoFs.

By derivation of (2.23) one can obtain:

$$\dot{S} = J_a \dot{Q} \quad (2.17)$$

where J_a is the *Analytical Jacobian Matrix* and $Q = \{q_1, q_2, \dots, q_n\}$.

Its dimension are very important, since kinematic singularities, which are configurations in which the end-effector frame loses the ability to move or rotate in one or more directions (i.e. it loses one or more DoFs), correspond to those configurations at which the Jacobian matrix fails to have maximal rank [11], where the rank is defined as the number of independent rows or columns of the matrix.

Singularities happen if the open-chain robot is kinematically redundant, meaning, as already mentioned, that it has more joints than the dimension of the task space; then the inverse of the jacobian doesn't exist anymore, since the inversion requires a squared full rank matrix, and the inverse kinematics is not possible ($\dot{Q} = J_a^{-1} \dot{S}$ and J_a^{-1} does not exist in singular configurations). When this occurs, the pseudoinverse or *Moore-Penrose Inverse* matrix, denoted by J^+ , must be used:

$$\begin{cases} J^+ = (J^T J)^{-1} J^T & \text{when } J \text{ has linearly independent columns} \\ J^+ = J^T (J J^T)^{-1} & \text{when } J \text{ has linearly independent rows} \end{cases} \quad (2.18)$$

In robotics, the most common case is the one with a Jacobian with linearly independent columns, since it's common to have more links than dimensions in the task space.

2.2 Planar Vehicle-Manipulator System

It will be now presented a simple case of Vehicle-Manipulator System, consisting in a base approximated with a disk and a planar serial manipulator whose revolute joints move around their z axis, orthogonal to the plane (as the one shown in Figure 2.6).

Since the aim of this work is to study the dynamics of the system after the catching of a tumbling object, its kinematics will be analyzed too, by considering it as a disk. The calculations will be shown as a reference for future more complex model, for which the matrices and joint positions will be omitted.

Firstly, let's define the DoFs of the VMS, which will be:

$$p = \{x_b, y_b, \theta_b, q_1, q_2\} \quad (2.19)$$

It is already clear that, being in the 2D case, the number of DoFs is greater than the dimension of the task space (i.e. $5 > 2$). In fact, from (2.1), given $m = 3$ (base and two arms) and $c_1 = 2$ (two revolute joints), we have $n = 2$, while $p \in \mathbb{R}^5$.

This means that a redundancy occurs and the pseudoinverse jacobian must be used instead of the inverse one, when needed.

The generalized coordinate of the tumbling object are the following:

$$\psi = \{x_O, y_O, \theta_O\} \quad (2.20)$$

Since the links rotate only around the local z-axis, given equations (2.5) and (2.7), the roto-translation matrices of the manipulator chain have the following form:

$${}^a_f M = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & p_1 \\ \sin \alpha & \cos \alpha & 0 & p_2 \\ 0 & 0 & 1 & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

where p_1, p_2, p_3 are the coordinates of the following moving frame in the chain projected in the actual frame.

Following the notation of Figure 2.6, the matrices are then given:

$$\begin{aligned} {}^f_b M &= \begin{bmatrix} 0 & 0 & 0 & x_b \\ 0 & 0 & 0 & y_b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^b_0 M &= \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 & 0 & 0 \\ \sin \theta_0 & \cos \theta_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0_1 M &= \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & l_1 \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & l_1 \sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2 M &= \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.22)$$

with l_1 and l_2 the lengths of the two arms.

The origin of the frames written in the fixed one can be found by multiplying the Homogeneous Matrix by $O_3 = \{0, 0, 0, 1\}$. Thus, the end-effector position is:

$$p_{EE} = {}^0_2 M O_3 = \begin{bmatrix} l_1 \cos(\theta_0 + q_1) + l_2 \cos(\theta_0 + q_1 + q_2) + x_b \\ l_1 \sin(\theta_0 + q_1) + l_2 \sin(\theta_0 + q_1 + q_2) + y_b \\ 0 \\ 1 \end{bmatrix} \quad (2.23)$$

with ${}^0_3 M = {}^f_b M {}^b_0 M {}^0_1 M {}^1_2 M$ for the chain rule (2.4).

The Jacobian Matrix is then obtained by derivation of p_{EE} with respect to the VMS DoFs p such that $\dot{p}_{EE} = J \dot{p}$:

$$J = \begin{bmatrix} 1 & 0 & -l_1 \sin(\theta_0 + q_1) - l_2 \sin(\theta_0 + q_1 + q_2) & -l_1 \sin(\theta_0 + q_1) - l_2 \sin(\theta_0 + q_1 + q_2) & -l_2 \sin(\theta_0 + q_1 + q_2) \\ 0 & 1 & l_1 \cos(\theta_0 + q_1) + l_2 \cos(\theta_0 + q_1 + q_2) & l_1 \cos(\theta_0 + q_1) + l_2 \cos(\theta_0 + q_1 + q_2) & l_2 \cos(\theta_0 + q_1 + q_2) \end{bmatrix} \quad (2.24)$$

As already forecasted, the Jacobian Matrix is rectangular, i.e. $J \in \mathbb{R}^{2 \times 5}$, thus not invertible.

Following equation (2.9), the Velocity Matrices can be found as ${}^fW = {}_m\dot{M} {}_f^mM$:

$$\begin{aligned} {}^fW_{fb} &= {}_b\dot{M} {}_f^bM = \begin{bmatrix} 0 & 0 & 0 & \dot{x}_b \\ 0 & 0 & 0 & \dot{y}_b \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & {}^bW_{b0} &= {}_0\dot{M} {}_b^0M = \begin{bmatrix} 0 & -\theta_0 & 0 & 0 \\ \theta_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ {}^1W_{01} &= \begin{bmatrix} 0 & -q_1 & 0 & 0 \\ q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & {}^2W_{12} &= \begin{bmatrix} 0 & -q_2 & 0 & 0 \\ q_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (2.25)$$

where the notation is the same of equation (4.9).

Since it is of our interest (see Chapter 3) to write all the velocity matrices projected in the fixed frame, following equation (2.12) it is possible to write:

$${}^fW_{b0} = {}_b^fM {}^bW_{b0} {}_f^bM = \begin{bmatrix} 0 & -\theta_0 & 0 & y_b\dot{\theta}_0 \\ \theta_0 & 0 & 0 & -x_b\dot{\theta}_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.26)$$

Finally, now that both ${}^fW_{fb}$ and ${}^fW_{b0}$ are projected in the same frame, following equation (2.4):

$${}^fW_{f0} = {}^fW_{fb} + {}^fW_{b0} = \begin{bmatrix} 0 & -\theta_0 & 0 & \dot{x}_b + y_b\dot{\theta}_0 \\ \theta_0 & 0 & 0 & \dot{y}_b - x_b\dot{\theta}_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.27)$$

and the same holds for ${}^fW_{f1}$ and ${}^fW_{f2}$, whose 3×3 submatrices, related to the angular velocity, are:

$${}^fW_{f1_{1:3,1:3}} = \begin{bmatrix} 0 & -\theta_0 - q_1 & 0 \\ \theta_0 + q_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad {}^fW_{f1_{1:3,1:3}} = \begin{bmatrix} 0 & -\theta_0 - q_1 - q_2 & 0 \\ \theta_0 + q_1 + q_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.28)$$

We could have calculated ${}^fW_{f0}$ using equation (2.9) as well (${}^fW_{f0} = {}_0\dot{M} {}_f^0M$). However, it is useful to have also the relative velocity matrices to calculate the

corresponding L matrices (see equation (2.15)):

$$\begin{aligned} {}^f L_{fb_x} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & {}^f L_{fb_y} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ {}^b L_{b0} = {}^0 L_{01} = {}^1 L_{12} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (2.29)$$

where ${}^f L_{fb_x}$ and ${}^f L_{fb_y}$ have been found such that ${}^f L_{fb} = {}^f L_{fb_x} + {}^f L_{fb_y}$.

By projecting in the fixed frame:

$${}^f L_{b0} = \begin{bmatrix} 0 & -1 & 0 & y_b \\ 1 & 0 & 0 & -x_b \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.30)$$

which is ${}^f W_{b0}/\dot{\theta}_0$ from equation (2.26).

2.2.1 Object's Kinematics

For the planar case, the object to be captured (might it be a satellite or a micro-meteorite) is approximated with a disk, in the same way we did for the base of the VMS.

The goal of the object's kinematics is to find the relationship between the velocity of its centre and the velocity of the contact point (c_p), as suggested in [12].

Hence, the Homogeneous Matrices related to the tumbling object are the following:

$$\begin{aligned} {}_{O_0}^f M &= \begin{bmatrix} 1 & 0 & 0 & x_O \\ 0 & 1 & 0 & y_O \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}_{O_0}^{O_1} M &= \begin{bmatrix} \cos \theta_O & -\sin \theta_O & 0 & 0 \\ \sin \theta_O & \cos \theta_O & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}_{O_1}^f M &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & r \cos \gamma \\ \sin \gamma & \cos \gamma & 0 & r \sin \gamma \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.31)$$

with the coordinates of the object as in (2.20), γ the angle between the the object local frame and the position of the contact point (see Figure 2.7) and r the radius of the disk.

As for the previous case, the position of the contact point with respect to the fixed

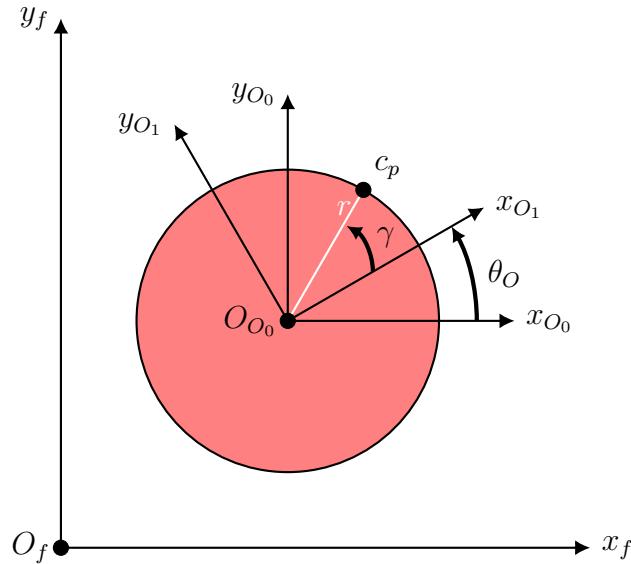


Figure 2.7: Tumbling object disk approximation.

frame is found with the chain rule:

$${}^f c_p = {}_{O_0}^f M {}_{O_1}^{O_0} M {}_{O_2}^{O_1} M {}^{O_1} c_p = \begin{bmatrix} x_O + r \cos(\theta_O + \gamma) \\ y_O + r \sin(\theta_O + \gamma) \end{bmatrix} \quad (2.32)$$

By deriving the position of the contact point with respect to the object coordinates ψ , we obtain the object Jacobian Matrix:

$$J_O = \begin{bmatrix} 1 & 0 & -r \sin(\theta_O + \gamma) \\ 0 & 1 & r \cos(\theta_O + \gamma) \end{bmatrix} \quad (2.33)$$

which is still not squared ($J_O \in \mathbb{R}^{2 \times 3}$), thus not invertible.

Regarding the velocity matrix, we are not interested in ${}^f W_{fO_2}$, since we can find the velocity of the contact point by just derivation with respect of time. Instead, it will be useful in the next chapter to have ${}^f W_{O_0 O_1}$, which is the velocity of frame one with respect to frame zero, projected in the fixed frame:

$${}^f W_{O_0 O_1} = \begin{bmatrix} 0 & -\theta_O & 0 & y_O \dot{\theta}_O \\ \theta_O & 0 & 0 & -x_O \dot{\theta}_O \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.34)$$

which is very similar to (2.26). Hence, in the same way we did in the previous section, we can retrieve the following matrices:

$$\begin{aligned} {}^f L_{fO_x} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & {}^f L_{fO_y} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ {}^f L_{O_0 O_1} &= \begin{bmatrix} 0 & -1 & 0 & y_O \\ 1 & 0 & 0 & -x_O \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (2.35)$$

Chapter 3

Dynamics

The dynamics analysis of a mechanical system consists in studying how the system evolves according to the present actions, given by motors and/or external world. If the kinematics analysis is used to describe the motion, the dynamics one studies its causes. For a robotic system, the forces involved are usually the torques applied at the joints and, in this case, also the forces applied to the base.

The analysis can be:

- Direct: from the torques applied by the motors, the motion is calculated;
- Inverse: given the motion we calculate the forces and torques that produce that motion.

The former problem requires differential calculation, while the latter algebraic.

In this thesis, both problems will be studied: given the impact of the tumbling object with the VMS, we want to analyse the motion of the system after the impact; however, we also want to find a control algorithm to be applied at the joints to stabilize the system after the impact.

Before writing the dynamic equations, we want to find a relationship between the forces at the end-effector and the forces at the joints, that is, given a *wrench* \mathbf{w} (force + moment) applied at the end-effector, compute the joints' torques τ_w generated by it.

The power at the joints is given by:

$$P_\tau = \tau_w^T \dot{q} \quad (3.1)$$

where $q \in \mathbb{R}^n$ is the vector of joint variables and $\tau_w \in \mathbb{R}^n$ is the vector of forces and torques applied to the joints because of w .

The power at the end-effector is:

$$P_e = w^T v \quad (3.2)$$

with v the velocity of the wrench. Since $P_\tau = P_e$ and given the relationship from (2.17), we can write:

$$\tau_w^T \dot{q} = w^T v = w^T J \dot{q} \quad \forall \dot{q} \quad (3.3)$$

with J the jacobian matrix. Hence:

$$\tau_w^T = w^T J \quad \Rightarrow \quad \tau_w = J^T w \quad (3.4)$$

The dynamic equations, also referred to as the equations of motion, are a set of second-order differential equations of the form:

$$M(q)\ddot{q} + C(q, \dot{q}) = u + J^T(q)w \quad (3.5)$$

where $u \in \mathbb{R}^n$ is the vector of joint forces and torques, $M(q) \in \mathbb{R}^{n \times n}$ is a symmetric positive-definite *Mass Matrix*, and $C(q, \dot{q}) \in \mathbb{R}^n$ are forces that lumped together centripetal, Coriolis, gravity and friction terms that depend solely on q and \dot{q} [11]. By solving these equation, given the nature of the problem (direct or inverse), we can find the motion or the torques respectively.

There are different methods to find the equations of motion, such as Newton-Euler method and the lagrangian one. In this thesis the latter will be used, both in a "classic" scalar approach and in the Homogeneous Matrix approach shown in [10].

3.1 Classic Approach

The lagrangian approach is an energy-based one, which allows to find the dynamic equations in a symbolic closed form. Particularly, the method involves the calculation of the kinetic and potential energy of the system.

For a point mass in \mathbb{R}^3 , the kinetic energy is expressed as:

$$T = \frac{1}{2}mv^T v \quad (3.6)$$

and for a system of point masses:

$$T = \frac{1}{2} \sum_{i=1}^N m_i v_i^T v_i \quad (3.7)$$

A rigid body that moves in space has:

- $v_c \in \mathbb{R}^3$: linear velocity (of the centre of mass);
- $\omega_c \in \mathbb{R}^3$: angular velocity (all points of the body).

Hence, the kinetic energy of a rigid body of volume V is:

$$\begin{aligned} T &= \frac{1}{2} \int_V v^T v dm = \frac{1}{2} \int_V (v_c + \omega \times P)^T (v_c + \omega \times P) dm \\ &= \frac{1}{2} \int_V v_c^T v_c dm + \frac{1}{2} \int_V v_c^T [\omega]_\times P dm + \frac{1}{2} \int_V (\omega \times P)^T (\omega \times P) dm \end{aligned} \quad (3.8)$$

where P is the position of the infinitesimal mass dm with respect to the centre of mass.

From the first term of equation (3.8) the velocity vectors can be taken out of the integral; from the second term the velocity vector and the skew matrix of the angular velocity can be taken out and the integral becomes zero, because of the centre of mass theorem (i.e. $\int_V P dm = 0$). Hence, the kinetic energy becomes:

$$\begin{aligned} T &= \frac{1}{2} v_c^T m v_c + \frac{1}{2} \int_V (-[P]_\times \omega)^T (-[P]_\times \omega) dm \\ &= \frac{1}{2} v_c^T m v_c + \frac{1}{2} \int_V -\omega^T [P]_\times [P]_\times \omega dm \\ &= \frac{1}{2} v_c^T m v_c + \frac{1}{2} \omega^T \int_V -[P]_\times [P]_\times dm \omega \\ &= \frac{1}{2} v_c^T m v_c + \frac{1}{2} \omega^T I \omega \end{aligned} \quad (3.9)$$

since $(-[P]_\times \omega)^T = \omega^T - [P]_\times^T = \omega^T [P]_\times$.

The first term of equation (3.9) refers to the translational part (i.e. the kinetic energy of the point mass at the centre of mass), while the second to the rotational

kinetic energy of the whole body, which takes into account the inertial distribution through the *Tensor of Inertia* with respect to the centre of mass: $I \in \mathbb{R}^{3 \times 3}$.

The potential energy, on the other hand, will not be taken into consideration in this Chapter, since in outer space there is no gravity and we are assuming the system to be composed only with rigid bodies: no elastic deformation is taken in account. This assumption will be removed in the next Chapter when dealing with elastic arms.

Finally, it is possible to write the lagrangian formulation:

$$\mathcal{L}(q, \dot{q}) \triangleq \sum_{i=1}^N T_i - U_i \quad (3.10)$$

where N is the number of bodies and L is the *Lagrangian Function*.

The equations of motion can now be expressed in terms of the Lagrangian as follows [11]:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = f_i \quad (3.11)$$

with f_i the non lagrangian, non conservative forces (external/dissipative) performing work on q_i . The number of equations obtained is equal to the number of DoFs of the system i .

Regarding the planar VMS seen in Section 2.2 it is possible to write the position of the centre of mass of each body using the same approach of equation (2.23): $p_{c_b} = O_b = {}^f_b M \{0, 0, 0, 1\}^T$, $p_{c_1} = {}^b_1 M \{-\frac{l_1}{2}, 0, 0, 1\}^T$ and $p_{c_2} = {}^b_2 M \{-\frac{l_2}{2}, 0, 0, 1\}^T$, since the position of the centre of mass of each arm is located in its middle point and the local frame is located at the end of the arm.

Now, v_{c_b} , v_{c_1} and v_{c_2} can be found by derivation of the position vectors and the translational part of the kinetic energy (3.9) can be computed.

In order to find the rotational part, the tensors of inertia of the disk and the two arms are needed. Table 3.1 reports the moments of inertia.

Since the rotation happens to be only around the orthogonal axis of the plane (following the Denavit-Hartenberg formulation, the z-axis), we will use only I_z :

$$\begin{aligned} T_b &= v_{c_b} m_b v_{c_b}^T + \frac{1}{2} I_{b_z} \omega_{b_z}^2 \\ T_1 &= v_{c_1} m_1 v_{c_1}^T + \frac{1}{2} I_{1_z} \omega_{1_z}^2 \\ T_2 &= v_{c_2} m_2 v_{c_2}^T + \frac{1}{2} I_{2_z} \omega_{2_z}^2 \end{aligned} \quad (3.12)$$

with ω_{b_z} , ω_{1_z} and ω_{2_z} the angular velocities extrapolated from the Velocity Matrices in equations (2.27, 2.28).

As already said, L is only the sum of the three contribution of equation (3.12) and,

Table 3.1: Moments of inertia of the planar VMS' bodies, with respect to their centre of mass (in the centre of the base and the arms), with $i = \{1, 2\}$.

Body	Tensor of Inertia
Base	$\begin{bmatrix} \frac{1}{4}m_b r^2 & 0 & 0 \\ 0 & \frac{1}{4}m_b r^2 & 0 \\ 0 & 0 & \frac{1}{2}m_b r^2 \end{bmatrix}$
Arms	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{12}m_i l_i^2 & 0 \\ 0 & 0 & \frac{1}{12}m_i l_i^2 \end{bmatrix}$

since we have five DoFs ($p \in \mathbb{R}^5$), there will be five lagrangian equations, one for each generalized coordinate.

The non lagrangian are called this way because they are not captured by the Lagrange equation: if the conservative part of the forces is captured by \mathcal{L} , the non lagrangian's are the ones which are not captured by any energy.

In this case they are the torques and forces applied to the base and the manipulator's

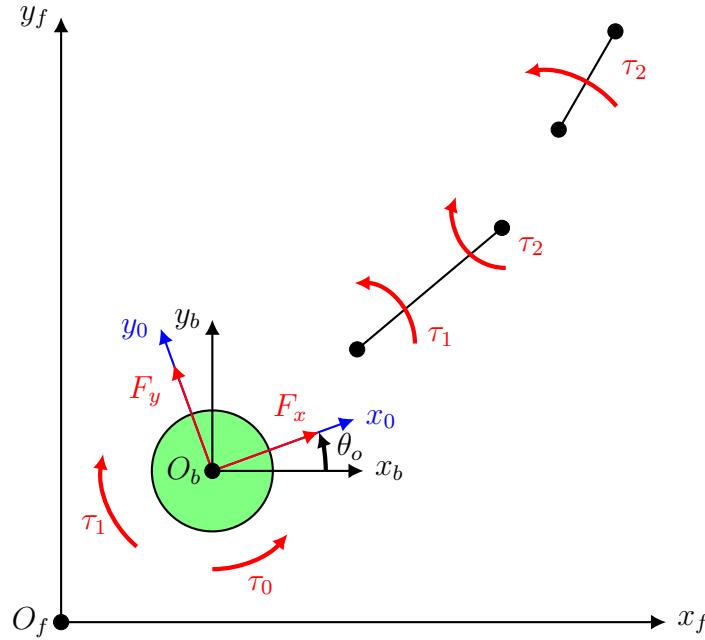


Figure 3.1: Forces and torques applied to the base (F_x, F_y, τ_0) and to the links (τ_1, τ_2). Note that the torque of the first arm affects the base and the torque of the second arm affects the first arm.

links (see Figure 3.1):

$$\begin{aligned} f_1 &= F_x \cos \theta_0 - F_y \sin \theta_0 \\ f_2 &= F_x \sin \theta_0 + F_y \cos \theta_0 \\ f_3 &= \tau_0 \\ f_4 &= \tau_1 \\ f_5 &= \tau_2 \end{aligned} \tag{3.13}$$

Since the kinetic energy is written with respect to the fixed frame, the forces are projected too in that frame through θ_0 .

Equation (3.11) can now be written as:

$$\left\{ \begin{array}{l} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}_b} - \frac{\partial \mathcal{L}}{\partial x_b} = F_x \cos \theta_0 - F_y \sin \theta_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{y}_b} - \frac{\partial \mathcal{L}}{\partial y_b} = F_x \sin \theta_0 + F_y \cos \theta_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_0} - \frac{\partial \mathcal{L}}{\partial \theta_0} = \tau_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} - \frac{\partial \mathcal{L}}{\partial q_1} = \tau_1 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_2} - \frac{\partial \mathcal{L}}{\partial q_2} = \tau_2 \end{array} \right. \tag{3.14}$$

After having solved these five equations, it is possible to gather them together in a vectorial form, such as in (3.5), with $M \in \mathbb{R}^{5 \times 5}$, $C \in \mathbb{R}^5$ and $u \in \mathbb{R}^{5 \times 5}$. The mass and Coriolis matrices can be seen in Appendix A.

3.1.1 Object's Dynamics

The same approach can be implemented for the object to be captured: since it is approximated as a disk too, its moment of inertia is equal to the base's one.

The kinetic energy is then:

$$T_O = v_O m_O v_O^T + \frac{1}{2} I_{O_z} \theta_O^2 \quad (3.15)$$

and the non lagrangian components:

$$\begin{aligned} f_{O_1} &= F_{O_x} \cos \theta_O - F_{O_y} \sin \theta_O \\ f_{O_2} &= F_{O_x} \sin \theta_O + F_{O_y} \cos \theta_O \\ f_{O_3} &= \tau_O \end{aligned} \quad (3.16)$$

supposing to have, as the VMS' base, two translational forces acting on the centre of mass and a torque around its inertial frame. However, since we are not really interested in the control of the external object, these are not necessary and we will use only the mass and Coriolis matrices.

By solving equation (3.11), we get the mass, Coriolis' and torques matrices:

$$\begin{aligned} M_O &= \begin{bmatrix} m_O & 0 & 0 \\ 0 & m_O & 0 \\ 0 & 0 & \frac{m_O r^2}{2} \end{bmatrix} & C_O &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ u_O &= \begin{bmatrix} \cos \theta_O & -\sin \theta_O & 0 \\ \sin \theta_O & \cos \theta_O & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.17)$$

3.2 Homogeneous Matrices Approach

The Homogeneous Matrices Approach introduced in [9,10] offers a quick, intuitive and scalable (for higher DoFs) method for robotic's dynamics.

It is base, as the name suggests, on the Homogeneous Matrices shown in Section 2.1 and on other characteristic matrices that will be here presented.

The goal of this approach is to write the lagrangian in a matrix form and, to do so, we need to define matrices for the inertia of the system, for the non-lagrangian actions present in the system and the velocity of the bodies.

The latter ones have already been discussed in Section 2.1.2, while the inertia and non-lagrangian forces need to be introduced.

The mass distribution of a body k , projected in the frame i , can be described through the *Pseudo Inertia Tensor* J :

$${}^i J_k = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} & mx_G \\ J_{yx} & J_{yy} & J_{yz} & my_G \\ J_{zx} & J_{zy} & J_{zz} & mz_G \\ mx_G & my_G & mz_G & m \end{bmatrix} \quad (3.18)$$

where m is the mass of the body, $\{x_G, y_G, z_G\}$ is the position of the centre of mass and the pseudo moments of inertia are defined as:

$$\begin{aligned} J_{xx} &= \int x^2 dm & J_{yy} &= \int y^2 dm & J_{zz} &= \int z^2 dm \\ J_{xy} &= \int xy dm & J_{xz} &= \int xz dm & J_{yz} &= \int yz dm \end{aligned} \quad (3.19)$$

These are different from the well known moments of inertia ($I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$) and their relation is shown in equation (3.20):

$$\begin{cases} J_{xx} = \frac{-I_{xx} + I_{yy} + I_{zz}}{2} \\ J_{yy} = \frac{-I_{yy} + I_{xx} + I_{zz}}{2} \\ J_{zz} = \frac{-I_{zz} + I_{xx} + I_{yy}}{2} \end{cases} \quad \begin{cases} J_{xy} = -I_{xy} \\ J_{yz} = -I_{yz} \\ J_{zz} = -I_{zz} \end{cases} \quad (3.20)$$

In order to evaluate the non lagrangian components of equation (3.11), [9] introduces the *Action Matrix*:

$${}^i \phi_k = \begin{bmatrix} 0 & -c_z & c_y & f_x \\ c_z & 0 & -c_x & f_y \\ -c_y & c_x & 0 & f_z \\ -f_x & -f_y & -f_z & 0 \end{bmatrix} \quad (3.21)$$

where $\{f_x, f_y, f_z\}$ is the resultant of the forces applied to a body k , $\{c_x, c_y, c_z\}$ holds

the torques calculated with respect to the origin of the frame i .

The pseudo-scalar product (see ??) between the action matrix of a body and its velocity matrix, gives the power developed by the actions included in ϕ as the object moves according to the velocities described by W :

$$P = {}^f\phi \otimes {}^fW \quad (3.22)$$

Notice that both the action and velocity matrix have been projected in the fixed frame.

A manipulator with two revolute links will have two contribution to the overall power, one for each link:

$$\begin{aligned} P &= {}^f\phi_1 \otimes {}^fW_{f1} + {}^f\phi_2 \otimes {}^fW_{f2} \\ &= {}^f\phi_1 \otimes {}^fL_{f1}\dot{\theta}_1 + {}^f\phi_2 \otimes \left({}^fW_{f1} + {}^fW_{12} \right) \\ &= \left({}^f\phi_1 + {}^f\phi_2 \right) \otimes {}^fL_{f1}\dot{\theta}_1 + {}^f\phi_2 \otimes {}^fL_{12}\dot{\theta}_2 \\ &= f_{\theta_1}\dot{\theta}_1 + f_{\theta_2}\dot{\theta}_2 \end{aligned} \quad (3.23)$$

whith f_{θ_1} and f_{θ_2} the non-lagrangian components of links one and two, respectively. It is now clear the importance of the L matrix, introduced in Section 2.1.2: we can write the non-lagrangian components as a pseudo-scalr product between the overall set of action applied to the system downstream and the relative motion permitted by the joint i :

$$f_{q_i} = \left(\sum_{j=1}^N {}^f\phi_j \right) \otimes {}^fL_{q_i} \quad (3.24)$$

It can be proved that ideal constraints between links don't affect the Lagrange equation: we can neglet the constraint forces in ϕ .

Finally, we can write the kinetic energy of the body j as follows:

$$T_j = \frac{1}{2} \text{Tr} \left({}^fW_{fj} {}^fJ_j {}^fW_{fj}^T \right) \quad (3.25)$$

and, from equation (3.10), we still get (3.11).

Regarding the planar VMS system, it is useful to write the pseudo inertial tensor starting from the local frame: since the body is rigid, the mass is supposed to remain in the same position with respect to the local frame.

The pseudo moments of inertia are found starting from the moments of inertia

Table 3.2: Moments of inertia of the planar VMS' bodies, with respect to their centre of mass (in the centre of the base and the arms), with $i = \{1, 2\}$.

Body	Tensor of Pseudo Inertia
Base	$\begin{bmatrix} \frac{1}{4}m_b r^2 & 0 & 0 \\ 0 & \frac{1}{4}m_b r^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Arms	$\begin{bmatrix} \frac{1}{3}m_i l_i^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

(Table 3.1) and following equation (3.20), as shown in Table 3.2. Thus:

$$\begin{aligned}
 {}^0 J_b &= \begin{bmatrix} \frac{m_b r^2}{4} & 0 & 0 & 0 \\ 0 & \frac{m_b r^2}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_b \end{bmatrix} & {}^1 J_1 &= \begin{bmatrix} \frac{l_1^2 m_1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{l_1 m_1}{2} \\ 0 & 0 & 0 & 0 \\ \frac{l_1 m_1}{2} & 0 & 0 & m_1 \end{bmatrix} \\
 {}^2 J_2 &= \begin{bmatrix} \frac{l_2^2 m_2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{l_2 m_2}{2} \\ 0 & 0 & 0 & 0 \\ \frac{l_2 m_2}{2} & 0 & 0 & m_2 \end{bmatrix}
 \end{aligned} \tag{3.26}$$

Then, following equation (2.12), we can project ${}^0 J_b$, ${}^1 J_1$ and ${}^2 J_2$ in the fixed frame, obtaining ${}^f J_b$, ${}^f J_1$ and ${}^f J_2$.

It is already possibile to write the kinetic energy (3.25) for every body, since we have both the velocity matrices and the tensors of pseudo inertia projected in the fixed frame:

$$\begin{aligned}
 T_b &= \frac{1}{2} \text{Tr} \left({}^f W_{f0} {}^f J_0 {}^f W_{f0}^T \right) \\
 T_1 &= \frac{1}{2} \text{Tr} \left({}^f W_{f1} {}^f J_1 {}^f W_{f1}^T \right) \\
 T_2 &= \frac{1}{2} \text{Tr} \left({}^f W_{f2} {}^f J_2 {}^f W_{f2}^T \right)
 \end{aligned} \tag{3.27}$$

To find the non lagrangian components, the action and the L matrices need to be calculated.

We have already found the L matrices associated with the permitted motion of each body in Chapter 2 (see equation (2.29)). The action matrices can be easily

found by looking at the definition of ϕ (3.21) and confronting it with Figure 3.1:

$$\begin{aligned} {}^0\phi_b &= \begin{bmatrix} 0 & -\tau_0 + \tau_1 & 0 & F_x \\ \tau_0 - \tau_1 & 0 & 0 & F_y \\ 0 & 0 & 0 & 0 \\ -F_x & -F_y & 0 & 0 \end{bmatrix} & {}^1\phi_1 &= \begin{bmatrix} 0 & -\tau_1 + \tau_2 & 0 & 0 \\ \tau_1 - \tau_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ {}^2\phi_2 &= \begin{bmatrix} 0 & -\tau_3 & 0 & 0 \\ \tau_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (3.28)$$

After the projection of the action matrices in the fixed frame, it is possible to find the non lagrangian components:

$$\begin{aligned} f_1 &= \left({}^f\phi_b + {}^f\phi_1 + {}^f\phi_2 \right) \otimes {}^fL_{fb_x} = F_x \cos \theta_0 - F_y \sin \theta_0 \\ f_2 &= \left({}^f\phi_b + {}^f\phi_1 + {}^f\phi_2 \right) \otimes {}^fL_{fb_y} = F_x \sin \theta_0 + F_y \cos \theta_0 \\ f_3 &= \left({}^f\phi_b + {}^f\phi_1 + {}^f\phi_2 \right) \otimes {}^fL_{f0} = \tau_0 \\ f_4 &= \left({}^f\phi_b + {}^f\phi_1 \right) \otimes {}^fL_{01} = \tau_1 \\ f_5 &= {}^f\phi_b \otimes {}^fL_{12} = \tau_2 \end{aligned} \quad (3.29)$$

which are the same of (3.13).

It is now possible to solve the Lagrange equation, with $\mathcal{L} = T_b + T_1 + T_2$:

$$\begin{cases} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}_b} - \frac{\partial \mathcal{L}}{\partial x_b} = f_1 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{y}_b} - \frac{\partial \mathcal{L}}{\partial y_b} = f_2 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_0} - \frac{\partial \mathcal{L}}{\partial \theta_0} = f_3 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} - \frac{\partial \mathcal{L}}{\partial q_1} = f_4 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_2} - \frac{\partial \mathcal{L}}{\partial q_2} = f_5 \end{cases} \quad (3.30)$$

3.2.1 Object's Dynamics

The object's pseudo inertia tensor follows the same structure of the base's one:

$${}^{O_1}J_{O_1} = \begin{bmatrix} \frac{m_b r^2}{4} & 0 & 0 & 0 \\ 0 & \frac{m_b r^2}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_b \end{bmatrix} \quad (3.31)$$

and ${}^f J_b = {}_{O_1}^f M {}^{O_1} J_{O_1} {}^{O_1} {}_f M$.

The velocity and L matrices have already been found in the previous chapter; the action matrix can be written with the same assumption made for the classic method approach:

$${}^{O_1} \phi_{O_1} = \begin{bmatrix} 0 & -\tau_O & 0 & F_{O_x} \\ \tau_O & 0 & 0 & F_{O_y} \\ 0 & 0 & 0 & 0 \\ -F_{O_x} & -F_{O_y} & 0 & 0 \end{bmatrix} \quad (3.32)$$

and ${}^f \phi_{O_1} = {}_{O_1}^f M {}^{O_1} \phi_{O_1} {}^{O_1} {}_f M$. Finally:

$$T_O = \frac{1}{2} \text{Tr} \left({}^f W_{fO_1} {}^f J_b {}^f W_{fO_1}^T \right) \quad (3.33)$$

and:

$$\begin{aligned} f_{O_1} &= {}^f \phi_{O_1} \otimes {}^f L_{fO_x} = F_{O_x} \cos \theta_O - F_{O_y} \sin \theta_O \\ f_{O_2} &= {}^f \phi_{O_1} \otimes {}^f L_{fO_y} = F_{O_y} \cos \theta_O + F_{O_x} \sin \theta_O \\ f_{O_3} &= {}^f \phi_{O_1} \otimes {}^f L_{O_0 O_1} = \tau_O \end{aligned} \quad (3.34)$$

By solving (3.11), we get the same matrices in (3.17).

Chapter 4

Impact Analysis

Once the Kinematics and the Dynamics of both the VMS and the external object are analysed, the impact analysis between the two is here performed.

In [13] and [14], two basic assumptions are made in the formulation of the impact model:

1. although the generalized velocities change substantially, the generalized coordinates of the system remain the same over the impact duration. This is justified not only for the rotational coordinates but for the elastic ones as well [15]. The latter conclusion can also be implied by reasoning that since the duration of impact (approximately 10^{-4} s) is very small compared with the period of the fundamental mode of vibration of the links (several seconds), the change in deflection of a flexible beam during impact is negligible;
2. at the contact point between the end-defector and the target there are forces but not moments. Impact occurs at a single point, which is unable to transmit a local moment.

Another assumption that has been already done in the previous chapters is that we know in advance the inertia and dynamic properties of the payload, as well as the position of the contact point on the payload.

Following the second assumption, during the impact the equations of motion of the VMS and the object can be expressed in a form similar to equation (3.5) except for the impact force f_I :

$$\begin{cases} M\ddot{p} + C = u + J^T f_I \\ M_O\ddot{\psi} + C_O = -J_O^T f_I \end{cases} \quad (4.1)$$

where $J^T f_I$ and $J_O^T f_I$ are the torques and forces applied to the VMS and object's generalized coordinates respectively, following the relation shown in (3.4). For the second assumption, the wrench is now composed of the force f_I only.

Notice that u_O has been omitted in the second equation, since, as already mentioned

in Section 3.1.1, we are not interested in controlling the object.

From the second equation of (4.1), we can write:

$$f_I = -(J_O^T)^+ (M_O \ddot{\psi}_O + C_O) \quad (4.2)$$

with $(J_O^T)^+$ the pseudoinverse of the transposed object's jacobian; in fact, since as found in (2.33) this jacobian is not squared, we can use the pseudoinverse to invert it and, by having linearly independent rows, it is defined accordingly to (2.18).

By substituting (4.2) in the first equation of (4.1), one obtaines:

$$\begin{aligned} M\ddot{p} + C &= u - J^T (J_O^T)^+ (M_O \ddot{\psi}_O + C_O) \\ &= u - J^T (J_O^+)^T (M_O \ddot{\psi}_O + C_O) \end{aligned} \quad (4.3)$$

Now, integrating the above equation over the period od impact π :

$$\int_0^\pi M\ddot{p} dt + \int_0^\pi C dt = - \int_0^\pi J^T (J_O^+)^T M_O \ddot{\psi}_O dt + \int_0^\pi (u - J^T (J_O^+)^T C_O) dt \quad (4.4)$$

Under the first assumption of no change of the generalized coordinates during the impact (but only of their velocities), all the contributions that depend only on the velocities and positions (i.e. $C, u, J^T (J_O^+)^T C_O$) can be ignored:

$$M(\dot{p}_f - \dot{p}_i) + J^T (J_O^+)^T M_O (\dot{\psi}_f - \dot{\psi}_i) = 0 \quad (4.5)$$

This equation can be used for all collisions ranging from plastic to perfectly elastic [15,16]. It is in fact the general property of conservation of momentum: given two point masses m_1 and m_2 , with linear initial velocity $v_{1,i}$ and $v_{2,i}$, the conservation of momentum law states:

$$m_1 v_{i,1} + m_2 v_{i,2} = m_1 v_{f,1} + m_2 v_{f,2} \quad (4.6)$$

or, equivalently:

$$m_1(v_{f,1} - v_{i,1}) + m_2(v_{f,2} - v_{i,2}) = 0 \quad (4.7)$$

Equation (4.7) is very similar to (4.5), with the difference that $M\dot{p} \in \mathbb{R}^{5 \times 1}$ and $M_O \dot{\psi} \in \mathbb{R}^{3 \times 1}$: they cannot be summed. That is why the object's mass matrix is multiply by the two jacobians, which comes directly from the equation of motion. In the following sections two different scenarios will be analysed by assuming the manipulator's arms as rigid or elastic bodies.

4.1 Rigid Bodies

In a plastic impact, the velocity of the contact point of each system is the same after the impact:

$$J\dot{p}_f = J_O\dot{\psi}_f \quad (4.8)$$

where the left-hand side term of the equation is the velocity of the contact point written as a function of the final VMS generalized coordinates and velocities p_f, \dot{p}_f (see (2.17)) and the right-hand side term is the velocity of the same point written with respect to the object generalized final coordinates and velocities $\psi_f, \dot{\psi}_f$.

From (4.8), the velocities of the object can be expressed in terms of those of the VMS as:

$$\dot{\psi}_f = J_O^+ J\dot{p}_f \quad (4.9)$$

By substituting equation (4.9) in equation (4.5), the following is obtained:

$$\dot{p}_f = G^{-1} H \quad (4.10)$$

where:

$$\begin{aligned} G &= M + J^T (J_O^+)^T M_O J_O^+ J \\ H &= M\dot{p}_i + J^T (J_O^+)^T M_O \dot{\psi}_i \end{aligned} \quad (4.11)$$

We can confront this result with the one obtained starting from equation (4.7) in the case of two point masses; since the final velocity in a plastic impact would have been the same v_f for both masses, we can write:

$$\begin{aligned} m_1 v_{i,1} + m_2 v_{i,2} &= (m_1 + m_2) v_f \\ \Rightarrow v_f &= g^{-1} h \end{aligned} \quad (4.12)$$

with:

$$\begin{aligned} g &= m_1 + m_2 \\ h &= m_1 v_{i,1} + m_2 v_{i,2} \end{aligned} \quad (4.13)$$

Notice that G, H are very similar to g, h , the difference is still due to the jacobians matrices which come from the kinematic relation.

Once the initial velocities $\dot{p}_i, \dot{\psi}_i$ have been determined, \dot{p}_f can be evaluated according to (4.10) and $\dot{\psi}_f$ can be solved from (4.9). These final values are used as the initial velocity conditions for the post-impact dynamical simulation of the system, whose model will be the combination of the equations of robot and object, so that equation (4.9) will always hold and we can get rid off the subscripts:

$$\dot{\psi} = J_O^+ J\dot{p} \quad (4.14)$$

While in [14] the post-impact analysis is not fully covered, [16] offers an intuitive procedure to write the dynamics of the new VMS, composed now by the object too, with respect of the VMS generalized coordinates alone (i.e. p).

By differentiating the above equation we can obtain:

$$\ddot{\psi} = J_O^+ J \ddot{p} + \frac{\partial J_O^+}{\partial t} J \dot{p} + J_O^+ \frac{\partial J}{\partial t} \dot{p} \quad (4.15)$$

We can now substitute (4.15) in the second equation of (4.1) (i.e. the dynamics of the object), obtaining:

$$M_O \left(J_O^+ J \ddot{p} + \frac{\partial J_O^+}{\partial t} J \dot{p} + J_O^+ \frac{\partial J}{\partial t} \dot{p} \right) + C_O = -J_O^T f_I \quad (4.16)$$

from which f_I can be written as:

$$f_I = - \left(J_O^{-T} M_O J_O^+ J \ddot{p} + J_O^{-T} M_O \frac{\partial J_O^+}{\partial t} J \dot{p} + J_O^{-T} M_O J_O^+ \frac{\partial J}{\partial t} \dot{p} + J_O^{-T} C_O \right) \quad (4.17)$$

Now, f_I can be substituted in the first equation of (4.1) (i.e. the VMS dynamics) and by doing so, we can write the overall dynamics with respect to p only:

$$M' \ddot{p} + C' = u \quad (4.18)$$

where:

$$\begin{aligned} M' &= M + J^T (J_O^T)^+ M_O J_O^+ J \\ C' &= C + J^T (J_O^T)^+ M_O \frac{\partial J_O^+}{\partial t} J \dot{p} + J^T (J_O^T)^+ M_O J_O^+ \frac{\partial J}{\partial t} \dot{p} + J^T (J_O^T)^+ C_O \end{aligned} \quad (4.19)$$

Equation (4.19) is the post-impact dynamic model of the VMS containing the object. Both M' and C' are still dependent on J_O , which depends on the object's coordinates, but since the object is now captured by the manipulator, its coordinates coincide with the end-effector ones, and its velocities are found by differentiation.

4.1.1 Free Motion

A simulation has been performed by means of the Mathematica software (version 13.2.1.0) with VMS parameters shown in Table 4.1 and the initial conditions shown in Tables 4.2, 4.3.

The VMS parameters have been chosen following a real case scenario: the lengths, radiiuses and masses of the manipulator's arms are the ERA ones (see Table 1.1 and [4]); the base mass is the mass of the ISS, on top of which the ERA is attached; the payload mass is the mass of a medium-size satellite; γ is an arbitrary value.

Regarding the VMS initial positions, the choice of the values will be explained in the next section. For this analysis, every initial position is valid. The initial velocities are zero for almost every VMS generalized coordinate (i.e we assume the base and manipulator still before the impact) and small initial velocities for the payload, as the case of satellite docking is taken into consideration: the ideal condition would be a null relative velocities, however, in a real case scenario, a non-zero relative velocities is present.

The position of the object in the pre-impact has been found by imposing that the contact point coincides with the initial end-effector position, while its orientation θ_O has been chosen as follows:

$$\theta_O = \theta_{0,0} + q_{1,0} + q_{2,0} + \pi - \gamma \quad (4.20)$$

which is, the orientation of the object is such that it is rotated of the same end-effector angle ($\theta_{0,0} + q_{1,0} + q_{2,0}$) minus γ to align the x-axis with the EE's one. In order to make the contact point coincide with the EE, 180° have been added.

As shown in Table 4.3, two simulations have been performed, with the only difference in the object's initial velocities: in the first simulation, the payload moved only in the negative x-direction, in the second it moves only in the negative y-direction. Given these initial conditions, the final velocities \dot{p}_f in (4.10) can be calculated.

For the post-impact analysis, the position of the object have been imposed again as a function of the EE position, but now as a function of time, and its orientation

Table 4.1: VMS parameters.

l_1	l_2	m_b	m_1	m_2	m_O	r	γ
5.59 m	5.59 m	419 725 kg	300 kg	300 kg	3000 kg	0.12 m	0.5 rad

Table 4.2: Simulation’s initial positions.

x_b	y_b	θ_0	q_1	q_2	x_O	y_O	θ_O
4 m	2 m	$\pi/2$ rad	0 rad	$\pi/2$ rad	-1.71 m	7.59 m	5.78 rad

Table 4.3: Simulation’s initial velocities.

	Simulation 1	Simulation 2
$\dot{\theta}_0$	0 rad s ⁻¹	0 rad s ⁻¹
\dot{x}_b	0 m s ⁻¹	0 m s ⁻¹
\dot{y}_b	0 m s ⁻¹	0 m s ⁻¹
\dot{q}_1	0 rad s ⁻¹	0 rad s ⁻¹
\dot{q}_2	0 rad s ⁻¹	0 rad s ⁻¹
\dot{x}_O	1 m s ⁻¹	0 m s ⁻¹
\dot{y}_O	0 m s ⁻¹	-1 m s ⁻¹
$\dot{\theta}_O$	0.01 rad s ⁻¹	0.01 rad s ⁻¹

similarly to equation (4.20):

$$\theta_O(t) = \theta_0(t) + q_1(t) + q_2(t) + \pi - \gamma \quad (4.21)$$

Having now the new post-impact ψ_p generalized coordinates as a function of p , the object’s jacobian can be expressed as a function of the VMS generalized coordinates too and M' and C' can be found according to (4.19).

A control-free dynamics has been performed using as initial conditions the initial positions in Table 4.2 and the final velocities \dot{p}_f found in the pre-impact analysis. As already stated at the beginning of this chapter, in fact, during the impact only the rates change, while the positions remain the same.

A quick and intuitive visual result can be seen in Figure 4.1, while the evolution of the generalized coordinates over time are reported in Figures 4.2, 4.3.

It is interesting to notice that, in both cases, θ_0 does not change over time: the absence of a torque applied to the base allows free rotation of q_1 around z_b , since the manipulator is directly attached to the CoM of the base (i.e. the arm does not apply any torque) and the base does not rotate.

Furthermore, given the higher mass of the base with respect to the payload and manipulator masses, it moves very little and we can approximate it as still in the next calculations.

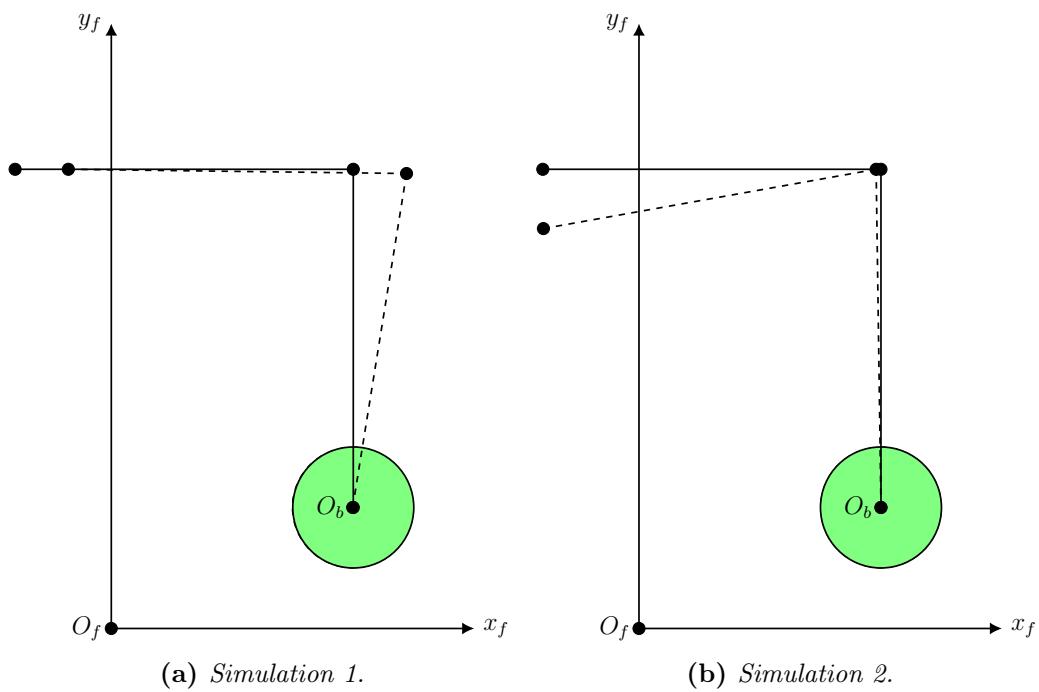


Figure 4.1: Initial and final (dashed) position of the VMS after the impact for the two different simulations. In the left image, the satellite approaches the VMS with a west to east motion, while in the right one with a north to south motion.

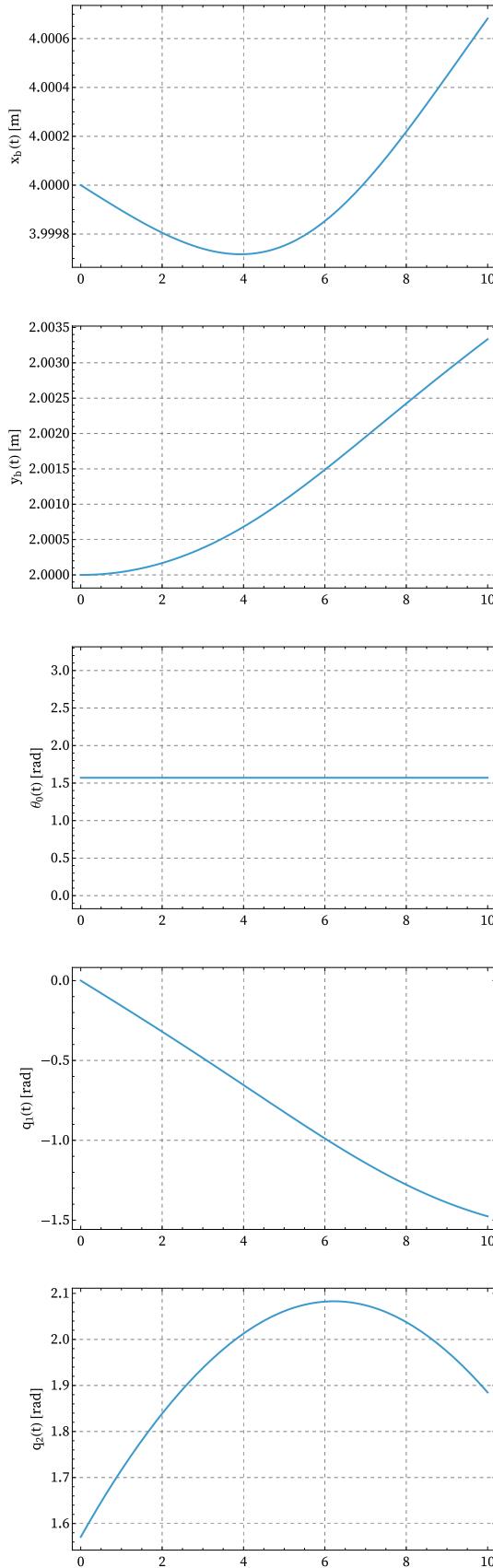


Figure 4.2: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, simulation one.

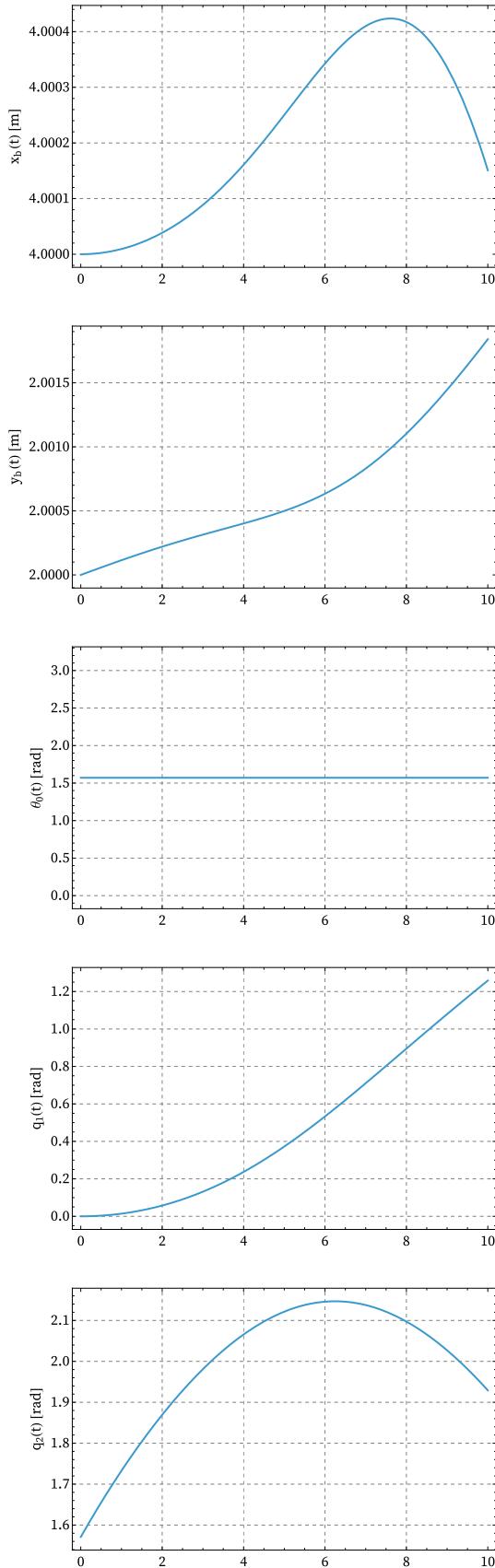


Figure 4.3: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, simulation two.

4.1.2 Control Based Motion

A typical control block diagram is shown in Figure 4.4. The sensors allow to keep track of the robot configuration q, \dot{q} such that the controller can provide a control law to follow a desired behaviour.

A remark needs to be done about the control of the translation of the base. It is highly improbable that it would be possible to calculate the exact displacement of the spacecraft where the manipulator is attached, considering also the too little displacement from Figures 4.2, 4.3. Hence, the control law is applied to θ_0, q_1 and q_2 only. The base is free to move but, as already mentioned, it can be approximate as still, given the huge inertia of the spacecraft.

Thus, we can rewrite the final equation of motion (4.18):

$$\begin{bmatrix} M'_{tt} & M'_{tr} \\ M'_{rt} & M'_{rr} \end{bmatrix} \begin{bmatrix} \ddot{p}_t \\ \ddot{p}_r \end{bmatrix} + \begin{bmatrix} C'_t \\ C'_r \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ u \end{bmatrix} \quad (4.22)$$

where $p_t = \{x_b, y_b\}$ refers to the translational coordinates, $p_r = \{\theta_0, q_1, q_2\}$ to the rotational ones and $\mathbf{0} \in \mathbb{R}^2$.

By writing the translational coordinate as a function of the rotational ones

$$\ddot{p}_t = -M'^{-1}_{tt}(M'_{tr}\ddot{p}_r + C'_t) \quad (4.23)$$

and by plugging it in the second row of (4.22):

$$\ddot{p}_r \tilde{M} + \tilde{C} = u \quad (4.24)$$

with

$$\tilde{M} = M'_{rr} - M'_{rt}M'^{-1}_{tt}M'_{tr}, \quad \tilde{C} = C'_r - M'_{rt}M'^{-1}_{tt}C'_t \quad (4.25)$$

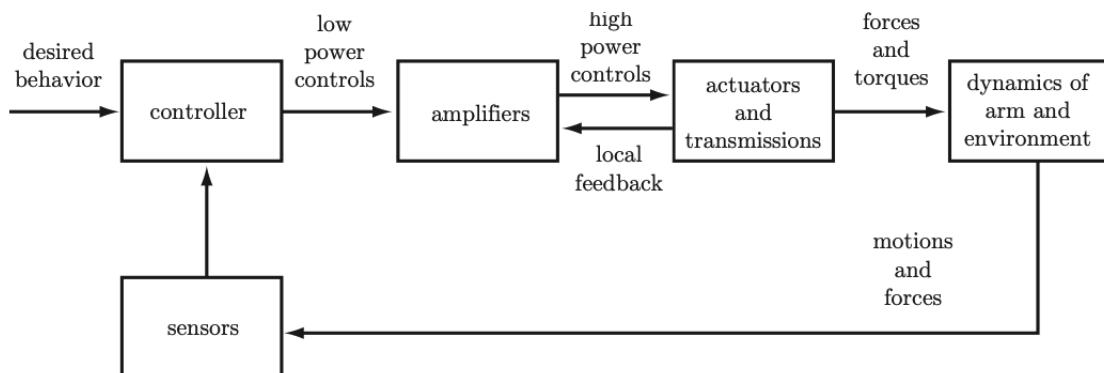


Figure 4.4: A typical robot control system. An inner control loop is used to help the amplifier and actuator to achieve the desired force or torque [11].

If we knew perfectly the robot dynamics, we could use the following input (*feed-forward*) to move the robot:

$$\hat{u} = \hat{M}\ddot{q}_d + \hat{C}(q_d, \dot{q}_d) \quad (4.26)$$

where the RHS of the equation is the estimated robot dynamics.

Upon substitution in (4.24), we would have, supposing $\hat{M} = \tilde{M}$ and $\hat{C} = \tilde{C}$:

$$\tilde{M}(\ddot{q}_d - \ddot{q}) = 0 \quad (4.27)$$

which is true when the desired configuration is reached.

Since in practice it is not possible to know the exact dynamics, a PD controller is often used in addition to the feed-forward term to compensate for inaccuracy and disturbances:

$$u = \hat{u} + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) \quad (4.28)$$

with K_p and K_d the proportional and derivative positive definite gain matrices.

Assuming then $\hat{M} \approx \tilde{M}$ and $\hat{C} \approx \tilde{C}$ and substituting in (4.24):

$$\tilde{M}(\ddot{q}_d - \ddot{q}) + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) = 0 \quad (4.29)$$

However, the above equation is coupled by the mass matrix. In order to have an uncoupled control scheme, a *feedback linearization* can be implemented, as suggested by [16]:

$$u = \hat{M}[\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)] + \hat{C} \quad (4.30)$$

such that the final equation of motion for the rotational coordinates becomes:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (4.31)$$

The tracking error e is governed by a second order dynamics that can be arbitrarily assigned (on each joint) by suitably selecting gains (K_p, K_d) .

The dynamics is now:

- decoupled: each joint coordinate evolves independently from others $\Rightarrow K_p, K_d$ are diagonal matrices.
- linear: we have global asymptotic ($e \rightarrow 0$) stability for any $K_p > 0, K_d > 0$. This is a sufficient and necessary condition since the state-space equation is given by:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} 0 & I_{2 \times 2} \\ -K_p & -K_d \end{bmatrix} \quad (4.32)$$

and, for second order differential equations, if $K_{p_i} > 0$ and $K_{d_i} > 0$, the

dynamics is stable [11].

- the time evolution is governed by the eigenvalues that are the roots of the polynomial:

$$s^2 + K_{d_i}s + K_{p_i} = 0 \quad \Rightarrow \quad s_{1,2} = \frac{-K_{d_i} \pm \sqrt{K_{d_i}^2 - 4K_{p_i}}}{2} \quad (4.33)$$

with $\omega_n = \sqrt{K_{p_i}}$ and $\xi = \frac{K_{d_i}}{2\sqrt{K_{p_i}}}$.

The control law shown in equation (4.30) has been chosen with zero desired velocities \dot{p}_d and with the initial positions (Table 4.2) as desired coordinates p_d . K_p has been chosen as the identity matrix (i.e. $I_{5 \times 5}$) and K_d has been calculated following the relation of the damping term shown above and choosing a critically damped behaviour, so that there are no overshoots of the links while reaching the desired position:

$$\xi = 1 \quad \Rightarrow \quad K_d = 2\sqrt{K_p} \quad (4.34)$$

The final results are shown in Figures 4.5, 4.6, where underdamped and overdamped behaviours have been plotted as well.

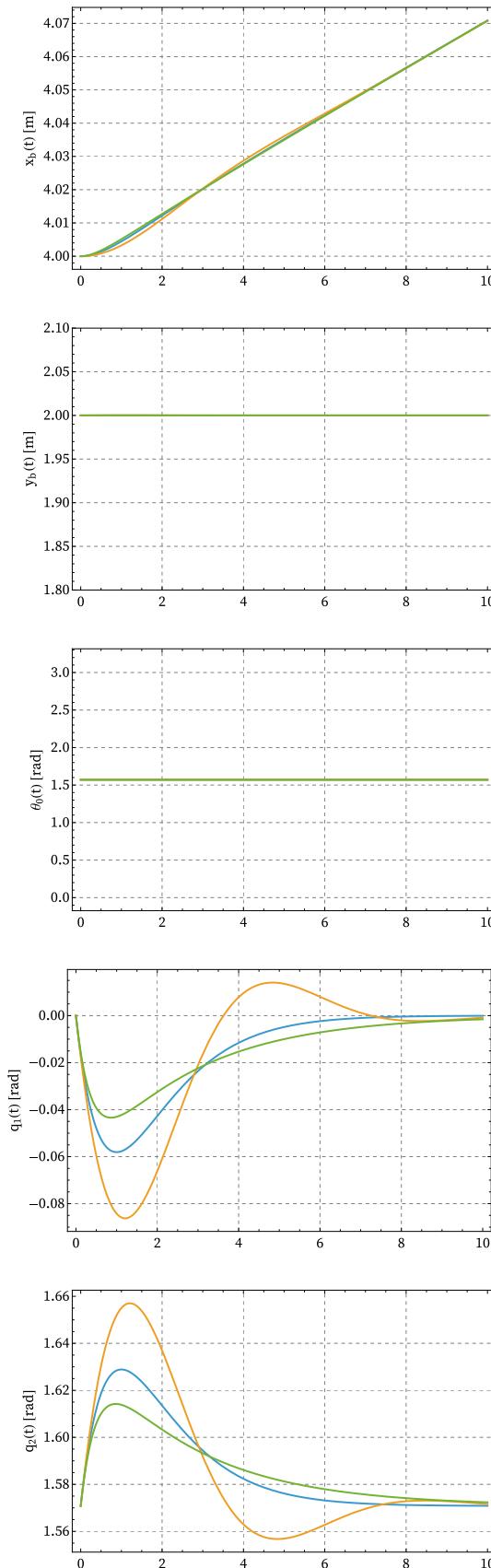


Figure 4.5: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, simulation one. In yellow, an underdamped behaviour ($\xi = 0.5$), in blue a critically damped behaviour ($\xi = 1$), in green an overdamped behaviour ($\xi = 1.5$).

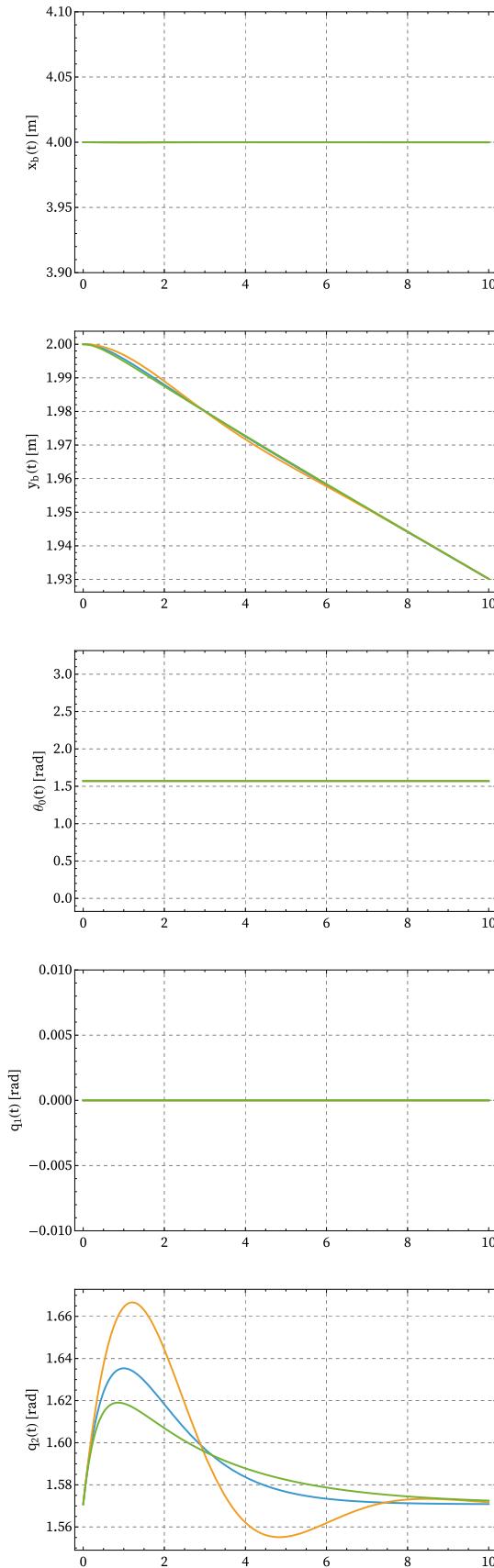


Figure 4.6: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, simulation two.

4.2 Elastic Bodies

In this section, a more realistic scenario is developed and the assumption of rigid bodies will be discarded: the links of the manipulator will be considered flexible. The robotic systems with flexible links are continuous dynamical systems characterized by an infinite number of degrees of freedom and are governed by nonlinear, coupled, ordinary, and partial differential equations. The exact solution of such a mathematical model is normally not feasible, and the infinite dimensional model imposes severe constraint on the design of controllers as well. Hence it is necessary that these continuous systems are discretized.

Flexible links with rigid joints in robot manipulators are discretized by using assumed modes, finite elements, or lumped parameters. The link deflections, with respect to the link's rigid configuration, are usually assumed to be small, and a linear theory of elasticity is selected. Researchers commonly utilize the Euler-Bernoulli beam theory to represent the dynamics of link flexibility, which is the one selected for this work [17, 18, 25].

The lateral vibration of a beam can be studied by considering it as a continuous system, characterized by an infinite number of degrees of freedom. The equation of motion is carried out by considering the forces and moments acting on an infinitesimal piece of the body (Figure 4.7), whose mass can be written as:

$$dm = \rho Adx \quad (4.35)$$

where ρ is the beam density and A its cross-section area, which is considered constant along the beam.

The equations of motion are:

$$\begin{cases} f(x, t)dx - (V + dV) + V = \rho A dx \frac{\partial^2 w(x, t)}{\partial t^2} \\ (M + dM) - (V + dV)dx + f(x, t)dx \frac{dx}{2} - M = 0 \end{cases} \quad (4.36)$$

where V is the shear force, M the bending moment and $f(x, t)$ the distributed force. The dominant motion is assumed to be the lateral one, which is true when the beam is thin.

By defining $dV = \frac{\partial V}{\partial x} dx$ and $dM = \frac{\partial M}{\partial x} dx$, and using the Euler-Bernoulli definition of moment $M(x, t) = ES \frac{\partial^2 w(x, t)}{\partial x^2}$, where E is the Young's modulus and S is the area moment of inertia (not to be confused with the mass moment of inertia I), the equation of motion becomes:

$$ES \frac{\partial^4 w(x, t)}{\partial x^4} + \rho A \frac{\partial^2 w(x, t)}{\partial t^2} = f(x, t) \quad (4.37)$$

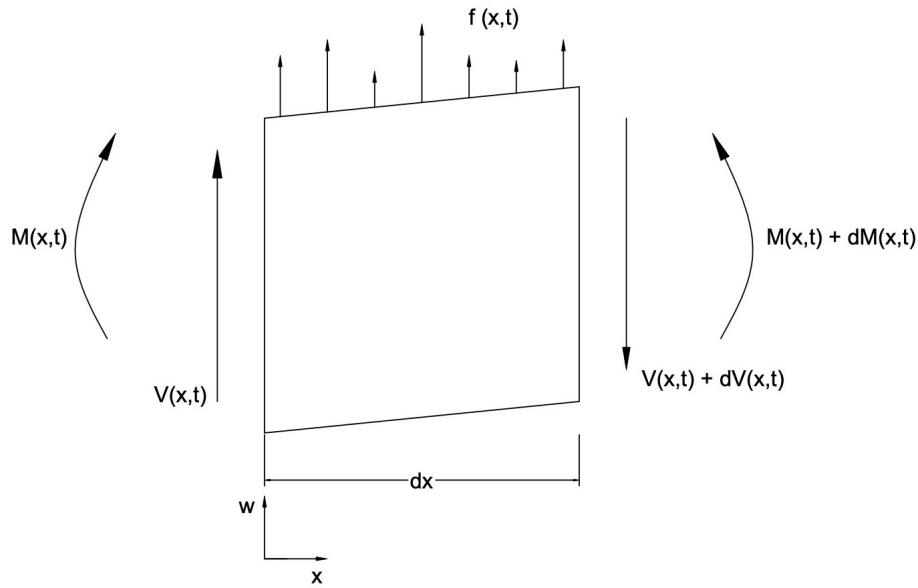


Figure 4.7: Forces acting on the infinitesimal piece of the beam.

and, by defining $c = \sqrt{\frac{ES}{\rho A}}$ as the speed of longitudinal deformation, we get:

$$c^2 \frac{\partial^4 w(x,t)}{\partial x^4} + \frac{\partial^2 w(x,t)}{\partial t^2} = f(x,t) \quad (4.38)$$

ES can also be referred to *flexural rigidity* of the beam.

The free vibration equation is obtained by imposing $f(x,t) = 0$:

$$c^2 \frac{\partial^4 w(x,t)}{\partial x^4} + \frac{\partial^2 w(x,t)}{\partial t^2} = 0 \quad (4.39)$$

which is also reported in [18]: we assume that no control is applied to the tips' deformations. This assumption will be tackled in the next sections.

The solution of the equation (4.39) needs two initial condition and four boundary condition to be found. It can be calculated by separation of variables, writing the solution as the product of a function depending on time and another one depending on space:

$$w(x,t) = W(x)Q(t) \quad (4.40)$$

It is now possible to rewrite the equation (4.39) as:

$$\frac{c^2}{W(x)} \frac{d^4 W(x)}{dx^4} = -\frac{1}{Q(t)} \frac{d^2 Q(t)}{dt^2} = const \quad (4.41)$$

and by choosing as the constant ω^2 , the time equation becomes the one degree of freedom oscillator, whose solution is known (4.42), while the space equation

solution can be written as (4.43), where $\beta^4 = \frac{\rho A}{EI} \omega^2$.

$$Q(t) = A \cos \omega t + B \sin \omega t \quad (4.42)$$

$$W(x) = c_1 \cos(\beta x) + c_2 \sin \beta x + c_3 \cosh \beta x + c_4 \sinh \beta x \quad (4.43)$$

The four coefficient of the space solution can be found with the boundary condition of the system. Every natural frequency of the system is associated to one space solution (mode shape) and time solution, and can be found imposing the boundary conditions.

The displacement of the beam is the sum of the product between the n mode shapes selected and their time solutions [16,17,18]:

$$w(x, t) = \sum_{n=1}^{\infty} W_n(x) Q_n(t) \quad (4.44)$$

As already mentioned, the assumption for the Euler-Bernoulli beams model is of small deformations. The kinematic analysis can be carried out in the same way we did in Section 2.2, with the only addition of homogeneous elasticity matrix, which accounts for the elastic deformation of the link [17]. Assuming null rotation and small tip deformation, we can write the elasticity matrix from link i to $i+1$ as:

$${}_{i+1}^i E = \begin{bmatrix} 1 & 0 & 0 & u(x, t) \\ 0 & 1 & 0 & w(x, t) \\ 0 & 0 & 1 & v(x, t) \end{bmatrix} \quad (4.45)$$

with $u(x, t)$, $w(x, t)$, $v(x, t)$ elastic deformations in the x , y , z direction respectively. Since the deformation is calculated at the arm's tip and only in the y -direction (following the Denavit-Hartenberg notation), we can rewrite the matrix:

$${}_{i+1}^i E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & W_i(x) Q_i(t) \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.46)$$

where $W_i(x) = \{W_{i,1}, W_{i,2}, \dots, W_{i,n}\}$ and $Q_i(t) = \{Q_{i,1}, Q_{i,2}, \dots, Q_{i,n}\}^T$ [16].

The joints' homogeneous transformation matrices, for the analyzed planar case, will be given as:

$$\begin{aligned} {}_1^f T &= {}_b^f M {}_0^b M {}_1^0 M {}_1^0 E \\ {}_2^f T &= {}_b^f M {}_0^b M {}_1^0 M {}_1^0 E {}_2^1 M {}_2^1 E \end{aligned} \quad (4.47)$$

Hence, the time-varying part of the free-vibration solution $Q(t)$ can be found by following the lagrangian approach as in the previous sections, where the generalized

coordinate vector of the system is now:

$$p = \{x_b, y_b, \theta_b, q_1, q_2, Q_1, Q_2\} \quad (4.48)$$

and its dimension depends on the number of modeshapes chosen (e.g. for $n = 1$, $p \in \mathbb{R}^7$).

For what regards the space-varying part of the free vibration solution $W(x)$, four boundary conditions need to be imposed. It is reasonable to suppose that the inertia of a lightweight link is small compared to the hub inertia, and then constrained mode shapes can be used. In particular, we assume each link to be clamped at the base:

$$w_i(0, t) = 0, \quad w'_i(0, t) = 0 \quad (4.49)$$

Furthermore, experiments and recent analytical studies have shown that the clamped assumption is even enforced when choosing a feedback control loop around the joint [18].

Concerning the remaining boundary conditions, we consider mass boundary conditions representing balance of moment and shearing force owing to other links of the serial structure of robot manipulators and the payload [17, 18]:

$$\begin{aligned} (ES)_i \frac{\partial^2 w_i(l_i, t)}{\partial x^2} &= -J_{L_i} \frac{d^2}{dt^2} \left(\frac{\partial w_i(l_i, t)}{\partial x} \right) - M_{D_i} \frac{d^2}{dt^2} w_i(l_i, t) \\ (ES)_i \frac{\partial^3 w_i(l_i, t)}{\partial x^3} &= -M_{L_i} \frac{d^2}{dt^2} w_i(l_i, t) + M_{D_i} \frac{d^2}{dt^2} \frac{\partial w_i(l_i, t)}{\partial x} \end{aligned} \quad (4.50)$$

where M_{L_i} , and J_{L_i} , are the actual mass and moment of inertia at the end of link i . Moreover, M_{D_i} accounts for the contributions of masses of distal links, i.e., noncollocated at the-end of link i , weighted by the relative distance from axis y , (shearing axis at the end of link i). Incidentally, these contributions are often not included in mode shape analyses: if the arm has only one link, M_{L_i} and J_{L_i} are directly the payload mass and inertia, while the additional terms on the right-hand side of (4.50) vanish ($M_{D_i} = 0$) only when the payload is balanced at the tip.

For the generic intermediate i th link in an open kinematic chain arrangement, instead, M_{L_i} is the constant sum of all masses beyond link i , but J_{L_i} , and M_{D_i} , depend on the position of successive links. Thus, for exact mode shapes computation, these quantities should be updated as functions of the arm configuration; this may considerably increase the complexity of model derivation, beside overloading the computational burden of on-line execution. Therefore, some practical approximation leading to constant-although nonzero-boundary conditions at the link end might be in order. For instance, a convenient position is to set $M_{D_i} = 0$ and compute J_{L_i} , for a fixed arm configuration.

In the case of a planar RR manipulator, like the one used in this thesis, we would have:

$$\begin{aligned} M_{L_1} &= m_2 + m_p \\ J_{L_1} &= I_{2,z} + I_{O,z} + m_O l_2^2 \\ M_{D_1} &= (m_2 \frac{l_2}{2} + m_O l_2) \cos q_2 - \chi \sin q_2 \end{aligned} \quad (4.51)$$

where χ is a term of an order of magnitude smaller than the first one, hence can be negllected.

Notice that in the considered case of only two links, J_{L_1} is a constant; for more than two links, J_{L_1} will become a function of the generalized coordinate of link 3 and following ones.

On the other hand for link 2:

$$\begin{aligned} M_{L_2} &= m_p \\ J_{L_2} &= I_{O,z} \\ M_{D_2} &= 0 \end{aligned} \quad (4.52)$$

The varying of M_{D_1} in time, since it depends on the arm configuration, implies that the mode shapes, which are spacial quantities, would become implicit function of time, thus conflicting with the original separability assumption (4.40).

However, M_{D_1} can be rendered constant for a fixed arm configuration: assuming small displacement and initial position $q_{2,0} = \pi/2$, leads to $M_{D_1} = 0$. That is why this has been chosen as the manipulator initial condition in both Section 4.1 and this one.

By substituting in the boundary conditions equations the solutions (4.42,4.43), the homogeneous solution for link i and mode shape j is [17, 18]:

$$W_{i,j}(x) = C_{i,j}[\cos(\beta_{i,j}x) - \cosh(\beta_{i,j}x) + \nu_i(\sin(\beta_{i,j}x) - \sinh(\beta_{i,j}x))] \quad (4.53)$$

where:

$$\nu_{i,j} = \frac{\sin(\beta_{i,j}l_i) - \sinh(\beta_{i,j}l_i) + M\beta_{i,j}[\cos(\beta_{i,j}l_i) - \cosh(\beta_{i,j}l_i)]}{\cos(\beta_{i,j}l_i) + \cosh(\beta_{i,j}l_i) - M\beta_{i,j}[\sin(\beta_{i,j}l_i) - \sinh(\beta_{i,j}l_i)]} \quad (4.54)$$

where

$$M = \frac{M_L}{\rho Al}, \quad J = \frac{J_L}{\rho Al^3} \quad (4.55)$$

and $\beta_{i,j}$ are solutions of the frequency equation:

$$\begin{aligned} 1 + \cos(\beta_{i,j}l_i) \cosh(\beta_{i,j}l_i) - M\beta_{i,j}[\cosh(\beta_{i,j}l_i) \sin(\beta_{i,j}l_i) - \sinh(\beta_{i,j}l_i) \cos(\beta_{i,j}l_i)] \\ - J\beta_{i,j}^3[\cosh(\beta_{i,j}l_i) \sin(\beta_{i,j}l_i) + \sinh(\beta_{i,j}l_i) \cos(\beta_{i,j}l_i)] \\ + MJ\beta^4[1 - \cosh(\beta_{i,j}l_i) \cos(\beta_{i,j}l_i)] = 0 \end{aligned} \quad (4.56)$$

The remaining constant in equation (4.53) (i.e. $C_{i,j}$) is found by mass normalizing the mode shape [19, 20]:

$$\int_0^{l_i} \rho_i A_i W_{i,j}^2(x) dx = 1 \quad (4.57)$$

where the RHS of the equation is the modal mass of the j th mode of the i th link and $\rho_i = m_i/l_i$.

Mode shapes, in fact, can be normalized in different ways (normalization to the unit length vector, normalization to the largest component equal to unity), but the mass normalization is the most common one [19, 20], since it allows to have modal displacements with a physical meaning, since they are related to the modal mass. The choice of the normalization, however, does not affect the value of the natural frequencies, which are the same for all the normalizations and depend only on equation (4.56).

Once $W_{i,j}(x)$ are known, the kinematics can be evaluated through (4.47), and the dynamics can be calculated as in the previous sections, with the awareness that it is not possible to neglect the potential energy anymore.

If in the previous sections the absence of gravity and the rigidity of the bodies allowed to ignore it, now the manipulators' arms are modeled as elastic bodies, thus the elastic potential energy needs to be taken into account.

The elastic energy for unit length stored in a bended beam can be written as [21]:

$$\frac{dU}{dx} = \frac{1}{2} \frac{M^2}{ES} \quad (4.58)$$

with M torque moment, which can be expressed as:

$$M = ES\kappa \quad (4.59)$$

where κ is the curvature of the deformation, defined as the derivative of the deformation angle φ :

$$\kappa = \frac{d\varphi}{dx} = \frac{d}{dx} \frac{\partial w(x, t)}{\partial x} = \frac{\partial^2 w(x, t)}{\partial x^2} \quad (4.60)$$

By substituting (4.60) in (4.59) and then in (4.58), by integration along the beam direction we get:

$$\begin{aligned} U &= \frac{1}{2} \int_0^{l_i} ES \left(\frac{\partial^2 w(x, t)}{\partial x^2} \right)^2 dx \\ &= \frac{1}{2} \int_0^{l_i} ES \left(\frac{\partial^2 W(x)}{\partial x^2} Q(t) \right)^2 dx \end{aligned} \quad (4.61)$$

which is the same equation reported in [17, 18].

The new lagrangian will be now:

$$\mathcal{L}_i = T_i - U_i \quad (4.62)$$

and the dynamics of the system is given by:

$$\left\{ \begin{array}{l} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}_b} - \frac{\partial \mathcal{L}}{\partial x_b} = F_x \cos \theta_0 - F_y \sin \theta_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{y}_b} - \frac{\partial \mathcal{L}}{\partial y_b} = F_x \sin \theta_0 + F_y \cos \theta_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_0} - \frac{\partial \mathcal{L}}{\partial \theta_0} = \tau_0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} - \frac{\partial \mathcal{L}}{\partial q_1} = \tau_1 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_2} - \frac{\partial \mathcal{L}}{\partial q_2} = \tau_2 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{Q}_1} - \frac{\partial \mathcal{L}}{\partial Q_1} = 0 \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{Q}_2} - \frac{\partial \mathcal{L}}{\partial Q_2} = 0 \end{array} \right. \quad (4.63)$$

since no torque is applied to control the vibrations at the arms' end.

We can directly find the stiffness matrix, of the system, without calculating from scratch the equations of motion, through the following considerations:

$$\frac{d}{dt} \frac{\partial U}{\partial \dot{Q}_i} - \frac{\partial U}{\partial Q_i} = 0 + \int_0^{l_i} E S \frac{\partial^2 W(x)}{\partial x^2} dx \quad (4.64)$$

Hence, as reported in [16], the new set of dynamical equation is obtained in the form:

$$M \ddot{p} + C + K p = u + J^T f_I \quad (4.65)$$

which is similar to the equation of motion of Chapter 4 (4.1), with the addiction of the stiffness matrix $K = \text{diag}(\mathbf{0}, K_1, K_2)$ where:

$$K_i = \int_0^{l_i} E_i S_i \frac{\partial^2 W_i(x)}{\partial x^2} dx \quad (4.66)$$

Following the same procedure shown in the previous Chapter and in Section 4.1, under the assumption that during the impact no change in the VMS' positions happens (i.e. we can neglect matrix K and follow the exact same calculus), the dynamic equation of the whole system VMS plus payload becomes [16]:

$$M' \ddot{p} + C' + K p = u \quad (4.67)$$

similar to (4.19) and with p as in (4.48).

4.2.1 Free Motion

The frequency equation (4.56) is a trascendental equation, so it has been solved with a finding-root algorithm (Mathematica provides it with a built-in command `FindRoot`).

A free motion simulation has been performed with the same initial condition and parameters of the previous section, adding the new ones exposed in Table 4.4. The arms flexural rigidity ES_i has been calculated by considering a circular cross section with diameter d , which gives us the value of the area moment of inertia:

$$S = \frac{\pi}{64} d^4 \quad (4.68)$$

and as material aluminum, which has a Young's modulus $E \approx 70$ GPa.

Only one mode has been taken into consideration for the complexity of the calculations (i.e. $n = 1$).

The joints' evolution can be seen in Figure 4.8 and 4.9.

As it can be seen, the rigid coordinates have the same behaviour as in the rigid case, as it should be, with the only difference that a little vibration is present.

Furthermore, the initial velocity of the elastic coordinates has the same sign (direction) of the rigid's ones. This makes sense, since when the arm starts to move, it starts to vibrate in the same direction.

It is interesting to notice that for both simulations, one arm's vibration seems to diverge. To better inspect this phenomenon, a longer temporal window of the two joints in the first simulation has been analyzed in Figure 4.10. For some instants, such as for $t = 40$ s, the two arms reach the highest amplitude at the same time, while for others, like $t = 50$ s, one's amplitude increases while the other's reaches its minimum.

If we look at the configuration of the arm at those instant we would see that in the first case the manipulator is fully stretched, meaning that it reaches its singular configuration, while in the second case the angle between the two arms is 90° . It is possible to conclude that, given the non-constant relative position between the manipulator's arms, the vibration of one influences the second's and that when they are aligned they enforce each other vibration, whereas when they are orthogonal one vibration is suppressed by the other.

Table 4.4: VMS elastic parameters.

ρ_1	ρ_2	ES_1	ES_2	A_1	A_2
$53.67 \frac{\text{kg}}{\text{m}}$	$53.67 \frac{\text{kg}}{\text{m}}$	$1.14 \times 10^7 \frac{\text{N}^2}{\text{m}}$	$1.14 \times 10^7 \frac{\text{N}^2}{\text{m}}$	$4.52 \times 10^{-2} \text{ m}^2$	$4.52 \times 10^{-2} \text{ m}^2$

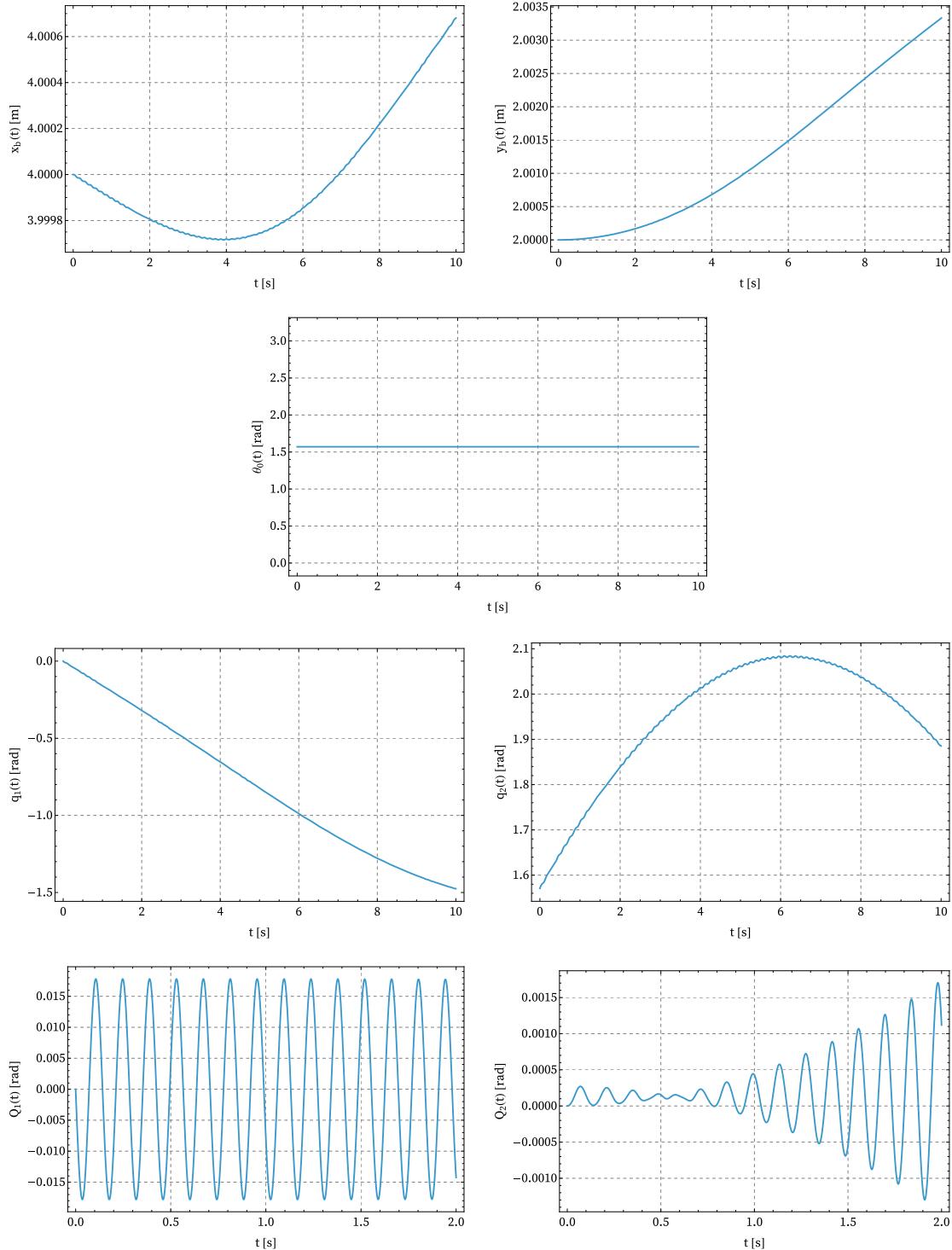


Figure 4.8: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, Simulation 1. Arms modeled as Euler-Bernoulli beams.

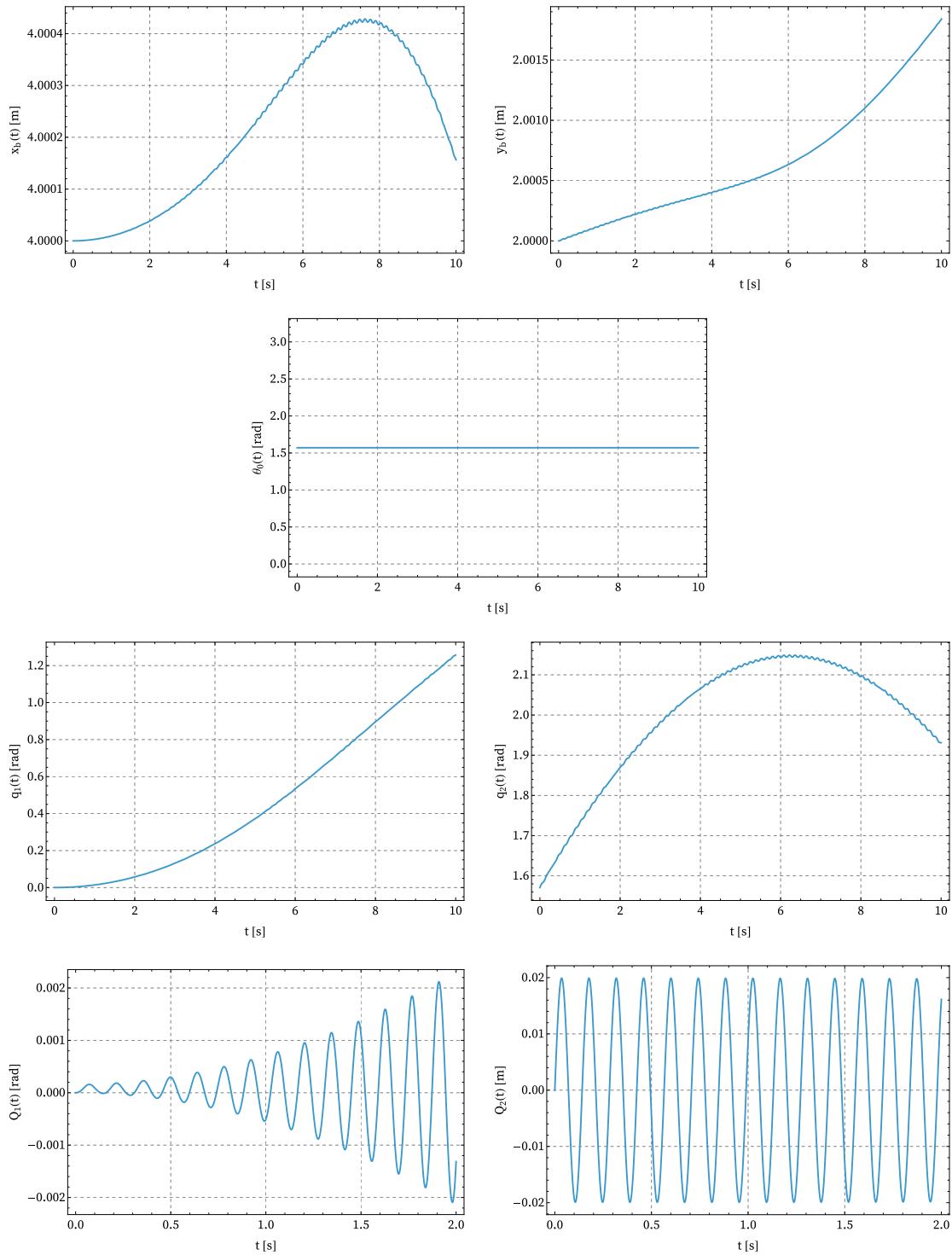


Figure 4.9: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, Simulation 2. Arms modeled as Euler-Bernoulli beams.

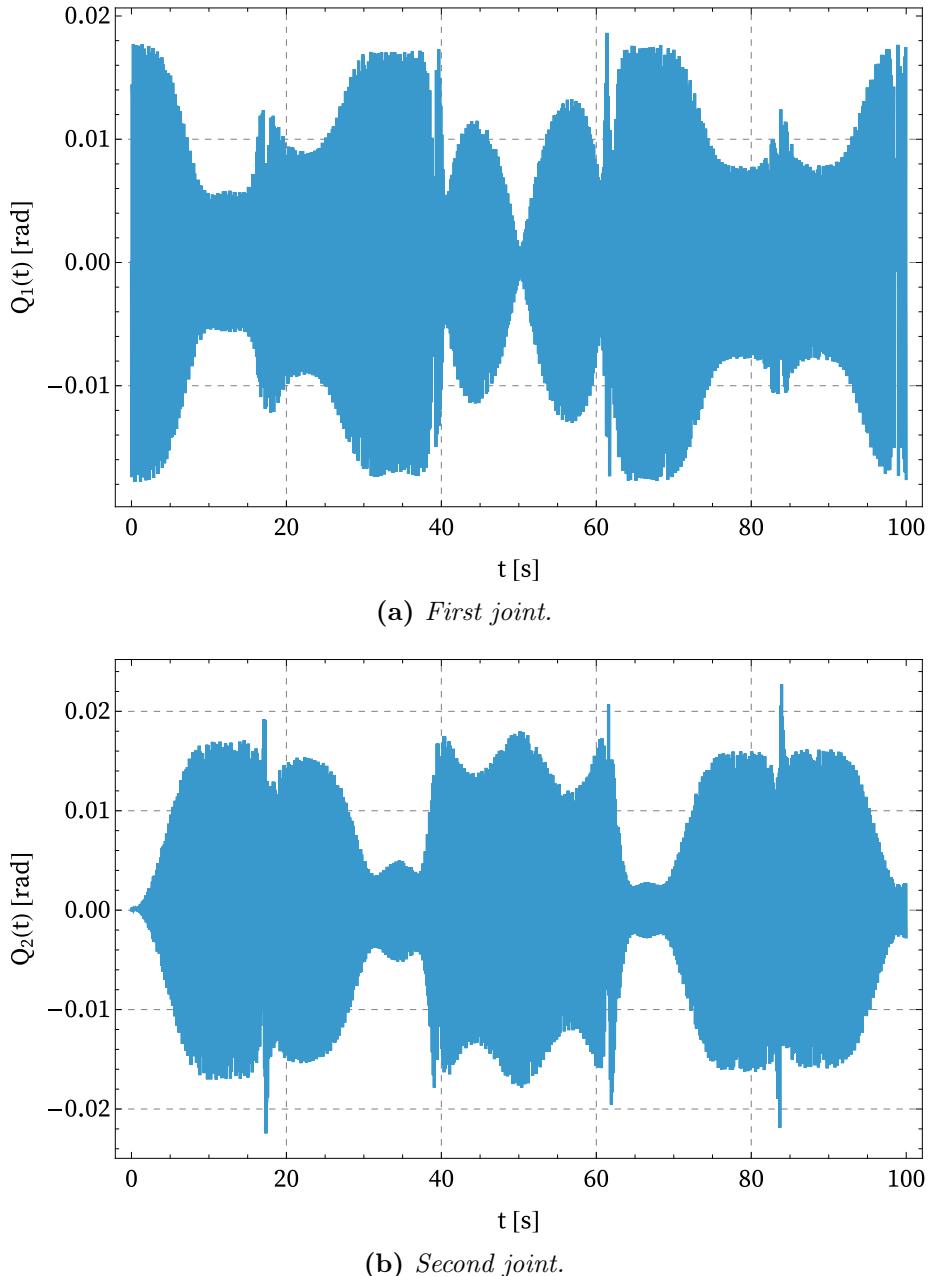


Figure 4.10: Extended temporal window of joints' elastic coordinates for Simulation 1. The oscillations change in time due to the movement of the arms.

4.3 Control Based Motion

As mentioned in [13], for flexible-link manipulators the control is not straightforward, because it involves the elastic degrees of freedom. Also, measurement of the angle between two links at a joint represent the sum of the angle between the two coordinate frames and the slopes of the elastic deformation at the joints.

There can be three approaches for the flexible system control:

- Elastic effects are ignored, i.e. the equation of motion is assumed to be as in (4.18). The torque computed by this model would give good results only if the elastic effects are negligible, otherwise it may result in gross inaccuracies in the positioning of the end-effector.
- The elastic effects are included in the mathematical model, and their amplitudes could be determined from the sensor readings by means of an estimator. This model treats the elastic deformations as known disturbances, but does not explicitly control them.
- Active control of both the rotational and bending co-ordinates, which is achieved by applying both torques and generalized forces. Such control is possible only if actuators capable of applying transverse forces to the links are present. Additionally, measurements of the elastic generalized coordinates and their rates must be available.

Since we have already analyzed the first (rigid) case and the third would require addiction sensors and further research in the "vibration control" field, the only feasible model is the second one.

Similarly to what has been done in Section 4.1.2 and equation (4.22), only the rotational coordinates q_1 and q_2 are actively controlled. This time, however, the distinction is between translational, rotational and elastic:

$$\begin{bmatrix} M'_{tt} & M'_{tr} & M'_{te} \\ M'_{rt} & M'_{rr} & M'_{re} \\ M'_{et} & M'_{er} & M'_{ee} \end{bmatrix} \begin{bmatrix} \ddot{p}_t \\ \ddot{p}_r \\ \ddot{p}_e \end{bmatrix} + \begin{bmatrix} C'_t \\ C'_r \\ C'_e \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ u \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} p_r \\ p_t \\ p_e \end{bmatrix} \quad (4.69)$$

For simplicity, we move the uncontrolled coordinates together:

$$\begin{bmatrix} M'_{rr} & M'_{rt} & M'_{re} \\ M'_{tr} & M'_{tt} & M'_{te} \\ M'_{er} & M'_{et} & M'_{ee} \end{bmatrix} \begin{bmatrix} \ddot{p}_r \\ \ddot{p}_t \\ \ddot{p}_e \end{bmatrix} + \begin{bmatrix} C'_r \\ C'_t \\ C'_e \end{bmatrix} = \begin{bmatrix} u \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (4.70)$$

with $M'_{jk} = M'_{kj}$ for the symmetry property of the mass matrix and its submatrices. Now, by setting:

$$\begin{bmatrix} p_t \\ p_e \end{bmatrix} = p_{nc}, \quad \bar{K} = \begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix} \quad (4.71)$$

where nc stands for "non-controlled", we get to the following equation:

$$\begin{bmatrix} M'_{rr} & M'_{r,nc} \\ M'_{nc,r} & M'_{nc,nc} \end{bmatrix} \begin{bmatrix} \ddot{p}_r \\ \ddot{p}_{nc} \end{bmatrix} + \begin{bmatrix} C'_r \\ C'_{nc} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \bar{K} \end{bmatrix} \begin{bmatrix} p_r \\ p_{nc} \end{bmatrix} = \begin{bmatrix} u \\ \mathbf{0} \end{bmatrix} \quad (4.72)$$

In this way we have arrived to a similar formulation as the one in (4.22). By writing the uncontrolled coordinate as a function of the rotational ones:

$$\ddot{p}_{nc} = -M'^{-1}_{nc}(M'_{nc,r}\ddot{p}_r + C'_{nc} + \bar{K}) \quad (4.73)$$

and pluggin it in the first row:

$$\bar{M}\ddot{p}_r + \bar{C} = u \quad (4.74)$$

with:

$$\bar{M} = M'_{rr} - M'_{nc,r}M'^{-1}_{nc,nc}M'_{nc,r}, \quad \bar{C} = C'_r - M'_{r,nc}M'^{-1}_{nc,nc}C'_{nc} - M'_{r,nc}M'^{-1}_{nc,nc}\bar{K}p_{nc} \quad (4.75)$$

Through a feedback linearization (4.30), we can control the manipulator's joints $p_r = \{q_1, q_2\}$ letting $p_{nc} = \{x_b, y_b, \theta_0, Q_1, Q_2\}$ free to move.

Figures 4.11 and 4.12 show the VMS behaviour after the impact. Now, since the arms do not move anymore after the system has reached the equilibrium position, the oscillations of the arms is constant in time (and do not damped out since no damping has been considered, see Figure 4.13).

Again, the rigid coordinates behave in a similar way as for the rigid case.

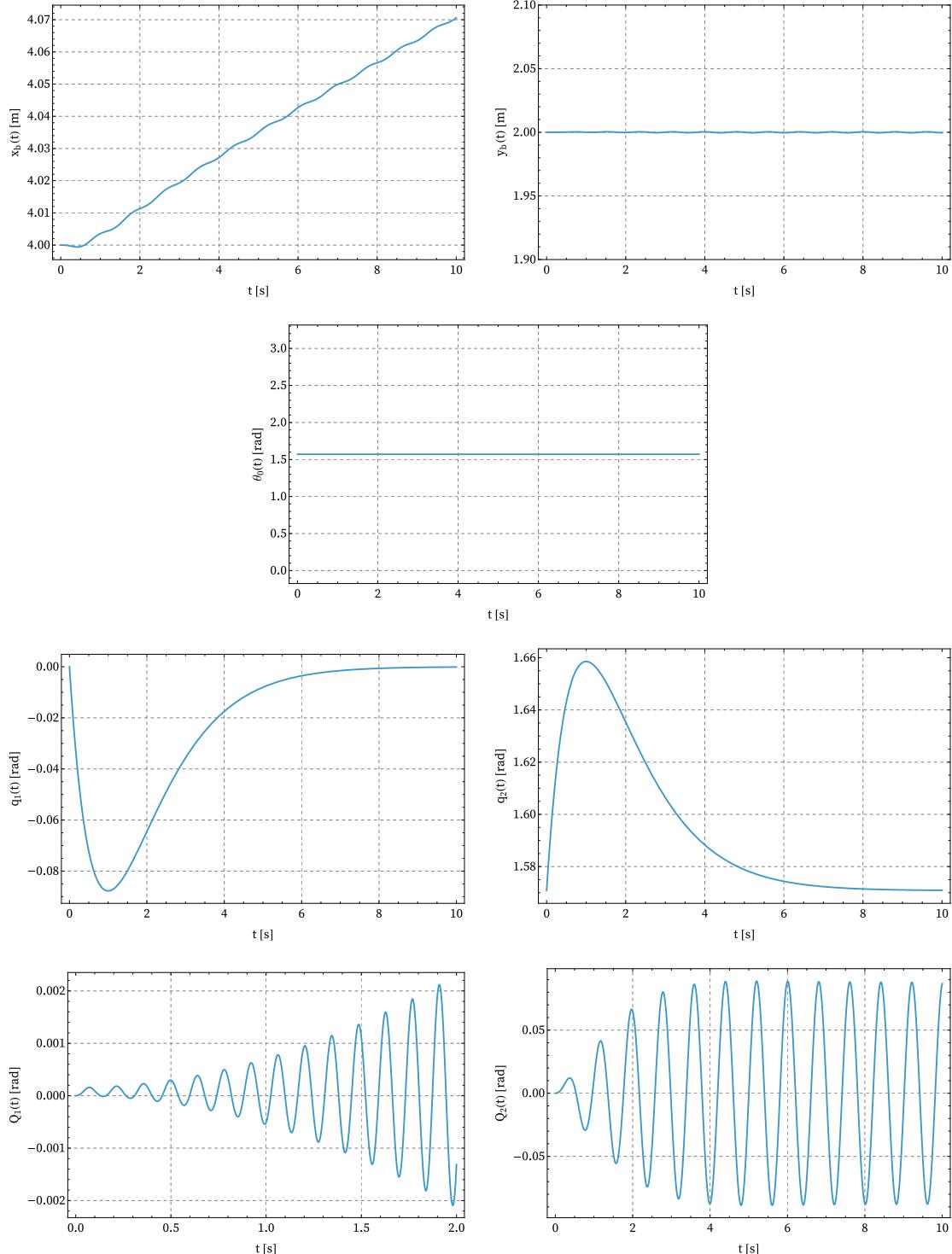


Figure 4.11: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, Simulation 1. Arms modeled as Euler-Bernoulli beams.

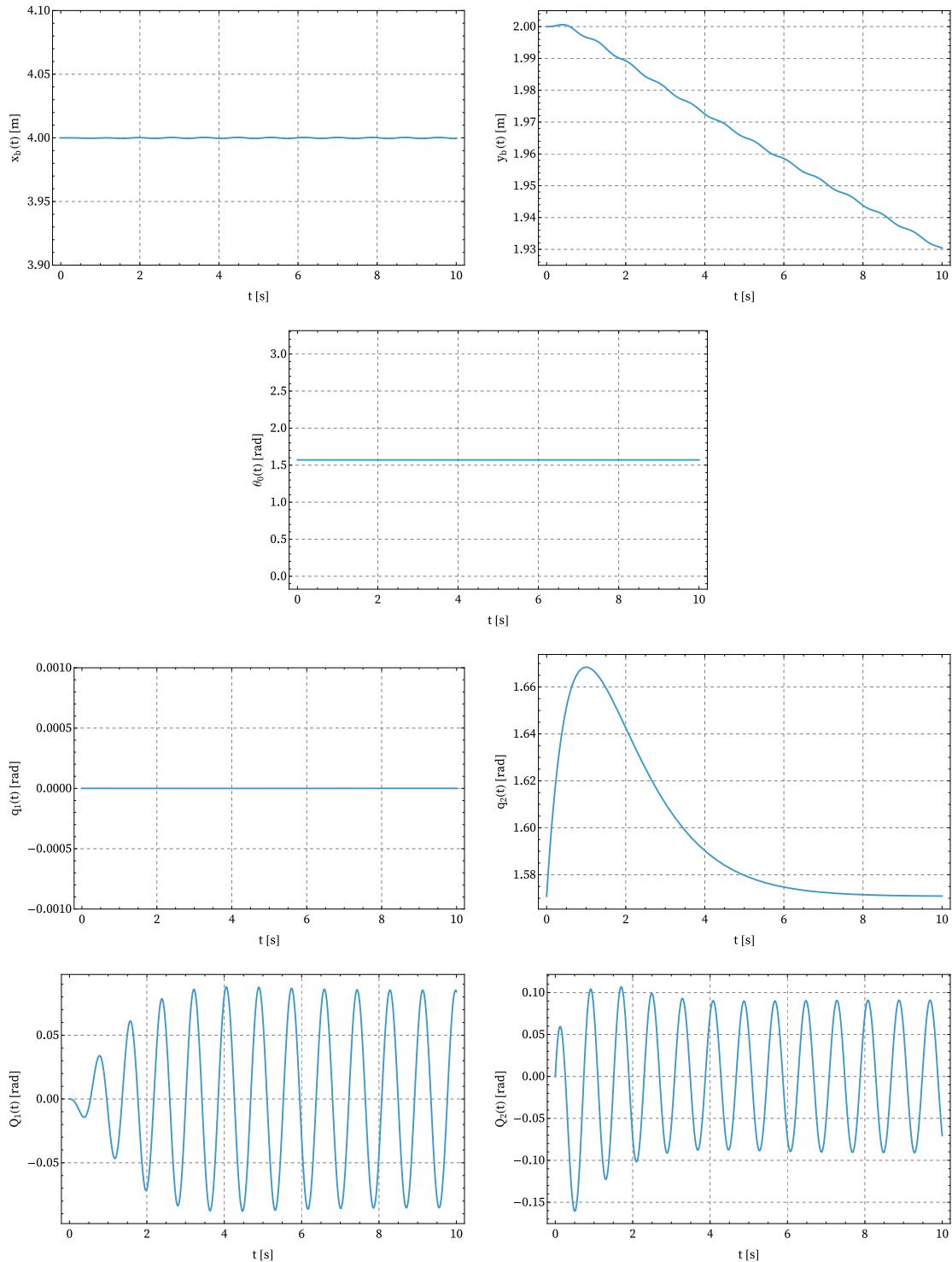


Figure 4.12: VMS generalized coordinates' displacement after the catching of the payload when no control is performed, Simulation 2. Arms modeled as Euler-Bernoulli beams.

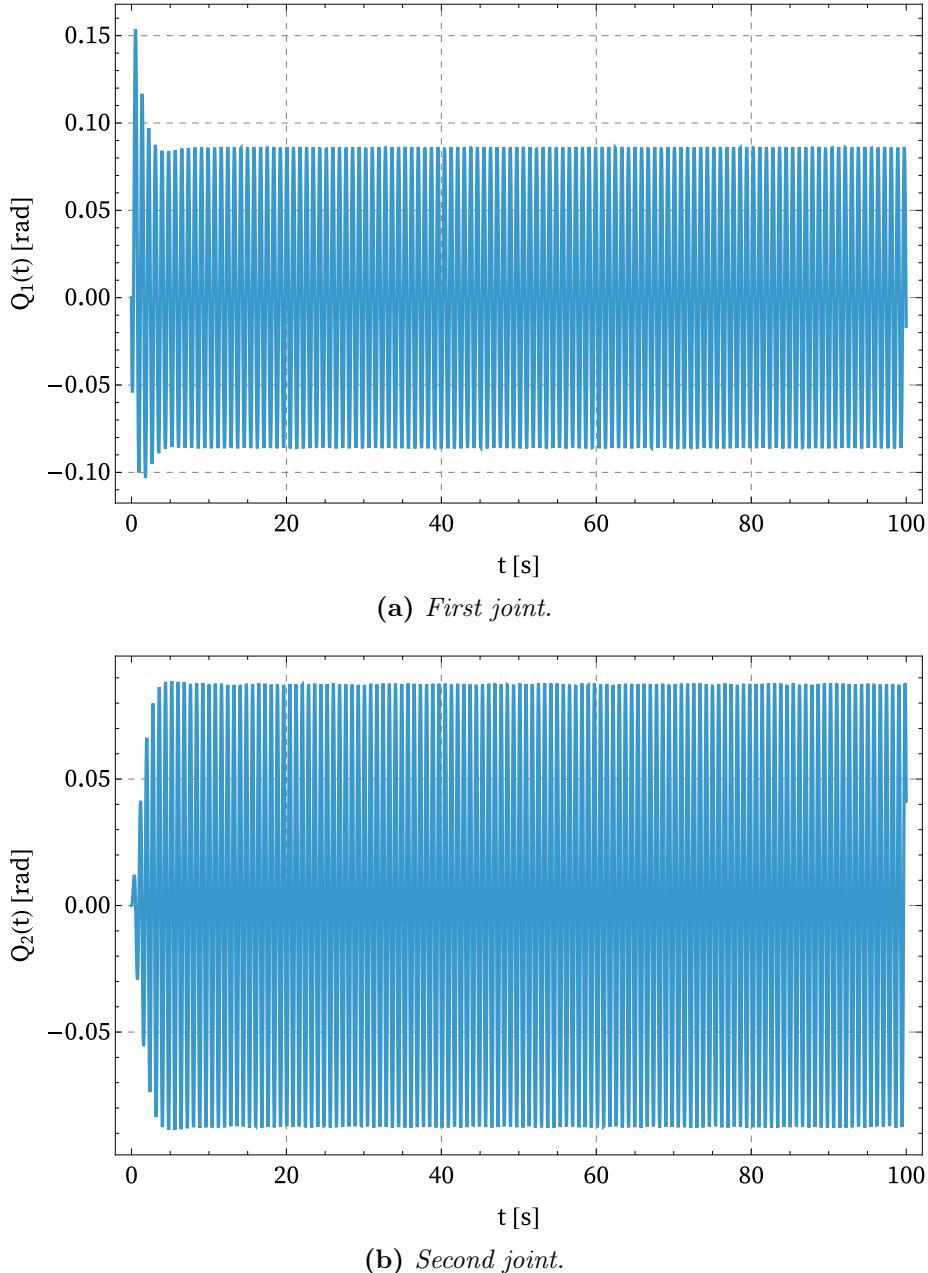


Figure 4.13: Extended temporal window of joints' elastic coordinates for Simulation 1. The oscillations are constant, since the arms do not move after the impact.

Chapter 5

Inertial and Kinetics Properties Retrieval

In the previous chapter, the captured object has been considered as a deterministic disturbance, i.e. the inertia and dynamic properties have been supposed known. However, in real case scenarios, the payload mass can be unknown (e.g. the captured of a small metoerite or of a satellite with estimated mass).

In this chapter, the same problem of payload docking will be analysed supposing unknown mass, trying to retrive it from the manipulator post-impact dynamics together with its initial velocities, which are not known a priori.

Only the rigid-bodies scenario will be taken into account, as the elastic-bodies one would require a lot of computational power and is very similar.

When the mass is not known, an estimate of the mass has to be injected in the control law (4.30). This value can be estimated by visual techniques or thanks to previus knowledge.

Before starting to calculate the real mass, some approximations has to be taken into account:

- the spacecraft does not move; this has already been shwon as a good approximation in the previous chapter;
- the spacecraft does not rotate; this has also been shown in the previuos calculations (see Figures 4.5, 4.6);
- given the little displacements after the impact, the dynamic can be linearized around the equilibrium position.

The VMS dynamics was described by the following equation:

$$M'\ddot{p} + C' = u \quad (5.1)$$

but under these assumptions, it can be rewritten as:

$$M'(q_1, q_2) \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + C'(q_1, q_2, \dot{q}_1, \dot{q}_2) = u(q_1, q_2, \dot{q}_1, \dot{q}_2) \quad (5.2)$$

Furthermore, the dynamics can be linearized around the equilibrium position (i.e. the initial one) through a Taylor expansion up to the first order [21]:

$$D(\mathbf{x}) = D(q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2) = M'(q_1, q_2) \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + C'(q_1, q_2, \dot{q}_1, \dot{q}_2) \quad (5.3)$$

$$D(\mathbf{x}) \approx D(\bar{\mathbf{x}}) + (\mathbf{x} - \bar{\mathbf{x}})^T \nabla T(\bar{\mathbf{x}}) \quad (5.4)$$

where $\bar{\mathbf{x}} = \{q_{1,0}, q_{2,0}, 0, 0, 0, 0\}$ is the equilibrium point and $\nabla T(\bar{\mathbf{x}})$ is the gradient of $T(\mathbf{x})$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$. The code used in Wolfram is available in the Appendix B. The control can now be found as in equation (4.30):

$$u = \hat{M}[\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)] + \hat{C} \quad (5.5)$$

with $q = \{q_1, q_2\}$, but now the mass inside the control is a guessed mass, different from the actual one and thus $\hat{M} \neq M'$, $\hat{C} \neq C'$, where the "hat" matrices are the ones where the real mass of the captures satellite has been substitute with the initial guess mass.

However, after linearization $C' = 0$ and the only error in the new equation is due to the mass matrix (along with the linearization error). The input becomes then:

$$u_{lin} = \hat{M}_{lin}[\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)] \quad (5.6)$$

and the final coupled dynamics:

$$M_{lin}\ddot{q} + C_{lin}\dot{q} + K_{lin}q - \delta = 0 \quad (5.7)$$

where C_{lin} and K_{lin} come from the linearized control input u_{lin} and δ from the desired final positions.

Different approaches have been investigated to find the actual mass and they will be hereafter investigate, from the worst to the best one.

5.1 Mass Extraction: Joints' Decoupling

The first method requires another strong assumption that will be validated a posteriori: the displacement of one arm in the controlled dynamics is negligible for the displacement of the other arm. From Figures 4.5, 4.6 it can be seen that also the arms displacement is little, so we can approximate them to zero when analyzing the other arm's dynamics.

This is true when $\hat{M} = M'$, for the input control has been design to automatically uncouple the equations, so when the error is small we can approximate the real mass matrix with the guessed one and decouple the equation. The nearer to the true mass the guessed one is, the more accurate will be its retrieval.

Conceptually, we can see the input control as follows:

$$u = -M'[K_d\dot{q} + K_p(q - q_d)] + C - dM[K_d\dot{q} + K_p(q - q_d)] + dC = u_e + du \quad (5.8)$$

where the effective control u_e is the one which decouples the equations and the error input du couple them again due to the error matrices dM and dC .

After linearization, however, we can make use of one fundamental property of linear systems, which is the superposition principle (i.e. a linear combination of inputs produces a linear combination of the outputs):

$$\begin{cases} M_{lin}\ddot{q} = u_{lin,e} \\ M_{lin}\ddot{q} = du_{lin} \end{cases} \quad (5.9)$$

The assumption made is then to neglect du_{lin} and consider only the effective one and

$$\lim_{\hat{m} \rightarrow m} du_{lin} = 0$$

With these suppositions, we can uncouple the dynamics of the two arms, since the accelerations and velocities of the all other coordinates are set to zero.

Figure 5.1 shows two time evolutions of the displacement of both arms with and without applying the approximation made above for the configuration assumed in the previous Chapter (i.e. $q_{1,0} = 0, q_{2,0} = \pi/2$). The error is really small.

We can now rewrite the controlled dynamics (4.31) of the two arms as follows:

$$\begin{aligned} \tilde{m}_1\ddot{q}_1 + \hat{m}_1 k_d \dot{e}_1 + \hat{m}_1 k_p e_1 &= 0 \\ \tilde{m}_2\ddot{q}_2 + \hat{m}_2 k_d \dot{e}_2 + \hat{m}_2 k_p e_2 &= 0 \end{aligned} \quad (5.10)$$

where $\tilde{m}(m, \hat{m})$.

Since the real mass is unknown, the approximation $M' = \hat{M}'$ is not possible anymore, that is why the supposed mass \hat{m} appears in the equation.

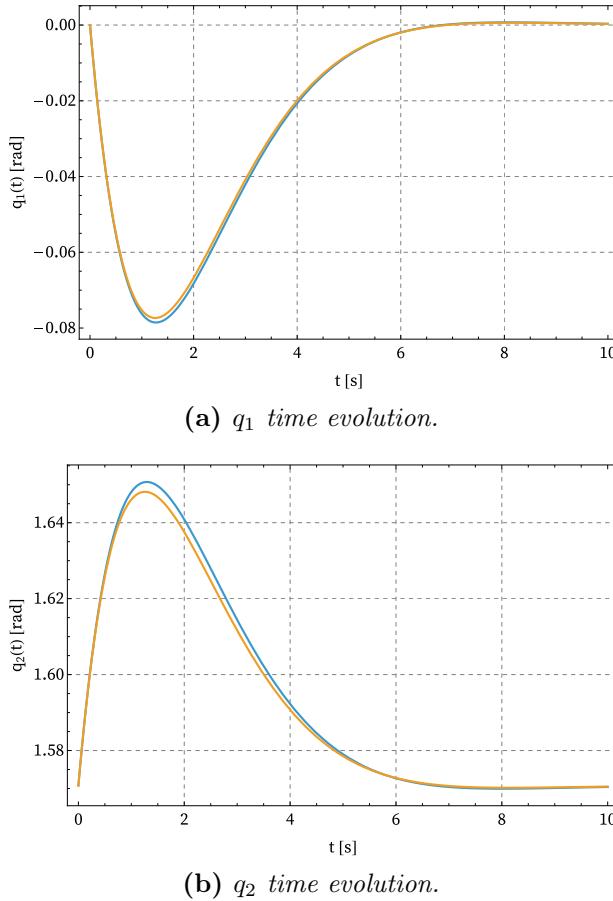


Figure 5.1: Arms behaviours with estimated mass 2000 kg, with (blue) and without (orange) approximations for the first simulation.

We can divide for m_1 and m_2 to get:

$$\begin{aligned}\ddot{q}_1 + \frac{\hat{m}_1}{\tilde{m}_1} k_d \dot{e}_1 + \frac{\hat{m}_1}{\tilde{m}_1} k_p e_1 &= 0 \\ \ddot{q}_2 + \frac{\hat{m}_2}{\tilde{m}_2} k_d \dot{e}_2 + \frac{\hat{m}_2}{\tilde{m}_2} k_p e_2 &= 0\end{aligned}\tag{5.11}$$

When the new control law with the estimated mass is applied, the behaviours is not critically damped anymore, since, as we can see from equations (5.11): the new damping coefficient is now given by:

$$\xi' = \frac{\hat{m}k_v}{2\omega_n \tilde{m}} = \frac{\hat{m}2\sqrt{k_p}}{2\sqrt{\frac{\hat{m}}{\tilde{m}}k_p}\tilde{m}} = \sqrt{\frac{\hat{m}}{\tilde{m}}}\tag{5.12}$$

5.1.1 Direct Solution

5.1.1.1 Position Roots

Given the simplicity of the equation of motions, a direct symbolic solution, containing \hat{m} as a parameter, can be easily found (i.e. $q(t, \hat{m})$).

Figure 5.2 shows how the time response of the first joint for the first simulation changes according to the estimated mass. It is of particular interest the increasing of the first peak as the estimated mass increases.

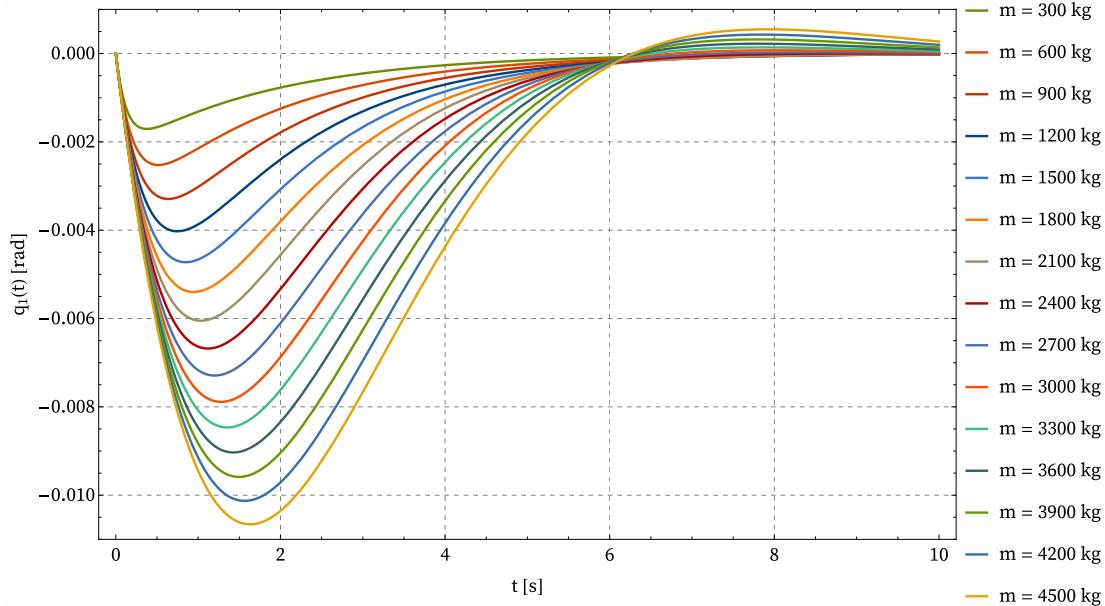


Figure 5.2: Parametric plot of the first joint's time evolution, first simulation. The real mass is 3000 kg. A visual parameter setting can be possible by confronting this plot with the real one (see Figure 4.5d).

Hence, once the peak time t^* is found, the real mass can be calculated by solving the following equation:

$$q(t^*, \tilde{m}) = q^* \quad (5.13)$$

with q^* the peak value.

5.1.1.2 Mass Fit

Furthermore, a minimization problem can be written to make the solution fit the available data. In the Wolfram software, the `NonLinearModelFit` command is available, allowing for nonlinear regression of a function by varying its parameters. In this case, the function to be optimized is the joint time evolution, and its parameter is the satellite mass.

An important aspect of this approach is that it allows to work with noisy data, since the regression tries to minimize the overall error, whereas noise in the previous

method seen above could disturb the overshoot's amplitude and time instant too much. Figure 5.3 shows the time evolution of the first joint in the first simulation with a white noise $w(t) \sim \mathcal{N}(0, 0.005)$.

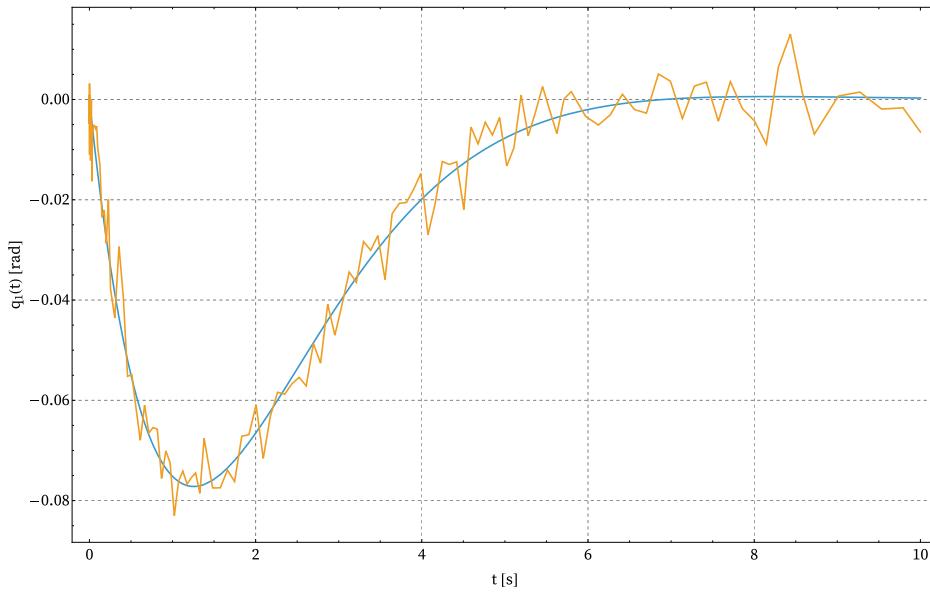


Figure 5.3: Noisy data to simulate a real scenario in orange.

5.1.2 Closed Form Solution

Now that we have linear second order differential equations (5.11), two methods have been introduced to retrieve the inertial properties of the payload (i.e. its mass). The first one, exposed in this section, is based on the fact that the solution of equations (5.11) is known.

5.1.2.1 Derivative Roots

Given zero initial velocities, from equation (5.11) one gets the generic equation for one joint:

$$\tilde{m}\ddot{q} + \hat{m}k_d\dot{q} + \hat{m}k_p(q - q_d) = 0 \quad (5.14)$$

The desired position is not zero in the general case. However, by moving the term on the RHS of the equation, we get a forced differential equation with constant external input:

$$\tilde{m}\ddot{q} + \hat{m}k_d\dot{q} + \hat{m}k_pq = \hat{m}k_pq_d \quad (5.15)$$

Equation (5.15) can be considered as a system subjected to a step function of amplitude $F_0 = \hat{m}k_pq_d$. The particular solution of such system is also known:

$$\begin{aligned} q_f(t) &= \frac{F_0}{k} \left[1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin(\omega_d t + \theta) \right] \\ &= q_d \left[1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin(\omega_d t + \theta) \right] \end{aligned} \quad (5.16)$$

with $\theta = \arccos \xi$ and $k = \hat{m}k_p$ elastic coefficient.

The total solution will be given by the sum of the homogeneous and the particular solution, which are both known:

$$q(t) = q_d \left[1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin(\omega_d t + \theta) \right] + \left[q_0 \cos(\omega_d)t + \frac{\dot{q}_0 + \xi\omega_n q_0}{\omega_d} \sin(\omega_d t) \right] e^{-\xi\omega_n t} \quad (5.17)$$

where q_0, \dot{q}_0 are the initial position and velocity of the joint and $\omega_d = \sqrt{1 - \xi^2}\omega_n$. Notice that \dot{q}_0 is now given by the velocity of the joint after the impact, which can be retrieved as the derivative of the data at the time $t = 0$.

It is easier to find a relationship between the real mass and the peak time t^* by differentiating equation (5.17), so that the RHS term becomes zero, since the peak is a stationary point:

$$\sqrt{1 - \xi^2} \cos(\sqrt{1 - \xi^2}\omega_n t^*) - \xi \sin(\sqrt{1 - \xi^2}\omega_n t^*) = 0 \quad (5.18)$$

which can be rearranged:

$$\begin{aligned}\tan(\sqrt{1-\xi^2}\omega_n t^*) &= \frac{\sqrt{1-\xi^2}}{\xi} \\ \Rightarrow \tan(\sqrt{1-\xi^2}\omega_n t^*) &= \beta(\xi)\end{aligned}\quad (5.19)$$

The intersection of the two functions allows to calculate the damping coefficient of the equation and thus the real mass from relation (5.12).

Note that equation (5.18) does not depend on the equilibrium position, hence it holds for the general case.

5.1.2.2 Damping Coefficient and Natural Frequency Fit

Similarly at what has been done for the mass fit, the same built-in command `NonLinearModelFit` is used inserting as interpolating function equation (5.17), with parameters ξ and ω_n , which are both equal to $\sqrt{\frac{\hat{m}}{m}}$ (see Section 5.1).

5.2 Mass Extraction: Coupled Solution

The problem with the joints' decoupling assumption is that it depends on the joints' configuration, as it will be better exposed in the next Chapter.

Without decoupling them it would be difficult to find a closed form solution and, given the computational power available, different approaches can be evaluated.

In this section two methods will be explored by solving the coupled linear equation directly and performing a minimization of a cost function.

5.2.1 Continuous Domain

Wolfram has a built-in command, `ParametricNDSolve`, which allows to solve numerically differential equations with unknown values (e.g. the actual mass m).

The output is a parametric function $q(m, t)$ that, given a mass value, returns a continuous function in time.

A cost function Γ can be written as the integral along the simulation period T of the squared residuals between the found solution and the actual evolution of the joint in time (given from the satellite captured):

$$\Gamma(m) = \int_0^T [q(m, t) - q_{data}(t)]^2 dt \quad (5.20)$$

By setting the gradient of Γ to zero we can find the mass that minimize the cost function, i.e. the true mass:

$$m^* = \min_m \Gamma(m) \quad (5.21)$$

Even if this method gives better and more consistent result than the previous ones (as it will be shown in the next Chapter), the computational power required to solve a minimization of an integral could be too demanding. That is why a second more efficient approach has been explored. The actual code used in Wolfram is available in Appendix B.

5.2.2 Discrete Domain

The same problem can be described by discretizing the available data and the differential equations solutions in a finite number of points in time.

However, to do so, a symbolic parametric solution of the coupled equations needs to be available and Wolfram does not manage to solve them.

For this section another software has been used to solve directly the equations, Maple 2024.2, whose commands are reported in Appendix B as well.

Maple manages to solve the joints' motion as a function of time and mass (the results are listed in Appendix B.1, B.2); it is now possible to discretize the solution:

$$\Gamma(m) = \sum_{k=0}^{N_p} \left[q(m, \frac{T}{N_p} k) - q_{data}(\frac{T}{N_p} k) \right]^2 \quad (5.22)$$

where N_p is the number of discretization points.

In the same way as for the continuous case, the gradient of the cost function is set to zero and the optimal mass can be calculated.

The number of discretization points should not be too low, in order not to have a huge discretization error, nor too high to avoid great computational resources.

For this thesis, it has been found that $N_p = 5$ gives better results than the Joints' Decoupling method. Conceptually it is very similar to the Position Root one, with the difference that no input error du is introduced and the function is evaluated on more than one point.

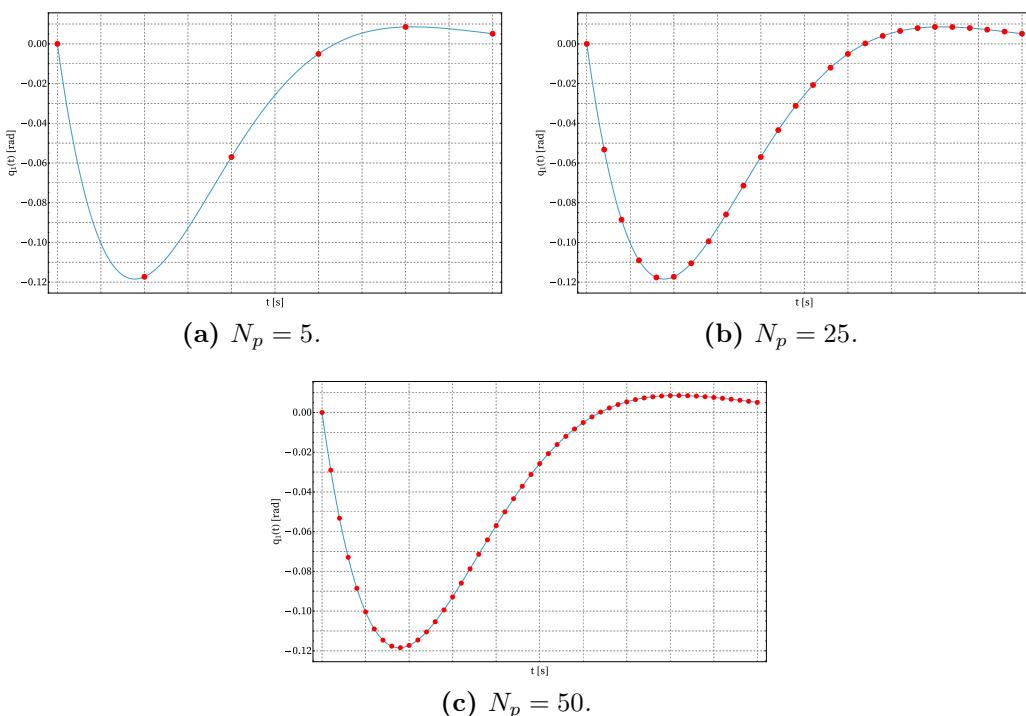


Figure 5.4: In blue the continuous solution, in red the discretized one.

5.3 Velocity Extraction

Once the mass is found, it would be interesting asking what was the satellite velocity before the impact.

Equation (4.10) set an important relation between final and initial velocities of the VMS and the satellite as well. In particular, the initial velocity of the VMS are known (since it is supposed to be still) and the velocities after the impact of the VMS (i.e. \dot{p}_f) can be found by derivating p_{data} in $t = 0$.

We can therefore isolate $\dot{\psi}_i$:

$$\begin{aligned} H &= G\dot{p}_f \\ \Rightarrow M\dot{p}_i + J^T(J_O^+)^T M_O \dot{\psi}_i &= G\dot{p}_f \\ \Rightarrow \dot{\psi}_i &= M_O^{-1} J_O^T (J^+)^T (G\dot{p}_f - M\dot{p}_i) \end{aligned} \quad (5.23)$$

but since the VMS is at rest before the impact:

$$\dot{\psi}_i = M_O^{-1} J_O^T (J^+)^T G\dot{p}_f \quad (5.24)$$

On Wolfram, solving this system of equations gives very good results regarding the translation velocity, while the angular velocity error is quite big due to numerical approximations.

In this thesis, however, the assumption that has been made is that the relative velocity between the satellite and the VMS is small, so by setting $\dot{\theta}_O = 0$ the output is consistent (notice that the real velocity in the simulation was set $\dot{\theta}_O = 0.01$, see Table 4.3).

Chapter 6

Results and Conclusion

In this Chapter a comparison between the methods suggested in the previous sections will be displayed as well as plots with the results found for each approach. The first section will focus on the mass outputs and considerations about the validity and efficiency of each method; the second section will focus on the error made in finding the initial velocity given different guessed masses; in the third one final considerations of this work and possible future improvements will be exposed.

6.1 Mass Retrieval

The joints' decoupling method, as already mentioned, relies on three assumptions:

1. Little displacement from initial position (linearization error);
2. When the guessed mass is near to the true one, we can decouple the differential equations (approximation error);
3. The guessed mass is not the real one (mass error).

The first two errors depend on the VMS arms' configuration, i.e. on the direction of impact of the satellite and its inclination with respect to the manipulator.

In Chapter 4 only one configuration and two different directions of impact (respectively called Simulation 1 and Simulation 2) have been analyzed. Here, different arms' configurations are studied.

However, M_{lin} , C_{lin} and K_{lin} do not depend on $q_{1,0}$, but on $q_{2,0}$ and \hat{m} only: we can study different scenarios by changing the second joint's initial position and the guess mass only.

Figures 6.2 and 6.3 show the behaviour of the first joint for the study case Simulation 1 (i.e. the satellite approaches the manipulator with positive linear velocity in the x direction and null velocity in the y direction) for different values of $q_{2,0}$ with $\hat{m} = 2000 \text{ kg}$ (the true satellite's mass is $m = 3000 \text{ kg}$, as shown in Table 4.1). The plots for the second joint follow the same behaviour and are listed in Appendix C. As it can be seen, when the displacement of one arm is little (e.g. when $q_{2,0} = \pi/2 \text{ rad}$ the second arm is almost still for the first simulation) the linear error is almost zero, while when there is a greater displacement (e.g. $q_{2,0} = 0 \text{ rad}$) it is more relevant. Similarly, the more the behaviour of one arm affects the movement of the other, the greater the approximation error is.

However, the linear error can be reduced by enabling a stronger control input, for example by increasing the proportional gain K_p , which assures faster convergence to the desired value but could enhance overshoots (if not critically damped). On the other hand, the approximation error relies on the geometry of the impact and it is more difficult to control.

Figures 6.4, 6.5, 6.6, 6.7 and 6.8 show the mass found with the joints' decoupling methods for different guessed masses \hat{m} for the first simulation. Since for each joint we can evaluate one mass (i.e. two retrieved masses), its final value was obtained as the mean of the two. However, there are cases where the joint does not move and it is not possible to find the real mass: in those cases only the final mass found with the other joint is used (e.g. for the first simulation and $q_{2,0} = \pi/2 \text{ rad}$, the second joint, which lies on the satellite velocity direction does not move or very little).

Furthermore, there could be cases where no arm moves: in the singular configuration

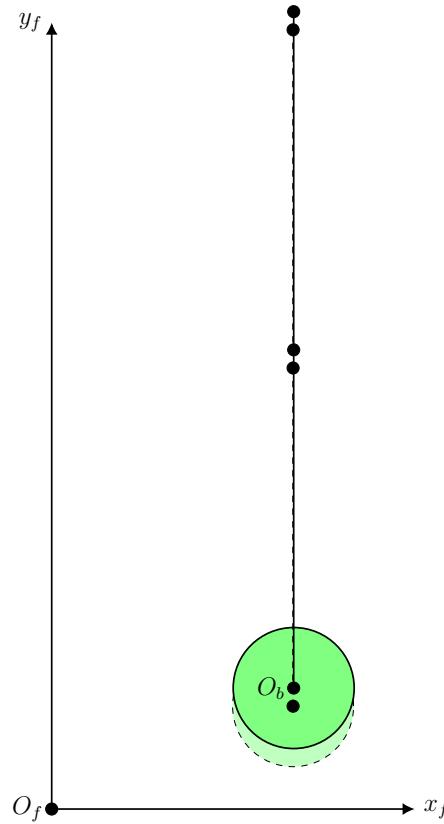


Figure 6.1: When the manipulator is in its singular configuration, no displacement at the joints happens if the payload approaches with a north-to-south direction.

($q_{1,0} = 0 \text{ rad}$ and $q_{2,0} = 0 \text{ rad}$), for the second simulation (i.e. when the satellite approaches with zero velocity in the x direction) both arms remain still and only translation of the VMS occurs (see Figure 4.6). This is why in the Appendix C, where the same Figures are shown for the second simulation, there is no plot of the configuration $q_{2,0} = 0 \text{ rad}$.

With the uncoupled method, only the mass and linearization error hold, since no approximation has been done. As a consequence, the result are much more reliable and consistent, as Figures 6.9 and 6.10 show for both simulation and all considered configurations.

As already mentioned, for the second simulation no mass can be retrieved when the manipulator's configuration is singular.

The method shown here is the continuous one only, since the discrete one gives similar results, as shown in Tables 6.1 and 6.2.

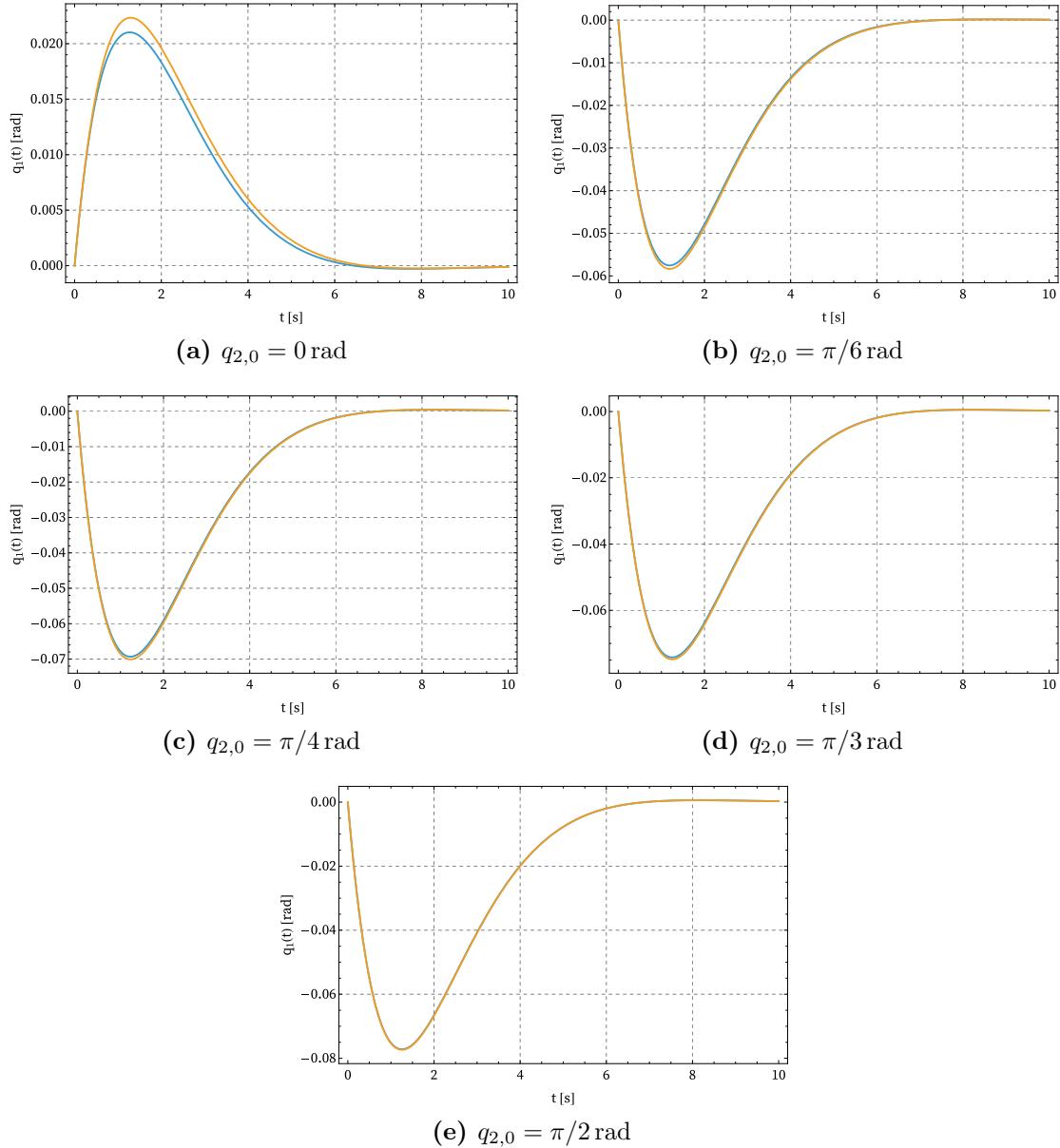


Figure 6.2: Linearization error for different first arm's configuration, Simulation 1: non-linear solution in blue, linearized solution in orange, with $\hat{m} = 2000 \text{ kg}$.

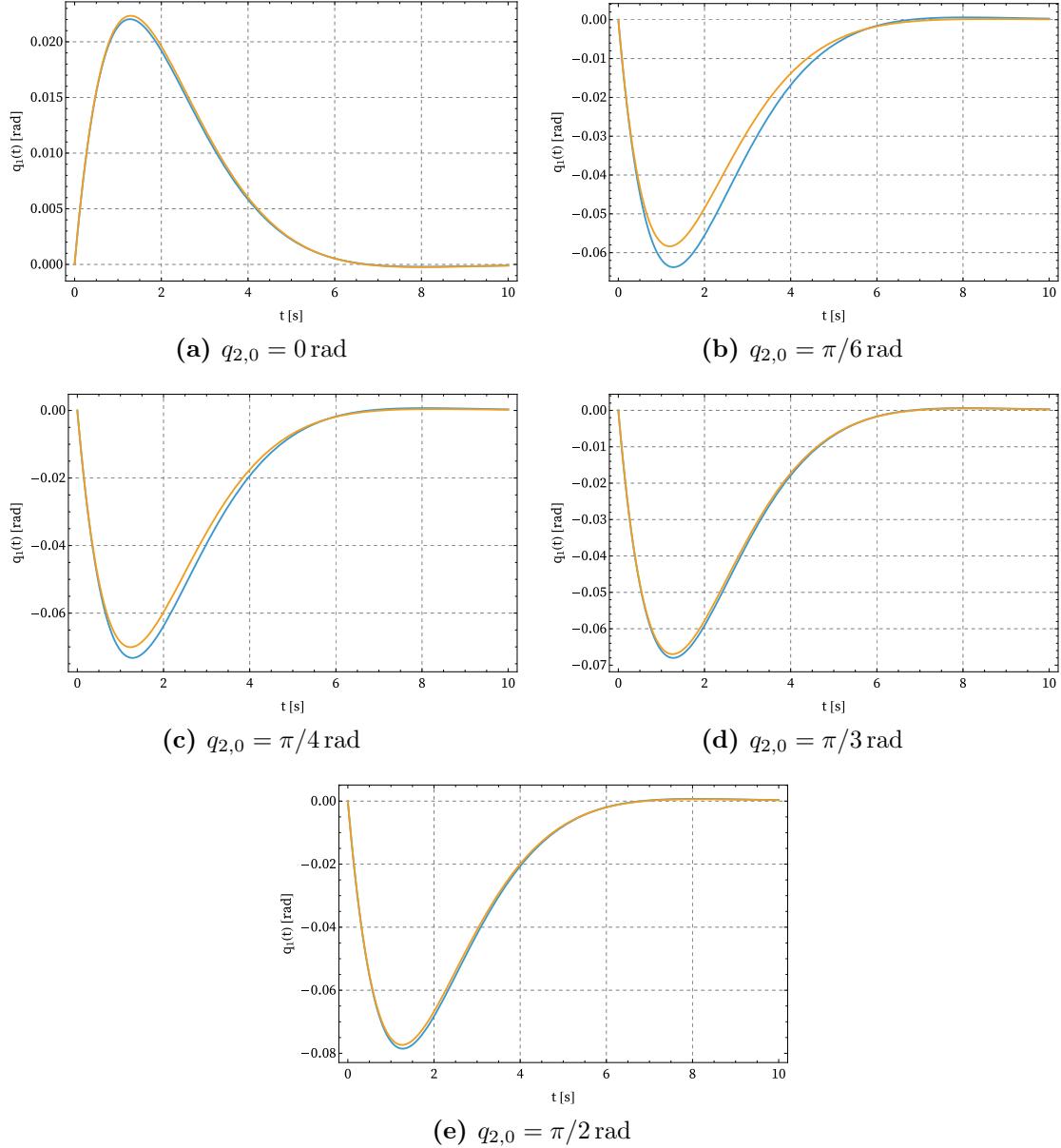


Figure 6.3: Approximation error for different first arm's configuration, Simulation 1: non-approximated linearized solution in orange, approximated solution in blue, with $\hat{m} = 2000 \text{ kg}$.

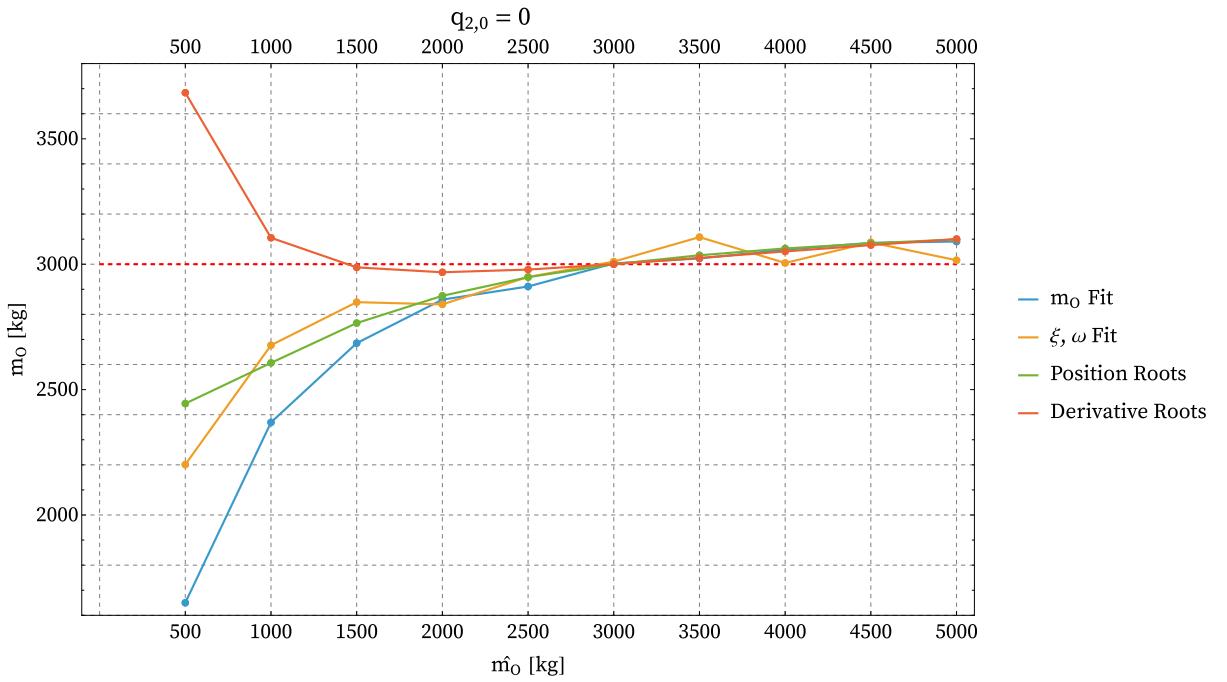


Figure 6.4

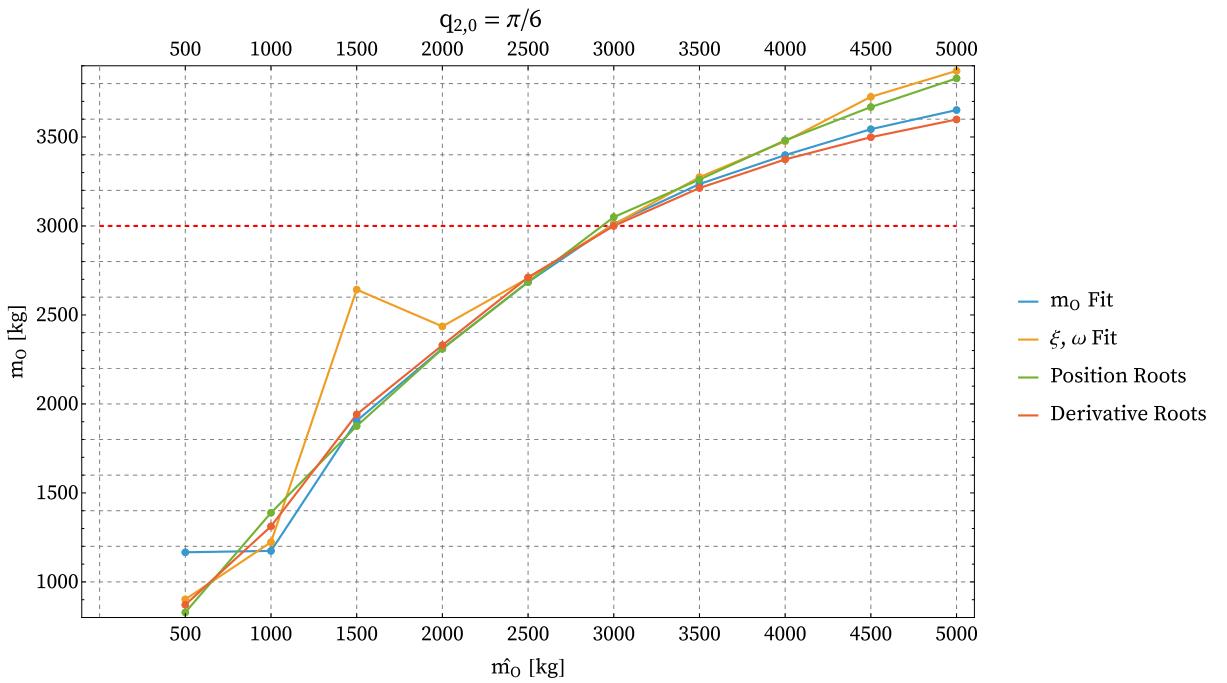


Figure 6.5

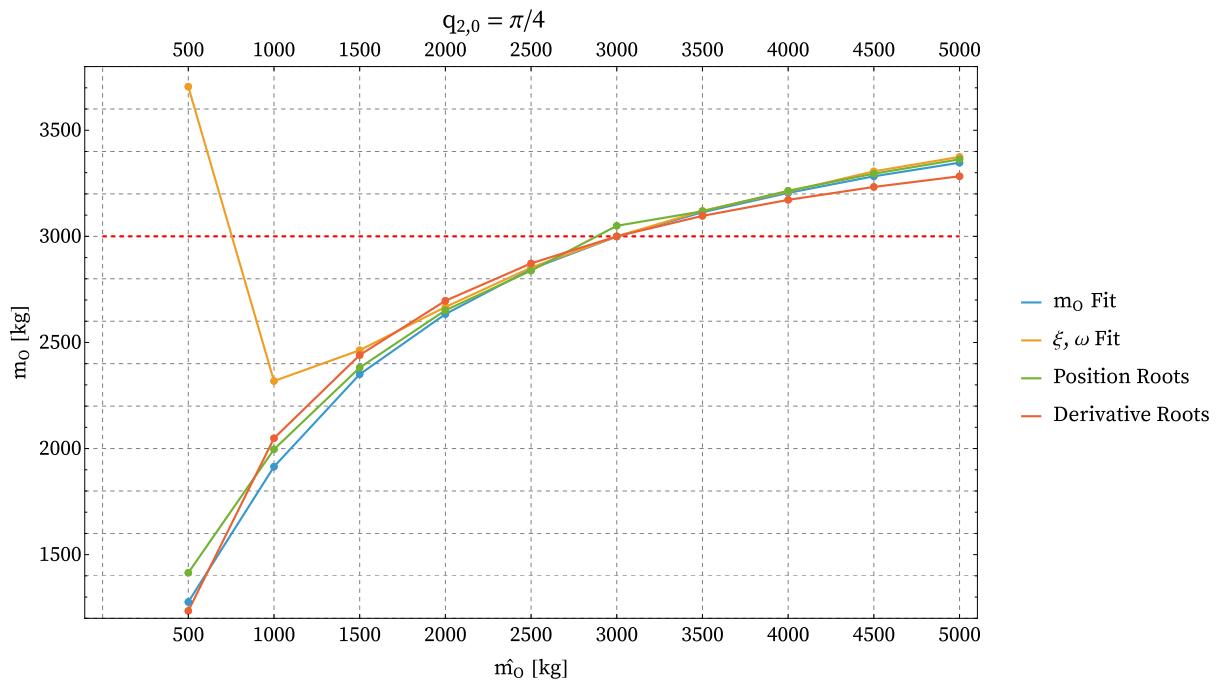


Figure 6.6

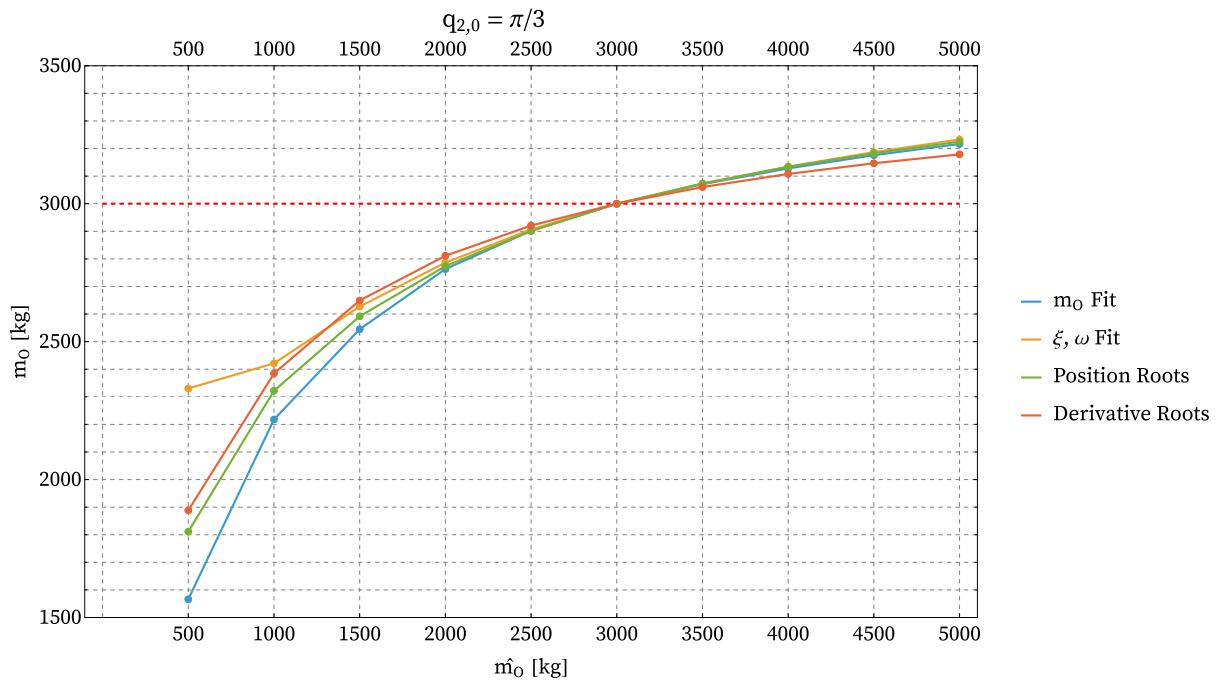


Figure 6.7

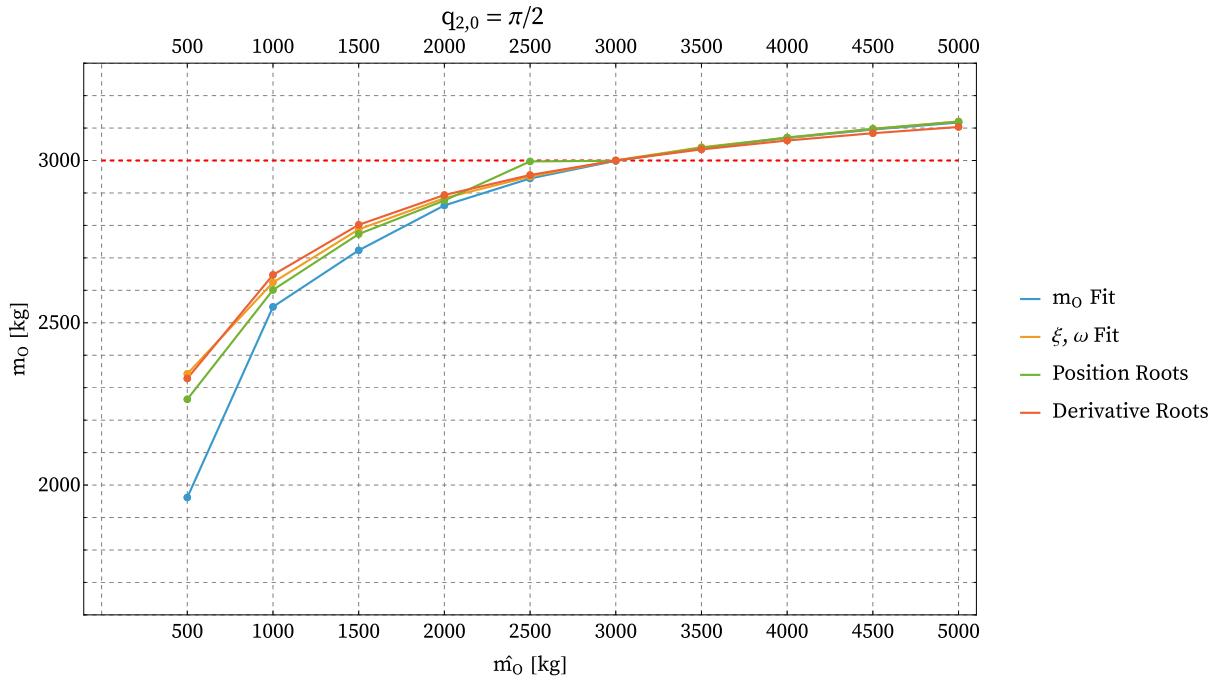


Figure 6.8

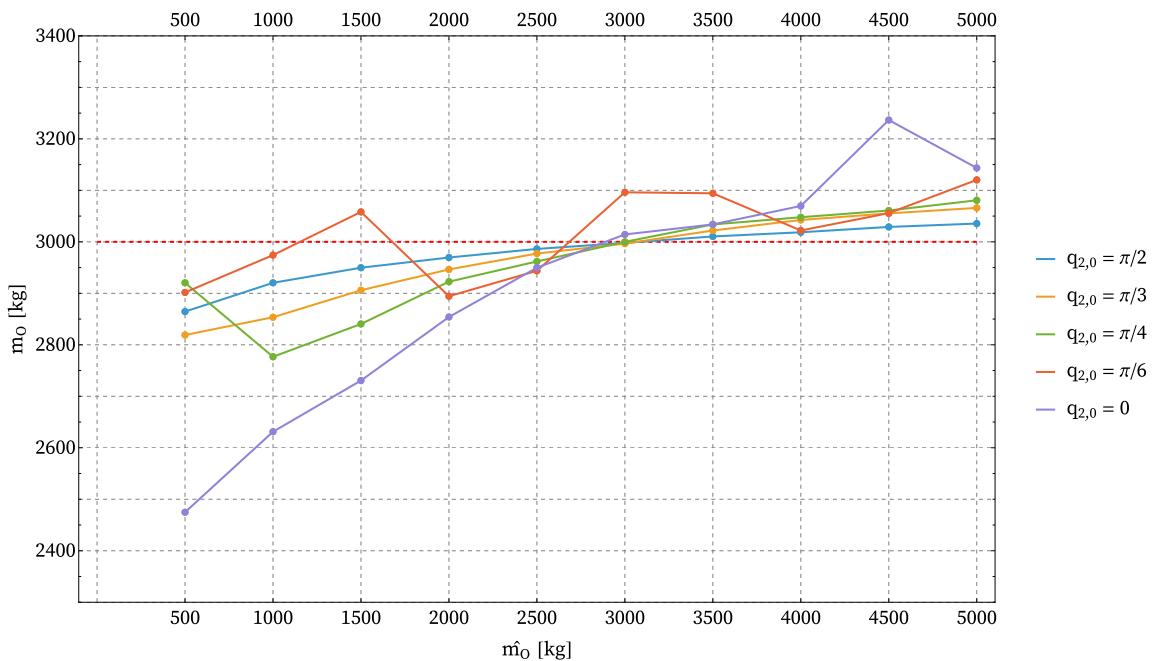


Figure 6.9

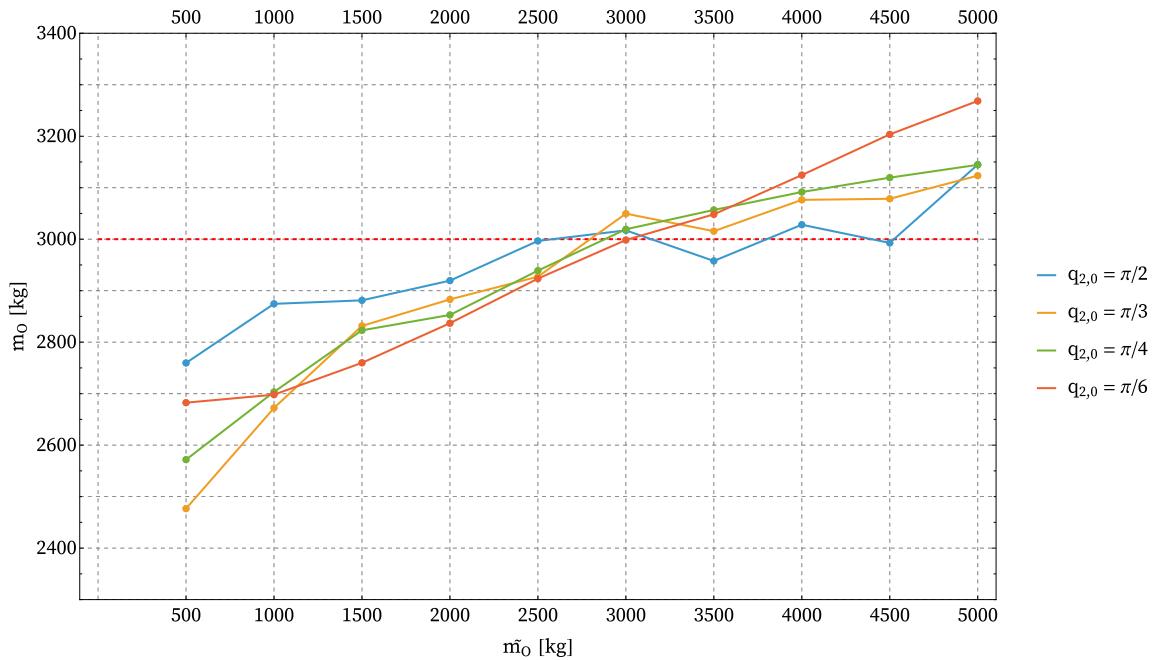


Figure 6.10

Table 6.1: Uncoupled methods comparison for Simulation 1 and $q_{2,0} = \pi/2$ rad.

Guess Mass	Discrete Method	Continuous Method
500 kg	2866.06 kg	2864.48 kg
1000 kg	2920.21 kg	2920.38 kg
1500 kg	2949.45 kg	2949.72 kg
2000 kg	2970.11 kg	2969.52 kg
2500 kg	2986.09 kg	2986.22 kg
3000 kg	2999.16 kg	2997.87 kg
3500 kg	3010.29 kg	3010.44 kg
4000 kg	3020.04 kg	3018.57 kg
4500 kg	3028.79 kg	3028.93 kg
5000 kg	3036.77 kg	3035.47 kg

Table 6.2: Uncoupled methods comparison for Simulation 2 and $q_{2,0} = \pi/2$ rad.

Guess Mass	Discrete Method	Continuous Method
500 kg	2770.30 kg	2759.61 kg
1000 kg	2862.58 kg	2874.43 kg
1500 kg	2913.02 kg	2881.21 kg
2000 kg	2949.30 kg	2918.53 kg
2500 kg	2977.09 kg	2996.60 kg
3000 kg	2999.18 kg	3017.16 kg
3500 kg	3017.32 kg	2957.75 kg
4000 kg	3032.64 kg	3028.08 kg
4500 kg	3045.89 kg	2992.76 kg
5000 kg	3057.56 kg	3144.85 kg

6.2 Velocity Retrieval

The velocity retrieval error has been calculated in two different ways, since for both simulations the satellite initial velocity is zero in one direction:

- As a relative error for the non-zero initial velocity:

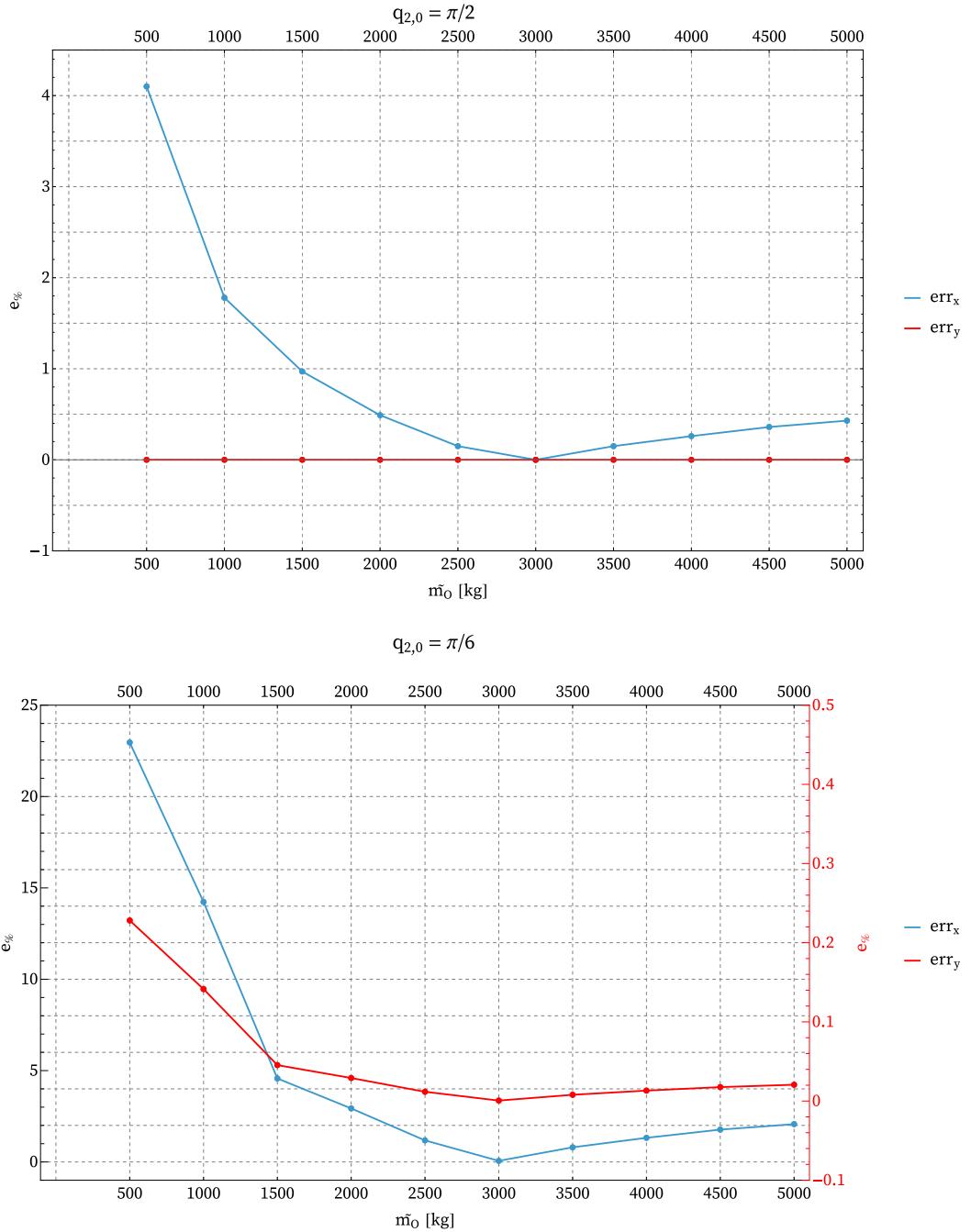
$$e\% = \frac{|\dot{\psi}_{i,t} - \dot{\psi}_{i,r}|}{\dot{\psi}_{i,t}} * 100 \quad (6.1)$$

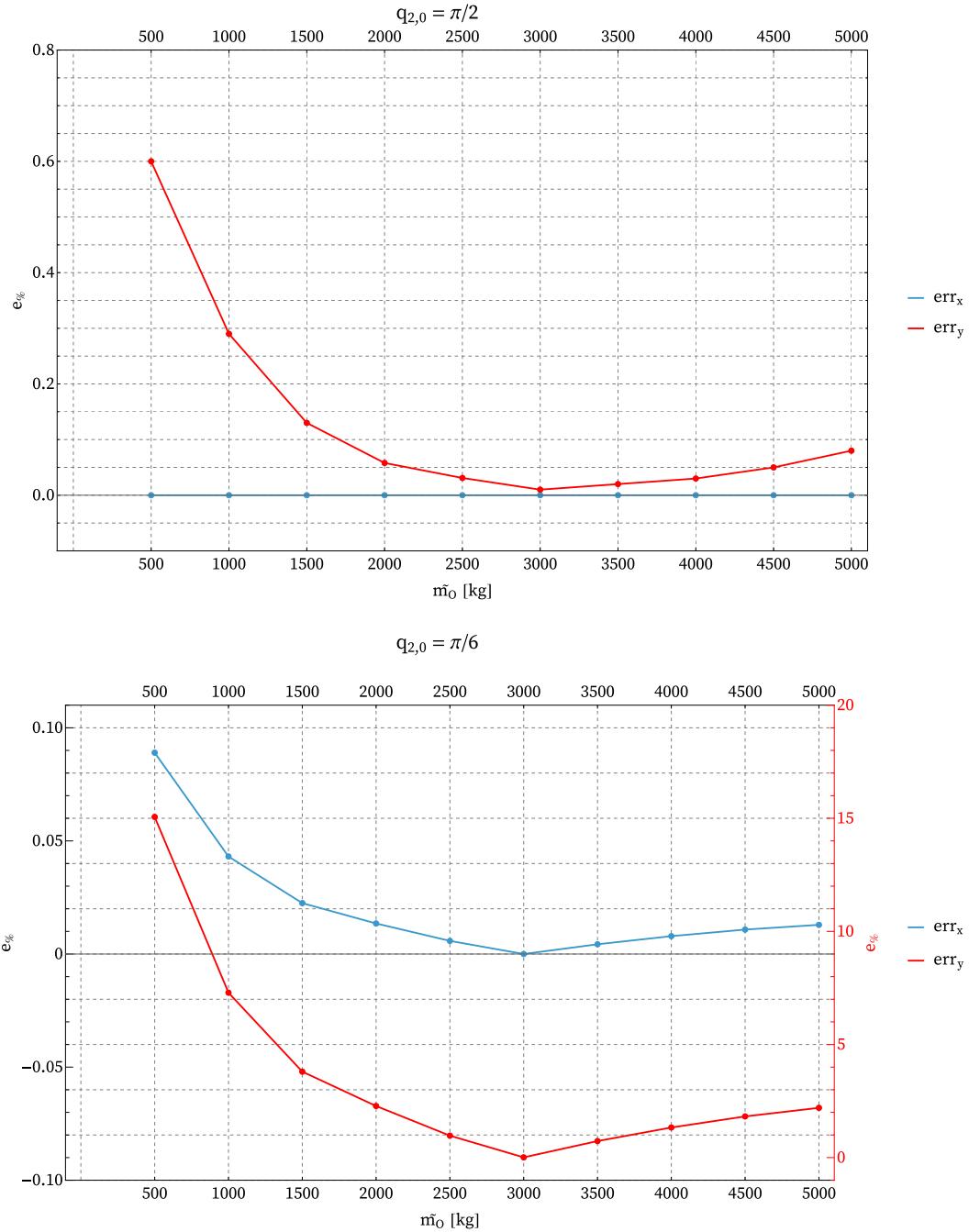
where the subscripts t and r stand for "true" and "retrieved" respectively;

- As an absolute error fro the zero initial velocity (as it is not possible to divide for zero):

$$e\% = |\dot{\psi}_{i,t} - \dot{\psi}_{i,r}| * 100 \quad (6.2)$$

Figures 6.11 and 6.12 show the error for two configurations only, which are the best and worst working scenario according to the linearization and approximation error (see Figures 6.2 and 6.3).

**Figure 6.11:** Velocity error for Simulation 1.

**Figure 6.12:** Velocity error for Simulation 2.

6.3 Results' Analysis

Different methods have been explored, each one with its strengths and weaknesses. Here, some considerations on the aforeshown results are done:

- The joints' decoupling approach allow to obtain closed form equation and simple time-dependent solution and can be used with different methods, such as the fitting ones or the position/derivative roots ones.

On the other hand they depends significantly on the VMS configuration and the more the configuration allows displacements, the less near the new mass is from the actual one.

- Among the decoupling methods, the damping coefficient and natural frequency fit seems to be the best one, as it gives the higher values when $\hat{m} < m$.

When $\hat{m} > m$, all the methods give better results: since the initial guessed mass is higher, the control is more aggressive and the displacement after the impact is smaller, riducing the linearization error.

- The uncoupled methods gives far more interesting results, as the final mass error is usually very small. The same consideration regarding the error linearization made above are valid, hence a higher initial mass is generally more preferable than a smaller one (i.e. an overestimation is preferable over an underestimation). Furthermore, these methods are very little sensitive to the VMS configuration, allowing for more freedom in the satellite docking.

- The continuos method and the discrete one give similar result for a sufficiently high number of discretization point, even though we have already seen that less than ten points give already a better result than the other methods. Furthermore, the discrete method is faster: in Wolfram, on a MacBook Air M3 with 16 GB of RAM, the mean execution time of the continuous minimization of the cost function take approximately 1.5 s, while the minimization of the discrete cost function only 0.03 s, which is ≈ 50 times faster.

Hence, a discrete method is more suitable in environment with no huge computational power, as it could be in space.

- The velocity error is lower for mass overestimation too and it is low for not too smaller initial guess as well.

6.4 Final Considerations

In this thesis, a planar VMS has been modeled, starting with the kinematics, then the dynamics through both the classic lagrangian approach and the homogeneous matrices method.

The impact analysis of a satellite docking has been explored introducing assumptions on the joints' displacement during the impact and contact-point forces/moment. After having found a relation between the initial satellite's velocities and final VMS' velocities, two scenarios have been studied: a rigid bodies ones, where the manipulator's arms were considered as perfectly rigid, and an elastic and more realistic one, where the arms' were approximated as Euler-Bernoulli Beams. This introduces the stiffness matrix in the equations of motion and more degrees of freedom related to the elastic vibration. The number of DoFs depends on the number of modes considered.

For both scenarios, both free and control-based motion simulations have been performed, considering as initial conditions a still VMS and a slow relative velocity between the VMS and the payload, with almost null angular velocity (since in the real case a null relative velocity is preferable, even though that is never the case and a non-zero velocity is always present.)

Regarding the control input, a feedback linearization was chosen along with a feed-forward term to account for the mass and Coriolis matrices. Furthermore, a critically damped behavior has been selected to have a smooth convergence.

Only the rigid rotational coordinates have been controlled, since the base's mass is far greater than the manipulator's arms and the satellite combined, so it can be considered as still, as the motion plots show. Moreover, the elastic coordinates for the elastic bodies scenario has been considered as uncontrollable noise and let free to evolve.

Finally, in the last chapters the problem we tried to solve the problem of unknown satellite's mass and initial velocities. Different methods have been proposed, which can be gathered in two main approaches: an approximated one, where linearization, mass and approximation error have been considered and closed form solution could be found, thanks to the simplicity of the final equations; a coupled one, where only linearization and mass error were considered and more reliable solution have been found, with different strengths and weaknesses.

Regarding the velocity retrieval, the theoretical relation between final and initial velocities that comes from the impact theory has been used.

The VMS' configuration plays an important role in these calculations, since the linearization error is as much smaller as the displacement after the impact is small, which may be not the case for some configurations where the initial displacement is bigger than others'. This can be controlled by regulation of the control proportional

and derivative parameters.

However, for singular configurations the joints do not move, and no method aforesown can be used to retrive the mass. Those configuration should be avoided in general, since the system loses a degree of freedom, as already seen in Section 2.1.4. The final results suggest that an overestimate of the satellite's mass is usually preferable over an underestimation and that discrete methods are to be chosen when lack of computation power is present.

6.4.1 Further Developments

There is still a lot of room for improvements and deeper analysis, such as:

- The chosen geometry is a planar (2D) RR manipulator. In the Chapter 1 real space manipulators are shown and they are redundant, meaning that they have more joints than degrees of freedom. On one hand, this allows for more manoeuvrability, e.g. to have more option when moving the EE in order to be able to avoid obstacles, and that is the reason why they are designed in that way. On the other hand this means a more complex direct kinematics and most importantly a complex inverse kinematics, since there are infinite solution to realize one EE position. There are a lot of possible solution, such as secondary tasks in the control law, which lies on the null-space of the EE and thus do not interfere with the main task, but allow to select, among the infinite solutions, the more appropriate one for the specific case. Other methods, such as the project gradient or the reduced gradient ones can be inspected [8].

Furthermore, a 3D model would be preferable, but it would have increased even more the complexity of the calculation which, given the available software and the aim of the thesis (i.e. to find mass retrival methods) would not have affect the result significantly.

All of this has not been considered in this work and an improvement like this would benefit of a more realistic and implementable study case.

- The dynamics has been developed with lagrangian-based approached, however a Newton-Euler algorithm could be faster and more useful in space.
- The satellite has been modeled as a disk, but in reality satellites have different shapes and usually not symmetric. A 3D real model of a medium-dimensions satellite would improve the study.
- In the elastic case, null rotation has been assumed to simplify the calculations: a more realistic model could be introduced. Furthermore, [18] simplifies the frequency equation by considering $M_{D_1} = 0$ for $q_{2,0} = \pi/2$. There are more

complex time-dependent frequency equations that can be studied to describe a more realistic case, as suggested by [17].

The arms have been modeled as Euler-Bernoulli beams, but a finite-element method could lead to better results [15].

- Only plastic impact has been considered in this thesis, while in reality a perfectly anelastic or elastic impact does not exist; a general non-plastic case lies between these two. A more realistic scenario has been analyzed in [14] where two parameters are used to characterize the impact: the energy loss parameter and the friction parameter, which represent fractions of the loss of kinetic energy and friction impulse, respectively, of the plastic case.
- Closed form solution of the linearized joints' equation can be solved as a function of not only the unknown mass but also the initial joints' position, to better inspect the relation between mass error and chosen configuration.
- Other control laws other than the feedback linearization can be implemented. A conservation of momenta during pre-impact and post-impact phases control law would allow not only to control the joints but also the base orientation: it is possible to transfer the whole angular momentum from the base toward the free-floating manipulator and in the same time reducing the joints' velocities [23].

In [24] an optimal control law is developed as to bring the tumbling non-cooperative satellite to rest in minimum time while ensuring that the magnitude of the interaction torque between the manipulator and target remains below a prescribed value.

Authors in [25] present a combined prediction and motion-planning scheme for robotic capturing of a drifting and tumbling object with unknown dynamics using visual feedback. A Kalman filter estimates the states and a set of dynamics parameters of the object needed for long-term prediction of the motion from noisy measurements of a vision system.

- The software used was Wolfram, which is very intuitive and free for students but lacks computational power and allows only for short and simple symbolic equations. Maple has been used in the final part to get a symbolic closed form solution of the coupled second order linearized equation of motions. This study would benefit from a more powerful software with built-in functions for robotic simulations (working entirely on Maple or Simulink could be a solution).

6.4.2 Conclusion

To conclude, an overview on space manipulators and their functionality has been studied. I would like to thank professor Bortoluzzi for the proposed topic and the attention, availability and help given in these months.

Thank you also to all my friends and family who always supported me in my choices and constantly remember me that no matter how hard sometimes the path is, there is always a good opportunity to improve ourselves.

The University of Trento has thought me so many lessons in these years that it would be too long to list them all (as if this thesis were not already too long), but one thing that is for sure is that I learned to face problems I have never encountered before and to take the good from bad situations. I learned how to manage stress and future uncertainty, and I will bring all these soft skills in my upcoming working life with enthusiasm and curiosity.

As my grandfather always says, *ad maiora!*

Appendices

Appendix A

Matrices

The VMS mass and Coriolis matrices are listed.

$$\begin{pmatrix}
 m1 + m2 + mb & 0 \\
 0 & m1 + m2 + mb \\
 \frac{1}{2}(-l1(m1 + 2m2) \sin(q1(t) + \theta0(t)) - l2m2 \sin(q1(t) + q2(t) + \theta0(t))) & \frac{1}{2}(l1(m1 + 2m2) \cos(q1(t) + \theta0(t)) + l2m2 \cos(q1(t) + q2(t) + \theta0(t))) \\
 \frac{1}{2}(-l1(m1 + 2m2) \sin(q1(t) + \theta0(t)) - l2m2 \sin(q1(t) + q2(t) + \theta0(t))) & \frac{1}{2}(l1(m1 + 2m2) \cos(q1(t) + \theta0(t)) + l2m2 \cos(q1(t) + q2(t) + \theta0(t))) \\
 -\frac{1}{2}l2m2 \sin(q1(t) + q2(t) + \theta0(t)) & \frac{1}{2}l2m2 \cos(q1(t) + q2(t) + \theta0(t)) \\
 \frac{1}{2}(-l1(m1 + 2m2) \sin(q1(t) + \theta0(t)) - l2m2 \sin(q1(t) + q2(t) + \theta0(t))) & \frac{1}{2}(-l1(m1 + 2m2) \sin(q1(t) + \theta0(t)) - l2m2 \sin(q1(t) + q2(t) + \theta0(t))) \\
 \frac{1}{2}(l1(m1 + 2m2) \cos(q1(t) + \theta0(t)) + l2m2 \cos(q1(t) + q2(t) + \theta0(t))) & \frac{1}{2}(l1(m1 + 2m2) \cos(q1(t) + \theta0(t)) + l2m2 \cos(q1(t) + q2(t) + \theta0(t))) \\
 \frac{1}{6}(2l1^2(m1 + 3m2) + 6l1l2m2 \cos(q2(t)) + 2l2^2m2 + 3mbr^2) & \frac{1}{3}(l1^2(m1 + 3m2) + 3l1l2m2 \cos(q2(t)) + l2^2m2) \\
 \frac{1}{3}(l1^2(m1 + 3m2) + 3l1l2m2 \cos(q2(t)) + l2^2m2) & \frac{1}{3}(l1^2(m1 + 3m2) + 3l1l2m2 \cos(q2(t)) + l2^2m2) \\
 \frac{1}{6}l2m2(3l1 \cos(q2(t)) + 2l2) & \frac{1}{6}l2m2(3l1 \cos(q2(t)) + 2l2) \\
 & -\frac{1}{2}l2m2 \sin(q1(t) + q2(t) + \theta0(t)) \\
 & \frac{1}{2}l2m2 \cos(q1(t) + q2(t) + \theta0(t)) \\
 & \frac{1}{6}l2m2(3l1 \cos(q2(t)) + 2l2) \\
 & \frac{1}{6}l2m2(3l1 \cos(q2(t)) + 2l2) \\
 & \frac{l2^2m2}{3}
 \end{pmatrix}$$

$$\left(\begin{array}{c} \frac{1}{2} \left(-11(m_1 + 2m_2) (q1'(t) + \theta0'(t))^2 \cos(q1(t) + \theta0(t)) + 12m^2 \sin(q2(t)) \sin(q1(t) + \theta0(t)) (q1'(t) + \theta0'(t)) + q2'(t) + \theta0'(t) \right)^2 - 12m^2 \cos(q2(t)) \cos(q1(t) + \theta0(t)) (q1'(t) + \theta0'(t))^2 \\ - \frac{1}{2} 111(m_1 + 2m_2) (q1'(t) + \theta0'(t))^2 \sin(q1(t) + \theta0(t)) - 12m^2 \sin(q2(t)) \cos(q1(t) + \theta0(t)) (q1'(t) + \theta0'(t)) + q2'(t) + \theta0'(t) \right)^2 \\ - \frac{1}{2} 1112m^2 q2'(t) \sin(q2(t)) (2q1'(t) + q2'(t) + 2\theta0'(t)) \\ - \frac{1}{2} 1112m^2 q2'(t) \sin(q2(t)) (2q1'(t) + q2'(t) + 2\theta0'(t)) \\ \frac{1}{2} 1112m^2 \sin(q2(t)) (q1'(t) + \theta0'(t))^2 \end{array} \right)$$

Appendix B

Code

Taylor expansion used to linearized the joints' equation of motions:

```
series[f_,v_,v_0,_order]:=Module[{e},Expand[Normal[Series[f/.Thread[v->v0+e*v],{e,0,order}]]/.e->1]]
```

Cost function for the continous coupled method:

```
errorFunction[q_, data_] := NIntegrate[(q - data)^2, {t, 0, 10}]
```

Cost function for the discrete coupled method:

```
FindMinimum[Sum[(DiscreteSol11[[k]] - DiscreteData11[[k]])^2, {k, 0, NP}], {m0,test}]
```

Maple code for copuled equations solution:

```
dsolve(eqns union {q1(0)=q__10,q2(0)=q__20,D(q1)(0)=vq__10,D(q2)(0)=vq__20},{q1(t),q2(t)},method=laplace)
assuming m0>0:
```

Coupled linearized joints' equations:

```
(3124.81 + 30.3704*m0)*diff(diff(q1(t), t), t) + (3124.81 + 30.3704*m0)*diff(diff(q2(t), t), t);
- 171878.0000 + 109421.0000*q1(t) + 109421.0000*q2(t) + 218733.0000*diff(q1(t), t) + 218733.0000*diff(q2(t), t);
(15624.1 + 61.6185*m0)*diff(diff(q1(t), t), t) + (3124.81 + 30.3704*m0)*diff(diff(q2(t), t), t)
- 171878.0000 + 231289.0000*q1(t) + 109421.0000*q2(t) + 462346.0000*diff(q1(t), t) + 218733.0000*diff(q2(t), t)
```

Solution of the coupled joints' equations obtained with Maple (next pages).

$$q_1(t) = \frac{\frac{312481}{312481} \left(\frac{\frac{1218065000 \sinh\left(\frac{200t\sqrt{3328389921325-9520358627mO}}{312481mO+124932900}\right)}{312481\sqrt{3328389921325-9520358627mO}} + \frac{200 \cosh\left(\frac{200t\sqrt{3328389921325-9520358627mO}}{312481mO+124932900}\right)}{312481} \right) e^{-\frac{1218065000t}{312481mO+124932900}} q_{1,0}} \\ + \frac{\frac{312481 \sinh\left(\frac{200t\sqrt{3328389921325-9520358627mO}}{312481mO+124932900}\right) (mO + \frac{124992900}{312481}) e^{-\frac{1218065000t}{312481mO+124932900}} v q_{1,0}}{200 \sqrt{3328389921325 - 9520358627mO}}}{(B.1)}$$

$$\begin{aligned}
q2(t) = & \frac{171878}{109421} - q_{10} e^{-\frac{1218065000 t}{312481 m O+124992900}} \cosh \left(\frac{200 t \sqrt{33283899921325}-9520358627 m O}{312481 m O+124992900}\right) \\
& +\frac{1}{25 \sqrt{29047778718100}-8307898846 m O} \frac{37963 \sinh \left(\frac{50 t \sqrt{29047778718100}-8307898846 m O}{75926 m O+7812025}\right)\left((v q_{10}+v q_{20}) m O+\frac{136708125 q_{10}}{37963}+\frac{7812025 v q_{10}}{75926}+\frac{7812025 v q_{20}}{4153949423}\right)-\frac{23497119108750}{75926 m O+7812025}}{37963} \\
& +\frac{\left((-312481 m O-124992900) v q_{10}-1218065000 v q_{10}\right) e^{-\frac{1218065000 t}{312481 m O+124992900}} \sinh \left(\frac{200 t \sqrt{33283899921325}-9520358627 m O}{312481 m O+124992900}\right)}{200 \sqrt{33283899921325}-9520358627 m O}+\frac{\cosh \left(\frac{50 t \sqrt{29047778718100}-8307898846 m O}{75926 m O+7812025}\right)\left(109421 \frac{273416250 t}{75926}-\frac{23497119108750}{75926 m O+7812025}\right)-109421}{109421} \\
\end{aligned}$$

Appendix C

Plots

Here, the same plots of Section 6.1 are listed for Simulation 2 (i.e. when the satellite approaches with a non-zero vertical velocity and zero horizontal velocity with north-to-south direction).

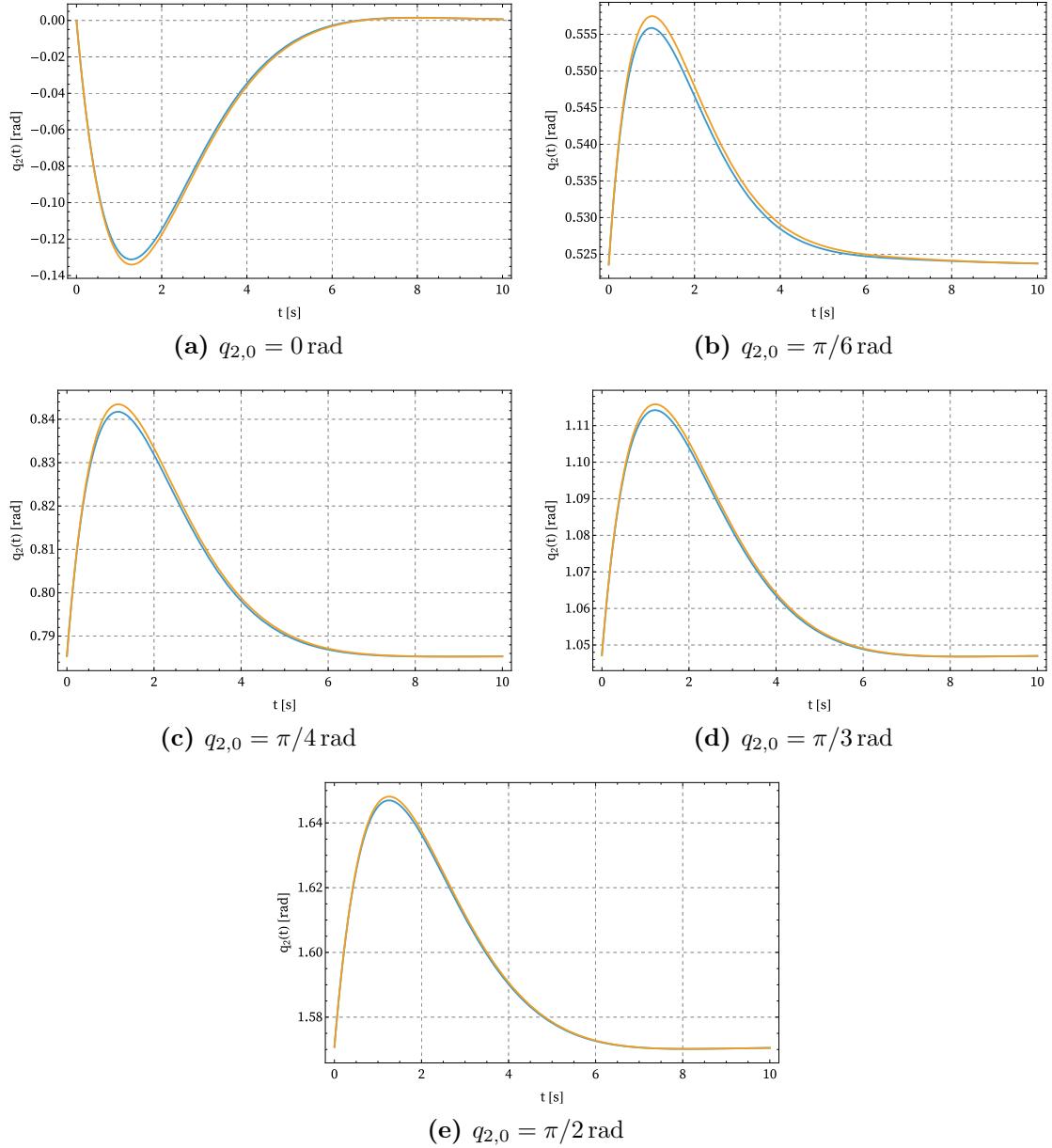


Figure C.1: Linearization error for different first arm's configuration, Simulation 2: non-linear solution in blue, linearized solution in orange, with $\hat{m} = 2000 \text{ kg}$.

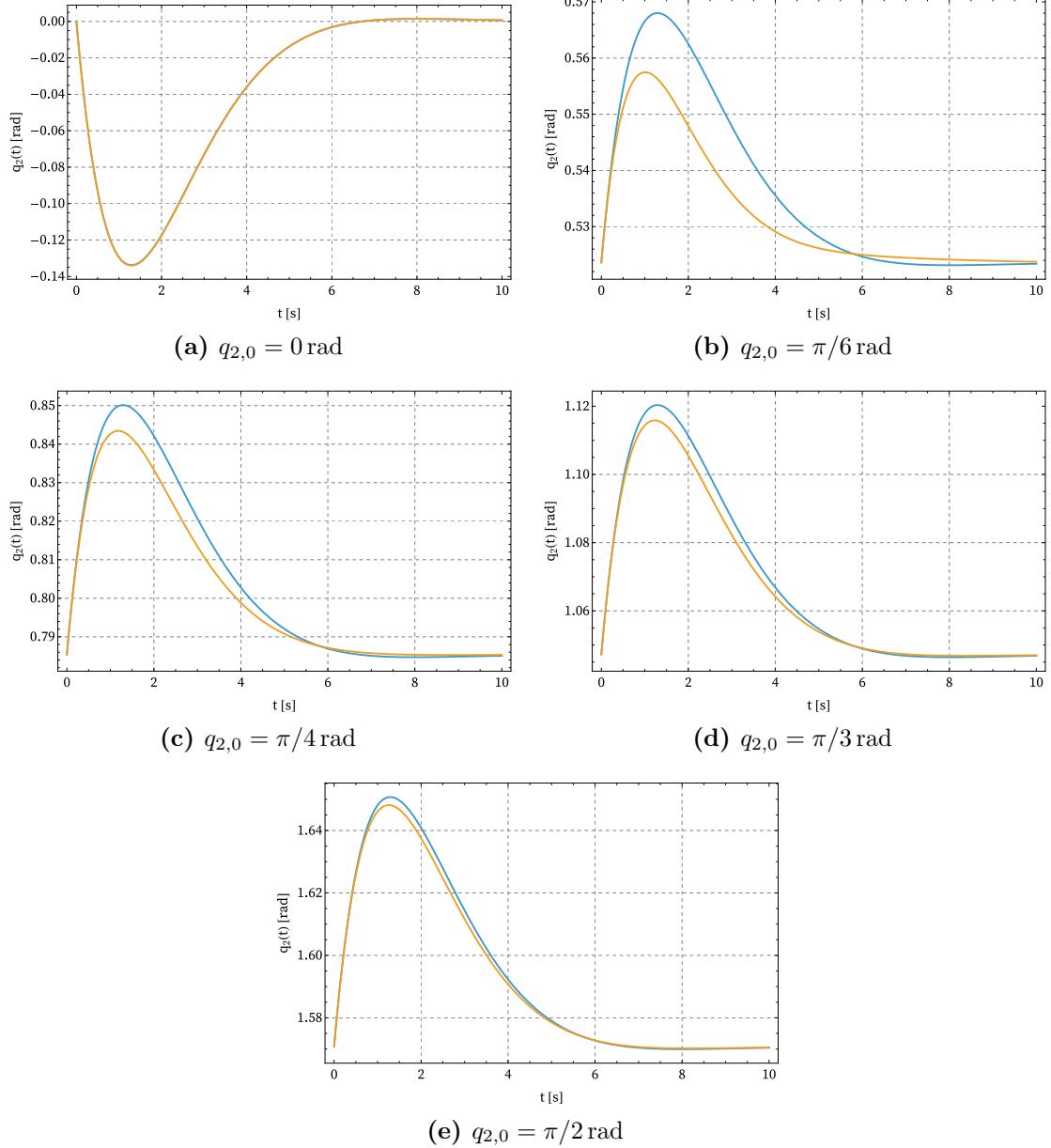


Figure C.2: Approximation error for different first arm's configuration, Simulation 2: non-approximated linearized solution in orange, approximated solution in blue, with $\hat{m} = 2000 \text{ kg}$.

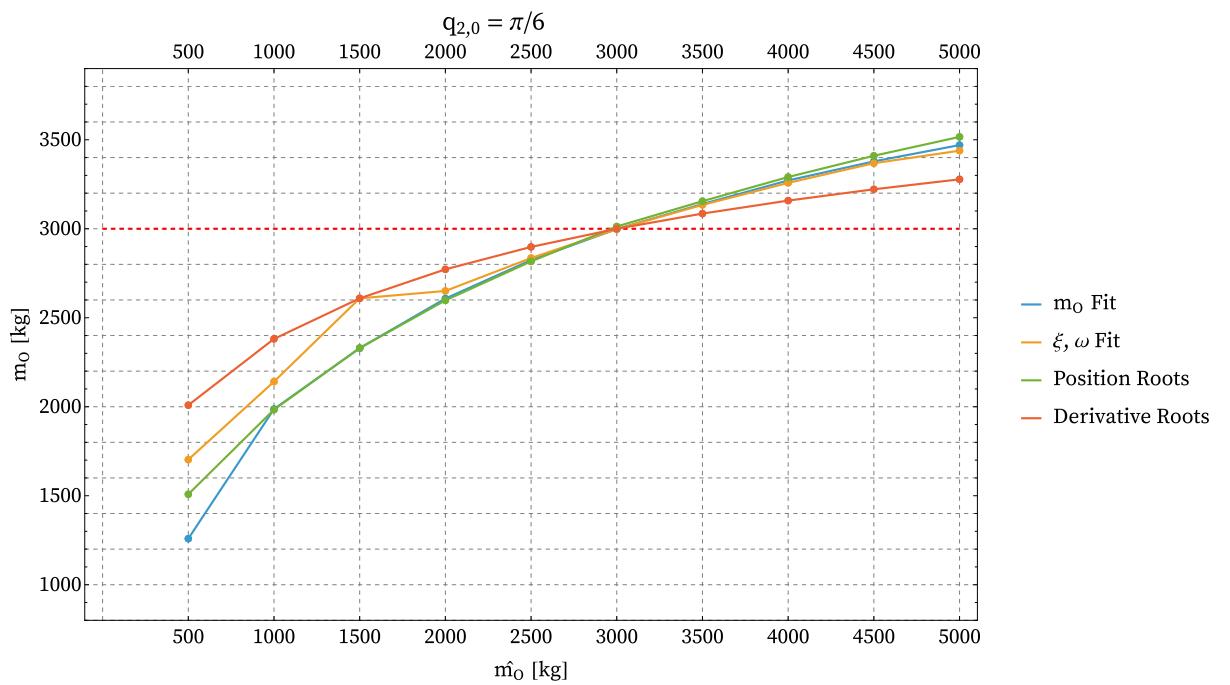


Figure C.3

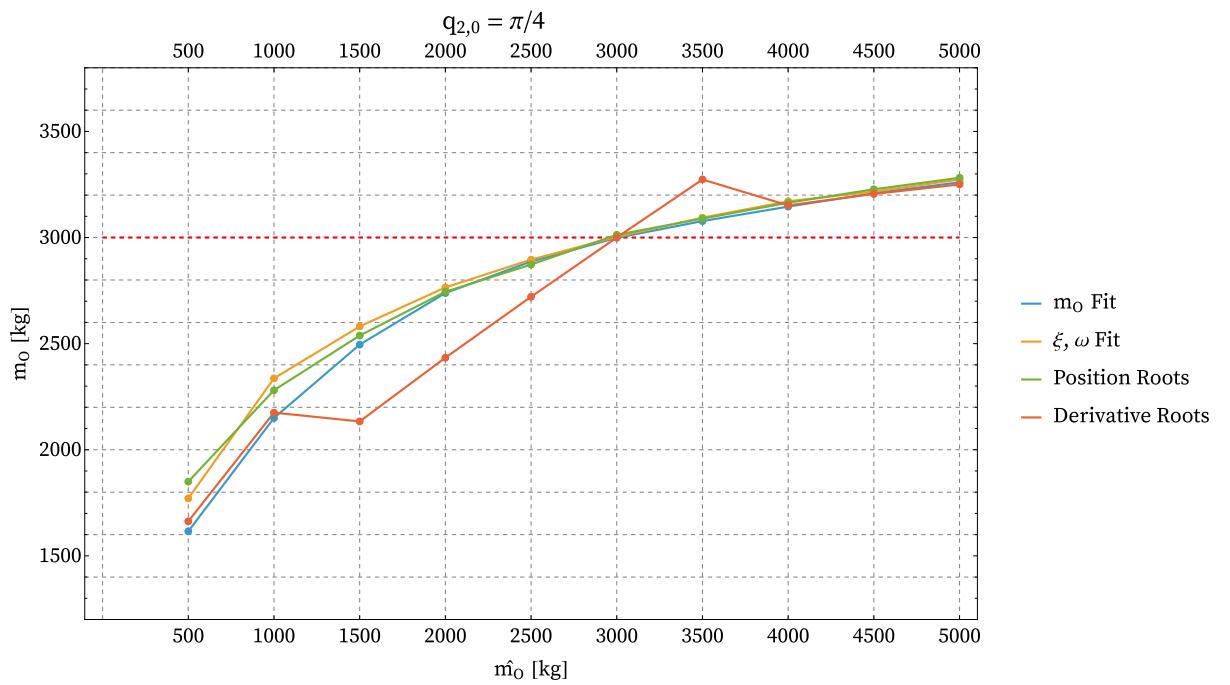


Figure C.4

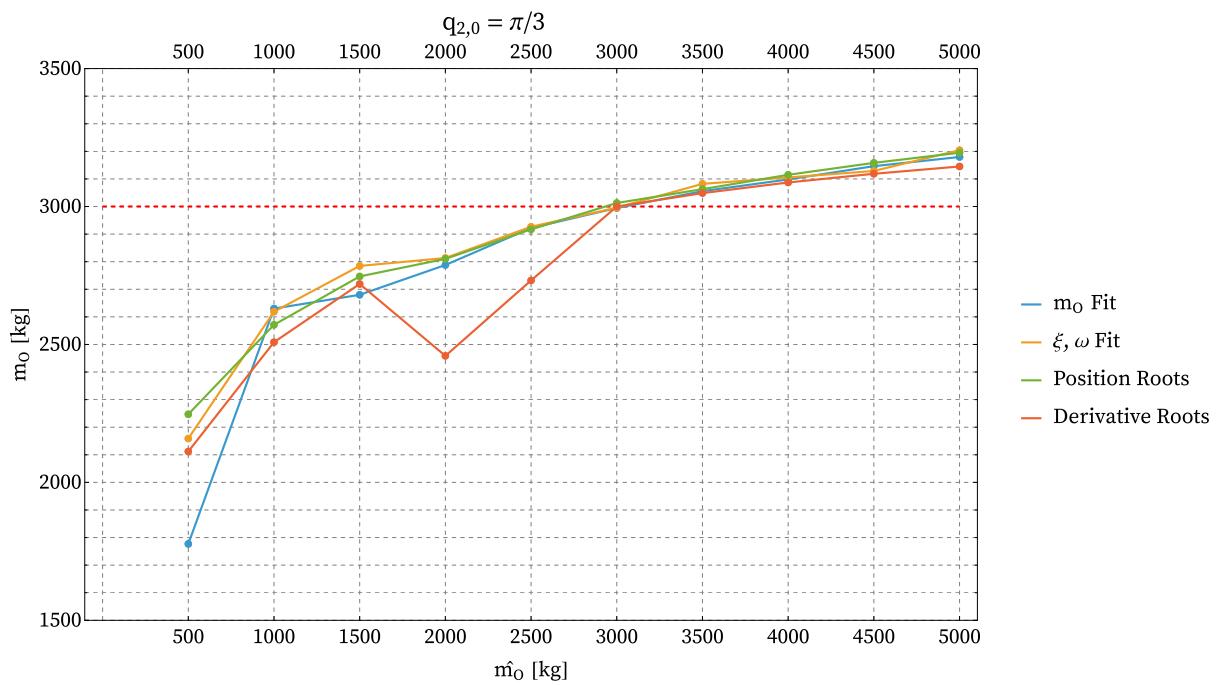


Figure C.5

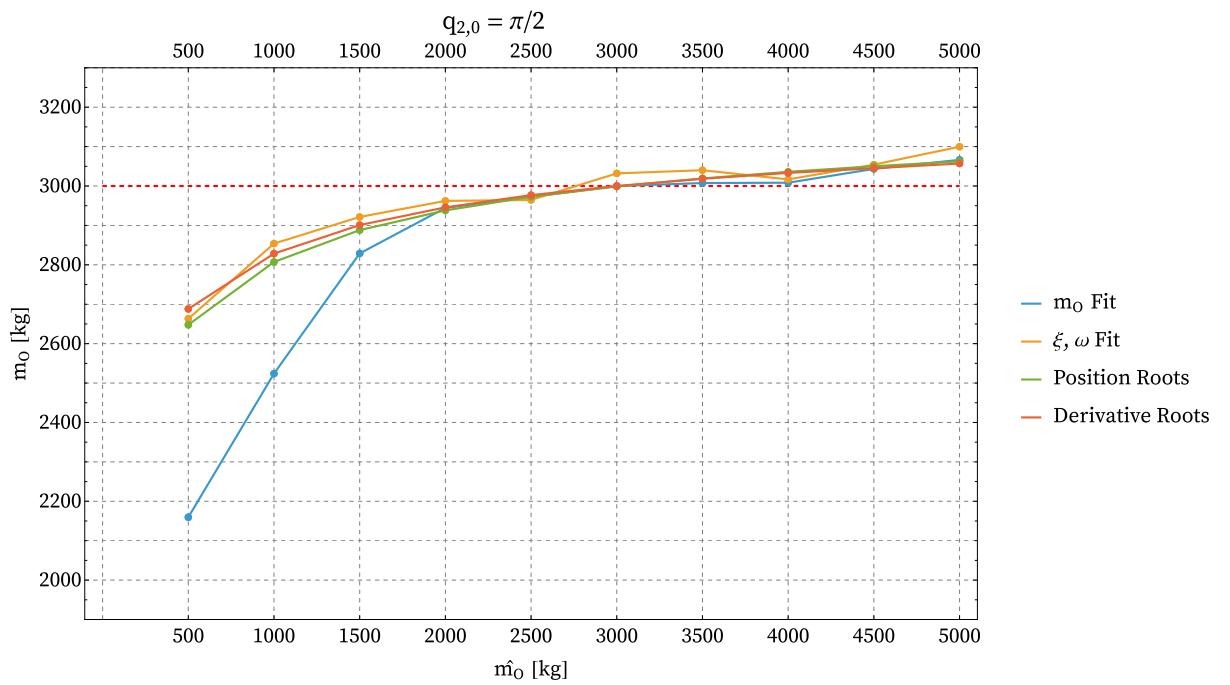


Figure C.6