POLITECNICO DI MILANO
DIPARTIMENTO DI SCIENZE E TECNOLOGIE AEROSPAZIALI
DOCTORAL PROGRAMME IN AEROSPACE ENGINEERING

GUIDANCE NAVIGATION CONTROL AND ROBOTICS FOR ON ORBIT SERVICING

Doctoral Dissertation of:
Aureliano Rivolta

Supervisor:
Prof. Michèle Lavagna

Tutor:
Prof. Lorenzo Dozio

The Chair of the Doctoral Program:
Prof. Luigi Vigevano

Cycle XXX

Aureliano Rivolta
Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano
via La Masa, 34
20156 Milano, Italy
E-mail: aureliano.rivolta@polimi.it

Abstract

A revolution is bound to happen in space in the near future as focus is shifting on colonization and commercialization of the Earth sphere and beyond. The Revolution takes the name of On Orbit Servicing (OOS) and is an old concept revived by the new spur of commercial endeavours in the space industry. Research on related topics like economics, liability, property rights, international cooperation and of course engineering is catching up again. Technical issues in OOS are not show-stoppers but can hinder the growth of the sector incredibly. Reliability, safety and costs of proximity operations are central in the development of OOS and the key aspects are related to robotics and GNC. In this thesis all topics related to Guidance Navigation Control & Robotics (GNCR) are covered simultaneously as disjoint analysis might prove to be inadequate to guarantee the proper synthesis of the control systems. Special focus is given to closed loop performance with camera images in the loop through high fidelity simulations of relevant OOS scenarios. Moreover, an adaptive control for both attitude and robotics operations is developed in order to track the desired performances under uncertain or time varying parameters like geometry or masses. Finally, after closed loop simulations of the proposed GNCR subsystem verify the expected performance, simple guidelines for future OOS mission are presented.

Acknowledgements

The first acknowledgement goes to my family that kept supporting me for all these years, giving me strength and comfort when I needed it the most. I have neglected you more than I ever wanted, hope you'll forgive me.

This work is dedicated to the ones that are still here, bearing my faults, and to those that still live in our most cherished memories.

I would like to thank from the bottom of my heart all the *colleagues* that shared with me the joy, sadness, frustration, rage and laughs of these intense three years. The feeling of not being alone in this journey helped me to focus on the target and overcome difficulties after difficulties. I am most grateful for that, I wouldn't have made this far without you.

Nomenclature

GNC Guidance Navigation & Control

GNCR Guidance Navigation Control & Robotics

OOS On Orbit Servicing

OOF On Orbit Refuelling

GEO Geostationary Earth Orbit

LEO Low Earth Orbit

MEO Medium Earth Orbit

LVLH Local Vertical Local Horizontal

ORU Orbit Replacement Unit

EVA Extra Vehicular Activity

CG Center of Mass

3D Tridimensional

2D Bidimensional

PID Proportional Integrative Derivative

LM Levenberg-Marquardt

INS Inertial Navigation System

RANSAC Random Sample Consensus

SURF Speeded Up Robust Features

LQR Linear Quadratic Regulator

NCC Normalized Cross Correlation

GPS Lobal Positioning System

ADI Adaptive Dynamic Inversion

PWM Pulse Width Modulation

SDM Sigma Delta Modulation

OR Orbital Robot

SISO Single Input Single Output

FOV Field of View

FFT Fast Fourier Transform

ADR Active Debris Removal

ETS-VII Engineering Test Satellite VII

LTI Linear time Invariant

ISS International Space Station

s/c Spacecraft

TRL Tecnology Readiness Level

DCM Direction Cosines Matrix

SD Steepest Descent

NG Newton-Gauss

fps frames per second

FPGA Field Programmable Gate Array

MRAC Model Reference Adaptive Control

ADI Adaptive Dynamical Inversion

MIMO Multiple Input Multiple Output

LMG Levenberg-Marquardt Guidance

LG Lyapunov Guidance

GJG Generalized Jacobian Guidance

Contents

CHAPTER 1

Introduction to On Orbit servicing

OOS is a hot topic in the space community and source of continuous research. Recently contracts have been signed by interested parties and some providers, others (Airbus for example) are starting to display the intent to bring OOS to life in the next couple of years.

Since the first OOS was achieved by astronaut's Extra Vehicular Activity (EVA) on Intelsat VI and the Hubble space telescope during the STS-49 and STS-61 missions there has been widespread research on how to expand the concept without EVAs. In the years the concept of automated resupply to the International Space Station (ISS) has been perfected (especially by ATV missions), however the ISS is one of a kind and commercial OOS involving civilian and non civilian endeavours requires a different perspective. Space industry is bound to be mostly driven by military and private companies and economic exploitation of OOS is the main concern, hence any OOS mission shall be sustainable in a market perspective without relying continuously on governmental agencies founds.

Earth observation, positioning systems and telecommunications are the main operators fields of interest and thus most of possible customers satellite orbits are in Low Earth Orbit (LEO) or Geostationary Earth Orbit (GEO) with the exception of GPS/Galileo orbits that are characterized by higher

eccentricity and pericenter than usual LEO missions, namely Medium Earth Orbit (MEO). The most lucrative area is probably GEO but any mission there has to pass through LEO area, thus many possibilities arise.

There are many technological challenges on the road to achieve robotic/automated OOS, which have sparked various research programs in the last decades. For example the DARPA funded Orbital Express mission successfully demonstrated autonomous rendezvous and docking [1–3] and refueling operations [4]. During the mission also berthing and servicing functionality have been accomplished [5]. Robotic manipulators can be used to grab, un-dock and replace Orbit Replacement Unit (ORU) and this capability has been also demonstrated by the Engineering Test Satellite VII (ETS-VII) of NASDA where autonomous rendezvous and berthing, visual servoing, ORU exchange and refueling have been accomplished [6–9]. Such feats were accomplished also thanks to previous missions like the STS-72 with the retrieval of the Space Flyer Unit [8].

1.1 The framework of OOS

In general, OOS sees an unmanned servicing satellite approaching a customer satellite in order to apply a somehow permanent modification that will increase the target lifetime, revenue, restore or increase functionalities or modify the target orbit. The natural appeal of increasing a satellite lifetime is that the customer does not have to build/buy and launch a new Spacecraft (s/c) to maintain its services, however servicing might not always be the best solution from the customer point of view.

Even if the cost of servicing is lower then replacement other factors might reduce the impact of servicing, for example obsolescence. Obsolescence of payload or subsystems as well as the risk connected to servicing operations are factors that may induce a cautious spacecraft owner to launch a new satellite, a practice that has always been implemented in the commercial space sector. Reference [10–12] analysed serviceability requests with focus on the customer needs and long term strategies.

Servicing might be the only solution for those satellite that have been inserted in the wrong orbit, since it could help achieve the mission at a lower cost compared to the launch of a new satellite. Since these events are not planned it is unlikely to constitute the core business of any company, rather is more believable that such emergency operations might be carried out by satellite with different main mission profile that have possibility to engage. The same reasoning can be applied to repairing a subsystem after a failure: programmed maintenance can be a good compromise. A more

comprehensive review of spacecraft failures and serviceability can be found in references [13, 14].

Nevertheless spacecraft servicing will be of importance in the space sector due to direct and collateral gains from the sustainability point of view. Increasing the lifetime of a spacecraft could reduce the risk of space debris proliferation if, and only if, the servicing companies are able to contain the risks of their own activities. Moreover, from Active Debris Removal (ADR) perspective, increasing OOS operation allows to increase the Tecnology Readiness Level (TRL) of grabbing and moving spacecraft in orbit and through heritage this could lead to the establishment of ADR technologies and endeavours in the common practice.

1.2 Orbital servicing scenarios

In this section the presentation of many OOS scenarios is carried out to establish the base of the subsequent technical analysis. Since OOS enables many different possible new perspective and innovative ideas the following list will not be extensive and will be limited to the main scenario currently pursued. As will be clear, many scenarios can also be solved with other scenario solutions.

1.2.1 Refueling

In the case of GEO satellites, the main lifetime limiting factor is the propellant mass needed for station keeping. Refueling allows the operator to extend control over the s/c for a longer duration postponing the need of a new launch. Also, refueling might enable the launch of satellites with next to empty fuel tanks as propellant can be fed to them on orbit as needed. The extension to exploration missions is straightforward and also seen as mandatory if one looks at the recent SpaceX claims [15].

As a matter of fact, fuel depletion is one of the major limiting factor of satellite lifespan and probably the less problematic to solve if a satellite is designed to be serviced. Retro compatibility on the fuel feed system can be very challenging and probably not worthily from a business perspective, however lifetime extension can be achieved by other means w.r.t. On Orbit Refuelling (OOF). A typical OOF mission requires a physical connection to be performed and maintained as fuel is passed from one satellite to the other.

A common practice among telecom operators is to sell nearly end-of-life GEO satellites to newcomers (such as developing countries) when a more performing satellite replacement is manufactured. In this case, an

OOF would greatly increase the market value of the old satellite providing benefits both to the operator (which would sell the satellite for a higher price) and/or to the buyer which would have a satellite with more lifetime ahead.

Since refueling operations do not require complex part substitution, all that is needed is to have a fuel storage line designed to accept external mass transfer. In any case a proper interface on the exterior part of the satellite is required for the link with the servicing satellite. Some companies are aiming to solve this problem with a different approach.

1.2.2 Payload Upgrade

Telecommunication satellites operator revenue is directly related to their transponders and payloads. This means that replacing or increasing the payload capabilities of a GEO telecom platform might increase the overall revenue. An already flying platform that is upgraded might have a shorter lifespan compared to a new platform altogether and the costs of serviceable design and performing the upgrade need to be considered to make a decision.

The first paradigm to be actually tested is the ORU handling, however such paradigm would lead to a very complex customer design architecture. Following reference [16] the concept of a Plug and Stay module could be preferred with respect to the Remove and Replace paradigm since it offers a more simple and safe approach.

1.2.3 Orbit re-insertion

Events where a s/c is put into the wrong orbit are not infrequent in the commercial space sector as pointed out in [13, 14] and by recent events connected to few Galileo satellites on 22/07/14 [17]. If this happens the satellite might see its lifetime decreased or its mission completely lost if it does not have enough propellant to correct the orbit.

Normally in such cases another spacecraft is prepared and launched, but this generates great economical loss both in terms of gross cost of the new spacecraft and in terms of loss revenue due to the delayed start of operations. Instead of launching a new spacecraft it might be more financially viable to ask a servicing company to intervene by catching the satellite and bringing it in the right orbit. This requires a satellite with increased docking capabilities, sufficiently close to reach the target and with enough propulsive ability to drag it to the correct orbit. Or, if feasible, use its propellant for the orbit change and then ask for refueling which is a more schedula-

ble operation. The operations are different and require a shorter interaction between spacecraft with respect to the nominal re-insertion case.

1.2.4 Repair

Repair scenarios where a robotic system repairs a pre-existing equipment on the serviced satellite is quite hard to realize. Substitution or integration of a new part is commonly recognized as a better strategy [16]. Only in few minor cases the use of repairs might prove the only viable solution to service a module, meaning that the concerned subsystem is not removable or bypassable. Such complicated operations are probably less appealing for both the customer and servicer, since the involved risk is high with low economical advantages.

If the customer satellite has a modular architecture it might as well be repairable. Some subsystem are easier to be replaced since they are usually positioned in the outer shell of the s/c. For example a star tracker can be replaced since it needs to have exposure to the outside and their small volume and power connections might allow for replacement if a proper connection mechanism is implemented. On the other hand, replacing a whole tank is much more difficult. Even the replacement of a solar array may prove challenging due to moving parts, rigid connection, power linkage and high rotational inertia variations.

1.2.5 Orbit inspection for insurance companies

Due to the increase in the number of spacecraft and space debris population a spur of insurance debates is expected: assessing damage due to collisions or orbit failures are of high interest to insurance companies in order to solve disputes. Assessing the damage on a satellite requires another satellite to orbit close by and able to take pictures or other measures with precision. This can also be applied to non-damaged satellite for a routine check, data that might be invaluable for satellite manufacturers as external measures are never available for commercial satellites. The concept of planned inspections has already been used during Shuttle missions for ablative protection checks prior to re-entry manoeuvre following the Columbia disaster.

Orbit inspection becomes appealing if not necessary in cases where the telemetry of the involved satellite is not available or might not help in discerning the reasons behind a failure [14]. Since no physical connection is required ant the amount of propellant used for the operation is not higher than that of other scenarios, it might be seen as a bonus in the servicing bundle proposed by a company.

1.2.6 On orbit transport

On orbit transport can be seen as the generalization of orbit insertion manoeuvres and is a service that some upper stages can already provide to some customers and up to some extent. A servicer that can dock and drag a misplaced satellite can be exploited also to transport satellites from the launcher injection orbit up to their target orbit, reducing substantially the weight of the customer satellite and the launch costs connected. The main issue is to reach a stage where the servicing companies is able to perform tugging in a profitable way, as high costs are involved. This could require several refueling operations of the tug which falls again on the OOF scenario and also shows that although the increase in OOS could reduce the number of launches of normal operators it will substantially increase the number of servicing launches as fuel is still prepared on ground. Hence, no sensible losses can be predicted for launch providers.

1.3 How to achieve OOS scenarios

Extending the lifetime of current orbiting satellite is not easy task and reasonably the only answer that has been proposed by industry is to attach a module to a satellite exploiting the adapter ring and perform station keeping and attitude control: there are several technical challenges even in this case. While seems reasonable to build new satellite with serviceability in mind, manufacturers still needs to be convinced to do so. Regardless, looking at the scenarios presented before, the main capabilities required for a servicing satellites are:

- rendezvous,
- formation flying with customer,
- repeated docking,
- target pointing,
- superior attitude control authority,
- intersatellite communication,
- robotic manipulation,
- fuel feeding system.

which require a dedicated Guidance Navigation & Control (GNC) subsystem as well as handling robotics arm and similar hardware. This will be the

main focus of this thesis and as will be shown later it might be appropriate to extend the acronym to GNCR as all aspects (GNC and robotics) are intertwined to achieve the performances needed for OOS.

Some technical issues concerning solely the servicing action are here presented without considering the engineering connected with the autonomous rendezvous, berthing, visual servoing, etc. The main issues are related to the difficulty of back-compatibility and module replacement or attachment as seen from the spacecraft system perspective.

1.3.1 Retro-compatibility difficulties

A major technical hurdle to servicing is the so called retro-compatibility: servicing an already flying satellite that was not designed to be serviced. The reason behind this general request from the market is the attempt to capitalize money from services as early as possible, while for a more refined version of OOS with new customers such revenues might be years in the future. Funding a servicing company is not easy task and the possibility to gain customers as well as proving capable is invaluable, although leads to sub-optimal solutions.

Retro compatibility issues may include: drain and fill valves are nominally sealed in orbit, non-return valve prevent ullage pressurant to flow back, multi-layer thermal insulator cover many instrument bays and so on. In order to service an old satellite a good understanding of the old design and a many new technologies are needed, thus reducing the appeal for a servicing company as development costs increase. Many tools, in fact, may have to be designed ad hoc for the target and a servicing s/c will become way more complex, heavy and less reliable. Hence the dilemma for new players.

What has been proposed and studied, is to think of serviceability in the design of new satellites with common international guidelines that would allow a flourishing new market with self sustainability.

1.3.2 Module augmentation

As mentioned before, a proposed solution is to use a module/satellite that can dock and increase capabilities of the old satellite. This increase the overall serviced s/c mass and inertia in orbit, requiring more propellant mass for station keeping and potentially for attitude control. Moreover, the increased external surface may also increase the drag for LEO satellites and gravity gradient related control issues. Since the mass of fuel needed for the station keeping of a satellite has linear dependence on the overall satellite

mass, increasing the mass of a satellite with the module requires more fuel. And the mass of the module can be contained up to a certain point since many subsystems are still needed. In case of an entire satellite clinging to the customer the fuel expenses might be doubling. Other issues in this regard are the connection and communication with the customer in order to provide the correct attitude control over all nominal mission phases.

1.3.3 Looking at the future

The best course of action, provided funds are found, is to prepare next generation of satellites to be serviceable. For example, the use of markers on the satellite outer appearance can greatly increase the estimation performances needed for close proximity operations. This road would lead to a more sustainable market, however it will require more years to be effectively functioning, hence what is expected in the near future is some activities carried out with module augmentation before the whole market convinces itself that a sustainable OOS might increase the business opportunities and the revenues. Looking at the present day perspective, what is clear is that the request for OOS is increasing and if not well inserted in the larger picture we could have another "debris issue" after damage has been done.

1.4 Peculiar mission phases of a standard OOS mission

A servicing mission can be divided in different phases that can differ based on applications. Regardless, there are some general guidelines that can help to understand why studying GNCR is mandatory. In this section the focus is given on close proximity operations, meaning all the phases that the servicing satellite is performing approximately within a hundred meter range from the customer satellite.

1.4.1 Far Inspection

In this phase the servicer describe one or more relative elliptical orbit around the customer satellite and spin in such a way that its cameras are pointing to the target satellite. Scanning the satellite looking for damages, failures or evidence of collision requires the servicer to be at a safe distance in order to avoid collision. In this part narrow angle cameras can be exploited, although for navigation purposes a wider camera is a better choice.

From the perspective of further phases, the far inspection can be used to derive or refine a 3D model of the satellite to be later use for relative nav-

igation purposes. In OOS scenario it is expected to have a good reference model of the satellite, however, limitations due to contract terms, human errors or failures could prevent the use of a good 3D reference model. Moreover, a wrong model can lead to possible inaccuracies or total divergence of the navigation and pose hazards on the customer.

1.4.2 Rendezvous & close approach

From the far inspection orbit, or a similar relative parking orbit, the servicer will move towards the target closing distances in the order of few meters. This phase is common to all approaches where the mission requires direct intervention or if a closer inspection is deemed necessary. The trajectory shall be designed to require the minimum amount of fuel expense in terms of position and attitude control as well as ensuring a safe abort in case of particular occurrences. The latter should be the driving requirement. In this phase the relative state control often has to place the servicer at the right position and on the right side of the target satellite. Depending on the distance from Earth, such phase might be more or less demanding in terms of fuel expense, duration and attitude control effort.

1.4.3 Docking/berthing

In a docking operation the servicer push forward to meet the customer rigidly linking itself to a docking port or ring adapter. At the completion of this phase the two satellites are rigidly linked and every action the servicer perform will affect the serviced satellite in terms of position and attitude disturbs as center of mass location and inertia changes.

The berthing options requires a robotic arm to connect to the serviced satellite and perform a softer docking or other operations. This is in general more appealing than a direct docking since customers satellite were not designed for direct docking operations and including that for future satellite might be too demanding.

The third option sees the servicer not exploiting a rigid docking connection but only robotic arms to perform all the operations. With good relative control and appropriately designed arms, this options permits to lower the influence on the customer satellite while executing tasks. In principle this options could burden less the nominal operations of the customer, however a full control with the proper safety measures are necessary. Drawbacks of this approach are higher costs related to fuel depletion and energy usage in the attitude and position control.

1.4.4 OOS operations

This phase is in general the riskier and thus it requires a special handling in order to deliver the service with minimum interference and maximum safety. During this phase all the servicing operations directly involving the customer are performed: a fluidic link is established in case of refueling, a module might be extracted and replaced in case of repair, a thrusting maneuver might be performed in case of tugging/relocation.

1.4.5 Un-docking

After all the OOS operations have been completed the servicer disengage the target. This includes un-docking and subsequent orbital manoeuvres to distance from target. In fact, it can be seen as the reverse of the approach phase where fuel expense and safety of the customer satellite are the main concerns. A second far inspection is unlikely to be requested unless a failure caused the mission to abort.

1.5 Customer perspective

The relation between customer and servicer is very important in the upcoming OOS revolution. Due to the rise of the new market and the risk adverse space industry the servicers need to provide a service with appealing features. Assuming that servicing operations are feasible both in terms of technology and costs, the issue here analysed is related to the possible request a customer might ask to a servicing company from the perspective of GNCR operations.

1.5.1 Customer satellite inactivity

The longer the inactivity period of a customer satellite is expected to be, the higher the losses for the operators and the lower the appeal of the proposed OOS is, hence diminishing such inactivity times can increase the potential of a OOS provider. OOS proximity activities might require days to be fulfilled and in some applications the customer satellite might be requested to switch off its pointing or even perform slew manoeuvres. From the customer perspective the ability to perform OOS operation while minimizing the influence on nominal activities is certainly appealing.

From a broader point of view, in case of repairs or orbit re-insertion the faster the servicing can rendezvous with the customer the better, however, this is a discussion for general OOS mission architectures rather than prox-

imity operations. Time optimization is negligible in the bigger picture with respect to the economics of achieving a profitable business.

On scheduled activities, like periodic refueling, the customer might find appealing the ability of the servicer to work without requiring the customer satellite to change its attitude. This poses heavy requirements in terms of flexibility of operations, meaning that more fuel needs to be carried on, robotics must have higher reach and so on.

1.5.2 Risks

OOS operations come with inherent risks associated to activities like docking, berthing or in general interaction with a satellite. Such risks need to be minimized as possible through a careful design of the servicer segment. The systems to be used during OOS operations need to have enough redundancy to allow operation in case of failures or at the very least to avoid jeopardizing the safety of the customer satellite. Safe escape in case of any potentially hazardous issue is also a good asset.

1.5.3 Restriction

Due to legal issues concerning patents or due to sensible equipment on the customer, servicer might have to provide operations while not inspecting certain parts of the satellite or avoiding the use of laser sensors (or other active measurements) or even perform operations with non-nominal attitude to avoid blocking radiators or similar equipment. This might impose strict requirements on the sensor suite and request the GNCR system to work with muted sensors or with special handling of the vision sensors, for example. The general design of the GNCR subsystem should be flexible enough to account for unexpected request and able to work without the need to re-design the sensor suite for each customer.

1.6 Research on GNCR

OOS scenarios require several operations to be performed, hence the development of a subsystem able to cope with most of the tasks is a key feature in the development of a servicer satellite. Although the “one for all” servicer satellite is highly unlikely, it is extremely plausible the design of a servicer satellite with more than one mission capability. Behind this reasoning there is the full exploitation of the lifetime of the servicer as many task can be planned beforehand and gaps of inactivity are expected.

The attitude control system of the satellite shall be able to cope with pointing accuracy needed for both inspection and proximity operations as well as increase the output when controlling larger ensemble or during phases with variable geometry. The control needs to adapt to changes and be as much as reliable as possible in faces of unforeseen events.

Since docking or proximity station keeping is involved, the position control of the satellite must be ON for many phases and this might create problem when it is coupling with the motion of a robotic arm as unwanted torque are generated due to shift in center of mass.

Also, when moving a robotic arm, the base position and attitude control needs to have precise performances and frequency influence, otherwise the different control parts could enter in conflict and jeopardize the whole mission.

Many operations require a precise pointing or pose determination, hence cameras are exploited. Due to hardware limitations the frequency of estimation using cameras might be quite low, hence proper filtering and propagation might be necessary in order to have higher frequency estimation.

An important part of the analysis and research connected is thus the study of closed loop performances to address the compatibility and limits of each part of the subsystem. For example if the output of camera estimation of the target direction is not well filtered for the attitude control it is very likely that the servicer satellite would be unable to follow the target and resulting in losing the target from the camera and switching mode to recover the pointing.

Many of these aspects are also difficult to test and require very complex experimental set-up like several robotic arms able to simulate the dynamics of two satellite in close proximity. There are few laboratories able to reproduce most of the scenarios here presented, however looking at the complete picture it is difficult to reproduce some effects caused by nominal rotation of satellites or to the use of thrusters. On the other hand simulating images takes way more time than using experimental equipment and more cumbersome to tailor.

1.7 Structure of the thesis

Tackling the guidance, navigation and control issues separately is easier but sometimes one loses the perspective and attachment to reality and capabilities. Study of the whole process are harder to find in literature due to the increase complexity and difficulties to disjoint effects and pinpoint problematic. Here the first chapters will deal with each part separately and

on the last part focus will be given to joint effects in order to show the performance of the elements introduced beforehand.

Chapter 2 will introduce all the modelling issues to provide a framework with enough detail to simulate most of OOS operations. In particular a multibody code used for simulations will be presented without going into the detail of the formulation used for the kinematics representation. The kinematic representation using dual quaternions will be presented in Appendix A as it is not the main focus of the research, rather a necessary and elegant tool for most of subsequent applications.

Chapter 3 will deal with guidance problematic. Here the design of relative trajectories for inspection will be addressed, as well as basic guidance laws for pointing or position control.

Chapter 4 focus on navigation in the sense of state determination. Here some vision based navigation topics are analysed at higher level, without entering in the details of computer vision as it is not the main focus of research. Then the filtering required for precise attitude determination as well as relative position is addressed and stress is put on the performance variation based on mission requirements. Namely performances might vary depending on pointing requirements, as faster moving references would reduce the expected disturbance rejection.

Finally, Chapter 5 will present the attitude and position control of the servicer satellite. Focus is given to Lyapunov stability and adaptation of the developed controller. Robustness and adaptivity are then tested and results shown in order to assure the capability of the attitude control to overcome variable geometry configuration issues or other connected to some OOS scenarios.

Chapter 6 is focused on the control of robotic arms. State determination, trajectory design and adaptive controller are studied together. In this chapter ideas from previous chapters are used and translated into the robotic framework without effort. Sometimes the point of view of robotics can give useful hints to satellite control and vice versa. The use of the same kinematic representation for robot and satellite could also help in this regard.

The study carried out in each chapter is then mixed in Chapter 7 where relevant simulation scenarios are presented. As hinted in the first chapter, there exist many OOS scenarios and testing all combinations requires a lot of time, however the key features to be analysed can be seen in few application scenarios. Simulations will be limited but richer in content trying also to show the coupling of effects (or the lack of, thanks to proper control) in the most demanding conditions. Such simulation will be limited to LEO as it is far more demanding in terms of control action, effort and higher fre-

quency of disturbs. The first of the two presented scenarios has been shown before in [18] while preliminary results of the complete GNCR loop were firstly presented in [19].

The final chapter, Conclusions, sums up what has been developed and some general guidelines for the research on GNCR as well as OOS are provided.

In the end Appendix A, Appendix B and Appendix C are included to address kinematic modelling, adaptive control and minimization algorithms that have been used though the development of the GNCR research.

CHAPTER 2

Orbital robot model

In this chapter a review of classical celestial mechanics, attitude dynamics and space robotics modelling will be presented. The purpose of this review and presentation of the theoretical background of the solver used is to indicate the characteristics of the dynamical system on which the GNCR is built on. The simulation of an orbital robots require special care and is the core base on which the control can be tested and implemented. Since the central aspect of the research is the closed loop performance of the whole system, the characterization of the iteration of each element is vital to provide a physical connection between all the subsystem involved. The decision of developing a new orbital robotics simulator using dual quaternions as kinematic core pays off in terms of capability, expandability and of course development of all features involved.

2.1 Review of Orbital & Attitude mechanics

2.1.1 Keplerian motion

The dynamics of the Center of Mass (CG) of a rigid body is given by Newton equation of motion stating that the variation in time of the linear mo-

mentum is equal to the external forces applied to the body, with respect to an inertial reference frame. Thus

$$\frac{d}{dt}\left(m\boldsymbol{v}\right) = \sum_i \boldsymbol{f}_i \tag{2.1}$$

where m is the mass of the body, $\boldsymbol{v}$ its velocity and $\boldsymbol{f}_i$ the forces applied to the system. Then, for a rigid body orbiting around a single massive body, the main force to take into account is the gravitational pull that can be modelled accordingly to Newton law of gravitation as follows

$$m\ddot{\boldsymbol{r}} = -\frac{\mu m}{\|\boldsymbol{r}\|^3}\boldsymbol{r} + \boldsymbol{f} \tag{2.2}$$

where μ is the gravitational constant of the main attractor, $\boldsymbol{r}$ the position vector of the orbiting body computed from the main attractor CG and $\boldsymbol{f}$ the sum of other forces applied to the system. The underlying assumption is to consider the main attractor to be still or moving in linear constant motion, which is never the case. Such assumptions holds well enough for many problem of interest. Considering null or negligible other forces it follows that the system motion is central, thus the angular momentum is conserved. Such quantity, for this system, is defined as follows

$$\boldsymbol{h} = \boldsymbol{r} \times \boldsymbol{v} \tag{2.3}$$

where $\times$ denotes the cross or vector product and $\boldsymbol{v}$ is of course the velocity, derivative of the position vector $\boldsymbol{r}$ expressed in the said reference frame. Angular momentum conservation can be proven easily by differentiating Eq. (2.3) and substituting Eq. (2.2) with null $\boldsymbol{f}$.

From this consideration, previous assumptions, crossing Eq. (2.2) with the angular momentum and considering that the gravitational field is conservative, it is possible to determine the position of the orbiting body in time through six parameters that represent the analytical solution of the restricted two body problem. For this research the influence of other massive bodies is not taken into account as for many of the applications here considered can be analysed considering satellites to be small bodies close to the planet.

The most used set of parameters are often referred to as Keplerian parameters, however different parametrization are possible. Of these six constant two represent the shape of the orbit, which must be a conic according to Kepler's studies and Newton's formulation, three identify the orientation of the orbit in a 3D space and the last one links the position along the orbit

with time. The shape is identified by the semi-major axis a of the conic and the eccentricity e, restricted to be between zero and one for closed orbits, being zero for circular orbits. In the reference frame with z axis aligned with angular momentum and x axis aligned with the minimum orbital distance (eccentricity vector) the position vector can be written as follows

$$\boldsymbol{r}_{orb} = \frac{\|\boldsymbol{h}\|^2/\mu}{1 + e\cos\vartheta} \begin{Bmatrix} \cos\vartheta \\ \sin\vartheta \\ 0 \end{Bmatrix} = \frac{a\,(1-e^2)}{1 + e\cos\vartheta} \begin{Bmatrix} \cos\vartheta \\ \sin\vartheta \\ 0 \end{Bmatrix} \tag{2.4}$$

Then the position vector $\boldsymbol{r}_{orb}$ in this frame can be translated back into the inertial $\boldsymbol{r}$ by using three sequential rotations in the order $z - x - z$ according to the values of the other three parameters: argument of pericenter ω , inclination i and right ascension of the ascending node Ω.

Looking at OOS applications we have GEO orbits that are almost circular and planar, LEO orbits often circular but inclined[1] and MEO orbit characterized by higher eccentricity. For many applications circular orbits are the only orbits involved, hence the main focus of subsequent analysis will be focused on circular orbits, however the software that implements the orbital motion shall not be restricted in any way to analyse circular orbits only.

2.1.2 Linearized orbit relative motion

In OOS robotics problems the relative motion between two satellite is of extreme importance and being able to obtain analytical or semi-analytical models can give very useful insights. The Chloessy-Wilthshire (or Hill) model [20] for circular orbits describe the position of a body with respect to a particular moving reference frame with a Linear time Invariant (LTI) system. The frame considered is fixed to a circular orbital motion, meaning that it has a nominal angular velocity equal to the orbital angular velocity with axes fixed to the orbital position and velocity vectors. The center of this frame is a geometrical point that is travelling trough space following a circular orbit, thus can be used to study the relative motion of objects in close proximity. The model holds for orbits with near zero eccentricity and in close proximity, as the linearity properties is lost if the axis aligned with the velocity is not adherent to the orbital track.

The Cartesian rotating reference frame has x as the outgoing radial direction, y directed as the orbital velocity and z parallel to the orbital angular

[1] Ex: polar orbits for Earth Observation

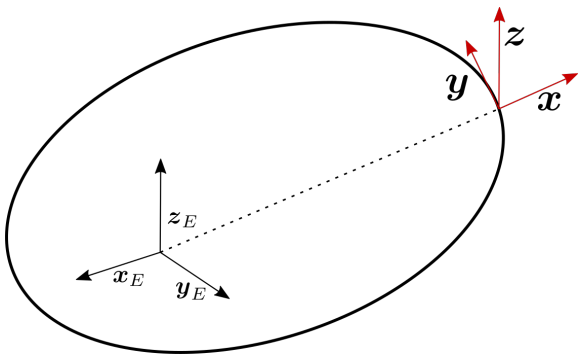


Figure 2.1: *Local vertical local horizontal reference frame*

momentum vector as shown in Figure 2.1. This frame is often referred to as Local Vertical Local Horizontal (LVLH) frame. The frame itself can be generalized for non circular orbits. Without considering forces other than the gravitational pull the result is the following

$$\begin{cases} \ddot{x} &= 2n\dot{y} + 3n^2 x \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2 z \end{cases} \tag{2.5}$$

Since it is a LTI the system admits an analytical solution in terms of velocity and position. Closed orbits and analytical paths are exploitable for rendezvous, inspection and other close proximity operations. It is also possible to add extra forces acting on the system of Eq. (2.5) in terms of accelerations in the LVLH frame.

This model is valid for almost circular orbits and can be seen as the linearisation of a more complete expansion of the relative equation of motion:

$$\begin{cases} \ddot{x} &= n^2\rho^3 \left(3 + e\cos\left(\vartheta\right)\right) x + 2n\rho^2\dot{y} - 2n\rho e \sin\left(\vartheta\right) y \\ \ddot{y} &= n^2\rho^3 e\cos\left(\vartheta\right) y - 2n\rho^2\dot{x} + 2n\rho e \sin\left(\vartheta\right) x \\ \ddot{z} &= -n^2\rho^3 z \end{cases} \tag{2.6}$$

with $\rho = 1 + e\cos\left(\vartheta\right)$. Assuming $e \simeq 0$ leads to Eq. (2.5). With respect to (2.5) this model is no longer LTI but a state transition matrix can still be found, although in terms of true anomaly (for example in [21]); literature in this regard is rather broad. In the above models it has been used a constant n, that represent the mean angular velocity of the rotating frame and can be written as

$$n = \frac{\mu^2}{\|\boldsymbol{h}\|^3} \tag{2.7}$$

which is related to the medium orbital motion. This can be put in relation to the true anomaly variation in time as

$$\dot{\vartheta} = n\rho^2 = n\left(1 + e\cos\left(\vartheta\right)\right)^2 \tag{2.8}$$

Integrating such relation with some manipulation leads to the implicit time of flight equation

$$\begin{cases} \tan\left(\frac{\Delta\vartheta}{2}\right) = \sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E}{2}\right) \\ n\Delta t = E - e\sin\left(E\right) \end{cases} \tag{2.9}$$

Also, a direct integration of (2.8) is possible but cumbersome.

2.1.3 Euler equations

The attitude dynamics of a rigid body in space can be easily represented by the Euler equation, having in mind that such equation is expressed in a non-inertial reference frame attached to the body itself in its CG. This allows to consider the inertia tensor constant and derive a non linear model with constant parameters. Moreover, there is a special reference frame where the inertia tensor becomes diagonal: the principal axes of inertia. The model does not require to use such a reference frame maintaining generality under the previous hypothesis.

$$\boldsymbol{I}_b\dot{\boldsymbol{\omega}}_b = -\boldsymbol{\omega}_b \times \left(\boldsymbol{I}_b\boldsymbol{\omega}_b\right) + \boldsymbol{\tau} \tag{2.10}$$

Eq. (5.1) present the Euler equations in vector form where $\boldsymbol{I}_b$ is the inertia tensor, $\boldsymbol{\omega}_b$ the angular velocity of the body expressed in that reference frame and $\boldsymbol{\tau}$ the external torques.

For a non rigid body, meaning also an ensemble of rigid bodies with relative motion allowed, the model would become

$$\boldsymbol{I}_b\dot{\boldsymbol{\omega}}_b = -\dot{\boldsymbol{I}}_b\boldsymbol{\omega}_b - \boldsymbol{\omega}_b \times \left(\boldsymbol{I}_b\boldsymbol{\omega}_b\right) + \boldsymbol{\tau} \tag{2.11}$$

where the new term $\dot{\boldsymbol{I}}_b\boldsymbol{\omega}_b$ is present due to the variation of the system inertia. This can be caused either by adding or removing mass or by internal movement: it can be caused by flexibility of parts of the satellite like solar panels, by internal fluid motion (sloshing) or by the motion of a robotic arm. It is clear that if the body is still, then such term would not have any

meaningful effect per se, however couplings are indeed possible. From the practical point of view, though, Eq. (2.11) is rarely used and a multibody representations is often preferred.

2.1.4 Perturbation in orbital mechanics and attitude dynamics

In Eqs. (2.2) and (5.1) have been inserted external forces and torques that influence the dynamics of the CG and the attitude. For the CG the disturbing actions can be due to a non uniform gravitational field, atmospheric drag, solar radiation pressure or third massive bodies gravitational pulls. Attitude can be influenced also by the satellite non uniform mass distribution and the coupling of magnetic field and residual internal currents. In these terms were included also control actions that are used to control the satellite attitude and position.

2.2 Multibody satellite model

Robotic manipulators in space are dynamically different from their counterparts on ground for several aspects. First of all the base they are mounted on is not fixed an unmovable, therefore the base motion influences the inertial forces of the arm and vice versa. As a matter of fact if the base is not attitude controlled any motion of the arm joints will produce a rotation of the base, causing it to tumble and preventing the end effector to reach the intended position with regular control paradigms. Another aspect is that the structure itself of the arms is much lighter due to the micro-gravity environment lesser influence causing flexibility to be an issue.

For this reasons there have been effort in the past to find a good model to represent the dynamics of a floating robot (in general a multibody spacecraft). In order to represent accurately the dynamics two conditions must be satisfied:

- the relative motion of all the bodies connected must be coherent with the CG position following the first law of dynamics;
- the CG must act as the center of rotation of the whole system.

Under these conditions it is possible to decouple the orbital motion, i.e. the CG motion, from the attitude motion. This means that a multibody satellite attitude can be described separately using a reference frame fixed with the CG. There are few methods to approach the problem, one is described in [22] , another in [23] where it makes extensive use of center of rotation for each limb and augmented body definition to reduce the problem on just

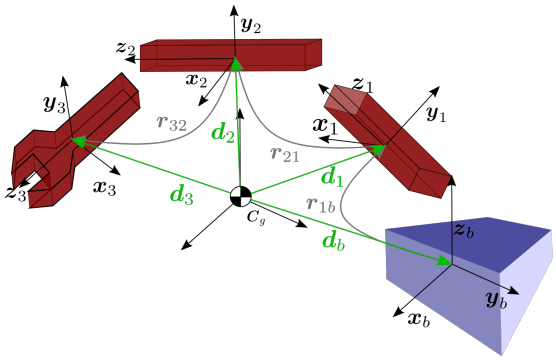


Figure 2.2: *Representation of a multibody satellite*

angular velocities. Should be noted that for robotic arm control there exist also a virtual reference control where an equivalent arm with base fixed with the CG is used to represent the end effector motion [24]. In the latter approach, as well as in the others, the focus is on revolute joints, while neglecting for the formulation prismatic joints. This can be a good approximation for robotic arms but certain linear mass motions can be encountered especially when dealing with refueling equivalent mass exchange.

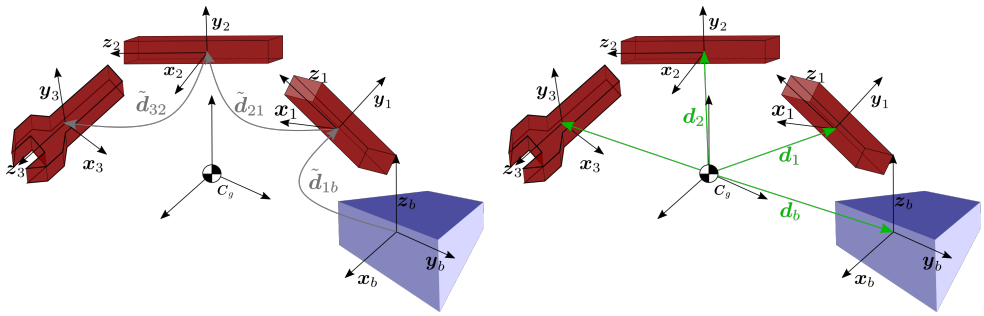
For this reason the model from [22] has been further taken to include several constraints and with a Lagrangian approach it is possible to include all external forces acting on the bodies so that only the forces and torques that have an influence of rotation around the CG are included in the dynamics. All net forces that would modify the CG position do not have influence on the rotational motion and can be taken into account separately.

Given an ensemble of n_b open chained connected rigid bodies orbiting around a main attractor, the constraints of CG motion and center of rotations must hold. Let us consider a reference frame fixed with the CG of the multibody satellite.The CG constraints can be written as follows

$$\sum_{i=1}^{n_b} m_i \boldsymbol{d}_i = \mathbf{0}_{3\times 1} \tag{2.12}$$

where m_i is body i mass, $\boldsymbol{d}_i$ the Cartesian position of the body with respect to the CG. It follows that the derivative of Eq. (2.12) must hold as well.

$$\sum_{i=1}^{n_b} m_i \boldsymbol{v}_i = \mathbf{0}_{3\times 1} \tag{2.13}$$



(a) *CG from the base body* **(b)** *Bodies position in the CG frame*

Figure 2.3: *Procedure to compute state in the center of mass frame*

where $\boldsymbol{v}_i$ is body i linear velocity. Thus we have formally 6 equations in a Cartesian representation of the linear position and velocities. Then it follows directly that the origin of the system is the instantaneous center of rotation of the system and all bodies can rotate around it and/or translate alongside the radial direction.

To enforce such constraint it is needed to compute the relative positions, attitude and velocities of all the bodies. It is possible to use the method from [22] in case of a single chain and taking care to consider the relative degrees of freedom regardless of their nature. In general it is possible to compute all the bodies position and attitude in a reference frame fixed with the base and then translate once the fictitious CG is computed. The rotation computed with this procedure are equal in both cases since the CG constraint does not involve rotations of frames. On the other hand the positions of all components with respect to the frame must be changed by adding a contribution due to all other elements. As a matter of fact, in free floating devices like space robots all the bodies influences each others.

Considering $\tilde{\boldsymbol{d}}_i$ the positions computed from the frame fixed with the base [2] then using Eq. (2.12) it gets that

$$\boldsymbol{d}_i = -\frac{1}{m_{tot}} \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_j \left(\tilde{\boldsymbol{d}}_j - \tilde{\boldsymbol{d}}_i \right)$$

leading to

[2] or any other points, the choice of the base is for easiness

$$\begin{cases} \boldsymbol{d}_i & = \left(1 - \frac{m_i}{m_{tot}}\right) \tilde{\boldsymbol{d}}_i - \frac{1}{m_{tot}} \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_j \tilde{\boldsymbol{d}}_j \\ \frac{d}{dt}\boldsymbol{d}_i & = \left(1 - \frac{m_i}{m_{tot}}\right) \frac{d}{dt}\tilde{\boldsymbol{d}}_i - \frac{1}{m_{tot}} \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_j \frac{d}{dt}\tilde{\boldsymbol{d}}_j \end{cases} \tag{2.14}$$

This generalization of the procedure seen in [22] is obtained by constructing a fictitious reference frame. In [22] it is highlighted the use of relative positions computed in such a way that the position vectors always points towards each body. In fact, in a straight chain where relative positions are computed from the base to the end effector the sign of each contribution depends on the order of the elements. Strictly generalizing this approach leads to complex programming, while the approach here adopted is quite straightforward.

The model is then re-written in terms of Dual Quaternions that have great advantages in dealing with multiple reference frames and singularities handling, not to mention the appeal of using the same entities to represent attitude and position kinematics and dynamics. The interested reader is suggested to use the Appendix A as reference for any dual quaternion related issue. Eq. (2.12) can be written in dual quaternion form as follows.

$$2 \sum_{i=1}^{n_b} m_i \left[\mathbf{q}_{i\otimes}^{-}\right] \boldsymbol{t}_i = \mathbf{0}_{4\times 1} \tag{2.15}$$

Then the system of Eq. (2.14) becomes

$$\boldsymbol{t}_i = \left(1 - \frac{m_i}{m_{tot}}\right) \tilde{\boldsymbol{t}}_i - \frac{1}{m_{tot}} \left[\mathbf{q}_{i\otimes}^{-}\right] \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_i \left[\mathbf{q}_{j\otimes}^{-}\right]^T \tilde{\boldsymbol{t}}_j \tag{2.16}$$

with derivatives easily obtained by differentiation as shown here after.

$$
\begin{cases}
\frac{d}{dt}\boldsymbol{t}_i & = \left(1 - \frac{m_i}{m_{tot}}\right) \frac{d}{dt}\tilde{\boldsymbol{t}}_i - \frac{1}{m_{tot}} \left(\mathbf{y}_1 + \mathbf{y}_2\right) \\
\mathbf{y}_1 & = \frac{d}{dt} \left[\mathbf{q}_{i\overline{\otimes}}\right] \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_i \left[\mathbf{q}_{j\overline{\otimes}}\right]^T \tilde{\boldsymbol{t}}_j \\
\mathbf{y}_2 & = \left[\mathbf{q}_{i\overline{\otimes}}\right] \sum_{\substack{j=1 \\ j \neq i}}^{n_b} m_i \frac{d}{dt} \left(\left[\mathbf{q}_{j\overline{\otimes}}\right]^T \tilde{\boldsymbol{t}}_j \right)
\end{cases}
\tag{2.17}
$$

In this formulation is clearer the influence of relative attitude on the position and velocities of the multibody satellite.

2.3 Multibody system solver

A multibody problem can be solved with many different numerical methods and paradigm, some more suited than others to assess an orbital robot. The equation of motion to be integrated can be computed using Lagrange equations, or through a Newton-Euler approach or even using a general purpose multibody code where the orbital robot constraints are inserted in discrete algebraic equations and solved. For the application in exams, the number of bodies to be considered is not very large and multiple chains with forces applied to each body must be assessed slightly reducing the appeal of the Newton-Euler approach. Moreover, in order to cope with the constraints seems reasonably more efficient to include the constraint directly in the dynamical formulation with respect to applying constraints on the bodies. A Lagrangian formulation, albeit less computationally efficient, is the selected procedure to develop a multibody code for orbital robots. During the development it has been observed that separating the orbital motion with respect to the local multibody attitude motion leads to less numerical problems related to different order of magnitude in the mass matrix when using a dual quaternion formulation. This is the price to pay for having a formulation without singularities and with consistent dimensions.

2.3.1 Lagrange formulation

The Lagrangian formulation uses a set of equations derived by energetic principles and the use of free coordinates/variable ϕ. For the mechanical problem in exam the energy at the core of the formulation is the sum of kinetic and potential energy. From the standpoint of using a somehow general approach it has been chosen to describe the gravitational force as an

external force without including it in the potential energy. This is due to the easier extension to new gravitational model in the software (ex: three bodies gravity influence, gravity gradient, etc.).

Hence, only the kinetic energy will be considered. For a body i the kinetic energy $E_{k,i}$ can be written in terms of Dual Quaternions as follows

$$E_{k,i} = \frac{1}{2}\dot{\boldsymbol{a}}_i{}^T \mathbf{S}_i^T \mathbb{M}_i \mathbf{S}_i \dot{\boldsymbol{a}}_i \tag{2.18}$$

where it has been used the generalized mass matrix $\mathbb{M}_i$, the derivative of the dual quaternion mapping the body attitude and position with respect to the CG reference $\dot{\boldsymbol{a}}_i$ and a mapping matrix $\mathbf{S}_i$ that depends on the dual quaternion $\boldsymbol{a}_i$ and is used to translate the dual quaternion derivative into the linear and angular velocities in the preferred reference frames. The frame attached to each body is considered centred in its CG and aligned with body axes. For the numerical implementation, to reduce the number of variables effectively stored, it is a principal reference frame. The generalized mass matrix is given by a composition of inertia tensor and mass

$$\mathbb{M}_i = \begin{bmatrix} \boldsymbol{I}_i & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 4} \\ \mathbf{0}_{1\times 3} & \frac{1}{2}\text{trace}\left(\boldsymbol{I}_i\right) & 0 \\ \mathbf{0}_{4\times 3} & 0 & \mathbf{I}_{4\times 4} m_i \end{bmatrix} \tag{2.19}$$

The extra term equal to the trace of the inertia, in this formulation is unnecessary, but has been formally kept since it is a term arising from the inertia tensor computation in $\mathbb{R}^4$ [25]. Should be noted that when the matrix inversion is computed, there could be singularity issues regardless, hence it is always advisable to append the quaternion normality constraints at acceleration level through double differentiation of the constraint itself. The matrix $\mathbf{S}_i$ depends on the frame position $\boldsymbol{a}_i$ and attitude with respect the inertial frame and is used to compute the body angular velocity from Eq. (A.18) and the inertial linear velocity from Eq. (A.22). Since $\boldsymbol{a}_i = \left\{\mathbf{q}_i^T \quad \boldsymbol{t}_i^T\right\}^T$.

$$\mathbf{S}_i = 2\begin{bmatrix} \left[\mathbf{q}_{i\otimes}^{+}\right]^T & \mathbf{0}_{4\times 4} \\ \left[\boldsymbol{t}_{i\otimes}^{-T}\right] & \left[\mathbf{q}_{i\otimes}^{-}\right]^T \end{bmatrix} \tag{2.20}$$

It is pointed out that it is possible to rewrite the product $\mathbf{S}_i\dot{\boldsymbol{a}}_i$ to obtain a form like $\mathbf{D}_i\boldsymbol{a}_i$. In this case

$$\mathbf{D}_i = 2\begin{bmatrix} \left[\dot{\mathbf{q}}_{i\otimes}^{+T}\right] & \mathbf{0}_{4\times 4} \\ \left[\dot{\boldsymbol{t}}_{i\otimes}^{-T}\right] & \left[\dot{\mathbf{q}}_{i\otimes}^{-}\right]^T \end{bmatrix} \tag{2.21}$$

then Eq. (2.18) can be written as

$$E_{k,i} = \frac{1}{2}\boldsymbol{a}_i^T \mathbf{D}_i^T \mathbb{M}_i \mathbf{D}_i \boldsymbol{a}_i \tag{2.22}$$

Lagrange equations for the whole system of n_b bodies are then written in terms of the free variables $\boldsymbol{\phi}$ describing the degrees of freedom of the system

$$\sum_{i=1}^{n_b} \left(\frac{d}{dt} \left(\frac{\partial E_{k,i}}{\partial \dot{\boldsymbol{\phi}}} \right) - \frac{\partial E_{k,i}}{\partial \boldsymbol{\phi}} \right) = \boldsymbol{f}^{'} \tag{2.23}$$

where $\boldsymbol{f}^{'}$ is the summation of all generalized forcing terms $\boldsymbol{f}_i^{'}$. This term is the product of the displacement induced by external forces and torques $\boldsymbol{f}$ projected on the degrees of freedom.

$$\boldsymbol{f}^{'} = \sum_{i=1}^{n_b} \left(\left(\frac{\partial \boldsymbol{a}_i}{\partial \boldsymbol{\phi}} \right)^T \left(\frac{\partial \left(\boldsymbol{d}_i, \mathbf{q}_i \right)}{\partial \boldsymbol{a}_i} \right)^T \boldsymbol{f}_i \right) \tag{2.24}$$

where in $\boldsymbol{f}_i$ are considered both forces and torques. The fourth component for each one is put to 0. The term $\frac{\partial (\boldsymbol{d}_i, \mathbf{q}_i)}{\partial \boldsymbol{a}_i}$ takes into account the virtual displacement with respect to the dual quaternion used for parametrization of the body.

$$\left(\frac{\partial \left(\boldsymbol{d}_i, \mathbf{q}_i \right)}{\partial \boldsymbol{a}_i} \right)^T = \begin{cases} \begin{bmatrix} \left[\mathbf{q}_{i\otimes}^{-} \right] & \left[\boldsymbol{t}_{i\otimes}^{+T} \right] \\ \mathbf{0}_{4\times 4} & \left[\mathbf{q}_{i\otimes}^{-} \right] \end{bmatrix} & \text{inertial} \\ \begin{bmatrix} \left[\mathbf{q}_{i\otimes}^{+} \right] & \left[\boldsymbol{t}_{i\otimes}^{+T} \right] \left[\mathbf{q}_{i\otimes}^{+} \right] \left[\mathbf{q}_{i\otimes}^{-} \right]^T \\ \mathbf{0}_{4\times 4} & \left[\mathbf{q}_{i\otimes}^{+} \right] \end{bmatrix} & \text{body} \end{cases} \tag{2.25}$$

To compute Eq. (2.23) two more objects must be computed: the Jacobian $\boldsymbol{J}_i$ and its derivative $\dot{\boldsymbol{J}}_i$.

$$\begin{cases} \boldsymbol{J}_i = \left(\frac{\partial \boldsymbol{a}_i}{\partial \boldsymbol{\phi}} \right) = \left(\frac{\partial \dot{\boldsymbol{a}}_i}{\partial \dot{\boldsymbol{\phi}}} \right) \\ \dot{\boldsymbol{J}}_i = \frac{d}{dt} \left(\frac{\partial \boldsymbol{a}_i}{\partial \boldsymbol{\phi}} \right) = \left(\frac{\partial \dot{\boldsymbol{a}}_i}{\partial \boldsymbol{\phi}} \right) \end{cases} \tag{2.26}$$

These terms are configuration dependent, meaning that depends on the links connecting bodies and how these are mapped to the free variables $\boldsymbol{\phi}$.

$$\begin{aligned} \boldsymbol{J}_i^T \mathbf{M}_i \boldsymbol{J}_i \ddot{\boldsymbol{\phi}} = &- \boldsymbol{J}_i^T \mathbf{M}_i \dot{\boldsymbol{J}}_i \dot{\boldsymbol{\phi}} - \boldsymbol{J}_i^T \dot{\mathbf{M}}_i \dot{\boldsymbol{a}}_i + \\ &\boldsymbol{J}_i{}^T \mathbf{D}_i^T \mathbb{M}_i \mathbf{D}_i \boldsymbol{a}_i + \boldsymbol{f}_i^{'} \end{aligned} \tag{2.27}$$

with

$$\begin{cases} \mathbf{M}_i = & \mathbf{S}_i^T \mathbb{M}_i \mathbf{S}_i \\ \dot{\mathbf{M}}_i = & \dot{\mathbf{S}_a}^T \mathbb{M}_a \mathbf{S}_a + \mathbf{S}_a^T \mathbb{M}_a \dot{\mathbf{S}_a} \end{cases} \tag{2.28}$$

The whole system then follows naturally

$$\sum_{i=1}^{n_b} \left(\boldsymbol{J}_i^T \mathbf{M}_i \boldsymbol{J}_i \right) \ddot{\boldsymbol{\phi}} = - \sum_{i=1}^{n_b} \left(\boldsymbol{J}_i^T \mathbf{M}_i \dot{\boldsymbol{J}}_i \right) \dot{\boldsymbol{\phi}} + \boldsymbol{f}^{'} + \sum_{i=1}^{n_b} \left(\boldsymbol{J}_i{}^T \mathbf{D}_i^T \mathbb{M}_i \mathbf{D}_i \boldsymbol{a}_i - \boldsymbol{J}_i^T \dot{\mathbf{M}}_i \dot{\boldsymbol{a}}_i \right) \tag{2.29}$$

If we consider a generic orbital robot we would have that the base attitude would need to be parametrized with quaternions, hence there will be the need to include the unitary norm constraint in the Lagrange equations. This can be solved by adding a Lagrange multiplier in Eq. (2.29) with the second derivative of the constraint. This method is not sufficient to solve the constraint issue and requires a constraint violation enforcing method post integration.

2.3.2 Center of mass

The CG motion of a multibody satellite ensemble is solved by projecting all the forces, gravity included, applied to the bodies to the CG. This permits to solve easily Eq (2.2) in Cartesian coordinates without involving others degrees of freedom for the solution of this. After several trial this has been deemed the most effective solution, as the force translation is a simple task and require less augmentation of the mass matrix. On the other hand the use of Eq (2.2) is the least precise in terms of orbital dynamics integration, as tends to accumulate more error over time. The use of Variation of Parameters is for sure a better strategy when longer time of simulations are involved, however for many scenarios here simulated the problem does not surface. Applications like on orbit assembly of large structures might require such a formulation.

2.3.3 Gravity gradient

A very important disturb for space robots, especially in LEO, is the *gravity gradient*. The term might result confusing since sometimes it might refer to the non uniform gravity field of Earth due to its own shape and

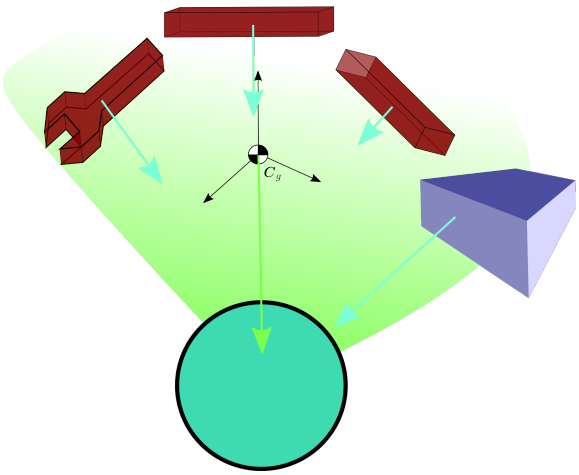


Figure 2.4: *Gravity gradient due to mass distribution*

mass distribution, however here is used with the intent of describing the small variations in gravity force acting on an ensemble of bodies orbiting in space. Intuitively we can recognize that in Eq (2.2) the gravity force depends on the object mass and distance from the center of the main attractor. Hence, even assuming Earth as perfect sphere and the bodies as points, all the bodies would experience a force slightly difference due to the small position variation and mass differences. Such variations might be small but the closer to the main attractor, the more influence may have on the bodies dynamics. The phenomena is well known in satellite attitude control as the reasoning here presented applies also to a single body due to its physical dimensions. Hence for a orbital robot the distributed mass of each link as well as its distribution in the link itself contributes to generate a small force variation hence a disturbing torque on all objects. In robotics the problematic of the gravity field is an open problem as constitutes a constant torque to be supplied to the joints in order to maintain position, requiring an integrator in the control loop. The difference for an orbital robot is that on Earth the gravity acceleration is so high that the gradient is negligible and acceleration can be considered constant in direction and intensity, while in space it all depends on the ensemble attitude. Of course the intensity of such disturb is lower with respect to Earth but not negligible as was pointed out in [6] looking at the Engineering Test Satellite VII robotics mission results.

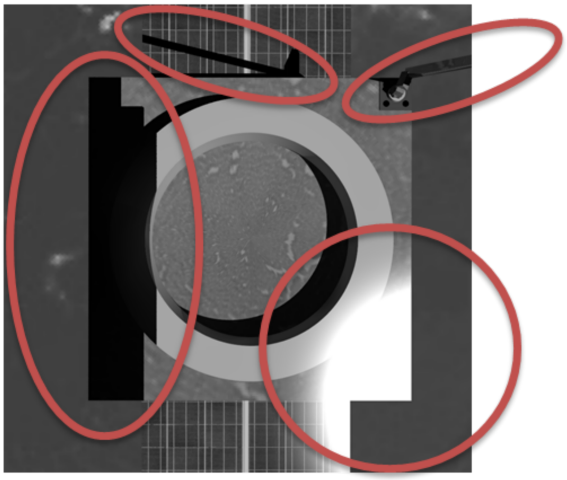


Figure 2.5: *Details of POV Ray based image rendering*

2.4 Camera and sensors models

When dealing with closed loop GNC aided by vision sensor, it is mandatory to have a good model to represent the camera but this is rarely done due to computational burden. For example in [26] images were not used in the loop but rather points and lines were assumed almost-perfect. In cases where images are generated, they might lack some realistic aspects. For example, images in [27] were not rendered with Earth in sight, self-shadows and shadows induced by servicer nor reflections, hence the quality of images is by far greater than in the real scenario. Increasing image sensor accuracy means to make use of advanced rendering techniques that require a lot of computation to be created, increasing notably by 5 or 10 times the total simulation time of the feedback loop.

In this work the images are rendered using POV Ray, a software based on ray-tracing. The photorealistic performance of POV Ray are testified in [28] where the commercial software PANGU, approved by ESA, has been validated using POV Ray itself.

The use of the software within Simulink is non trivial as POV Ray input has to be updated run and post-processed each iteration and uses a frustrating left hand, clockwise positive convention for kinematics. Even the simple image used in this work require about 5 seconds to be generated using several cores, increasing the details in the model would increase the rendering time by several factors.

The important features that a POV Ray generated image can make are

realistic reflections and shadows cast by several objects. By adding an Earth textured sphere one can include the disturbance for computer vision software of a background. Should be noted that the texture used is by far not high-definition, hence closeups might look poor. Enhancing the textures would require more processing power, hence the compromise. Such features can be seen directly in Figure 2.5.

Difficult aspects to include in POV Ray images are the sun in sight and the image noise, hence they are added afterwards in the simulation. The presence of the sun in direct view of the camera, when the Earth is not shadowing it, saturates the image, and this is done artificially by analysing at the sun position from the camera. Noise is added as a positive bias and random integer addition for each pixel. This allows to create noise that can reduce gradient based feature extraction performance of an otherwise polished image and shift the histogram of the image as perfect blackness is rarely found in navigation cameras.

2.4.1 Other Sensors

Aside from cameras, other sensors to be simulated that are relevant to the GNCR are star trackers and gyroscopes. It is possible to model the star tracker measurement using simulated images and a sat map, however this would increase the computational load more than necessary. In the present work star tracker are modelled using a simplified approach but with an approach close to reality. In practice the only part not simulated is the image, as a map of stars is randomly generated and at each iteration perfectly matched but noisy measurement are taken and analysed. Measurements are generated from the map using a pinhole camera model and translated into 2D features. Then random noise, affected by angular velocity, is added and vector measurements are finally generated. At the end a quaternion based solver of the Wahba problem provides attitude estimation. The last stage makes use of singular value decomposition and the quaternion estimated can shift sign without warning, creating a more realistic measurement case.

Gyroscopes measures body mounted angular velocities and are usually affected by static bias and random walk at low frequency. They are often sensitive to temperature, although this depends on the type of gyroscope considered. In simulations the noise is added as filtered band limited white noises with different power for low and high frequency. Completely static bias is finally added.

Other sensors modelled are encoders for angular velocity and position. Here noises are added either using a constant distribution with width equal

to bit dimension or with band limited white noises.

For all the sensors considered it is possible to derive a more accurate model if a particular component is considered, otherwise the simplified approach can still hold as main features of the measurements, especially in the frequency range, are maintained and characterized.

CHAPTER 3

Guidance

The term *Guidance* used here is intended broadly as a reference state or trajectory that is meant to achieve a mission objective. It might assume the form of orbital parameters, state space trajectory or even optimal path in a feedback loop. Looking at the broad perspective, it gives the reference for any control system such that, if followed closely by the system, the objective are reached. The methods to obtain such references might be different, going from optimal control theory to kinematics and geometry, using insights coming from applied computational mathematics or closed form solutions derived by human analytical skills. The main topics of this chapter are the study of relative trajectory for inspection and depot rendezvous, quaternion based Lyapunov optimal velocity reference, pointing for inspection and a general Dual Quaternion feedback velocity guidance.

Guidance for rendezvous of satellite literature is quite broad and rarely focuses only on inspection, rather inspection-like manoeuvres are inserted as an intermediate step. In many cases rendezvous is studied using vision sensors, like in [26] or in [29, 30]. Here a step back is taken in order to design inspection orbit with geometrical constraints. A brief analysis on a possible depot refueling trajectory is proposed based on simple orbital mechanics considerations.

When dealing with attitude control, the guidance here considered gives the reference velocity for the controller in order to follow a certain attitude reference. Typically the two loops are grouped together, but in some cases it might be wiser to divide the two. It is also straightforward to pass from one form to the other so in this section we can consider for quaternion based guidance past example like [31] where few form of quaternion guidance are analysed or in [32,33]. The problem of unwinding is, however, not addressed and stabilization might not occur for some or optimality might be lost. In general the phenomena is regulated using a sign function, that can cause chattering in some cases, the issue has been solved within certain boundaries in [34], although most of the discussion is about the global attractivity property of quaternion based control laws. In [35] modified Rodrigues parameters are used instead, however no minimal representation of SO3 (rotation group) is singularity free and thus arguably not globally converging.

3.1 Inspection orbit

The inspection orbit has to be designed in such a way that important requirements are satisfied and the safety of the customer satellite is ensured.

For circular orbits the model of Chapter 2 in the LVLH frame can be exploited and an analytical solution for periodic orbits can be found. From these solutions it is possible to design the proper trajectory to obtain the wanted results. The degrees of freedom in this problem are in general 6, usually position and velocities in time (initial, final or a mix), however if the target is a closed orbit one parameter is unnecessary, thus the d.o.f. are five. The goal is to determine at least five state conditions that allows the servicer to describe a closed orbit around the target. These conditions are determined as follows:

$$\begin{cases} \dot{x}_0 &= \pm n\frac{a}{2} \\ \dot{y}_0 &= 0 \\ \dot{z}_0 &= -n\frac{a}{2}\tan\eta \\ x_0 &= 0 \\ y_0 &= y_c \pm a \\ z_0 &= 0 \end{cases} \tag{3.1}$$

The relative trajectory here defined is an ellipse and the starting point is one of the absidal points. Here y_c is the geometrical center of the ellipse and

a is the semi-major axis of the said ellipse that can be put in relation with the minimum safety distance the servicer needs to keep from the serviced center of mass d_{min}.

$$a = 2\sqrt{d_{min}^2 + \frac{y_c^2}{3}} \tag{3.2}$$

For practical inspection seems legit to suggest $y_c = 0$ to have the customer satellite at the center, however if particular hardware limitations for the attitude control are present, then the strategy might be adjusted differently.

The angle η represents the inclination of the ellipse with respect the main orbital plane and can be put in relation with customer requirements of clearance in the nadir direction of the customer satellite. During inspection it is possible to let the customer satellite to be operational, thus a proper cone of semi-aperture δ in front of the satellite is required to be kept clear from the servicing satellite. Since the proposed orbit is symmetric, the same guaranteed aperture cone is free also in the zenith direction. Such angle η is thus given by

$$\eta = \pm \text{arctan}\left(\sin\delta + \frac{2\Delta z}{\cos\delta\sqrt{a^2 - y_c^2}}\right) \tag{3.3}$$

where the term Δz takes into account the maximum size of the servicing satellite. The sign ambiguity takes into account the two possible orbits that satisfy the clearance requirement. If we want to consider Δz as the maximum size of servicer and include it in the minimum distance requirement also for inclined orbit, then Eq (3.2) should be modified as follows:

$$\frac{r}{2}\sqrt{1 + \left(\sin\delta + \frac{2\Delta z}{r\cos\delta}\right)^2} - \Delta z > d_{min} \tag{3.4}$$

with

$$r = \sqrt{a^2 - y_c^2} \tag{3.5}$$

and η should be computed once again. Figure 3.1 presents an example for the trajectory here designed. In red the off-limits areas: a sphere from the target and two cones in the radial direction representing the space to be left clear for communication, observation and such activities. In green the sphere ideally representing the space occupied by the servicer satellite. Distances are respected according to the requirements and Eq.(3.4).

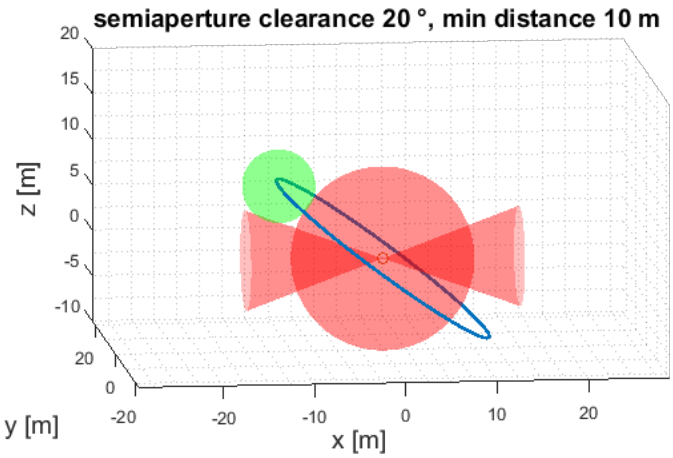


Figure 3.1: *Inspection trjectory example*

Fuel Depot rendezvous

In many OOS scenarios two satellites need to physically interact with each other through a connection. In order to achieve this feat the two have to come close and establish a connection though a mechanism. Typically one satellite is seen as a chaser and the other as a target but depending on the OOS scenario considered the role of servicer and customer might be reversed. Thinking about a GEO refueling the servicer reaches the client and provide the service, while for an early stage LEO refueling it is more likely to see the customer reaching the refueling station. The last scenario can also be applied in GEO in cases where a refueling satellites self-refuel at a station, an attractive economical option.

In general, the case where the client rendezvous with the depot is preferable to limit the amount of instruments and navigation tools the client satellite has to have in order to perform the rendezvous. This means that the depot should be equipped with all the sensors and communication systems to be able to autonomously guide the client to a safe docking.

Figure (3.2) present two version of the same docking procedure. The client approaches from the tangential direction and through simple phasing approaches the station at safe distance. In point 1 the first manoeuvre is performed, a simple burn in the orbital velocity direction moves the client into a relative orbit around the station. This radial approach has been chosen for safety: if the client is not able to manoeuvre due to unforeseen causes the natural drift will avoid possible collisions. The minimum distance is called x_{appr} and has also been used as a scale factor in the plots. From the

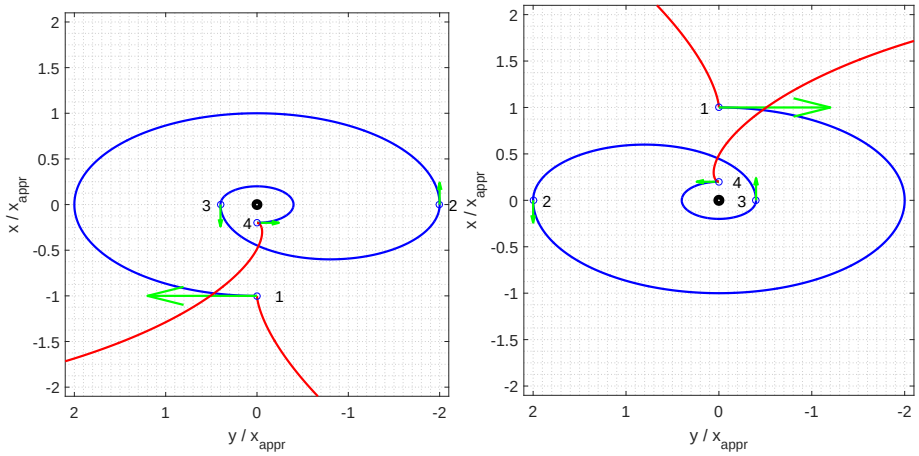


Figure 3.2: *Trajectories to Depot Docking*

relative orbits the station should have the possibility and time to refine the state.

Step 2 and 3 are two radial burns that are used to reduce the size of the relative orbit to the wanted dimensions. Should be noted that in case of great difference between the outer and inner orbits a repetition of intermediate steps can be used to lower the effort.

The capture, by means of a Canadarm-size robotic arm is performed in 4. The customer satellites does not arrive at null speed, which would be preferable, because that would require a V-bar final approach where the possible drift in case of miss can be less predictable or observable. In 4 a non docking collision would more likely put the customer in a drifting lower track. Moreover, once the refueling operations are finished a simple detachment with minimum robotic arm motion can be performed and the customer satellite can safely escape from any possible collisions. A V-bar approach can still be implemented in the final part if the robotic arm has not be designed or is unable to absorb the equivalent velocity variation of the customer.

3.2 Attitude guidance

In order to control the attitude of a satellite it is necessary to know the current satellite attitude, the reference attitude and a proper control law that steers the satellite in the chosen direction. Let us divide the dynamical system of the attitude in two loops, one for the attitude and one for the angular velocity. The guidance control determines the velocity reference that, if fol-

lowed perfectly, would set the attitude pointing error to zero. In this session a Lyapunov based quaternion guidance is presented. Its goal is to determine the reference angular velocity $\boldsymbol{\omega}_{br}$ such that the attitude of the satellite follows a certain reference. Using quaternions for the attitude parametrization means that the quaternion error $\mathbf{q}_e$, obtained from the satellite quaternion $\mathbf{q}_b$ and the reference attitude quaternion $\mathbf{q}_r$, should tend to the null rotation quaternion. A brief reference on quaternions for attitude parametrization can be found in Appendix A.

3.2.1 The attitude pointing error

Given the measured quaternion representing the attitude of the satellite with respect a preferred inertial frame $\mathbf{q}_b$ and the desired attitude quaternion $\mathbf{q}_r$, the error can be easily expressed as

$$\mathbf{q}_e = \left[\mathbf{q}_{b\otimes}^{+}\right]^T \mathbf{q}_r = \left[\mathbf{q}_{r\otimes}^{+T}\right] \mathbf{q}_b \tag{3.6}$$

and its variation in time can be expressed in terms of angular velocities as follows

$$\dot{\mathbf{q}}_e = \frac{1}{2}\left[\mathbf{q}_{e\otimes}^{+}\right] \begin{Bmatrix} \boldsymbol{\omega}_r \\ 0 \end{Bmatrix} + \frac{1}{2}\left[\mathbf{q}_{e\otimes}^{+T}\right] \begin{Bmatrix} \boldsymbol{\omega}_b \\ 0 \end{Bmatrix} \tag{3.7}$$

where it has been applied the derivative rule of quaternions and some permutations. $\boldsymbol{\omega}_r$ is the reference angular velocity and $\boldsymbol{\omega}_b$ the angular velocity of the satellite in body axes. Now let $\mathbf{q}_e = \begin{Bmatrix} \boldsymbol{\eta}^T & \epsilon \end{Bmatrix}^T$ and express the error quaternion in the two components. The unitarity condition must hold and is expressed as $\boldsymbol{\eta}^T\boldsymbol{\eta} + \epsilon^2 = 1$. Eq. (3.7) can be written in terms of $\boldsymbol{\eta}$ and ϵ as follows

$$\begin{cases} 2\dot{\boldsymbol{\eta}} & = \boldsymbol{\eta} \times (\boldsymbol{\omega}_b + \boldsymbol{\omega}_r) + \epsilon\,(\boldsymbol{\omega}_r - \boldsymbol{\omega}_b) \\ 2\dot{\epsilon} & = \boldsymbol{\eta}^T (\boldsymbol{\omega}_b - \boldsymbol{\omega}_r) \end{cases} \tag{3.8}$$

The system has two non-trivial equilibrium solution families:

- $\boldsymbol{\omega}_b = \boldsymbol{\omega}_r$ with $\boldsymbol{\eta} = \mathbf{0}$ and consequently $\epsilon = \pm 1$;
- $\boldsymbol{\omega}_b = \boldsymbol{\omega}_r \parallel \boldsymbol{\eta}$ and $\epsilon = 0$ (rather unstable).

For any solution there exist two specular equilibrium points as a result to use quaternions for the attitude parametrization and without proper care this can lead to unwanted phenomena leading to extra effort and delays.

3.2.2 Reference velocity for static attitude reference

Let us first consider the case where $\boldsymbol{\omega}_r = \mathbf{0}$. Eq. (3.8) simplifies to

$$\begin{cases} 2\dot{\boldsymbol{\eta}} & = \boldsymbol{\eta} \times \boldsymbol{\omega}_b + \epsilon \boldsymbol{\omega}_b \\ 2\dot{\epsilon} & = \boldsymbol{\eta}^T \boldsymbol{\omega}_b \end{cases} \tag{3.9}$$

Any condition with $\boldsymbol{\omega}_b = \mathbf{0}$ is an equilibrium point of the system regardless of the attitude orientation. To reach the target condition of $\boldsymbol{\eta} = \mathbf{0}$ and $\epsilon = \pm 1$ the reference velocity should depend on $\boldsymbol{\eta}$ and $|\epsilon|$.

Consider the following Lyapunov candidate function

$$\mathfrak{L} = \boldsymbol{\eta}^T \boldsymbol{\eta} = 1 - \epsilon^2 \tag{3.10}$$

that is null at the desired target condition and always positive otherwise. Its derivative with respect to time is

$$\frac{d}{dt}\mathfrak{L} = -2\epsilon\dot{\epsilon} = -\epsilon \boldsymbol{\eta}^T \boldsymbol{\omega}_b \tag{3.11}$$

The simplest $\boldsymbol{\omega}_{br}$ to drive the derivative of the Lyapunov candidate function to be negative definite would be

$$\boldsymbol{\omega}_{br} = \mathbf{K}\boldsymbol{\eta} \tag{3.12}$$

When $\boldsymbol{\omega}_b \to \boldsymbol{\omega}_{br}$ the derivative assumes the following expression

$$\frac{d}{dt}\mathfrak{L} = -\epsilon \boldsymbol{\eta}^T \mathbf{K} \boldsymbol{\eta} \tag{3.13}$$

That is negative semi-definite only for $\epsilon_{\mathbf{K}}$ positive definite. Then, and only then, the system would be stable and tends toward the desired attitude with exponential convergence rate; however, if $\mathbf{K}$ has no dependencies on the sign of ϵ then there might be cases where the sign is reversed and stability is locally lost: the consequence is the phenomena called *unwinding*. In [36] the problem is dealt with more insight and considerations are made on the impossibility to have continuous global stabilization for rotations due to topological limitations. Despite the theoretical relevance of the issue, in practice no controller is actually continuous, hence in practice rougher solutions are adopted. One potential solution to the issue is looking at hybrid systems [34]. To solve this issue a common practice is to incorporate a sign function in $\mathbf{K}$. However, such sign function has to deal with the case $\epsilon = 0$, that represents a 180° angle error where both clockwise and counter clockwise rotations can result in the same but opposite path. Then, since

the attitude does not depend on any geometrical or dynamical variable, it is safe to assume that using a scalar valued $\mathbf{K}$ would suffice. Hence the simplest guidance law would be

$$\boldsymbol{\omega}_{br} = -\beta\varphi(\epsilon)\,\boldsymbol{\eta} \tag{3.14}$$

with $\beta > 0$ and the classical implementation of $\varphi(\epsilon)$ as a modified sign function $\mathsf{s}(\epsilon)$

$$\varphi(\epsilon) = \mathsf{s}(\epsilon) = \begin{cases} 1 & \epsilon_{\geq} 0 \\ -1 & \epsilon_{<} 0 \end{cases} \tag{3.15}$$

Then

$$\frac{d}{dt}\mathfrak{L} = -\epsilon\beta\mathsf{s}(\epsilon)\,\boldsymbol{\eta}^T\boldsymbol{\eta} = -\epsilon\beta\mathsf{s}(\epsilon)\left(1-\epsilon^2\right) \leq 0 \qquad \forall(\boldsymbol{\eta},\,\epsilon) \tag{3.16}$$

The parameter β can be chosen by looking at the closed loop linearised system. Close to the equilibrium we have that the quaternion can be approximated as $\mathbf{q}_b \simeq \left\{\frac{1}{2}\boldsymbol{\vartheta}^T \quad 1\right\}$ and its derivative $\dot{\mathbf{q}}_b \simeq \left\{\frac{1}{2}\dot{\boldsymbol{\vartheta}}^T \quad 1\right\}$ with $\dot{\boldsymbol{\vartheta}} \simeq \boldsymbol{\omega}_b$. Hence the closed loop system is a low pass filter for the reference with a cutoff frequency equal to β.

Remarks on unwinding and chattering

It is important to notice that $\varphi(\epsilon) = \mathsf{s}(\epsilon)$ is not the only possibility, it is just the first that has been developed and implemented capable of offering Lyapunov stability and equilibrium at $\epsilon = \pm 1$. One could use $\varphi(\epsilon) = \epsilon$ and attain the same stability with a gentler action when the error is high, however the unwanted solution manifold $\epsilon = 0$ is a reachable but unstable solution, thus requiring an ad-hoc set-point insertion. The problem with the presented solution is the possible chattering when starting with $\epsilon_e \simeq 0$ as error in the error estimation might lead to keep the error in that neighbourhood. This lead to the proposition in [34] of a switching discrete function that alternates between -1 and 1

$$\varphi(\cdot) = h_k \tag{3.17}$$

with update that depends on the product $(\epsilon_k h_k)$

$$h_{k+1} = \begin{cases} h_k & \epsilon_k h_k \geq -\delta \\ -h_k & \epsilon_k h_k < -\delta \end{cases} \tag{3.18}$$

with a threshold δ such that $\delta \in (0,\, 1)$ and with initial condition $h_0 = 1$. The parameter δ set the behaviour of the error as increasing the noise rejection would increase the chances and extent of unwinding phenomena. The update law of [34] has been designed for hybrid systems and did not address the full discrete controller design, but the adaptation is straightforward. The premise of the work was based on a previous work about disconnected sets, hence topological problem easily relatable to the redundancy of the quaternion and sign invariance. Peering more in the depths of the argument, one might search for the relation between noise frequency spectrum and the unwanted chattering of $\mathsf{s}(\epsilon)$ as low frequency errors, with respect to the controller sample time, are arguably able to produce chattering or keep the system in the initial error state. The guidance law of [34] does provide better convergence and noise rejection when $\epsilon_e \simeq 0$ and the estimation of the error quaternion has significant high frequency components, otherwise the difference is negligible.

3.2.3 Time varying attitude reference

In many circumstances for OOS scenarios the servicing satellite has to follow a target that is moving, hence the set-point guidance is no longer satisfactory as a zero tracking error is never reached. Using Lyapunov theory again we can determine the modification to the set-point control. Let us consider the same Lyapunov candidate function (3.10) and consider (3.8) for the derivative of the candidate function.

$$\frac{d}{dt}\mathfrak{L} = -2\epsilon\dot{\epsilon} = -\epsilon\boldsymbol{\eta}^T\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_r\right) \tag{3.19}$$

Then for the same reasoning applied before we can achieve exponential stability with a velocity reference such as

$$\boldsymbol{\omega}_{br} = \beta\mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} + \boldsymbol{\omega}_r \tag{3.20}$$

In fact if we impose the body velocity $\boldsymbol{\omega}_b$ to be equal to $\boldsymbol{\omega}_{br}$ we get the derivative of the Lyapunov function to be again

$$\frac{d}{dt}\mathfrak{L} = -\beta\mathsf{s}\left(\epsilon\right)\epsilon\boldsymbol{\eta}^T\boldsymbol{\eta} \leq 0 \qquad \forall\left(\boldsymbol{\eta},\, \epsilon\right) \tag{3.21}$$

The derivative of the Lyapunov function can be null for the equilibrium point or on the manifold of uncertainty with $\epsilon = 0$. In the latter case the derivative would be null, but can be shown that the second derivative in that

neighbourhood is negative and non zero. In conclusion, to achieve exponential global asymptotically stability[1] the velocity reference to be tracked by the controller of the velocity loop is composed by a term proportional to the attitude error and a derivative term on the reference signal. Should be noted that such term should not be computed with a mere numerical derivative of the reference since a set-point input, assimilable to a step function, would cause an undesired impulse.

3.3 Relative pointing control

During an inspection manoeuvre the servicing satellite should be able to keep the serviced satellite in view of its instruments while performing a relative orbit. On the attitude control this translates to a single vector pointing, thus allowing virtually one single degree of freedom that can be exploited for solar panel pointing, communication or other tasks. In order to keep the controller structure unchanged from the previous case a proper reference signal has to be derived in terms of quaternions. Given the target vector in body reference frame $\mathbf{v}_{b,r}$ and the measured vector $\mathbf{v}_{b,m}$ in the same reference frame, the quaternion error can be written using the following procedure

$$\begin{cases} \vartheta = \frac{1}{2}\text{acos}\left(\mathbf{v}_{b,r}^T \mathbf{v}_{b,m}\right) \\ \mathbf{e} = \frac{\mathbf{v}_{b,r} \times \mathbf{v}_{b,m}}{\left\| \mathbf{v}_{b,r} \times \mathbf{v}_{b,m} \right\|} \end{cases} \tag{3.22}$$

Then the error quaternion is

$$\begin{cases} \boldsymbol{\eta}_e &= \mathbf{e} \sin\left(\frac{\vartheta}{2}\right) \\ \epsilon_e &= \cos\left(\frac{\vartheta}{2}\right) \end{cases} \tag{3.23}$$

then, since this quaternion is expressed in body reference frame, the actual quaternion reference shall be computed as

$$\mathbf{q}_r = \left[\begin{Bmatrix} \boldsymbol{\eta}_e \\ \epsilon_e \end{Bmatrix}_{\otimes}^{-} \right] \mathbf{q}_b \tag{3.24}$$

Should be noted that the reference quaternion needs to have frequency content below the reference tracking cutoff frequency of the controller to be fully tracked.

[1] In [34] issues regarding the global properties have been exposed. Theoretically speaking the above guidance law is not global due to topological restrictions, however in practical applications such restrictions do not hinder the controller

3.3.1 Pointing

Assuming a planar inspection orbit with $y_c = 0$ it can be shown that the deputy satellite on that orbit, to point always at the serviced satellite needs to have the following angular velocity

$$\omega = -\frac{4n}{(3\cos(2nt)+5)} + n \tag{3.25}$$

That is the composition of the nominal orbital motion and the LVLH relative angular velocity tailored on the inspection orbit. It is important to notice that the angular velocity does not depend on the inspection orbit size but only on the altitude of the serviced orbit. Integrating directly Eq. (3.25) gives

$$\vartheta(t) = nt + \text{atan}(2\cot(nt)) + \vartheta_0$$

Eq. (3.25) gives the angular accelerations

$$\dot{\omega} = -\frac{24n^2\sin(2nt)}{(3\cos(2nt)+5)^2} \tag{3.26}$$

this allows to estimate the maximum torque required for the inspection pointing, located at $\cos(2nt^*) = \pm\frac{5-\sqrt{97}}{6}$

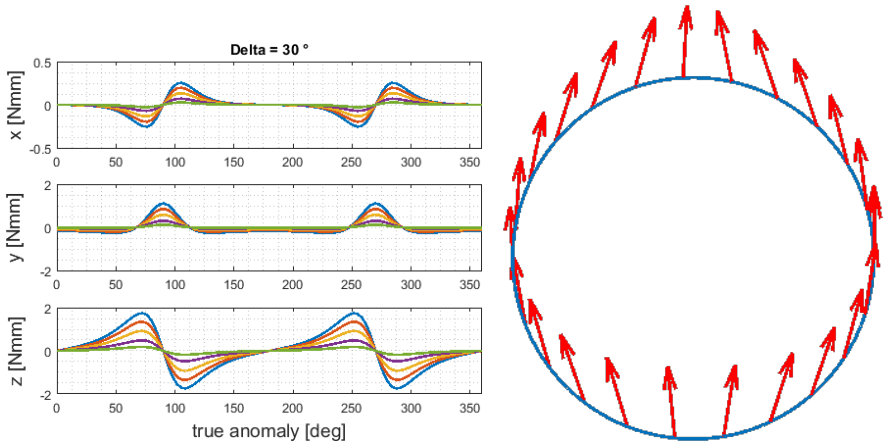
$$\tau_{max} \simeq 2.1\boldsymbol{I}n^2 \tag{3.27}$$

That is usually a small amount of torque required, however the critical part is the frequency spectrum of the reference signal. Looking at the spectrum of the term atan $(2\cot(nt))$ it is possible to notice that has some relatively high frequency content that is not easily followed by any attitude control. Hence, some pointing error is to be expected in any case and the magnitude depends on the altitude of the serviced orbit, the higher the lower such error would be.

3.3.2 Considerations on Inspection angular velocities

Here an inspection manoeuvre has been implemented with focus on the pointing demand on actuators. The goal is to observe the customer satellite with one or more cameras in a elliptical relative orbit, thus allowing for mapping and damage inspection. A view of angular velocities in time is presented in order to appreciate the torques required to maintain pointing.

Figure 3.3a shows that no constant torques need to be provided thus no wheel saturation is foreseen. The maximum torque requested is function



(a) *Theoretical torques for optimal pointing at different or-* **(b)** *Pointing in inertial frame*
bits

Figure 3.3: *Pointing and angular velocities for inspection*

of the nominal orbital angular velocity, thus for GEO satellite inspection, the torque requested is minimal with respect to LEO inspection. On the other hand the duration of a single orbit is much higher, thus the inspection operations might take several days. This also hints that for LEO a more agile satellite is preferred and that is unlikely to use the same servicing satellite class for both missions.

Figure 3.3b presents the pointing direction of the servicer for a planar circular orbit in a Earth centred inertial frame. This shows the nature of the attitude motion for inspection with several swing in opposite directions. This helps to understand the variation of attitude during a single orbit, thus permitting to assess the possible light variation during the inspection. In the worst case for GEO inspection the variation of light direction in the Earth centred frame is one degree, thus almost negligible. The variation in attitude of the servicer is bounded thus no risk of abrupt light changes. Changes will be present due to the main rotation, however the effect can be minimized.

3.4 Dual Quaternion velocity guidance

Finally, consider the problem of positioning a body with respect to another body or frame. Let the kinematics be represented by dual quaternions, where $\boldsymbol{a}_b$ represent position and attitude of the body, $\boldsymbol{a}_r$ the reference position and attitude and the error $\boldsymbol{a}_e$ is defined through the following relation

$$\boldsymbol{a}_r = \left[\boldsymbol{a}_{e_{\epsilon\times}^{-}}\right] \boldsymbol{a}_b \tag{3.28}$$

Of course form this relation is clear that $\boldsymbol{a}_e$ is the error expressed in the body reference frame, while with $\boldsymbol{a}_r = \left[\boldsymbol{a}_{e_{\epsilon\times}^{+}}\right] \boldsymbol{a}_b$ would have been expressed in the inertial frame. A guidance law would need to define the linear and angular velocities required to reduce the error dual quaternion to the identity. Consider the following Lyapunov function

$$\mathfrak{L} = \left(1 - \epsilon^2\right) + \frac{1}{2}\boldsymbol{d}_e^T \boldsymbol{d}_e \tag{3.29}$$

where ϵ is the error quaternion scalar component and $\boldsymbol{d}_e = \boldsymbol{d}_r - \boldsymbol{d}_b$ the position error in the inertial frame. Should be noted that considering $\boldsymbol{d}_e$ the displacement of $\boldsymbol{a}_e$ would have lead to the same $\mathfrak{L}$ as the scalar valued function does not depend on the attitude of the two reference frames, being the norm of a 3D vector symmetric to pure rotations.

By differentiating one obtains

$$\frac{d}{dt}\mathfrak{L} = -\left\{\epsilon\boldsymbol{\eta}^T \quad \boldsymbol{d}_e^T\right\} \begin{Bmatrix}\boldsymbol{\omega}_r \\ \boldsymbol{v}_r\end{Bmatrix} \tag{3.30}$$

Thus with the desired velocities

$$\begin{cases}\boldsymbol{\omega}_r &= \mathbf{K}_\omega \mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} \\ \boldsymbol{v}_r &= \mathbf{K}_d \boldsymbol{d}_e\end{cases} \tag{3.31}$$

exponential convergence should be guaranteed. A discontinuous $\mathsf{s}\left(\epsilon\right)$ would theoretically pose problems to draw global stability claims as stated before.

CHAPTER 4

Navigation

In this chapter solutions to the determination of a satellite attitude as well as the relative state of a target satellite are proposed. First, some technique used for vision based estimation will be addressed. A lot of research is involved in developing a vision based algorithm that is capable of determining the relative pose of a satellite. This is crucial for ADR implementations.

Vision based angle only navigation for rendezvous using Kalman filtering has been used in [30] although observability using only angles is an issue that has been lifted by the same author few years later in [37], not to mention the general difficulties of dealing with quaternions in a Kalman filter [38].

Computer vision algorithm for navigation can assume in general three different forms: image stream, template/map/model matching or target identification.

An image stream has the drawback that diverges over time, but this can be contained using model based filtering or other sensors. The use of complementary sensors for vision based navigation has been seen by many as a possible solution to increase old sensor suite performance, however the precision attainable by vision based system might be higher. Anyway, camera aided Inertial Navigation System (INS) have become a popular topic in the

recent years [39–45]. One effective way to correct INS is to use an Indirect Kalman Filter [46], tailored on the INS error, and exploiting the rigid transformation between two camera views [43]. Looking at the results obtained in [43] it is speculated that the camera alone reconstruction might prove lighter while slightly less precise w.r.t. the INS filtered version. Most of Camera aided INS method augment the state with measured features, that is always computationally expensive. In ego motion about one hundred of tracked points are enough, therefore the already high dimension state is often augmented with twice or three times the number of feature found in the images [42, 47–49] getting few hundreds of variables. Given recent advancement in vision based navigation the most relevant issue to be solved are robustness to light variation and enhancing the estimation data-rate to be able to reach required control frequency to compensate expected disturbs. Another time-related issue is possible delays between image acquisition and information utilization, some research on this issue with experimental activities can be found in [50].

Template matching for proximity operations using cameras has been studied in the past. For example [26] has implemented a 3D matching using non linear minimizers to solve the pose (see Appendix C), although images were not simulated and the computer vision part has been neglected. A more recent and comprehensive reference with experimental validation can be found in [51] where 3D model matching flourish within a visual servoing approach. Further development on exploiting the maximum return from images can be found in [52] while the basis for 3D matching have been posed more than two decades ago, the interested reader should refer to [53].

In earth robot navigation in an unknown environment the *Simultaneous Localization And Mapping* (SLAM in short) [54, 55] is often employed and in this approach the map is formed through repeated observations. In this case convergence is attained through multiple view of the same zone. Partial mapping exploited through high framerate might still be effective but a proper assessment of hardware impact has to be performed, in fact this approach should not be able to attain the same convergence of a 3D reference map/model. When mapping approaches are considered another important tool is *Bundle Adjustment* [56] where also cameras intrinsic parameters are updated through time. This will become increasingly necessary in OOS applications due to the long lifetime and harsh environment foreseen for cameras on servicing satellites that also require good precision.

Some research groups are working on the use of vision based relative state estimation based on known features on the target satellite, for example

in [57, 58] matching is done with a rectangular shape, while in [27] the adapter ring is exploited. A more sophisticated and robust implementation of similar principles is found in [59]. The latter approach is promising enough to be partially adopted here.

4.1 Vision based navigation

Given two consecutive images from the same camera there are two possible way of computing the relative movement: feature tracking and optical flow. The first method requires similar feature points to be extracted from both images and then matched to determine the rigid transformation between the two views. The quality of the estimation increases with the number of features and matching accuracy process and this procedure is capable of addressing wide motion. Optical flow is based on the images intensity differentiation to be able to determine the motion. The great limitation of the latter is the weakness to large displacement.

Feature tracking requires to detect features of the same kind in two images and then apply a matching procedure, typically a *Normalized Cross Correlation* (NCC) [60] between the neighbourhood areas of the feature points in both images [60, 61]. The pairs are then fed to an algorithm that determines the rigid transformation. Some methods, like *RANdom SAmple Consensus* (RANSAC) [62] or *eight point algorithm* [63], make use of the epipolar constraint [63–65] determining a combination of rotation and translation direction, however they can be computationally expensive and give multiple and/or inaccurate results.

Feature selection and matching is a costly operation and a proper analysis has to be carried out to decide which features provide enough points and in the shortest time available. Features can be divided roughly in low level (edges, corners) and high level (ex: landmarks). High level feature are usually detected starting from low level feature, making the process computationally more expensive and requiring a priori knowledge of the terrain. In the last decade the new concept of region based feature has been studied. *Scale Invariant Feature Transform* (SIFT in short) [66] and *Speeded Up Robust Features* (SURF) [67, 68] lie within this category. Basically the principle is to take interesting points such as corners and identifying those with better probability to be traced in the other image.

4.1.1 Stereo relative positioning

With a single camera it is possible to identify an object in space and extrapolate its direction. For inspection purposes this might be enough, but some

problem arise. First of all with only angle measurement is in general not possible to obtain the full state [37] and some additional image processing must be considered, since other object might be seen by the camera, like Earth in LEO scenarios. On the other hand it might be wise to have a redundancy, thus two camera on board for proximity operations. Hence considering a relative positioning with a stereo camera pair might be a wise solution.

When a stereo pair is considered two possible implementations might be enforced: consider the two camera as one sensor and perform a calibration or consider them separately and perform matching operations between the two views. The second possibility might be computationally heavier but permits to easily filter out objects that are outside the considered range. Here this approach is followed. The images of the stereo pair are passed through a SURF [67,68] detector and extractor, then matched through NCC [60]. The outlier matches are filtered out in two stages, first applying a RANSAC [62] run, then through baseline (akin to epipolar constraint) and finally through triangulation a small point cloud is obtained.

Due to parallax and limited baseline, Earth would appear equally placed in the two images, thus the displacement of features in the stereo cameras would be null with almost perfect matching. Figure 4.1 exemplifies the fact that background features can be easily spotted and subtracted from the computation as they appear in the same place for both cameras. In proximity operations, say between 100 and 20 meters, there is no need to use pose estimation to determine attitude and absolute relative positioning, thus a simple centroid estimation would suffice. Figure 4.2 presents the estimation from the simulated image example. It is possible to appreciate that a good estimation of the centroid is possible. Due to the several combinations of parameters for feature extraction, matching and filtering, it is clear that sometimes the estimation may be off if some features are not extracted. Looking at the image will be clear that if a feature on the solar panels is detected then the centroid would move off from the center of mass. The resulting measure is thus affected by high frequency noise as from frame to frame it is not recovered the same centroid position but in lower frequency this shift has less importance. One might argue that using feature based matching it is possible to match also across subsequent shots and it is absolutely true, however this can create a problem as two subsequent measurement would be connected and error accumulation impose a drift in the estimation, leading to target loss. Keeping the measurements completely uncorrelated allows to have a more robust estimation as errors do not accumulate over time.

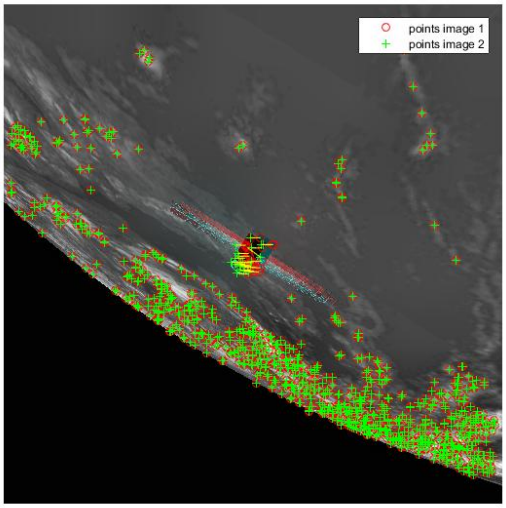


Figure 4.1: *Stereo matching*

Of course, it might be requested a filter in order to reduce the noise and limit high frequency oscillations for the pointing guidance.

4.1.2 Dual quaternion estimation from camera views

Dual quaternions can be used also to estimate the relative pose between two subsequent image frames [69]. The algorithm is based on the solution of the Wahba problem with quaternions, then the position quaternion is recovered afterwards. Should be noted that the algorithm requires one particular configuration of dual quaternions in order to exploit properties of the dual quaternion matrix forms. Regardless, it is always possible to revert to any formulation. Using stereo camera is always possible to match a 3D point cloud between two subsequent time instants and estimate the relative displacement. However, this procedure does not guarantee a good matching or estimation, as errors would still increase over time. In other words it is pretty much similar to a velocity estimator affected by noise.

The naive implementation of the process exhibits random walk that may be dangerous for operations, hence a mapping approach is often enforced to be able to reduce the drift. The problem of mapping and memory are relevant and well addressed in literature with some solution found satisfactory in some applications.

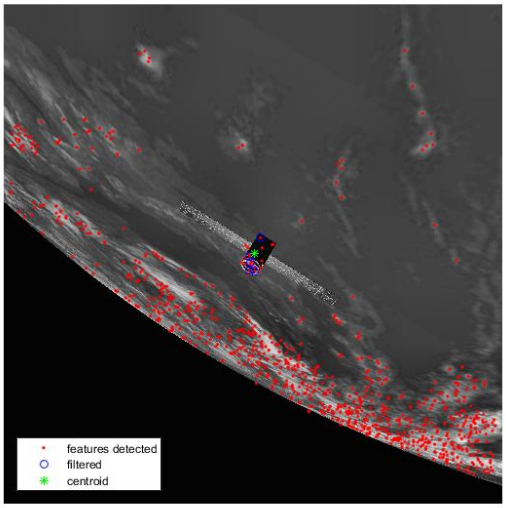
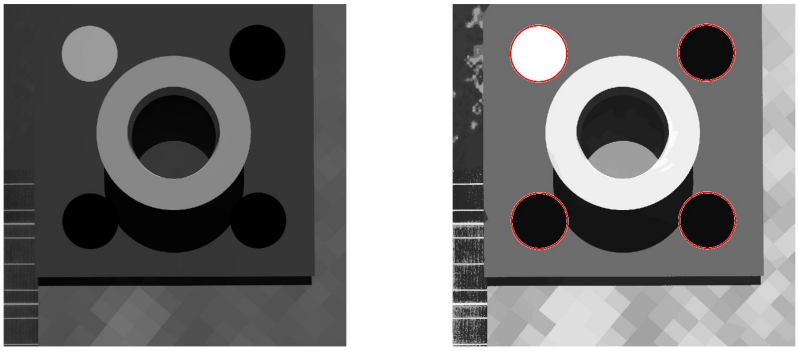


Figure 4.2: *Filtering and centroid estimation*

4.1.3 Reference matching

In many OOS scenarios that requires close proximity operations it is possible to exploit inter-satellite communication to at least receive an estimation of the target attitude in order to use template matching or shape based matching techniques that are able to eliminate the drifting problem. As a matter of fact the matching from a template and a model, intended as an ensemble of points or lines, gives back a punctual estimation of the relative position and attitude with a un-correlated error with respect previous estimates, which avoids the drift in the visual odometry. The estimation error in this framework can be of few millimetres at about 3 meters of distance and using very few points. See Chapter 7 for application results.

When markers of any kind are available on the target, especially if the target port has been designed for servicing purposes, it is possible to estimate precisely the port location with respect the currently computed relative pose as some inaccuracies might be possible. The focus of this thesis is not on visual navigation but rather the combination of visual navigation with other GNCR subsystems, hence the registration of images and the use of templates is rather simple.



(a) *Original image* **(b)** *Processed image with found markers*

Figure 4.3: *Marker detection*

Robotic arm docking port

Since camera mounted on a robotic arm tip cannot provide measures up-close the camera is instead used to refine the relative port displacement from the estimated target satellite pose. Through chain of transformation the measures can be used to estimate the port location on the target satellite.

The markers used are four circles of two different colors (white and black) on an intermediate background. Using Hough transform and a initially estimated distance from the target it will find circles in a neighbourhood of the estimated circle diameter. Figure 4.3 presents the original simulated image and the processed image. In the latter the image histogram has been adjusted adaptively and in red the circles match perfectly the marker.

The search conducted without pre-estimation will be more cumbersome and prone to error, so robustness of the estimate relies on the constraint given to the error of previously estimated poses. Once all four circle position are computed in camera frame, the relative attitude is computed using four directions of the markers with respect to their center and the normal of the plane identified by the 3D positions of the markers in the model reference of the target satellite. The 3D position of the markers is recovered by using the radius of the circle found in the images as scale factor, thus the estimation becomes really fast.

Figure 4.4 shows the robustness of the method to occlusion, meaning that if part of the marker is not seen by the camera or is covered it can still be found by the algorithm. On the other hand Figure 4.5 shows that with high error in depth estimation leads to a difficulties in finding the circles of

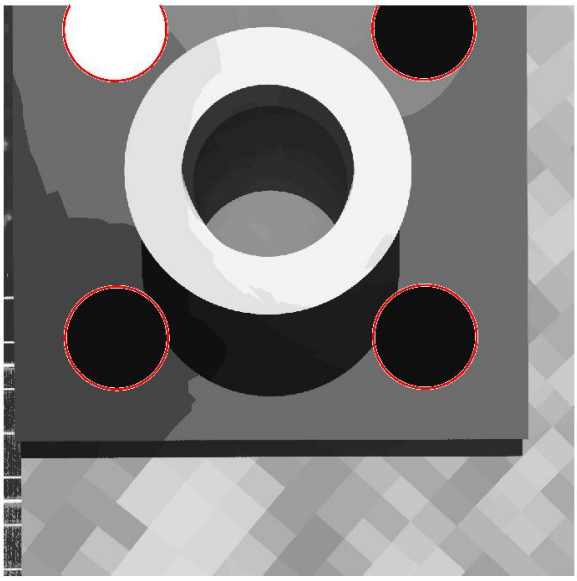
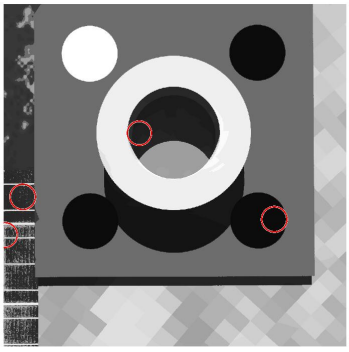
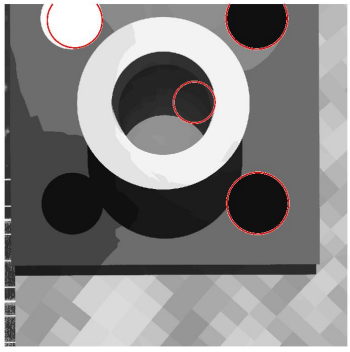


Figure 4.4: *Robustness to occlusion*



(a) *Wrong depth*



(b) *Wide range of radius search*

Figure 4.5: *Mismatch*

the correct size, and that using a wide range of radius search might lead to inaccuracies.

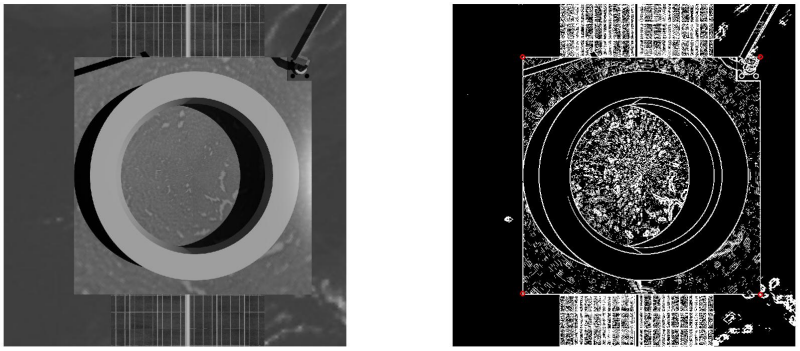
The final estimation is performed by a weighted mean on the minimal representation of the estimated dual quaternions. The weights are computed as the overall fitting error of the four markers while the minimal representation links the dual quaternion to the Cartesian position vector and the Euler axis and angle. To reduce potential errors in the estimated dual quaternions, the attitude estimation is computed with respect the first port estimate of the batch as in some cases the Euler angle can be 180 degrees. The finalization of the port location requires the port to be visible, hence the expected error is small and any simplification made in the final estimation is justified.

Satellite relative state

On the other hand, for the pose estimation of the whole satellite relative position and attitude it has been used a slightly different approach. Knowing the rough outer appearance of the target, a Hough transform has been used to determine lines and edges of the target satellite. First the image is processed with a Sobel edge detector becoming a binary image where lines are easily found with the Hough transform. In Figure 4.6 the whole process and the results are displayed. From the Hough transform four lines are extracted, two vertical and two horizontal, within a $\pm 5^{\circ}$ domain. When dealing with outer edges detection and Hough transform with space object, light issues and shadows might reduce the robustness. Even in [59] the issue is addressed and tracked lines are found in a subspace of the whole search space. The simplified idea of tracking a relatively known rectangular shape has also been applied experimentally with compatible results in [70] although with different premises and finite reliability.

From the four lines tracked in this implementation are then computed the four corner points in the image. Using a non linear minimizer the depth of the features is determined minimizing the projection error. Displacement and attitude are computed using a combination of the dual quaternion estimation of [69] and the slightly older [71] used for points sets. The estimation of the relative dual quaternion is direct and analytical when depth are estimated, thus for each iteration of the minimizer the depths are used to construct the 3D projection of the features found in the image and from there the error.

The limits of search for lines, as well as the search for maxima in the Hough space, needs to be tailored as many disturbs might be added. For instance consider Fig 4.6a where the shadow of the robotic arm projects a straight line on the serviced satellite face, in Figure 4.7 a possible bad



(a) *Original Image* **(b)** *Processed image and corners found*

Figure 4.6: *Satellite corner determination*

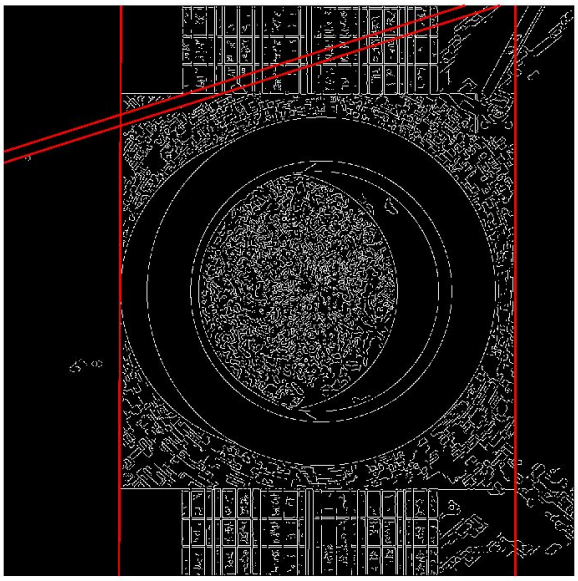


Figure 4.7: *Example of mismatch due to shadows*

interpretation of the image due to shadows is presented. Even with those precautions the estimation might fail, for example if the line that has been found is on the solar panel instead of the base or if some other line with stronger image gradient is found. Hence, as safety factor, the previous line estimation is used when the criteria are not found in the image without having to change the parameters, as this might happen just in few images.

Light variation issues

Light variations can have dramatic impact on the cameras, as it is possible to see from Figure 4.8. When in eclipse the servicer should switch on lights in order to see the other satellite and although this burdens batteries, it is not the worst situation as no shadows are cast and the background is almost black. On the other hand, having the sun behind the servicer satellite cast shadows on the serviced satellite, altering the image histogram and creating troubles in the segmentation process. It is indeed possible to adjust parameters ad hoc for the situation when operations are really performed in orbit. Another approach could be simply to schedule the operations when the sun is in favourite directions, including the variations due to the rotation of the two satellites if nadir pointing. Another source of errors, as shown before, is the presence of other objects, like a robotic arm, casting shadows that can be interpreted wrongly by the software. Moreover, on highly inclined orbits the sun could generate some self shadows on the servicer satellite that covers important areas for the segmentation if not a docking port itself. It is not difficult to predict such behaviour as to some extent it is possible to simulate images, as done in the development of this research, but creating a truly robust vision based relative state estimation capable of adjusting itself without much effort from the ground is still an issue.

Hardware limitations

Image processing involves heavy computations and this can hinder the use of vision sensors in space where space proven hardware are generally few generations back with respect to Earth-borne applications. Even more so, the use of cameras in a closed loop control requires a steady sample time that cannot be lower than a minimum threshold. However, recent development in space hardware for lunar landing, planetary exploration and satellite to satellite applications permits to state that such concerns are no longer relevant. Many of the computer vision applications like feature extraction and matching can be implemented using Field Programmable Gate Array (FPGA) and run in parallel at medium/high sampling time. In [72] is

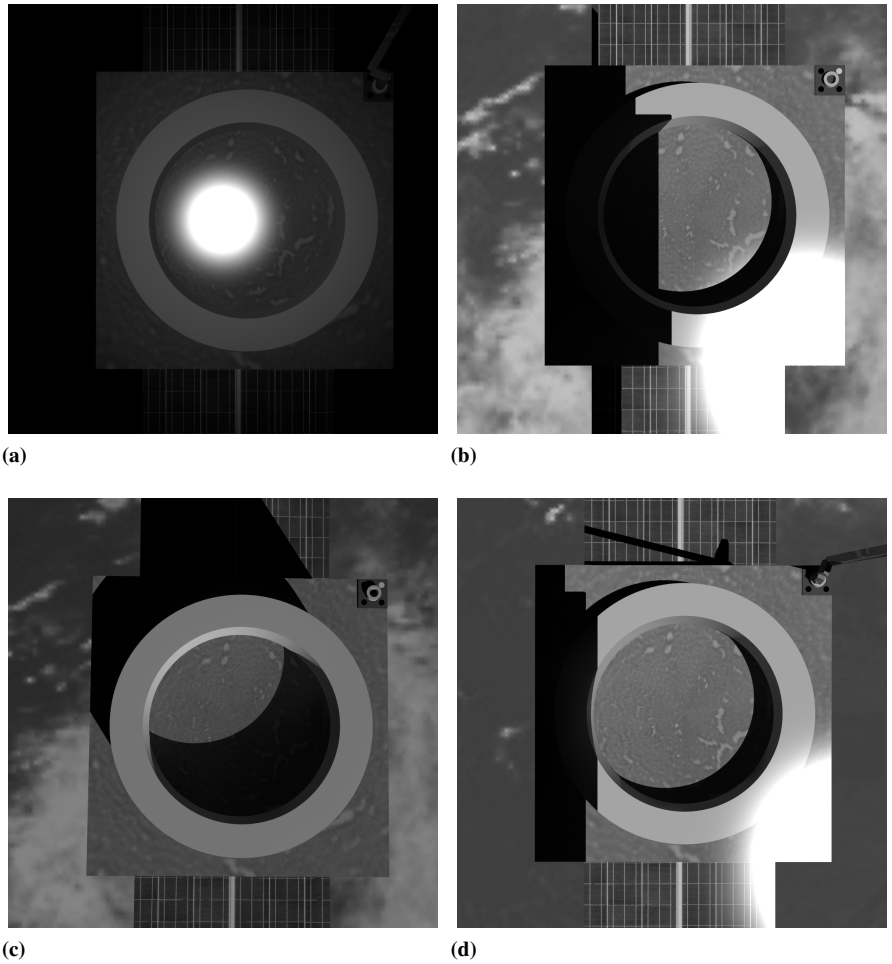


Figure 4.8: *Relative sun direction impact on images*

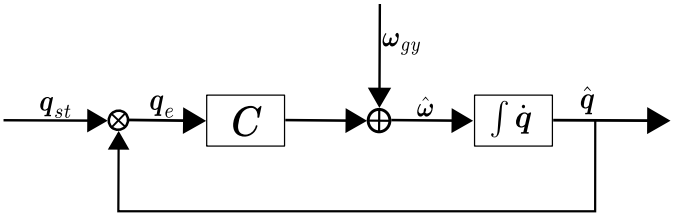


Figure 4.9: *Quaternion Complementary filter scheme*

presented a chip for monocular vision with lunar landing applications capable of reaching 20 fps with included feature extraction and matching (up to 200) of a 1024 x 1024 pixels image. Looking at Italian development in the field we can cite [73] and the subsequent [74] with both monocular and stereo applications. Here the achievable framerate ranges from 10 to 30 fps and all equipment is space qualified and available from the supplier. The image processing applications for the GNCR here proposed can be hard coded in FPGA, but to be conservative the framerate here considered is bounded to 1 fps. This ensures that the required computations are expected to be delivered to the control system with real time constraints.

4.2 Complementary filtering

A complementary filter is a filter that weights in frequency two or more sensors information to refine a state estimate [75]. The simplicity and elegance of the approach can be used also for satellite attitude determination, for example fusing information from a gyroscope and a star tracker. In [75] proof is given for the convergence of the estimate and locally for gyro bias estimation (as Lyapunov theory cannot assure absolute convergence of the bias estimate) also using quaternions. In Fig 4.9 a block scheme of this approach for a simple satellite attitude control is presented. Here $\mathbf{q}_{st}$ and $\boldsymbol{\omega}_{gy}$ are respectively the measured quaternion from the star tracker and the gyro measured angular velocity. With $\hat{\mathbf{q}}$ and $\hat{\boldsymbol{\omega}}$ are indicated the filter estimated outputs and the quaternion error $\mathbf{q}_e$ is computed and fed to the filtering function C with sign consistency.

This filter has some interesting features:

- it is able to fuse measurement from two sensors at different data rates, providing an accurate estimation at high sampling frequency. Should be noted that some high frequency noise might be introduced with the straight implementation of the filter if the data rate difference is too high. Up-sampling might introduce excessive delay and down-

sampling would reduce the benefits of high frequency estimation;

- it uses a gyro measure to integrate the attitude that can give estimate even when other attitude sensors are switched off or unable to provide measurements (ex: eclipses);

- can have good gyro bias rejection (theoretically demonstrable only on linearised case);

- with respect to a full state Kalman filter it is much simpler and easily tunable with the proper sensor suite;

- provides a quaternion estimation with no sign jump since it is obtained from integration if C takes into account possible jumps in $\mathbf{q}_e$;

- Attitude measurement can be provided by quaternion or vector measurements.

In some aspects this attitude filter is not much different from an indirect Kalman filter [46] since the core principle is the same, but the tuning and updates may differ. If one wants to compare the two approaches, it should be clearer that the complementary filter of [75] has fewer parameters to be determined and the computations needed are way lower than those for a Kalman filter with iterative formulation. Of course, the drawback is that it requires good complementarity of the sensor suite. It can be demonstrated that a proportional gain in C gives convergence to the filter while adding an integrative action can compensate for low frequency errors in the gyroscope measurement, thus locally estimating its bias and random walk. The proportional term drives the cutoff frequency of the filter: below such frequency the star tracker measurements are weighted more while the low frequency component of the gyro integration are almost not considered.

The filter estimated values are used for attitude control, hence an higher sampling rate allows for higher control gain and possibility to track attitude reference with low frequency content. High frequency noise in $\hat{\mathbf{q}}$ or $\hat{\boldsymbol{\omega}}$ translates in higher control effort but if limited it is not going to jeopardize the attitude control robustness. If the star sensor measure is highly accurate, then the estimated quaternion of the filter might not reduce the error but the signal will have higher data-rate useful to track relatively fast signals, operation that can be impossible for a more slower estimation. This is even more true if an up-sampling procedure is implemented, since a delay in the signal will be incorporated.

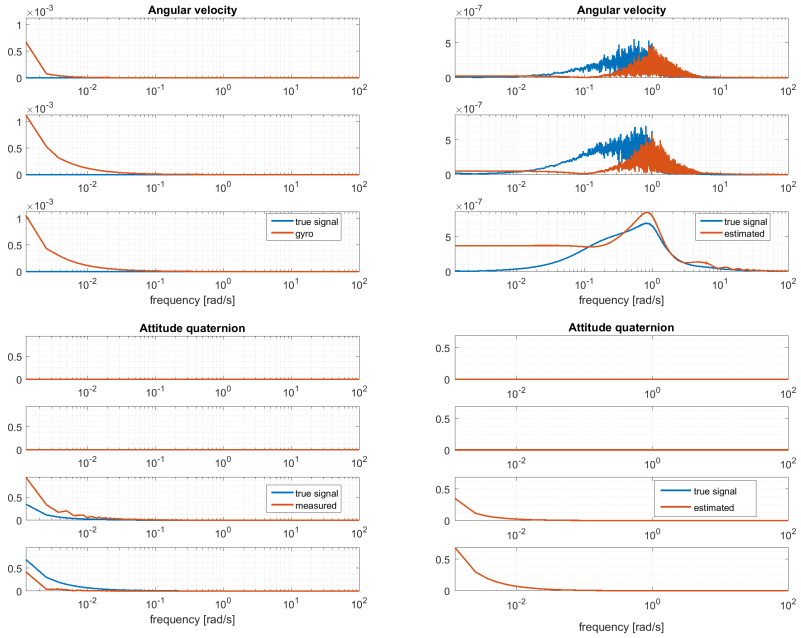


Figure 4.10: *Closed loop performance*

4.2.1 Closed loop

Another important issue to be addressed is the closed loop performances of such non linear estimator. Let us use an example to clarify the importance of such problem.

In Figure 4.10 are shown the results from a simulation where it has been set the control closed loop frequency to 0.1 rad/s and with a dynamical loop control frequency set to 1 rad/s. The satellite is requested to spin of an angular velocity equal to 0.001 rad/s around the z axis. From the FFT of the gyro it is clear that the dynamical loop control frequency is the most relevant in the true signal and that the modelled gyroscope has higher lower frequency error that is rejected by the complementary filter. In this case the cutoff frequency of the filter is set to 1 rad/s. Just by switching the filter frequency we have two different behaviour, in one case there is a resonance phenomena occurring between closed loop and dynamical frequency. For further insight, results can be found in application scenario in Chapter 7.

4.3 Relative position estimation

Relative position estimation is of utmost importance for any OOS operation, hence it is requested to use all possible sensors, based on the relative distance, to obtain the best estimation possible. Possible sensors are

- relative Lobal Positioning System (GPS), that requires an inter-satellite communication link to share the inertial position estimation from GPS;
- cameras and stereo cameras, that works only when the sun is not in the Field of View (FOV) and the subject is inside the FOV, and might work in eclipses in close distance with an artificial light;
- ranging sensors (radar/lidar) that are able to obtain relative distances and in rare case a 3D position is obtainable.

The relative dynamics of two satellite in a Keplerian orbit is quite well understood and state transition matrices are obtainable for the circular orbit case and for the elliptical orbit case, albeit the latter is more cumbersome to compute due to the dependency on Kepler time equation. To exploit such solutions the linear position and velocity need to be expressed in the LVLH frame and then can be transformed in any reference frame needed for control or other operations. For any orbit the discrete system is linear and potentially time dependent and can be expressed as follows

$$\begin{Bmatrix} \mathbf{v}_{k+1} \\ \mathbf{x}_{k+1} \end{Bmatrix} = \boldsymbol{\Phi}_{k+1,k} \begin{Bmatrix} \mathbf{v}_k \\ \mathbf{x}_k \end{Bmatrix} + \boldsymbol{B}_k \boldsymbol{u}_k \tag{4.1}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_k \begin{Bmatrix} \mathbf{v}_k \\ \mathbf{x}_k \end{Bmatrix} \tag{4.2}$$

where k is the discrete index of the discrete time, $\boldsymbol{\Phi}_{k+1,k}$ is the state transition matrix evaluated at the discrete system sample time that connects the state to its next value. The state is composed by velocity $\mathbf{v}$ and position $\mathbf{x}$, while $\boldsymbol{u}_k$ are the control forces and $\boldsymbol{y}_k$ the measures.$\boldsymbol{B}_k$ and $\boldsymbol{C}_k$ are the matrices connecting the control and measures to the state and its derivatives. Considering only Gaussian noise acting on measures and control forces we can develop a steady state Kalman Filter whose update is

$$\begin{Bmatrix} \mathbf{v}_{k+1,k+1} \\ \mathbf{x}_{k+1,k+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{v}_{k+1,k} \\ \mathbf{x}_{k+1,k} \end{Bmatrix} + \boldsymbol{K}_\infty \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}_{k,k}} \right) \tag{4.3}$$

which is an update on the foreseen projection of the state based on previous information by a constant matrix $\boldsymbol{K}_\infty$ weighting the residual of estimated measures versus actual input registered measures $\boldsymbol{y}_k$. As long as the system sample time is consistent, the matrices are constant (circular case) matrix $\boldsymbol{K}_\infty$ is constant. The constant Kalman gain $\boldsymbol{K}_\infty$ is of course the result of a Riccati matrix equation weighting the model uncertainties in form of covariance of white noises and the measures error covariance. Since the procedure is well known in literature here will be reported only few aspects connected to the issue in exam.

In $\boldsymbol{y}_k$ we consider relative GPS measurements and camera position estimation. As mentioned before there are some conditions where cameras might not work properly, hence the filter that is requested to estimate the state needs to operate even when camera measures are not available. Since the gain are considered constant it is possible to partition matrix $\boldsymbol{K}_\infty$ and add the contribution of cameras only when it is available. A test case of this principle is available in Chapter 7.

CHAPTER 5

Control

Here the word *control* will be used to represent the algorithms that determine the command to physical actuators in order to act on dynamical variables representing part of a mechanical system state. For example, in case of satellite attitude control the actions are torques and the dynamical variables are the angular velocities of the satellite. Usually the outer loop in position or attitude and the inner loop on velocities are grouped together, here the distinction has been made for explanatory purposes. In this chapter it will be derived an adaptive control law such that is able to track a reference velocity with uncertain or variable inertia parameters.

When dealing with control laws we can divide application from control schemes as framework can be used, although adapted, to different systems. Satellite attitude control goes back from the dawn of space era and in many cases a simple Proportional Integrative Derivative (PID) architecture on Euler angles was enough to achieve the required performance [76]. Later on, alongside new kinematic representations, the field flourished with research on more modern paradigm, although in industrial application PID control is often favoured for its simplicity and consequent robustness. In general, the more parameters have to be tuned, the brittle is the consistency of the controller and performance greatly change with minimal gain

variation. Regardless, in literature several other paradigms have been studied and applied, at least theoretically, to the problem of attitude control of satellites and similar vehicles.

Among these paradigm we can list *Linear Quadratic Regulators* (LQR) [77], *Input-Output Feedback Linearization*, *Backstepping* [78], *Adaptive control*s, H_∞/μ *control*s [79], *Model Predictive Controls* [80], *Sliding mode controls [81, 82], soft computing control* (neural networks, fuzzy systems, etc) and all their variations and combinations. LQR coupled with Kalman filters have been popular in the field, however in practical applications the strife was to bring back the LQR to a PID like control and in some extent it is another way to determine the PID weights. In that regard also structured H_∞control can be used to derive robust weights of a PID and has more success in the field with respect to the non-structured version. Sliding mode controllers have found more applications in controls where small thrusters are involved, as the actuators output cannot be modulated in amplitude but the chattering they induce might create problems in attitude control where higher precision is required. Input-Output Feedback Linearization techniques have been popular in the field as they provide a framework to eliminate nonlinearities and adopt stabilizing simple linear control laws through a non linear transformation of the system state. However, such controllers are heavily dependent on the accuracy of the system model and might be less robust to uncertainties.

An adaptive control has been derived in [83] and incorporates in the control the estimation of the full inertia matrix that may result in excessive computations, especially since in the adaptive framework the convergence to true values is not guaranteed unless persistent excitation is exerted. Moreover the estimation might be corrupted by measurement noises [84].

5.1 Quaternion based attitude control

OOS operations require a robust attitude control in order to guarantee the safety of all requested operations. Hence, a control law that guarantees stability under uncertainties and in different phases is mandatory. Among many possibilities here it has been chosen to derive an attitude control law using quaternions to avoid singularities and simplifying as much as possible the controller. The latter is due to the axiom “the simpler the robust” that has been chosen as guidelines for many application cases. PD-like controllers even with quaternions have been studied in the past decades [33] but some aspects have been overlooked and some other issues have been re-opened with respect to convergence properties.

Let us divide the attitude control in two parts: a kinematic part and a dynamic part or inner and outer loop if one prefers. The distinction comes handy since the kinematic loop has no uncertainties with respect to the dynamic loop that has uncertainties connected to the inertia of the satellite and the nature of disturbances acting on it. Then, let us focus on the dynamical loop and assume a general reference velocity $\boldsymbol{\omega}_{br}$ for the reference input. Later, such angular velocity will be equal to the reference guidance studied in Chapter 3.

The equation of motion for a satellite can be expressed by the introduced Euler equations rewritten as follows

$$\boldsymbol{I}_b\dot{\boldsymbol{\omega}}_b = -\boldsymbol{\omega}_b \times (\boldsymbol{I}\boldsymbol{\omega}_b) + \boldsymbol{\tau}_\mathbf{d} + \boldsymbol{\tau}_\mathbf{u} \tag{5.1}$$

where $\boldsymbol{I}_b$ is the 3 by 3 inertia tensor of the satellite expressed in body frame, $\boldsymbol{\tau}_\mathbf{d}$ is the vector of disturbing torques and $\boldsymbol{\tau}_\mathbf{u}$ is the vector of control torques. In absence of disturbs and control torques there are some cases where the system has no dynamical equilibrium thanks to the non linear term $\boldsymbol{\omega}_b \times (\boldsymbol{I}_b\boldsymbol{\omega}_b)$.

Analysing Eq. (5.1) permits to derive the control law such that the angular velocity of the satellite $\boldsymbol{\omega}_b$ follows a reference angular velocity $\boldsymbol{\omega}_{br}$ that can assume a constant value, a null value or a time varying value depending on the attitude control mode of the satellite.

5.1.1 Null velocity reference

In satellite attitude control a null angular velocity reference corresponds usually to the *de-spin* case, an operation often performed after launcher detachment or in safe modes. Assuming no disturbances the simplest control law to drive angular velocity to zero is

$$\boldsymbol{\tau}_\mathbf{u} = -\mathbf{K}\boldsymbol{\omega}_b \tag{5.2}$$

Substituting Eq. (5.2) in Eq. (5.1) and considering no disturbs it is straightforward to see that the only possible equilibrium point is $\boldsymbol{\omega}_b = \mathbf{0}$. To assess the stability of such point let us take as Lyapunov function candidate the following:

$$\mathfrak{L} = \frac{1}{2}\boldsymbol{\omega}_b^T\boldsymbol{I}_b\boldsymbol{\omega}_b \tag{5.3}$$

The function is positive for any $\boldsymbol{\omega}_b$, zero for null velocity and it is unbounded in $\boldsymbol{\omega}_b$. The time derivative of $\mathfrak{L}$ is

$$\frac{d}{dt}\mathfrak{L} = \boldsymbol{\omega}_b^T \boldsymbol{\tau}_\mathbf{u} \tag{5.4}$$

Substituting Eq. (5.2) leads to

$$\frac{d}{dt}\mathfrak{L} = -\boldsymbol{\omega}_b^T \mathbf{K} \boldsymbol{\omega}_b \tag{5.5}$$

that is always negative for any positive definite $\mathbf{K}$, regardless of $\boldsymbol{\omega}_b$. Using Barbalat lemma we can conclude that, under the disturb-less case hypothesis, the simple proportional feedback of Eq. (5.2) is globally stabilizing.

The simplest value of $\mathbf{K}$ would be $\mathbf{K} = \alpha \mathbf{I}_{3\times 3}$ with α a positive scalar. However, the inertia tensor of a satellite is rarely equivalent to a sphere, hence it might be wise to use instead $\mathbf{K} = \text{diag}\{\boldsymbol{\alpha}\}$. The choice of $\boldsymbol{\alpha}$ can be computed easily looking at the linearised version of Eq. (5.1) that holds for small angular velocities (or tri-symmetric bodies)

$$\boldsymbol{I}_b \dot{\boldsymbol{\omega}}_b \simeq -\mathbf{K} \boldsymbol{\omega}_b \tag{5.6}$$

$$\dot{\boldsymbol{\omega}}_b \simeq -\boldsymbol{I}_b^{-1} \mathbf{K} \boldsymbol{\omega}_b \tag{5.7}$$

Such equation exhibits no couplings if $\mathbf{K} = \text{diag}\{\boldsymbol{\alpha}\}$, therefore each axis can be considered separately. The equivalent SISO system is straightforward and can give useful insights.

Let ω_{br} be the reference angular velocity of the axis under exam and ω_{bj} the actual angular velocity of axis j, then the closed loop transfer function is

$$\frac{\omega_{bj}}{\omega_{br}} = \frac{\alpha_j}{\mathbb{I}_j s + \alpha_j} \tag{5.8}$$

where s is the Laplace variable. Taking $\alpha_j = \mathbb{I}_j \lambda_j$ is equivalent to assign the cutoff frequency of the closed loop system to be λ_j. The system behaves like a low pass filter for the reference ω_{br}. Another interesting insights is looking at the disturbance to output transfer function

$$\frac{\omega_{bj}}{\tau_{d_j}} = \frac{1}{\mathbb{I}_j s + \alpha_j} \tag{5.9}$$

Figure (5.1) presents its bode diagram, showing that all disturbances are reduced at least by a factor $\frac{1}{\alpha_j}$. Interesting to note that constant disturbances are not entirely rejected, hence this hints that for static $\boldsymbol{\tau}_\mathbf{d}$ the controller

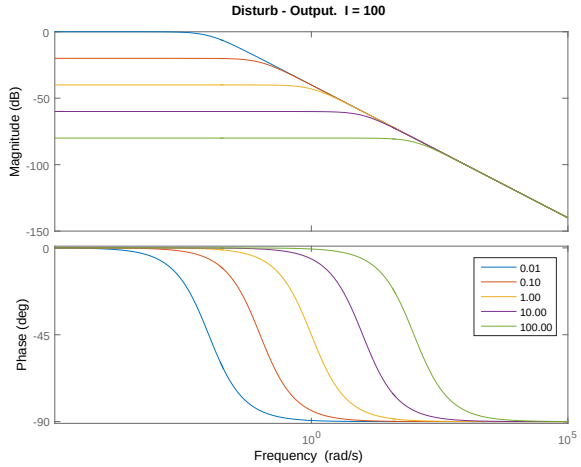


Figure 5.1: *Bode diagram of disturbance rejection*

of Eq. (5.2) is not capable of reaching zero error. To solve this issue a proportional and integrative controller should be devised. The disturbance attenuation factor is $\frac{1}{\alpha_j}$ in the lower frequencies.

5.1.2 Velocity tracking

In case the reference velocity is not null, then the controller of Eq. (5.2) is not guaranteed to achieve global stability. Let us denote the angular velocity error as

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_b - \boldsymbol{\omega}_{br} \tag{5.10}$$

$$\dot{\boldsymbol{\omega}}_e = \dot{\boldsymbol{\omega}}_b - \dot{\boldsymbol{\omega}}_{br} \tag{5.11}$$

Then the candidate Lyapunov function will be

$$\mathfrak{L} = \frac{1}{2}\boldsymbol{\omega}_e^T \boldsymbol{I}_b \boldsymbol{\omega}_e \tag{5.12}$$

whose derivative is given by adding Eq. (5.1) with $\boldsymbol{\tau_d} = \mathbf{0}_{3\times 1}$.

$$\frac{d}{dt}\mathfrak{L} = \boldsymbol{\omega}_e^T \left(-\boldsymbol{\omega}_b \times (\boldsymbol{I}_b \boldsymbol{\omega}_b) + \boldsymbol{\tau_u} - \boldsymbol{I}_b \dot{\boldsymbol{\omega}}_{br}\right) \tag{5.13}$$

Then the control law to achieve Lyapunov stability should be

$$\boldsymbol{\tau_u} = -\mathbf{K}\boldsymbol{\omega}_e + \boldsymbol{\omega}_b \times (\boldsymbol{I}_b \boldsymbol{\omega}_b) + \boldsymbol{I}_b \dot{\boldsymbol{\omega}}_{br} \tag{5.14}$$

whose block scheme is shown in Figure 5.2.

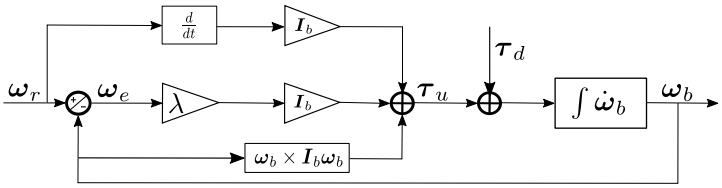


Figure 5.2: *Block scheme of Lyapunov based controller*

Integral action

The controller presented above is not capable of completely rejecting constant disturbs and the rejection capabilities can only be proven in a linearised setting, as will be shown in Section 5.1.4. From linear theory we do know that the addition of an integral action can reduce disturb influence on the output and reject the static component. From the non linear point of view should be investigated if the presence of an integral action alters the stability properties of the controlled system.

Let us modify the control law and the Lyapunov function as follows

$$\begin{cases} \mathfrak{L} & = \frac{1}{2}\boldsymbol{\omega}_e^T \boldsymbol{I}_b \boldsymbol{\omega}_e + \frac{1}{2}\boldsymbol{\xi}^T \kappa^{-1} \boldsymbol{\xi} \\ \boldsymbol{\tau}_\mathbf{u} & = -\mathbf{K}\boldsymbol{\omega}_e + \boldsymbol{\omega}_b \times (\boldsymbol{I}_b \boldsymbol{\omega}_b) + \boldsymbol{I}_b \dot{\boldsymbol{\omega}_{br}} + \boldsymbol{\xi} \end{cases} \tag{5.15}$$

Then the time derivative of $\mathfrak{L}$ is

$$\frac{d}{dt}\mathfrak{L} = -\boldsymbol{\omega}_e^T \mathbf{K} \boldsymbol{\omega}_e + \boldsymbol{\xi}^T \left(\boldsymbol{\omega}_e + \kappa^{-1} \dot{\boldsymbol{\xi}} \right) \tag{5.16}$$

Hence for any $\kappa > 0$ if we set $\dot{\boldsymbol{\xi}} = -\kappa \boldsymbol{\omega}_e$ the stability of the system is not altered. If the system is not well known, has disturbs or the derivative of the reference is not available, then a simplified feedback law could be used

$$\begin{cases} \boldsymbol{\tau}_\mathbf{u} & = -\mathbf{K}\boldsymbol{\omega}_e + \boldsymbol{\xi} \\ \dot{\boldsymbol{\xi}} & = -\kappa \boldsymbol{\omega}_e \end{cases} \tag{5.17}$$

Then $\boldsymbol{\xi}$ will not tend to zero but would be required to tend to a specific function of reference and angular velocity. Let us assume that a final $\overline{\boldsymbol{\xi}}$ will be reached, then a proper candidate Lyapunov function could be

$$\mathfrak{L} = \frac{1}{2}\boldsymbol{\omega}_e^T \boldsymbol{I}_b \boldsymbol{\omega}_e + \frac{1}{2}\left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}} \right)^T \kappa^{-1} \left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}} \right) \tag{5.18}$$

Differentiating Eq. (5.18) and substituting Eq. (5.17) with the assumption $\frac{d}{dt}\overline{\boldsymbol{\xi}} = \mathbf{0}$ leads to

$$\frac{d}{dt}\mathfrak{L} = \boldsymbol{\omega}_e^T \left(-\mathbf{K}\boldsymbol{\omega}_e - \boldsymbol{\omega}_b \times (\boldsymbol{I}_b\boldsymbol{\omega}_b) - \boldsymbol{I}_b\dot{\boldsymbol{\omega}}_{br} + \overline{\boldsymbol{\xi}}\right) \tag{5.19}$$

Stability holds if

$$\overline{\boldsymbol{\xi}} = \boldsymbol{\omega}_b \times (\boldsymbol{I}_b\boldsymbol{\omega}_b) + \boldsymbol{I}_b\dot{\boldsymbol{\omega}}_{br} \tag{5.20}$$

At convergence $\overline{\boldsymbol{\xi}}$ will be equal to the desired reference angular momentum. If $\frac{d}{dt}\overline{\boldsymbol{\xi}} \neq \mathbf{0}$ then the term $\left(\boldsymbol{\xi}\_\overline{\boldsymbol{\xi}}\right)^T \kappa^{-1}\frac{d}{dt}\overline{\boldsymbol{\xi}}$ will appear in the derivative of Eq. (5.18) and stability will be achieved only if $\boldsymbol{\xi} = \overline{\boldsymbol{\xi}}$.

Taking the second row of Eq. (5.17) and differentiating leads to

$$\ddot{\boldsymbol{\xi}} = -\kappa\dot{\boldsymbol{\omega}}_e = -\kappa\left(\dot{\boldsymbol{\omega}}_b - \dot{\boldsymbol{\omega}}_{br}\right) \tag{5.21}$$

$$\kappa^{-1}\boldsymbol{I}_b\ddot{\boldsymbol{\xi}} = -\boldsymbol{\xi} - \kappa^{-1}\mathbf{K}\dot{\boldsymbol{\xi}} + \left(\boldsymbol{\omega}_b \times (\boldsymbol{I}_b\boldsymbol{\omega}_b) + \boldsymbol{I}_b\dot{\boldsymbol{\omega}}_{br}\right) \tag{5.22}$$

That is a damped harmonic oscillator excited by the term in brackets. If resonances are avoided, as time tends to infinity $\boldsymbol{\xi}$ will tend to $\overline{\boldsymbol{\xi}}$, that is the requested value for stability. The damped oscillator system can be written as

$$\ddot{\boldsymbol{\xi}} + \lambda\dot{\boldsymbol{\xi}} + \kappa\boldsymbol{I}_b^{-1}\boldsymbol{\xi} = \kappa\boldsymbol{I}_b^{-1}\left(\boldsymbol{\omega}_b \times (\boldsymbol{I}_b\boldsymbol{\omega}_b) + \boldsymbol{I}_b\dot{\boldsymbol{\omega}}_{br}\right) \tag{5.23}$$

the frequencies of the oscillator are $\sqrt{\kappa\boldsymbol{I}_{ii}^{-1}}$, hence the reference angular momentum time derivative frequency content can be managed thanks to κ.

Should be noted that even if the integral term will eventually converge to the disturbing terms if the frequency content of such disturbs is below the critical frequency of the integral update, this does not assures that the derivative of the controller Lyapunov function to be always semi-negative definite. In fact, in general, $\boldsymbol{\xi} = \overline{\boldsymbol{\xi}}$ is not verified at all times and at the beginning could be an unstabilizing term. However, under the reasonable assumption that disturbs and non modelled dynamics are bounded, it can be proven under the above mentioned hypothesis that a bounded convergence is attainable. As will be clear later, the addition of an integral action can cause a phase delay in the controller response, thus lowering the convergence rate.

5.1.3 Complete feedback loop

Now, let us consider the velocity controller of Eq. (5.14) with the velocity reference of . The full attitude controller becomes

$$\begin{cases} \boldsymbol{\tau}_{\mathbf{u}} & = -\mathbf{K}\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br}\right) + \boldsymbol{\omega}_b \times \left(\boldsymbol{I}_c \boldsymbol{\omega}_b\right) + \boldsymbol{I}_c \dot{\boldsymbol{\omega}}_{br} \\ \boldsymbol{\omega}_{br} & = \beta \mathsf{s}\left(\epsilon\right) \boldsymbol{\eta}_+ \boldsymbol{\omega}_r \\ \dot{\boldsymbol{\omega}}_{br} & \simeq \beta \mathsf{s}\left(\epsilon\right) \left(\boldsymbol{\eta} \times \left(\boldsymbol{\omega}_b + \boldsymbol{\omega}_{br}\right) + \epsilon \left(\boldsymbol{\omega}_{br} - \boldsymbol{\omega}_b\right)\right) + \dot{\boldsymbol{\omega}}_r \\ \mathbf{K} & = \boldsymbol{I}_c \lambda \end{cases} \tag{5.24}$$

where $\boldsymbol{I}_c$ is the reference inertia used for the controller, often taken as diagonal. The term $\dot{\boldsymbol{\omega}}_{br}$ can be computed numerically or using the approximated expression above that is impervious to quaternion sign change.

The control action $\boldsymbol{\tau}_{\mathbf{u}}$ has been proven previously to uniformly asymptotically converge $\boldsymbol{\omega}_b$ to $\boldsymbol{\omega}_{br}$. $\boldsymbol{\omega}_{br}$ has been proven in Chapter 3 to drive the error in attitude towards zero. If we define the error in angular velocity as $\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br}$ we can conclude that $\boldsymbol{\omega}_b \to \beta \mathsf{s}\left(\epsilon\right) \boldsymbol{\eta} + \boldsymbol{\omega}_r$ and through we can conclude that $\boldsymbol{\eta} \to \mathbf{0}$ exponentially.

Should be noted that $\dot{\boldsymbol{\omega}}_{br}$ has not been computed exactly, since the derivative of $\mathsf{s}\left(\epsilon\right)$ in the neighbourhood of $\epsilon = 0$ is hill defined. This topological barrier formally decays the global convergence properties of the controller and this cannot be avoided unless a switching control is enforced . For practical applications this might be neglected and we can limit ourselves to check if unwinding might take place: thanks to $\mathsf{s}\left(\epsilon\right)$ no terms in $\boldsymbol{\tau}_{\mathbf{u}}$ changes sign if both $\boldsymbol{\eta}$ and ϵ change sign. Thus $\boldsymbol{\tau}_{\mathbf{u}}\left(\boldsymbol{\eta}, \epsilon\right) = \boldsymbol{\tau}_{\mathbf{u}}\left(-\boldsymbol{\eta}, -\epsilon\right)$. The price to pay is the possible chattering around $\epsilon = 0$ but the situation is quite unlikely and often can be solved using a good reference generation algorithm.

Another very important issue is the robustness to non modelled issues and parametric uncertainties. A robustness analysis must be conducted while keeping the uncertainties in a reasonable range. As a matter of fact no controller is able to fully control in case of completely underestimated phenomena concerning the system in exam. In this case we can consider the inertia tensor to be affected by uncertainties, while unknown disturbing forces can be dealt with a linear approximation of the system. The term $\mathbf{K}$ in the controller set the velocity loop cutoff frequency and in practical terms with linear control theory we can conclude that disturbs with lower frequency are rejected.

The minimum rejection is approximately given by $\mathbf{K}^{-1}\boldsymbol{\tau}_{\mathbf{d}}$ when no integral action is considered. An integral term would be able to completely nullify constant disturbs and increase the rejection of lower frequency disturbs, however, since the addition of an integral action reduces the phase margin it is advisable to use it only for the terminal phases of a manoeu-

vre or for fine pointing. It can be shown that using the integral action for large manoeuvres slows down the convergence considerably. A practical implementation of the controller would be

$$\begin{cases} \boldsymbol{\tau}_{\mathbf{u}} & = -\mathbf{K}\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br} - \kappa\psi\boldsymbol{\xi}\right) + \boldsymbol{\omega}_b \times \left(\boldsymbol{I}_c\boldsymbol{\omega}_b\right) + \boldsymbol{I}_c\dot{\boldsymbol{\omega}}_{br} \\ \boldsymbol{\omega}_{br} & = \beta \mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} + \boldsymbol{\omega}_r \\ \dot{\boldsymbol{\omega}}_{br} & \simeq \beta \mathsf{s}\left(\epsilon\right)\left(\boldsymbol{\eta} \times \left(\boldsymbol{\omega}_b + \boldsymbol{\omega}_{br}\right) + \epsilon\left(\boldsymbol{\omega}_{br} - \boldsymbol{\omega}_b\right)\right) + \dot{\boldsymbol{\omega}}_r \\ \dot{\boldsymbol{\xi}} & = -\psi\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br}\right) \\ \psi & \neq 0 \qquad \text{if} \frac{d}{dt}\left(1-\epsilon^2\right) \simeq 0 \end{cases} \tag{5.25}$$

where the condition on ψ is meant to formally imply that the integral action is active only near convergence in order to not slow down the controller. Should be added also that each time the integral action is triggered there should be a reset of the initial conditions in order to avoid excessive overshooting. This implies that the switching criteria should allow the integral action to keep acting when it is required to act and not keep switching.

5.1.4 Local stability and disturbance rejection analysis

In order to easily determine the disturbance rejection conditions as well as to tune parameters a linear analysis can be very insightful. Let us consider the feedback system with no uncertainties and no integral action and rearrange the terms. By setting $\boldsymbol{\omega}_e = \boldsymbol{\omega}_b - \boldsymbol{\omega}_r$ we find

$$\boldsymbol{I}_b\dot{\boldsymbol{\omega}}_e = \boldsymbol{\tau}_{\mathbf{d}} - \mathbf{K}\boldsymbol{\omega}_e + \mathbf{K}\beta \mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} + \boldsymbol{I}_c\beta \mathsf{s}\left(\epsilon\right)\dot{\boldsymbol{\eta}} \tag{5.26}$$

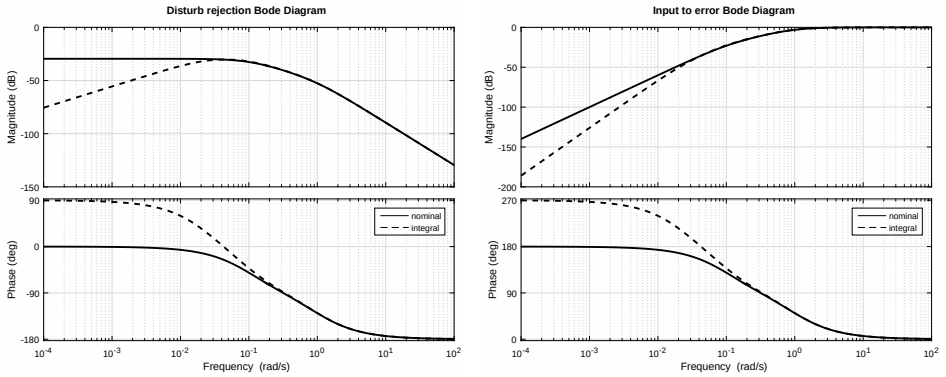
Linearising the system around the equilibrium condition and setting the linearised variables $\boldsymbol{\vartheta}$ as follows

$$\begin{cases} \epsilon & \simeq \pm 1 \\ 2\mathsf{s}\left(\epsilon\right)\dot{\boldsymbol{\eta}} & \simeq -\boldsymbol{\omega}_e \simeq -\dot{\boldsymbol{\vartheta}} \\ \boldsymbol{\eta} & \simeq -\frac{1}{2}\boldsymbol{\vartheta} \end{cases} \tag{5.27}$$

lead to the simplified second order time invariant system

$$\boldsymbol{I}_b\ddot{\boldsymbol{\vartheta}} + \left(\mathbf{K} + \boldsymbol{I}_c\frac{\beta}{2}\right)\dot{\boldsymbol{\vartheta}} + \mathbf{K}\frac{\beta}{2}\boldsymbol{\vartheta} = \boldsymbol{\tau}_{\mathbf{d}} \tag{5.28}$$

The linearised system has no coupling if we consider principal axis and $\mathbf{K} = \lambda\boldsymbol{I}_c$ and $\boldsymbol{I}_c = \boldsymbol{I}_b$.



(a) *Disturb rejection* **(b)** *Tracking error*

Figure 5.3: *Linearized system Bode plots*

$$\ddot{\boldsymbol{\vartheta}} + \left(\lambda + \frac{\beta}{2}\right)\dot{\boldsymbol{\vartheta}} + \lambda\frac{\beta}{2}\boldsymbol{\vartheta} = \boldsymbol{I}_b^{-1}\boldsymbol{\tau}_{\mathbf{d}}$$

we can now consider each axis independently and write the disturb to tracking error transfer function for generic axis i

$$\frac{\vartheta_i}{\tau_{d_i}} = \frac{{}^{1}\!/\!_{\mathbb{I}_i}}{s^2 + \left(\lambda + {}^{\beta}\!/\!_{2}\right)s + \left(\lambda {}^{\beta}\!/\!_{2}\right)} \tag{5.29}$$

from which is clear that a constant disturbing torque is rejected by a factor $\mathbb{I}_i\lambda\frac{\beta}{2}$.

The integral term in the linearised Eq. (5.26) would appear as $\boldsymbol{\xi} \simeq -\kappa\boldsymbol{\vartheta} - \frac{\kappa\beta}{s}\boldsymbol{\vartheta}$ and leads to the following modification of Eq. (5.29)

$$\frac{\vartheta_i}{\tau_{d_i}} = \frac{{}^{s}\!/\!_{\mathbb{I}_i}}{s^3 + \left(\lambda + {}^{\beta}\!/\!_{2}\right)s^2 + \left(\lambda {}^{\beta}\!/\!_{2} + {}^{\kappa}\!/\!_{\mathbb{I}_i}\right)s + {}^{\kappa}\!/\!_{\mathbb{I}_i}\beta} \tag{5.30}$$

Now it is easy to see with the aid of Fig 5.3 that for a constant disturbing torque Eq. (5.30) tends to zero. Looking at the bode diagram of Fig (5.1) it is clear that the integral action does reduce the phase margin and delays the system response although it reduce the tracking error. As such it should be advisable to insert the integral action only for fine pointing when the error is below a threshold or when the derivative of the Lyapunov function approaches zero.

From the linearised system with no integral action it is easy to determine a relation between the control weights and the resulting frequencies f and damping coefficients of the closed loop system:

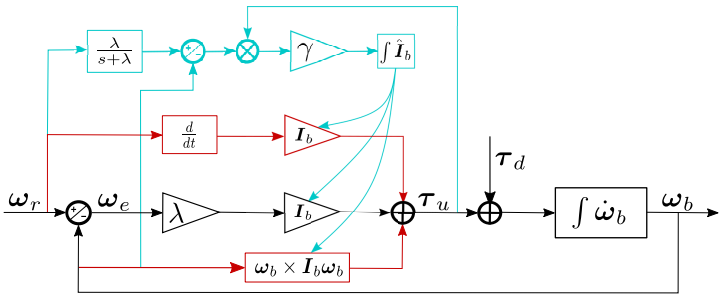


Figure 5.4: *Inertia - adaptive Lyapounov controller block scheme*

$$\begin{cases} f &= \sqrt{\frac{\lambda\beta}{2}} \\ \xi &= \frac{\sqrt{2}}{2}\frac{\lambda+\beta/2}{\sqrt{\lambda\beta}} \end{cases} \tag{5.31}$$

however should be noted that $\beta/2$ locates the first pole, thus is equal the reference following cutoff frequency. The controller is able to track attitude reference with frequency content lower than $\beta/2$ with a decrease in error tracking of 20 dB per decade as the frequency of the reference lowers. With integral action, as seen in Fig (5.1), the error reduces accordingly.

5.2 Adaptive Dynamic Inversion Controller

The control law derived so far is capable of achieving global convergence in absence of disturbs. The linearised analysis shows the rejection of disturbs and general performance and guidelines have been derived. The result is dependent on the physical parameters and their estimation. There are cases in OOS framework where the system parameter may change due to geometrical re-configuration of the orbital robot or due to addition of another satellite mass. In such cases the parameter should change in order to keep the desired performance, otherwise problems might arise due to the complex influence of the various feedback loops. An adaptive control law is able to reduce the errors due to non modelled variations or estimation errors.

The idea of keeping the control law as simple as possible should still be maintained, therefore the control law here proposed has been developed to minimize the complexity and retain in the limits the converging properties of the previous controller.

With the exception of the non linear term, the control law presented is purely decoupled, thus we can derive the adaptive control law on each

axis independently for explanatory purposes. The feed forward term will also not be considered, as it does not compromise the stability properties of the controller. The adaptive control paradigm used in this derivation is the *Adaptive Dynamic Inversion* (ADI) that shows more flexibility with respect to the *Model Reference Adaptive Control* for the studied problem. The choice is mainly due to the better ability of the ADI to incorporate feed forward and to comply better with time varying reference signals. The cost to pay is in general an increased control action, but in OOS scenarios where high precision is required such effort is justified. A more detailed comparison with some theoretical background can be found in Appendix B.

Let us write the simplified single axis model of the system by simplifying Eq. (5.1) over axis i

$$\dot{\omega}_i \simeq \frac{1}{\mathbb{I}_i}\tau_i \tag{5.32}$$

while the reference model with exponential convergence to be tracked is

$$\dot{\omega_{i,m}} = \lambda_i \left(\omega_{i,r} - \omega_{i,m}\right) \tag{5.33}$$

Let us modify (5.32) as follows

$$\dot{\omega}_i = \frac{1}{\mathbb{I}_i}\tau_i + \frac{1}{\hat{\mathbb{I}}_i}\tau_i - \frac{1}{\hat{\mathbb{I}}_i}\tau_i \tag{5.34}$$

$$\dot{\omega}_i = \frac{1}{\hat{\mathbb{I}}_i}\tau_i + \Delta b \tau_i \tag{5.35}$$

with

$$\Delta b = \frac{1}{\mathbb{I}_i} - \frac{1}{\hat{\mathbb{I}}_i} \tag{5.36}$$

Let us define two separate errors in order to prove nominal convergence and convergence to the reference model behaviour.

$$\begin{cases} e_1 &= \omega_{i,r} - \omega_i \\ e_2 &= \omega_{i,m} - \omega_i \end{cases} \tag{5.37}$$

whose derivatives, assuming ω_r constant for the time being, are

$$\begin{cases} \dot{e_1} &= -\dot{\omega}_i \\ \dot{e_2} &= \dot{\omega_{i,m}} - \dot{\omega}_i \end{cases} \tag{5.38}$$

Should be noted that since $\omega_{i,m} \rightarrow \omega_r$ by construction, then $e_2 \rightarrow e_1$. For the proof only one error would suffice, here the duplication is taken to stress that the nominal stability is maintained. Expanding the derivatives with Eq. (5.33) and Eq. (5.35) gives

$$\begin{cases} \dot{e_1} & = -\frac{1}{\hat{\mathbb{I}}_i}\tau_i - \Delta b \tau_i \\ \dot{e_2} & = \lambda_i \left(\omega_{i,r} - \omega_{i,m}\right) - \frac{1}{\hat{\mathbb{I}}_i}\tau_i - \Delta b \tau_i \end{cases} \tag{5.39}$$

The adaptive control law in this framework assumes the following expression

$$\boldsymbol{\tau}_i = -\hat{\mathbb{I}}_i \lambda_i \left(\omega_i - \omega_{i,r}\right) = \hat{\mathbb{I}}_i \lambda_i e_1 \tag{5.40}$$

Substituting it into the error derivatives leads to

$$\begin{cases} \dot{e_1} & = -\lambda_i e_1 - \Delta b \tau_i \\ \dot{e_2} & = -\lambda_i e_2 - \Delta b \tau_i \end{cases} \tag{5.41}$$

Let us take the following candidate Lyapunov function

$$\mathfrak{L} = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}\Delta b^2 \tag{5.42}$$

The derivative is

$$\dot{\mathfrak{L}} = -\lambda_i e_1^2 - \lambda_i e_2^2 + \Delta b \left(-e_1\tau_i - e_2\tau_i + \dot{\Delta} b\right) \tag{5.43}$$

since $\lambda_i > 0$ as it represent the wanted loop cutoff frequency, the straightforward $\dot{\Delta} b$ that does not affect the Lyapunov stability is

$$\dot{\Delta} b = \left(e_1 + e_2\right)\tau_i \simeq e\tau_i \tag{5.44}$$

where it is also possible to add a gain γ without changing the convergence of the Lyapunov function. Assuming that $\dot{\mathbb{I}}_i = 0$ we can substitute $\dot{\Delta} b$ and get the adaptive inertia term as

$$\frac{d}{dt}\hat{\mathbb{I}}_i = \gamma e_i \tau_i \hat{\mathbb{I}}_i^{\,2} \tag{5.45}$$

however this law might have some issues and a simplified version is instead used. In fact, integrating (5.45) by parts leads to

$$\hat{\mathbb{I}}_i = \hat{\mathbb{I}_{i,0}} - \frac{1}{\int \gamma e_i \tau_i dt} \tag{5.46}$$

hence if the integral converges to zero the inertia estimated skyrockets and can only be avoided by limiting the inertia, which is a common practice. The problem is that if the integral oscillates between two values with opposite sign, it will cross 0 several times, and each time the inertia will reach its maximum value. Hence, the control law that has been found to be most suitable for the examined problem is

$$\frac{d}{dt}\hat{\mathbb{I}}_i = \gamma e_2 \tau_i \tag{5.47}$$

Substituting $\dot{\Delta}b = \hat{\mathbb{I}}_i^{-2}\frac{d}{dt}\hat{\mathbb{I}}_i$, Δb and τ_i in Eq. (5.43) gives

$$\dot{\mathfrak{L}} = -\lambda_i e_1^2 - \lambda_i e_2^2 + \left(\frac{1}{\mathbb{I}_i} - \frac{1}{\hat{\mathbb{I}}_i}\right)\left(-e_1 - e_2 + \hat{\mathbb{I}}_i^{-2}\gamma e_2\right)\hat{\mathbb{I}}_i \lambda_i e_1$$

to simplify the expression recall that $e_2 \to e_1 = e$

$$\dot{\mathfrak{L}} = -\lambda_i e^2 \left(2 - \left(\frac{\hat{\mathbb{I}}_i}{\mathbb{I}_i} - 1\right)\left(\gamma\hat{\mathbb{I}}_i^{-2} - 2\right)\hat{\mathbb{I}}_i\right) \tag{5.48}$$

then requirement for asymptotic stability is

$$2 - \left(\frac{\hat{\mathbb{I}}_i}{\mathbb{I}_i} - 1\right)\left(\gamma\hat{\mathbb{I}}_i^{-2} - 2\right)\hat{\mathbb{I}}_i > 0 \tag{5.49}$$

The discriminant line between stability and instability can be expressed either as function of γ or $\mathbb{I}_i$.

$$\mathbb{I}_i = \frac{\hat{\mathbb{I}}_i}{\gamma}\left(\gamma - 2\hat{\mathbb{I}}_i^{2}\right) \tag{5.50}$$

We can conclude that it always exist in a neighbourhood of $\mathbb{I}_i$ a maximum value of γ such that for all possible inertia estimates $\hat{\mathbb{I}}_i$ the adaptive controller is globally stabilizing. Deriving Eq. (5.50) with respect to $\hat{\mathbb{I}}_i$ and set it to zero helps to find the maximum value of γ as function of $\hat{\mathbb{I}}_i$. Interestingly the maximum value of the instability region lies on the line $\hat{\mathbb{I}}_i = {}^3\!/_2\mathbb{I}_i$ and consequently the maximum value of the gain γ to prove asymptotic stability is related to the minimum value one assumes the inertia can take

$$\gamma_{max} = {}^{27}\!/_2\mathbb{I}_{i,min}^2 \tag{5.51}$$

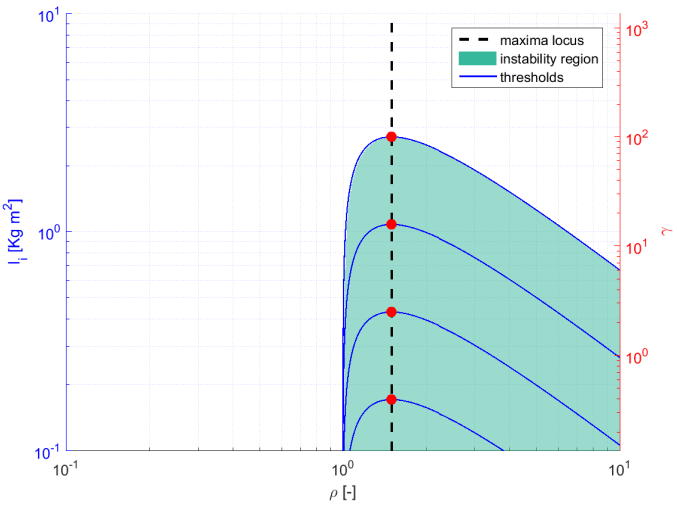


Figure 5.5: *Adaptive control convergence domain*

Figure 5.5 presents the region of stability and instability as function of γ. To give a better understanding of the regions, it has been used the parameter $\rho = \hat{\mathbb{I}}_i \mathbb{I}_i^{-1}$ on the x axis of the plot. Notably, when the estimator underestimates the inertia of the system, convergence it is still attained. Should be noted that in practice there is a lower bound due to the presence of disturbs. Then, as expected, there is a value of γ that guarantees stability for any estimates if the inertia of the system is above $\mathbb{I}_{i,min}$. Hence the gain γ can be tailored based on the estimation of the minimal value of the inertia to guarantee stability.

The single axis implementation is faster and simpler with respect to the full problem that would require to solve a matrix Lyapunov matrix equation and might be not necessary in many cases. This study can also be extended to the case with an integrative action by simply including the integral term inside the reference to be tracked, since in the end everything depends on the inertia properties and the integral term itself can be independent on the physical properties of the system.

The complete controller, to achieve Lyapunov stability requires other inertia dependent terms. Under the simplifying, yet plausible, assumption that the static velocity gain $\mathbf{K}$ is dominant with respect the other terms it is possible to update the non linear compensation terms $\boldsymbol{\omega}_b \times (\boldsymbol{I}\boldsymbol{\omega}_b)$ and $\boldsymbol{I}\dot{\boldsymbol{\omega}}_{br}$ with the ADI estimated inertia. As long as the estimation is bounded and the update gains are reasonably small the proposed scheme should not

be prone to instability. Namely, the simplified adaptive law of Eq. (5.47) acts on increasing the values of the gains if the expected response is not followed good enough.

The complete control law for attitude control with quaternion based guidance is

$$\begin{cases} \boldsymbol{\tau}_{\mathbf{u}} & = -\hat{\boldsymbol{I}}_b \text{diag}\left(\boldsymbol{\lambda}\right)\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br} - \kappa\psi\boldsymbol{\xi}\right) + \boldsymbol{\omega}_b \times \left(\hat{\boldsymbol{I}}_b\boldsymbol{\omega}_b\right) + \hat{\boldsymbol{I}}_b\dot{\boldsymbol{\omega}}_{br} \\ \boldsymbol{\omega}_{br} & = \beta \mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} + \boldsymbol{\omega}_r \\ \dot{\boldsymbol{\omega}}_{br} & \simeq \beta \mathsf{s}\left(\epsilon\right)\left(\boldsymbol{\eta} \times \left(\boldsymbol{\omega}_b + \boldsymbol{\omega}_{br}\right) + \epsilon\left(\boldsymbol{\omega}_{br} - \boldsymbol{\omega}_b\right)\right) + \dot{\boldsymbol{\omega}}_r \\ \dot{\boldsymbol{\xi}} & = -\psi\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_{br}\right) \\ \psi & \neq 0 \qquad \text{if} \frac{d}{dt}\left(1-\epsilon^2\right) \simeq 0 \\ \frac{d}{dt}\hat{\boldsymbol{I}}_b & = \gamma \boldsymbol{e} \odot \boldsymbol{\tau}_{\mathbf{u}} \end{cases} \tag{5.52}$$

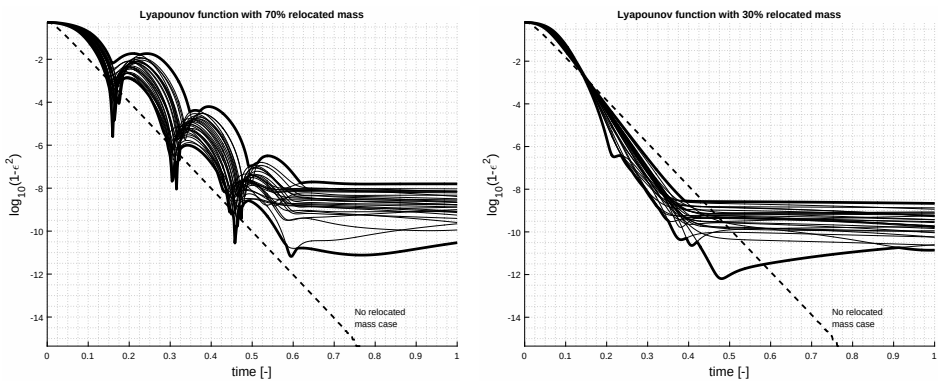
where it has been used the operator $\odot$ to denote the element wise vector multiplication and the error vector $\boldsymbol{e}$ is the variation from predicted state from the one measured and has components as defined beforehand in the single axis procedure.

5.3 Robustness analysis

The physical entities, the inertia tensor, may be poorly estimated or might change during the lifetime of the satellite due to different operative modes (ex: rotating panels, robotic arm moving, detachment or deployment of instruments, refueling) or unexpected events (non deployed solar panel, detachment of components) and this can pose huge problems if the controller is not robust enough.

It is difficult to pick up a test to simulate many of these scenarios and retain general conclusions thus some restriction must be enforced. The following test has been devised focusing on the effect of unknowingly relocated masses and for such only gravity force contribution is considered for all simulations.

The satellite is modelled as a cube of uniform density plus eight concentrated masses fixed at the center of each sub-cube generated by the three planes of symmetry parallel to the faces of the main body. The total mass is fixed and the inertia of the main body is computed using uniformity condition with mass equal to a fixed percentage of the total mass. The rest of the mass is distributed using a normalized uniform distribution among the other



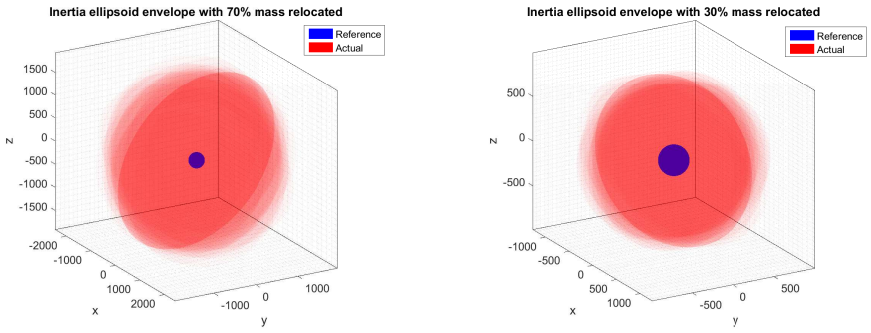
(a) *70% relocated mass* **(b)** *30% relocated mass*

Figure 5.6: *Robustness to parametric uncertainty*

eight masses. A direct consequence of this is that the resulting inertia principal components are higher than the nominal case where the whole mass is distributed uniformly in a single cube. Principal axes also varies from nominal condition, which is expected after docking of during the movement of a robotic arm. Designing the controller using as reference inertia the nominal cubic case would result in a underestimated inertia, the worse case with respect to overestimation as would not normally compromise the controllability.

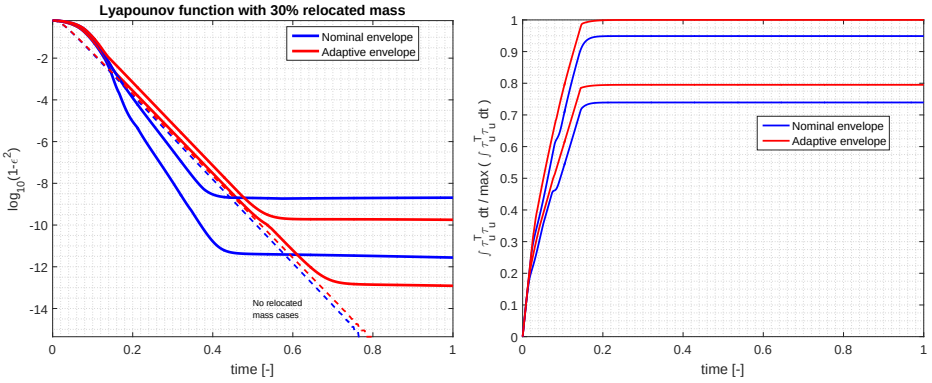
Two simulation campaigns are conducted with a relocated mass of 70 and 30 percent respectively. Gravity gradient is not inserted as a body inertia torque as each relocated mass has no inertia and the base is tri-symmetrical, instead, since it has been used the multibody approach, the gravity force acts on each mass, thus generating a discretised gravity gradient. The controller has no integral action active thus a residual error is expected in any non-nominal case. Initial conditions are set randomly at the beginning of the simulations and kept constant as mass distribution varies in order to retain a comparison in the output graphs. No external torques or measurement noises are considered.

From the graphs of Fig (5.6) it is clear that the convergence is not lost even for a 70% mass relocation but the performance is of course degraded. The Lyapunov function derivative is no longer always negative semi-definite but bounded convergence seems to be attained. In either case a delay in response is visible and expected. As reference, Fig (5.7) presents a visual representation and comparison of inertia ellipsoids of the nominal case and all the cases with relocated mass in order to put in perspective the



(a) *70 % relocated mass* **(b)** *30% relocated mass*

Figure 5.7: *Inertia ellipsoid envelopes of robustness test*



(a) *Lyapounov function envelopes* **(b)** *Control effort comparison*

Figure 5.8: *Adaptive controller robustness test*

uncertainties the control can overcome.

The same procedure is then applied to the ADI controller considering the inertia estimation to be bounded within a order of magnitude from the initial estimation and with non linear parameters updated accordingly. The simulations are performed with 30% relocated mass and for each mass distribution the nominal and adaptive controller are tested.

From Fig (5.8) we can appreciate the ADI controller ability to stick to the reference behaviour more than the nominal case. The adaptation of the inertia allows to reduce the final error at the cost of an evident but justifiable control effort. The adaptive version of the controller is expected to stick more to the expected behaviour of the system, increasing the predictability of the attitude controller, which can be a desired feature when looking at

the complete GNC loop or with respect to other systems like robotic arms.

5.4 Position control

Controlling the position of a satellite is not easy task, however there are fewer problems in terms of kinematics. Using dual quaternions it is easy to make a chain of transformations between reference frames to control a spacecraft with respect to another one. In order to avoid unwinding problems it is wiser to use the position as control variable, that is already independent from the change in sign of any dual quaternion that can be used to determine the relative state.

The relative dynamics of the center of mass of a satellite with respect to the LVLH frame presented in Chapter 2 can be linearised and in case of circular orbit it is also time invariant. Regardless, in this reference frame closed form solutions can be found and a trajectory can be devised to minimize fuel expense or maximize safety of operations.

Let the rotating orbital frame be attached to the center of mass of the satellite and $\mathbf{q}_r$ and its derivative represent the attitude of the satellite with respect to this frame given the quaternions for the satellite $(\mathbf{q}_b)$ and frame $(\mathbf{q}_h)$ with respect to Earth as follows

$$\mathbf{q}_r = \left[\mathbf{q}_{h\otimes}^{+}\right]^T \mathbf{q}_b \tag{5.53}$$

Then the dual quaternion representing the target body in frame h using a measurement relative to the satellite is given simply by

$$\boldsymbol{a}_h = \left[\begin{Bmatrix} \mathbf{q}_r \\ \mathbf{0}_{4\times 1} \end{Bmatrix}_{\epsilon\times}^{+}\right] \boldsymbol{a}_r \tag{5.54}$$

such that the dual quaternions of target and satellite are connected by

$$\boldsymbol{a}_t = \left[\boldsymbol{a}_{r\epsilon\times}^{-}\right] \boldsymbol{a}_b \tag{5.55}$$

should be noted that $\boldsymbol{a}_r$ contains info on the relative attitude between the two satellite, information that can be neglected when looking only at relative position. Moreover when two satellites are far, the estimation of a relative attitude might not be sufficiently precise and introduce more errors. Simplifying gives

$$\begin{Bmatrix} \boldsymbol{d}_h \\ 0 \end{Bmatrix} = \left[\mathbf{q}_{h\otimes}^{-}\right] \left[\mathbf{q}_{b\otimes}^{-}\right]^T \left[\mathbf{q}_{h\otimes}^{+}\right]^T \left[\mathbf{q}_{b\otimes}^{+}\right] \begin{Bmatrix} \boldsymbol{d}_b \\ 0 \end{Bmatrix} \tag{5.56}$$

that can be further simplified in the obvious $\boldsymbol{d}_h = \mathbf{R}_{hb}\boldsymbol{d}_b$ where the position vector in body frame $\boldsymbol{d}_b$ is simply rotated using the relative quaternion $\mathbf{q}_r$ as one would expect. Then using the information of $\boldsymbol{d}_h$ a simple PD controller can be implemented to follow a trajectory easily designed in that reference frame, remembering to translate back the computed forces in the body reference frame. Alternatively it is possible to rotate the trajectory from frame h to body frame, although it is more effective to build the controller in frame h since the dynamics is well known and understood.

When decoupling the attitude control and position control, assuming the two set of actuators are independent, there could be cases where the attitude variation might influence the position control, however if the transformations are well performed there should be no problem.

When thinking about an orbital rendezvous and docking it is necessary to achieve relative attitude and position, hence finding relative velocities such that a relative position and attitude error tends to zero.

If we consider $\boldsymbol{a}_r$ in Eq. (5.55) to be the error computed from the body frame to the target frame $\boldsymbol{a}_t$ we can express the position error in as

$$\boldsymbol{d}_{b,r} = \mathbf{R}_{bi}\left(\boldsymbol{d}_{i,t} - \boldsymbol{d}_{i,b}\right) \tag{5.57}$$

Then a candidate Lyapunov function would be

$$\mathfrak{L} = \boldsymbol{d}_{b,r}^T\boldsymbol{d}_{b,r} = \left(\boldsymbol{d}_{i,t} - \boldsymbol{d}_{i,b}\right)^T\left(\boldsymbol{d}_{i,t} - \boldsymbol{d}_{i,b}\right) \tag{5.58}$$

hence by following an appropriate reference velocity in the base frame we can enforce $\dot{\mathfrak{L}} \leq 0\,\forall \boldsymbol{d}_{b,r}$

$$\dot{\mathfrak{L}} = \boldsymbol{d}_{b,r}^T\left(\dot{\boldsymbol{d}_{i,t}} - \dot{\boldsymbol{d}_{i,b}}\right) \tag{5.59}$$

$$\dot{\boldsymbol{d}_{i,b}} = \dot{\boldsymbol{d}_{i,t}} + \mathbf{K}\boldsymbol{d}_{b,r} \tag{5.60}$$

Then the reference velocity in the body frame is of course

$$\dot{\boldsymbol{d}_b} = \mathbf{R}_{bi}\left(\dot{\boldsymbol{d}_{i,t}} + \mathbf{K}\boldsymbol{d}_{b,r}\right) \tag{5.61}$$

hence we can determine a reference velocity in any reference frame and with any intermediate reference frame transformation and then translate in the reference frame where the velocity controller is able to translate the input into the proper action and achieve exponential convergence.

In relative GNC the estimation of distance and direction of the other satellite is of primary importance to fulfil mission objective safely. Sensors

suite varies in terms of distances and availability of a inter-satellite communication. To guarantee the minimum safety for the customer it is almost mandatory to have a communication, albeit contained in data rate and with possible delays in the connection chain. Using relative GPS measurement can give a reasonable estimation of satellite relative position and with a good dynamical model it is possible to propose a filter.

5.4.1 Thrusters control

In order to move the center of mass from the current orbit a thrusters are required. Big thrusters can provide enough accelerations to change orbits, while smaller ones are able to perform more refined manoeuvres. When dealing with proximity manoeuvre, the use of thrusters for position control is mandatory. Those thrusters are positioned usually near the corners and sides of a satellite so that they are also able to produce substantial torques. Reaction wheels are more suited for precise control when the control authority required is sufficiently small, in other cases thrusters are required. In case when the center of mass is moving due to fuel sloshing or transfer, robotic arm movement or satellite capture small disturbs will be added to the system. We can write the scheme that relates thrusters ON/OFF commands to actual forces and torques as follows

$$\begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{Bmatrix} = \mathbf{M}\psi_{th} \tag{5.62}$$

where the matrix $\mathbf{M}$ has dimension six times n, where n is the number of thrusters. In order to be able to deliver forces and torques with both signs and with the possibility to produce forces with nominal null torques the number of thrusters is more than six. The minimum number is 12 but for redundancy in OOS mission is highly recommended to have a full 24 thrusters configuration. ψ_{th} are the ON/OFF binary signal to the thrusters and usually are determined through PWM.

A naive determination of ψ_{th} could be

$$\psi_{th} = \mathbf{M}^* \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{Bmatrix} \tag{5.63}$$

However, with a least square solution the value of ψ_{th} will not be binary nor positive. In cases where for each engine exist another one capable of delivering the opposite forces and torques the sign might still be recovered. Another possible way is to write an optimization problem and solve for the

values of ψ_{th}, bounded to be between 0 and 1 while trying to minimize the sum of its element, closely linked to the fuel expense. An attempt using the LM algorithm has been made and used for later simulations of Chapter 7. Another possible strategy is to use Lyapunov-like thrusters selection as done in [85].

Then, once the value of ψ_{th} that closely match the requested forces and torques are computed, a PWM or any algorithm with similar behaviour, translates into actual command to the thrusters. In the modulation category we can list the Sigma Delta Modulation (SDM) that has been successfully implemented and ground tested for space applications in [86]. It has been shown that comparing with classical PWM the SDM can achieve a lower steady state error at the cost of higher control expense. Since for OOS applications the precision of estimation is closely linked to the safety of operations it is completely plausible to prefer the SDM over classic PWM at an increased control cost and subsequently of propellant mass expenses if that is the case. A direct comparison is included in Chapter 7.

Actuation errors

For the proposed GNCR suite the position control relies solely on thrusters for actuation and including errors in the actuation helps to assess the robustness of the position control. Two main factors might be source of errors in the actuation: the thrust module and the positioning and orientation of each thruster. The latter can be introduced by computing a perturbation of matrix M considering a random error with uniform distribution and bounded limits for the position and direction of thrust. This introduces a constant bias in the actuation, hence every time a thruster is selected to be switched ON it generates the nominal force plus a disturbing action on other axis. Errors in thrust magnitude can be treated as gain errors, hence the thrust magnitude is multiplied by a factor that is the composition of a constant part, a sinusoidal part with high or low frequency content and a random bounded factor with uniform distribution [87]. Such analysis are usually done when dealing with actuation failure to test if a control law is able to adapt to fluctuation in the actuation or reduced performances. Again, even in this case the perturbing effect is present only if the thrusters are fired. Since the proposed GNCR do not uses a model for the center of mass positioning, each time a mass moves the thrusters contribution changes from the nominal values that are stored and equivalent forces are also exerted on the main body, hence the algorithms could prove robust even in the baseline configuration.

CHAPTER 6

Robotics

This chapter is intended to address robotics for OOS from a high level perspective. Since, at this stage, there is no reference arm for commercial OOS operations, any attempt to dive into electro-mechanical/mechatronic details would not be supported by real data or insights. On the other hand without knowing the boundaries of servicing robotics it is unlikely to be able to design a full fledged robotic system. This means that in this chapter elements of robot control (path generation, state estimation, control) will be derived in order to perform a closed loop analysis for the whole Orbital Robot (OR). An OR is a free floating vehicle with robotic appendages orbiting in space, hence moving a manipulator would cause disturbances and motion on the base and vice versa. To some extent the attitude perturbations acting on a orbital robot might be negligible, hence a popular way to control the arm was to exploit angular momentum conservation [22, 23]. Pure conservation is never attained though and correction for gravity gradient should be included based on experimental data [6]. Drawback of this free-floating control are the need for torque control, the centralized control strategy and the motion of the OR base. The latter is more concerning because sensors for relative positioning are mounted on the base, hence rotation of such base might compromise state estimation thus jeopardizing the whole mis-

sion. The first two problems are a general concern in the implementation of such approaches as stated before in Chapter 5. Since the robot dynamics is complex any attempt to use it in a feedback loop would result in potential robustness weaknesses.

In space robotics particular attention has been given to trajectory planning to reduce attitude disturbances [88], however this concept rely on kinematic redundancy and thus can be applied with some limitations. In general the free floating control assumptions are not applicable in cases where position control is involved to maintain a precise station keeping near the target, as many OOS scenario request in order to maintain safety of customer satellite.

In [89] a review of Model Reference Adaptive Control has been proposed to control a robot and estimate the non linear terms that can be used to compensate an otherwise decentralized control [90]. Here the simplified control of Chapter 5 has been adapted to robotic arm control and is based on the Adaptive Dynamical Inversion framework. Adaptive manipulator schemes like [91] have been studied and implemented, although complexity still remained an issue. The idea of slightly modifying PID-like controller with more advanced techniques has been harboured in the past, for example sliding mode [92], or dynamic adaptation [90] even with neural networks [93].

A lot of research has been invested also in the so called *visual servoing* [94, 95], where the feedback loop of the manipulator is closed by a camera. The approach is very relevant to the OOS tasks but some further insights are needed. First of all, if a connection has to be performed, the camera mounted on the robotic arm is not able to provide measurements in the terminal phases due to out of focus issues and potential illumination occlusion.

Here the grasping control will not be covered since it has been assumed for the most part that the connection in OOS has been designed beforehand, thus specifics on connections for a servicing port are supposed to be known and not requiring grasping or similar actions. All the literature about impedance matching and control will not be covered as out of the scope of this research.

6.1 Robotic arm guidance

Any robotic arm used for OOS purposes requires a trajectory to be followed in the task space. Namely the tip, equipped with specific tools, needs to be moved relative to a target that can sometimes be in motion. In literature

different strategies for different applications have been devised, here focus will be given to fast computation of a reasonable path within the limits of operations of the robotic arm. Although path constraints due to self contact will not be covered it is important to stress out that in some rare case it could be an important matter. For example in the substitution of a part in orbit the robotic arm should take a component from its base and then plug into the target, however such cases are less likely and with proper attention errors should be avoided.

First of all the position and attitude of the tip of a robotic arm can be written as follows through dual quaternions. For reference on dual quaternions please refer to Appendix A.

$$\begin{cases} \boldsymbol{h} = \boldsymbol{f}\left(\boldsymbol{\vartheta}\right) \\ \dot{\boldsymbol{h}} = \boldsymbol{f}\left(\boldsymbol{\vartheta},\, \dot{\boldsymbol{\vartheta}}\right)' \end{cases} \tag{6.1}$$

it is possible to express the dual quaternion velocity with a linear relationship

$$\dot{\boldsymbol{h}} = \frac{\partial \boldsymbol{f}\left(\boldsymbol{\vartheta}\right)}{\partial \boldsymbol{\vartheta}} \dot{\boldsymbol{\vartheta}} = \boldsymbol{J}_{\boldsymbol{\vartheta}} \dot{\boldsymbol{\vartheta}} \tag{6.2}$$

where $\boldsymbol{J}_{\boldsymbol{\vartheta}}$ is the Jacobian of the manipulator. The end effector linear and angular velocities can be recovered by multiplying for the matrix $\mathbf{S}_h$ defined as

$$\mathbf{S}_h = 2 \begin{bmatrix} \left[\mathbf{q}_{h\otimes}^{+}\right]^T & \mathbf{0}_{4\times 4} \\ \left[\boldsymbol{t}_{h\otimes}^{-T}\right] & \left[\mathbf{q}_{h\otimes}^{-}\right]^T \end{bmatrix} \tag{6.3}$$

Due to the linearity of the relation between joint velocity and end effector velocities a popular control approach is the tip velocity control. This approach does not directly lead to a specific position and attitude, hence a proper reference trajectory would need to take into account the position error.

6.1.1 Classic implementation (L-Guidance)

To assess a guidance control laws that guarantees to move the end effector towards its target consider the following Lyapunov function

$$\mathfrak{L} = \left(1 - \epsilon^2\right) + \frac{1}{2} \boldsymbol{d}_e^T \boldsymbol{d}_e \tag{6.4}$$

where ϵ is the error quaternion scalar component and $\boldsymbol{d}_e = \boldsymbol{d}_r - \boldsymbol{d}$ the position error in the inertial frame. Following the approach on Chapter 3 the end effector desired velocities should be equal to

$$\begin{cases} \boldsymbol{\omega}_r & = \mathbf{K}_\omega \mathsf{s}\left(\epsilon\right) \boldsymbol{\eta} \\ \mathbf{v}_r & = \mathbf{K}_d \boldsymbol{d}_e \end{cases}$$

to guarantee exponential convergence. However the end effector velocities are dependent on the joint velocities, true variables of the problem. Such relations depends on the current configuration and current position error but is linear

$$\begin{Bmatrix} \boldsymbol{\omega}_r \\ \mathbf{v}_r \end{Bmatrix} = \mathbf{G}\dot{\boldsymbol{\vartheta}}_r \tag{6.5}$$

with $\mathbf{G} = \mathbf{S}_h \boldsymbol{J}_{\boldsymbol{\vartheta}}$ that is a matrix of dimensions six by n_ϑ that depends only on the joint position and the geometric correlations of the links. The derivative of the Lyapunov candidate function now depends on the reference joints velocity

$$\frac{d}{dt}\mathfrak{L} = -\begin{Bmatrix} \boldsymbol{\eta}^T \epsilon & \boldsymbol{d}_e^T \end{Bmatrix} \mathbf{G}\dot{\boldsymbol{\vartheta}}_r \tag{6.6}$$

and to achieve convergence the joints velocities must be put in correlation with the errors

$$\dot{\boldsymbol{\vartheta}}_r = \mathbf{U} \begin{Bmatrix} \mathsf{s}\left(\epsilon\right) \boldsymbol{\eta} \\ \boldsymbol{d}_e \end{Bmatrix} \tag{6.7}$$

If $\mathbf{GU}$ is symmetric semi positive definite, then $\frac{d}{dt}\mathfrak{L} < 0$ for any error position. A popular strategy is to use the Moore-Penrose pseudoinverse of $\mathbf{G}^*$

$$\mathbf{G}^* = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T \tag{6.8}$$

unfortunately there are some configurations where $\mathbf{G}^T\mathbf{G}$ is almost singular and convergence is lost. Another strategy is to use the transpose of $\mathbf{G}$ with less computation. When dealing with Jacobians a computationally effective way to handle them is to apply a column scaling.

$$\mathbf{G}_s = \mathbf{GD} \tag{6.9}$$

where the diagonal matrix $\mathbf{D}$ has each element elements equal to the inverse of the norm of the respective column of $\mathbf{G}$. Thus we can rewrite

$$\mathbf{GU} = \mathbf{GDK}_{\vartheta} \left(\mathbf{GD}\right)^{T} \tag{6.10}$$

with $\mathbf{K}_{\vartheta}$ constant and s.d.p. we would have convergence. The resulting reference velocity is thus

$$\dot{\boldsymbol{\vartheta}}_r = \mathbf{DK}_{\vartheta}\mathbf{DG}^{T} \begin{Bmatrix} \mathsf{s}\left(\epsilon\right)\boldsymbol{\eta} \\ \boldsymbol{d}_e \end{Bmatrix} \tag{6.11}$$

If the robotic arm can track effectively $\dot{\boldsymbol{\vartheta}}_r$ then The error would converge to zero exponentially. Should be noted that this reference, depending on the position error, thus on the joint position, can be inserted in the control loop directly without requiring integration of $\dot{\boldsymbol{\vartheta}}_r$. This formulation does not take into account joint limits, but could theoretically limit $\dot{\boldsymbol{\vartheta}}_r$ to be under the joints speed limits. The continuity of the reference is ensured up to speed level, so on the acceleration level some discontinuities might arise and excite the system in the high frequency range. A similar strategy has been used in [96] although the handling of dual quaternion is slightly different.

6.1.2 Filtered Levenberg-Marquardt robotic guidance

In order to generate a trajectory that is compliant with the joints limits, smooth, with frequency content that can be followed by the control and is not prone to Jacobian singularities, a new guidance has been designed. The new guidance is composed of two main blocks: one determine the final position the joints have to reach in order to follow the target and the second one translates the final values in a smooth trajectory that the arm controller can follow. A small addition can be made in order to achieve null error in case of moving target.

Final state determination

First the final state is computed using a non linear minimizer able to work under Jacobian singularities. In formal terms we search for the final joint state $\boldsymbol{\vartheta}_t$ such that the end effector position $\boldsymbol{h}$ is equal to the target location $\boldsymbol{h}_t$.

$$\boldsymbol{\vartheta}_t = \arg\min \left(\boldsymbol{\eta}^T\boldsymbol{\eta} + \boldsymbol{\eta}_t^T\boldsymbol{\eta}_t\right) \tag{6.12}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\eta}_t$ are respectively the vector part of the attitude and position quaternions of the dual quaternion error $\boldsymbol{e}_h$ given by

$$\boldsymbol{e}_h = \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right] \boldsymbol{h} = \left\{\boldsymbol{\eta}^T \quad \epsilon \quad \boldsymbol{\eta}_t^T \quad \epsilon_t\right\}^T \tag{6.13}$$

where the end effector position $\boldsymbol{h}$ is given by Eq. (6.1) and is function of the (desired) joint variables only. In order to stick to the matrix representation we can define a matrix $\boldsymbol{V}$ as follows

$$\boldsymbol{V} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{6.14}$$

so that we can express the objective function as

$$\boldsymbol{\vartheta}_t = \arg\min \left(\boldsymbol{f}\left(\boldsymbol{\vartheta}_t\right)^T \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right]^T \boldsymbol{V}^T \boldsymbol{V} \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right] \boldsymbol{f}\left(\boldsymbol{\vartheta}_t\right) \right) \tag{6.15}$$

where the dependency on $\boldsymbol{\vartheta}_t$ is better seen. This is not sufficient as $\boldsymbol{\vartheta}_t$ is bounded by the physical joints limits of the considered robotic arm, hence

$$\begin{aligned} \boldsymbol{\vartheta}_t = & \arg\min \left(\boldsymbol{f}\left(\boldsymbol{\vartheta}_t\right)^T \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right]^T \boldsymbol{V}^T \boldsymbol{V} \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right] \boldsymbol{f}\left(\boldsymbol{\vartheta}_t\right) \right) \\ & \text{s.t. } \boldsymbol{\vartheta}_{min} < \boldsymbol{\vartheta}_t < \boldsymbol{\vartheta}_{max} \end{aligned} \tag{6.16}$$

Considering computational efficiency and robustness, one of the fastest and most popular methods that can be used is the Levenberg-Marquardt (LM) minimizer . For implementation and theoretical background, refer to Appendix C. The algorithm can be seen as a damped Newton-Gauss method for non linear minimization and is able to solve with superlinear or quadratic convergence rate problems like the one in Eq. (6.15). From a robotics background could be seen as similar to the damped Moon-Penrose pseudo inverse method, however it does about 20 iterations per step, granting higher accuracy. The algorithm is fast and accurate in the determination of the desired final joint position, but requires extra care to introduce the joint limits of Eq. (6.16).

Two possibilities using a variable change are explored. Let us define a mapping function $g_m(\,)$ from the new variable vector $\boldsymbol{\xi}$ to the joint variables $\boldsymbol{\vartheta}_t$ such that we have

$$\boldsymbol{h} = \boldsymbol{f}\left(\boldsymbol{\vartheta}_t\right) = \boldsymbol{f}\left(g_m\left(\boldsymbol{\xi}\right)\right) = \boldsymbol{f}'\left(\boldsymbol{\xi}\right) \tag{6.17}$$

Then the problem in Eq. (6.15) changes to

$$\boldsymbol{\xi} = \arg\min \left(\boldsymbol{f}'\left(\boldsymbol{\xi}\right)^T \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right]^T \boldsymbol{V}^T \boldsymbol{V} \left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right] \boldsymbol{f}'\left(\boldsymbol{\xi}\right) \right) \tag{6.18}$$

Eq. (6.19) presents this option using the arctangent function which is a upper and lower bounded function of a parameter that can freely change with no limits.

$$\boldsymbol{\vartheta}_t = g_m\left(\boldsymbol{\xi}\right) = \frac{2}{\pi}\tan^{-1}\left(\boldsymbol{\xi} + \frac{\boldsymbol{\vartheta}_{max} + \boldsymbol{\vartheta}_{min}}{2}\right)\frac{\boldsymbol{\vartheta}_{max} - \boldsymbol{\vartheta}_{min}}{2} \tag{6.19}$$

With good initial guess this parametrization is able to convey the solution, however the search space is no longer well defined and multiple restart of the algorithm shall be performed especially when there is no prior information on possible solutions. The selling point of this transformation is that it does not increase the number of variables nor the number of equations, which is six with the $\boldsymbol{V}$ matrix considered. Should be noted that instead of a constant $\boldsymbol{V}$ it could have been introduced a matrix mapping the dual quaternion to $\boldsymbol{e}_h$ to a different position and attitude error representation, but for the minimization point of view this only increases the number of operations without adding anything to the solution.

Another strategy is to add dummy variables $\boldsymbol{\nu}$ to the problem and extend the cost function of Eq. (6.15) to account for joint limits. The mapping function is linear and still depends only on $\boldsymbol{\xi}$

$$\boldsymbol{\vartheta}_t = g_m\left(\boldsymbol{\xi}\right) = \boldsymbol{\xi} + \frac{\boldsymbol{\vartheta}_{max} + \boldsymbol{\vartheta}_{min}}{2} \tag{6.20}$$

but the cost function is modified as follows

$$\left(\boldsymbol{\xi},\, \boldsymbol{\nu}\right) = \arg\min\left(\boldsymbol{f}'\left(\boldsymbol{\xi}\right)^T\left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right]^T\boldsymbol{V}^T\boldsymbol{V}\left[\boldsymbol{h}_{t_{\epsilon\div}^{+T}}\right]\boldsymbol{f}'\left(\boldsymbol{\xi}\right) + \boldsymbol{e}_\vartheta{}^T\boldsymbol{e}_\vartheta\right) \tag{6.21}$$

with the constraint error function $\boldsymbol{e}_\vartheta$ expressed as follows

$$\boldsymbol{e}_\vartheta\left(\boldsymbol{\xi}, \boldsymbol{\nu}\right) = \boldsymbol{\xi}^2 + \boldsymbol{\nu}^2 - \left(\frac{\boldsymbol{\vartheta}_{max} - \boldsymbol{\vartheta}_{min}}{2}\right)^2 \tag{6.22}$$

This procedure increases both the number of equations and the number of variables by the number of joints composing the robotic arm. The LM algorithm is set to drive to zero the cost function thus at the same time it will modify both $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$ such that also Eq. (6.22) is verified. The constraint error function has been chosen in such a way that $\boldsymbol{\nu}$ is equal to the remaining margin of the joint. In fact with null error we have

$$\boldsymbol{\nu}^2 = \left(\frac{\boldsymbol{\vartheta}_{max} - \boldsymbol{\vartheta}_{min}}{2}\right)^2 - \boldsymbol{\xi}^2 \tag{6.23}$$

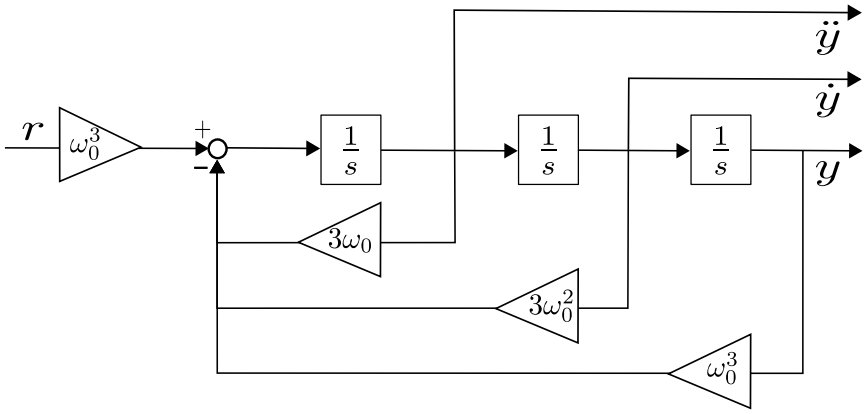


Figure 6.1: *Smoother block scheme*

a non negative function that is zero only for $\boldsymbol{\xi}^2 = \left(\frac{\boldsymbol{\vartheta}_{max}-\boldsymbol{\vartheta}_{min}}{2}\right)^2$ thus for $\boldsymbol{\xi} = \pm\left(\frac{\boldsymbol{\vartheta}_{max}-\boldsymbol{\vartheta}_{min}}{2}\right)$. If $|\boldsymbol{\xi}| > \left(\frac{\boldsymbol{\vartheta}_{max}-\boldsymbol{\vartheta}_{min}}{2}\right)$, meaning $\boldsymbol{\vartheta}_t > \boldsymbol{\vartheta}_{max}$ or $\boldsymbol{\vartheta}_t < \boldsymbol{\vartheta}_{min}$ we would have $\boldsymbol{\nu}^2 < 0$ hence $\boldsymbol{e}_\vartheta \neq \mathbf{0}$, thus the constraint would not be satisfied.

This parametrization with the addition of a constraint error function is heavier per iteration with respect to the first option but does not require any re-start as the solution space is well defined.

Should be noted that in case of redundant manipulators there exist multiple solutions and with the current framework is not possible to shift the problem to constrained minimization as the algorithm does not deliver good precision estimates. Forcibly adding an equation would reduce the accuracy of final pose and constraints. Theoretically it is possible to add a non linear constraint that keep under a user defined threshold the displacement from current configuration, however this generates a potential flaw in case of large manoeuvres. The solution that has been implemented is different and rather than focusing on minimizing a displacements it minimizes the jump from solutions as uses the previously computed solution as first guess for the new iteration. Then, the attractivity of the solution does the rest, as LM algorithm fast converges to the nearest solution which will be very close to the previous. Only one solution jump is possible at the start when no previous solution is present.

Smoother

Then the second step is to feed a proper trajectory to the controller, avoiding to excite the system with step or impulses using instead a proper trajectory.

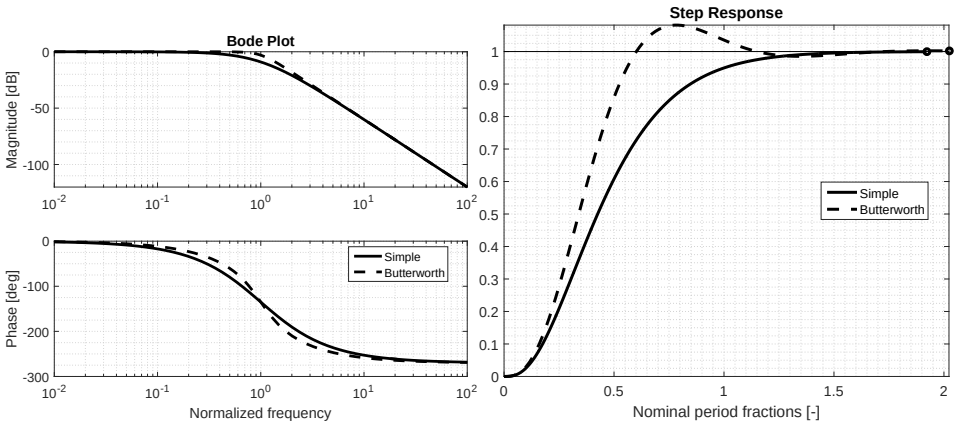


Figure 6.2: *Smoother filter comparison*

Take a lowpass filter of the third order

$$G(s) = \frac{\omega_0^3}{(s+\omega_0)^3} = \frac{\omega_0^3}{s^3 + 3\omega_0 s^2 + 3\omega_0^2 s + \omega_0^3} \tag{6.24}$$

It has cutoff frequency ω_0 and slope -3 afterwards. This can be read as

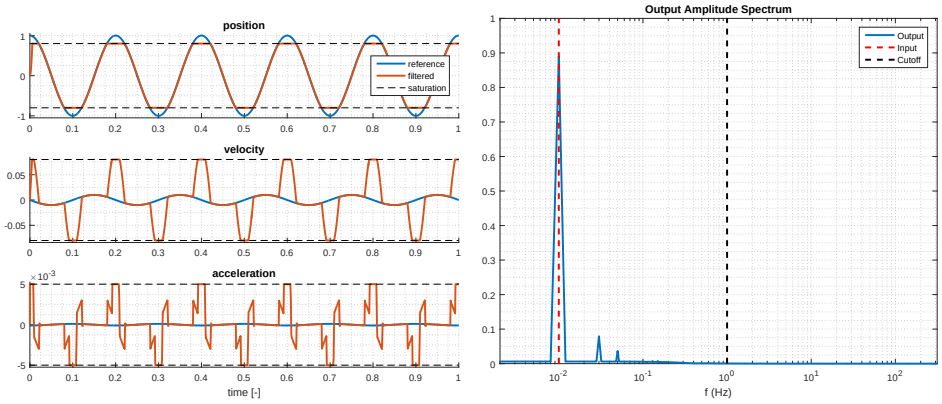
$$\frac{d}{dt}\dddot{y} = \omega_0^3 (r - y) - 3\omega_0 \ddot{y} - 3\omega_0^2 \dot{y} \tag{6.25}$$

and implemented in the block scheme in Figure 6.1. y will be the filtered signal of reference r, $\dot{y}$ its velocity and $\ddot{y}$ the acceleration. Then, by simply adding at each stage of integration a saturation with anti-windup it is possible to generate a reference signal whose derivatives are bounded and whose frequency content is essentially limited up to ω_0. It is possible to modify a bit the weights of the filter, for example the corresponding Butterworth filter would be

$$G(s) = \frac{\omega_0^3}{s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3} \tag{6.26}$$

Figure 6.2 present a brief comparison between the Butterworth version and the simpler version in terms of magnitude and phase diagrams and step response. Phase delay and filtering properties are quite close, however the Butterworth version has overshoot in the response, something that is not preferable for precise motion of the end effector.

This procedure can be applied to the reference $\boldsymbol{\vartheta}_t$ coming from the LM generating a reference trajectory that is continuous until the acceleration



(a) *Input - output* **(b)** *Spectrum*

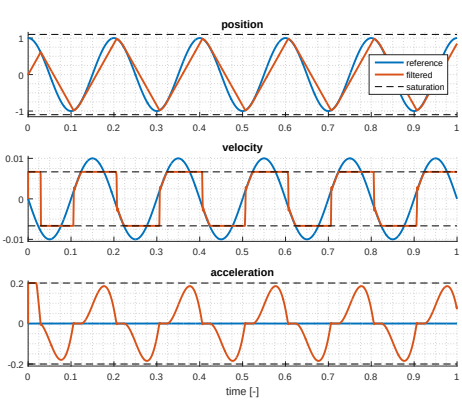
Figure 6.3: *Position saturation example*

level, that has acceleration, velocity and position limits and filter out the high frequency content.

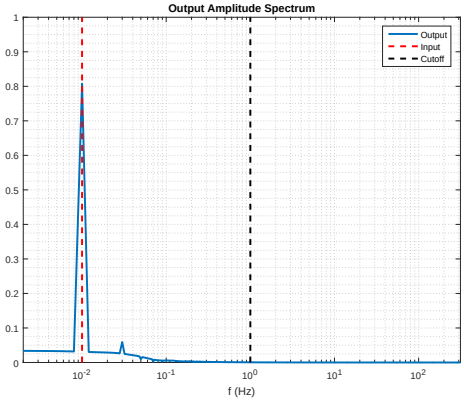
Figure 6.3 present an example of the filter with saturation of the reference position signal. The input signal is a unitary sinusoidal of frequency $0.01\ ^{rad}/_{s}$ with the filter with cutoff frequency of $1\ Hz$. The implementation is discrete with sampling time $0.01\ s$ and executed in Simulink. The saturation in position causes an abrupt increase in the reference velocity testified by discontinuities in the acceleration of the output. This is the main reason that lead to the choice of using a LM that takes care already of the position limits. Looking at the output frequency response it is clear that the signal contains some noise with frequency greater than the reference but lower than the cutoff frequency of the filter.

Figure (6.4) simulates a velocity saturation. First of all the position output keeps tracking the reference but due to velocity saturation cannot follow the reference in the high velocity region. This is also testified by the lower frequencies content in the output. Looking at the velocity confront between input and output seems that the anti-windup is not working, but in reality the lasting saturation is due to the filter trying to track the reference signal. Accelerations are continuous but greatly increased.

The input signal in both cases, being sinusoidal with no phase shift, has non zero velocity at the initial point, thus a spike in initial acceleration is required to start tracking completely the signal. In figure (6.5) the signal has a 90° phase shift and acceleration saturates at the beginning while the filter tries to follow the reference. The consequent frequency noise in the

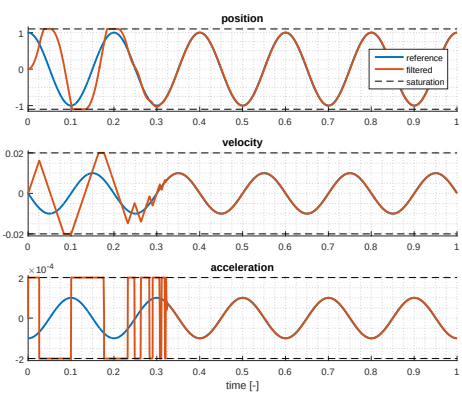


(a) *Input - output*

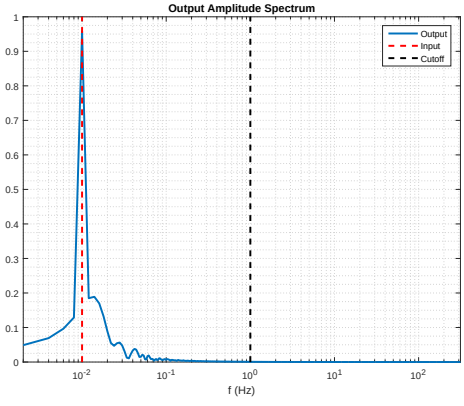


(b) *Spectrum*

Figure 6.4: *Velocity saturation example*



(a) *Input - output*



(b) *Spectrum*

Figure 6.5: *Acceleration saturation example*

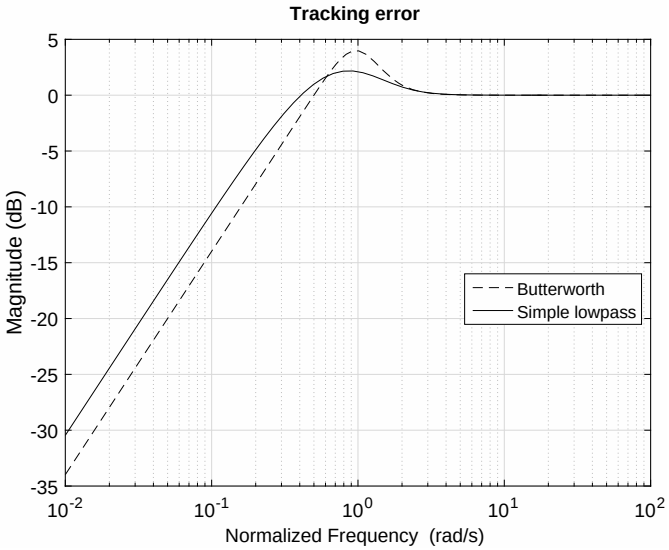


Figure 6.6: *Filter tracking error*

signal can be noticed in the spectrum plot.

When dealing with a moving target, should be noted that with the filtered approach there will be small phase delays as well as a tracking error that depends on the chosen filter. This can be observed in Fig (6.6) where comparison between the tracking error of the Butterworth version versus the simple lowpass implementation. Also, if using the LM strategy, the frequency of the LM update can influence the tracking of the moving target since it can track signals only up to its Nyquist limit.

Moving target modification

In case of moving target the solution LM plus smoother is not able to perfectly track a moving target, hence a modification must be made. When the end effector is almost in position a feedforward velocity signal is computed such that the velocity of the smoother is augmented with the velocity needed to match the moving target velocity. This can be achieved using the Jacobian of the end effector thus linking joint velocity with end effector velocity. This requires an estimation of the target velocity and the inversion (or transpose) of the Jacobian as popular robot kinematic references are generated, even with dual quaternion [96]. Let us take the derivative of the dual quaternion error (6.13) and substitute (6.2)

$$\left[\boldsymbol{h}_{t_{\epsilon\div}}^{+T}\right] \boldsymbol{J}_{\vartheta} \dot{\boldsymbol{\vartheta}} = \dot{\boldsymbol{e}_h} - \left[\boldsymbol{h}_{\epsilon\div}^{+}\right] \dot{\boldsymbol{h}_t} \tag{6.27}$$

Then posing the error variation in time to be null and exploiting some properties of dual quaternion the target joint velocities would be

$$\dot{\boldsymbol{\vartheta}} = \boldsymbol{J}_{\boldsymbol{\vartheta}}{}^{*} \left[\boldsymbol{e}_{h_{\epsilon \div}^{-}} \right] \dot{\boldsymbol{h}}_{t} \tag{6.28}$$

That is the feedforward velocity that is supplied to the end effector control in addition to the one computed by the smoother. To avoid instability or constraint issues, such feedforward shall be added only in terminal phases, when the error has converged to a somehow steady values or in general only when the error is small. A simple frequency analysis permits to assess the maximum target velocity that the end effector can actively follow, in many OOS applications relative velocities are kept at minimum by position control of the satellite base. In fact, with good base position control this addition can be irrelevant.

6.2 Robotic arm control

In many applications a service module requires a robotic arm or docking mechanism to fulfil some tasks, hence a proper control paradigm has to be found in order to be used in different scenarios. The dynamical system of a ground based robot can be written as

$$\mathbf{M}\left(\boldsymbol{\vartheta}\right) \ddot{\boldsymbol{\vartheta}} = \mathbf{f}\left(\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}\right) + \mathbf{d} + \mathbf{u} \tag{6.29}$$

where $\mathbf{M}$ is the mass matrix depending on mass distribution, inertia and the configuration given by the joint angles $\boldsymbol{\vartheta}$. $\dot{\boldsymbol{\vartheta}}$ and $\ddot{\boldsymbol{\vartheta}}$ are respectively joints velocity and acceleration, $\mathbf{f}$ a vector of non linear terms and $\mathbf{u}$ the vector of control torques, $\mathbf{d}$ the non modelled disturbing actions. To control such system two approaches are generally accepted: centralized and decentralized control. By considering each joint as a decoupled system a simple PD controller can be derived, in some cases it is possible to include the non linear coupling terms adding them to the control torques. The latter approach might suffer from parametric uncertainties and be less robust.

The simplified linear decoupled system is very akin to the attitude control scheme of Chapter 5 and the considerations made above still hold. To put this in a formal expression:

$$\begin{cases} \mathbf{u} & = -\mathbf{K}_2 \left(\dot{\boldsymbol{\vartheta}}_m - \dot{\boldsymbol{\vartheta}}_r \right) \\ \dot{\boldsymbol{\vartheta}}_r & = \mathbf{K}_1 \left(\boldsymbol{\vartheta}_r - \boldsymbol{\vartheta}_m \right) \end{cases} \tag{6.30}$$

with the gains $\mathbf{K}_1$ and $\mathbf{K}_2$ as diagonal positive definite matrices, $\boldsymbol{\vartheta}_r$ the reference joint position, $\boldsymbol{\vartheta}_m$ and $\dot{\boldsymbol{\vartheta}}_m$ the measured or estimated joint position and velocity. Considering no measurement errors the system becomes now

$$\mathbf{M}\ddot{\boldsymbol{\vartheta}} + \mathbf{K}_2\dot{\boldsymbol{\vartheta}} + \mathbf{K}_2\mathbf{K}_1\boldsymbol{\vartheta} = \mathbf{f} + \mathbf{d} + \mathbf{K}_2\mathbf{K}_1\boldsymbol{\vartheta}_r \tag{6.31}$$

That is a nonlinear second order coupled system. Let us rewrite the mass matrix with a constant diagonal matrix and a variation matrix

$$\mathbf{M}(\boldsymbol{\vartheta}) = \text{diag}(\boldsymbol{m}) + \Delta\mathbf{M}(\boldsymbol{\vartheta}) \tag{6.32}$$

then the system becomes

$$\text{diag}(\boldsymbol{m})\ddot{\boldsymbol{\vartheta}} + \mathbf{K}_2\dot{\boldsymbol{\vartheta}} + \mathbf{K}_2\mathbf{K}_1\boldsymbol{\vartheta} = \mathbf{f} + \mathbf{d} + \mathbf{K}_2\mathbf{K}_1\boldsymbol{\vartheta}_r - \Delta\mathbf{M}\ddot{\boldsymbol{\vartheta}} \tag{6.33}$$

that tends towards the value $\boldsymbol{\vartheta}_r + (\mathbf{K}_2\mathbf{K}_1)^{-1}(\mathbf{f} + \mathbf{d})$ if the accelerations are damped and tends to zero. The higher the control gains the lower the tracking error due to disturbs and non linear effects. If the system is well known, it could be possible to estimate the term $\mathbf{f} - \Delta\mathbf{M}\ddot{\boldsymbol{\vartheta}}$, but it has not been considered for robustness issues. The weights for the linear controllers can be chosen as before looking at the disturbing actions and reference frequency content. If $\mathbf{K}_2 \gg \mathbf{K}_1$ then both values are equal to each loop cutoff frequency.

The addition of an integrative term is sometime used to reject a constant torque acting on the robot, namely due to gravity, however there is still no formal proof that such integrative term ensures stability and usually are switched on only for when the system is close to the solution.

6.2.1 Lyapunov stability & integral action

Consider only the dynamical loop of (6.29) and let us study the convergence using Lyapunov theory. Define the joint velocity error as

$$\dot{\boldsymbol{\vartheta}}_e = \dot{\boldsymbol{\vartheta}} - \dot{\boldsymbol{\vartheta}}_r \tag{6.34}$$

$$\ddot{\boldsymbol{\vartheta}}_e = \ddot{\boldsymbol{\vartheta}} - \ddot{\boldsymbol{\vartheta}}_r \tag{6.35}$$

the candidate Lyapunov function

$$\mathfrak{L} = \frac{1}{2}\dot{\boldsymbol{\vartheta}}_e^T\mathbf{M}\dot{\boldsymbol{\vartheta}}_e \tag{6.36}$$

its time derivative is

$$\frac{d}{dt}\mathfrak{L} = \dot{\boldsymbol{\vartheta}}_e^T \left(\mathbf{f} + \mathbf{d} + \mathbf{u} - \mathbf{M}\ddot{\boldsymbol{\vartheta}}_r\right) \tag{6.37}$$

to achieve stability the control law should look like

$$\mathbf{u} = -\mathbf{K}_v\dot{\boldsymbol{\vartheta}}_e - \mathbf{f} - \mathbf{d} + \mathbf{M}\ddot{\boldsymbol{\vartheta}}_r \tag{6.38}$$

again we can add an integrative action and remove the unknown or computationally expensive parts of the dynamics and get the following control law

$$\mathbf{u} = -\mathbf{K}_v\dot{\boldsymbol{\vartheta}}_e + \boldsymbol{\xi} \tag{6.39}$$

then the modified Lyapunov candidate function would be

$$\mathfrak{L} = \frac{1}{2}\dot{\boldsymbol{\vartheta}}_e^T\mathbf{M}\dot{\boldsymbol{\vartheta}}_e + \frac{1}{2}\left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}}\right)^T\mathbf{K}_i\left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}}\right) \tag{6.40}$$

and its time derivative

$$\begin{aligned}\frac{d}{dt}\mathfrak{L} = &-\dot{\boldsymbol{\vartheta}}_e^T\mathbf{K}_v\dot{\boldsymbol{\vartheta}}_e + \dot{\boldsymbol{\vartheta}}_e^T\left(\mathbf{f} + \mathbf{d} - \mathbf{M}\ddot{\boldsymbol{\vartheta}}_r\right) + \boldsymbol{\xi}^T\left(\dot{\boldsymbol{\vartheta}}_e + \mathbf{K}_i\dot{\boldsymbol{\xi}}\right)\\ &-\overline{\boldsymbol{\xi}}^T\mathbf{K}_i\dot{\boldsymbol{\xi}} - \left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}}\right)^T\mathbf{K}_i\dot{\overline{\boldsymbol{\xi}}}\end{aligned} \tag{6.41}$$

The variation in time of $\boldsymbol{\xi}$ is computed to nullify the term in brackets

$$\dot{\boldsymbol{\xi}} = -\mathbf{K}_i^{-1}\dot{\boldsymbol{\vartheta}}_e \tag{6.42}$$

and thus remains

$$\frac{d}{dt}\mathfrak{L} = -\dot{\boldsymbol{\vartheta}}_e^T\mathbf{K}_v\dot{\boldsymbol{\vartheta}}_e + \dot{\boldsymbol{\vartheta}}_e^T\left(\mathbf{f} + \mathbf{d} - \mathbf{M}\ddot{\boldsymbol{\vartheta}}_r + \overline{\boldsymbol{\xi}}\right) - \left(\boldsymbol{\xi} - \overline{\boldsymbol{\xi}}\right)^T\mathbf{K}_i\dot{\overline{\boldsymbol{\xi}}} \tag{6.43}$$

stability is ensured if

$$\overline{\boldsymbol{\xi}} = \mathbf{M}\ddot{\boldsymbol{\vartheta}}_r - \mathbf{f} - \mathbf{d} \tag{6.44}$$

and $\boldsymbol{\xi} \to \overline{\boldsymbol{\xi}}$ regardless of the variation of $\overline{\boldsymbol{\xi}}$ in time. Then differentiating (6.42) and substituting the control law (6.39) and the theoretical value of $\overline{\boldsymbol{\xi}}$ results in a n-dimensional coupled harmonic oscillator

$$\mathbf{M}\mathbf{K}_i\ddot{\boldsymbol{\xi}} + \mathbf{K}_v\mathbf{K}_i\dot{\boldsymbol{\xi}} + \boldsymbol{\xi} = \overline{\boldsymbol{\xi}} \tag{6.45}$$

This time the difference is that the mass matrix is not fixed in time and thus formally the convergence can be guaranteed only in quasi-static conditions and with frequencies lower than the cutoff frequency of the above system. The frequency and the damping of the system are affected by the choice of $\mathbf{K}_i$.

6.3 Adaptive control

Like for the attitude case, even here an adaptive law can be used one the dynamic loop to adapt to parameter change in time. Should be stressed out again that only in some special cases the estimated parameters are equal to their true value, however the controller is much more adaptable to changes and practically extends the converging region of the controller.

The controller developed in Chapter 5 is now applied to the control of a robot, taking into account that unlike for attitude control, a OR is extremely coupled. The decoupled controller sees as disturbances any other torque applied to the link, meaning that coupling terms in velocity and acceleration are labelled as disturbs and rejected. The adaptive control is set to do the same but it is expected a more centralized answer to the adaptation.

In other words, each time a joint controller increases its action it also increases the disturbing torques on other joints, that on their end will probably increase their own action. Would this happen all the times, it would mean complete divergence of the control as a whole, however what happens is that all gains increase at the same time and converge to a value that is maintained if the arm operates in that neighbourhood. These values do not represent the decoupled inertia parameters $\boldsymbol{m}$ of Eq (6.32) as the same trajectory might lead to different values depending on the initial estimation of the inertia related term due to the expected choral increase stated before.

CHAPTER 7

Simulations

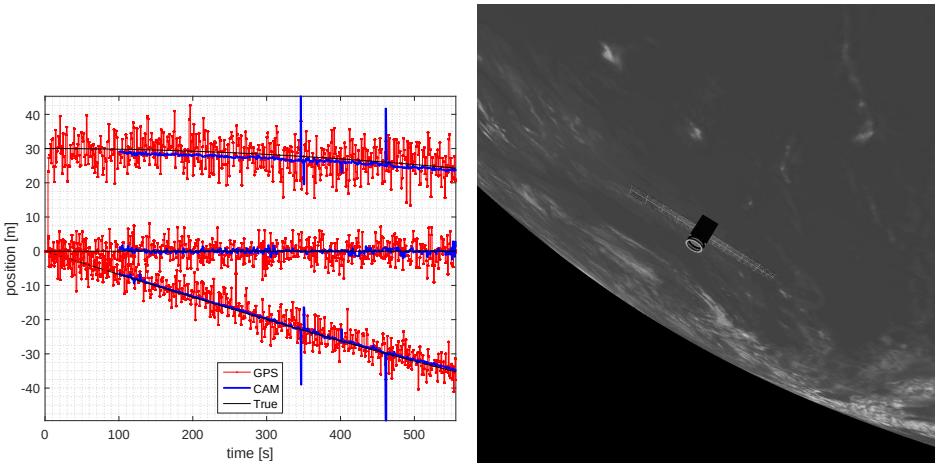
In this section two main scenarios will be analysed. First an inspection manoeuvre with recovery after safe mode making use of relative GPS and stereo camera estimation will be presented and results analysed. Finally, a simulation using vision estimation in the loop is performed and results analysed for a close proximity operation common to many OOS scenarios.

7.1 Inspection

The goal of this section is to show the proposed kinematic parametrization and GNC infrastructure applied to OOS scenario. Peculiarity of OOS are close proximity operations, formation flying with a non dedicated satellite and inspection. Of the many possible scenarios to analyse here will be included a scenario to test attitude control and position estimation using GPS and camera measurements. Simulations are carried out with the multibody orbital robotics simulator of Chapter 2.

The servicing satellite is already in a stable relative orbit at a safe distance form the client satellite but cameras are not pointed towards the target and all filters are reset. This scenario would represent a follow up of a safety mode exit and resume the nominal operation. The servicer would

acquire GPS signal from the target at slow data-rate and with some delays, estimate thus the position of the other through a Kalman filter and steer to point cameras towards the target. Then through camera estimation a precise inspection is conducted. All the images when the target is on sight can be used on board, or streamed down on Earth, to assess for damages and estimate, if close enough, the attitude of the other satellite. This aspect, as is not properly related to the GNC aspects of the operations, is not here addressed. The minimum requirement for the attitude control is to track a reference that has higher frequency content than usual as some inversion of rotations with respect to the inertial frame are expected.



(a) *Measurements* **(b)** *Simulated image example*

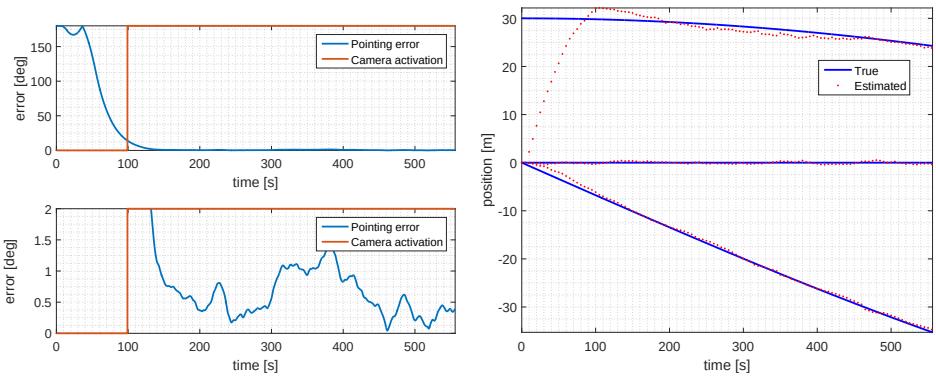
Figure 7.1: *Simulation sensor output*

Fig 7.1a presents the comparison between GPS simulated measurements and stereo camera estimation. GPS measurement has been obtained by computing the difference between on board GPS and GPS position transmitted by the other satellite. Uncorrelated white noise has been added on azimuth, elevation and altitude for each satellite separately, a delay of 4 seconds in the transmission and a delay compensation of 2 seconds: the resulting signal is thus affected by noise and a small phase delay.

Features are extracted from both frames, matched and filtered in order to compute a geometrical centroid in 3D space and use that as measurement. The target satellite for the simulation is between Earth and servicer, as can be seen from an example image in Fig 7.1b, thus the planet is in the background. Using a threshold it is possible to eliminate features that appears to have no displacement with respect to the stereo baseline. The simulated

stereo pair cameras have 30 degrees of semi-aperture and are turned on only when the estimated target position is within 15 degrees.

The camera switch, alongside the true pointing error are shown in Fig 7.2a. From that instant onwards camera estimate are added to the filter and refine the estimation. Here we can appreciate the attitude control capability to steer 180 degrees following a rather noisy signal. It can also be proven that in a close relative orbit the angular velocity profile needed to perfectly point towards target has a frequency spectrum with small high frequency content, thus some error is bound to happen even in a perfect scenario. Here the signal reference is obtained by determining the minimum relative rotation between the current and target direction in body frame then multiplied by estimated satellite quaternion and then filtered using a modified complementary filter in order to estimate at higher frequency both quaternion and velocity reference needed for the attitude control. This operation also filter the signal so that the reference has lower high frequency content and be more traceable by the controller. On the other hand this include a small delay and the pointing will have a small error regardless of the reference noise.



(a) *Pointing error* **(b)** *Estimation*

Figure 7.2: *Pointing & estimation for inspection test case*

Finally, Fig 7.2b compares the true relative position signal in the rotating frame against the estimated value coming from fusing the GPS and camera measurements with the Chloessy-Wiltshire model for the linearised system through a Kalman filter, partitioned to accept camera measurements only when the target is in sight as developed in Chapter 4. The outcome of this estimation is then used for attitude pointing.

7.2 Proximity operations

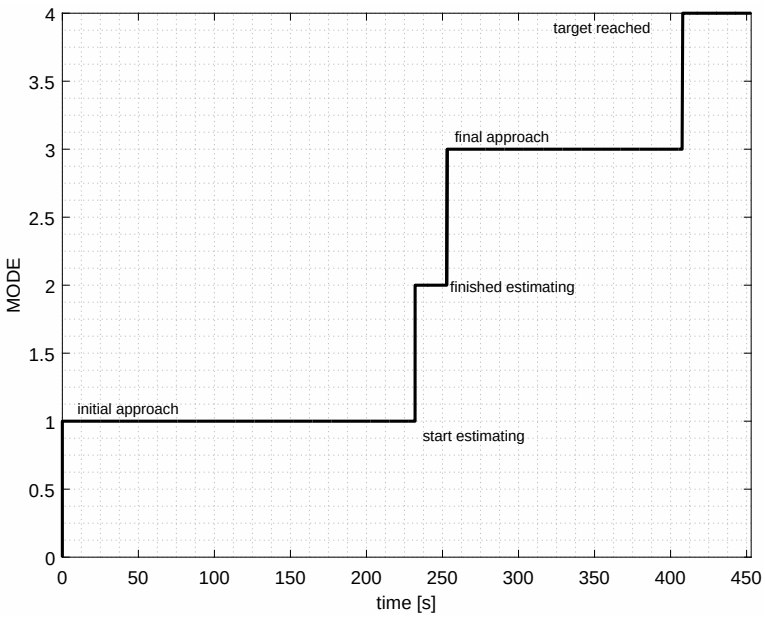


Figure 7.3: *Modes*

The main scenario to test the GNCR closed loop performance sees proximity operations performed by a servicer robot to a customer satellite in LEO. The satellite needs to autonomously station keep at few meter distance form the target, maintain relative attitude and operate the robotic arm to reach a docking port. The operative modes of the robotic arm sequence are correlated with time in Figure 7.3. The first task (1) sees the arm approach the target port at a hovering distance of 20 cm. The position of the port with respect to the target is known with an error of 2, 3 and 1 cm for x, y and z axis respectively. Once this stage is reached, the phase (2) starts and the camera on the robotic arm takes 21 pictures of the port and an estimation of the real location is made. Then phase (3) sees the final approach until the port is reached. The final stage (4) has been added to verify the capability of keeping the arm in position while connection is made. The threshold for reaching stage (4) is mainly related to the estimated distance from the target and is set to 0.5 centimetres. The effective error in positioning of the end effector depends mainly on the s/c ability to keep a fixed distance with relative estimation errors lower than 5 millimetres and below 0.5 degrees. For some applications the requirements could be even stricter and

require a more elaborate estimation scheme, but the goal set is meant to be representative for the early design phases of a GNCR subsystem for OOS.

Attitude results

The satellite attitude control makes use of 3 reaction wheels, one per axis, and a gyroscope and a star tracker for the attitude determination. Data on the base-only inertia parameters can be found in Table 7.1 alongside the parameters of the serviced satellite. The latter is also feedback controlled but with the intent to have nadir pointing.

The quaternion and angular velocity measured are then compared to the complementary filter output in Figure 7.4. What is possible to appreciate is the reduction in maximum error for the attitude estimation once the biases of the gyroscope are estimated. The bias rejection is self-evident in Figure 7.4b. The initial small drift of quaternion estimation is due to the integral action of the filter to reject the bias and is present only at the beginning of operations.

Table 7.1: *Satellites physical properties*

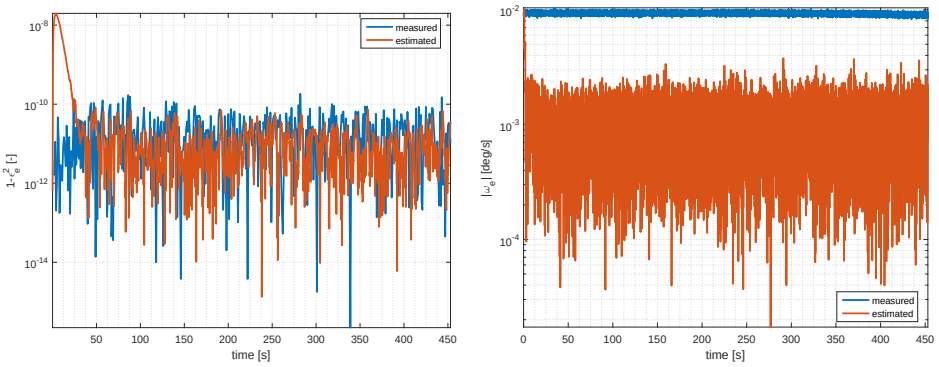
	Servicer base	Client
m $[\mathrm{kg}]$	1500	1000
$\mathbb{I}_{xx}$ $\left[\mathrm{kg}\cdot\mathrm{m}^2\right]$	250	500
$\mathbb{I}_{yy}$ $\left[\mathrm{kg}\cdot\mathrm{m}^2\right]$	250	500
$\mathbb{I}_{zz}$ $\left[\mathrm{kg}\cdot\mathrm{m}^2\right]$	250	500

Table 7.2: *Attitude controller parameters*

Sampling Frequency $[\mathrm{Hz}]$	100	Torque saturation $[\mathrm{Nm}]$	1
Velocity cutoff frequency$[\mathrm{Hz}]$	1	Adaptive gain$\left[\mathrm{s}^2\right]$	100000
Guidance loop frequency$[\mathrm{Hz}]$	0.1	Deadzone wide $\left[{}^{rad}/\mathrm{s}\right]$	$2\cdot10^{-5}$
Reference inertia $\left[\mathrm{kg}\cdot\mathrm{m}^2\right]$	250	Inertia limits$\left[\mathrm{kg}\cdot\mathrm{m}^2\right]$	50-1250

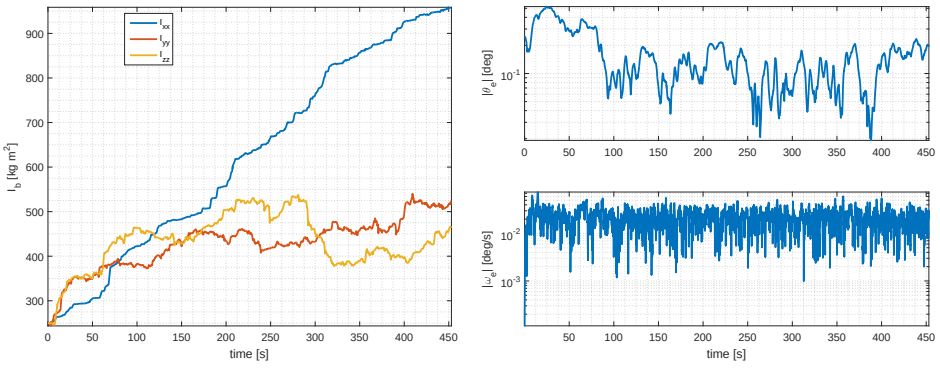
The torques required for the attitude control are computed using the adaptive control law of Chapter 5 and the optimal quaternion guidance of Chapter 4. The parameters shown in Table 7.2 were used for the simulations. The true error computed using true values and the reference fed to the controller, hence without estimation errors of Figure 7.4, can be seen

in Figure 7.5b. The attitude controller is thus able to keep the system stable against disturbances coming from the position control and the robotics combined. it is also clear that for these simulations the integral action of the controller has been switched off and the errors in both attitude and velocity is stable but not tending to zero. Even with the integrative term such behaviour would kick with a settling time higher than the whole manoeuvre. On the left, in Figure 7.5a, we can see the inertia parameter estimated by the controller through the whole simulation. It is possible to appreciate the capability of the estimator to reduce the gains if the output is higher than necessary.



(a) *Attitude estimation error* **(b)** *Angular velocity estimation error*

Figure 7.4: *Complementary filtering performance*



(a) *Attitude adaptive inertia* **(b)** *Attitude control error*

Figure 7.5: *Attitude control performance*

Robotics results

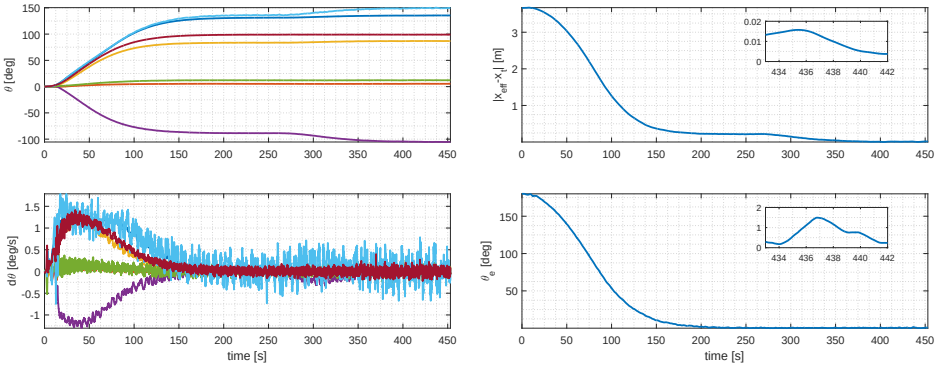
Table 7.3: *Robotic arm parameters*

	1	2	3	4	5	6	7
link mass $[\mathrm{kg}]$	2.12	42.4	4.24	16.96	2.12	2.12	1.06
link length $[\mathrm{m}]$	0.1	2	0.2	0.8	0.1	0.1	0.05
link max inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	0.003	14.16	0.017	0.915	0.003	0.003	0.001
link min inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	0.003	0.053	0.0053	0.0212	0.003	0.003	0.001
Abs. Enc. max err $[\mathrm{deg}]$	0.088	0.088	0.088	0.088	0.088	0.088	0.088
Encoder max error $[^{deg}/_{\mathrm{s}}]$	0.088	0.088	0.088	0.088	0.088	0.088	0.088
Max joint position $[\mathrm{deg}]$	180	90	180	0	90	180	180
Min joint position $[\mathrm{deg}]$	-180	-90	0	-180	-90	0	-180
Max joint velocity[1] $[^{deg}/_{\mathrm{s}}]$	1.2	1.2	1.2	1.2	1.2	1.2	1.2

Table 7.4: *Robotic arm controller parameters*

	1	2	3	4	5	6	7
Reference inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	7.5	3	1.5	3	1.5	0.15	0.6
Maximum inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	300	300	30	300	3	0.3	0.3
Minimum inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Reference cutoff frequency $[\mathrm{Hz}]$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Velocity cutoff freq. $[\mathrm{Hz}]$	0.7	0.6	0.5	0.4	0.3	0.2	0.1
Torque saturation $[\mathrm{Nm}]$	1	1	1	1	1	1	1

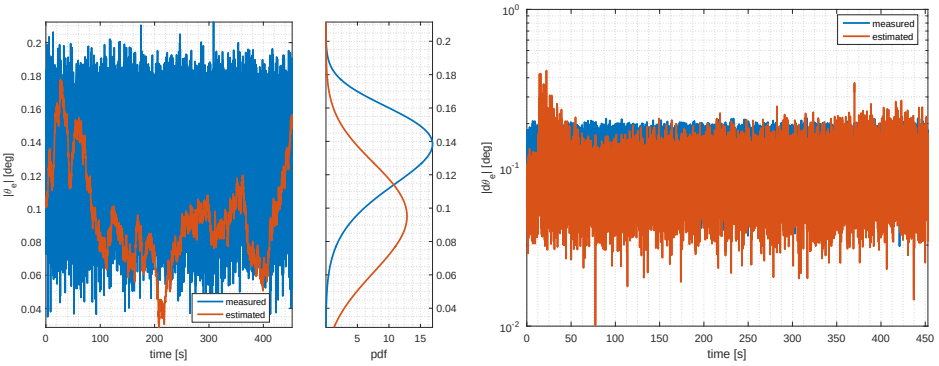
The robotic arm mounted on the satellite base has mechanical parameters shown in table 7.3 and consist of seven serial joints with alternate axis of rotation starting from axial (joint 1) then lateral (joint 2) and repeating till the last link that has axial rotation. Inertia parameters assumed are also listed in table 7.3 with also joints limits.



(a) *Joint path* **(b)** *End effector position and attitude error*

Figure 7.6: *End effector trajectory in joint space*

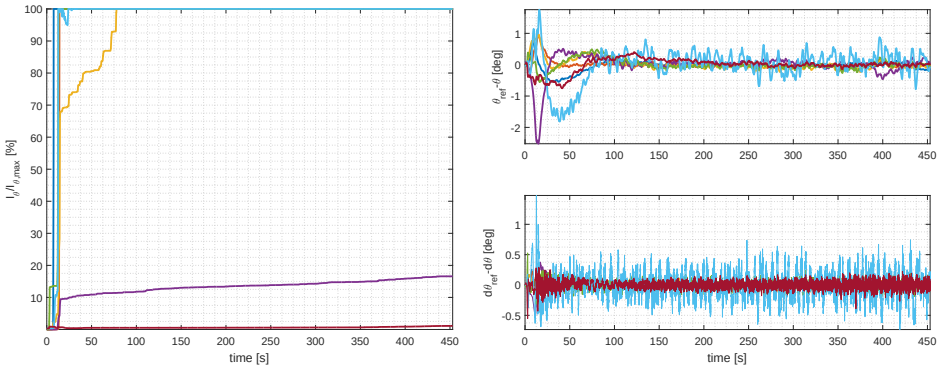
The path to reach the port is translated by the LM guidance of Chapter 6 to a smoothed reference. The actual path and velocities of the joints are shown in Figure 7.6a. In the same figure is also presented the distance and angular displacement of the end effector from the true port location. In Figure 7.6a it is possible to see the effect of velocity saturation in the trajectory of the joints. This testifies the reliability of the guidance system of the OR. Spikes and noise, especially in the first instants, of the velocity plot are due to the controller underestimation of the parameters. The chattering of joints velocity is due to the numerous sources of disturbance in forms of disturbing torques and noisy measurements.



(a) *Joint estimation* **(b)** *Joint velocity estimation*

Figure 7.7: *Estimation of joint state*

The filtering of the encoder measurements is shown in 7.7 and although



(a) *Adaptive Inertia* **(b)** *Joint control error*

Figure 7.8: *Robotics control performance*

it was not covered much, it has been used a complementary-like filter. The presence of noise is of the utmost importance when dealing with the adaptive control and to correlate velocity noises that translates back into torque disturbances.

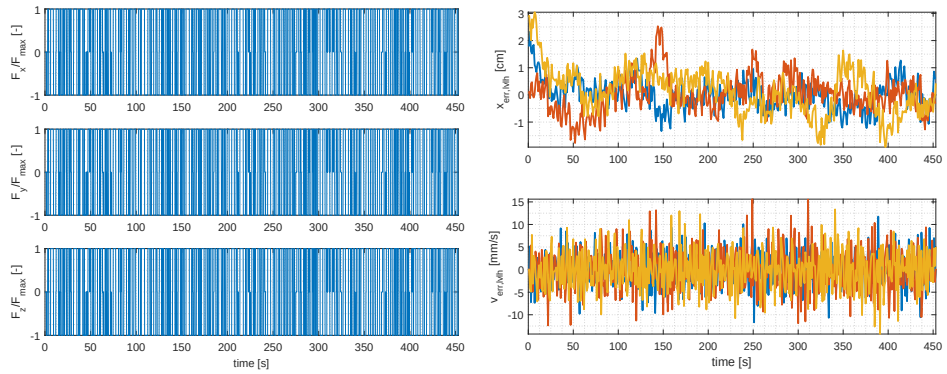
The performance of the adaptive controller tailored for the robotic arm are shown in Figure 7.8; its parameters are instead listed in Table 7.4. It is possible to see the error increasing at the beginning of the manoeuvre as the equivalent inertia has been severely underestimated. The inertia adaptation kicks in rather soon as testified by the increase, up to saturation, of some joints controller inertia as it is visible from the graph of Figure 7.8a. After the first rough tuning phase, the joints error are effectively kept in check by the controller for the rest of the simulation.

Position control

Table 7.5: *Position controller parameters*

Sampling Frequency [Hz]	100
Frequency[Hz]	0.5
Damping	5
Reference mass [kg]	1661
Force saturation [N]	4

Table 7.5 presents the parameters used for the relative position control. The force only request is then translated into thrusters firings whose activation



(a) *Thrusters firings* **(b)** *Position control error*

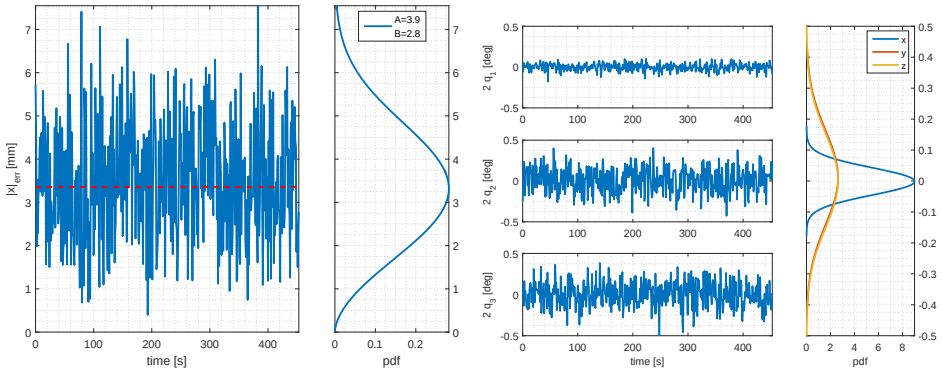
Figure 7.9: *Position control performance*

command can be seen in Figure 7.9a. Although it is rather difficult to see from the plot, in each firing area the forces are almost impulsive, as expected. The length of clustered firings varies through the simulation as the center of mass of the ensemble moves with respect to the base due to arm movement. The back and forth movement on the axis is better seen in Figure 7.9 looking at the position error. Such error is the error processed on-board, hence relative to the estimation error of Figure 7.10a. The error is around 1 centimetre per axis which is expected considering the 1N of thrust for each of the on-board thrusters. Being able to use modulable thrusters could really make the difference, although the use could probably be limited to GEO orbits as such thrusters are most likely to be electric. By increasing or decreasing the frequency of the position control we can have a more precise station keeping at the cost of more attitude disturbances.

Vision based estimation

Table 7.6: *Cameras Parameters*

	Left	Right	Arm
Field of view [deg]	60	60	60
n° Pixel	1024x1024	1024x1024	1024x1024
Framerate [fps]	1	1	1
Duration of activity [s]	always	always	20



(a) *Relative position estimation error* **(b)** *Relative attitude estimation error*

Figure 7.10: *Vision based relative state estimation performance*

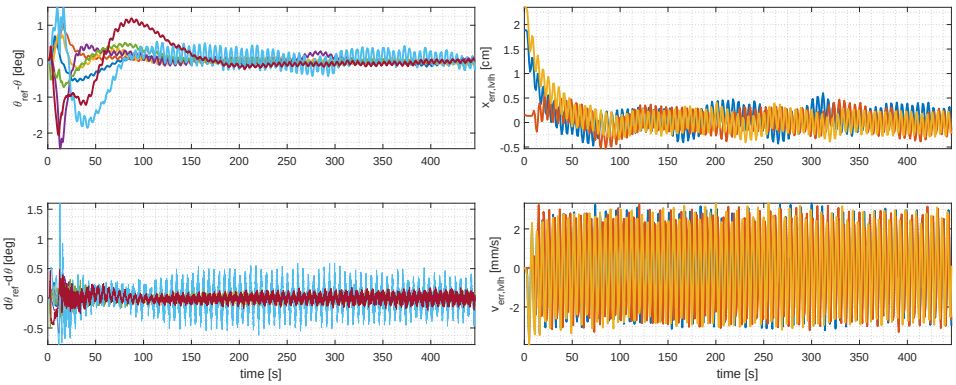
The images simulated with POV-Ray makes use of a simple camera whose data can be found in Table 7.6. The left and right camera can be used for stereo vision but in this simulation only the left camera is used for position estimation. The camera on the robotic arm is used only in Phase (2) and is mounted on the tip with a little displacement.

From Figure 7.10a we can see that the expected error with the algorithm presented in Chapter 4 is about 3 millimetres when relative distances are around 3 meters. The method used for the state estimation is rather raw but for relying only on 4 points it is still capable of a good amount of precision. As for attitude is concerned, Figure 7.10b presents the attitude estimation errors through small angle approximation from the error quaternion. Looking at the Gaussian probability density function of the error and the values for all the simulation it is clear that two axis are more prone to errors as one would expect from close distances.

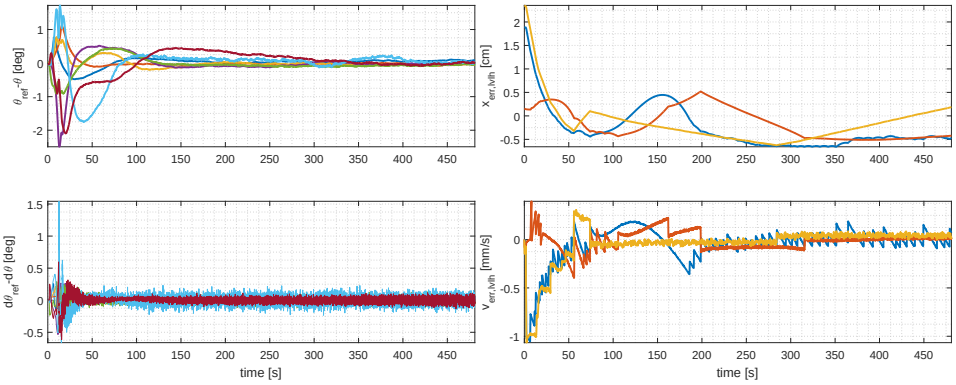
The port estimation during phase (2) drops the position error from 3.7 centimetres to 5.5 millimetres (lower per axis). The initial error is thus reduced by an order of magnitude and is in line with the position error of Figure 7.10a by which is also affected. A more rigorous estimation algorithm should be devised in order to reduce further the influence of the relative estimation. Increasing the number of images taken in phase (2) can reduce the effect up to an expected lower limit.

Datarate effects

Figure 7.11 shows a confront on the datarate of the relative pose estimation with perfect measurements. The first row shows the error that a perfect estimation of the relative pose would cause and one can appreciate the differ-



(a) *1 Hz*



(b) *10 Hz*

Figure 7.11: *Datarate analysis for relative pose*

Table 7.7: *Planar robot data*

	base	wheel	link 1	link 2	link 3
Mass $[\mathrm{kg}]$	1500.00	5.00	21.21	16.96	4.24
Inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	100.00	10.00	1.78	0.92	0.02
Characteristic lengths	2.00	-	1.00	0.80	0.20
Ref. Inertia $\left[\mathrm{kg} \cdot \mathrm{m}^2\right]$	100.00	-	25.00	2.00	3.85
Joint lower limits $[\mathrm{deg}]$	-	-	-180	-90	-90
Joint upper limits $[\mathrm{deg}]$	-	-	0	90	90

ence by comparing the two images with Figures 7.9 and 7.8b. As expected the one with more realistic measurement exhibits higher noise in velocity error. The second row of Figure 7.11 reports the position and joints error when the sampling time of the perfect vision based reconstruction is increased ten times and closer to the state of the art. The upsampling filters used for the slower version are not modified in weights and acts as simple lowpass filters. As one could expect, increasing the sampling time allows for a better velocity estimation and reduced efforts that consequently reduces errors in joint positioning. With a proper study and realization of the vision based algorithms it is possible to reduce errors and vibrations as well.

7.3 Comparison

The proposed methods for GNCR for OOS have shown good performance with a time varying model and in presence of noises and simulated images. The closed loop performance are satisfying, but a comparison with other methods can further show the proposed GNCR algorithms performance increase.

In order to show that a simplified planar model of the previous case has been prepared. The reference orbit is circular equatorial and the space robot consist of a base, a reaction wheel and 3 links that all have a vertical axis of rotation. This ensures the motion to be planar and the disturbs due to gravity gradient stays planar as well. Target physical properties and control are the same as the previous case; this case also keeps unchanged sensors and estimation of the previous one in order to focus more on the adaptive control and guidance of the space robot. Relevant physical and control data are shown in Table 7.7.

Table 7.8: *Simulation cases*

Case	1	2	3	4	5	6	7	8	9	10	11	12	13
RW	☑	☑	☑	☑	☑	☑	☑	☑	☒	☑	☑	☑	☑
Thr	☑	☑	☒	☒	☑	☒	☑	☑	☑	☑	☒	☒	☒
Guid	LM	L	GJ	GJ	LM	L	L	L	LM	LM	GJ	LM	GJ
Att	☑	☑	☒	☑	☑	☒	☑	☑	☑	☑	☒	☒	☑
Mod	SD	SD	-	-	PW	-	SD	SD	SD	SD	-	-	-
Adpt	☑	☒	☒	☒	☑	☑	☒	☑	☑	☑	☑	☑	☑

Should be noted that the reference inertia of link 2 has been deliberately underestimated to show the capability of the adaptive controller to reduce error with a poorer parameter estimation. It should be clear that an adaptive controller has almost the same performance as the non adaptive version if the parameters are estimated with high precision. The base attitude control, on the other hand, has a good initialization of the inertia in order to show the adaptive control enhancing performance when inertia changes due to arm movement and with actuators errors.

Another issue is the modulation, hence a comparison between SDM and PWM is performed using the same sampling time of 0.1 seconds. Unlike [86], here will be shown that PWM is more consuming with respect to SDM.

The last part to be tested is the robot guidance. Three different algorithms will be tested and differences will be shown. The LMG with filtered input is compared with classical velocity guidance (here named Lyapunov Guidance or LG for short) and a modified version of [22] where Generalized Jacobian Guidance (GJG) is used. In order to better show the latter algorithm capabilities the rendezvous approach is on v-bar, meaning the direction where drift due to orbital motion is lower and an approach that request no position control can be used. Unlike the first one, the other two algorithms generates a velocity reference and do not require a joint position control hence the robot controller will not use position as feedback when these two guidance algorithms are used.

The comparison and the following graphics are generated with different test cases. The correspondence between cases, algorithms and modes is listed in Table 7.8.

The simulations see the robotic arm deploying from a stowed position and reach a target position on the other satellite. Each simulation concludes

if the end effector positioning error goes below 1 millimetre for five seconds straight or if a maximum time (a tenth of the orbital period) has passed. This level of precision is reachable in this simplified scenario because the relative state estimation is perfect.

Adaptive vs non adaptive control

Figure 7.12 Shows the confront of the proposed GNCR algorithm with its non-adaptive version. The difference is not extreme but it is clearly shown in attitude and joint angles errors where CASE 7, the non adaptive version, has clearly an higher error through the whole simulation. The difference can be seen also on the velocities plots, however is less readable from the graphs without filtering results, an operation avoided to maintain uniformity in all results. Although the non adaptive case seems to be faster, should be noted that meeting the requirement for stopping condition when errors in control are higher may be pure and sheer luck.

What can be drown from this comparison is that adaptive control is effectively able to reduce errors without increasing much the effort. There are up to three figure of merit with respect to effort and those are the fuel mass expense, the arm effort and the reaction wheel velocity profile. Time can also be included, however the first two figure of merit are result of an integration, hence time is indirectly included.

The fuel mass is computed through integration of the mass flow rate computed from the famous rocket equation using the thrust and specific impulse written in the figure. This is not the straightforward mass expense and is not to be read as the effective mass consumption but as a figure of merit as the above mentioned quantities are kept constant for all simulations. Moreover, in order to increase readability, the expense is computed using the total thrust commanded from the modulation and not the effective thrust produced which includes also disturbs in the actuation process.

The arm effort is the integration of the norm of the vector of commanded torques to the robotic arm joints. The energy usage of arm control is linked to the power budget and not on the mass budget of a mission, but since the estimation would be more cumbersome than explanatory, here is preferred this version equally readable and understandable.

The reaction wheel velocity profile is not shown because there is no sensible difference.

The difference between adaptive and non adaptive control is more clearly seen when using a less refined guidance strategy for the robot. The reported Lyapunov stable guidance law is used here and fed to the same controller,

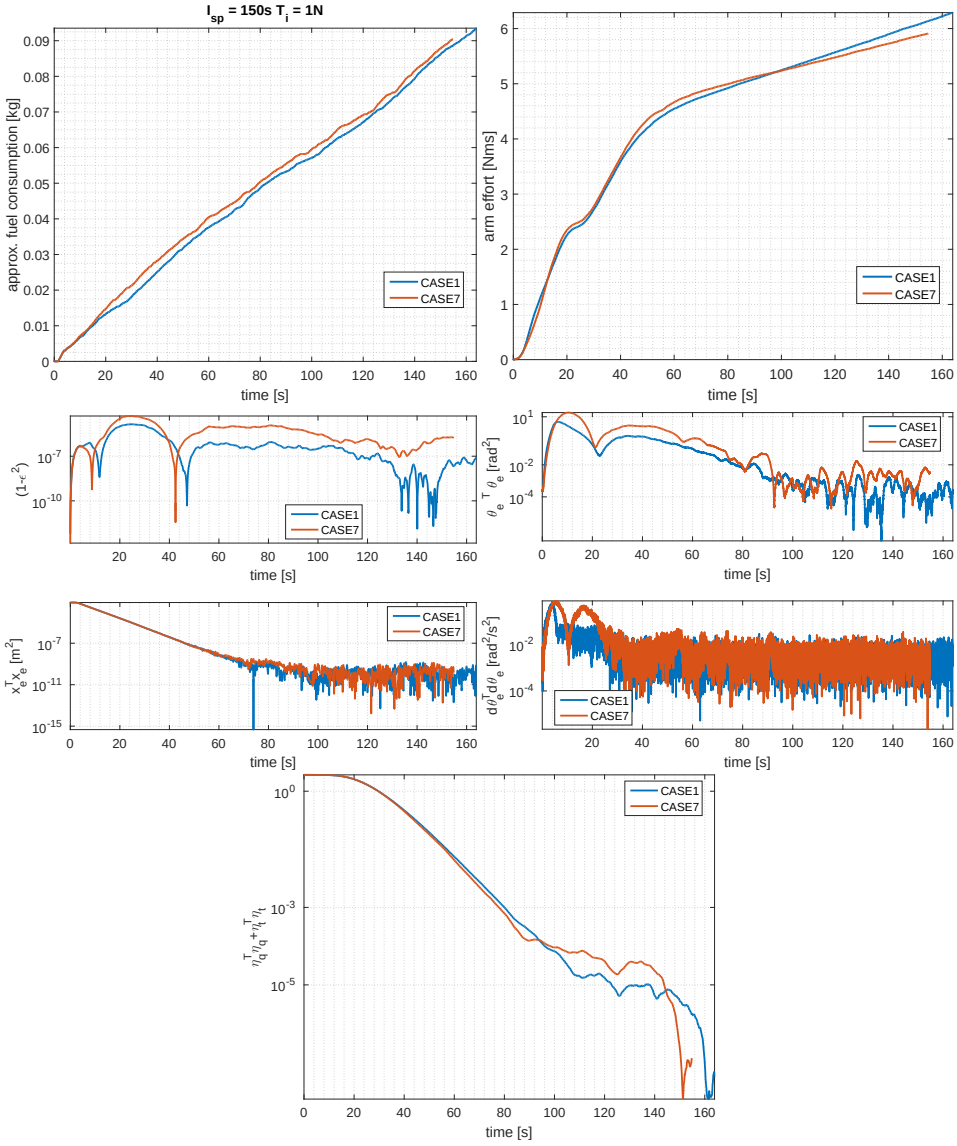


Figure 7.12: *Adaptation in proposed GNCR algorithm*

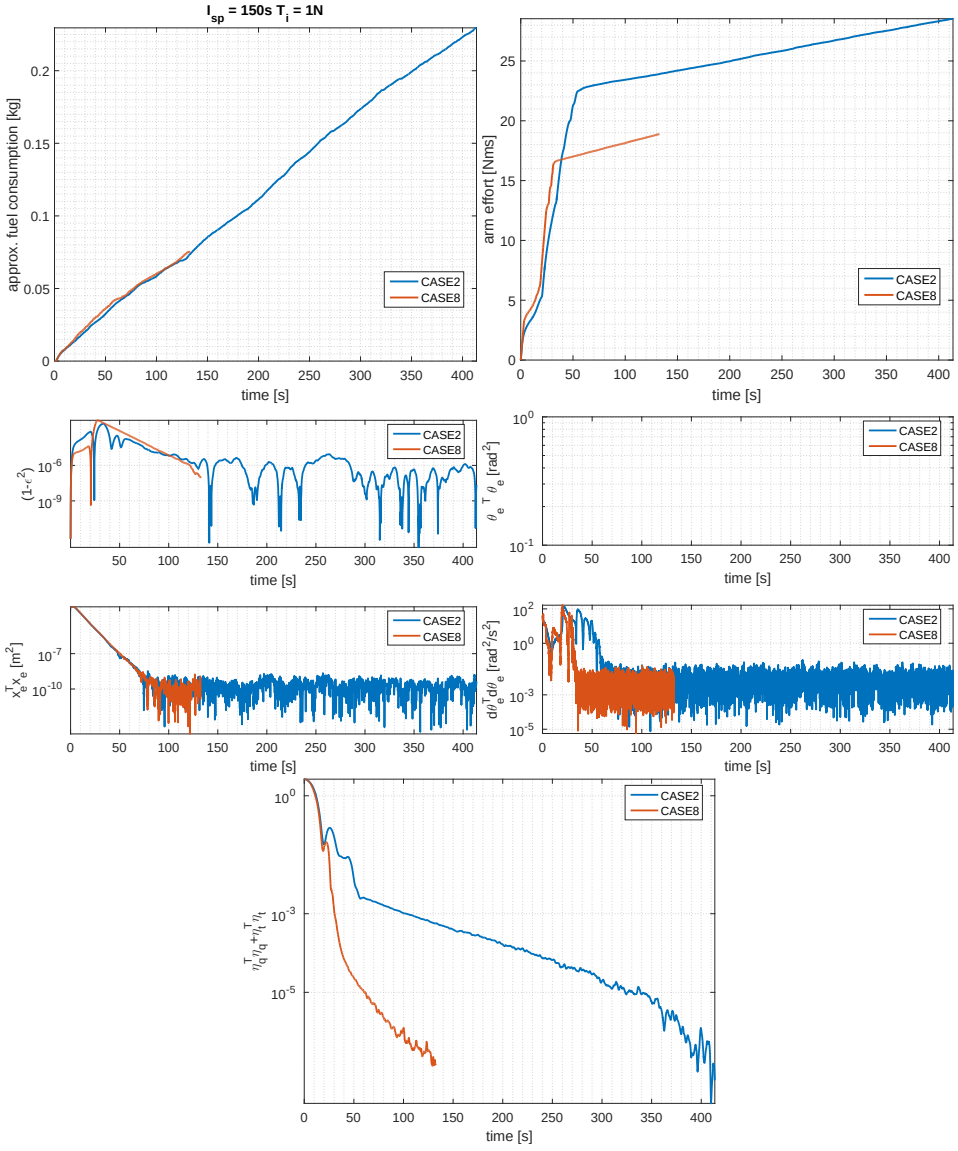


Figure 7.13: *Adaptation with classical robot guidance*

although no position or acceleration information coming from the guidance. Hence, position and acceleration constraints are not enforced and velocity constraints are satisfied by scaling the velocity reference vector in such a way that the maximum value of velocity lies on the boundary for any given computation.

This means that the reference is less smooth and followable than the previous case and this result in degraded performance of the non adaptive control. This implies that LMG through the filtering of the reference is able to reduce the uncertain effects. Also, the previous guidance adds more information and that is able to reduce errors as well.

In Figure 7.13 is presented the staggering difference that an adaptive control brings in the picture. Thanks to the adaptation the end effector is able to reach the target way faster as shown by the Lyapunov function of the vector dual quaternion quantities (η_t, η_q) in the lowest figure. The adaptive control is able to overcome the poor initial estimation of one parameter and adjust accordingly, showing the great difference in terms of time and effort.

Interestingly, yet expected, the mass usage by thrusters has a similar trend in both cases because the position control is non-adaptive and thus the sooner the target is reached (grasping for example) the sooner the position control can be switched off. The error trend (x_e) is of course very similar. More interesting is the reduction of control effort for the robotic arm, since in the non adaptive case in order to try to reach a good level of precision an higher torque is required.

Robotic guidance confront

A representative comparison between the proposed and reported guidance systems for orbital robots can be found looking at Figure 7.14 where CASE 1, 8 and 11 are compared. All of them uses the adaptive control and the one with GJG approach has no position or attitude control involved, as per natural application of its capabilities.

From a quick look one can see that even when the approach is on the V-bar the GJG takes too much time to reach the target and finally it slips away from reach, as seen in the last figure where error in end effector position increases towards the end. The drift is caused by a misalignment of the center of mass from the theoretical stationary point of few centimetres, which can be expected normally. Hence the classical implementation of the GJG , although appealing from the effort point of view, might require an unbearable increase in velocity of the arm or adjustment.

Then, looking at the confront of CASE 1 and 8 one can see the differ-

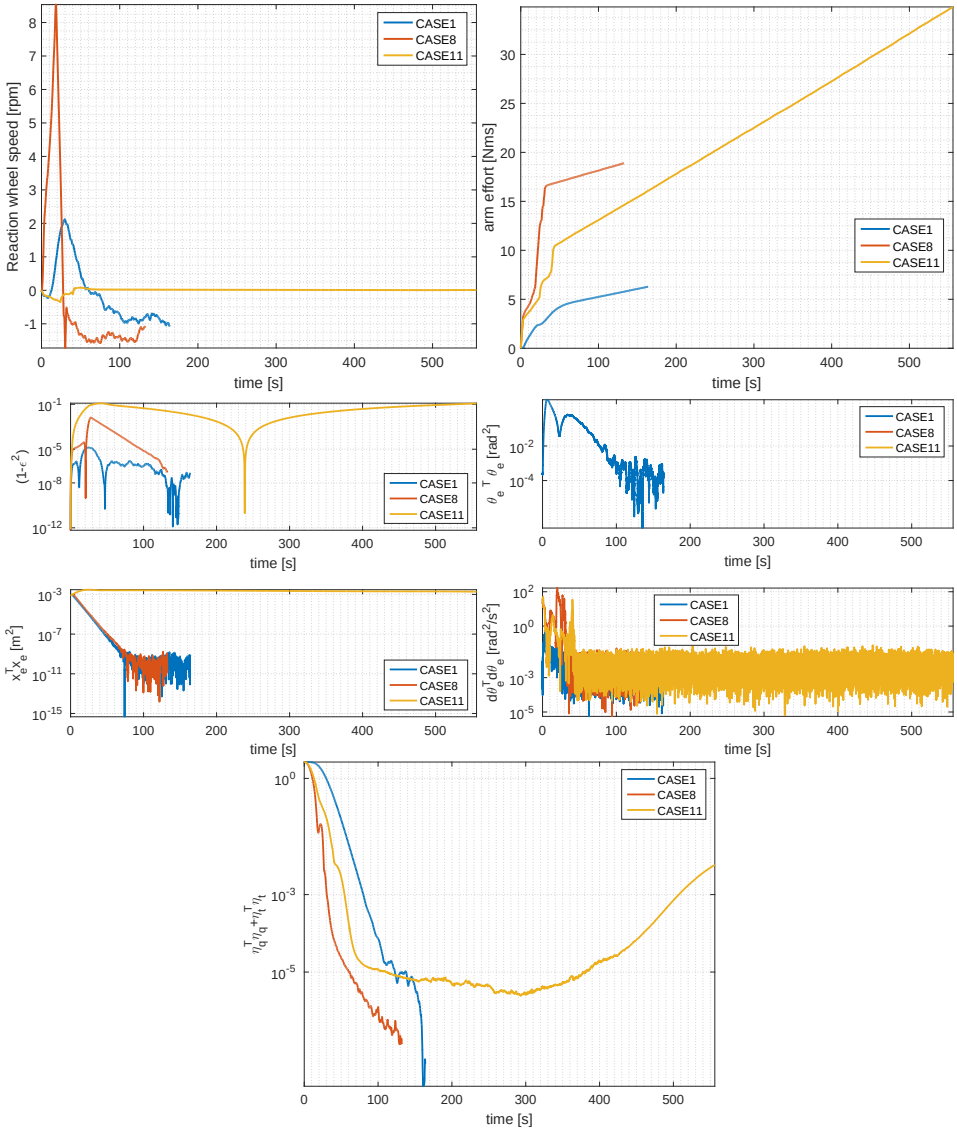


Figure 7.14: *Robot guidance strategies with adaptive control*

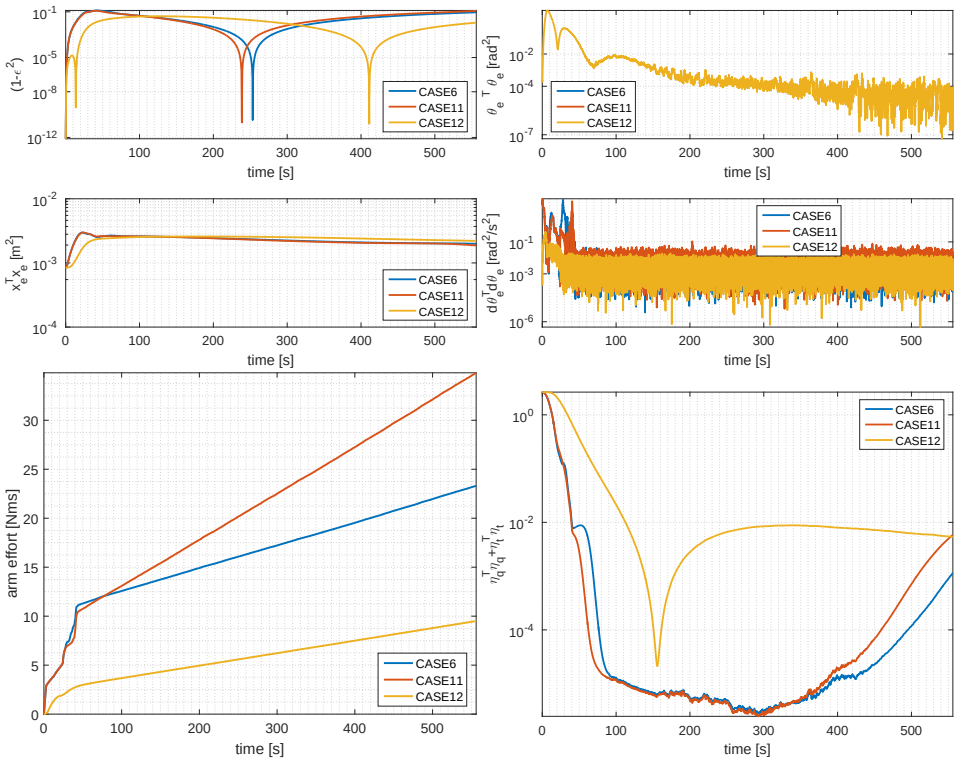


Figure 7.15: *Robot guidance strategies without position and attitude control*

ences induced by the two guidance systems. The classical Lyapunov velocity guidance of CASE 8 disturbs way more the attitude of the system and as a consequence the effort of the robotic arm is greater than the one of the LMG. On the other hand the L guidance provides a faster convergence of the end effector towards the target as expected from its Lyapunov analysis and from the delays introduced by the filtering in the LMG version. Such delay can be adjusted but at the cost of higher error in the early stages or by using more expensive and powerful avionics to increase the computational power of the system.

Fuel expense is not reported because is quite similar and depends mostly on the duration of the manoeuvre.

To give a better perspective on the near-zero effort perspective of the GJG, a comparison of methods is presented in Figure 7.15. The cases in exam are 6, 11 and 12, all with adaptive control included for robotic arm, but no control in position and attitude.

The GJG and the Lyapunov based guidance behaves closely to the LG (and is implemented as a variation where the Jacobian of the LG is modified

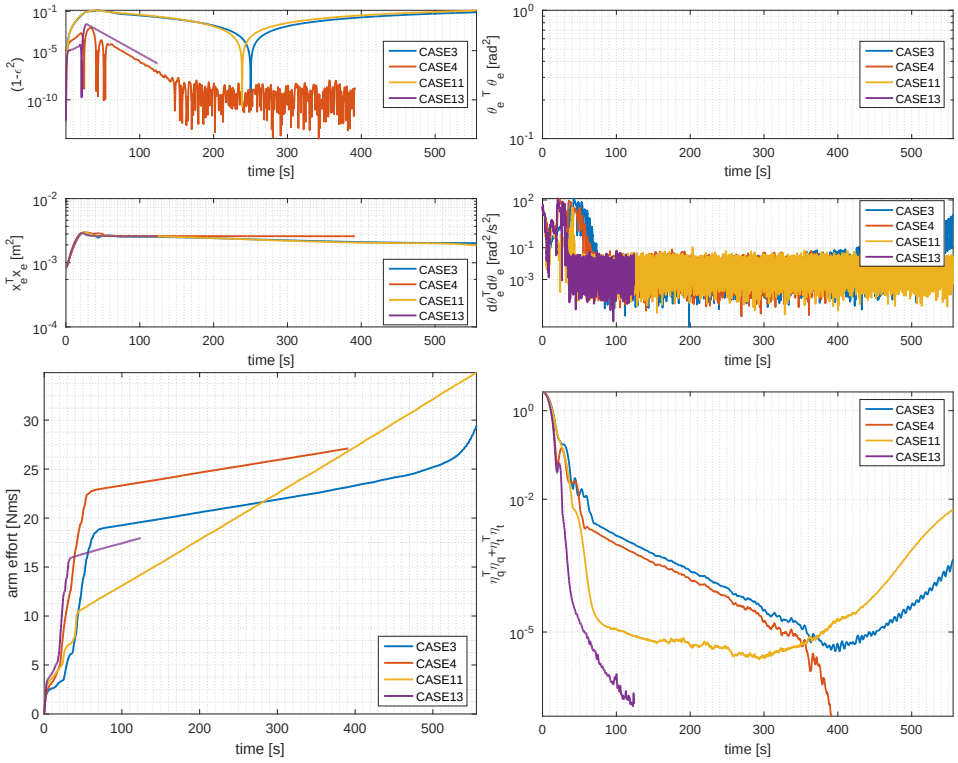


Figure 7.16: *Generalized Jacobian guidance strategies*

using the information of angular momentum) but has a faster convergence due to the incorporation of the base expected motion through angular momentum conservation. Should be noted that even in this simple case the assumption of conservation of angular momentum is not verified as gravity gradient due to a distributed mass system is still present. The slight improvement of GJG with respect to the L guidance is mostly due to this fact. In this case, neither method can achieve the desired target in time and then it slips away.

The LMG behaves poorly with respect to the other two methods because of inherent slower tracking of moving target with respect to its L guidance counterpart as expected by previous comparison.

Finally, modified versions of the GJG are included to show that by adding attitude control, through reaction wheels, it is possible to reduce the time to reach the target before it slips away. CASE 3, 4, 11 and 13 are presented in Figure 7.16.

CASE 3 and 11 do not use attitude control but 11 has adaptive control

for the robotic arm and the results are impressive as it is able to reach a minimum distance from the target sensibly before the non-adaptive counterpart. Even here the adaptive control shows the increase in performance with lower effort, although in both cases the target is not reached in time. The expense of the two cases becomes equal as both start to slip away, however the adaptive version reaches a lower error before the other case and with lower effort.

The simple addition of attitude control though reaction wheel changes the results and the system is able to reach the target in time. By looking at CASE 4 and CASE 3 it is possible to see that keeping the attitude more stable permits to reach the target in time whereas a non controlled case would not. Again, the addition of adaptation reduces the error way faster as shown here by the curves of CASE 13. Graphs are not generated, however there is a close matching between CASE 13 and CASE 8 with the L guidance instead. This is expected as both algorithms work using the same principle, however CASE 8 also control position and thus has a higher fuel expense. A modification of CASE 8 with no position control is not included but the results have very high correlation with CASE 13, hence L guidance and GJG in the end can provide the same results when attitude control is switched ON.

The GJG approach should be more performing but, as stated before, the conservation of angular momentum is never verified in reality and small modifications need to be made. Moreover it results less appealing from the robustness point of view as it requires a good model and errors in masses, inertia and lengths might reduce the performances. Such errors are not taken into account here but reduced performance are expected nevertheless. GJG approach, and similar approaches, were not developed for position feedback but rather to follow a velocity profile, hence closing the loop with a target pose can yield similar results for L and GJG guidance algorithms.

Modulation & Thrusters considerations

Considerations about modulation for thruster activation is made here by looking at Figure 7.17 where CASE 1 and 5 are compared. The only difference is that CASE 5 uses PWM while CASE 1 uses SDM. Unlike [86] the sampling time of both is the same and taken as 0.1 seconds. The result shown here is the increase in effort and error with the PWM as one would expect if both sample the signal at the same frequency. In [86] the PWM sampling time appears to be 10 times slower than the other one so results might be affected by this difference. Should be noted that unlike [86] here

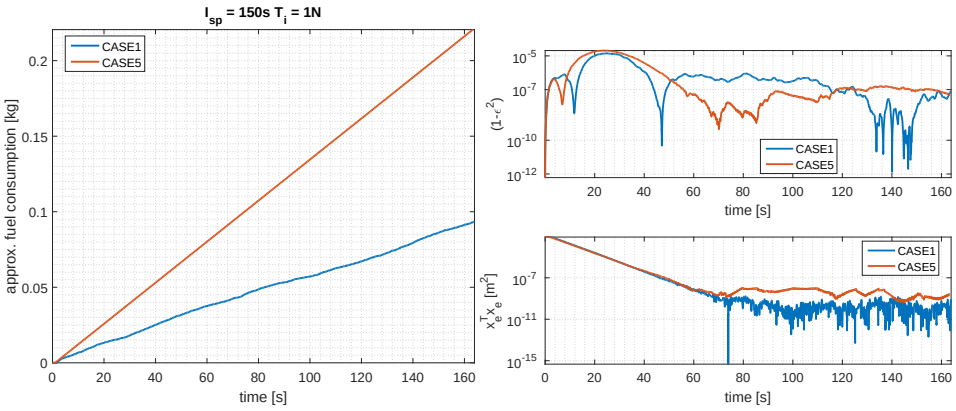


Figure 7.17: *PWM versus SDM performance at same sampling time*

the thrusters are used only for position control and not for attitude control.

For almost all cases of Table 7.8 the attitude is controlled with a reaction wheel, however the attitude can be controlled equally also using thrusters. Figure 7.18 compare CASE 1 with CASE 9 and the results are interesting. Through SDM the precision reached using thrusters (position and orientation errors are taken into account as per all cases) is on the same level as the one with a reaction wheel and with the same fuel expense.

This is caused by the fact that torques can be generated by a proper activation of thrusters by producing the same amount of net thrust. However, what is notable is the increase in the arm effort at convergence caused by the higher disturbs caused by base motion on the arm itself.

Even if the precision in attitude is the same as the one with reaction wheels, the use of thrusters induces locally higher torque impulses. A trade off between the two strategies can be made based on reaction wheel saturation level and power/energy on board, connected to the control effort of the robotic arm.

Comparison conclusion

Figure 7.19 sums up the improvement possible using the proposed adaptive control and robotic guidance scheme. In presence of parametric uncertainties and time varying quantities the adaptive controller allows both attitude and robotic arm to improve the convergence rate displaying better performance. Although the LMG filtered scheme is slower with respect to LG, it converge faster thanks to adaptivity and to lower disturbs transmitted to the base. The guidance scheme is also able to reduce the control effort regardless from adaptation as the filtered output and the consequent feedforward

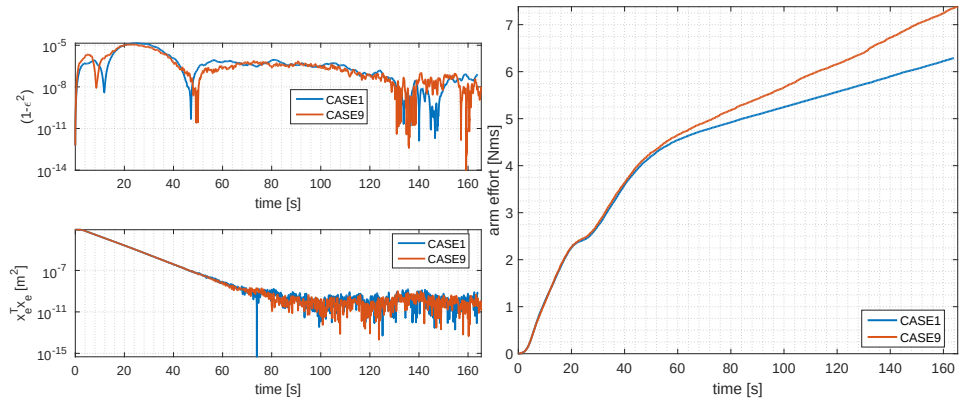


Figure 7.18: *Attitude control actuation method differences*

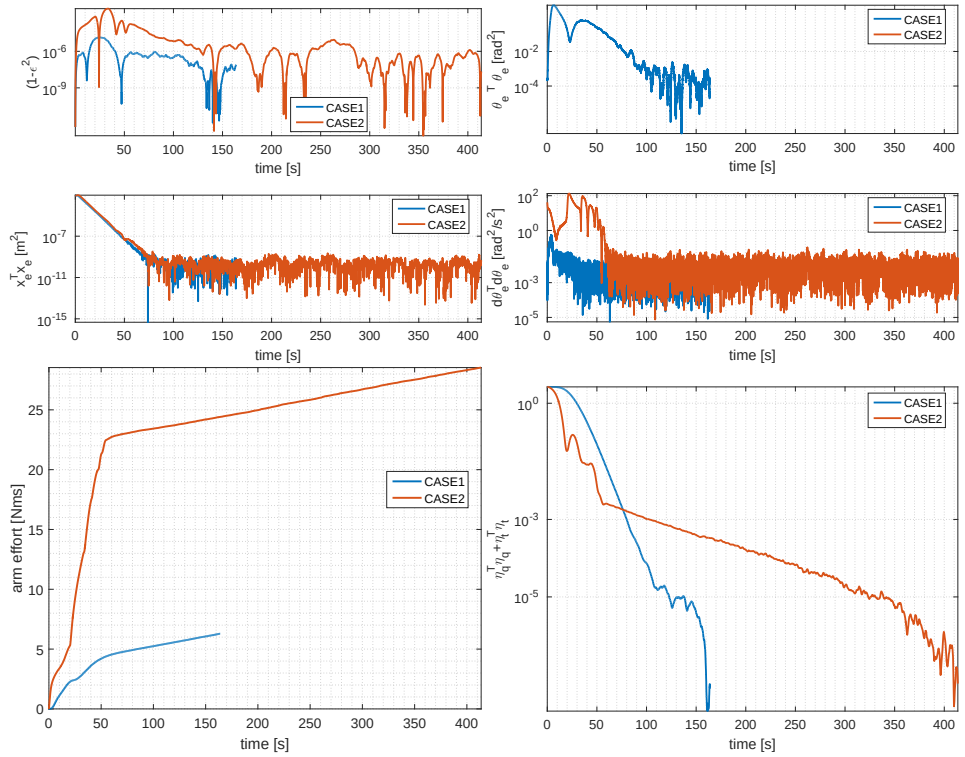


Figure 7.19: *GNCR Improvements*

permits to follow a more compliant reference signal. The proposed scheme requires position and attitude control to be active for all the manoeuvre in order to give the desired performance; such hypothesis is coherent within a practical context as the previously shown comparison highlights difficulties in using a free floating approach even when approaching from the less demanding tangential direction.

Conclusions

The first OOS commercial mission will mark the beginning of a new era in the space industry and possibly be marked as a *revolution.* This might sounds excessive at first, but if one looks at all possible applications and the radical shift in the design of satellites that will happen as a direct consequence of the serviceability paradigm the definition of revolution is not at all resounding. While big players and small disruptive endeavors fight against the current system in order to establish OOS, research in many fields has to follow.

In OOS the intertwining of multidisciplinary area of interest is so deep that one can hardly disjoint them during research. Any OOS mission would require a deep planning of a servicer satellite lifetime in order to be profitable, hence mission analysis optimization and economical analysis should be looked at together. Another example is the use of cameras in close proximity hovering applications that could also generate issues about property rights. Before diving into the multidisciplinary pool a researcher, as well as an entrepreneur, must first look at his own skills in order to find the best way to contribute to this revolution.

According to that principle, this thesis represents the analysis on one core aspect of OOS operations: the GNCR subsystem that provides the means to deliver the OOS service. Literature on this specific topic, especially covering the whole control loop, is rather scarce and often some parts of the loop are overlooked. What happens then is that actual performance are below the expected values or lack trustworthiness, especially when cameras are considered as sensors.

The first step, presented in Chapter 2, is to simulate the mechanical system at the highest level of detail permitted by hardware and engineering judgment. In this work a multibody software with capability of simulating many sensors, including cameras with photorealistic features, has been coded, validated and used to simulate a space robot. Equations of motion have not been linearized and for the timeframe and orbits considered delivers expected performance.

The second step was to prepare all blocks that compose the GNCR loop in order to perform a full simulations. This required to plunge into the guidance and control of satellites and robots, but also to look at attitude estimation and relative vision based estimation techniques. Neglecting one of these aspects would have compromised the final analysis and peering too much into the depths of each field would have made impossible to get the big picture. Hence, focus was given to characterize all parts at the highest level of precision for a preliminary assessment of the GNCR system. For example, an effective computer vision algorithm for relative pose estimation requires years of development and many people working on it before it gets to the required level of precision for real OOS applications. However, overlooking the effect of this part on the relative control would not expose the frequency estimation problematic and subsequent problems of closed loop performance.

One of the main contributions in subsystems is probably the Principal Inertia Adaptive Dynamic Inversion Controller developed in Chapter 5 also applied to robotics in Chapter 6. The strife was to find a suitable adaptive scheme that would not require estimation of many parameters and was easily implementable. Being de facto a PD controller with adaptive derivative terms, it is more than suitable to be looked at in the industry. The main reason behind this study is the acknowledgment that in many OOS scenarios considered the servicer satellites undergoes geometrical variation or mass/inertia variation due to docking/capture. All possible sources of errors have been pointed out (noise, drift, overestimation, stability) and verified through simulations and the structure has been simplified at the maximum level possible. The controller can be further improved and connected to a robust estimation of the nominal parameters and performance.

In Chapter 7 few simulation studies, making use of all that has been developed from previous chapters, were presented and results analyzed. Vision sensors prove to be effective in the estimation of target position and pose with encouraging level of precision. However, precision is not the only concern, as the robustness of image processing in space is a huge problematic. What is recommended to possible OOS provider/OOS system

manufacturers is to spend time in preparing a robust vision system capable of doing both pose estimation in close proximity and centroid estimation for long range approach. The two approaches presented here are compatible but would require more study to provide a fusion of the two that is able to guarantee the level of robustness capable of not jeopardizing the OOS mission they were designed for. In general it has been seen that, with the provided control architecture, it gets fundamental to reduce the noise in velocities estimation, especially for robot control. This would reduce the effect on flexibility and cross-system frequency interference. For example, the frequency of the position control (force determination and thrusters wise) has huge impact on the attitude control and the robotic arm movement shall be slow enough to keep the adaptive controller in the global convergence region. The frequency content of position control is then influenced by the pose estimation sampling time, which should be increased hardware wise and not only through a upsampling process (that can either add delays or high frequency noises).

The bottom line of this consideration is that the GNCR system is heavily coupled and it is easy to reduce global performance by overlooking a simple aspects in one of the loop parts or the influence on other subsystems. It is here stated and stressed the need for complete loop simulations when dealing with OOS scenarios because, unlike many commercial scenarios, the GNCR system plays a key role for the economical growth and stability of a OOS provider company. In the end the company that delivers the best and more reliable OOS service will be the victor as consequences of a failed OOS mission would be catastrophic especially in the erly development stages of the next two decades.

APPENDIX $\mathcal{A}$

Dual Quaternions

This Appendix is meant to provide the basics of quaternions and dual quaternions algebra and matrix notation. This extract has been presented before in [18] to introduce the GNCR dual quaternion based framework. Let us start by reviewing what a quaternion is.

A quaternion $\mathbf{q}$ is a four-element entity belonging to a space where three components are imaginary. The group of quaternions is defined as

$$\mathbb{H} = \{\mathbf{q} = q_4 + iq_1 + jq_2 + kq_3 : \; q_4, \, q_1, \, q_2, \, q_3 \in \mathbb{R}\} \tag{A.1}$$

where $i, \; j, \; k$ follow the rules of Eq. (A.2).

$$\begin{cases} i^2 + j^2 + k^2 = -1 \\ ij = k = -ji \\ jk = i = -kj \\ ki = j = -ik \end{cases} \tag{A.2}$$

In this group addition and multiplication are defined and form a skew field, often called "Division Algebra". It is possible to represent a quaternion using a $\mathbb{R}^4$ vector so that

$$\mathbf{q} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\eta} \\ \epsilon \end{Bmatrix} \tag{A.3}$$

In the latter representation it has been introduced also the separation of the scalar real (ϵ) and vector imaginary $(\boldsymbol{\eta})$ part of the quaternion as will be particularly helpful later on. The use of the vector and matrix forms of quaternions allows for a easier understanding of the underlying nature of their representation capabilities.

The quaternion group is closed with respect to the summation, subtraction, multiplication and division. Multiplication and division must be handled in a peculiar way due to the rules of (A.2). A quaternion product can be written as

$$\mathbf{q}_c = \mathbf{q}_b \otimes \mathbf{q}_a = \left[\mathbf{q}_{b\overline{\otimes}}\right] \mathbf{q}_a = \left[\mathbf{q}_{a\overset{+}{\otimes}}\right] \mathbf{q}_b \tag{A.4}$$

where the matrix forms $\left[\mathbf{q}_{\overline{\otimes}}\right]$ and $\left[\mathbf{q}_{\overset{+}{\otimes}}\right]$ are no other than canonical left and right transformation or Hamilton operators. Should be noted that a symmetric definition of the quaternion product would be fine as well, since the result is still a quaternion. However the consistency must be maintained through the whole development. The matrix notation has been different through literature, the form adopted here is meant to suggest the matrix form of a general quaternion and the duality of the product representation. In terms of components those four by four matrices are given as

$$\left[\mathbf{q}_{\overline{\otimes}}\right] = \begin{bmatrix} \mathbf{I}_{3\times3}\epsilon - \left[\boldsymbol{\eta}_{\times}\right] & \boldsymbol{\eta} \\ -\boldsymbol{\eta}^T & \epsilon \end{bmatrix} \tag{A.5}$$

$$\left[\mathbf{q}_{\overset{+}{\otimes}}\right] = \begin{bmatrix} \mathbf{I}_{3\times3}\epsilon + \left[\boldsymbol{\eta}_{\times}\right] & \boldsymbol{\eta} \\ -\boldsymbol{\eta}^T & \epsilon \end{bmatrix} \tag{A.6}$$

where $\mathbf{I}_{3\times3}$ is the identity matrix and $\left[\boldsymbol{\eta}_{\times}\right]$ is the skew matrix form of the vector $\boldsymbol{\eta}$[1]. To better comprehend the duality of the two matrices let us look at the components of the quaternion multiplication (A.4)

$$\begin{cases} \boldsymbol{\eta}_c &= \epsilon_b\boldsymbol{\eta}_a - \left[\boldsymbol{\eta}_{b\times}\right]\boldsymbol{\eta}_a + \epsilon_a\boldsymbol{\eta}_b \\ \epsilon_c &= -\boldsymbol{\eta}_b^T\boldsymbol{\eta}_a + \epsilon_a\epsilon_b \end{cases} \tag{A.7}$$

[1]Cross product and notation: $\boldsymbol{\eta}_a \times \boldsymbol{\eta}_b = \left[\boldsymbol{\eta}_{a\times}\right]\boldsymbol{\eta}_b$

Rearranging the terms and remembering the property of the cross product $\boldsymbol{\eta}_a \times \boldsymbol{\eta}_b = -\boldsymbol{\eta}_b \times \boldsymbol{\eta}_a$ we can see that the duality of the two matrix form with respect to multiplication is indeed verified. This keeps being valid for any extension that will be made hereafter.

The division operation follows quite straightforwardly from (A.4)

$$\begin{cases} \mathbf{q}_a &= \left[\mathbf{q}_{b\overline{\otimes}}\right]^{-1} \mathbf{q}_c \\ \mathbf{q}_b &= \left[\mathbf{q}_{a\overset{+}{\otimes}}\right]^{-1} \mathbf{q}_c \end{cases} \tag{A.8}$$

where it can be shown that

$$\left[\mathbf{q}_{b\overline{\otimes}}\right]^{-1} \mathbf{q}_c = \frac{1}{\mathbf{q}_b^T \mathbf{q}_b} \left[\mathbf{q}_{b\overline{\otimes}}\right]^T \mathbf{q}_c \tag{A.9}$$

Being de facto an inverse multiplication there is also another matrix form that allows to switch the position of $\mathbf{q}_c$ in the equation above.

Unitary quaternion and attitude representation

We already know that a unitary quaternion can represent the attitude of a given body. From the quaternion group it is possible to consider a subgroup consisting of quaternion with unitary norm. In this group the norm considered is the root of the squared sum of quaternion components. Briefly the condition that must hold for unitary quaternion is

$$\|\mathbf{q}\| = 1 = \boldsymbol{\eta}^T \boldsymbol{\eta} + \epsilon^2 \tag{A.10}$$

This allows the quaternion multiplication to be considered a rotation of a given angle around the Euler axis. Limiting the quaternion to lie on the $\mathbb{R}^4$ unitary sphere permits to gain few properties that can be exploited in rotation parametrization, but on the other hand it loses the operations of sum and subtraction: the unitary quaternion group is not closed with respect to sum and subtraction. Summing two quaternion with unitary norm will generate a quaternion with non unitary norm.

A unitary quaternion can be written in terms of Euler angle and axis as follows.

$$\begin{cases} \boldsymbol{\eta} &= \mathbf{e} \sin \frac{\vartheta}{2} \\ \epsilon &= \cos \frac{\vartheta}{2} \end{cases} \tag{A.11}$$

where $\mathbf{e}$ is the Euler axis and ϑ the Euler angle. It is possible to recover a rotation matrix (or Direction Cosines Matrix (DCM)) $\mathbf{R}_{bi}$ (that rotates a

tridimensional vector from frame i to frame b) with one of the following expressions, consistent with Kayley-Rodrigues formula

$$\begin{cases} \begin{bmatrix} \mathbf{R}_{bi}^{-} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} & = \left[\mathbf{q}_{-\otimes}^{+}\right]^{T} \left[\mathbf{q}_{-\otimes}^{-}\right] \\ \begin{bmatrix} \mathbf{R}_{bi}^{+} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} & = \left[\mathbf{q}_{+\otimes}^{-}\right]^{T} \left[\mathbf{q}_{+\otimes}^{+}\right] \end{cases} \tag{A.12}$$

From (A.12) we note that the direction cosines matrix is "quadratic" in terms of quaternion components. This implies that $\mathbf{R}_{bi}\left(\mathbf{q}\right) = \mathbf{R}_{bi}\left(-\mathbf{q}\right)$ meaning that $\pm\mathbf{q}$ represent the same attitude. This is the source of the unwinding phenomena that may occur with quaternion based controllers. Should be clear now that the two possibilities of (A.12) are one the transpose of the other and follows also that $\mathbf{q}_{-}$ is the quaternion conjugate of ${\mathbf{q}_{+}}^{2}$.

From Cartan-Dieudonne theorem we do know that the angular velocity $\boldsymbol{\omega}_i$ of the frame i can be recovered from the following relation

$$\frac{d}{dt}\left(\mathbf{R}_{ib}\right) \cdot {\mathbf{R}_{ib}}^{T} = \left[\boldsymbol{\omega}_{i\times}\right] \tag{A.13}$$

By differentiation Eq. (A.10) it is clear that the derivative of a quaternion with respect to time is bounded by the following relation and thus is the quaternion representation of a tridimensional entity, the angular velocity.

$$\boldsymbol{\eta}^{T}\dot{\boldsymbol{\eta}} + \epsilon_{\dot{\epsilon}} = 0 \tag{A.14}$$

This can be made clearer by using the two matrix forms of the quaternion $\mathbf{q}$ and its derivative $\dot{\mathbf{q}}$. Depending on which matrix form is used the result will be different, hence two different $\dot{\mathbf{q}}$ are possible given the same general tridimensional vector $\boldsymbol{\xi}$.

$$\begin{Bmatrix} \boldsymbol{\xi} \\ 0 \end{Bmatrix} = \left[\mathbf{q}_{\otimes}^{-}\right]^{T} \dot{\mathbf{q}_{-}} = \left[\mathbf{q}_{\otimes}^{+}\right]^{T} \dot{\mathbf{q}_{+}} \tag{A.15}$$

The result is still a quaternion but its norm depends on the norm of $\dot{\mathbf{q}}$, hence will be connected to the norm of the relative angular velocity of the frames. Applying Cartan-Dieudonne theorem to (A.12) gives respectively[3]

[2] A quaternion conjugate has vector part with opposite sign: conj $\left(\begin{Bmatrix} \boldsymbol{\eta} \\ \epsilon \end{Bmatrix}\right) = \begin{Bmatrix} -\boldsymbol{\eta} \\ \epsilon \end{Bmatrix}$

[3] It can be proven using properties of the matrix forms.

$$\begin{cases} \frac{d}{dt}\left(\mathbf{R}_{ib}^{-}\right)\cdot\mathbf{R}_{ib}{}^{-T} & = \left[2\boldsymbol{\xi}_{\times}\right] \\ \frac{d}{dt}\left(\mathbf{R}_{ib}^{+}\right)\cdot\mathbf{R}_{ib}{}^{+T} & = -\left[2\boldsymbol{\xi}_{\times}\right] \end{cases}$$

hence we have four different possibilities linked to the choice of mapping from the quaternion $\mathbf{q}$ to the DCM $\mathbf{R}_{ib}$.

$$\begin{Bmatrix} \boldsymbol{\omega}_i \\ 0 \end{Bmatrix} = \pm 2 \begin{Bmatrix} \boldsymbol{\xi} \\ 0 \end{Bmatrix} = \begin{cases} \pm 2 \left[\mathbf{q}_{\otimes}^{-}\right]^T \dot{\mathbf{q}}_{-} \\ \pm 2 \left[\mathbf{q}_{\otimes}^{+}\right]^T \dot{\mathbf{q}}_{+} \end{cases}$$

then the velocity of frame b with respect to frame i is given by $\boldsymbol{\omega}_b = \mathbf{R}_{bi}\boldsymbol{\omega}_i$ that gives four different possibilities

$$\begin{Bmatrix} \boldsymbol{\omega}_b \\ 0 \end{Bmatrix} = \begin{cases} 2 \left[\mathbf{q}_{-\otimes}^{+}\right]^T \dot{\mathbf{q}}_{-} \\ 2 \left[\mathbf{q}_{-\otimes}^{+}\right]^T \left[\mathbf{q}_{-\otimes}^{-}\right] \left[\mathbf{q}_{-\otimes}^{+}\right]^T \dot{\mathbf{q}}_{+} \\ -2 \left[\mathbf{q}_{+\otimes}^{-}\right]^T \left[\mathbf{q}_{+\otimes}^{+}\right] \left[\mathbf{q}_{+\otimes}^{-}\right]^T \dot{\mathbf{q}}_{-} \\ -2 \left[\mathbf{q}_{+\otimes}^{-}\right]^T \dot{\mathbf{q}}_{+} \end{cases} \tag{A.16}$$

To recap what has been said, there are four different possibilities to link a quaternion and its derivative to a DCM and angular velocities. Any choice is valid and depending on the problem in exam one might be computationally better than the other. From here on the attitude kinematics will be represented using the following

$$\begin{cases} \begin{bmatrix} \mathbf{R}_{bi} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \left[\mathbf{q}_{\otimes}^{+}\right]^T \left[\mathbf{q}_{\otimes}^{-}\right] \\ \begin{Bmatrix} \boldsymbol{\omega}_i \\ 0 \end{Bmatrix} = 2 \left[\mathbf{q}_{\otimes}^{-}\right]^T \dot{\mathbf{q}} \end{cases} \tag{A.17}$$

Looking at the derivative of the quaternion components we can link the derivative in terms of $\boldsymbol{\omega}_i$ or $\boldsymbol{\omega}_b$

$$\begin{cases} 2\dot{\boldsymbol{\eta}} & = \epsilon\boldsymbol{\omega}_i - \left[\boldsymbol{\eta}_{\times}\right]\boldsymbol{\omega}_i = \epsilon\boldsymbol{\omega}_b + \left[\boldsymbol{\eta}_{\times}\right]\boldsymbol{\omega}_b \\ 2\dot{\epsilon} & = -\boldsymbol{\eta}^T\boldsymbol{\omega}_i = -\boldsymbol{\eta}^T\boldsymbol{\omega}_b \end{cases} \tag{A.18}$$

Dual quaternion

Dual quaternions are mathematical entities that can be used to fully represent rigid body kinematics and are an extension of the quaternion group

to include in one entity both attitude and position parametrization. A dual quaternion is a quaternion with a dual part: the real part represent rotation, the dual part include also the position of the body. In order to understand the idea behind such formulation, one should first address dual numbers and the properties that are linked to the representation of kinematics through same entities. For the sake of brevity here is omitted, the interested reader should refer to . Should be noted that the said approach is of course closely linked to the screw theory popular in robotics. The bottom line of the approach is that it is possible to use a non-unitary quaternion to represent displacement between reference frames. Since the translation group has not the specific problematic of the rotation group there should be no need to increase the dimensionality of the entities at hand, however using the same arithmetic, linear algebra and others for the whole body kinematics is appealing and general.

The goal, now, is to link any displacement $\boldsymbol{d}_i$ to a position quaternion $\boldsymbol{t}$, taking into account the attitude represented by the rotation quaternion $\mathbf{q}$. A relation between $\mathbf{q}$ and $\boldsymbol{t}$ must hold since only three components of $\boldsymbol{t}$ are linearly independent. The possibilities are

$$\begin{Bmatrix} \boldsymbol{d}_i \\ 0 \end{Bmatrix} = 2 \begin{cases} \left[\mathbf{q}_{-\otimes}^{-}\right]^T \boldsymbol{t}_a \\ \left[\mathbf{q}_{+\otimes}^{-}\right]^T \boldsymbol{t}_b \\ \left[\mathbf{q}_{-\otimes}^{+}\right]^T \boldsymbol{t}_c \\ \left[\mathbf{q}_{+\otimes}^{+}\right]^T \boldsymbol{t}_d \end{cases} \tag{A.19}$$

and the displacement in the other reference frame $\boldsymbol{d}_b = \mathbf{R}_{bi}\boldsymbol{d}_i$

$$\begin{Bmatrix} \boldsymbol{d}_b \\ 0 \end{Bmatrix} = 2 \begin{cases} \left[\mathbf{q}_{-\otimes}^{+}\right]^T \boldsymbol{t}_a \\ \left[\mathbf{q}_{+\otimes}^{-}\right]^T \left[\mathbf{q}_{+\otimes}^{+}\right] \left[\mathbf{q}_{+\otimes}^{-}\right]^T \boldsymbol{t}_b \\ \left[\mathbf{q}_{-\otimes}^{+}\right]^T \left[\mathbf{q}_{-\otimes}^{-}\right] \left[\mathbf{q}_{-\otimes}^{+}\right]^T \boldsymbol{t}_c \\ \left[\mathbf{q}_{+\otimes}^{-}\right]^T \boldsymbol{t}_d \end{cases} \tag{A.20}$$

As before, the choice depends on the uses and its arbitrary, to maintain symmetry with the previous choice here will be used the following

$$\begin{Bmatrix} \boldsymbol{d}_i \\ 0 \end{Bmatrix} = 2 \left[\mathbf{q}_{\otimes}^{-}\right]^T \boldsymbol{t} \tag{A.21}$$

Should be noted that like for rotations a simultaneous change in sign of both $\mathbf{q}$ and $\boldsymbol{t}$ gives back the same $\boldsymbol{d}_i$. If we do use $\boldsymbol{t}$ in a control loop

we could incur in unwinding phenomena, however unlike for rotations a tridimensional exact parametrization can still be made and be exploited.

Differentiating Eq. (A.21) gives

$$\begin{Bmatrix} \mathbf{v}_i \\ 0 \end{Bmatrix} = 2\left[\dot{\mathbf{q}}_{\otimes}^{-}\right]^T \boldsymbol{t} + 2\left[\mathbf{q}_{\otimes}^{-}\right]^T \dot{\boldsymbol{t}} \tag{A.22}$$

The result shows that the derivative of the position quaternion $\dot{\boldsymbol{t}}$ depends on both linear velocity, attitude and angular velocity, as one would expect. By introducing a new matrix form which is the dual of $\left[\overset{-}{\otimes}\right]^T$ we can link the velocities in frame i as follows

$$\begin{Bmatrix} \boldsymbol{\omega}_i \\ 0 \\ \mathbf{v}_i \\ 0 \end{Bmatrix} = 2\begin{bmatrix} \left[\mathbf{q}_{\otimes}^{-}\right]^T & \mathbf{0}_{4\times 4} \\ \left[\boldsymbol{t}_{\otimes}^{-T}\right] & \left[\mathbf{q}_{\otimes}^{-}\right]^T \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}} \\ \dot{\boldsymbol{t}} \end{Bmatrix} \tag{A.23}$$

Now, only the transformation from a reference frame to another is left. The dual quaternion can be represented as a eight-dimensional vector and the product operation can be used again to represent rotation and translation from a frame to another. Rotation and translation are not commutative and can be defined in two different ways. Given a common frame i let there be two reference frame, 1 and 2, whose inertial position and attitude from i are given by dual quaternion $\boldsymbol{a}_1 = \begin{Bmatrix} \mathbf{q}_1^T & \boldsymbol{t}_1^T \end{Bmatrix}^T$ and $\boldsymbol{a}_2 = \begin{Bmatrix} \mathbf{q}_2^T & \boldsymbol{t}_2^T \end{Bmatrix}^T$ here represented as vectors of eight components. The relative displacement from one to the other is given by

$$\boldsymbol{d}_{2i} = \boldsymbol{d}_{ri} + \boldsymbol{d}_{1i} \tag{A.24}$$

and must holds true whichever dual quaternion operation we can define. We can also express $\boldsymbol{d}_{ri}$ in frame 1 using $\boldsymbol{d}_{ri} = \mathbf{R}_{i1}\boldsymbol{d}_{r1}$. To codify the operation in dual quaternion terms we have two possibilities

$$\begin{cases} \boldsymbol{d}_{2i} = \mathbf{R}_{i1}\boldsymbol{d}_{r1} + \boldsymbol{d}_{1i} \\ \boldsymbol{d}_{2i} = \mathbf{R}_{i1}\left(\boldsymbol{d}_{r1} + \boldsymbol{d}_1\right) \end{cases} \tag{A.25}$$

with of course

$$\boldsymbol{d}_{1i} = \mathbf{R}_{i1}\boldsymbol{d}_1 \tag{A.26}$$

thus the choice of which of the two is connected to the position quaternion influence the form of the transformation matrix between frame 1 and

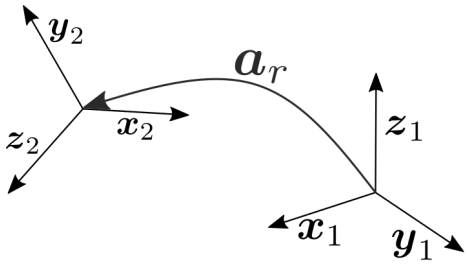


Figure A.1: *Example of reference frame transformation*

2. In other terms if the translation is performed before or after the rotation during the transformation. Looking at (A.19) and (A.20) the first approach seems more natural, although this is still arbitrary. Following this choice, the position quaternion of 2 is given by

$$\boldsymbol{t}_2 = \left[\boldsymbol{t}_{r\otimes}^{-}\right] \mathbf{q}_1 + \left[\mathbf{q}_{r\otimes}^{-}\right] \boldsymbol{t}_1 \tag{A.27}$$

The result can be obtained easily by exploiting some properties of the matrix representation of quaternions and remembering that for rotation holds

$$\mathbf{q}_2 = \left[\mathbf{q}_{r\otimes}^{-}\right] \mathbf{q}_1 \tag{A.28}$$

The complete transformation from frame 1 to frame 2 can be represented by the dual quaternion $\boldsymbol{a}_r$. The transformation is thus a multiplication defined in the dual quaternion group and its matrix representation encapsulates the quaternion matrix forms previously defined.

$$\boldsymbol{a}_2 = \boldsymbol{a}_r \otimes \boldsymbol{a}_1 \tag{A.29}$$

$$\boldsymbol{a}_2 = \left[\boldsymbol{a}_{r\epsilon\times}^{-}\right] \boldsymbol{a}_1 \tag{A.30}$$

$$\begin{Bmatrix} \mathbf{q}_2 \\ \boldsymbol{t}_2 \end{Bmatrix} = \begin{bmatrix} \left[\mathbf{q}_{r\otimes}^{-}\right] & \mathbf{0}_{4\times4} \\ \left[\boldsymbol{t}_{r\otimes}^{-}\right] & \left[\mathbf{q}_{r\otimes}^{-}\right] \end{bmatrix} \begin{Bmatrix} \mathbf{q}_1 \\ \boldsymbol{t}_1 \end{Bmatrix} \tag{A.31}$$

The extension of the quaternion space to the dual quaternion space inherits many matrix form properties, with the exception of the division operation. For unitary quaternion the inverse is equal to the transpose, while for non unitary quaternions like $\boldsymbol{t}$ this does not apply. On the other hand, the inversion operation still retains the linear structure presented before. For example looking at Figure A.1 we can define easily

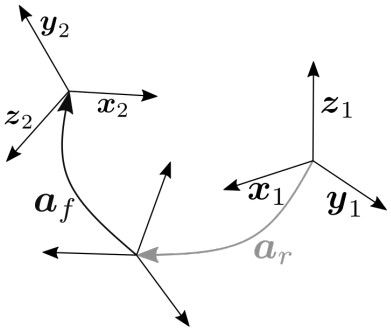


Figure A.2: *Chain of transformations example*

$$\begin{cases} \boldsymbol{a}_2 = \left[\boldsymbol{a}_{r\epsilon\times}^{-}\right] \boldsymbol{a}_1 = \left[\boldsymbol{a}_{1\epsilon\times}^{+}\right] \boldsymbol{a}_r \\ \boldsymbol{a}_1 = \left[\boldsymbol{a}_{r\epsilon\div}^{-}\right] \boldsymbol{a}_2 = \left[\boldsymbol{a}_{2\epsilon\div}^{-T}\right] \boldsymbol{a}_r \\ \boldsymbol{a}_r = \left[\boldsymbol{a}_{1\epsilon\div}^{+}\right] \boldsymbol{a}_2 = \left[\boldsymbol{a}_{2\epsilon\div}^{+T}\right] \boldsymbol{a}_1 \end{cases} \tag{A.32}$$

The linearity and the duality of matrix forms can be easily exploited, for example a chain of transformation can be rearranged in many different ways. Using Figure A.2 as visual reference we have

$$\begin{cases} \boldsymbol{a}_2 = \left[\boldsymbol{a}_{f\epsilon\times}^{-}\right] \left[\boldsymbol{a}_{r\epsilon\times}^{-}\right] \boldsymbol{a}_1 \\ \boldsymbol{a}_2 = \left[\boldsymbol{a}_{f\epsilon\times}^{-}\right] \left[\boldsymbol{a}_{1\epsilon\times}^{+}\right] \boldsymbol{a}_r \\ \boldsymbol{a}_2 = \left[\boldsymbol{a}_{1\epsilon\times}^{+}\right] \left[\boldsymbol{a}_{f\epsilon\times}^{-}\right] \boldsymbol{a}_r \\ \boldsymbol{a}_2 = \left[\boldsymbol{a}_{1\epsilon\times}^{+}\right] \left[\boldsymbol{a}_{r\epsilon\times}^{+}\right] \boldsymbol{a}_f \end{cases} \tag{A.33}$$

This is pretty useful when dealing with many different reference frames and chains of transformations. Similar result can be obtained also using exponential mapping and a consistent minimal representation of the dual quaternion consisting in six components. It can be easily verified that

$$\begin{cases} \left[\boldsymbol{a}_{r\epsilon\times}^{-}\right] = \exp\left(\left[\boldsymbol{\xi}_{r\epsilon\times}^{-}\right]\right) \\ \boldsymbol{\xi}_r = \left\{\boldsymbol{\xi}_q^T \quad 0 \quad \boldsymbol{\xi}_t^T \quad 0\right\}^T \end{cases} \tag{A.34}$$

However the relation with displacement and rotation of $\boldsymbol{\xi}_q$ and $\boldsymbol{\xi}_t$ is highly non-linear, thus more cumbersome to be dealt with.

APPENDIX $\mathcal{B}$

Adaptive control

Adaptive control is a system control paradigm where some parameters are updated considering the evolution in time of the controlled system. These controllers are inherently non-linear and are meant to adapt to uncertainties or to track a reference behaviour. Two main strategies are used for adaptive control: Model Reference Adaptive Control (MRAC) and Adaptive Dynamical Inversion (ADI) which can be seen, depending on the classification, as an indirect MRAC. Both can account for unknown and bounded parameters of a dynamical system with known basis functions in the system state variables and are set to track a reference response of the system. A combination of the two can be devised and is usually referred to as CM-RAC. Literature on the subject is quite vast and the interested reader can look into [97] or into [98] that has been widely used for this Appendix and subsequent work.

Let us consider a linear system as our system plant

$$\dot{x} = ax + b\boldsymbol{u} \tag{B.1}$$

and a stable reference model that we want the controller to follow

$$\dot{x_m} = a_m x_m + b_m r \tag{B.2}$$

with $a_m < 0$ the reference model has exponential convergence towards $-\frac{b_m}{a_m} r$.

Let us take the simpler, yet more desirable, model to be

$$\dot{x_m} = \lambda (r - x_m) \tag{B.3}$$

with positive λ. The system that results is a first order lowpass filter with cutoff frequency λ. It is equivalent to take $a_m = -b_m = -\lambda$.

Direct MRAC estimates and adapt the control gains while the ADI or indirect MRAC estimates the plant uncertain parameters.

Model Reference Adaptive Control

The MRAC law can be written for system of Eq. (B.3) as

$$\boldsymbol{u} = \hat{k}_x x + \hat{k}_r r \tag{B.4}$$

substituting this control law in the system real model (B.1) gives

$$\dot{x} = ax + b\hat{k}_r r + b\hat{k}_x x \tag{B.5}$$

$$\dot{x} = \left(a + b\hat{k}_x\right) x + b\hat{k}_r r \tag{B.6}$$

hence we can conclude that there exist an ideal control gain k such that

$$\begin{cases} -\lambda &= a + bk_x \\ \lambda &= bk_r \end{cases} \tag{B.7}$$

Let us define the tracking error and its derivative as function of $\Delta k_x = \hat{k}_x - k_x$ and $\Delta k_r = \hat{k}_r - k_r$

$$\begin{cases} e &= x - x_m \\ \dot{e} &= a_m e + b\Delta k_x x + b\Delta k_r r \end{cases} \tag{B.8}$$

Consider the Lyapunov function candidate

$$\mathfrak{L} = \frac{1}{2} e^2 + \frac{1}{2} |b| \left(\gamma_x^{-1} \Delta k_x^2 + \gamma_r^{-1} \Delta k_r^2\right) \tag{B.9}$$

Then its derivative is

$$\frac{d}{dt}\mathfrak{L} = a_m e^2 + |b|\,\Delta k_x \left(e\,\text{sign}\,(b)\, x + \gamma_x^{-1} \frac{d}{dt}\Delta k_x \right) + |b|\,\Delta k_r \left(e\,\text{sign}\,(b)\, r + \gamma_r^{-1} \frac{d}{dt}\Delta k_r \right) \tag{B.10}$$

Considering that the system of Eq. (B.1) is LTI, then $\frac{d}{dt}\Delta k_x = \frac{d}{dt}\hat{k}_x$ and $\frac{d}{dt}\Delta k_r = \frac{d}{dt}\hat{k}_r$, hence to achieve the stability condition we have to null the terms in the parentheses, giving the adaptation laws as

$$\begin{cases} \frac{d}{dt}\hat{k}_x &= -\gamma_x e x \text{sign}\,(b) \\ \frac{d}{dt}\hat{k}_r &= -\gamma_r e r \text{sign}\,(b) \end{cases} \tag{B.11}$$

with $a_m < 0$ this gives $\frac{d}{dt}\mathfrak{L} < 0$ in all the considered domain, hence parameter tracking is stable and through a chain of considerations we have that error and its derivatives are bounded. Differentiating Eq. (B.10) together with the previous considerations leads to the boundedness of the second derivative of the Lyapunov function and consequent continuity of the first derivative. Then, thanks to Barbalat's lemma we can conclude asymptotic tracking.

Adaptive Dynamic Inversion control

In the ADI version the adaptive system estimates directly the parameters of the plant. The system plant of (B.1) is rewritten as follows

$$\dot{x} = ax + b\boldsymbol{u} + \hat{a}x - \hat{a}x + \hat{b}\boldsymbol{u} - \hat{b}\boldsymbol{u} \tag{B.12}$$

$$\dot{x} = \hat{a}x + \hat{b}\boldsymbol{u} + (a - \hat{a})\, x + \left(b - \hat{b}\right)\boldsymbol{u} \tag{B.13}$$

$$\dot{x} = \hat{a}x + \hat{b}\boldsymbol{u} - \Delta a x - \Delta b \boldsymbol{u} \tag{B.14}$$

the ADI feedback control is

$$\boldsymbol{u} = \frac{1}{\hat{b}} \left((a_m - \hat{a})\, x + b_m r \right) \tag{B.15}$$

substituting (B.15) in (B.14) gives the closed loop dynamics

$$\dot{x} = \lambda\,(r - x) + \Delta a x + \Delta b \boldsymbol{u} \tag{B.16}$$

The tracking error is now defined as

$$\begin{cases} e &= x - x_m \\ \dot{e} &= -\lambda e - \Delta a x - \Delta b \boldsymbol{u} \end{cases} \tag{B.17}$$

The Lyapunov function candidate is

$$\mathfrak{L} = \frac{1}{2}e^2 + \frac{1}{2}\left(\gamma_a^{-1}\Delta a^2 + \gamma_b^{-1}\Delta b^2\right) \tag{B.18}$$

its derivative is

$$\frac{d}{dt}\mathfrak{L} = -\lambda e^2 + \Delta a\left(-ex + \gamma_a^{-1}\frac{d}{dt}\Delta a\right) + \Delta b\left(-e\boldsymbol{u} + \gamma_b^{-1}\frac{d}{dt}\Delta b\right) \tag{B.19}$$

Considering that the system of (B.1) is LTI, then $\frac{d}{dt}\Delta a = \frac{d}{dt}\hat{a}$ and $\frac{d}{dt}\Delta b = \frac{d}{dt}\hat{b}$, hence

$$\begin{cases} \frac{d}{dt}\hat{a} &= \gamma_a e x \\ \frac{d}{dt}\hat{b} &= \gamma_b e \boldsymbol{u} \end{cases} \tag{B.20}$$

should be noted that in this case the reference signal does not enter in the adaptive laws directly as was in the MRAC. Stability and tracking properties demonstrations follows as in the previous case.

Comparison: ADI vs MRAC

Let us compare the two control laws and point out the differences.

$$\begin{cases} \boldsymbol{u}_{MRAC} &= \hat{k}_x x + \hat{k}_r r \\ \boldsymbol{u}_{ADI} &= \frac{1}{\hat{b}}\left(\lambda\left(r - x\right) - \hat{a}x\right) \end{cases} \tag{B.21}$$

The MRAC law is simpler but is slightly more complex to estimate the initial values for the parameters and reasonable boundaries, while on the other hand the ADI control law requires the inversion of $\hat{b}$ which poses a zero crossing issue as well as limited uses for a MIMO system since not always is possible to invert the control to state matrix due to different dimensions, although a Pseudo Inverse might be used. The adaptive parameters in the ADI framework are closely linked to parametric uncertainty in the systems and thus it is more likely to find a suitable initial guess and boundaries for the parameters. Nevertheless the relation is quickly found as follows

Table B.1: *Comparison simulation data*

	a/k_x	b/k_r
System	$\pm$0.10	0.10
perfect controller	$\pm$0.10	0.10
MRAC	-7.34	6.67
ADI	1.00	1.50
λ	1.00	
r	1.00	
ω	1.00	

$$\begin{cases} \hat{k}_x & = -\frac{(\lambda+\hat{a})}{\hat{b}} \\ \hat{k}_r & = \frac{\lambda}{\hat{b}} \end{cases} \tag{B.22}$$

the difference lies in the update of the parameters. In the MRAC the updates for reference and state are not connected while in the ADI through control variable $\boldsymbol{u}$ the update of $\hat{b}$ relies on a combination of state, reference and estimation of parameters.

In the classical implementation both versions makes use of a reference model, however it is possible to reach the same conclusion using as error $e = x - r$ for the ADI controller, inherently assuming a unitary transfer function between the reference and the model; stability and convergence properties still holds but system response is faster.

Making use of the structure of the system, the ADI allows for simplifications in cases where the parameter $a \simeq 0$ by simply eliminating the update on $\hat{a}$. This operation is more cumbersome in the MRAC design.

Example of comparison

Let us address other differences through a simple example. Let us take the system of Eq. (B.1) with two kind of input: a filtered step function of amplitude r and a sinusoidal signal of frequency ω and maximum amplitude r. Values for system and controllers can be found in Table (B.1). The gains have been selected to have a similar behaviour for the step response.

The two adaptive controller are devised with the same system parameter and the same reference controller, both modified to accept a feed forward information. Should be noted that such modification needs to be included in the reference model as well and is straightforward only in the ADI design, while for the MRAC a weighting parameter is added, namely a factor $\hat{k}_r/\lambda$.

This implies that for time varying reference the ADI is supposed to be

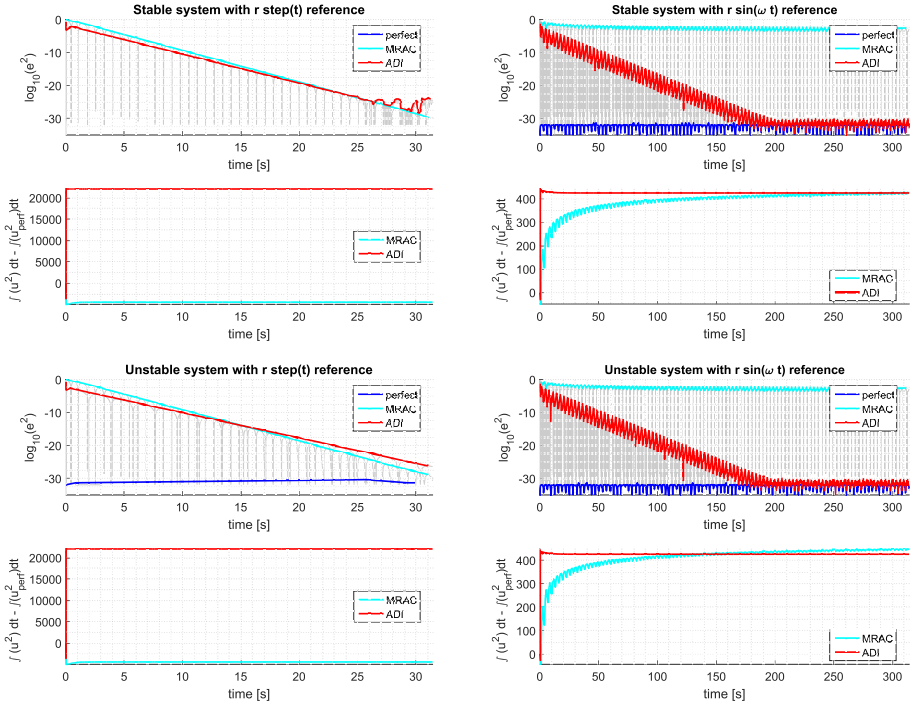


Figure B.1: *MRAC vs ADI*

more performing than the MRAC. Simulation for stable and unstable plant with step and sinusoidal entry are performed and results can be found in Figure B.1.

In Figure B.1 for each simulations the quadratic errors and the control effort are presented. The control effort is shown with respect to the baseline of a perfect controller. The MRAC offers better performance than ADI for a step input while for a constantly exciting reference the ADI parameters converge faster and outperform greatly the MRAC controller.

The MRAC performance could be improved by adding another gain to be estimated for the feed forward, however that would increase the complexity and the number of computations needed for such approach. On the other hand it can be seen that for a step entry the ADI controller requires more effort to bring the controller to operative regime. The MRAC has a gentler action and sometimes lower than the optimal but with higher error.

The bottom line of this confront is to show that for different applicative cases the choice of the architecture is important. In cases where a time varying reference is expected it is more straightforward to choose the ADI controller, while for reference with little frequency content the MRAC is

a more reasonable choice. Also, system with negligible dynamics can be treated better with a simplified ADI with respect to a full MRAC.

Confront with baseline controllers

Let us take the Lyapunov candidate function for the ADI controller with plant (B.1) and suppose to freeze the parameter update. The derivative of the Lyapunov function can be rewritten as follows

$$\frac{d}{dt}\mathfrak{L} = e\left(-\lambda e - \Delta a x - \Delta b \boldsymbol{u}\right) = e\left(\left(D-\lambda\right)e - Dx_m + Cr\right)$$

even assuming bounded reference and reference model state, since at low frequency it perfectly follows the reference, we might have a condition where $D > \lambda$ and thus convergence might not be guaranteed. It can be shown that such instability condition can be written as follows

$$\frac{\hat{b}}{b}a - \hat{a} > \lambda \tag{B.23}$$

That coherently nullifies for $\hat{a} \to a$ and $\hat{b} \to b$. A properly formulated adaptive control can reduce the value of D to ensure stability if the system knowledge is affected by high uncertainties or lower control capability. As a matter of fact λ is connected to the maximum control action and limited when dealing with discrete implementation.

Assuming $D < \lambda$ we have that the state to reference error boundary for a non adaptive controller, considering that $x_m \to r$ and thus $e \to x - r$ is given by

$$\frac{\|e_{max}\|}{\|r_{max}\|} = \frac{\|C - D\|}{\lambda - D} \tag{B.24}$$

and again is function of the uncertainties. From this also follows that the higher λ the lower $\|e_{max}\|$ as one would expect. A proper designed adaptive controller reduces inherently the uncertainties and thus is able to reduce the boundary of the state error with respect to the non-adaptive law. For simple system with null a the boundary becomes

$$\frac{\|e_{max}\|}{\|r_{max}\|} = 2\left\|\frac{\hat{b}}{b} - 1\right\| \tag{B.25}$$

and considering that we assume $\frac{\hat{b}}{b} > 0$ we can conclude that underestimating $\hat{b}$ is slightly better than overestimating it too much. The same

conclusion can be achieved by simply looking at the control law where the error is multiplied by a factor $\frac{\lambda}{\hat{b}}$, hence the lower $\hat{b}$ the higher the control action and equivalent stiffness of the system resulting in lower errors. An adaptive control law would try to reduce the estimation error in such a way that an optimal control effort is achieved. In case of $\hat{a} = a = 0$ we can relate approximately

$$\boldsymbol{u}_{ADI} = \frac{b}{\hat{b}} \boldsymbol{u}_{perfect} \tag{B.26}$$

From which follows that overestimating $\hat{b}$ reduces the control action with respect to the theoretical optimum (at fixed reference model) with reduced performance while, on the other hand, underestimating $\hat{b}$ leads to needlessly higher control effort.

Miscellaneous issues of Adaptive Control

Both ADI and MRAC parameter update equations rely on the state determination. Since measurements are always affected by noise it is straightforward to see that noise can affect negatively the variation of the parameters, hence a common solution is to apply a Dead-Zone to the error in parameters estimates. In this case the tracking error e will never be zero but will converge to a confined zone. The limit of the dead-zone should take into account the noise content of the measured state, typically in the higher frequencies as a static non-modelled bias would be unstabilizing.

Another issue with Adaptive control is that parameters may shift too much and require a saturation. Since the parameters are estimated through an integration process what is often required is an Anti-Windup operator. In literature the projection operator is often used, but virtually any Anti-Windup would suffice.

Other parameters estimation

In the adaptive control framework it is also possible to include estimation of state dependent uncertain functions. Let us re-write the plant of Eq. (B.1) with the addition of such contribution

$$\dot{x} = ax + b\boldsymbol{u} + \theta^T \Phi(x) \tag{B.27}$$

where θ is the vector containing the parameters and $\Phi(x)$ is the vector of basis functions of the state. If $\Phi(x)$ are known and the parameters θ are

constant, then the adaptive framework allows to include such uncertainties in the controller. The MRAC and ADI control laws are then modified as follows

$$\begin{cases} \boldsymbol{u}_{MRAC} & = \hat{k}_x x + \hat{k}_r r + \hat{\theta}^T \Phi(x) \\ \boldsymbol{u}_{ADI} & = \frac{1}{\hat{b}} \left(\lambda (r - x) - \hat{a} x - \hat{\theta}^T \Phi(x) \right) \end{cases} \tag{B.28}$$

with update laws that can be determined through the same procedure presented before. This leads to the following, quite similar, adaptive parameters update.

$$\begin{cases} \frac{d}{dt}\hat{\theta} = -\Gamma_\theta e \Phi(x) \operatorname{sign}(b) & \text{MRAC} \\ \frac{d}{dt}\hat{\theta} = \Gamma_\theta e \Phi(x) & \text{ADI} \end{cases} \tag{B.29}$$

Convergence of parameters

Both ADI and MRAC solutions, as other adaptive paradigms, do not guarantee convergence of parameters unless a persistent excitation is exerted on the system. Even in cases where perfect measurements are available, parameters converge to their true values only with specific classes of reference signals in a time frame deeply varying. Stability does not imply convergence to the real values and a bounded convergence is expected.

The result is expected as estimating parameters of a dynamical system requires a reference signal capable of exciting a response dependent on the said parameters. This applies also to adaptive controls where a state observer is used to estimate the uncertain parameters.

Time varying model

If the system is no longer LTI but changes in time, then the used schemes for MRAC and ADI are no longer asymptotically converging to null error but to a bounded error. This can be easily seen looking at the Lyapunov function candidate derivatives. Take the ADI scheme for example as parametric variations are more easily seen.

$$\frac{d}{dt}\mathfrak{L} = -\lambda e^2 + \Delta a \gamma_a^{-1} \frac{d}{dt} a + \Delta b \gamma_b^{-1} \frac{d}{dt} b \tag{B.30}$$

If we assume Δa and Δb to be bounded and both $\frac{d}{dt}a$ and $\frac{d}{dt}b$ to be bounded as well we can state that

$$\frac{d}{dt}\mathfrak{L} \leq -\lambda e^2 + C \tag{B.31}$$

where $C > 0$ is equal to the maximum contribution of the time varying terms and errors and is meant to define the error boundary. The maximum error expected is thus

$$\|e_{max}\| = \sqrt{\frac{C}{\lambda}} \tag{B.32}$$

which is rather intuitive as λ defines the static disturbance rejection of the linear system.

APPENDIX C

Levenberg-Marquardt algorithm

Given a non-linear function $\mathbf{r} = f(\mathbf{a})$ in the parameters vector $\mathbf{a}$, a general minimization algorithm search for the solution $\mathbf{a}^*$ such that

$$\mathbf{a}^* = \arg\min \left(\mathbf{r}^T \mathbf{r}\right) \tag{C.1}$$

Hence a good indication of the goodness of a solution at step k is the chi squared criterion

$$\chi^2(\mathbf{a}_k) = \mathbf{r}_k^T \mathbf{r}_k = \sum_{i=1}^{n} r_i^2 \tag{C.2}$$

The function to be minimized can either be scalar or vector and may be used for data fitting, solving system of equations, training neural networks and so on. A popular algorithm is the LM method, which is a non-linear recursive minimization algorithm that can shift between Newton-Gauss (NG) method and the gradient method (also Steepest Descent (SD)). Let us address these two methods first. All these methods are able to find *local minima* not *global minima* though recursive iterations:

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \delta \mathbf{a}_{k+1} \tag{C.3}$$

and will achieve convergence at a certain iteration under some user defined threshold in most cases.

Gradient descent method

The gradient method is inherently straightforward: the solution is found looking at the direction that minimizes χ^2. At each iteration the solution is updated as follows

$$\delta \mathbf{a}_{k+1} = -\alpha \hat{\mathbf{g}} \tag{C.4}$$

where $\hat{\mathbf{g}}$ is the gradient direction of the squared criterion and α is the length of the step. The latter might be changed iteration by iteration to adjust to the solution, if needed.

Taking the gradient of the squared criterion is simple

$$\frac{\partial \chi^2 (\mathbf{a})}{\partial \mathbf{a}} = 2\mathbf{J}^T \mathbf{r} \tag{C.5}$$

where $\mathbf{J} = \frac{\partial (\mathbf{r})}{\partial \mathbf{a}}$ is the Jacobian of the residual $\mathbf{r}$. Following this principle, the opposite direction of the gradient is the direction of minimization of the residual, therefore the SD step is given by

$$\delta \mathbf{a}_{k+1} = -\alpha \frac{\mathbf{J}_k^T \mathbf{r}_k}{\|\mathbf{J}_k^T \mathbf{r}_k\|} \tag{C.6}$$

Should be noted that with a constant α there could be situations where the SD method chatters around the solution or diverges.

Newton-Gauss method

The Newton-Gauss method can be seen as the generalization of the Newton method to solve non-linear problems with higher order. First let us expand χ^2 in a neighbourhood of state $\mathbf{a}$ with a small perturbation $\delta \mathbf{a}$

$$\chi^2 (\mathbf{a} + \delta \mathbf{a}) = \mathbf{r} (\mathbf{a} + \delta \mathbf{a})^T \mathbf{r} (\mathbf{a} + \delta \mathbf{a}) \tag{C.7}$$

Expanding the residual and stopping at the first order term leads to

$$\mathbf{r} (\mathbf{a} + \delta \mathbf{a}) \simeq \mathbf{r} + \mathbf{J} \delta \mathbf{a} \tag{C.8}$$

substituting (C.8) in (C.7) lead to

$$\chi^2 (\mathbf{a} + \delta \mathbf{a}) \simeq \mathbf{r}^T \mathbf{r} + 2\delta \mathbf{a}^T \mathbf{J}^T \mathbf{r} + \delta \mathbf{a}^T \mathbf{J}^T \mathbf{J} \delta \mathbf{a} \tag{C.9}$$

The variation of the perturbed solution with respect to the perturbation is

$$\frac{\partial \chi^2 (\mathbf{a} + \delta\mathbf{a})}{\partial \delta\mathbf{a}} \simeq 2\mathbf{J}^T\mathbf{r} + 2\mathbf{J}^T\mathbf{J}\delta\mathbf{a} \tag{C.10}$$

when the variation of the perturbed solution is null, then $\mathbf{a}$ maps to a local minimum, maximum or a saddle. The variation that satisfy the condition at each step is given by

$$\delta\mathbf{a}_{k+1} = -\left(\mathbf{J}_k^T\mathbf{J}_k\right)^{-1}\mathbf{J}_k^T\mathbf{r}_k \tag{C.11}$$

The solution of the NG step is simply given by the pseudoinverse of the Jacobian and the residual. It is easy to see that the results is equivalent to the least square solution of a linear problem, but for the NG method the procedure is repeated until convergence.

Levenberg - Marquardt method

The Levenberg-Marquardt algorithm owes its name to Kenneth Levenberg who initially proposed the algorithm in 1944 [99] and to Donald W. Marquardt who revised the method in 1963 [100]. Since then it has been a popular method to solve non linear least square problems and has been applied to computer vision, Neural Network training and much more. Following the initial premises, the canonical form of the LM algorithm step is as follows

$$\left(\mathbf{J}_k^T\mathbf{J}_k + \lambda_k \text{diag}\left(\mathbf{J}_k^T\mathbf{J}_k\right)\right)\delta\mathbf{a}_{k+1} = -\mathbf{J}_k^T\mathbf{r}_k \tag{C.12}$$

where λ_k is the LM damping parameters that shifts the algorithm from SD to NG. In case where $\mathbf{J}_k^T\mathbf{J}_k$ could be singular it is often used the primal version with the identity matrix instead of the diagonal of $\mathbf{J}_k^T\mathbf{J}_k$. The latter has been introduced in the past because represent the perfect scaling of the problem for numerical reasons. From (C.12) it is straightforward to see that for $\lambda_k = 0$ the LM step is equal to the NG step, while for high values of λ_k the step is a close approximation of the SD step as on the right hand side of (C.12) appears the gradient of χ^2.

In general, at each step the new solution is tested with a metric function [101] that usually is given by

$$\rho(\mathbf{a}_{k+1}) = \frac{\chi^2(\mathbf{a}_k) - \chi^2(\mathbf{a}_k + \delta\mathbf{a}_k)}{2\delta\mathbf{a}_{k+1}^T(\lambda_k\delta\mathbf{a}_{k+1} - \mathbf{J}_k^T\mathbf{r}_k)} \tag{C.13}$$

and if it is higher than a threshold the step is accepted and the damping parameter reduced to approach the NG algorithm. Otherwise λ_k is increased for the next iteration and the algorithm approaches the SD method, useful when far from the solution. The update of λ_k is indeed a thrust region algorithm. In order to waste less iterations a step is often accepted as soon as $\chi^2(\mathbf{a}_k + \delta\mathbf{a}_{k+1}) < \chi^2(\mathbf{a}_k)$.

Damping parameter update

In the years the LM algorithm has been studied and new more performing ways to update λ_k have been found. Let us introduce the method used in this work. The damping factor is always computed as function of the norm of the residual vector as follows

$$\lambda_k = \mu_0 \|\mathbf{r}_k\| \tag{C.14}$$

where the parameter μ_0 can be kept constant or evolve with a thrust region algorithm. It has been demonstrated that $\lambda_k \propto \|\mathbf{r}_k\|^a$ has superlinear convergence for $a < 1$ and quadratic convergence for $1 \leq a \leq 2$ [102]. In [103] a generalization of such approach is presented, with μ_0 updated using a thrust region technique, but μ_0 assumes also the role of a scaling factor. Since the scale of the error might change during the descent it is important that the method keep the initial λ_k in a suitable range for fast convergence.

$$\mu_0 = \frac{\lambda_0}{\|\mathbf{r}_k\|} \tag{C.15}$$

where λ_0 is the initial value of the damping parameter, usually chosen by experience, initial guess error estimation and/or a robust initialization. In the implemented LM $\lambda_0$is chosen between a subset of possible values: the one that produces the highest metric for the first step is taken as initial damping. To reduce the computational cost it is implemented the procedure of [104,105] where the LM step of Eq. (C.12) is re-written as two stage QR decomposition. This procedure is heavier than the previous one but does not require a full re-computation when the damping is changed, making it appealing for the first initialization.

From experience in using the LM algorithm with this formulation it has been observed that in case of well defined problems μ_0 can be kept constant and convergence is attained in few steps. On the other hand, in some cases where the function is inherently numerically hill defined the thrust region update of μ_0 proves to be fundamental.

Two step solution

In the version of [104, 105] the LM algorithm is solved with another perspective. The LM step can be rewritten as follows

$$\begin{bmatrix} \mathbf{J}_k \\ \lambda_k \mathbf{I}_{n\times n} \end{bmatrix} \delta \mathbf{a}_k = - \begin{bmatrix} \mathbf{r}_k \\ \mathbf{0} \end{bmatrix} \tag{C.16}$$

The solution is obtained using a recursive QR decomposition. The first step is to compute the QR decomposition of the Jacobian, such that

$$\mathbf{J}_k = \mathbf{Q}_1 \mathbf{U}_1 \tag{C.17}$$

where $\mathbf{Q}_1$ is an orthogonal matrix with dimension equal to the rows of the Jacobian and $\mathbf{U}_1$ an upper triangular matrix. Since ${\mathbf{Q}_1}^{-1} = {\mathbf{Q}_1}^T$ it is easy to compute

$$\begin{bmatrix} \mathbf{U}_1 \\ \lambda_k \mathbf{I}_{\times} \end{bmatrix} \delta \mathbf{a}_k = - \begin{bmatrix} {\mathbf{Q}_1}^T \mathbf{r}_k \\ \mathbf{0} \end{bmatrix} = - \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix} \tag{C.18}$$

Then the QR decomposition is used again as follows

$$\begin{bmatrix} \mathbf{U}_1 \\ \lambda_k \mathbf{I}_{\times} \end{bmatrix} \delta \mathbf{a}_k = \mathbf{Q}_2 \mathbf{U}_2 \tag{C.19}$$

The system is thus

$$\mathbf{U}_2 \delta \mathbf{a}_k = -{\mathbf{Q}_2}^T \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix} = - \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \tag{C.20}$$

and the solution of the problem is

$$\delta \mathbf{a}_k = -\mathbf{U}_2^{-1} \mathbf{u} \tag{C.21}$$

where the last step can be computed with back substitution since $\mathbf{U}_2$is upper triangular. In order to avoid unnecessary computation and get $\mathbf{u}$ it is possible to take the upper left portion of the matrix $\mathbf{Q}_2$. The advantage of this function is that if the step is not successful and λ_k changes only the second QR step must be computed again. To reduce numerical errors is thus also possible to scale the Jacobian to have column of unitary length by setting

$$\mathbf{J}_1 = \mathbf{J}_k \mathbf{D} \tag{C.22}$$

$$\mathbf{D} = \text{diag}\left(\begin{bmatrix} \|J_1\|^{-1} & \cdots & \|J_i\|^{-1} & \cdots & \|J_m\|^{-1} \end{bmatrix} \right) \tag{C.23}$$

Of course the state variation solution has to be modified accordingly

$$\delta \mathbf{a}_k = -\mathbf{D}\mathbf{U}_2^{-1}\mathbf{u} \tag{C.24}$$

List of Figures

List of Tables

Bibliography

[1] R. M. Pinson, R. T. Howard, and A. F. Heaton, "Orbital express advanced video guidance sensor: ground testing, flight results and comparisons," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 1, pp. 1–9, 2008.

[2] R. T. Howard, A. F. Heaton, R. M. Pinson, and C. K. Carrington, "Orbital express advanced video guidance sensor," in *Aerospace Conference, 2008 IEEE*, pp. 1–10, IEEE, 2008.

[3] A. F. Heaton, R. T. Howard, and R. M. Pinson, "Orbital express AVGS validation and calibration for automated rendezvous," in *AIAA guidance, navigation and control conference and exhibit, Honolulu, United States*, pp. 18–21, 2008.

[4] N. F. Dipprey and S. Rotenberger, "Orbital express propellant resupply servicing," in *Proceedings of the 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, pp. 20–23, 2003.

[5] A. Ogilvie, J. Allport, M. Hannah, and J. Lymer, "Autonomous satellite servicing using the orbital express demonstration manipulator system," in *Proc. of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp. 25–29, 2008.

[6] K. Yoshida, "Engineering test satellite VII flight experiments for space robot dynamics and control: theories on laboratory test beds ten years ago, now in orbit," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 321–335, 2003.

[7] T. Kasai, M. Oda, and T. Suzuki, "Results of the ETS-7 mission-rendezvous docking and space robotics experiments," in *Artificial Intelligence, Robotics and Automation in Space*, vol. 440, p. 299, 1999.

[8] Y. Ohkami and M. Oda, "NASDA's activities in space robotics," in *Artificial Intelligence, Robotics and Automation in Space*, vol. 440, p. 11, 1999.

[9] K. Yoshida, "Space robot dynamics and control: To orbit, from orbit, and future," in *Robotics Research*, pp. 449–456, Springer, 2000.

[10] A. Long, M. Richards, and D. E. Hastings, "On-orbit servicing: a new value proposition for satellite design and operation," *Journal of Spacecraft and Rockets*, vol. 44, no. 4, pp. 964–976, 2007.

[11] A. M. Long *et al.*, *Framework for evaluating customer value and the feasibility of servicing architectures for on-orbit satellite servicing*. PhD thesis, Massachusetts Institute of Technology, 2005.

[12] J. H. Saleh, E. S. Lamassoure, D. E. Hastings, and D. J. Newman, "Flexibility and the value of on-orbit servicing: New customer-centric perspective," *Journal of Spacecraft and Rockets*, vol. 40, no. 2, pp. 279–291, 2003.

[13] B. R. Sullivan and D. L. Akin, "A survey of serviceable spacecraft failures," *Databases*, vol. 1, p. 4540, 2001.

[14] A. Ellery, J. Kreisel, and B. Sommer, "The case for robotic on-orbit servicing of spacecraft: spacecraft reliability is a myth," *Acta Astronautica*, vol. 63, no. 5, pp. 632–648, 2008.

[15] SpaceX, "Making life multiplanetary." `http://www.spacex.com/mars`. accessed 2017-10-24.

[16] S. Moynahan, S. Touhy, *et al.*, "Development of a modular on-orbit serviceable satellite architecture," in *Digital Avionics Systems, 2001. DASC. 20th Conference*, vol. 2, pp. 8D4–1, IEEE, 2001.

[17] ArianSpace, "Galileo satellites experience orbital injection anomaly on soyuz launch: Initial report." `http://www.arianespace.com/mission-update/galileo-satellites-experience-orbital-injection-anomaly-on-soyuz-launch-initial-report/`. accessed 2017-10-24.

[18] A. Rivolta and M. Lavagna, "On orbit servicing GNC through a dual quaternion approach," *68th International Astronautical Congress, Adelaide, Australia*, 2017.

[19] A. Rivolta and M. Lavagna, "On orbit servicing mission: GNC architectures driven by client requirements," *67th International Astronautical Congress, Guadalajara, Mexico*, 2016.

[20] V. A. Chobotov, *Orbital Mechanics*. AIAA, 2002.

[21] K. Yamanaka and F. Ankersen, "New state transition matrix for relative motion on an arbitrary elliptical orbit," *Journal of Guidance,Control, and Dynamics*, vol. 25, no. 1, pp. 60–66, 2002.

[22] Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized jacobian matrix," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 303–314, 1989.

[23] E. Papadopoulos and S. Dubowsky, "On the nature of control algorithms for free-floating space manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 750–758, 1991.

[24] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems," *IEEE Transactions on robotics and automation*, vol. 9, no. 5, pp. 531–543, 1993.

[25] J. C. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Transactions on robotics and automation*, vol. 8, no. 1, pp. 53–64, 1992.

[26] G. Arantes, E. M. Rocco, I. M. da Fonseca, and S. Theil, "Far and proximity maneuvers of a constellation of service satellites and autonomous pose estimation of customer satellite using machine vision," *Acta Astronautica*, vol. 66, no. 9, pp. 1493–1505, 2010.

[27] W. Xu, Q. Xue, H. Liu, X. Du, and B. Liang, "A pose measurement method of a non-cooperative GEO spacecraft based on stereo vision," in *12th International Conference on Control Automation Robotics & Vision*, pp. 966–971, IEEE, 2012.

[28] M. McCrum, M. Dunstan, and S. Parkes, "Realistic image generation for testing vision-based autonomous rendezvous," in *7th International ESA Conference on Guidance, Navigation and Control Systems*, ESA, 2008.

[29] X. Du, B. Liang, W. Xu, X. Wang, and X. Gao, "A semi-physical simulation system for binocular vision guided rendezvous," in *12th International Conference on Control Automation Robotics & Vision*, pp. 853–858, IEEE, 2012.

[30] D. C. Woffinden and D. K. Geller, "Relative angles-only navigation and pose estimation for autonomous orbital rendezvous," *Journal of Guidance Control and Dynamics*, vol. 30, no. 5, pp. 1455–1469, 2007.

[31] B. Wie and P. M. Barba, "Quaternion feedback for spacecraft large angle maneuvers," *Journal of Guidance*, vol. 8, no. 3, pp. 360–365, 1985.

[32] S. Joshi, A. Kelkar, and J.-Y. Wen, "Robust attitude stabilization of spacecraft using nonlinear quaternion feedback," *IEEE Transactions on Automatic control*, vol. 40, no. 10, pp. 1800–1803, 1995.

[33] J.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic control*, vol. 36, no. 10, pp. 1148–1162, 1991.

[34] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Robust global asymptotic attitude stabilization of a rigid body by quaternion-based hybrid feedback," in *Proceedings of the 48th IEEE Conference on Decision and Control, 2009*, pp. 2522–2527, IEEE, 2009.

[35] P. Tsiotras, "Stabilization and optimality results for the attitude control problem," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 772–779, 1996.

[36] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[37] D. C. Woffinden and D. K. Geller, "Observability criteria for angles-only navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, 2009.

[38] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.

[39] D. D. Diel, P. DeBitetto, and S. Teller, "Epipolar constraints for vision-aided inertial navigation," in *Seventh IEEE Workshops on Application of Computer Vision, Volume 1.*, vol. 2, pp. 221–228, IEEE, 2005.

[40] M. George and S. Sukkarieh, "Inertial navigation aided by monocular camera observations of unknown features," in *IEEE International Conference on Robotics and Automation*, pp. 3558–3564, IEEE, 2007.

[41] L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart, "Closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence," in *IEEE International Conference on Robotics and Automation*, pp. 4546–4553, IEEE, 2011.

[42] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation*, pp. 3565–3572, IEEE, 2007.

[43] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4326–4333, IEEE, 2002.

[44] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, J. Montgomery, A. Ansar, and L. Matthies, "Coupled vision and inertial navigation for pin-point landing," in *NASA Science and Technology Conference*, 2007.

[45] D. Zachariah and M. Jansson, "Camera-aided inertial navigation using epipolar points," in *IEEE/ION Position Location and Navigation Symposium*, pp. 303–309, IEEE, 2010.

[46] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1656–1663, IEEE, 1999.

[47] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement," in *IEEE International Conference on Robotics and Automation*, pp. 2935–2942, IEEE, 2013.

[48] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Observability-constrained vision-aided inertial navigation," *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, vol. 1, 2012.

[49] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, 2013.

[50] H. Benninghoff, F. Rems, and T. Boge, "Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing," *Acta Astronautica*, vol. 102, pp. 67–80, 2014.

[51] A. Petit, N. Despré, E. Marchand, K. Kanani, F. Chaumette, S. Provost, and G. Flandin, "3D model-based tracking for space autonomous rendezvous," in *8th International ESA Conference on Guidance and Navigation Control Systems,2011*, 2011.

[52] A. Yol, E. Marchand, F. Chaumette, K. Kanani, and T. Chabot, "Vision-based navigation in low Earth orbit," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2016.

[53] P. J. Besl, N. D. McKay, *et al.*, "A method for registration of 3-D shapes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[54] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[55] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[56] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment, a modern synthesis," in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[57] X. Gao, B. Liang, and W. Xu, "Attitude determination of large non-cooperative spacecrafts in final approach," in *11th International Conference on Control Automation Robotics & Vision*, pp. 1571–1576, IEEE, 2010.

[58] X. Du, B. Liang, W. Xu, and Y. Qiu, "Pose measurement of large non-cooperative satellite based on collaborative cameras," *Acta Astronautica*, vol. 68, no. 11, pp. 2047–2065, 2011.

[59] T. Tzschichholz, T. Boge, and H. Benninghoff, "A flexible image processing framework for vision-based navigation using monocular image sensors," in *8th International ESA Conference on Guidance and Navigation Control Systems,2011*, 2011.

[60] J. Lewis, "Fast template matching," in *Vision interface*, vol. 95, pp. 15–19, 1995.

[61] J. Lewis, "Fast normalized cross-correlation," in *Vision interface*, vol. 10, pp. 120–123, 1995.

[62] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[63] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[64] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pp. 61–62, 1987.

[65] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 13–27, 1984.

[66] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[67] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, pp. 404–417, Springer, 2006.

[68] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[69] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: image understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[70] H. Liu, Z. Wang, B. Wang, and Z. Li, "Pose determination of non-cooperative spacecraft based on multi-feature information fusion," in *IEEE International Conference on Robotics and Biomimetics*, pp. 1538–1543, IEEE, 2013.

[71] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[72] M. Dunstan and K. Hornbostel, "Image processing chip for relative navigation for lunar landing," in *9th International ESA Conference on Guidance, Navigation, and Control Systems*, (Porto, Portugal), June 2014.

[73] G. Capuano, M. Severi, E. Della Sala, R. Ascolese, C. Facchinetti, and F. Longo, "Compact and high-performance equipment for vision-based navigation," in *63rd International Astronautical Congress (IAC)*, (Napoli, Italy), Oct. 2012.

[74] G. Capuano, R. Ascolese, D. Titomanlio, P. Longobardi, M. De Nino, and G. Formicola, "A MULTI-OCULAR SMART SYSTEM FOR VISION-BASED SPACE NAVIGATION," in *65rd International Astronautical Congress (IAC)*, (Toronto, Canada), 2014.

[75] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[76] M. H. Kaplan, "Modern spacecraft dynamics and control," *New York, John Wiley and Sons, Inc., 1976. 427 p.*, 1976.

[77] M. L. Psiaki, "Magnetic torquer attitude control via asymptotic periodic linear quadratic regulation," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 386–394, 2001.

[78] R. Kristiansen, P. J. Nicklasson, and J. T. Gravdahl, "Satellite attitude control by quaternion-based backstepping," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 227–232, 2009.

[79] T. Ohtani, Y. Hamada, T. Nagashio, T. Kida, S. Mitani, I. Yamaguchi, T. Kasai, and H. Igawa, "Robust attitude control using mu-synthesis for the large flexible satellite ETS-VIII," *The Journal of Space Technology and Science*, vol. 25, no. 1, pp. 1_27–1_40, 2009.

[80] Ø. Hegrenæs, J. T. Gravdahl, and P. Tøndel, "Spacecraft attitude control using explicit model predictive control," *Automatica*, vol. 41, no. 12, pp. 2107–2114, 2005.

[81] J. L. Crassidis and F. L. Markley, "Sliding mode control using modified rodrigues parameters," *Journal of Guidance, Control and Dynamics*, vol. 19, no. 6, p. 1, 1996.

[82] E. Capello, E. Punta, F. Dabbene, G. Guglieri, and R. Tempo, "Sliding-mode control strategies for rendezvous and docking maneuvers," *Journal of Guidance, Control, and Dynamics*, 2017.

[83] J. Ahmed, V. T. Coppola, and D. S. Bernstein, "Adaptive asymptotic tracking of spacecraft attitude motion with inertia matrix identification," *Journal of Guidance Control and Dynamics*, vol. 21, no. 5, pp. 684–691, 1998.

[84] L. Eugene and W. Kevin, "Robust and adaptive control with aerospace applications," 2013.

[85] F. Curti, M. Romano, and R. Bevilacqua, "Lyapunov-based thrusters' selection for spacecraft control: analysis and experimentation," *Journal of guidance, control, and dynamics*, vol. 33, no. 4, p. 1143, 2010.

[86] R. Zappulla, J. Virgili-Llop, and M. Romano, "Spacecraft thruster control via sigma–delta modulation," *Journal of Guidance, Control, and Dynamics*, pp. 1–6, 2017.

[87] W. Cai, X. Liao, and Y. Song, "Indirect robust adaptive fault-tolerant control for attitude tracking of spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, p. 1456, 2008.

[88] S. Dubowsky and M. A. Torres, "Path planning for space manipulators to minimize spacecraft attitude disturbances," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2522–2528, IEEE, 1991.

[89] T. Hsia, "Adaptive control of robot manipulators-a review," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 183–189, IEEE, 1986.

[90] L.-C. Fu, "Robust adaptive decentralized control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, no. 1, pp. 106–110, 1992.

[91] H. Seraji, "Decentralized adaptive control of manipulators: theory, simulation, and experimentation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 183–201, 1989.

[92] V. Parra-Vega, S. Arimoto, Y.-H. Liu, G. Hirzinger, and P. Akella, "Dynamic sliding PID control for tracking of robot manipulators: Theory and experiments," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 967–976, 2003.

[93] A. Karakasoglu, S. I. Sudharsanan, and M. K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 919–930, 1993.

[94] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[95] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[96] H.-L. Pham, V. Perdereau, B. V. Adorno, and P. Fraisse, "Position and orientation control of robot manipulators using dual quaternion feedback," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 658–663, IEEE, 2010.

[97] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM, 2010.

[98] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and Adaptive Control*, pp. 317–353, Springer, 2013.

[99] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[100] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[101] H. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems," *Department of Civil and Environmental Engineering, Duke University*, 2011.

[102] J.-y. Fan and Y.-x. Yuan, "On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption," *Computing*, vol. 74, no. 1, pp. 23–39, 2005.

[103] J. Fan and J. Pan, "A note on the Levenberg-Marquardt parameter," *Applied Mathematics and Computation*, vol. 207, no. 2, pp. 351–359, 2009.

[104] M. Osborne, "Nonlinear least squares the Levenberg algorithm revisited," *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, vol. 19, no. 03, pp. 343–357, 1976.

[105] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*, pp. 105–116, Springer, 1978.