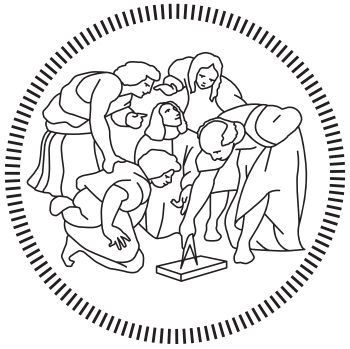
Politecnico di Milano
School of Industrial and Information Engineering
Master of Science course in Space Engineering



Thesis
Search, implementation and testing of adaptive methods for spacecraft's attitude control under actuator faults

Advisor: Marco LOVERA

Thesis by:
Antonio GONZÁLEZ CARVAJAL Matr. 935880

Academic Year 2020–2021

To my parents, without whose support this would not have been possible.

And thanks to Majo, Samu, Sofi, Cedric, Joha, Dani, Ana, Veli, Pedro, Cris and so many others, for making this second year in Milan more wonderful than I could have ever imagined.

Acknowledgments

Abstract

The evolution of the space sector towards smaller and cheaper spacecrafts, combined with the increase in usage of Commercial of the Shelf components, has given rise to new typologies of failures in the spacecraft's Attitude Control System. There is a need to develop fault-tolerant attitude control techniques to counteract the increase in failure probability that comes with the reduction of the component's costs. Nowadays, a series of innovative techniques are being proposed with the potential of outperforming the classical techniques used to deal with these failures. The purpose of the thesis is to investigate one or two innovative techniques and select the most promising methods from sources in the literature. After that, the objective is to test them onto two baselines to determine their applicability and potential to be used in real-world applications. In the thesis, a literature review is presented on the Adaptive control and Model identification techniques. Four adaptive methods are selected among the sources, of which three are successfully implemented. They are tested onto the LUMIO CubeSat baseline and the AGILE mission baseline, which are both cases of spacecrafts controlled with four reaction wheels. Three scenarios are implemented: slew manoeuvre, detumbling, and long-term pointing. The adaptive methods show better behaviour than the original controller (when available for comparison) and they are fault-tolerant, displaying some mission-saving characteristics.

Sommario

L'evoluzione del settore spaziale verso satelliti più piccole e più economiche, insieme all'aumento dell'utilizzo di componenti commerciali e da scaffale, ha provocato la crescita di nuove tipologie di rotture nel sistema di controllo d'assetto delle navicelle. Esiste la necessità di sviluppare tecniche di controllo d'assetto resistenti al guasto per contrastare l'aumento della probabilità di rottura che sopraggiunge con la riduzione dei costi dei componenti. Al giorno d'oggi, vengono proposte una serie di tecniche innovative con il potenziale di superare le classiche tecniche impiegate per gestire queste rotture. Lo scopo della tesi è di investigare una o due tecnologie innovative e di selezionare i metodi più promettenti da fonti della letteratura. Successivamente, l'obiettivo è di testarli su due baselines per determinare la loro applicabilità e potenziale per poterli usare in applicazioni del mondo reale. Nella tesi, è presentata una revisione della letteratura sul Controllo Adattivo e sulle tecniche di Identificazione del Modello. Quattro metodi adattivi sono selezionati tra le fonti, dei quali tre sono implementati con successo. Essi sono testati sulla baseline del CubeSat LUMIO e la baseline della missione AGILE, che sono entrambi casi di satelliti controllate con quattro ruote di reazione. Sono implementati tre scenari: manovra di slew, detumbling e puntamento a lungo termine. I metodi adattivi mostrano un comportamento migliore del controllore originario (quando disponibile per un confronto) e sono resistenti al guasto, mostrando alcune caratteristiche che salvano la missione.

Contents

List of Figures

List of Tables

List of acronyms

ACS Attitude Control System

ASI Italian Space Agency

BLS Batch Least Squares

CoM Center of Mass

COTS Commercial of the Shelf

DCM Desaturation Control Mode

EOL End of Life

ESA European Space Agency

FDI Fault Detection and Isolation

IMU Inertial Measurement Unit

ISS International Space Station

KF Kalman Filter

LEO Low Earth Orbit

LQ Linear Quadratic

LS Least Squares

MRAC Model Reference Adaptive Control

PD Proportional Derivative

PI Proportional Integral

PID Proportional Integral Derivative

RCS Reaction Control System

RLS Recursive Least Squares

RW Reaction Wheels

SCM Science Control Mode

SEU Single Event Upsets

SRP Solar Radiation Pressure

UI User Interface

Introduction

The control of a spacecraft's attitude is a complex but crucial problem for the success of many if not all space missions. It is typically achieved through a dedicated subsystem inside the spacecraft, the Attitude Control System (ACS), which comprises of attitude sensors, actuators able to apply torques, and a control logic that determines the actuation from the available data. Due to the characteristics of space missions, the ACS has to be autonomous for certain periods of time, usually long, and must require minimum human intervention, none if possible. That is why the control logic in the ACS is algorithmic in nature, instead of controlled by humans.

Many techniques have been proposed and refined for the attitude control problem, but until recently most of them assumed no failures in the ACS. The approach followed to guarantee the success of a mission was to ensure that the components had an extremely small possibility of failure. This created highly reliable *ad hoc* components for each mission, but at the expense of great costs.

However, in the last two decades there has been a strong push for the "democratization" of space, that meaning granting access at significantly reduced costs to stakeholders previously excluded from the sector due to the high investments required. The CubeSat [2] platform is the most prominent example of this trend of smaller, cheaper spacecraft. The new participants such as small companies, student associations, and academic institutions such as Politecnico di Milano, cannot afford or are not willing to cover the expensive traditional *ad hoc* design process of ACS components to ensure the previously achieved low probability of failure, particularly in the actuators field. This brought the need to develop control techniques that took into account and could compensate for failures, and several branches have been developed particularly in the last two decades.

Another factor is that, to further reduce the cost, a move towards Commercial of the Shelf (COTS) components has occurred. They are components not designed for an specific mission, but for generic applications. Then the designers of each mission buy and adapt them to their spacecraft. These COTS components are vulnerable to particular types of failures, such as transient events and even complete failures. These are typically treated differently in traditional ACS due to their higher reliability.

Crucially, the countermeasures that the traditional spacecraft deploy against the harsher failures, like having redundancy against a complete sensor/actuator failure, depend heavily on the presence of Fault Detection and Isolation (FDI) systems. FDI would provide information to the spacecraft on the specific failure that is occurring, yet their presence cannot be guaranteed for COTS components on these smaller spacecrafts.

The state of the art to deal with these particular failures are a series of "classical techniques" that are already in use, which consist on: filtering of measurements, rejection of measurements, default commands (such as "safe" modes), and accelerated restart procedures. Beyond them, there is a series of innovative techniques, in development or experimentation phases, which show promising potential to outperform the "classical techniques". A list of the innovative techniques considered will be provided in section 1.1. Among them it is highlighted the presence of Adaptive control and of Model identification, because they are the ones that were explored deeply in this Thesis. While Adaptive control is a modern technique and its application to the spacecraft attitude has almost completely been developed in the last fifteen years, Model Identification has a longer history, although its proposed application to failure tolerance is recent.

There is extensive literature published for most of these methods. However, the many variables that change from one publication to another, and the lack of publications replicating the findings of previous ones, makes it hard to select good technique for any single mission. A given research can focus on a specific case of application, type of actuator, of failure, availability or not of certain sensor types, etc. Moreover, it is no secret that many publications lack the necessary detail to ensure the replicability of their results, or even the implementation of their techniques. It is necessary to develop a criteria to evaluate sources which can lead to selecting techniques suitable to a given application.

Thus, the objectives of this thesis are stated as follows:

1) Identify the particular failures that affect the ACS when COTS components are used in their architecture.

2) Identify the innovative techniques that are currently proposed to deal with the specific kind of failures discovered on 1).

3) Select a small number (one, two or three) of techniques from the ones identified in 2) and conduct a literature review on them.

4) Develop a criteria to evaluate the sources found and shortlist a few (between three and five) of the most promising sources.

5) Implement the methods of the sources and try to replicate their results through simulation.

6) Apply the successfully implemented methods of 5) into a first baseline to check their capability to adapt to multiple missions.

7) Apply the methods that result successful from 6) into a second baseline to confirm their flexibility.

The entire project will be developed by a single author under the supervision of the Thesis mentor.

The Thesis is divided into four chapters. In the first, the problem is stated, and an investigation onto the COTS failure types and innovative techniques is performed. Two techniques are selected, Adaptive control and Model identification, and a literature review is conducted onto them. The sources found are analyzed and four methods of the Adaptive control category are shortlisted for implementation. This corresponds to objective 1) though 4).

The second chapter explains the mathematical layout of the common parts of the four shortlisted methods, then the unique parts of each method specifically, and shows the results of the attempted replication of the originals, corresponding to objective 5).

The third chapter applies the three successful methods (one was unsuccessful) onto the LUMIO CubeSat baseline [1] and compares the results among them as well as with the original controller proposed for the mission, tackling objective 6).

Finally, the fourth chapter implements the three methods onto the AGILE mission spacecraft [3], and discusses the results in comparison with that was obtained in Chapter 3, dealing with objective 7).

The Thesis finishes by discussing the obtained results in the Conclusions summary.

Chapter 1

Problem statement, research and scope

This chapter begins by explaining the motivation for this research, laying out the groups of COTS components and the challenges that each of the COTS group represents. After narrowing down the group of COTS to focus on and the failures to be expected of them, it lists several algorithmic innovative techniques that could be used to mitigate them, selecting two of them to investigate, Adaptive control and Model identification. In the subsequent two sections, after a quick broad explanation of each technique and some organizational clarifications, it presents each source found, first of Adaptive control and later of Model identification. The chapter ends with the conclusions and the selection of the techniques candidates for implementation in the next chapter.

1.1 Applications of innovative techniques on modern ACS

New applications developed in the space environment in the last two decades have favoured a move towards smaller, standardized and cheaper spacecrafts. The rise of the CubeSat platform [2] is the quintessential example of this spacecraft trend, with over 1500 launched so far [4]. Today, many academic and non-academic organizations, including Politecnico di Milano, are in the process of designing CubeSats due to its affordability and versatility in function.

This drive in the reduction of cost for new spacecrafts is enabled also by the surge of COTS hardware. The philosophy behind this concept is that the functionalities of some spacecraft subsystems are similar enough across a wide range of satellites to support development of common components. These components would thus be chosen by the designers according to the mission needs, instead of designed *ad hoc*, and then integrated into the design, reducing the cost of said component. The ACS subsystem is one example that benefits from this premise.

This approach has enabled amateur and academic designs to come to fruition that otherwise would not have had the funds to develop the parts, significantly advancing the space sector. However, it has also come with problems.

The use of COTS supposes a departure from typical spacecraft design and creates new challenges. A joint inquiry of AIRBUS and the European Space Agency (ESA) on the topic [5] discovered that COTS on the ACS field could be classified in two broad families:

- Firstly, there are those designed for high-end applications such as commercial constellations, whose cost is not reduced in such a dramatic fashion compared to the second group, but that offer similar reliability compared to traditionally developed hardware. The main problem with this group are Single Event Upsets (SEU) that could significantly alter transient behaviour of the component.

- Secondly, the most low-cost COTS components suffer from significantly lower reliability, being susceptible to permanent failure. However, they offer the greatest cost-saving opportunity so in this research they are deemed the most indicated to amateur and student-developed applications.

For reasons of their applicability and capability of the author to analyse the related potential failures, in this research it was decided to focus on the second group. This same inquiry [5] also reached two conclusions relevant to the later conduct of this research. The first of them is that the best way to mitigate the risks posed by using COTS components is at system level. The second, that the Reaction Wheels (RW) component is the one that poses the most "threats" but, tied with another component, offers the most "opportunities" with its usage.

One of the examples of mitigation at "system level" is that of algorithmic software techniques. These have several advantages, one of them being that they can be added to currently operating satellites as software-only updates. They can also be adequately explored using simulations, before subsequent implementation and testing. Classical techniques used for this end such as filtering and data rejection have been shown to have limitations, thus a set of innovative techniques was proposed in [5] to deal with the effects of COTS:

- Machine Learning

- Hardware anomaly identification techniques (other than ML based)

- FDIR based on structural analysis

- Multiple-model control design

- LPV modelling, identification and control
- Adaptive control
- Adaptable estimation techniques
- Model identification
- Non-holonomic feedback control techniques of under-actuated spacecraft
- Hybrid control

It was decided to explore two methods in the initial research stage, those being Adaptive control and Model identification, to later focus on the one that showed the most promising outcomes. It is worth mentioning that the above-listed methods are not necessarily mutually exclusive. Many proposals have been made of techniques that use a combination of methods to maximize their respective strengths. Indeed, some articles found during the literature review phase make use of one of the selected methods in combination with another, and there are even examples of using Adaptive control and Model identification together.

The approach followed to find publications for the literature review was double fold. Searches by keywords such as "fault", "'tolerance", "spacecraft", "attitude", "adaptive", "control", *etc.*, was carried out in the main scientific publishers' online libraries. The searches have spanned the sites [6], [7], [8], and [9]. Also, the sources mentioned in [5] have been investigated. To expand further the search, the references cited by all publications found have also been explored. To select the sources to explore in depth, the title and the abstract have been the deciding factors, and preference was given to the sources the closer they were to fault tolerant application, using the selected methods, and dealing with spacecraft's attitude dynamics.

1.2 Literature review on adaptive control

The family of techniques referred to as Adaptive control have in common the presence of a controller structure with parameters that are time-varying. These parameters "adapt" to the conditions of the system by the implementation of differential equations called "adaptation laws", which can use any number of available signals. Among the measurements to drive the adaptation process could of course be the presence of failures in the sensors or the actuators of the ACS. The adaptation can be assisted by the use of a reference model, possibly obtained by Model identification, and in that case it is called Model Reference Adaptive Control (MRAC).

On the process of making the literature review, preference has been given to the information sources that directly deal with the attitude control problem.

Adaptive control has multiple applications, even within the space sector, and papers can be found in many areas. For instance in orbit control, the series of papers [10], [11] and [12] uses an adaptive update law with the objective of controlling the spacecraft orbit using aerodynamic drag. These applications are very worth looking into, but if any cross-adaptation of the method to the attitude problem could be made at all, it would require significant effort and expertise.

Before starting to introduce the findings of this phase, it is necessary to explicitly state that any criticism intentionally or unintentionally directed to the sources found is made from a humble point of view, and can be the result of misunderstanding or lack of expertise on the side of the author of this Thesis. It is at no point the intention of this Thesis to undermine or attack the work of expert researchers, specially given the inexperience of this author on the fields treated. This also applies to the rest of the chapter and by extension to the entirety of the document.

A significant praise has to be devoted to the archival efforts of [13] for making a comprehensive review of the most recent developments in fault-tolerant control using several methods. On the way of structuring the results of this review, it has been difficult to select a criterion to group the sources in a consistent way, as they can vary in so many independent dimensions (application, method used, faults dealt with, simulation limitations applied or not, results, *etc.*). In the end it has been decided to lay out the sources one by one keeping a consistent narrative by maintaining the ones that use similar methodology close together.

1.2.1 Sources on Adaptive control methods

The first and oldest paper that has been found of interest is [14]. It uses an adaptive law with the inertia matrix components as adaptive parameters to tackle the problem of attitude tracking with unknown inertia. This has the added benefit of identifying the spacecraft inertia, which can then be available for other uses. Results show that stable tracking is achieved, but with severe oscillations during transient behaviour. The control law would be relatively easy to implement. The main criticisms are that it does not include disturbances, it does not model actuators, nor features fault-tolerance. It also needs the commanded attitude to be "rich enough" (see the definition in the paper) for the inertia estimate to obtain the correct values. Actually, the contributions it does achieve, global convergence and inertia identification, can be done separately, as it is going to be seen during the rest of the chapter. Finally, the formulation is done in Euler parameters, and the demonstrations carried out by Lyapunov stability analysis.

In fact, it is going to be said now that the majority of the studies discussed here use Lyapunov methods [15] to prove their convergence, sometimes recurring to highly complicated Lyapunov candidates. The approach has the advantage that it can prove local or even global stability, however it does not offer any *a pri-*

ori guarantees on the transient behaviour, speed of convergence, or the behaviour inside the stable defined region, so these analyses should always be backed up by simulations or experiments.

A more modern application is found in [16]. It undertakes the attitude tracking of a microsatellite with changing targets. The adaptive control law is of a structured form that resembles a Proportional Derivative (PD) controller, but with the coefficients as time-varying adaptive parameters. The objective is to substitute the original switching "PD-speed bias" controller of a PICARD satellite with an adaptive one that transitions smoothly between the two. The method managed to replicate both original controllers in their realm of applicability, removing the "jump" between the two. Actuators this time are modelled, one magnetometer and three RWs, and it is taken into account the wheels saturation and possible positioning measurements outage.

The method was tested on an End of Life (EOL) PICARD satellite with better results than the original switching controller. The main criticisms are that it treats the three axes independently, relying on swapping to a "safe mode law" if errors grow too large, and that it is again not an application demonstrating fault-tolerance. The controller would also be moderately difficult to implement. Although some controller parameters can be computed by optimization problems, many others have to be chosen by trial and error. Even the authors say finding a good combination is challenging, which would not be helped by the fact they do not seem to report the controller parameters used in the tests.

The first application of this review that treats fault-tolerance is that of [17]. With quaternion formulation, it addresses the attitude tracking with changing target of a rigid body spacecraft. The adaptation law features an extra dedicated parameter to control the transient response. The simulations are carried out with actuator loss of effectiveness and saturation. It is worth noting that in this context, "loss of effectiveness" can be the result of complete failure of an actuator in a redundant configuration. For instance, if a redundant two thruster configuration in one axis suffers an outage, the equivalent "loss of effectiveness" would be of 0.5. It does not necessarily mean any actuator is working at half the commanded value.

It also claims that it can deal with an unknown inertia, and its control indeed does not use the inertia matrix, although control under different inertias is not demonstrated. Results show improved response in any case compared to a PD controller. Also, the improvement of transient response is best for moderate values of the transient control parameter. The drawbacks are that it does not model the actuators, and it needs upper and lower bound of actuator effectiveness. However, these limits can be selected very aggressively in case FDI is not present, but it nonetheless would improve performance. Implementation at first glance would be easy, with only one first-order differential equation for an adaptation parameter

called k.

A similar method with some extended capabilities is found in [18]. Again, the method is applied to an attitude tracking rigid body spacecraft with quaternion formulation. The (unmodelled) actuators suffer from loss of effectiveness and bias. The main novelty of this paper is that it introduces a command filter for velocity tracking based on the hyperbolic tangent. This allows to explicitly impose a maximum limit in the angular velocity of any manoeuvre done under this control. Results of simulations also showed a decrease in steady state errors for fault-free and specially for faulty case, and the constraint on angular velocity was respected. Together with the unmodelled actuators, a criticism could be that it does not show the behaviour under varying inertia, although the inertia is not used in the control scheme so it presumably could withstand it. The method is expected to be easy to implement, with two additional first order equations and only five scalar parameters to tune.

Departing from the methodology of the first four sources discussed, the method in [19] proposes an adaptive neural network to estimate a nonlinear term, inside a smooth function used to simulate saturation. It also uses the same hyperbolic tangent angular velocity command filter as before to control the maximum angular velocity reached during manoeuvres. Simulation results show that control saturation implemented resembles real behaviour, and the angular velocity limit is respected. The scheme is applied to attitude pointing, instead of tracking, which is a criticism. Also drawbacks are the fact it does not feature fault-tolerance and does not model actuators. The source claims that the controller would be able to deal with model uncertainties and disturbances. However, it does not implement them in the simulations, so that claim would have to be checked. The difficulty of implementation, given the knowledge of the author on the neural network field, could be stated as "unclear", but would probably be hard due to the concise explanation of the source, without an step-by-step guide to replicate it. In terms of computational power, once it is designed, it seems to require one differential equation per network node for parameter weight adaptation. For another interesting source also proposing adaptive neural networks, which is not going to be discussed here because it was deemed far too complex for an implementation, see [20].

Two of the authors of [20] also proposed another approach in [21] using adaptive fuzzy logic for the fault-tolerant attitude tracking controller of a rigid body spacecraft. It uses fuzzy rules to estimate nonlinear terms in the error dynamics. It also uses a virtual controller for that same error dynamics. Because it bases the adaptation on the norm of the weight matrix, it only has adaptation laws of two parameters, with other three additional differential equations for the error vector used in the method. In simulations, it confronts control saturation, distur-

bances, uncertain inertia, actuator loss of effectiveness, outage, and even control sign reversal, which is very interesting, although for this last case the settling time is sensibly longer. The criticisms would be that actuators are not modelled and the transient response is very bad, with no control on the slew trajectory, so it is not useful for systems with forbidden directions. Implementation would be moderately difficult, with six constants and one 3×3 matrix to select, and the fuzzy logic functions to design.

Another line of research for adaptive methods has been that of sliding mode techniques, of which [22] is an example. The problem of attitude pointing of a rigid spacecraft with four thrusters is addressed, with an additional integral term added to the traditional sliding surface combination of error quaternion and its derivative. The result is an adaptive control law with four tuning parameters. Implemented on the simulations shown are thruster outage, limitation, unknown inertia, and the case when the thruster gets stuck, at 50% and 100% of actuation. The response is improved on nominal condition when compared to a Proportional Integral Derivative (PID) controller, and to the method in [23], that is going to be discussed later. Response is heavily improved on outage and 50% stuck condition, and at 100% stuck condition this method stabilizes the system while the others cannot. As drawbacks, it can be said that it does not explicitly impose limitations on the actuators inside the control scheme (it does apply limitations in the simulations), and performs attitude pointing, instead of tracking. Implementation difficulty would be medium, with four first order differential equations and also four scalar and four matrix parameters to tune.

From one of the same author as [22] is [24]. Being an earlier paper, it uses a normal sliding surface, only error quaternion and its derivative, on the attitude tracking of a flexible body spacecraft with four RWs. Flexibility effects, disturbances, actuator loss of effectiveness and complete failure are taken into account. The method uses adaptation to estimate the unknown inertia matrix norm. Results show attenuation of flexible oscillations and stabilization when one RW fails. However, the method is not well explained on the source, with the results showed being very few, and after intense reading, major doubts still remain, like the dimensions of some control variables. Implementation would be also be made harder by severe restrictions on controller parameters and not reporting the controller numbers used on the simulations.

One of the methods that [22] compared its results to was the one in [23]. This source, also one using a standard sliding surface variable, uses adaptation to estimate for unknown nonlinear terms that appear during its theoretical layout. It comes up with three stages of development of what could be argued is the same controller, one for unmodelled three-axis control, another one accounting for the matrix of actuator distribution, and a last one explicitly implementing limitations

on the actuators. The method is applied to the problem of attitude tracking of a rigid spacecraft with six thrusters. Simulations are carried out under thruster loss of effectiveness and up to three outages, also considering thruster limits and unknown inertia. Even in the worst failure cases, tracking is successfully achieved. However, it does not compare its results with any other alternative scheme. The implementation would be easy, with one added differential equation and four tuning parameters.

One final block of adaptive spacecraft's attitude control papers would be those using MRAC, the first of the three discussed here being [25]. For the attitude tracking of a rigid spacecraft, it uses MRAC to generate a virtual control torque, and then performs control allocation based on a 1-dimensional minimisation problem for the unconstrained case. This method needs Fault Identification and Magnitude Estimation to function, so that is not ideal given what is expressed in Section 1.1. The problem is that for the saturated case the minimization turns into a convex Second-Order Cone problem, relying on a software called CVX for solving. Although results show an improvement of convergence speed and tracking steady-state errors, the PD it is compared against already performs well. It is not tested on more challenging cases, and the procedure behind implementing the unconstrained (no saturation) case is already quite complex, so implementation would be hard.

The second source using MRAC is [26], on both the problem of translational positioning and that of attitude pointing of the SPHERES satellites. These are reconfigurable satellites with the possibility of attaching add-ons with different tool sets. This causes variation in mass and inertia, so they remain unknown. The MRAC was design with a sliding surface variable in both position and attitude control. An advantage is that the scheme was experimentally tested on board the International Space Station (ISS) using the real SPHERES hardware. Results show a significant performance increase compared to the PD baseline controller, especially when more add-ons are plugged in, which vary the inertia and mass properties significantly. As drawbacks of the scheme, it does not feature fault-tolerance, and it assumes perfect knowledge of the position, attitude, velocity, and angular velocity, although later in the real-world test the method seems to be able to handle those uncertainties. Implementation difficulty would be moderate. Two three-by-three matrices and four scalar parameters have to be determined. The authors recognise finding the correct parameters for the application was a challenge, and mainly done by trial and error using simulations. The controller parameters used in the tests are reported.

The last source explored is [27], from one of the same authors as [26], and it deals with the problems of translational position tracking and mass determination. For this source, mass is uncertain, and it combines an Extended Kalman Filter

(KF) with a MRAC together into a composite adaptation technique. Simulation mainly show that the mass parameter is correctly identified through continuous estimation. That means the extended filter design prevents the estimation from closing into wrong values. As drawbacks, obviously that it deals with translational dynamics instead of attitude, and that the source provides very little explanation on the method itself. Implementation would be very hard due to the authors not providing numbers nor parameters used for the results and the explanation being so high level.

1.3 Literature review on Model identification

A Model identification technique is that one which uses data collected during the real life experience of a system to determine the value of parameters of a model that is already predetermined beforehand. Many techniques have been proposed over the years to do that, depending on data quality, application, known information on the dynamics to propose a less or more accurate model, *etc.*, giving place to the many families inside the identification world. Of key importance for Model identification techniques is that the predetermined model is able to sufficiently closely replicate the behaviour of the real world system. If that condition is not met, more often than not the identification process will be completely worthless.

A remarkable characteristic to differentiate techniques is if they are performed offline, once all data has been collected, or they allow for online implementation, which is much more suitable for on-board application. Of course the online methods must involve no more than a manageable amount of computational power in order to be applied to satellites. Other distinction is that between grey-box techniques when the structure of the model is already known, normally derived from physical principles, and black-box techniques, when the structure is unknown. The main approach for black-box identification is through trial of generic-system dynamics of multiple levels, or analysing the data beforehand to try determining the level of the system, so at first glance that makes them harder for online use.

For Model identification is has been much harder to find applications pertaining fault-tolerance than for Adaptive control. Rather, when searching for sources using identification in spacecraft and fault related topics, the majority found referred to FDI, diagnosis of sensors, actuator alignment estimation, and above all mass and inertia estimation. If the reader remembers, some applications discussed in Section 1.2 already dealt with this last topic. The consequences of this lack of sources in pure fault-tolerance will be discussed in the chapter conclusions, Section 1.4. As of the presentation of the sources, the same methodology followed for Adaptive control is used here, keeping the similar sources following one another to add continuity, despite the many-dimensional differences between them.

1.3.1 Sources on Model identification methods

The first method dates from the early two thousands and is developed in [28], which lays out the mathematical procedures for estimation of an spacecraft direct and inverse inertia matrices, Center of Mass (CoM) position, thruster magnitude bias, and total mass. It does it by simplifying the system equations of motion through neglecting terms until they are set in linear form. The method is offline, although a modification to turn it to online via Recursive Least Squares (RLS) is proposed, and implementation is deemed easy.

The section of [28] on inverse inertia identification and CoM position is applied in [29] for both offline identification, using Batch Least Squares (BLS), and online identification, using RLS. The source implements a KF to reduce angular velocity measurements noise, and resets the filter after each time the firing configuration of the thrusters changes. One challenge is that the source wanted to do all with the version that only uses angular velocity measurements, with no position measurements. Results show that offline and online methods converged on the same values, but for some tests the converged values are incorrect, which is blamed on other unknown effects. Also, just after each filter reset, the angular velocity variance is very high. Implementation would be easy because the method is well explained at length, but it does not feature fault-tolerance.

The source [30] addresses the problem of a spacecraft's thrusters FDI online. The procedure is to select a preset limited combination of failure modes (in this case forty), and then compute the nominal acceleration disturbance that would correspond to them if they were to happen. After that, it uses Maximum Likelihood estimation to determine which among the failure conditions, or the no-failure condition, is more likely. In simulations it implements thrusters pulse-to-pulse strength variability, constant thruster bias, inertia matrix bias and mass bias, CoM location offset, and again only uses angular velocity measurements, supposing attitude measurements are not available. The results show the typical failure detection time is 0.5 seconds after firing. Most failures are detected before 1 second. Failures with similar effects take up to 5s to distinguish. Also, 99.98% of the times the failures are correctly identified. The main drawback is that failure modes have to be predetermined beforehand, with failure modes not in the set impossible to be identified, even with the possibility for them to be confused with erroneous predetermined ones. It also does not feature fault-tolerance, only diagnosis. Implementation is deemed moderately difficult, mainly due to the need of identifying the failure modes.

The same application of spacecraft thrusters' FDI is targeted in [31]. Considering uncertain inertia, thruster misalignment, partial or total failure, and only attitude measurements available, the authors propose two methods applied on the linearized dynamics. The first is a FDI algorithm that computes the expected

total moment of inertia and compares it to the measured one to isolate failures. The second is an Unknown Input Observer that results in a structure sensitive to certain failures and not to others. It also features efficiency estimation, actuator torque matrix, and mean thrust calibration. In simulations, most of the time both methods detect and isolate the correct failure, but in both cases there is a non-negligible number of wrong identifications, thus they are unreliable. It is a work in progress, and was tested on real telemetry data with moderately good results. Implementation though would be hard, due to a high level explanation of the method, with no concrete numbers to put it into practice.

Another master's thesis like [29] is that of [32]. In this case the offline estimation of direct inertia matrix and actuator misalignment is confronted. It considers noisy and biased measurements, as well as possible non-availability of angular velocity measurements. The author proposes an Instrumental Variable method to tackle types of noises that deviate from normal Gaussian white noise, a State-Variable low pass filter to obtain angular acceleration when angular velocity measurements are available, and a Second-order Butterworth Filter combined with central differentiation for when only position measurements are available. Simulations show improved estimation with respect to Least Squares (LS) and other methods, and the technique was experimentally tested. Implementation would be complex, with necessity to design the instrument "Z" to apply the method. It is also not very applicable to the target problem, being offline and not used for fault-tolerance.

The method in [33] handles the case of the capture of an asteroid using a spacecraft, with the severe uncertainties on inertia and CoM that this scenario carries. Also takes into account the inability to produce net torques on the combined spacecraft-asteroid system. The process is offline, by reformulating the problem as a convex optimization problem and using MATLAB's CVX solver. It adds convex constraints on the CoM positioning and inertia characteristics to the convex solver to ease the solution finding. Results showed the ability of the technique to estimate the target parameters, while the standard LS method failed to do it. The time to estimation was reduced with each added convex constraint. Of course, a criticism is that it uses a software solver, so the technique is not actually explained. It also assumes perfect knowledge of all measurements and no disturbance. Implementation would be moderately difficult.

The second to last method, proposed in [34], utilizes a two-stage KF in the failure estimation of a three-RWs rigid spacecraft. Also considering gyro sensor bias and failure, it includes some of the system states inside the state evolution matrix "A" of the filter. The actuator effectiveness factor is then included in the filter for estimation. The source also proposes two control laws with fault-tolerance characteristics, first a reconfiguration controller, and second a direct accommodation

controller. Results show that they comfortably deal well with RWs failures and with gyro bias separately, but no combined failures are attempted. As criticisms, it assumes a principal inertia matrix for the mathematical development, and it is unclear how to extend the controllers to other actuators other than one reaction wheel per axis. Implementation is assessed as moderate to hard, with some steps being unclear and no justification given to the parameter values used, although at least they are provided.

Finally, the method in [35] proposes an augmented linearized KF, of enormous proportions, including as states all the target parameters for estimation. To work around the observability problem of both spacecraft inertia and RWs inertia, it sets the value of the term (1,1) of the inertia to the nominal one. This supposedly does not affect attitude estimation nor the control. The scheme adds a torque in the null space of the RWs distribution matrix to be able to estimate the direction of the RWs misalignment. In the source it is stated that it necessitates angular momentum bias to estimate the inertias. Another procedure followed is amplifying process and measurement noise at the beginning of the simulations to avoid the KF getting closed around wrong values. Starts at 100 times amplification, then drops to 10 times, and later to nominal. The Scheme is applied in a four RWs spacecraft with residual magnetic pole, attitude measurements temporary outage and no gyro sensors present. The main problem is that it needs numerical tricks to avoid the filter from closing and estimating the wrong values, it needs custom calibration manoeuvres, and does not deal with fault-tolerance. Although the KF can be reduced if estimation of some states is not needed, the number of states is still high, so implementation difficulty is considered medium.

1.4 Chapter conclusions

For the large number of sources introduced in this chapter, now it is necessary to filter them down and come to useful conclusions to advance on the project. As is was stated in Section 1.1, the objective is to find a shortlist of a hand-full of papers that show promising potential to handle complete actuator failures of the kind that could experiment COTS components of group two.

The efforts of the last part of the literature review phase have been focused on organizing and classifying the sources to come up with a list of between three and five methods to implement. This is thought this way so as if there are one or more techniques that fail at implementation, work can still be done with enough material to move forward with the rest of them. The results are presented in the table inside Appendix A.

The table presents in the first column the problem that the sources deal with. The second column names the year, author and reference of the source. A single source may appear many times in different problem categories if it tackles more

than one problem. The third column tries to present in few words, as a reminder, the method used by the source to deal with that problem. Finally the numbering columns summarizes the analyses made to the sources in three dimensions. The evaluation has been performed with respect to the criteria of: Completeness, Easiness, and Results Quality. The Completeness category analyses how well the method is explained in the source. For instance, step by step explanations achieve higher score than high level ones. Easiness represents the complexity of the technique, and Results Quality assesses how good the results presented are using the technique, also weighting in the applicability to this Thesis. Valuations of each method may vary depending on the specific problem category. Finally the last column presents the Overall score achieved.

Thus, after all this analysis, it was time to make a decision. As is expressed at the beginning of this chapter, two broad innovative techniques were going to be researched, to then select the one that was the most convincing. Well, given the results of the sources review in Appendix A, the lower applicability, as was mentioned before, of Model identification to fault-tolerance, and taking into account the personal preference of the project's author, after consultation with the Thesis mentor it was decided to proceed with Adaptive control. Moreover, due to the sources evaluation results outlined in Appendix A, the methods selected for tentative implementation have been [18], [17], [23], and [22]. For clarity, this methods will be referred to as 2018 Qiang Shen et al.[18], 2014 Danyal Bustan et al.[17], 2008 Wenchuan Cai et al.[23], and 2011 Qinglei Hu et al.[22] occasionally throughout the rest of this Thesis. In the next chapter, each method will be explored and explained in-depth, and the original results reported will try to be replicated.

Chapter 2

Selected methods

This chapter introduces first the mathematical layout that is common for all four selected methods, followed by a high level explanation of the simulation software models used for implementing the techniques. After that it is presented, one method at a time in different sections, the mathematical equations that allow for the implementation of each technique, and the results obtained by simulation compared to the original results reported by the sources. The chapter ends with the conclusions where all discoveries are summarized.

2.1 Common framework for all methods

To begin this chapter with, it is going to be discussed the generalities that apply to all selected methods. The first explanation will be of the attitude dynamics of a rigid body spacecraft. Arising from the Principle of Conservation of Angular Momentum, the equations of motion that describe the rotation of a rigid body, known as Euler Equations, are:

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}x} = \mathbf{M} \tag{2.1}$$

$$\mathbf{I}\dot{\mathbf{w}} + \dot{\mathbf{w}} \times \mathbf{h} = \mathbf{M} = \mathbf{u} + \mathbf{d} \tag{2.2}$$

where $\mathbf{h}$ is the 3×1 vector of spacecraft's total angular momentum, $\mathbf{M}$ is the 3×1 vector of total external moments (torques) applied on the spacecraft, $\mathbf{I}$ is the 3×3 inertia matrix, $\mathbf{w}$ is the 3×1 vector of angular velocity, $\mathbf{u}$ the 3×1 control torque vector, and $\mathbf{d}$ the 3×1 disturbance vector. Note that $\mathbf{h} = \mathbf{I}\mathbf{w}$, and that equation (2.2) only applies when the magnitudes are measured with respect to body axes.

It has not been a purposely selected coincidence that all four chosen sources use the formulation in quaternions as their choice of attitude parameters. To be fair, most of the sources investigated used them because they are universally accepted

as a versatile, singularity-free attitude representation. The attitude kinematics in this representation are governed by the equations (2.3) and (2.4).

$$\dot{q_0} = -\frac{1}{2}\mathbf{q}^T\mathbf{w} \tag{2.3}$$

$$\dot{\mathbf{q}} = \frac{1}{2}(\mathbf{q}^\times + q_0 I_{3\times 3})\mathbf{w} \tag{2.4}$$

where q_0 refers to the scalar part of the quaternion, $\mathbf{q}$ to the 3×1 vectorial part of the quaternion, $I_{3\times3}$ denotes the 3×3 identity matrix, $*$ is the normal multiplication operator, and the operator $\mathbf{w}^\times$ is the skew symmetric matrix that is equivalent to the vectorial product if it is applied to the first vector, that is:

$$\mathbf{w}^\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{2.5}$$

To address the difference between the current attitude and the commanded desired attitude, and thus be able to achieve pointing on different directions and tracking, the sources also use the concept of quaternion error $(q_{0_e}; \mathbf{q}_e)$, that is defined as the relative orientation between the body frame B and the desired frame D. The quaternion error kinematics can be computed through equations (2.6) and (2.7).

$$\mathbf{q}_e = q_{0_d}\mathbf{q} - q_0\mathbf{q}_d + \mathbf{q}^\times\mathbf{q}_d \tag{2.6}$$

$$q_{0_e} = q_{0_d}q_0 + \mathbf{q}_d^T\mathbf{q} \tag{2.7}$$

where $(q_{0_d}, \mathbf{q}_d)$ indicates the quaternion representation of the desired attitude. In short, $\mathbf{Q}_e = \mathbf{Q}_d^{-1} \bigotimes \mathbf{Q}$, where the operator $\bigotimes$ represents the quaternion multiplication and $\mathbf{Q}$ represents the full quaternion with scalar and vectorial part $(q_0, \mathbf{q})$. Note that 2018 Q. Shen *et al.* [18] has a slight error when defining the error quaternion, and its math actually corresponds to the quaternion multiplication in the opposite order than is reported there. That is why its equations are different from [17] and [23]. However, the source is consistent throughout the document, and this discrepancy does not affect the method. In the implementation, a sign switch on the desired attitude has been added for this technique so it uses the same formulation as the rest.

The angular velocity error is defined in equation (2.8):

$$\mathbf{w}_e = \mathbf{w} - \mathbf{C}\mathbf{w}_d \tag{2.8}$$

where $\mathbf{w}_d$ is the 3×1 vector of angular velocity of the desired frame, and $\mathbf{C}$ is the 3×3 rotation matrix corresponding to $\mathbf{Q}_e$, defined as $\mathbf{C} = (q_{0_e}^2 - \mathbf{q}_e^T\mathbf{q}_e)\mathbf{I} + 2\mathbf{q}_e\mathbf{q}_e^T - 2q_{0_e}\mathbf{q}_\mathbf{e}^\times$.

In the following sections, the methods are going to be explained only to the point that allows for its successful implementation. That is, content like the stability demonstrations and the regions to which the techniques are guaranteed convergence are going to be left to the sources if the reader wants to have more information about the mathematical formalities of the techniques.

2.1.1 The simulation environment

The aim of this subsection is to provide a quick explanation on the simulation resources developed in MATLAB and Simulink used to test the methods and implement the baselines. An example of the Simulink environment that carries out the simulations is presented in Figure 2.1. In blue, the subsystem "Euler Dynam-

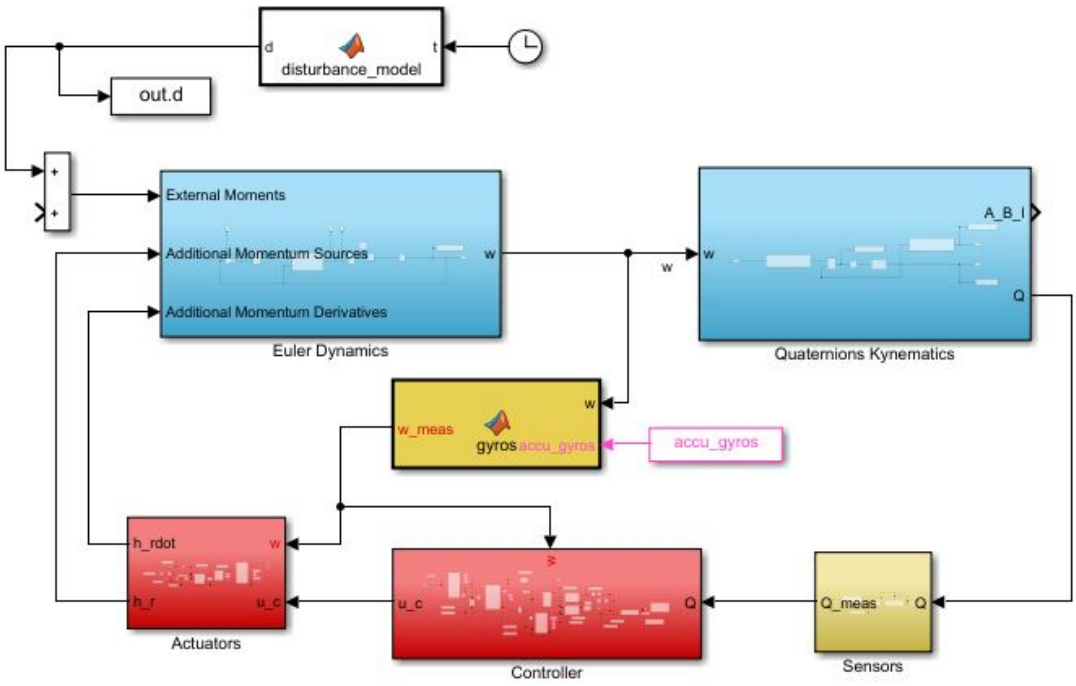


Figure 2.1: Example of Simulink blocks.

ics" implements the Euler Equations (2.2), inputting the external moments, and additional momentum sources (that do not correspond to the free body rotation of the spacecraft) and their derivatives, and outputting the spacecraft's angular velocity $\mathbf{w}$. The "Quaternion Kinematics" subsystem, also in blue, inputs the angular velocity and outputs the attitude quaternion $\mathbf{Q}$ of the spacecraft, as well as its equivalent 3×3 rotation cosines matrix $\mathbf{A}_I^B$.

In red, the subsystem block "Controller" is where the different control laws are implemented. It inputs the spacecraft's angular velocity and attitude quaternion, and outputs the commanded control torque $\mathbf{u}_c$, which feeds into the "Actuators"

subsystem block. Note that, for this block, 2008 Wenchuan Cai *et al.* is an exception, as the method computes directly each actuator's required torque, instead of the 3×1 vector of three-axis virtual control torque.

The "Actuators" block determines the behaviour of the actuators from the commanded control torque and angular velocity. The output can be in the form of angular momentum (and its derivative) in case of, for instance, control by RWs, or in pure moment applied to the spacecraft in case of, for instance, control by thrusters. Note that this block can be missing entirely when the actuators are not modelled, like in the original versions of the sources, and then the commanded control torque inputs directly onto the "Euler Dynamics".

In yellow, the subsystem block "Sensors" represents the attitude position sensors, and the function "gyros" represents the angular velocity sensors. Their purpose is to introduce the error measurements of the attitude sensors and angular velocity sensors, respectively. They have been a late addition to analyze the scenarios "Long Term", and thus they only appear in these cases.

Finally, the function "disturbance_model" explains by its very name its functionality, creating the external disturbance affecting the spacecraft. For the second baseline, in Chapter 4, the simple model has been discarded in favour of a more refined physics-base model. This modification is explained in detail in Chapter 4.

In terms of organization, for each study that has been carried out, a different folder has been created that allowed for the modifications required by each method and scenario. This approach has resulted in a large number of folders with very similar, if not almost identical, Simulink models. This is not ideal, and is it recognized by the author that there may have been better ways to organize the simulations. Maybe, implement in the same file several models with a "switch" to select which one to use. Also, it is noted that Simulink cannot open two project files with the same name, which is definitely a nuisance. It is encouraged to anyone expanding onto this Thesis work to not repeat the same mistakes in organization as where committed here.

The data to set up each simulation is generated a MATLAB ".m" file in the same folder as each Simulink model. The file also launches the simulation itself, except for the "Long Term" scenarios. This is done this way because the extended simulations take more time and in the Simulink User Interface (UI) the progress can be tracked, whereas in the standard MATLAB windows this cannot be done. Detail instructions on how to use them are provided at the beginning of each ".m" file, as exemplified in Figure 2.2.

2.2 2018 Qiang Shen *et al.*

The presentation of this method will begin by explaining the filter for the commanded angular velocity based on the hyperbolic tangent. It uses the value of

```
%% CubeSat Results of Antonio Gonzalez Thesis

% Scenario: Slew Manoeuvre
% Method: 2014 Danyal Bustan et al.

% Utilization instructions:
%   1. Select desired conditions in the data below. One can change
%   parameters, select time and severity of failures, simulation time, etc.
%   2. Run this scrip (includes the simulation inside it)
```

Figure 2.2: Example of instructions provided at the beginning of each ".m" file.

the maximum angular velocity w_{max} (coming from the spacecraft's requirements), and the maximum angular velocity of the desired frame $w_{d,max}$ (which is known because it comes from the imposed attitude to track), to define the maximum magnitude allowed to the angular velocity error as of equation (2.9).

$$w_{e,max} = w_{max} - w_{d,max} \tag{2.9}$$

The filter is of the form in equation (2.10)

$$T_0 \dot{\mathbf{w}}_v + \mathbf{w}_v = \alpha \mathbf{w}_v^0 \tag{2.10}$$

where T_0 is the filter's time constant, α is a safety scale value ($0 < \alpha \leq 1$), and $\mathbf{w}_v^0$ is the input of the filter, which is given by equation (2.11)

$$\mathbf{w}_v^0 = -w_{e,max} tanh(c\mathbf{q}_e) \tag{2.11}$$

with c being a constant parameter selected by the designer. It is noted that the filter's input $\mathbf{w}_v^0$ converges to zero when the attitude error is zero, and it saturates approaching $-w_{e,max}$ for large positive values of $\mathbf{q}_e$, and $w_{e,max}$ for large negative values of $\mathbf{q}_e$. This filter guarantees that $\|\mathbf{w}_v\| \leq \alpha w_{e,max}$.

For the construction of the controller, the filter's output is used to define the virtual tracking error $\mathbf{w}_a$ in equation (2.12).

$$\mathbf{w}_a = \mathbf{w}_e - \mathbf{w}_v \tag{2.12}$$

The operator $\psi(\cdot)$ is defined in equation (2.13).

$$\psi(\cdot) = \|\mathbf{w}\|^2 + \|\mathbf{w}\| + 1 \tag{2.13}$$

The adaptive scalar parameter $\hat{b}$ update law is proposed as equation (2.14)

$$\dot{\hat{b}} = -\sigma \varrho \hat{b} + \frac{\sigma \psi^2(\cdot) \|\mathbf{w}_a\|^2}{\psi(\cdot) \|\mathbf{w}_a\| + \iota} \tag{2.14}$$

where $\hat{b}$ is the estimator of an unknown constant b used in the source's mathematical analysis, σ and ϱ are positive design parameters chosen by the designer, $\iota > 0$ is a small constant to avoid singularity, and the initial value $\hat{b}(0) > 0$.

Finally, the controller is proposed as equation (2.15)

$$\mathbf{u}_c = -\left(k + \frac{\hat{b}\psi^2(\cdot)}{\psi(\cdot)\left\|\mathbf{w}_a\right\| + \iota}\right)\mathbf{w}_a \tag{2.15}$$

where k is a positive constant design parameter, and $\mathbf{u}_c$ is the 3×1 commanded control torque vector, which will later be transformed to the real control torque $\mathbf{u}$ transmitted to the spacecraft via the actuators. The authors provide a summary of the effects of modifying the parameters in the overall high level behaviour of the controller, and also detail the parameter values used to obtain their results, which is going to be very valuable to replicate them in subsection 2.2.1.

2.2.1 Replication of results for 2018 Qiang Shen *et al.*

In the source [18], the technique is tested onto a spacecraft with inertia matrix

$$\mathbf{I} = \begin{bmatrix} 20 & 1.2 & 0.9 \\ 1.2 & 17 & 1.4 \\ 0.9 & 1.4 & 15 \end{bmatrix}$$

a perturbation model

$$\mathbf{d} = 0.001 \times [sin0.8t \quad cos0.5t \quad sin0.2t]^T$$

and a desired attitude to track

$$\mathbf{q}_d(t) = [\frac{1}{\sqrt{30}}sin{-}0.1t, \quad \frac{1}{\sqrt{60}}sin{-}0.2t, \quad 0.1cos{-}0.1t]^T$$

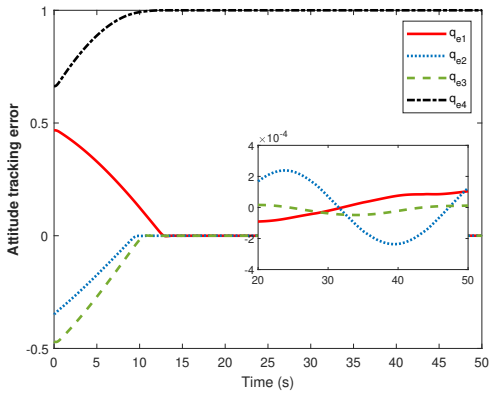
that corresponds to a value of $w_{d,max} = 0.055$ rad/s. The limitation on angular velocity is $w_{max} = 0.155$ rad/s, so due to equation (2.9), $w_{e,max} = 0.1$ rad/s.

The controller parameters used are reported as $k = 100$, $\alpha = 0.92$, $T_0 = 0.005$, $c = 80$, $\varrho = 10$, $\sigma = 0.1$, and $\iota = 0.005$. It is applied a hard limit of $4N$ in the control torque in each axis.

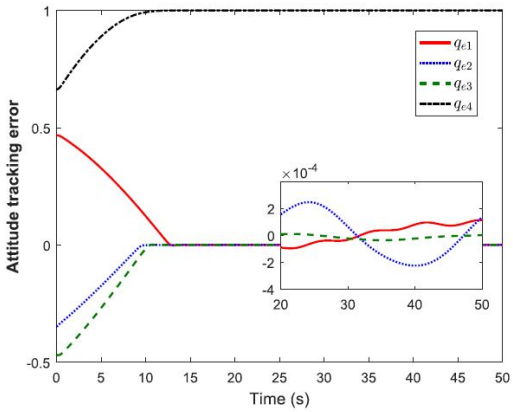
Initial attitude is given by the quaternion $\mathbf{Q}(0) = [0.7071, 0.5, -0.3, -0.4]^T$ (despite the source stating the first vectorial component with a negative sign, that is a mistake, and later in simulations this is the correct initial value), initial angular velocity is zero $\mathbf{w}(0) = [0, 0, 0]^T$ and initial adaptive parameter value is $\hat{b}(0) = 0.1$.

Absence of actuator faults. Results for the error quaternion when actuators do not have faults are shown in Figure 2.3, where the stile of the original source in the Figures has been imitated to ease comparison. That is why, although it is not recommended, the four quaternion components are shown in the same image. In this case notation is slightly different from the one introduced in section 2.1, note the change in quaternion numeration $q_{e4} = q_{0_e}$. In Subfigure 2.3a it is represented the results of the simulation created in this Thesis work, which can be compared to the equivalent results provided by the source, shown in 2.3b. As can be seen, results are almost identical, with the same magnitude and behaviour of the steady state error.

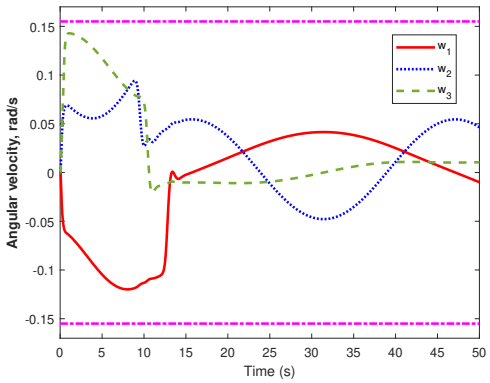
In 2.3c are shown the components of the angular velocity for the implemented simulation, and in 2.3d it is shown the same graph presented by the source. Again, results are absolutely similar.



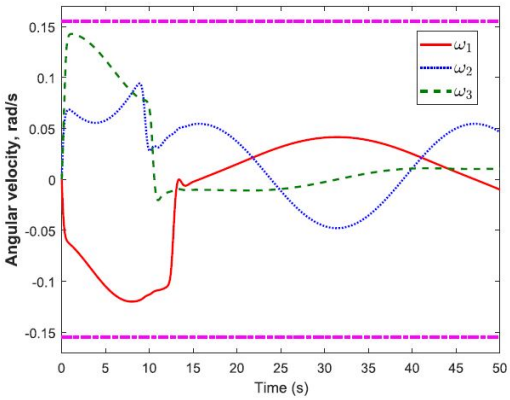
(a) Attitude error, simulation result



(b) Attitude error, original (from [18])



(c) Angular velocity, simulation result



(d) Angular velocity, original (from [18])

Figure 2.3: Error quaternions (2.3a)(2.3b) and angular velocities (2.3c)(2.3d) without actuator failures.

Presence of actuator failures. The source tests the case of actuator failures with the following three-axis loss of effectiveness fault:

$$e_1(t) = \begin{cases} 1, & \text{if } t < 5 \\ 0.5 + 0.09sin0.05t + 0.005rand(\cdot) & \text{if } t \geq 5 \end{cases}$$
$$e_2(t) = \begin{cases} 1, & \text{if } t < 5 \\ 0.6 + 0.10cos0.08t + 0.008rand(\cdot) & \text{if } t \geq 5 \end{cases}$$
$$e_3(t) = \begin{cases} 1, & \text{if } t < 5 \\ 0.4 + 0.08sin0.06t + 0.005rand(\cdot) & \text{if } t \geq 5 \end{cases}$$

where $rand(\cdot)$ represents a random value generated from the standard normal distribution with mean 0 and standard deviation 1. As the update rate of this $rand(\cdot)$ function is not indicated in [18], for the simulation it has been selected as 0.05 seconds. Also, the three-axis additive bias fault is added:

$$f_{a1}(t) = \begin{cases} 0, & \text{if } t < 10 \\ 0.75 + 0.25sin0.04t & \text{if } t \geq 10 \end{cases}$$
$$f_{a2}(t) = \begin{cases} 0, & \text{if } t < 10 \\ 0.95 + 0.05sin0.08t & \text{if } t \geq 10 \end{cases}$$
$$f_{a3}(t) = \begin{cases} 0, & \text{if } t < 10 \\ 0.85 + 0.15sin0.06t & \text{if } t \geq 10 \end{cases}$$

where all magnitudes are in Newton $[N]$.

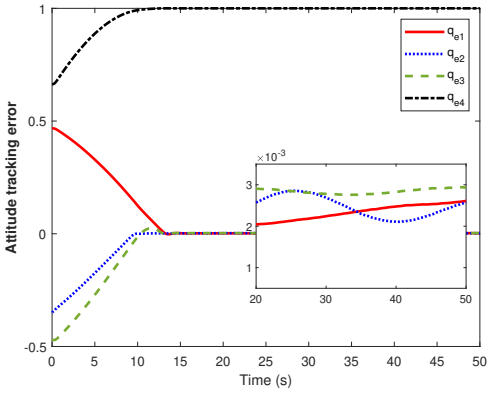
The same results as before are presented in Figure 2.4. Comparison of 2.4a and 2.4b demonstrate the same behaviour between the custom implementation and the original source, with only small differences in the steady state, probably due to a different update rate of the $rand(\cdot)$ function chosen.

Angular velocity plots 2.4c and 2.4d also show complete equivalence between results obtained and the source.

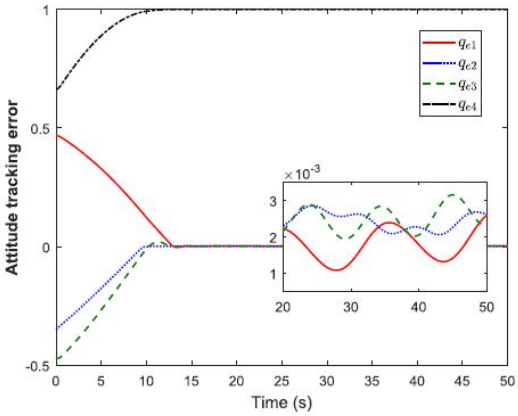
2.3 2014 Danyal Bustan *et al.*

The method [17] makes use of two auxiliary variables defined in equations (2.16) and (2.17).

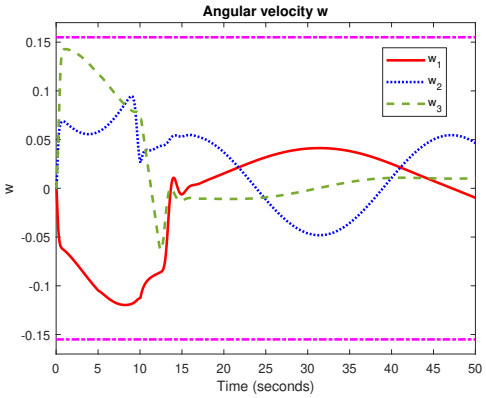
$$\mathbf{s}_1 = \mathbf{w}_e + (k^2 + 1)\mathbf{q}_e \tag{2.16}$$
$$\mathbf{s}_2 = (k^2 + 1)\rho q_{0_e}\dot{\mathbf{q}}_e \tag{2.17}$$



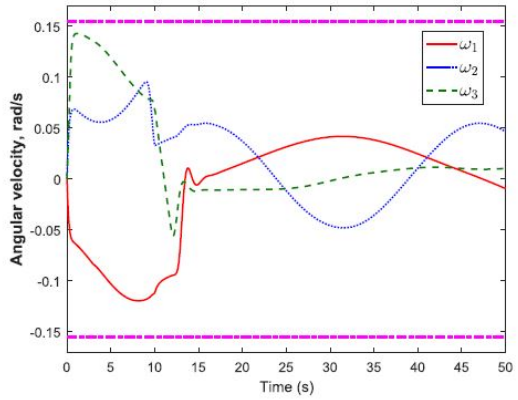
(a) Attitude error, simulation result



(b) Attitude error, original (from [18])



(c) Angular velocity, simulation result



(d) Angular velocity, original (from [18])

Figure 2.4: Error quaternions (2.4a)(2.4b) and angular velocities (2.4c)(2.4d) in presence of actuator failures.

where ρ is the positive constant that is claimed to control the transient response of the system, and $k(t)$ is the time-varying adaptive parameter used in the technique. Its adaptive update law is defined in equation (2.18)

$$\dot{k} = \begin{cases} \frac{-\gamma\chi}{k(\mathbf{w}_e^T \mathbf{I}\mathbf{w}_e + 2\gamma(1-q_{0_e}))} & \text{if } \mathbf{w}_e \neq 0 \\ 0 & \text{if } \mathbf{w}_e = 0 \end{cases} \tag{2.18}$$

where γ is a positive constant chosen by the designer, and χ is given by equation (2.19)

$$\begin{aligned} \chi = &\sum_{i=1}^{3} \left(\frac{k^2 \gamma_{min} u_{max}}{2\gamma} \frac{|w_{ie}|(k^2+1)(\delta+1)}{|w_{ie}| + (k^2+1)(\delta+1)} \right) \\ &+ \sum_{i=1}^{3} \left(\frac{k^2 \gamma_{max} u_{max}}{2\gamma} \frac{|w_{ie}||q_{ie}|(k^2+1)}{|s_{1_i}| + (k^2+1)\delta} \right) + \frac{k^2}{2} \mathbf{q}_e^T \mathbf{s}_1 \end{aligned} \tag{2.19}$$

where the subindex i denotes the corresponding vector i-th component, δ is a positive control constant, γ_{max} and γ_{min} represent respectively the maximum and minimum effectiveness of the combined actuators in any of the three body axis, and u_{max} is the lowest of the control torque limits provided by the actuators on the three body axis. γ_{max} and γ_{min} must be known, but they can be selected very aggressively (for instance as 0.1 and 1 respectively) if information on failure severity is not available.

Finally, the controller is proposed as equation (2.20)

$$u_i = -\frac{u_{max}}{2} \times \left(\frac{s_{1_i}}{|s_{1_i}| + (k^2+1)\delta)} + \frac{s_{2_i}}{||\mathbf{s}_2|| + (k^2+1)\delta)} \right) \tag{2.20}$$

where $||\cdot||$ denotes the euclidean norm of a vector or matrix.

Looking more carefully at equation (2.18), it can be seen that this is a first order differential equation with no forcing term. In fact, it is acknowledged by the source that if γ and δ are selected so that $\gamma_{min} u_{max}(k^2+1) > \gamma(|\bar{s_{1_i}}| + (k^2+1)\delta$, where $|\bar{s_{1_i}}|$ is the bound value of $\mathbf{s}_1$, then k has an exponentially decreasing solution, tending to zero. This is puzzling for a method that claims to regulate a changing target tracking. That is, because if the controller is asked to perform a manoeuvre at one time, and later is asked the same manoeuvre at a later time, the response would not be the same because the value of k would have decreased in the mean time. Thus, doubts are seeded on up to what point this technique is an adaptive controller, which does not truly "adapt" to the dynamic attitude situation.

It was also observed during simulations that k tended quickly to zero up to machine precision. That is a problem because looking again at equation (2.18), it has a singularity at $k = 0$. However, in equation (2.18), χ is located at the

numerator, and looking at equation (2.19) all terms of Ξ are multiplied by k^2. One of these can be cancelled with the k in the denominator of (2.18). Empiric evidence shows that this change does not affect the technique's implementation.

2.3.1 Replication of results for 2014 Danyal Bustan *et al.*

In the source [17], the technique is tested onto a spacecraft with inertia matrix

$$\mathbf{I} = \begin{bmatrix} 20 & 0 & 0.9 \\ 0 & 17 & 0 \\ 0.9 & 0 & 15 \end{bmatrix} \quad \mathbf{kg\,m^2}$$

a perturbation model

$$\mathbf{d} = (|\mathbf{w}|^2 + 0.05)[sin0.8t, cos0.5t, cos0.3t]^T \quad \mathbf{N \cdot m}$$

and a desired attitude to track

$$\mathbf{Q}_d = \frac{1}{2} \begin{cases} [0, cos0.15t, -\sqrt{3}, sin0.15t]^T & t < 50 \\ [sin0.15t, 0, cos0.15t, -\sqrt{3}]^T & 50 \leq t < 100 \\ [-\sqrt{3}, sin0.15t, 0, cos0.15t]^T & 100 \leq t < 150 \\ [cos0.15t, -\sqrt{3}, sin0.15t, 0]^T & 150 \leq t < 200 \end{cases}$$

The controller parameters used are reported as $u_{max} = 5$, $\gamma = 0.002$, $\delta = 0.02$, $\gamma_{min} = 0.1$, and $\gamma_{max} = 0.1$. Strangely, [17] reports a value for parameter used during the mathematical derivation of the method as $J_M = 20.1571$, although this parameter is later never used in the actual implementation of the technique.

When failures are applied, they are set to a loss of effectiveness for the actuators given by:

$$e_i = \begin{cases} 1 & \text{if } f_i > 1 \\ 0.1 & \text{if } f_i < 1 \\ f_i & otherwise \end{cases}$$

where

$$f_i = 0.3 + 0.1(sin0.5t + i\pi/3 + rand(\cdot), \quad i = 1, 2, 3$$

Initial attitude is given by the quaternion $\mathbf{Q}(0) = [0.96, -0.1, 0.15, -0.2]^T$, initial angular velocity is zero $\mathbf{w}(0) = [0, 0, 0]^T$, and the initial adaptive parameter value is $k(0) = 2.5$.

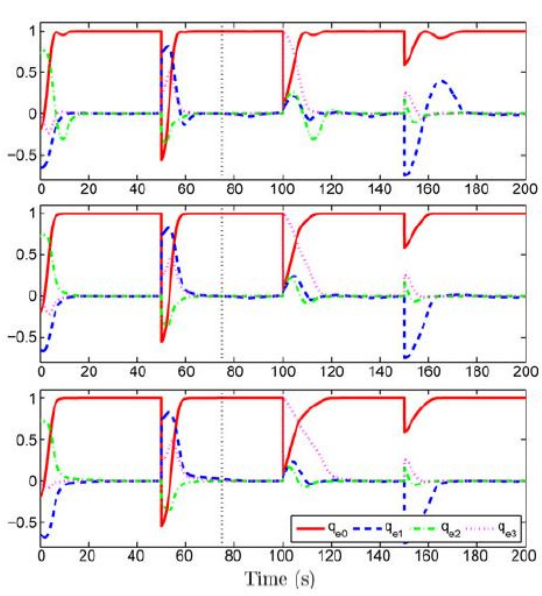
Results in [17] are reported for three values of the transient parameter, $\rho = 15$, $\rho = 1$, and $\rho = 0$, and this is what is going to be imitated here, imitating the stile of the figures in the source as much as possible.

Figure 2.5a shows the components of the attitude error quaternion obtained in [17]. They can be compared with the ones obtained by the implemented simulation

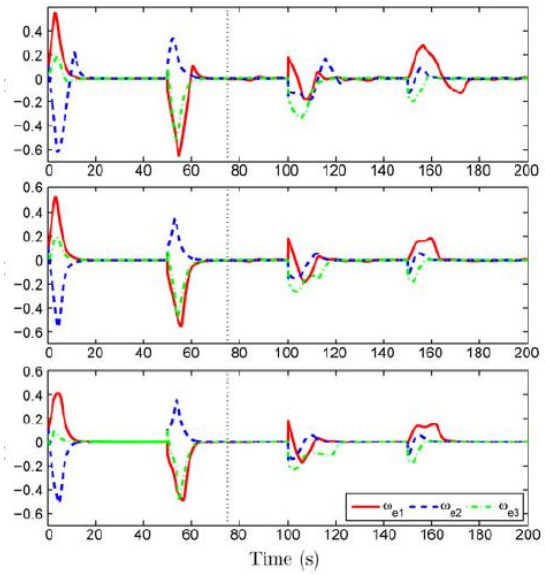
shown in Figure 2.6. Results are identical for the case with $\rho = 0$, and substantial overshooting can be observed on the manoeuvres.

However, a difference starts to be observed for the case of $\rho = 1$, where the response in the simulation is more affected by the increase in ρ than what is reported in the source. Figure 2.5a still has small overshoot, in contrast to Figure 2.6b where overshoot is all but entirely eliminated, and response is slower to converge.

This phenomenon continues and amplifies in Figure 2.6d compared to the bottom plot of Figure 2.5a. Note that in Figure 2.6d, q_{e_0} does not reach unity. The behaviour is like for some reason the implemented simulation was the same than the source but with the parameter ρ multiplied by a factor. Implementation has been reviewed and no failure has been spotted that could cause this effect.



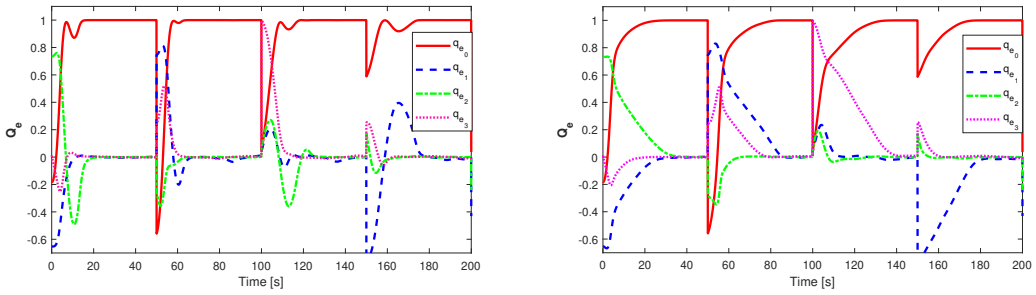
(a) Attitude errors $\mathbf{Q}_e$, for $\rho = 0$ (top), $\rho = 1$ (middle), and $\rho = 15$ (bottom)



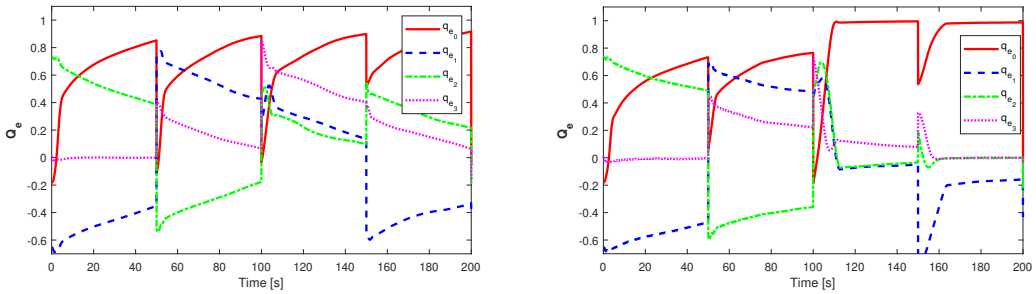
(b) Angular velocity errors $\mathbf{w}_e$ in rad/s, for $\rho = 0$ (top), $\rho = 1$ (middle), and $\rho = 15$ (bottom)

Figure 2.5: Source results from 2014 Danyal Bustan *et al.*

Plottings of the angular velocity errors w_e in Figures 2.5b and 2.7 reinforce this analysis, with results for $\rho = 0$ being equal. The shattering effect on the controller that [17] mentions occurring for high values of ρ, starts happening in the implemented simulation for $\rho = 15$, as can be seen in Figure 2.7d.

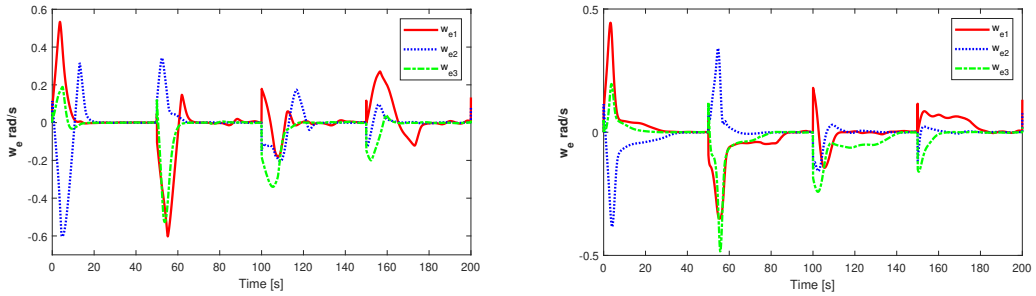


(a) $\rho = 0$ (b) $\rho = 1$

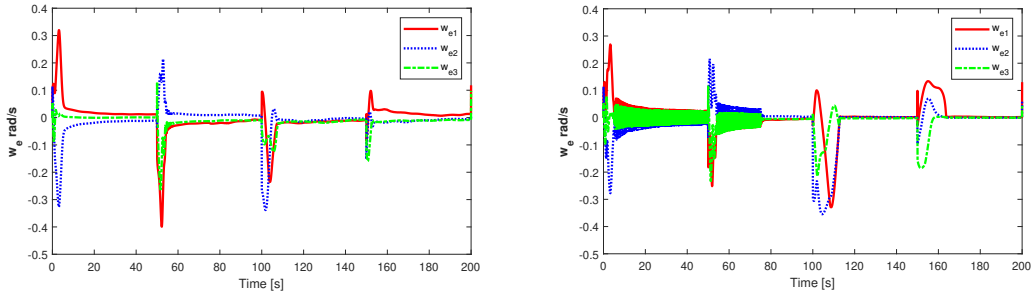


(c) $\rho = 5$ (d) $\rho = 15$

Figure 2.6: Attitude errors $\mathbf{Q}_e$ from implemented simulation



(a) $\rho = 0$ (b) $\rho = 1$



(c) $\rho = 5$ (d) $\rho = 15$

Figure 2.7: Angular velocity errors $\mathbf{w}_e$ from replicated simulation

2.4 2008 Wenchuan Cai *et al.*

To begin with, [23] defines a sliding mode-type variable $\mathbf{s}$ that in the source is called a "filtered error variable"

$$\mathbf{s} = \mathbf{w}_e + \beta \mathbf{q}_e \tag{2.21}$$

where β is a positive free design scalar parameter. It is also defined the same operator Φ than what was called ψ in section 2.2.

$$\Phi = \|\mathbf{w}\|^2 + \|\mathbf{w}\| + 1 \tag{2.22}$$

As mentioned in Chapter 1, the method in [23] is developed in three steps of increasing complexity. The first step is for unmodelled actuators, its output is the three-axis control torque τ, and the proposed controller is the one in equation (2.23)

$$\begin{aligned} \tau &= -[k_0 + \kappa(t)]\mathbf{s}, \qquad \kappa(t) = \frac{\hat{b}\Phi}{||\mathbf{s}|| + \epsilon} \\ \dot{\hat{b}} &= -\sigma_1 \hat{b} + \sigma_2 \frac{||\mathbf{s}||^2 \Phi}{||\mathbf{s}|| + \epsilon}, \qquad \epsilon = \frac{\mu}{1 + \Phi} \end{aligned} \tag{2.23}$$

where $k_0 > 0$, $\mu > 0$, $\sigma_1 > 0$ and $\sigma_2 > 0$ are freely chosen design parameters.

The second iteration of the controller accounts for the modelling of actuators with a column-wise distribution matrix $\mathbf{D}$ of dimensions $3 \times n$, where n is the number of actuators. $\mathbf{F}$ is the nx1 vector that denotes each individual actuator's required torque. In this case, the control force $\mathbf{u}$ would be equal to the multiplication of $\mathbf{D}$ and $\mathbf{F}$, $\mathbf{u} = \mathbf{DF}$, and the only change to the controller would be that instead of τ, the output would be $\mathbf{F}$ as in equation (2.24).

$$\mathbf{F} = -[k_0 + \kappa(t)]\mathbf{D}^T \mathbf{s} \tag{2.24}$$

The final version of the controller, which is the one implemented in this Thesis, takes explicitly into account the limitation on maximum actuator control torque. This is made defining $F_{max} = min\{F^1_{max}, F^2_{max}, ..., F^n_{max}\}$, where F^i_{max} is the scalar maximum torque generated by the i-th actuator. The controller is then modified as in equation (2.25)

$$\mathbf{F} = -F_{max} \frac{\mathbf{D}^T}{||\mathbf{D}||} sat([k_0 + \kappa(t)]\mathbf{s}) \tag{2.25}$$

with

$$sat([k_0 + \kappa(t)]\mathbf{s}) = \begin{cases} \frac{\mathbf{s}}{||\mathbf{s}||} & \text{if } ||\mathbf{s}|| \geq F_{max}/(k_0 + \kappa) \\ \frac{[k_0 + \kappa(t)]\mathbf{s}}{F_{max}} & \text{if } ||\mathbf{s}|| \leq F_{max}/(k_0 + \kappa) \end{cases} \tag{2.26}$$

and it is remembered that $||\cdot||$ denotes the euclidean norm of the matrix.

2.4.1 Replication of results for 2008 Wenchuan Cai *et al.*

The technique is tested in [23] on a spacecraft with six actuators with corresponding distribution matrix

$$\mathbf{D} = \begin{bmatrix} 0.8 & -0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & -0.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.7 \end{bmatrix}$$

and an inertia matrix taken up by two parts $I = I_0 + I_u(t)$, where I_0 is a nominal constant part given by

$$\mathbf{I_0} = \begin{bmatrix} 20 & 0 & 0.9 \\ 0 & 17 & 0 \\ 0.9 & 0 & 15 \end{bmatrix}$$

and I_u is an unknown time-varying part that is reported in the source by Figure 2.8. Because this time-varying part is not quantitatively reported, only with

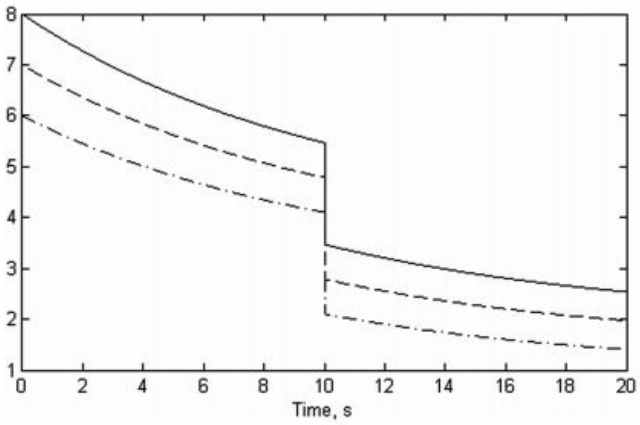


Figure 2.8: Unknown time-varying part of inertia matrix, I_u, in 2008 Wenchuan Cai *et al.*

Figure 2.8, it cannot be replicated in the implemented simulation, so an equivalent inertia matrix

$$\mathbf{I} = \begin{bmatrix} 26 & 0 & 0.9 \\ 0 & 22 & 0 \\ 0.9 & 0 & 19 \end{bmatrix}$$

has been used instead.

The desired attitude to track is given by

$$\mathbf{q}_d(t) = \left[\frac{\sqrt{3}}{3} sin{-}0.1t, \quad \frac{\sqrt{6}}{6} sin{-}0.1t, \quad \frac{1}{2} sin{-}0.1t \right]^T$$

and the disturbance torque is given by

$$\mathbf{d} = (||\mathbf{w}||^2 + 0.5)[sin0.8t, \quad cos0.5t, \quad cos0.3t]^T$$

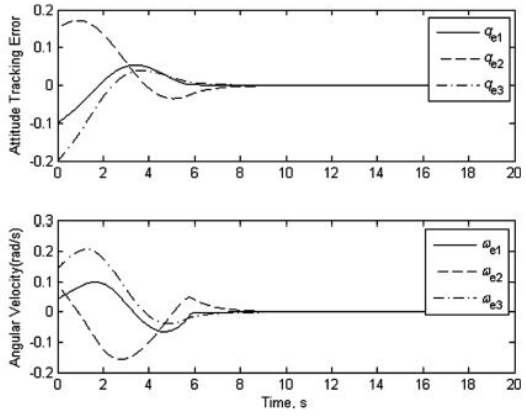
Actuator parameters used in [23] are reported as $k_0 = 20$, $\sigma_2 = 100$, $\sigma_1 = 0.01$, $\beta = 2$, $\mu = 0.1$, and $F_{max} = 5$. Initial attitude is given by $\mathbf{q}(0) = [-0.1, 0.15, -0.2]^T$, and the initial value of the adaptive parameter $\hat{b}(0) = 0$. The actuator failures applied are a loss of effectiveness of the form

$$\delta_i = 0.7 + 0.15rand(\cdot) + 0.1sin0.5t + i\pi/3. \qquad (i = 1, ..., 6)$$

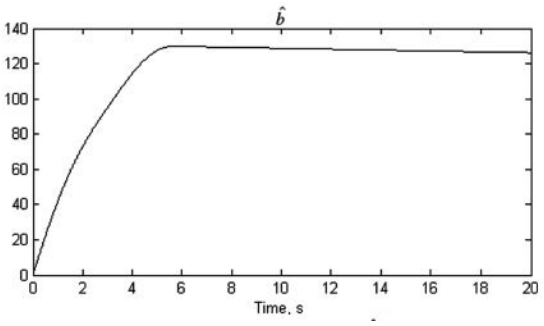
with the function $rand(\cdot)$ updating time being 2.4 seconds. Additionally, exactly as it happens in the source, at $t = 8$ the thruster 3 effectiveness is set permanently to 0.2, and thrusters 4 and 5 are shut down completely at times $t = 10$ and $t = 12$ respectively.

The results obtained in [23] are presented in Figure 2.9, while the results from the replication of the technique are presented, trying to imitate as much as possible the original style, in Figure 2.10. It can be seen from the comparisons of Figure 2.9a with Figures 2.10a and 2.10b that the results are absolutely analogous.

However, the behaviour of the adaptive parameter $\hat{\beta}$ is not, as reflected by Figures 2.9b and 2.10c. In the source it reaches values of 130 and then decreases very slowly, while in the replication it does not quite reach 80 and then decreases much faster. Looking at the adaptive update law in equation (2.23), it can be deduced why. The forcing term is directly proportional to the variable $\mathbf{s}$, which is a filtered combination of the errors. When the spacecraft approaches the target and the errors become close to zero, $\mathbf{s}$ also becomes close to zero, and the forcing term disappears. This leaves only the exponentially decreasing term to affect $\hat{\beta}$. It is observed for this technique that after a longer simulation time the value of $\hat{\beta}$ reaches an equilibrium where it oscillates (around 15 for this simulation set up), and the steady state errors remain bounded by $q_{e_i} < 0.01$ for $i = 1, 2, 3$.

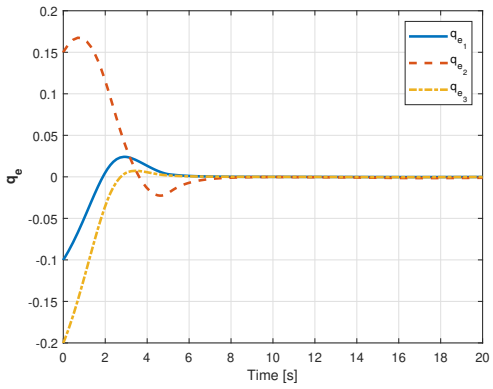


(a) Angular velocity error and attitude tracking error

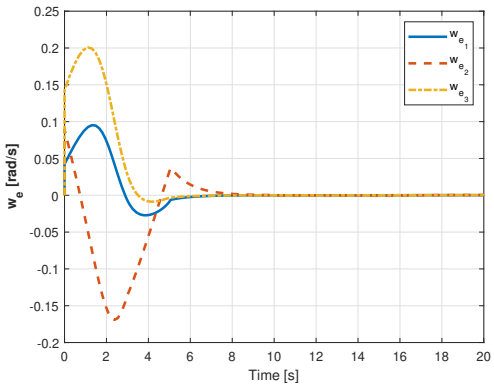


(b) Adaptive parameter $\hat{\beta}$

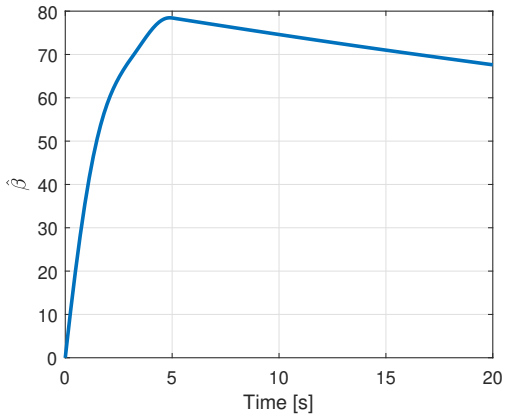
Figure 2.9: Source results from 2008 Wenchuan Cai *et al.*



(a) Attitude tracking error



(b) Angular velocity error



(c) Adaptive parameter $\hat{\beta}$

Figure 2.10: Simulation results from the implemented technique

2.5 2011 Qinglei Hu *et al.*

The main instrument of the technique develop in [22] is the addition of an integral term to the classical sliding surface variable, shown in (2.27). Note that in this case the sliding surface is created using the unaltered attitude quaternion and its derivative, instead of the error equivalents, because the source only addresses the pointing problem to align with the inertial frame.

$$\mathbf{S} = \dot{\mathbf{q}} + \mathbf{K}_P\mathbf{q} + \mathbf{K}_I \int_0^t \mathbf{q} d\tau \tag{2.27}$$

A new variable $\dot{\mathbf{q}}_r$ is introduced as

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}} - \mathbf{S} = -\mathbf{K}_P\mathbf{q} - \mathbf{K}_I \int_0^t \mathbf{q} d\tau \tag{2.28}$$

Then, the following definitions are made; $\mathbf{P} \equiv \mathbf{\Xi}^{-1}$ with $\mathbf{\Xi} \equiv \mathbf{q}^{\times} + q_0 I_{3\times 3}$. The constants a_0 and a_1 are introduced as unknowns that define the upper bounds of the inertia matrix and the cosines rotation matrix respectively. See the source [22] for more detail. Similarly, a_2 and a_3 are unknown constants that bound the disturbance torque. Finally, $\mathbf{D}$ in this formulation is the $3\times n$ actuator distribution matrix, with n being the number of actuators present, and $\mathbf{F}_f$ being the nx1 vector denoting the values of actuation at which the actuators get stuck when a fault of this type occurs. For instance, in the case of outage-only in the ith actuator, F_{f_i} would be equal to zero.

With this concepts, another four additional variables are introduced as

$$\left.\begin{aligned} \mathbf{Y}_1 &= [\quad ||\ddot{\mathbf{q}}_r - \lambda\mathbf{S}|| \quad , \quad ||\dot{\mathbf{q}}_r||||\dot{\mathbf{q}}|| \quad]^T \\ \mathbf{\Theta}_1 &= [\quad a_0 \quad , \quad a_1]^T \end{aligned}\right\} \tag{2.29}$$

$$\left.\begin{aligned} \mathbf{Y}_2 &= \tfrac{||\mathbf{P}||}{2}[\quad 1 + ||\mathbf{D}|| \quad , \quad ||\dot{\mathbf{q}}||^2 \quad]^T \\ \mathbf{\Theta}_1 &= [\quad a_2 + ||\mathbf{F}_f|| \quad , \quad a_3]^T \end{aligned}\right\} \tag{2.30}$$

where $\lambda > 0$ is a constant chosen to specify the speed of convergence of the system.

The controller proposed is in equation (2.31)

$$\mathbf{u} = -2\mathbf{D}^T\mathbf{\Xi}(\mathbf{KS} + \hat{\alpha}_1||\mathbf{Y}_1||sgn(\mathbf{S}) + \hat{\alpha}_2||\mathbf{Y}_2||sgn(\mathbf{S})) \tag{2.31}$$

with adaptation laws in equations (2.32) and (2.33)

$$\left.\begin{aligned} \dot{\hat{\alpha}}_1 &= -\beta_1^2\hat{\alpha}_1 + \gamma_1||\mathbf{Y}_1||||\mathbf{S}|| \\ \dot{\beta}_1 &= -K_{\beta_1}\beta_1 \end{aligned}\right\} \tag{2.32}$$

$$\left.\begin{aligned} \dot{\hat{\alpha}}_2 &= -\beta_2^2\hat{\alpha}_2 + \gamma_2||\mathbf{Y}_2||||\mathbf{S}|| \\ \dot{\beta}_2 &= -K_{\beta_2}\beta_2 \end{aligned}\right\} \tag{2.33}$$

where $\mathbf{K}$ is a positive definite matrix chosen by the designer, γ_i and K_{β_i} $(i = 1, 2)$ are arbitrary positive constants, and $\hat{\alpha}_i$ is the parameter estimation of a value α_i which satisfies $\alpha_i \geq ||\mathbf{\Theta_i}||$ for $i = 1, 2$. The sign function satisfies

$$sgn(\upsilon) = \begin{cases} 1 & \text{if } \upsilon > \epsilon \\ \upsilon & \text{if } |\upsilon| \leq \epsilon \\ -1 & \text{if } \upsilon < \epsilon \end{cases} \tag{2.34}$$

where ϵ is a small constant chosen to avoid the chattering effect arising from the imperfect implementation of a typical sign function.

Immediately, looking at the adaptation laws in equations (2.32) and (2.33), it is shocking that the β_i parameters have no forcing term inside their equations. It is similar to what happened for equation (2.18), however in this case the effect is different, and at first glance it could be argued that greater. The parameters β_i will, no matter their starting value, fade to zero, and when that happens, the parameters α_i lose their decreasing term that counteracts the forcing one!

It seems that the source trusts that an initial transient would make a perfect estimation of the values α_i using $\hat{\alpha}_i$. But careful examination would suggest that this is not possible. The presence of the absolute value operators guarantees the second term on the adaptation equations for $\dot{\hat{\alpha}}_i$ will always be positive. Thus, $\hat{\alpha}_i$ will forever increase, making the whole scheme unstable. On top of that, the source [22] makes a crucial mistake for the repeatability of their results that has not been noticed until now. It does not report the initial values of these adaptation parameters. The consequences of this lack of reporting will be seen soon.

2.5.1 Replication of results for 2011 Qinglei Hu *et al.*

To show the capabilities of the technique, in [22] it has been applied to a spacecraft with inertia matrix

$$\mathbf{I_0} = \begin{bmatrix} 1543.9 & -2.3 & -2.86 \\ -2.3 & 471.6 & -35 \\ -2.86 & -35 & 1713.3 \end{bmatrix}$$

disturbance of the form

$$\mathbf{d} = [3cos0.01t + 1, \quad 5sin0.02t + 3cos0.025t + 2, \quad 3sin0.01t + 3] \times 10^{-3} \quad \mathbf{N \cdot m}$$

maximum actuator force of 1 Newton, and distribution matrix

$$\mathbf{D} = \begin{bmatrix} -d & -c & c \\ -d & c & c \\ -d & c & -c \\ -d & -c & -c \end{bmatrix} \times \begin{bmatrix} -cos\vartheta & \rho sin\vartheta & \rho sin\vartheta \\ -cos\vartheta & -\rho sin\vartheta & \rho sin\vartheta \\ -cos\vartheta & -\rho sin\vartheta & -\rho sin\vartheta \\ -cos\vartheta & \rho sin\vartheta & -\rho sin\vartheta \end{bmatrix}$$

where $\rho = 1/\sqrt{2}$, $\vartheta = 5°$, $d = 0.5$, $c = 0.2$, and the operator $\times$ denotes the vectorial product of the ith row of the first column with the same ith row of the second column. This is a very complex distribution matrix, where it has been deduced that the result of the vectorial product of the ith rows shall be placed on the ith column of matrix $\mathbf{D}$ (remember that $\mathbf{D}$ is of dimensions $3 \times n$, so in this case 3×4). However, as this was not completely clear, other arrangements of constructing matrix $\mathbf{D}$ with the sources data have been tried, like placing the ith result in rows instead of columns, and also a substitution with the simple identity matrix $I_{3\times3}$.

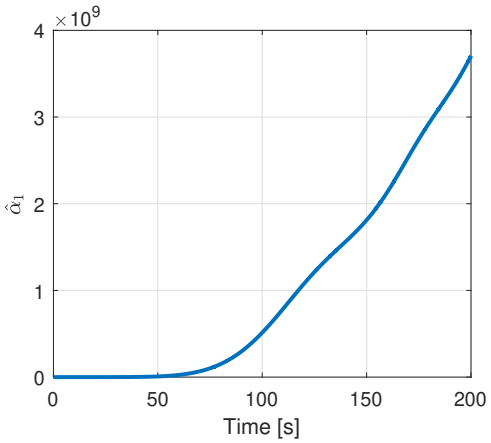
Controller parameters are reported are $\mathbf{K} = 1000I_{3\times3}$, $\mathbf{K}_P = [10]_{3\times3}$, $\mathbf{K}_I = [10]_{3\times3}$, $\gamma_1 = \gamma_2 = 10$, $K_{\beta_1} = K_{\beta_2} = 50$, $\epsilon = 0.001$.

Initial attitude is taken as $Q(0) = [0.173648, -0.263201, 0.786030, -0.526402]$ and initial angular velocity is zero. Crucially, the initial adaptive parameter values are not reported, nor is λ, and although in Fig.3 of the source [22] it can be observed that $\hat{\alpha}_1(0) = \hat{\alpha}_2(0) = 0$, the values for $\beta_1(0)$, $\beta_2(0)$ and λ are completely unknowable. Thus, a wide range of combinations have been tried for them, varying $\beta_1(0)$ and $\beta_2(0)$ independently from 0.01 to 10^4, and λ from 0.001 to 100.

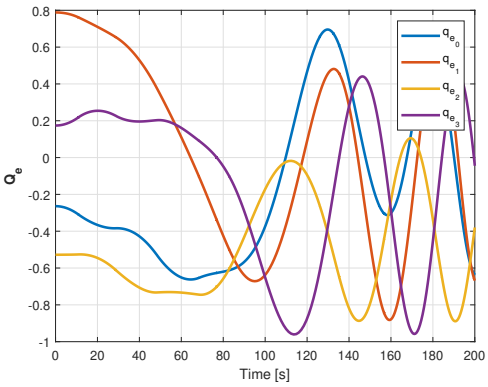
Despite all of these different combinations, no case has been found that results in a satisfactory solution for this technique. Firstly, it was tried all the combinations of $\beta_1(0)$, $\beta_2(0)$ and λ that have been mentioned before. Later, it was changed the matrix $\mathbf{D}$ to the simple 3×3 identity and to the other matrices $\mathbf{D}$ mentioned, fearing that an error in the implementation was the cause of the bad results. After that, a thorough review of the implementation was attempted, trying to identify any possible mistakes. Lastly, when that did not have any result, extensive changes on the technique were attempted, trying to spot any mistake the source may have done when reporting the technique.

None of these steps gave positive results. The simulations of this method always show the same behaviour. As predicted in section 2.5, the parameters $\hat{\alpha}_i$ increase indefinitely with nothing opposing their growth, as it can be seen in Figure 2.11a. The system becomes unstable and starts to oscillate uncontrollably in a short period of time, as shown in 2.11b. This happens in every case, despite the numerous attempts to find a successful solution.

It is a sad state of affairs that no satisfactory implementation of this technique was possible, because the method claimed to have very interesting results for the actuator stuck case, controlling even at full stuck actuation. In fact, this was the main reason why this technique was selected for implementation.



(a) $\hat{\alpha}_1$



(b) Attitude quaternion Q

Figure 2.11: Results of simulations with 2011 Qinglei Hu *et al.*

2.6 Chapter conclusions

The results of replicating the techniques in the sources selected have been presented, and the main conclusions that can be drawn from them are:

- Three out of the four techniques have been implemented with satisfactory outcome, those been 2018 Qiang Shen *et al.* [18], 2014 Danyal Bustan *et al.* [17] and 2008 Wenchuan Cai *et al.* [23].
- The replication of 2018 Qiang Shen *et al.* [18] has produced the exact same results as the ones presented by the source in every case for all variables.
- For 2014 Danyal Bustan *et al.*, the case of $\rho = 0$ produces the exact same results as reported. For cases of $\rho > 0$ the overall behaviour is the same but the effect of ρ are bigger in the replication than reported in the source, so effects appearing with large values on the source appeared with smaller values in the implementation. The technique worked despite lack of forcing term in the adaptation law.
- In the case of 2008 Wenchuan Cai *et al.* [23], the reports of attitude and angular velocity on the source are exactly the same as the ones obtained with its replication. However, the adaptive parameter $\hat{\beta}$ behaves differently, not reaching such high values and decreasing faster, although this discrepancy seems to not have any effect on the rest of the results. An explanation for this behaviour is proposed in section 2.4.
- Sadly for 2011 Qinglei Hu *et al.* [22] no successful implementation has been achieved. Multiple reviews, combinations of set up parameters, and modifications where made with no positive results. In all simulations the dynamics

become unstable and an uncontrolled oscillation develops in a short period of time. A hypothesis on why this happens, based on the technique's equations structure, is presented in section 2.5. It is related to the lack of counter-balance term in the adaptation laws that makes unavoidable an indefinite growth of the adaptive parameters.

In the next chapter, the three techniques that produced positive results are going to be tested onto the first baseline of the project, the LUMIO CubeSat.

Chapter 3

First baseline: the LUMIO CubeSat

This chapter is dedicated to testing the three adaptive methods selected and successfully implemented in Chapter 2 into the LUMIO CubeSat mission baseline. Three scenarios of the spacecraft's life-cycle have been carried out: slew manoeuvre, detumbling and long-term pointing. The chapter begins by explaining the baseline, and then has one section per scenario where each method, together with the original controller from the source, are tested. Finally, the discoveries obtained are summarized in the conclusions section.

3.1 Outline of the baseline

The LUMIO (Lunar Meteoroid Impact Observer) mission is a 12U CubeSat put forward by a consortium that includes the Politecnico di Milano, and is one of the winners of the ESA's General Studies Program SysNova contest to design CubeSat missions to the Moon. Its aim is to orbit the Moon and observe the meteoroid impacts landing on its dark side.

Provided by the Thesis mentor, the document Phase A System Design Report [1] details the design of the ACS subsystem and its requirements. For purpose of confidentiality, only the details required for the implementation of the baseline are going to be reported here, not mentioning for instance the sensor and actuator brands chosen, or any other data that is not strictly necessary.

To begin with, the spacecraft's inertia matrix is reported for the packed configuration in the detumbling phase as

$$\mathbf{I}_{pack} = \begin{bmatrix} 0.1701 & 0 & 0 \\ 0 & 0.2200 & 0 \\ 0 & 0 & 0.2310 \end{bmatrix} kg\, m^2 \tag{3.1}$$

and for the deployed configuration in the rest of the mission phases as

$$\mathbf{I}_{depl} = \begin{bmatrix} 0.4029 & 0 & 0 \\ 0 & 0.2204 & 0 \\ 0 & 0 & 0.4516 \end{bmatrix} kg\, m^2 \tag{3.2}$$

The spacecraft uses as actuators four RWs and a Reaction Control System (RCS), comprised of thrusters, to desaturate the RWs. The only relevant data of the actuators that is needed to be known are the maximum actuation torque of the RWs, $\dot{h}_{rmax}$

$$\dot{h}_{rmax} = 2 \times 10^{-3} \quad [N\, m] \tag{3.3}$$

the RWs maximum angular momentum storage h_{rmax}

$$h_{rmax} = 30 \times 10^{-3} \quad [N\, m\, s] \tag{3.4}$$

and the RWs distribution matrix $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} -1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} \\ -1/\sqrt{3} & -1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} \tag{3.5}$$

As of sensors, the spacecraft counts with six fine Sun sensors, one Inertial Measurement Unit (IMU), and two star trackers. Because the start trackers are the most precise sensors and no information is given on how the data measurement fusion is carried out, it is going to be assumed that the measurements are obtained only by them. In any case, this would be an upper bound to the errors, because other sensors are being discarded that could improve precision.

The individual sensor accuracy ϑ_{acc} is reported as $9\, arcseconds$ for the yaw axis, that corresponds to the Y axis, and $51\, arcseconds$ for the roll and pitch axis, that correspond to the X and Z axis respectively. Taking into account that there are two of them, the overall measurement accuracy is given by

$$\vartheta_{acc} = [51, \quad 9, \quad 51]/\sqrt{2} \quad arcseconds \tag{3.6}$$

The data on angular velocity is surely obtained by a fusion of the IMU measurements and a derivation of the position measurements, but no reference to this is mentioned in [1]. As no information is provided, it is assumed that angular velocity measurements are only provided by the IMU, which offers an accuracy of $13\, arcseconds/s$ on all three axis. The form of implementing these sensors has been simple. At each time step, with an update rate of 0.05 seconds, a random value is selected inside the uncertainty region using a random number generator with equal probability for the entire number range.

The relevant requirements imposed on the ACS for the simulations implemented are:

1) Detumbling maximum starting angular velocity value $w_{i,max}(0) = 10\,deg/s$.

2) Detumbling maximum final angular velocity value $w_{i,max}(t_{end}) = 0.5\,deg/s$.

3) Steady state maximum absolute attitude tracking error of $\theta_{i,max} = 0.18\,deg$ half cone for the science phase.

4) Steady state maximum angular velocity error of $w_{i,max} = 64.9\,arcsec/s$ for the science phase.

5) Maximum angular velocity during slew manoeuvres of $w_{i,max} = 0.5\,deg/s$.

The failures that the system is going to be tested against have been selected to be, all of them, complete failures of one or more RWs, from the beginning of the simulation.

Three scenarios have been selected to test the techniques on during the life-cycle of the spacecraft: a slew manoeuvre, the detumbling phase, and the long-term pointing during the scientific phase. Besides the selected techniques, the ACS design in [1] proposes the following control laws for each of the three mentioned scenarios.

For the detumbling phase, the ideal control torque is computed as in equation (3.7)

$$\mathbf{u}_{id} = -0.01\,\mathbf{I}_{pack}\mathbf{w} \tag{3.7}$$

and the mapping from the desired control torque to the real RWs angular momentum derivative $\dot{\mathbf{h}}_r$ is done as shown in equation (3.9)

$$\dot{\mathbf{h}}_r = -\mathbf{R}^{\perp}(\mathbf{u}_{id} + \mathbf{w} \times \mathbf{R}\mathbf{h}_r) \tag{3.8}$$

where the $\times$ operator denotes the vectorial product and $\mathbf{R}^{\perp}$ is the distribution matrix pseudoinverse.

$$\mathbf{R}^{\perp} = \mathbf{R}^T(\mathbf{R}\mathbf{R}^T)^{-1} \tag{3.9}$$

It is worth mentioning now, that for all the methods implemented that produce a virtual three-axis control torque, namely [18] and [17], the mapping onto the actuators has been done in the way of equations (3.8) and (3.9).

The control law during the science phase, that from now on is going to be called Science Control Mode (SCM), is given by equation (3.10)

$$\mathbf{u}_{id} = -10^{-2}\,\mathbf{I}_{pack}\mathbf{w_e} - 10^{-5}\mathbf{A}_e^{\vee} + \mathbf{w} \times \mathbf{I}\mathbf{w} + \mathbf{I}(\mathbf{A}_e\mathbf{w}_d - \mathbf{w}_e^{\times}\mathbf{A}_e\mathbf{w}_d) \tag{3.10}$$

where $\vee$ is the operator that does the inverse mapping than $\times$ in equation (2.5), and $\mathbf{A}_e$ is the attitude cosines matrix equivalent to the quaternion error, see $\mathbf{C}$ in equation (2.8).

The control law during the desaturation phases, from now on called Desaturation Control Mode (DCM), is very similar, only changing the coefficients to the ones shown in equation (3.11)

$$\mathbf{u}_{id} = -\mathbf{I}_{pack}\mathbf{w_e} - 10^{-2}\mathbf{A}_e^{\wedge} + \mathbf{w} \times \mathbf{I}\mathbf{w} + \mathbf{I}(\mathbf{A}_e\mathbf{w}_d - \mathbf{w}_e^{\times}\mathbf{A}_e\mathbf{w}_d) \tag{3.11}$$

As for disturbances, the original source [1] reports the external disturbance torque affecting the spacecraft in Figure 3.1, which is completely a result of the Solar Radiation Pressure (SRP), as no other disturbances are considered in the deep-space orbit of the mission.

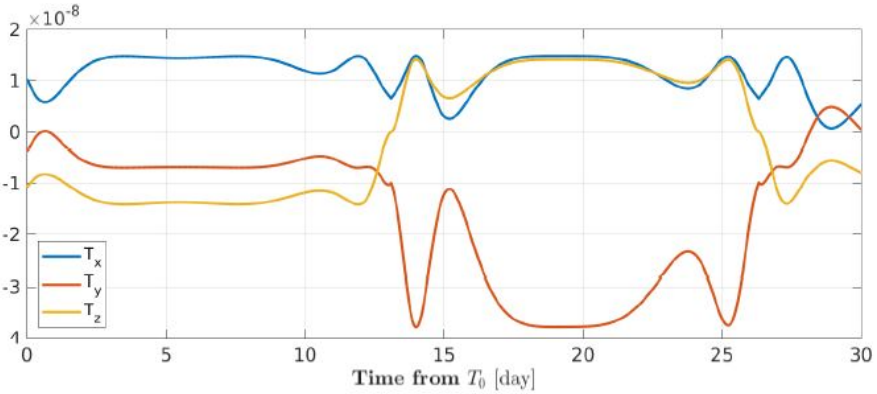


Figure 3.1: External disturbance torque acting on the CubeSat, from [1]

As it can be seen, variations on disturbance occur in the time frame of days. It would have been interesting to model the actual SRP torque given the attitude of the spacecraft, as the source [1] does provide the surfaces areas, positions and normals, which is all that is needed by a SRP model. However, no information is given on the orbit, so the Sun's direction cannot be computed. Thus, what has been done instead is implement the following disturbance model:

$$\mathbf{d} = 10^{-8} \times \begin{bmatrix} 0.8 + 0.7sin\frac{2\pi t}{3 \cdot 86400} \\ -1.9 - 1.9cos\frac{2\pi t}{3 \cdot 86400} \\ -sin\frac{2\pi t}{20 \cdot 86400} \end{bmatrix} \quad N\,m \tag{3.12}$$

The slew and detumbling manoeuvres take minutes at most, instead of days. This means for those scenarios the disturbance will essentially be constant $\mathbf{d} = [0.8,\ -0.38,\ 0] \times 10^{-8}$ Nm, which can be considered a worst-case scenario value from the one presented in Figure 3.1. Anyway, it has been observed that for the transient behaviour, such a small disturbance of order 10^{-8} is completely negligible, only really affecting the steady state of the system.

3.2 First scenario: Slew manoeuvre

For all controllers, the slew manoeuvre has been selected to be from an initial attitude represented by

$$\mathbf{q}(0) = [0.4, \quad -0.3, \quad -0.5]^T \tag{3.13}$$

with the target attitude being $\mathbf{q}(0) = [0, 0, 0]^T$. The target attitude has been selected as not time-varying because the manoeuvres take minutes, while a full orbit of the spacecraft around the Moon takes seven days, so the orbit angular rate is considered negligible.

3.2.1 Original controller

As it is unclear whether or not the slew manoeuvre would be performed using SCM or DCM, both versions of the control law will be tested in this subsection.

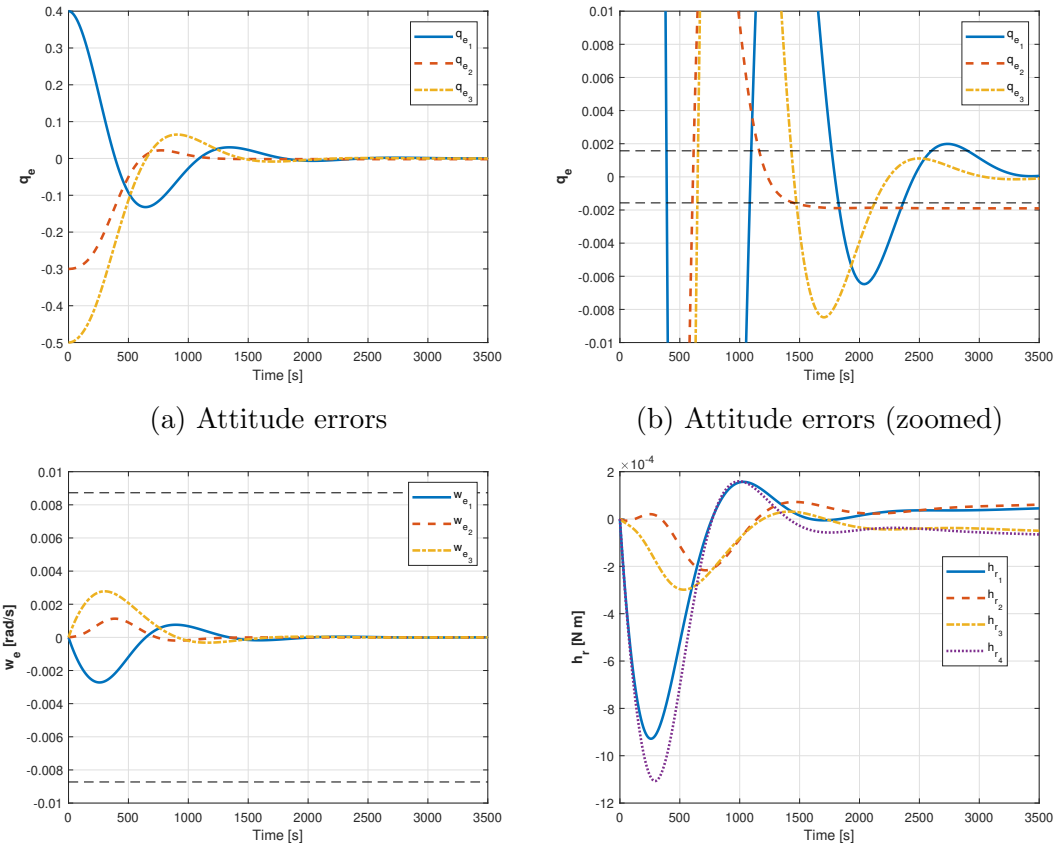
All the numerical results of of the subsection have been summarized in table 3.1. There, it is reported the time it takes for the simulation to enter the attitude error requirement in the fourth column, the total sum of control moment $\sum \dot{h}_{r_i}$ used by all RWs in the fifth column, the sum of the RWs angular momentum at the end of the simulation $\sum |h_{r_i}|$ in the sixth column, and the maximum value reached by the angular velocity $|w_{i,max}|$ during the manoeuvre in the last and seventh column. The value of $\sum |h_{r_i}|$ is important because it is the angular momentum to desaturate after the manoeuvre, and the wheels have to be desaturated using limited thruster propellant. Thus, it is convenient to have control laws that balance out the wheels automatically, without needing another layer of control logic to do it.

When there are errors, the values depend on which RW presents error or errors, so a range is given for the lower and higher values obtained from all combinations.

No failures. The results with SCM are shown in Figure 3.2. The response is very slow, taking over 45 minutes to complete and with high overshoot. On top of that, the disturbance makes the attitude stay just outside of the requirement in the steady state. Angular velocity stays very far away from the limits.

The results with DCM are shown in Figure 3.3. Response is much more aggressive, taking 133 seconds, but angular velocity limits are wildly exceeded.

1 RW failure. The effect of having one failure is that the controllers become less precise. For SCM, shown in Figure 3.4a, the response is even slower than before, taking over three hours, and now the attitude oscillates sensibly outside the requirement region. For DCM, shown in Figure 3.5a, the response is degraded and now shows considerable overshooting, when before it did not. The angular velocity is almost unaffected and is still exceedingly large.



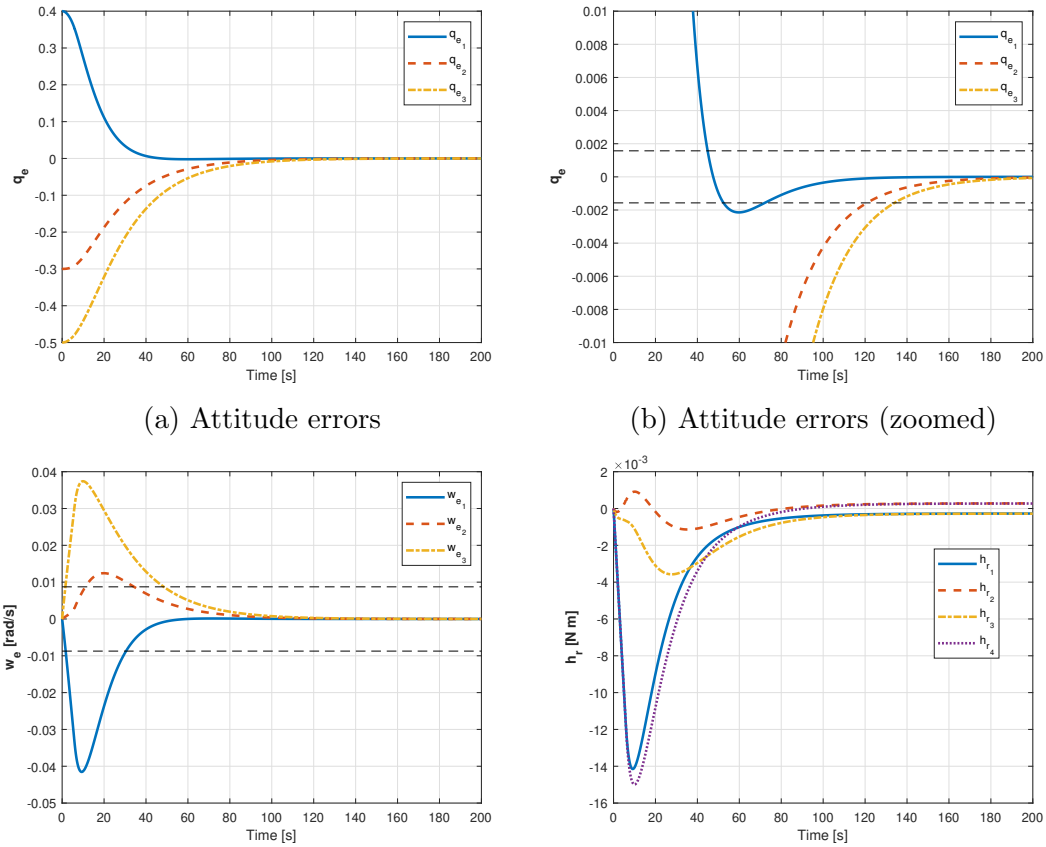
(a) Attitude errors

(b) Attitude errors (zoomed)

(c) Angular velocity errors

(d) RWs angular momentum

Figure 3.2: Results for slew manoeuvre with no failures: SCM original controller



(a) Attitude errors

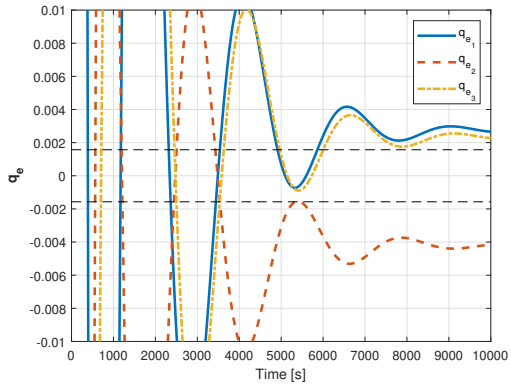
(b) Attitude errors (zoomed)

(c) Angular velocity errors

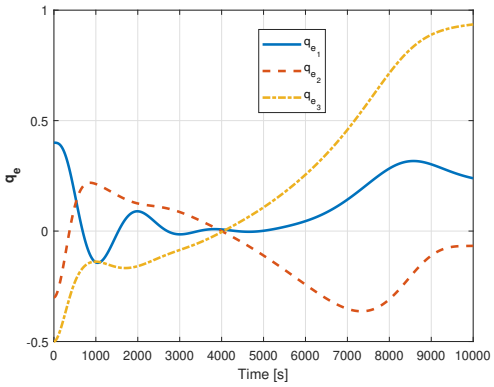
(d) RWs angular momentum

Figure 3.3: Results for slew manoeuvre with no failures: DCM original controller

2 RWs failures. None of the controllers achieve the objective, as was expected because now the spacecraft is under-actuated. For SCM the attitude drifts uncontrollably, as seen in Figure 3.4b, while for DCM at least the attitude stabilizes, as seen in Figure 3.5b.

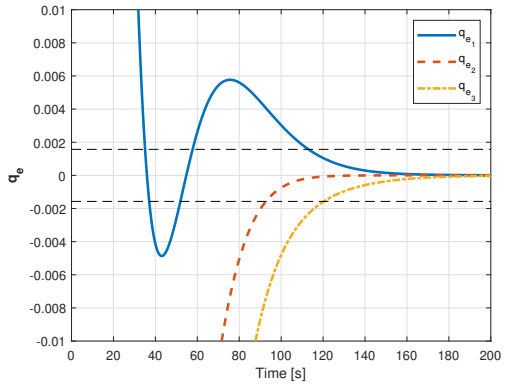


(a) Attitude errors (zoomed) with 1 RW failure

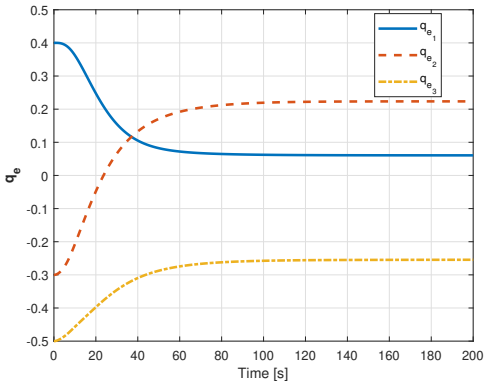


(b) Attitude errors with 2 RWs failures

Figure 3.4: Results for slew manoeuvre with failures: SCM original controller



(a) Attitude errors (zoomed) with 1 RW failure



(b) Attitude errors with 2 RWs failures

Figure 3.5: Results for slew manoeuvre with failures: DCM original controller

3.2.2 2018 Qiang Shen *et al.*

The results of the subsection are summarized in table 3.2, where instead of the mode like in table 3.1, now it is reported the value of the parameter k. This is done this way because due to the LUMIO baseline having a much smaller inertia

Mode	**Fail-ures**	**Success**	**Time to target [s]**	**Total control used [Nms]** $\times 10^{-2}$	**Final** $\sum \lvert h_{r_i} \rvert$ **[Nms]** $\times 10^{-3}$	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
SCM	0	Almost	2850	0.6	0.22	0.28
	1	Almost	10000+	0.7 to 1.1	0.64 to 0.77	0.20 to 0.22
	2	No, unstable	N/A	Increasing	N/A	N/A
DCM	0	Yes	133	7	1.1	4.2
	1	Yes	120 to 132	6.7 to 10.9	0.01 to 0.02	3.5 to 4.3
	2	No, stable	N/A	0.3 to 6.3	0.01 to 0.03	0.2 to 4.2

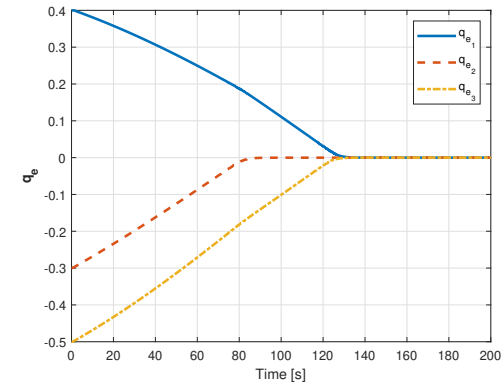
Table 3.1: Results summary for slew manoeuvre with original controller

matrix than the used in the original, it was though that the controller had to be modified to make it less aggressive. Later, it was shown the difference of changing k inside a wide range is pitifully small. The rest of the parameters remain the same except for $w_{e,max}$, which is changed to $0.5\, deg/s$ to fit the requirements.

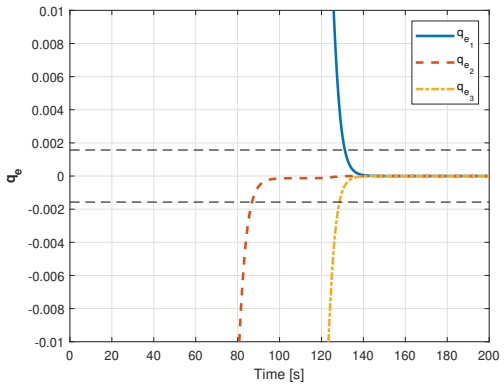
No failures. As seen in Figure 3.6, the response for $k = 1$ is as fast as with the more aggressive DCM controller with two crucial differences: now no overshoot happens whatsoever, and the angular velocity limitation in cleanly respected. The same response is observed for $k = 100$. For $k = 0$ the response is too undamped and it oscillates greatly.

1 RW failure. The effect of having a failure with $k = 1$ is that, at the beginning, appears a peak in velocity that in some cases makes it go outside the limit for a very small time. See in $\mathbf{w}_{e_2}$ in Figure 3.7. Also, extremely small overshoots appear, see again 3.7. However, for $k = 100$ this disadvantages disappear completely.

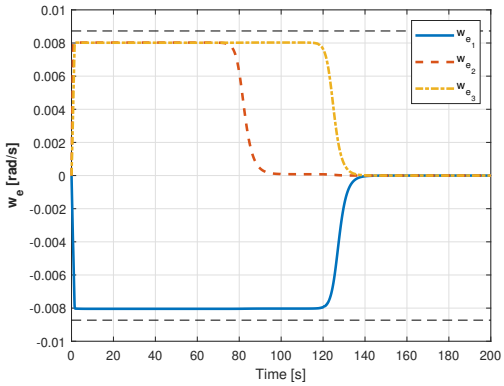
2 RWs failures. The behaviour with two failures is the same for $k = 1$ and $k = 100$. The controller tries to approach the desired attitude as much as it can with the under-actuated spacecraft, as can be seen in Figure 3.8. The angular velocity may get outside of the boundaries, but at least the response remains stable and the angular velocity violation is not very bad.



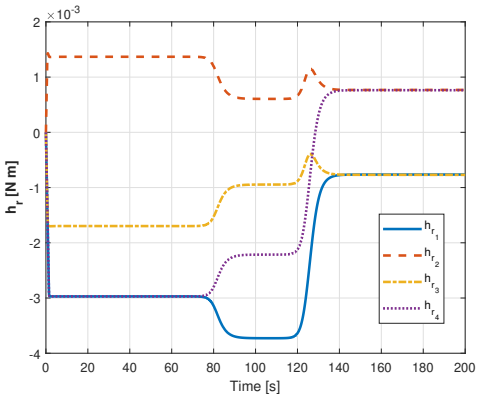
(a) Attitude errors



(b) Attitude errors (zoomed)

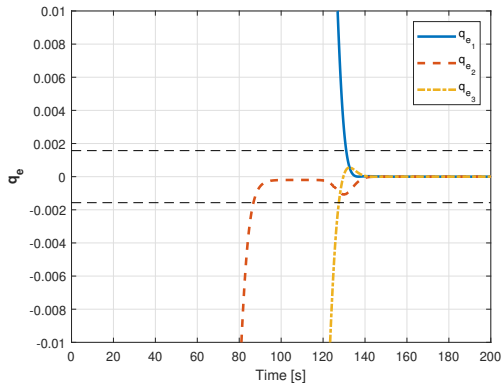


(c) Angular velocity errors

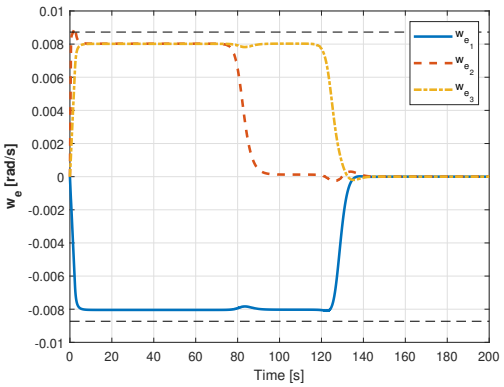


(d) RWs angular momentum

Figure 3.6: Results for slew manoeuvre with no failures: 2018 Qiang Shen *et al.* controller ($k = 1$)

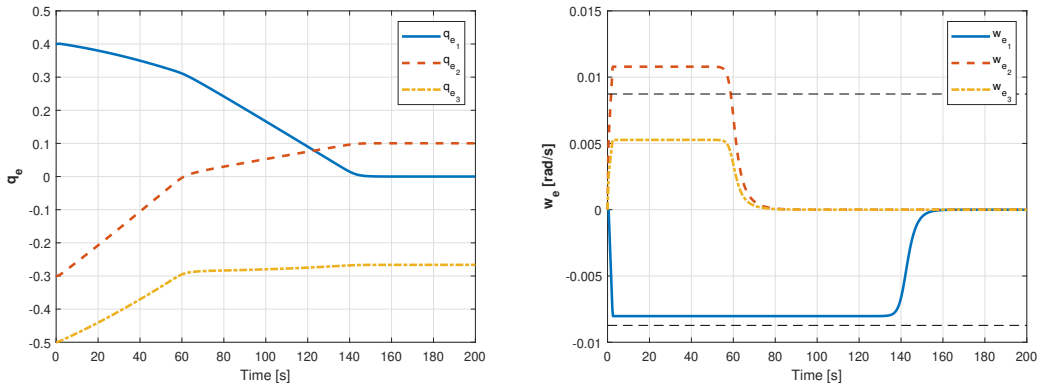


(a) Attitude errors (zoomed)



(b) Angular velocity errors

Figure 3.7: Results for slew manoeuvre with 1 failure: 2018 Qiang Shen *et al.* controller ($k = 1$)



(a) Attitude errors (b) Angular velocity errors

Figure 3.8: Results for slew manoeuvre with 2 failures: 2018 Qiang Shen *et al.* controller ($k = 100$)

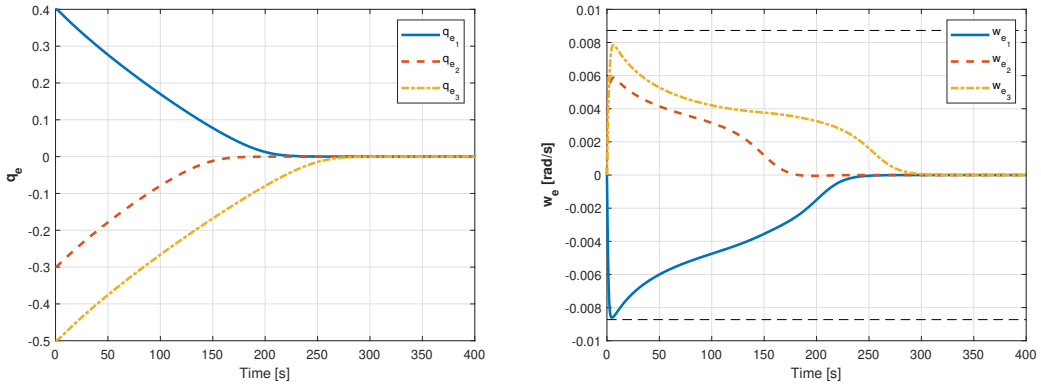
Failures	k	**Success**	**Time to target [s]**	**Total control used [Nms]** $\times 10^{-2}$	**Final** $\sum \lvert h_{r_i} \rvert$ **[Nms]** $\times 10^{-3}$	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	0	Yes	380	3.9	3.1	0.94
	1	Yes	130	2	3.1	0.80
	100	Yes	132	2	3.1	0.80
1	1	Yes	130 to 131	2.14 to 3.1	0.01 to 0.02	0.86 to 0.9
	100	Yes	132 to 133	2.1 to 3	0.01 to 0.02	0.80
2	1	No, stable	N/A	0.6 to 1.81	0.01 to 0.07	0.80 to 1.08
	100	No, stable	N/A	0.6 to 1.8	0.01 to 0.067	0.8 to 1.08

Table 3.2: Results summary for slew manoeuvre with 2018 Qiang Shen *et al.* controller

3.2.3 2014 Danyal Bustan *et al.*

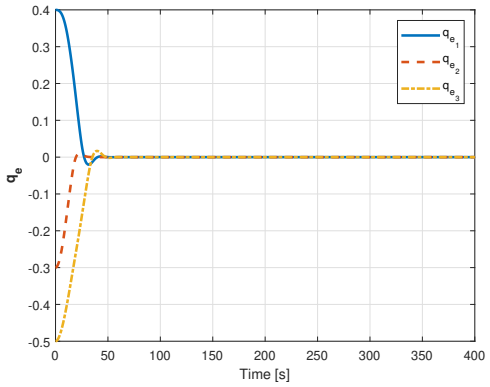
Now in the summary of table 3.3 it is reported the value of the parameter ρ. The rest of the parameters remain the same except for the value of u_{max}, which is changed to 5×10^{-3}.

No failures. The parameter ρ plays a trade-off effect between respecting the limitation on angular velocity and the time it takes to complete the manoeuvre. As is shown in Figure 3.9, with $\rho = 10$ it is exactly respected the angular velocity limit. Lower values of ρ make the response faster but with higher angular velocities and higher control torque used, as summarized in table 3.3. With $\rho = 1$ the response takes only 53 seconds but a slight overshooting starts to appear. With $\rho = 0$ the improvement on time is reversed due to very bad oscillations in the response.

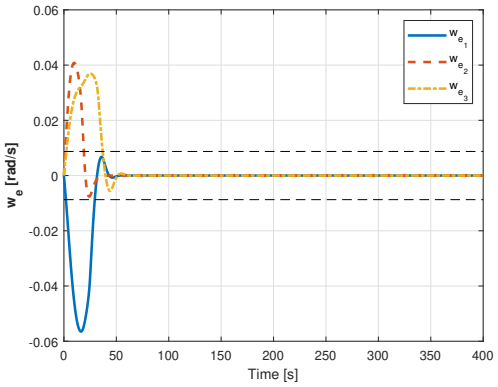


(a) Attitude errors ($\rho = 10$)

(b) Angular velocity errors ($\rho = 10$)



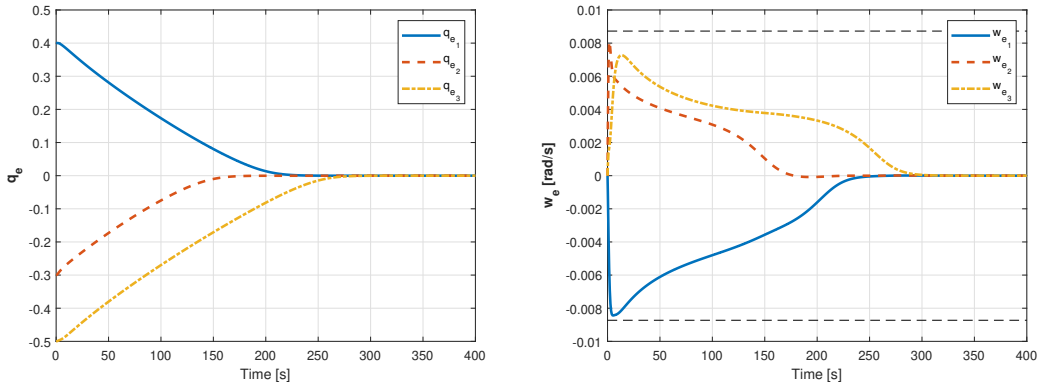
(c) Attitude errors ($\rho = 1$)



(d) Angular velocity errors ($\rho = 1$)

Figure 3.9: Results for slew manoeuvre with no failures: 2014 Danyal Bustan *et al.* controller

1 RW failure. The response to one failure for $\rho = 10$ is almost unaffected, as it is represented in Figure 3.10, only with a slight increase in the control used. For $\rho = 1$ however the failure causes oscillations, which in turn causes a significant increase in the convergence time.



(a) Attitude errors ($\rho = 10$) (b) Angular velocity errors ($\rho = 10$)

Figure 3.10: Results for slew manoeuvre with 1 failure: 2014 Danyal Bustan *et al.* controller

2 RWs failures. The response for two failures is very similar to that of the method 2018 Qiang Shen *et al.*: the controller tries to minimize the error all that it can with the under-actuated control, and the attitude stabilizes after a short time.

3.2.4 2008 Wenchuan Cai *et al.*

For this method there has been discovered a coupling in the effects of the parameters β, F_{max}, and the magnitude of the inertia matrix. These three aspects have to be balanced very carefully because when the controller enters the "saturation" mode, the response starts having extreme overshoot and oscillations. Check table 3.4 for more details on the effect that changing β and F_{max} has on the response. The remaining parameters are found to have no significant effect and remain unchanged.

No failure. As a general rule, it is wanted F_{max} to be as large as possible, so values around 2.1×10^{-3} are the best, because if larger it would exceed the actuation of the RWs. The inertia matrix $\mathbf{I}$ is imposed, so the only parameter left to vary is β. When increasing it beyond $\beta = 0.25$, the controller enters saturation and it oscillates wildly, as shown in Figure 3.11 with the case of $\beta = 0.5$. The $\beta = 0.25$ value seems to be a good compromise to achieve low manoeuvre time,

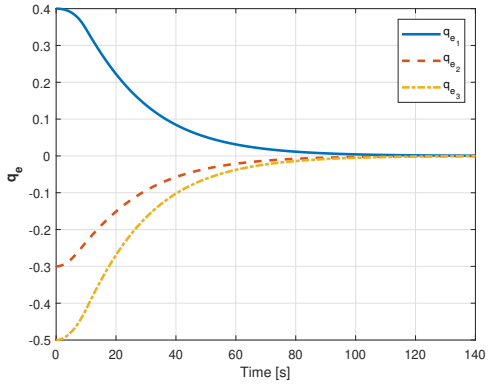
Failures	ρ	**Success**	**Time to target [s]**	**Total control used [Nms]** $\times 10^{-2}$	**Final** $\sum \lvert h_{r_i} \rvert$ **[Nms]** $\times 10^{-3}$	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	0	Yes	151	52.5	8.3	10.43
	1	Yes	53	12.96	2.7	5.65
	5	Yes	142	3.06	0.59	1.62
	10	Yes	281	1.65	0.30	0.86
1	1	Yes	76 to 128	10.62 to 19.48	0.02 to 0.03	4.21 to 5.79
	10	Yes	279 to 280	1.57 to 2.65	0.02 to 0.03	0.79 to 0.87
2	1	No, stable	N/A	6.00 to 11.30	0.02 to 0.09	3.33 to 6.73
	10	No, stable	N/A	0.49 to 1.55	0.02 to 0.13	0.43 to 1.31

Table 3.3: Results summary for slew manoeuvre with 2014 Danyal Bustan *et al.* controller

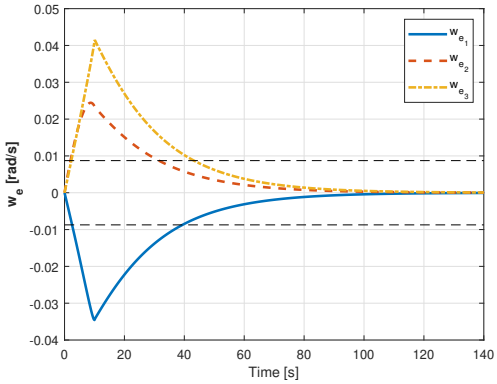
but it exceeds the angular velocity limit, as seen in Figure 3.11. Reducing β further makes the angular velocity lower and the response take more time, as shown in Figure 3.11 for the case of $\beta = 0.1$. Not shown in the figures is the value of $\beta = 0.018$, that makes the angular velocity exactly respect the boundaries, but for a exceedingly long time of 656 seconds.

1 RW failure. A failure affects more this method than the other two adaptive ones. The result is still reaching the objective, but with an increase in time taken, control torque used, and final RWs angular momentum left to desaturate. Now even with $F_{max} = 2.1 \times 10^{-3}$ and $\beta = 0.25$, there is significant overshoot, as seen in Figure 3.12.

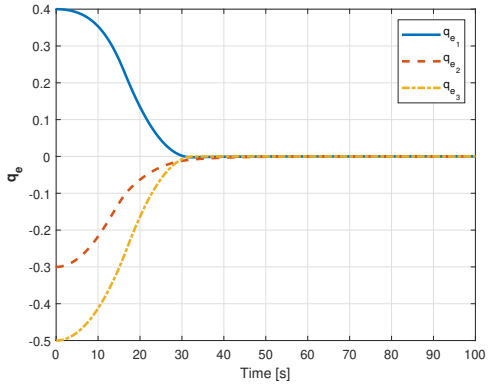
2 RW failures. With two failures the result is an increase in oscillations and time taken. Of course now it does not reach the objective, as expected, but it remains stable, as seen in Figure 3.13.



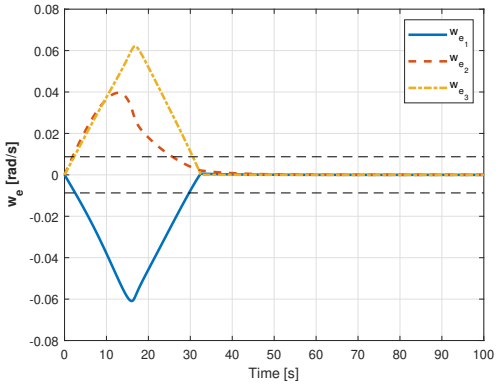
(a) Attitude errors ($\beta = 0.1$)



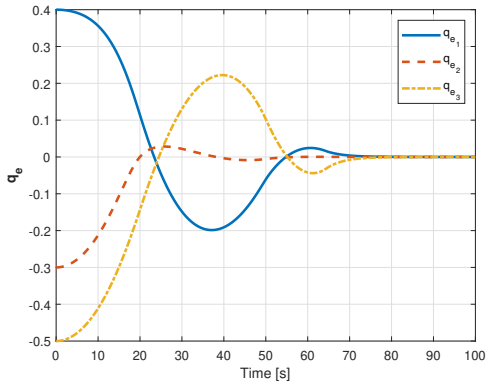
(b) Angular velocity errors ($\beta = 0.1$)



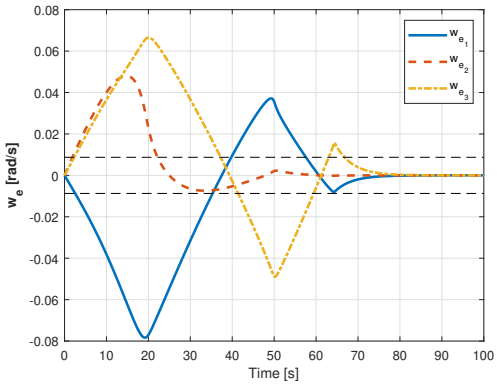
(c) Attitude errors ($\beta = 0.25$)



(d) Angular velocity errors ($\beta = 0.25$)

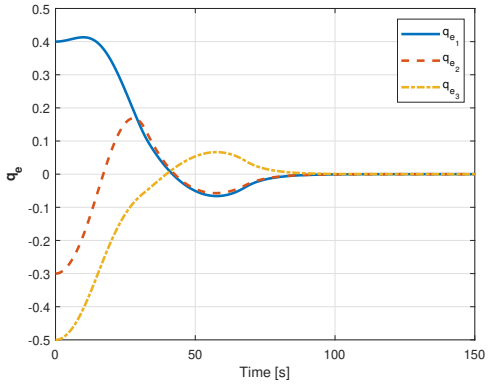


(e) Attitude errors ($\beta = 0.5$)

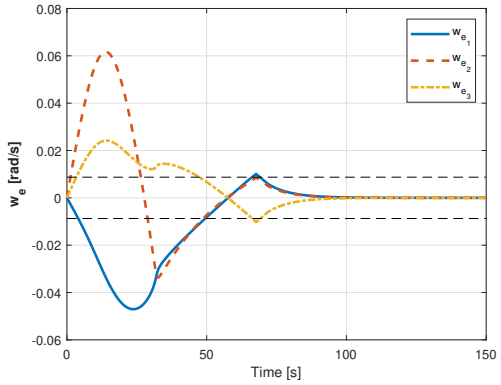


(f) Angular velocity errors ($\beta = 0.5$)

Figure 3.11: Results for slew manoeuvre with no failures: 2008 Wenchuan Cai *et al.* controller (all with $F_{max} = 2.1 \times 10^{-3}$)

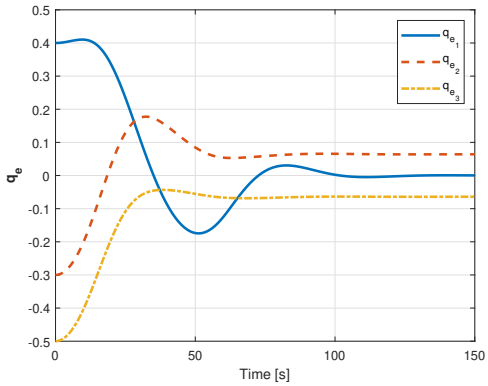


(a) Attitude errors ($\beta = 0.25$)

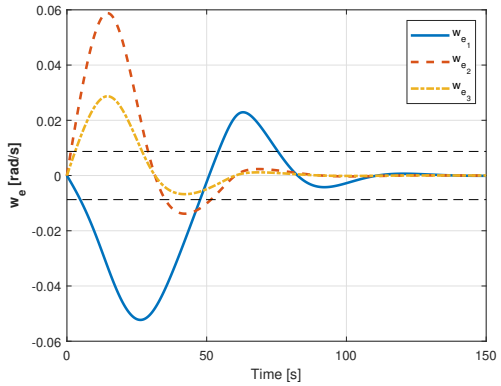


(b) Angular velocity errors ($\beta = 0.25$)

Figure 3.12: Results for slew manoeuvre with 1 failure: 2008 Wenchuan Cai *et al.* controller ($F_{max} = 2.1 \times 10^{-3}$)



(a) Attitude errors ($\beta = 0.25$)



(b) Angular velocity errors ($\beta = 0.25$)

Figure 3.13: Results for slew manoeuvre with 2 failures: 2008 Wenchuan Cai *et al.* controller ($F_{max} = 2.1 \times 10^{-3}$)

Failures	F_{max} $\times 10^{-3}$	β	**Success**	**Time to target [s]**	**Total control used [Nms]** $\times 10^{-2}$	**Final** $\sum \lvert h_{r_i} \rvert$ **[Nms]** $\times 10^{-3}$	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	2.1	0.25	Yes	46	10.9	0.01	6.2
	2.1	0.5	Yes	76	23.6	0.01	7.8
	2.1	0.1	Yes	123	6.7	0.06	4.1
	2.1	0.018	Yes	656	1.44	0.06	0.89
	5	0.25	Yes	54	13.2	6.10	7.2
	0.5	0.25	Yes	159	11.8	0.01	3.8
1	2.1	0.25	Yes	50 to 137	10.4 to 22.1	0.01 to 0.15	4.0 to 6.4
2	2.1	0.25	No, stable	N/A	11.8 to 17.7	0.02 to 0.07	3.0 to 6.5

Table 3.4: Results summary for slew manoeuvre with 2008 Wenchuan Cai *et al.* controller

3.3 Second scenario: Detumbling

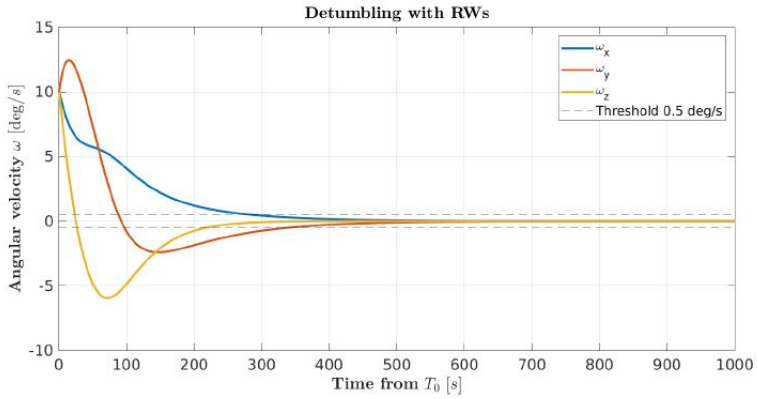
For this scenario the objective is to reduce the angular velocity in all axes to less than the $w_{i,max} = 0.5\, deg/s$ given by requirement 2). The primary strategy for detumbling is to perform it using the RCS of the spacecraft. However it is also explored the possibility to do it with the RWs, which is what is going to be replicated in this section. For this scenario the source actually reports results obtained with the controller. In Figure 3.14 it is shown the angular velocity and RWs angular momentum that is reported in [1].

The initial angular velocity $w_i(0)$ is going to be kept at $10\, deg/s$ in all axis as it was done in the source reported results and as is indicated by the requirements. Because there is not any attitude requirements and the objective is $\mathbf{w} = \mathbf{0}_{3\times1}$, in the adaptive techniques what has been done is set the attitude error $\mathbf{q}_e = \mathbf{0}_{3\times1}$, and the angular velocity error is directly equal to the angular velocity, $\mathbf{w}_e = \mathbf{w}$.

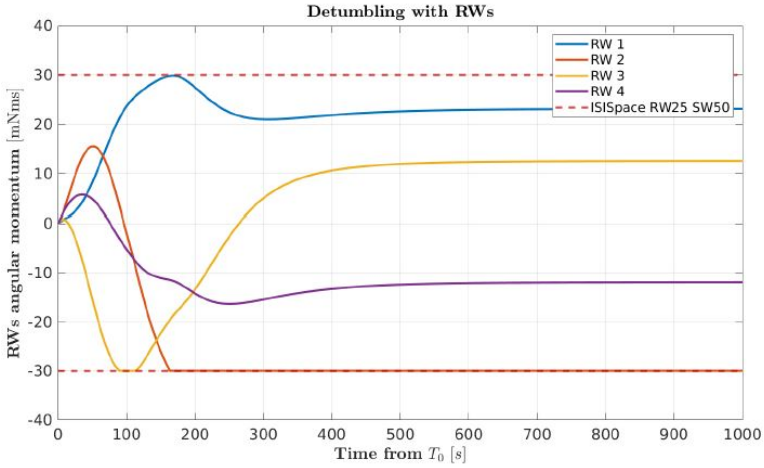
3.3.1 Original controller

No failures. Immediately for this baseline, a serious discrepancy has been found. The results on angular velocity have been replicated exactly, as is shown in Figure 3.15, which can be seen is the same as in Figure 3.14. However, this results are ONLY achieved when the total RWs angular momentum capacity is not limited. Looking again at Figure 3.15, the values of $h_{r,i}$ go outside the limits of $\pm 0.03\, N\, m$.

In fact, the result presented by the source is impossible, and this can be proven.



(a) Angular velocity w_i



(b) RWs angular momentum $h_{r,i}$

Figure 3.14: Results for detumbling with no failures reported by the source [1]

The disturbance can be neglected for its small magnitude over this short period of time, so the system becomes isolated. That means the total angular momentum vector must remain constant measured in an inertial frame, $\mathbf{h}|_I = const$. The results are given in body frame, but what must remain constant in all frames is the modulus $|\mathbf{h}| = const$. The initial value of this magnitude is $|\mathbf{h}(0)| = |\mathbf{I}\mathbf{w}(0)| = 6.33 \times 10^{-2}\ N\ m\ s$.

Well, taking the values reported by [1] to be $\mathbf{h}_r(end) = [2.3,\ -3,\ 1.3,\ -1.2] \times 10^{-2}$, the final value would be $|\mathbf{h}(end)| = |\mathbf{R}\mathbf{h}_r(end)| = 1.7 \times 10^{-2}$. Even making all them positive and being generous with the values so the modulus can grow as much as possible, with $\mathbf{h}_r(end) = [2.5,\ 3,\ 1.5,\ 1.5] \times 10^{-2}$ the value reached is 5.12×10^{-2} as an absolute upper boundary. This does not comply with the principle of conservation of angular momentum. By comparison, with the values obtained by the replication and shown on Figure 3.15, the results is $|\mathbf{R}\mathbf{h}_r(end)| = 6.33 \times 10^{-2}$, which is exactly what is expected.

The results when $h_{r,i}$ is limited are very different, and are going to be reported later. For now and for the sake of consistency, in this and later subsections it is going to be presented the case when the value of $h_{r,i}$ is not limited. The cases when $h_{r,i}$ is limited will be presented in subsection 3.3.5.

1 RW failure. In presence of a single failure, the effect is a sudden and complete loss of control. As shown in Figure 3.16, the spacecraft starts to oscillate with increasing magnitude.

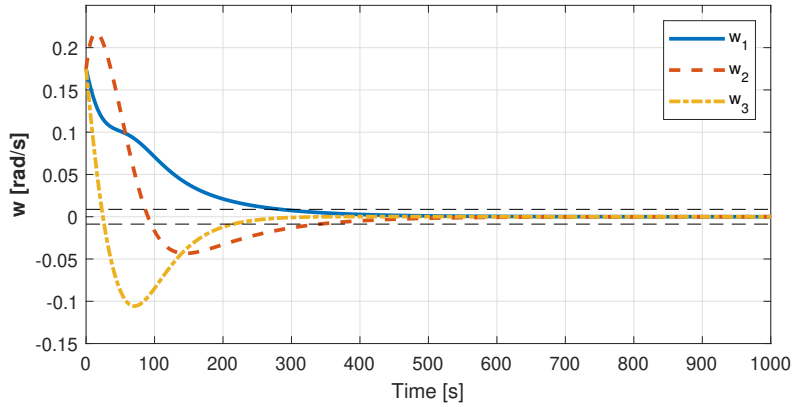
Failures	**Success**	**Time to target [s]**	**Total $\sum \dot{h}_{r_i}$ used [Nms] $\times 10^{-2}$**	**Final $\sum h_{r_i}$ [Nms] $\times 10^{-2}$**
0	Yes	340	21	10
1	No, unstable	N/A	N/A	N/A

Table 3.5: Results summary for detumbling with original controller

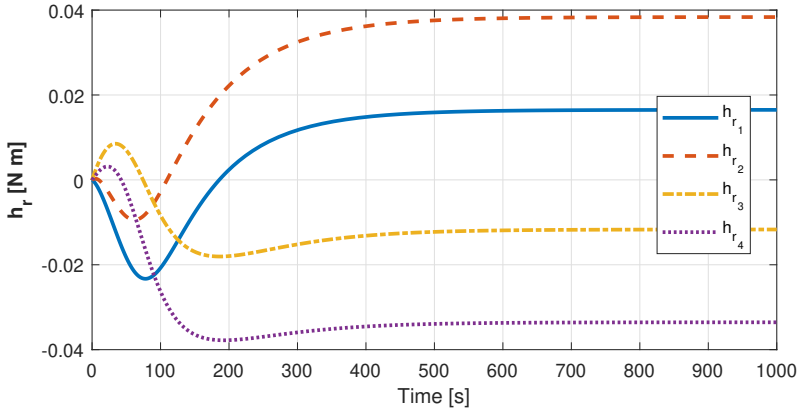
3.3.2 2018 Qiang Shen *et al.*

No failures. As summarized in table 3.6, this time changing the controller has no effect on the results. The response is very fast and satisfactory, with no overshoot. See Figure 3.17.

1 failure. The response is degraded with overshoot, but the objective is still reached, with slight increases in time and control used. See Figure 3.17.



(a) Angular velocity w_i



(b) RWs angular momentum $h_{r,i}$

Figure 3.15: Results for detumbling with no failures: original controller

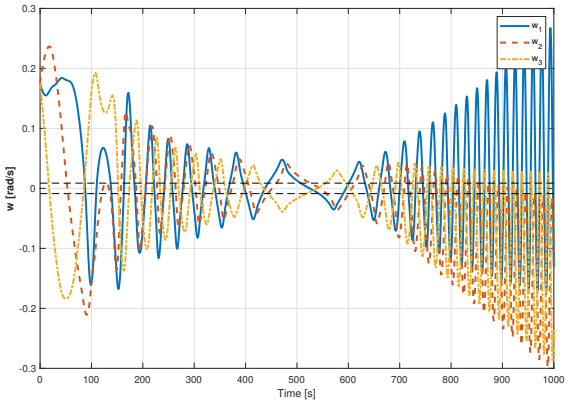
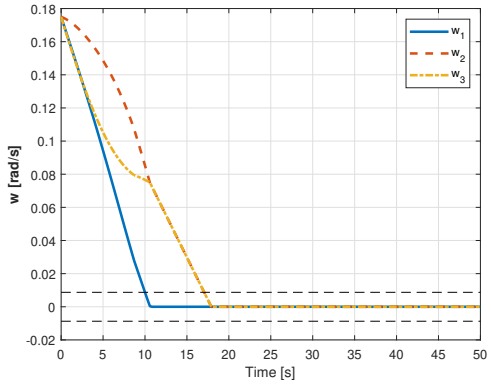
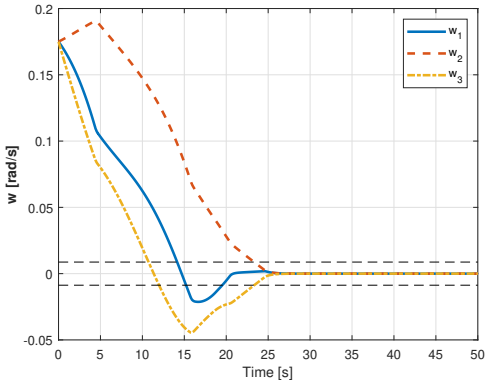


Figure 3.16: Angular velocity for detumbling with 1 failure: original controller

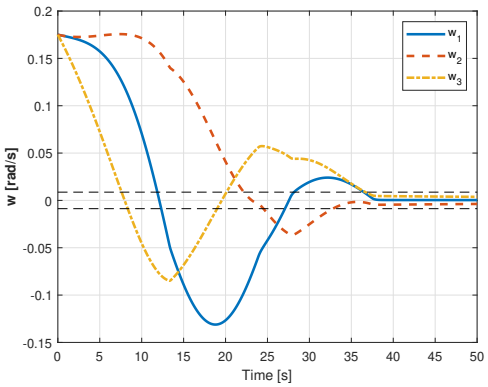
2 failures. The response is degraded further, with more time and overshoot. When selecting $k = 100$ there are cases in which the objective is not reached, with the control torque increasing over time due to the wheels balancing back and forth. See Figure 3.17 and table 3.6.



(a) No failures



(b) 1 failure



(c) 2 failures

Figure 3.17: Angular velocity w_i results for detumbling: 2018 Qiang Shen *et al.* controller

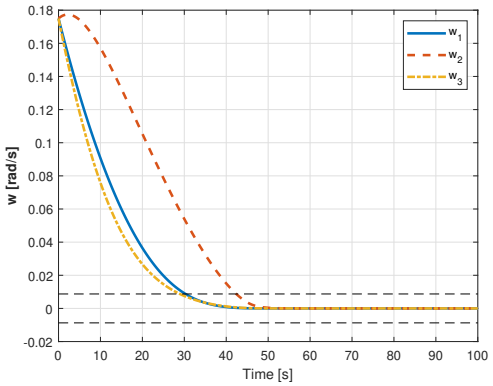
3.3.3 2014 Danyal Bustan *et al.*

Now the parameter ρ does not have any effect because $\dot{\mathbf{q}}_e$, which it is multiplying to in equation (2.17), is set to zero.

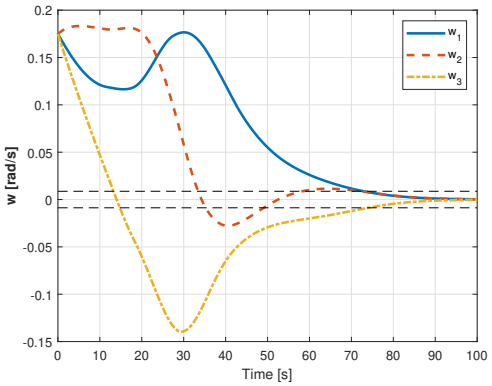
With **no failures**, the response is very fast and smooth. With **1 RW failure** it is degraded in time taken and control used, and even with **2 RWs failures** the objective is achieved, although in much more time and with significant oscillations. All of these behaviours can be seen in Figure 3.18 and table 3.7.

Failures	k	**Success**	**Time to target [s]**	**Total $\sum \dot{h}_{r_i}$ used [Nms] $\times 10^{-2}$**	**Final $\sum h_{r_i}$ [Nms] $\times 10^{-2}$**
0	0.01	Yes	17	12.2	11.1
	1	Yes	17	12.5	11.0
	100	Yes	17	12.5	11.0
1	0.01	Yes	23 to 39	11.1 to 18.2	8.2 to 18.2
	1	Yes	20 to 34	11.4 to 18.0	8.6 to 18.0
	100	Yes	20 to 34	11.6 to 17.8	8.6 to 17.7
2	0.01	Yes	40 to 55	14.5 to 18.4	7.2 to 7.8
	1	Yes	37 to 55	14.9 to 21	6.3 to 7.7
	100	Almost always	41 to ∞	17.8 to ∞	7.3 to 9.2

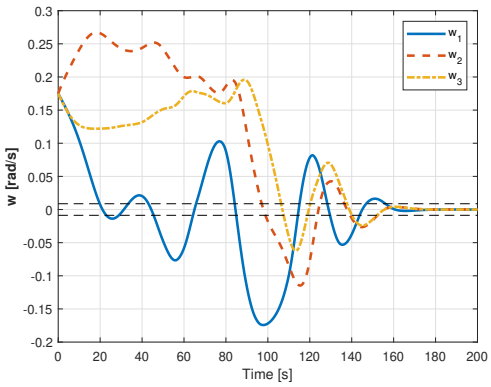
Table 3.6: Results summary for detumbling with 2018 Qiang Shen *et al.* controller



(a) No failures



(b) 1 failure



(c) 2 failures

Figure 3.18: Angular velocity w_i results for detumbling: 2014 Danyal Bustan *et al.* controller

Failures	Success	Time to target [s]	Total $\sum \dot{h}_{r_i}$ used [Nms] $\times 10^{-2}$	Final $\sum h_{r_i}$ [Nms] $\times 10^{-2}$
0	Yes	42	11.0	10.5
1	Yes	49 to 112	11.5 to 32.9	8.4 to 18.5
2	Yes	71 to 168	13.2 to 37.7	6.5 to 8.8

Table 3.7: Results summary for detumbling with 2014 Danyal Bustan *et al.* controller

3.3.4 2008 Wenchuan Cai *et al.*

Similarly here, the parameter β does not have any effect because $\mathbf{q}_e$, which it is multiplying to in equation (2.21), is set to zero. The value of F_{max} is left to be 2.1×10^{-3}.

With **no failures** this method is fast, although it is not better than [18], and uses the least control effort. However, it is relatively more affected by failures, increasing considerably the time, control used, and oscillations with **1 RW failure**, and even more with **2 RWs failures**. See Figure 3.19 and table 3.8 for details.

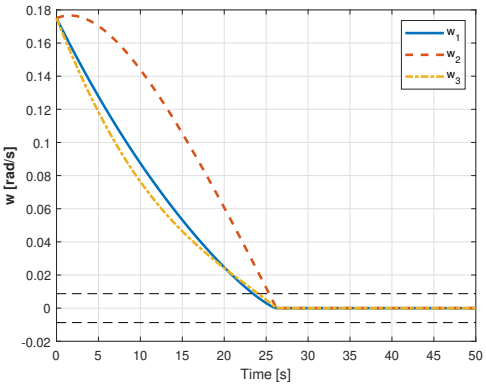
Failures	Success	Time to target [s]	Total $\sum \dot{h}_{r_i}$ used [Nms] $\times 10^{-2}$	Final $\sum h_{r_i}$ [Nms] $\times 10^{-2}$
0	Yes	25	9.8	9.3
1	Yes	29 to 95	9.1 to 27.3	7.5 to 18.9
2	Yes	43 to 303	12.8 to 66.2	7.1 to 8.4

Table 3.8: Results summary for detumbling with 2008 Wenchuan Cai *et al.* controller

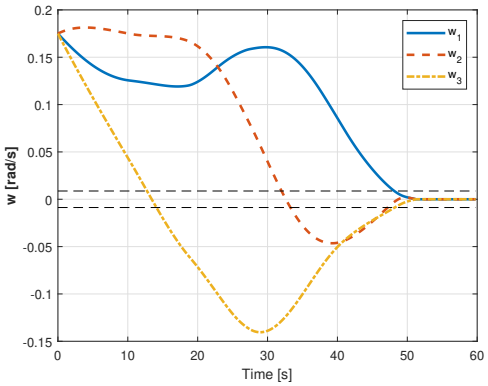
3.3.5 Detumbling with limited RWs angular momentum $h_{r,i}$

This subsection reports what happens when it is applied a limit in the RWs angular momentum $h_{r,i}$, as it was intended from the selection of the component in the ACS design of [1]. Results are summarized in table 3.9.

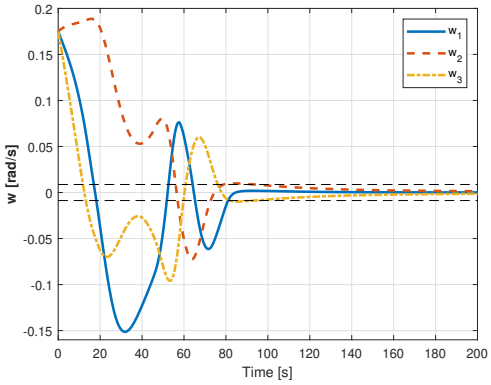
No failures. The original controller is not able to achieve the detumbling even under no failures. As is seen in Figure 3.20, the spacecraft remains rotating indefinitely. With the other three adaptive methods, detumbling is achieved without any problems. The only change that had to be done in these controllers has been to lower k to 0.1 in the 2018 Qiang Shen *et al.* method.



(a) No failures

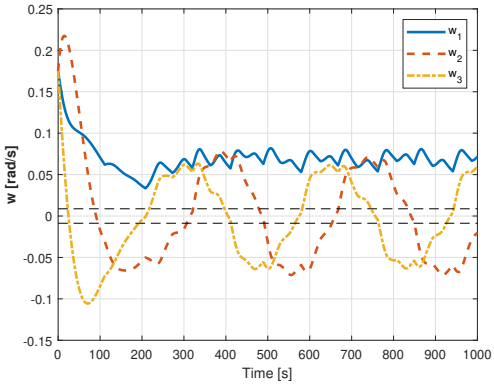


(b) 1 failure

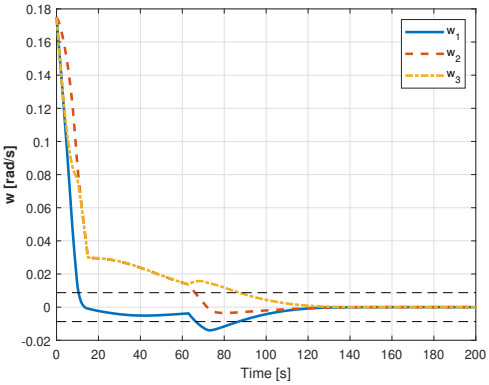


(c) 2 failures

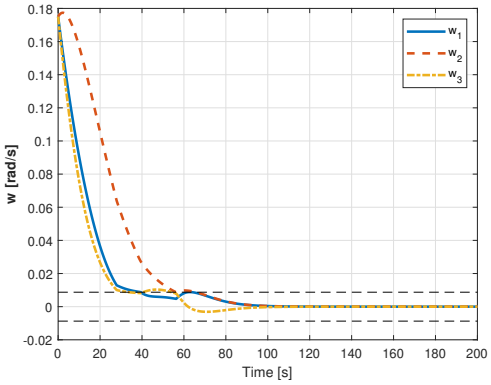
Figure 3.19: Angular velocity w_i results for detumbling: 2008 Wenchuan Cai *et al.* controller



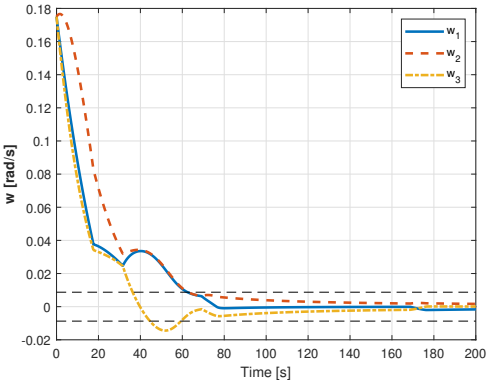
(a) Original controller



(b) 2018 Qiang Shen *et al.* controller



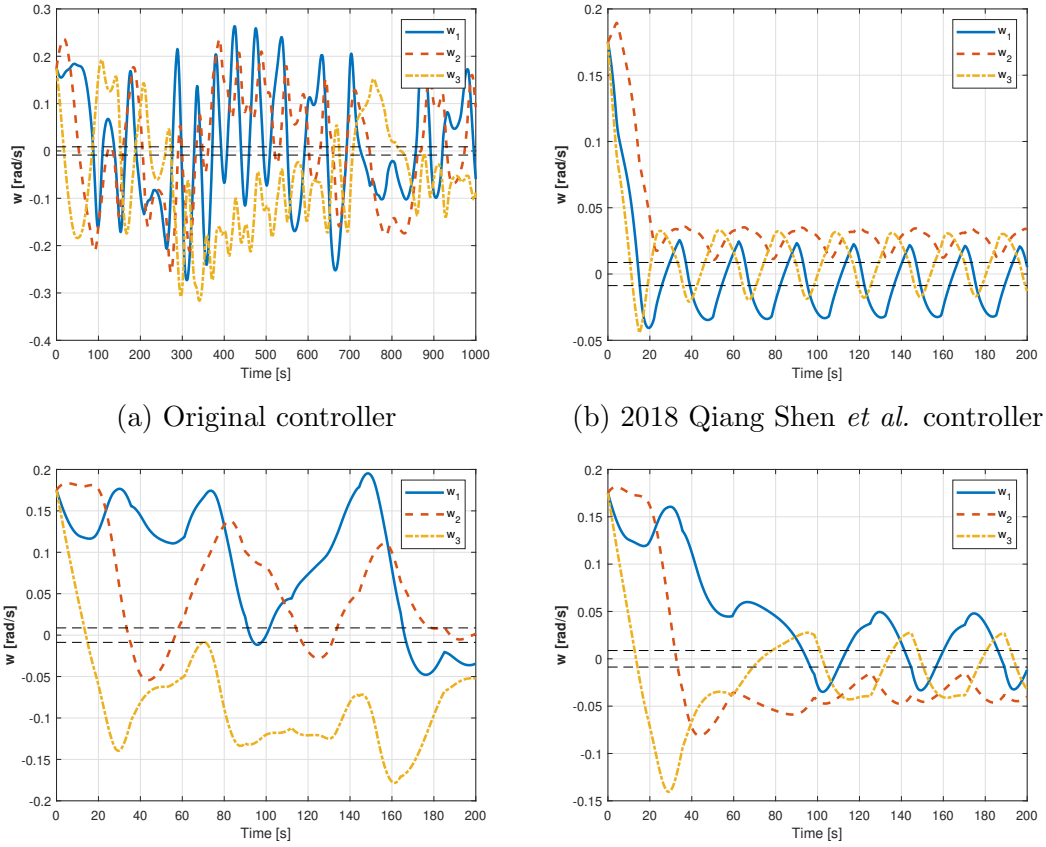
(c) 2014 Danyal Bustan *et al.* controller



(d) 2008 Wenchuan Cai *et al.* controller

Figure 3.20: Angular velocity results for detumbling with limited $h_{r,i}$ and no failures

1 RW failure. What happens when there is one or more failures is that there is not physically enough angular momentum available on the wheels to achieve detumbling. The most that can be expected is for the methods to reduce the angular velocity and make it hover around the value zero. As seen in Figure 3.21, the original controller is totally unstable. 2014 Danyal Bustan *et al.* is also unstable but with slow changes on angular velocity. Finally, 2018 Qiang Shen *et al.* and 2008 Wenchuan Cai *et al.* are able to confine the angular velocity to a region close to the zero value.



(a) Original controller

(b) 2018 Qiang Shen *et al.* controller

(c) 2014 Danyal Bustan *et al.* controller

(d) 2008 Wenchuan Cai *et al.* controller

Figure 3.21: Angular velocity results for detumbling with limited $h_{r,i}$ and no failures

Failures	**Method**	**Success**	**Time to target [s]**	**Total $\sum \dot{h_{r_i}}$ used [Nms] $\times 10^{-2}$**	**Final $\sum h_{r_i}$ [Nms] $\times 10^{-2}$**
0	Original	No, unstable	N/A	N/A	N/A
	2018 Qiang Shen *et al.*	Yes	87	20.4	10.9
	2014 Danyal Bustan *et al.*	Yes	65	12.7	10.9
	2008 Wenchuan Cai *et al.*	Yes	60	16.7	10.8

Table 3.9: Results summary for detumbling with limited RWs angular momentum $h_{r,i}$

3.4 Third scenario: Long-term pointing

For the long term pointing, as it has been said at the beginning of this chapter, the mission has a time-frame of days. In fact, the longest cycles of science according to the source [1] are of 15 days. As much as it was wanted to make simulations with such a long time, the computational power to perform them is not available to the author of this Thesis, specially if multiple ones with different methods and scenarios are to be done.

What has been done instead is a compromise in simulation time and the scenario has been computed for a time of 10000 seconds. Now, in the summary tables, it is reported the total control moment used during the 10000 seconds, and the RWs angular momentum to desaturate at the end of that period. Also, this time it is reported the maximum absolute value of the error quaternion reached $|q_{e_{i,max}}|$.

Because the disturbance torque is the only external influence over the system, it is known that any control that follows the desired trajectory done only with the RWs will make the angular velocity of said wheels to follow the integral of the disturbance. In that sense, there would be no difference over the controllers, only on the way the total momentum is distributed among the wheels.

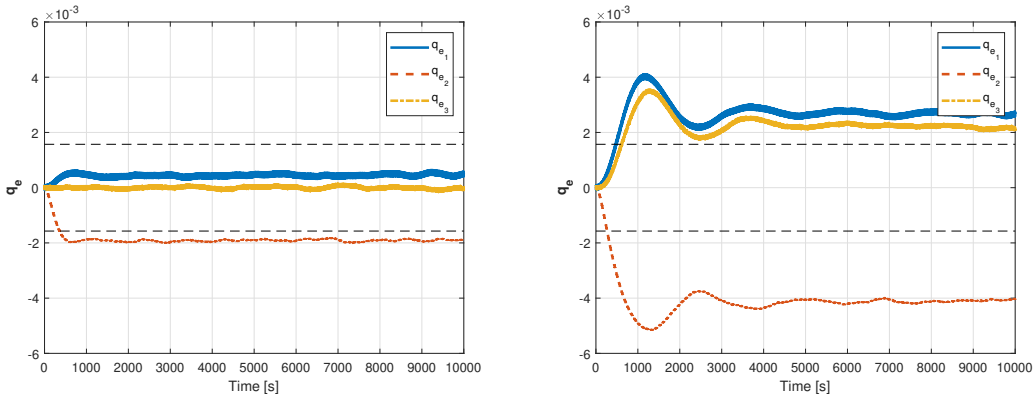
The trajectory chosen, as no specific information is available about the orbit, consists on a circular rotation around the Z axis. At least the orbit semi-axis is reported as $a = 35878\ [km]$, which results in an orbit with an angular rate of $\dot{\theta} = 2.13\ [arcseconds/s]$.

3.4.1 Original controller

The results show that with **no failure** the system stays just outside the boundaries of the requirement. This probably indicates that the controller parameters were purposefully selected this way, and that the worst-case disturbance imposed is taking it to its limits.

With **1 RW failure** the response is significantly degraded, although stable, and the errors stay well outside the requirement range, as seen in Figure 3.22. Sadly, as an effect of the measurement errors added for this scenario, now the style of the lines in the figures is not distinguishable in most cases.

In the case with **2 RWs failures**, which is not shown in the figures, the response starts drifting slowly but uncontrollably due to the spacecraft's under-actuation.



(a) No failures

(b) 1 RW failure

Figure 3.22: Attitude error results $\mathbf{q}_e$ for long-term pointing: original controller

Failures	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-3}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-4}$
0	Almost	2	1.9	6.6
1	Almost	4 to 5.2	1.5 to 1.6	6.6 to 7.8
2	No, unstable	N/A	N/A	N/A

Table 3.10: Results summary for long-term pointing with original controller

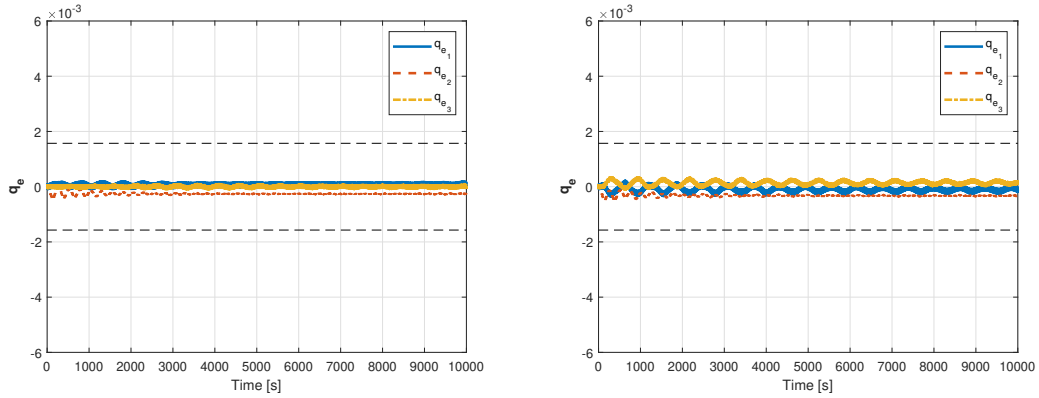
3.4.2 2018 Qiang Shen *et al.*

To avoid using too much control torque in long periods of time, arising from the measurement errors introduced, the controller in [18] had to be made much less aggressive than for the slew and detumbling manoeuvres. The best results are obtained reducing the parameter k to 10^{-4}. Also, the variable controlling the speed of convergence of the virtual velocity filter, T_0 is increased to 0.5.

The results are very satisfactory. As seen in table 3.11 and in Figure 3.23, with **no failures** the control torque is less than half that of the original controller, and the errors stay well within the allowed region.

With **1 RW failure** the control used and the attitude errors increase, but still remain less than with the original controller and inside the acceptable range.

With **2 RWs failures** the response is, as expected, the same as with the original controller, an uncontrolled drift, although less so than with the original.



(a) No failures

(b) 1 RW failure

Figure 3.23: Attitude error results $\mathbf{q}_e$ for long-term pointing: 2018 Qiang Shen *et al.* controller

3.4.3 2014 Danyal Bustan *et al.*

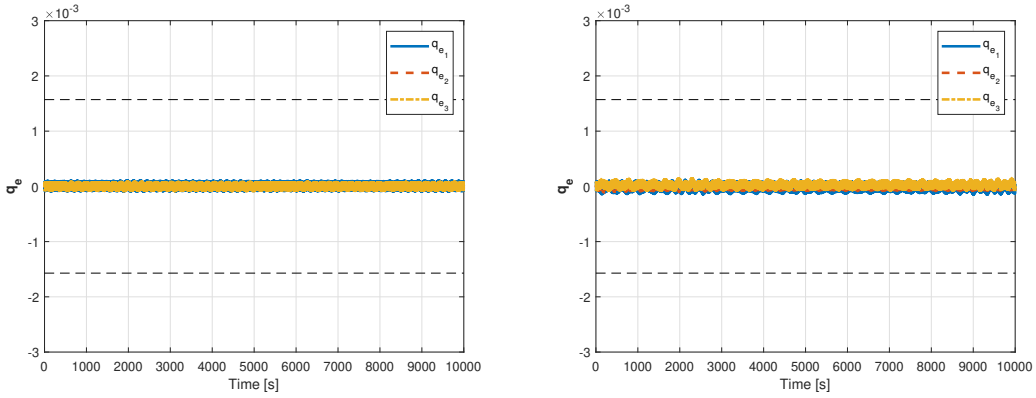
This time it has been observed that the change of parameter ρ does not have any effect on the controller. What has been done to tone down the controller aggressiveness is changing the maximum control effort to $u_{max} = 5 \times 10^{-5}$. This value has been chosen because it still offers a good margin compared to the magnitude of the disturbance. Note that this value of u_{max} could be reduced even more to reduce the control effort used, but a compromise has to be achieved with the attitude errors reached. See the results for $u_{max} = 5 \times 10^{-5}$ and $u_{max} = 5 \times 10^{-7}$ in table 3.12.

Failures	k	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-3}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-4}$
0	10^{-4}	Yes	0.41	0.67	6.6
1	10^{-4}	Yes	0.49 to 0.52	0.74 to 0.91	6.6 to 7.8
2	10^{-4}	No, unstable	N/A	N/A	N/A

Table 3.11: Results summary for long-term pointing with 2018 Qiang Shen *et al.* controller

With **no failures**, the controller takes more effort than 2018 Qiang Shen *et al.*, but still less than the original and with better results in accuracy. Looking at Figure 3.24, the requirement is respected.

The effect of **1 RW failure**, is just a small increase in control effort and attitude error, still inside the limits. With two errors the response is again a slow drift in quaternions.



(a) No failures

(b) 1 RW failure

Figure 3.24: Attitude error results $\mathbf{q}_e$ for long-term pointing: 2014 Danyal Bustan *et al.* controller

3.4.4 2008 Wenchuan Cai *et al.*

The process of adapting the parameters for this method is a bit more cumbersome than for the others, but it can be done. After a thorough review of the equations presented in section 2.4, and several trials, the controller has been able to be made less aggressive by changing the parameters to $k_0 = 0.001$, $\sigma_2 = 10$, $\beta = 1$ and $F_{max} = 2.1 \times 10^{-5}$. An even further modification of these parameters could

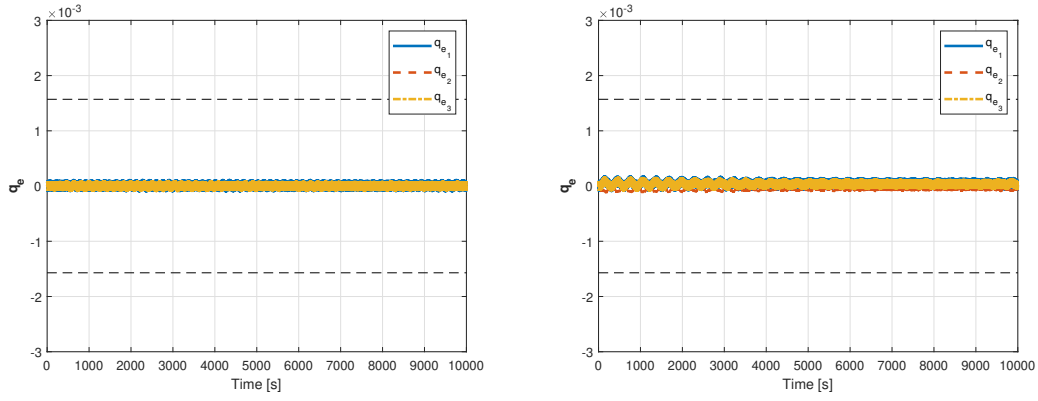
Failures	ρ	u_{max} $\times 5$	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-3}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-4}$
0	0	10^{-5}	Yes	0.11	1.36	6.6
	1	10^{-5}	Yes	0.11	1.32	6.6
	0	10^{-7}	Almost	7.6	0.69	6.5
1	0	10^{-5}	Yes	0.14 to 0.20	1.06 to 1.27	6.4 to 7.7
2	0	10^{-5}	No, unstable	N/A	N/A	N/A

Table 3.12: Results summary for long-term pointing with 2014 Danyal Bustan *et al.* controller

be attempted to reduce the control used, as there is still margin inside the error tolerance region.

As can be seen in Figure 3.25 and table 3.13, with **no failure** the controller takes less effort than the original and has better results, although it is still inferior to the other two adaptive methods explored.

The effect of **1 RW failure** is only a slight increase in the attitude errors, and the control used is even decreased. As expected, the system cannot handle **2 RWs failures** and the quaternions drift uncontrollably.



(a) No failures (b) 1 RW failure

Figure 3.25: Attitude error results $\mathbf{q}_e$ for long-term pointing: 2008 Wenchuan Cai *et al.* controller

Failures	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-3}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-4}$
0	Yes	0.11	1.7	6.6
1	Yes	0.15 to 0.18	1.4 to 1.7	6.6 to 7.7
2	No, unstable	N/A	N/A	N/A

Table 3.13: Results summary for long-term pointing with 2008 Wenchuan Cai *et al.* controller

3.5 Chapter conclusions

In this chapter the LUMIO CubeSat baseline described in [1] has been simulated for three scenarios: slew manoeuvre, detumbling and long-term pointing. Each scenario has been tested with the three selected adaptive methods and the original controller proposed. The discoveries made can be summarized in these points:

- For the **slew manoeuvre scenario**, the original SCM has a very slow response and stops meeting the pointing error requirements when there is a failure, while the original DCM is faster and robust to failures, but exceeds wildly the limitation on angular velocity for the manoeuvre. The adaptive methods 2014 Danyal Bustan *et al.* and 2018 Qiang Shen *et al.* are found to be robust, faster, take less control, and can be easily modified to comply with the limit in angular velocity. Particularly, 2018 Qiang Shen *et al.* is the one that produces the best results, with the ability of explicitly setting the maximum value of $|w_{max}|$ being very useful. On the other hand, 2008 Wenchuan Cai *et al.* is found to be more difficult to modify and has worst results, but it is still robust and better than the original DCM.

- For the **detumbling scenario**, the source results reported in [1] were replicated exactly, but these results are only achieved when not limiting the angular momentum capacity of the RWs. Moreover, it has been demonstrated theoretically that the results reported in [1] are impossible, based on the principle of conservation of angular momentum.

- In the **detumbling**, when there is no RWs angular momentum capacity limitation, all three adaptive methods are shown to be much faster, take less control effort, and be more reliable than the original proposed. They have fairly similar performance, but again 2018 Qiang Shen *et al.* is remarkably consistent and stable, and 2008 Wenchuan Cai *et al.* has the worst results of the three. While the original was not robust to any failure, the adaptive

methods are shown to be robust to one and two failures, only suffering from minor degradation in performance.

- When **detumbling** with the RWs angular momentum capacity limited, the original controller does not manage to detumble, while the adaptive methods all nicely achieve detumbling. When there is a failure, there is not enough angular momentum left available in the wheels to achieve detumbling, but 2018 Qiang Shen *et al.* and 2008 Wenchuan Cai *et al.* manage to reduce the angular velocity to small values were it oscillates periodically.

- For the **long-term pointing**, the original is right in the limit of error tolerance when subjected to a worst-case disturbance scenario, and when there is a failure the errors grow outside the tolerance region. The three adaptive methods show much better levels of performance, with errors very far from the non-compliance, while using even less control than the original for this better results, and being almost not affected by failures.

As a summary, the adaptive methods have been proven better than the original controller proposed for the mission in all scenarios, making some crucial mission-saving differences in come cases when failures occur. Particularly, 2018 Qiang Shen *et al.* has shown an excellent performance across the board, while 2008 Wenchuan Cai *et al.* has consistently the worst results of the three in almost all circumstances. To confirm these findings, in the next chapter the methods will be tested in the second baseline, the AGILE mission spacecraft.

Chapter 4

Second baseline: the AGILE mission spacecraft

This chapter will be used to test the three adaptive methods selected into a second baseline, the AGILE mission [3], to confirm their applicability to missions of varying size and characteristics. The chapter opens with an overview of the relevant baseline data and the modifications made to the simulation environment. Then the methods are applied to the same three scenarios as in Chapter 3: slew manoeuvre, detumbling and long term-pointing, one in each subsection. Finally, the discoveries are summarized in the conclusions section.

4.1 Outline of the baseline

The AGILE (Astrorilevatore Gamma a Immagini, Leggero) mission is a mission of the Italian Space Agency (ASI) with the scientific objective of, among others, observing the gamma rays proceeding from outside the Solar System. It was based on the MITA (Italian Advanced Technology Minisatellite) platform, a low cost platform for small Low Earth Orbit (LEO) missions. Initially planned for launch in 2002, it finally was launched in 2007 with a cheaper design based on degraded requirements. The spacecraft is still in operation as of June 2021 in a LEO near-equatorial orbit.

As the mentor of this Thesis was involved in the design of the ACS for the mission, it has been a privilege to have access to the Phase B design document [3] of the ACS subsystem, concerning the initial non-degraded requirements for the 2002 launch date. Now the adaptive methods selected will be applied onto this baseline, mainly to test their capacity to accommodate to different missions and verify the good performance obtained in Chapter 3.

The spacecraft has an inertia matrix in the packed configuration, correspond-

ing to the detumbling phase, of

$$\mathbf{I}_{pack} = \begin{bmatrix} 27 & 0 & 0 \\ 0 & 17 & 0 \\ 0 & 0 & 25 \end{bmatrix} kg\, m^2 \tag{4.1}$$

and for the rest of the mission phases, the deployed configuration inertia is

$$\mathbf{I}_{depl} = \begin{bmatrix} 29 & 0 & 0 \\ 0 & 23 & 0 \\ 0 & 0 & 31 \end{bmatrix} kg\, m^2 \tag{4.2}$$

The document [3] is lacking some data about the RWs which was additionally provided by its author. The RWs maximum control torque is given as

$$\dot{h_{r}}_{max} = 0.02 \quad [N\, m] \tag{4.3}$$

the RWs inertia is

$$I_{RW} = 0.011 \quad [kg\, m^2] \tag{4.4}$$

and the maximum wheel velocity is

$$\dot{\varphi}_{max} = 400 \quad [rad/s] \tag{4.5}$$

which gives a RWs angular momentum capacity of

$$h_{r\,max} = 4.4 \quad [N\, m\, s] \tag{4.6}$$

The RWs distribution matrix is not reported, so it has been left the same as in the LUMIO baseline:

$$\mathbf{R} = \begin{bmatrix} -1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} \\ -1/\sqrt{3} & -1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \end{bmatrix} \tag{4.7}$$

It is mention that besides the four RWs, the mission has another three redundant magnetic coils for total angular momentum management and RWs desaturation. These are not going to be included in the simulations and will not be used.

As of sensors, in [3] it is mentioned that one of the requirements of the ACS is ensuring a attitude determination precision of 0.02°. Because sensor precision is not reported, it has been then assumed that this is the measurement precision of the attitude in all three axes. The angular velocity measurements' precision is not referenced either, so it has been left with the same value of 13 $arcseconds/s$ than with the LUMIO baseline on all three axes. The method of implementation of this measurement errors is the same than in Chapter 3.

This time the relevant requirements on the ACS derived from [3] are:

1) Steady state maximum angular velocity error of $w_{i,max} = 0.1\, arcsec/s$.

2) Steady state maximum absolute attitude pointing error of $\theta_{i,max} = 0.5\, deg$.

3) Slew manoeuvre total time of less than 1 day.

As no requirements have been imposed regarding the detumbling nor the maximum angular velocity during slew manoeuvres, the following requirements have been additionally taken, some of them the same as for the LUMIO baseline:

4) Detumbling maximum starting angular velocity value $w_{i,max}(0) = 10\, deg/s$.

5) Detumbling maximum final angular velocity value $w_{i,max}(t_{end}) = 0.1\, deg/s$.

6) Maximum angular velocity during slew manoeuvres of $w_{i,max} = 0.5\, deg/s$.

The scenarios to test the methods on are selected the same as in Chapter 3 for maximum comparative power. This time there is no data available on an original controller that allows for an implementation to compare it with the adaptive techniques. [3] explains in detail the structure of the originally proposed controller: the RWs are mediated by a intermediary Proportional Integral (PI) controller, and then the whole dynamics are under a PD controller, which is selected using optimal Linear Quadratic (LQ) control theory applied to the linearised dynamics of the spacecraft. But as the parameters are not reported, implementation is not possible.

About disturbance, the data provided allows for the implementation of models. The orbit data is given in [3], with altitude $h = 550\, km$, nominal inclination $i = 0$, and eccentricity $e = 0$. The spacecraft dimensions are reported as $917 \times 988 \times 871\, mm$, and an illustration of the spacecraft is provided. With this, the following values have been defined that are necessary for the models.

The ten spacecraft surfaces normals, indicated column-wise by three-dimensional vectors, are

$$normals = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

The surfaces centre distances to the CoM are

$$normals = \begin{bmatrix} 91/2 & 0 & 0 & -91/2 & 0 & 0 \\ 0 & 98/2 & 0 & 0 & -98/2 & 0 \\ 0 & 0 & 87/2 & 0 & 0 & -87/2 \end{bmatrix} \begin{matrix} 130.5 & -130.5 & 130.5 & -130.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \; cm$$

And the surfaces areas are

$$Areas = [852,\ 792,\ 892,\ 853,\ 792,\ 892,\ 1583,\ 1583,\ 1583,\ 1583]\ cm^2 \tag{4.9}$$

Finally, the surfaces diffusivity (ρ_d) and absorptivity (ρ_d) factors have been chosen as

$$\rho_s = [0.5,\ 0.5,\ 0.5,\ 0.5,\ 0.5,\ 0.5,\ 0.8,\ 0.8,\ 0.8,\ 0.8] \tag{4.10}$$

$$\rho_d = [0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1] \tag{4.11}$$

The modifications done to the Simulink models are shown in Figure 4.1, where the blocks showed substitute the earlier disturbance block implemented. In yellow, the "Spacecraft Orbit" block outputs the inertial position vector r_I along the orbit and its norm R. Also in yellow, the "Sun Attitude" block outputs the direction if the Sun S_I.

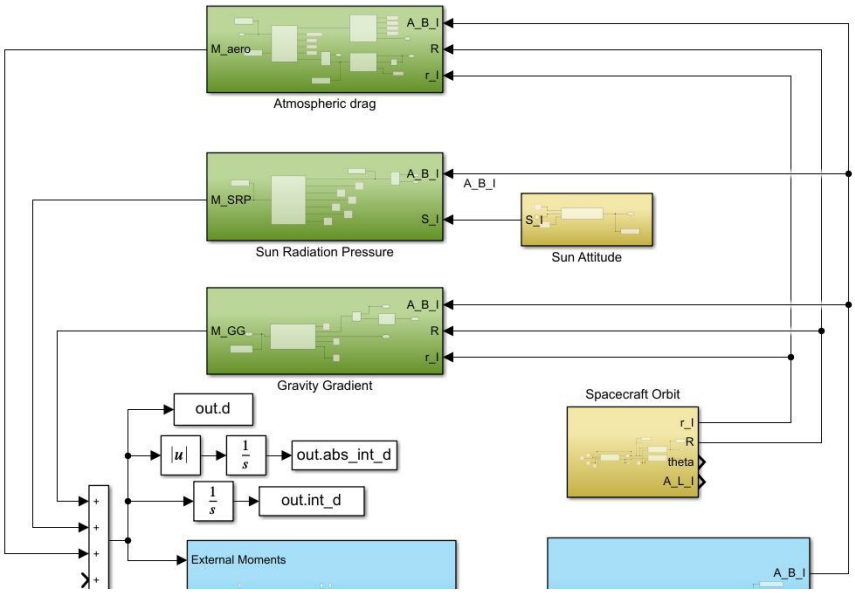


Figure 4.1: Simulink models modification for implementing disturbance models

The green blocks implement the disturbance models selected. They have been extracted from [36], are as the following.

The gravity gradient torque M_{GG} is

$$M_{GG} = \frac{3GM_t}{R^3} \begin{Bmatrix} (I_z - I_y)c_2c_3 \\ (I_x - I_z)c_1c_3 \\ (I_y - I_x)c_1c_2 \end{Bmatrix} \tag{4.12}$$

with c_1, c_2 and c_3 being the components of the unitary position vector from the center of the Earth, in body frame: r_B/R.

The aerodynamic torque is

$$M_{aero} = -\frac{1}{2}\rho C_D \sum_{i=1}^{n} \mathbf{r_i^B} \times \mathbf{v_{rel}^B}(\mathbf{N_i^B} \cdot \mathbf{v_{rel}^B}) A_i \tag{4.13}$$

if $\mathbf{N_i^B} \cdot \mathbf{v_{rel}^B}$ is positive, and zero otherwise. $\mathbf{N_i^B}$ are the normals to the panels, $\mathbf{r_i^B}$ their distance to the CoM, and A_i the panels areas. C_D has been selected as 2.2, the equivalent to a flat plate. A model of the atmosphere has been implemented to compute the density ρ.

Finally, the Sun radiation torque is modelled as

$$M_{SRP} = \sum_{i=1}^{n} \mathbf{r_i} \times (-P A_i (\mathbf{S^B} \cdot \mathbf{N_i^B}) \left[(1-\rho_S)\mathbf{S^B} + (2\rho_S(\mathbf{S^B} \cdot \mathbf{N_i^B}) + \frac{2}{3}\rho_d)\mathbf{N_i^B} \right]) \tag{4.14}$$

if $\mathbf{S^B} \cdot \mathbf{N_i^B}$ is positive, and zero otherwise. $\mathbf{S^B}$ is the Sun direction, P the SRP on Earth equal to $4.53 \times 10^{-6}\, N/m^2$, and the effect from Earth's albedo is neglected.

4.2 First scenario: Slew manoeuvre

For the sake of consistency, the slew manoeuvre has been selected to be from the same initial attitude than in Chapter 3, given by the quaternion

$$\mathbf{q}(0) = [0.4, \quad -0.3, \quad -0.5]^T \tag{4.15}$$

with the target attitude being $\mathbf{q}(0) = [0, 0, 0]^T$.

4.2.1 2018 Qiang Shen *et al.*

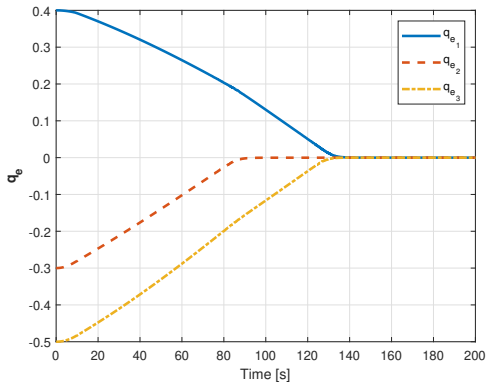
No failure. The situation for $k = 100$ and no failure is essentially the same than for the LUMIO baseline. Besides the obvious discrepancy on control used, the angular velocity reached and time of the manoeuvre are left unchanged, and the response is adequate as seen in Figure 4.2. However, for $k = 1$ and $k = 0.1$ the spacecraft experiences heavy oscillations, as seen in Figure 4.3 and table 4.1. This probably has to do with a different balance between the control torque available and the magnitude of the spacecraft's inertia, modifying the acceptable range for the parameter k.

1 RW failure. The effect of a single failure is slightly more noticeable than for LUMIO, with a larger performance degradation, but still the response reaches the objective and the angular velocity is most of the times within compliance, as shown in Figure 4.4, although not always.

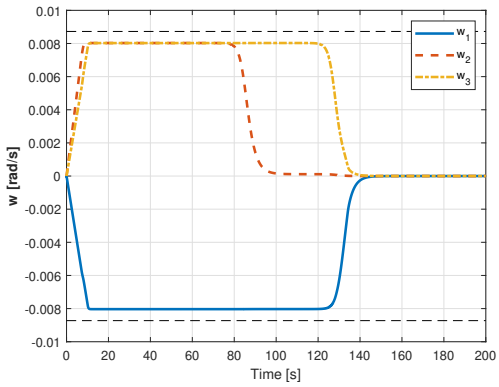
2 RWs failures. Although the manoeuvre cannot be completed due to the under-actuation, the response is stable.

Failures	k	**Success**	**Time to target [s]**	**Total control used [Nms]**	**Final** $\sum \lvert h_{r_i} \rvert$ **[Nms]** $\times 10^{-2}$	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	1	Yes	286	2.769	3	0.83
	100	Yes	133	1.65	32	0.80
1	100	Yes	132 to 160	1.74 to 3.29	0.2 to 0.3	8.3 to 8.6
2	100	No, stable	N/A	0.64 to 1.49	0.0 to 0.8	0.80 to 0.92

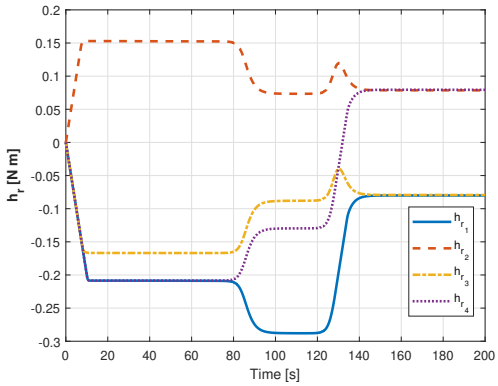
Table 4.1: Results summary for slew manoeuvre with 2018 Qiang Shen *et al.* controller



(a) Attitude errors

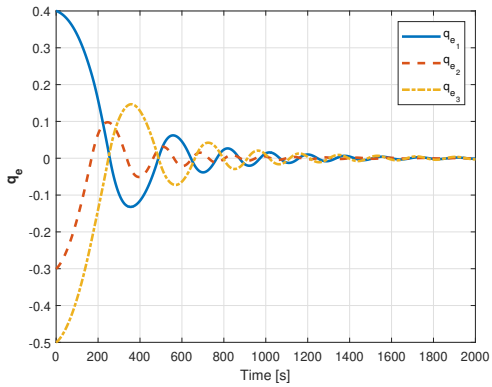


(b) Angular velocity errors

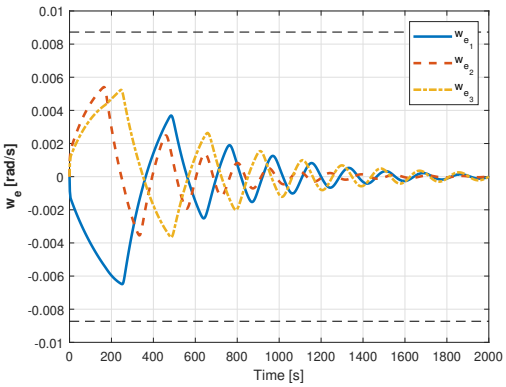


(c) RWs angular momentum

Figure 4.2: Results for slew manoeuvre with no failures: 2018 Qiang Shen *et al.* controller ($k = 100$)

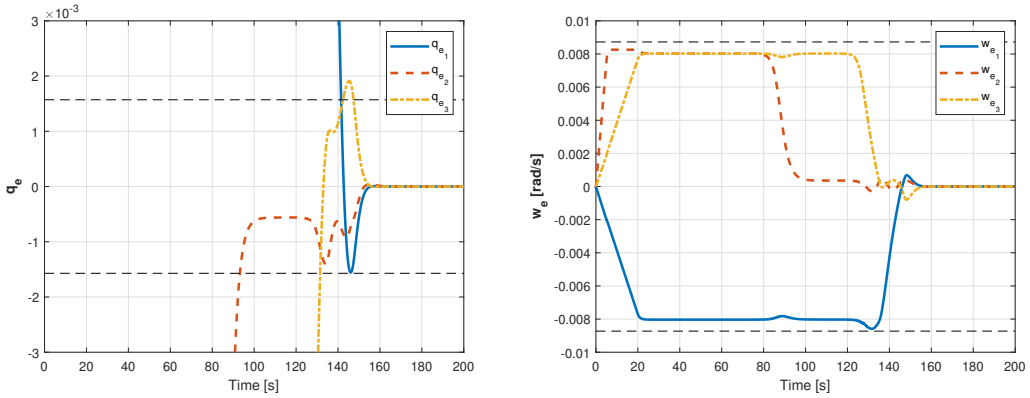


(a) Attitude errors



(b) Angular velocity errors

Figure 4.3: Results for slew manoeuvre with no failures: 2018 Qiang Shen *et al.* controller ($k = 0.1$)

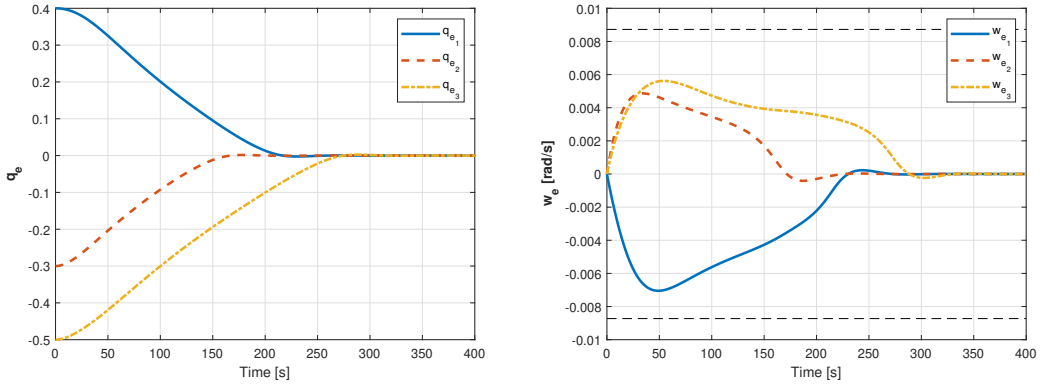


(a) Attitude errors (zoomed) (b) Angular velocity errors

Figure 4.4: Results for slew manoeuvre with 1 RW failure: 2018 Qiang Shen *et al.* controller ($k = 100$)

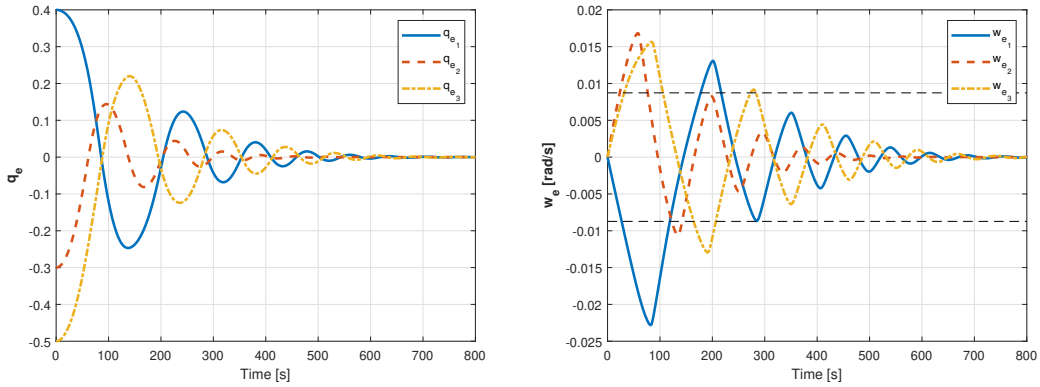
4.2.2 2014 Danyal Bustan *et al.*

No failure. Now, according to the baseline, the parameter u_{max} has been changed to 0.02. The behaviour with $\rho = 10$ is satisfactory as showed on Figure 4.5, although slower than for 2018 Qiang Shen *et al.*. However, when trying to increase the speed of the response reducing ρ to 5 or 1, the small overshooting present in Figure 4.5 increase dramatically, as shown in Figure 4.6. Finally for $\rho = 0.1$, an instability is developed. On the other hand, when increasing the value of ρ to 50, no shattering effect is observed unlike in the LUMIO baseline. See table 4.2.



(a) Attitude errors (b) Angular velocity errors

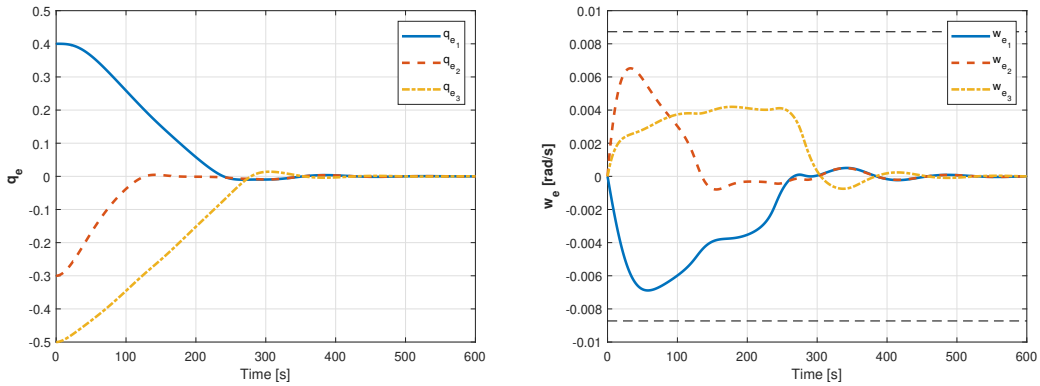
Figure 4.5: Results for slew manoeuvre with no failures: 2014 Danyal Bustan *et al.* controller ($\rho = 10$)



(a) Attitude errors (b) Angular velocity errors

Figure 4.6: Results for slew manoeuvre with no failures: 2014 Danyal Bustan *et al.* controller ($\rho = 1$)

1 RW failure. When there is a failure the response is degraded as shown in Figure 4.7, but the objective is still satisfactorily achieved.



(a) Attitude errors (b) Angular velocity errors

Figure 4.7: Results for slew manoeuvre with 1 RW failure: 2014 Danyal Bustan *et al.* controller ($\rho = 10$)

2 RWs failures. The response is as expected, not able to reach the objective, but is stable.

4.2.3 2008 Wenchuan Cai *et al.*

As was explored in Chapter 3, the only parameters that have a sensible effect are found to be F_{max} and β. According to the subsystem actuation capacity, F_{max} has been changed to 0.02.

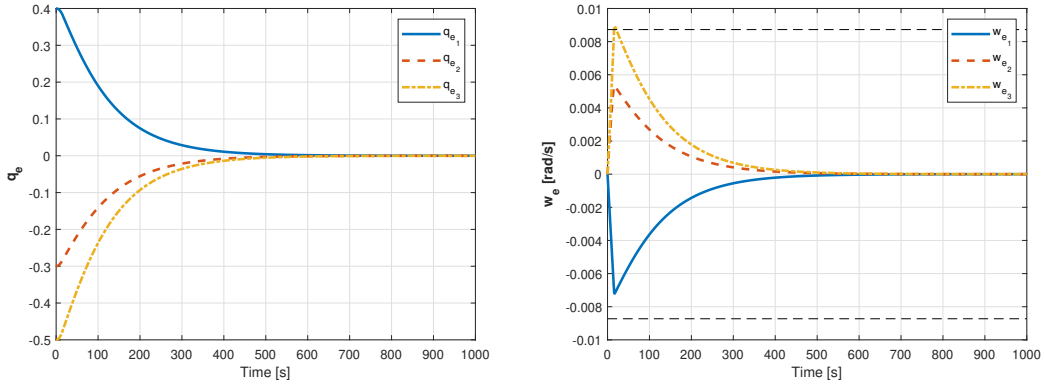
Failures	ρ	**Success**	**Time to target [s]**	**Total control used [Nms]**	**Final $\sum \lvert h_{r_i} \rvert$ [Nms] $\times 10^{-2}$**	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	0	No, unstable	N/A	N/A	N/A	2.8
	1	Yes	618	8.6	1.1	2.8
	5	Yes	231	2.3	1.1	1.2
	10	Yes	267	1.1	1.2	0.7
	50	Yes	1281	0.29	1.6	0.2
1	10	Yes	333 to 389	1.1 to 1.6	1.7 to 3.2	0.6 to 0.7
	20	Yes	525 to 531	0.54 to 0.85	0.7 to 0.9	0.34 to 0.4
2	10	No, stable	N/A	0.71 to 0.92	0.3 to 1.2	0.4 to 0.8

Table 4.2: Results summary for slew manoeuvre with 2014 Danyal Bustan *et al.* controller

No failure. The parameter β regulates the aggressiveness of the response, but now the range of values has changed. With $\beta = 0.019$ the response is just compliant with the angular velocity limitation as seen in Figure 4.8, but this takes over 500 seconds to converge. Increasing the value of β to increase the speed makes it go outside the allowed angular velocity, and it is effective until $\beta = 0.091$, see table 4.3. At this point, increasing β further degrades the response due to the appearance of oscillations, as seen in Figure 4.9.

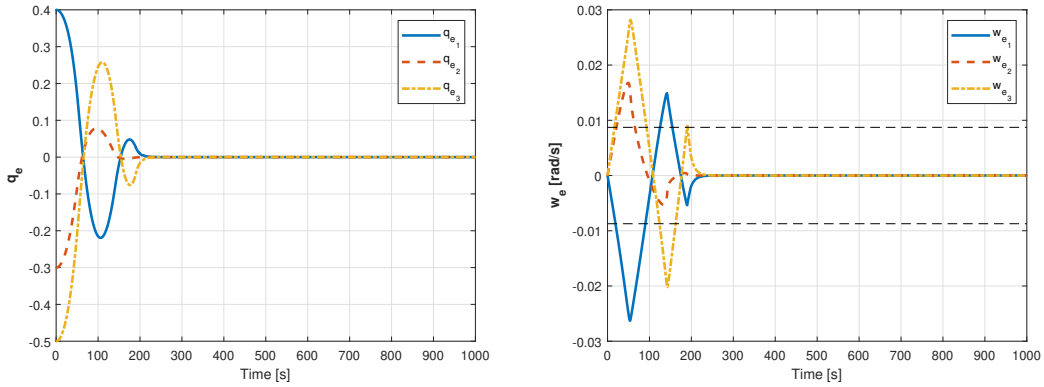
1 RW failure. The performance with one failure is slightly degraded on time and effort used with respect to no failure, but still achieves the objective and respects the angular velocity limit, as shown in Figure 4.10.

2 RWs failures. The response does not reach the objective due to the underactuation but remains stable.



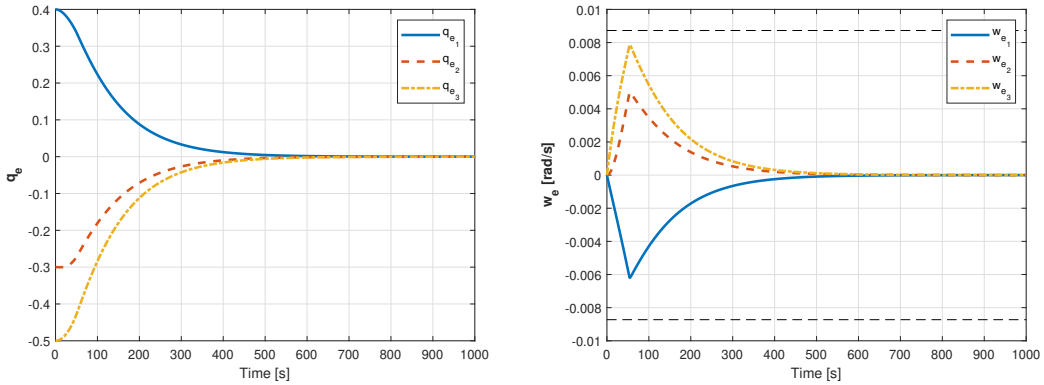
(a) Attitude errors (b) Angular velocity errors

Figure 4.8: Results for slew manoeuvre with no failures: 2008 Wenchuan Cai *et al.* controller ($\beta = 0.019$)



(a) Attitude errors (b) Angular velocity errors

Figure 4.9: Results for slew manoeuvre with no failures: 2008 Wenchuan Cai *et al.* controller ($\beta = 0.2$)



(a) Attitude errors

(b) Angular velocity errors

Figure 4.10: Results for slew manoeuvre with 1 RW failure: 2008 Wenchuan Cai *et al.* controller ($\beta = 0.019$)

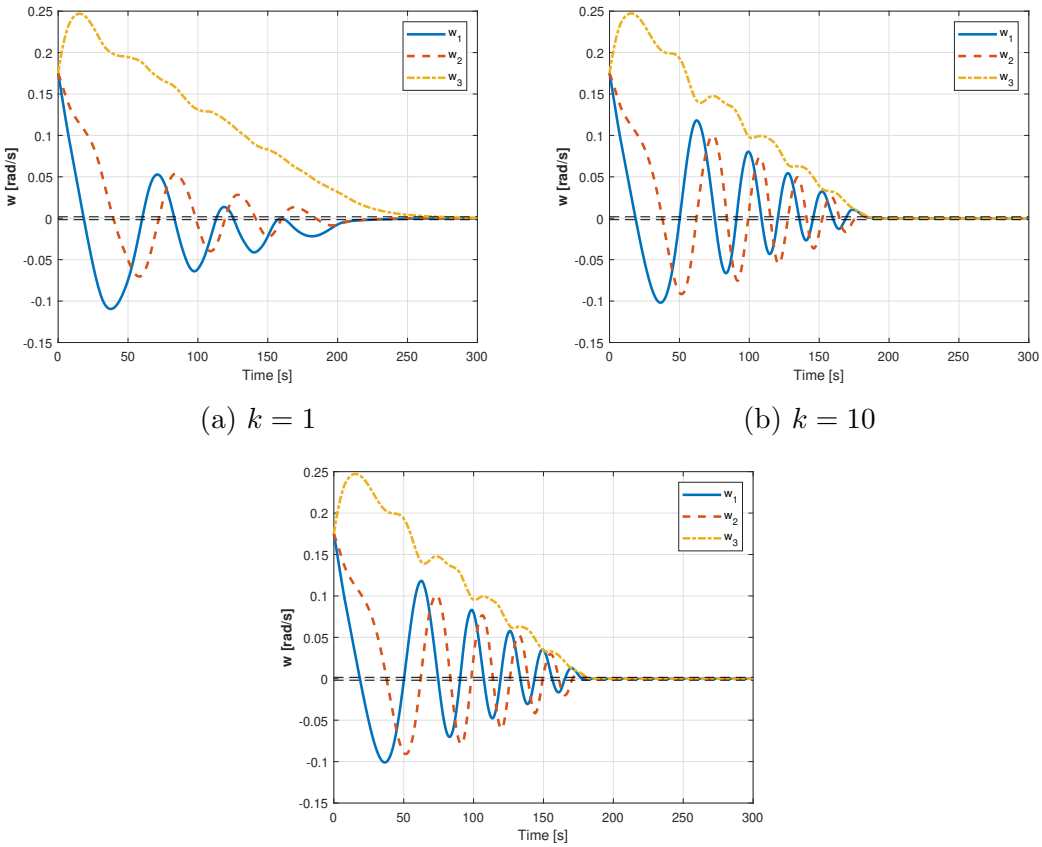
Failures	β	**Success**	**Time to target [s]**	**Total control used [Nms]**	**Final $\sum \lvert h_{r_i} \rvert$ [Nms] $\times 10^{-2}$**	$\lvert w_{i,max} \rvert$ **[rad/s]** $\times 10^{-2}$
0	0.019	Yes	517	1.06	1.5	0.89
	0.05	Yes	214	2.24	1.4	1.9
	0.091	Yes	97	2.95	1.4	2.4
	0.2	Yes	213	6.8	1.3	2.8
1	0.019	Yes	517 to 532	0.98 to 1.86	1.5 to 1.9	0.77 to 0.90
2	0.019	No, stable	N/A	0.45 to 0.9	0.6 to 1.5	0.46 to 0.81

Table 4.3: Results summary for slew manoeuvre with 2008 Wenchuan Cai *et al.* controller

4.3 Second scenario: Detumbling

4.3.1 2018 Qiang Shen *et al.*

No failures. It is seen that for a value of $k = 10$ the control saturates and no improvements occur increasing k further to, for instance, 100. Reducing k below 10 degrades the time and control effort used. These results are shown on Figure 4.11 and table 4.4.

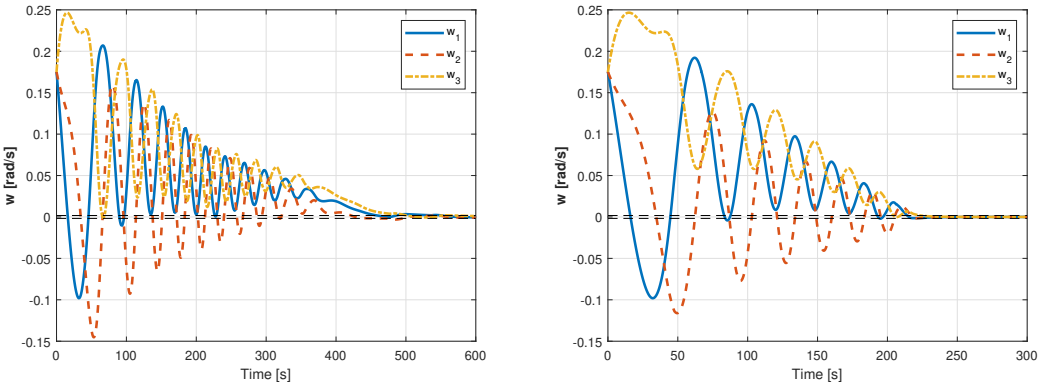


(a) $k = 1$ (b) $k = 10$

(c) $k = 100$

Figure 4.11: Angular velocity w_i results for detumbling manoeuvre with no failures: 2018 Qiang Shen *et al.* controller

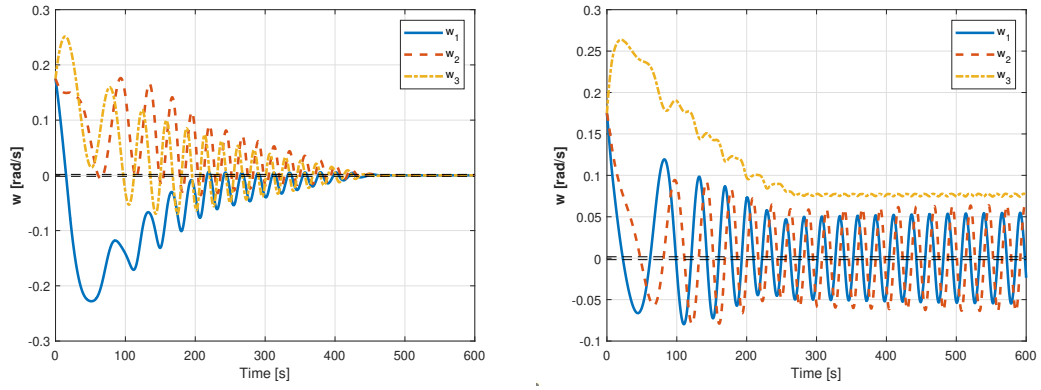
1 RW failure. The result with one failure keep being satisfactory, and, as shown in Figure 4.12, the value of $k = 10$ is still the optimum. Thus, this settles the question that was left from the LUMIO baseline: when there are failures in detumbling the 2018 Qiang Shen *et al.* method achieves it even with limited RWs angular momentum capacity, as long as the total momentum available is enough to actually detumble the spacecraft.



(a) $k = 1$ (b) $k = 10$

Figure 4.12: Angular velocity w_i results for detumbling manoeuvre with 1 RW failure: 2018 Qiang Shen *et al.* controller

2 RWs failures. When there are two failures most of the time the detumbling is achieved in a degraded response. However, there is a particularly harsh failure combination for which, for some reason, the method does not achieve detumbling, shown in Figure 4.13. Confusingly, the combination that causes the no detumbling changes with the value of k, and no single value was found that worked for all failure combinations.



(a) Failures on the 1st and 3rd RWs (b) Failures on the 2nd and 4th RWs

Figure 4.13: Angular velocity w_i results for detumbling manoeuvre with 2 RWs failures: 2018 Qiang Shen *et al.* controller ($k = 10$)

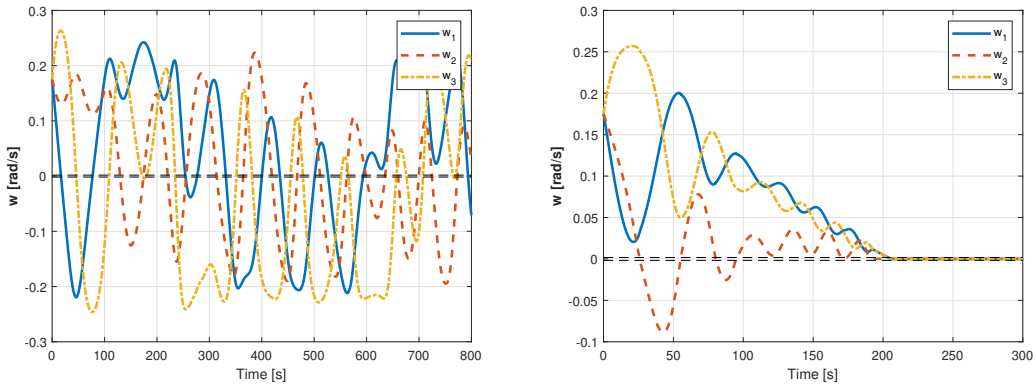
4.3.2 2014 Danyal Bustan *et al.*

No failures. The only parameter that makes a difference in the controller is u_{max}. Contrary to what could be thought, a value of $u_{max} = 0.02$ as the system

Failures	k	Success	Time to target [s]	Total $\sum h_{r_i}$ used [Nms]	Final $\sum h_{r_i}$ [Nms]
0	1	Yes	270	15.7	12.1
	10	Yes	185	14.5	12.0
	100	Yes	181	14.5	12.1
1	10	Yes	221 to 237	13.1 to 13.9	9.6 to 10.5
	100	Yes	214 to 221	12.9 to 14.3	9.3 to 10.7
2	10	Not always	408 to ∞	14.2 to ∞	8.7 to 8.8
	100	Not always	380 to ∞	15.1 to ∞	8.7 to 8.8

Table 4.4: Results summary for detumbling with 2018 Qiang Shen *et al.* controller

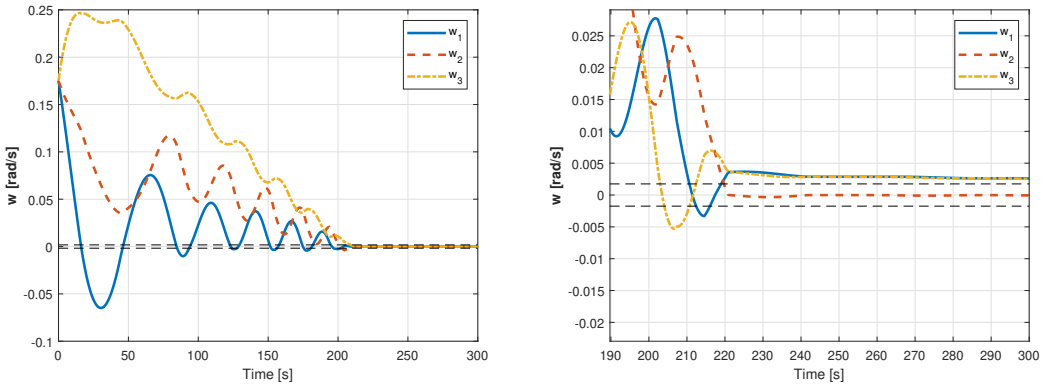
architecture could suggest does not achieve detumbling, as seen in Figure 4.14. Instead, it has been investigated that a value of $u_{max} = 2$ that allows for the saturation of the control effort obtains the best results. A further increase of u_{max} beyond the value 2 only starts to degrade the performance, as shown in table 4.5.



(a) $u_{max} = 0.02$ (b) $u_{max} = 2$

Figure 4.14: Angular velocity w_i results for detumbling manoeuvre with no failures: 2014 Danyal Bustan *et al.* controller

1 RW failure. When there is a failure the controller with $u_{max} = 2$ achieves detumbling with only slight degradation of performance. This settles the question left in Chapter 3: the 2014 Danyal Bustan *et al.* method is able to achieve detumbling with both failures and limitation of RWs angular momentum capacity simultaneously. Increasing the value of u_{max} to 20 does not alter much the response with the exception that in one case the angular velocity does not reach the required minimum value, shown in Figure 4.15.



(a) $u_{max} = 2$ (b) (zoomed) $u_{max} = 20$

Figure 4.15: Angular velocity w_i results for detumbling manoeuvre with 1 RW failure: 2014 Danyal Bustan *et al.* controller

2 RWs failures. The controller with $u_{max} = 2$ has an even more degraded performance but still achieves detumbling, as shown in table 4.5.

Failures	u_{max}	**Success**	**Time to target [s]**	**Total $\sum \dot{h}_{r_i}$ used [Nms]**	**Final $\sum h_{r_i}$ [Nms]**
0	0.02	No, unstable	N/A	N/A	N/A
	2	Yes	195	15.5	12.7
	20	Yes	203	16.2	12.2
1	2	Yes	214 to 289	12.8 to 21.1	9.7 to 10.6
	20	Not always	208 to ∞	12.6 to 15.5	9.6 to 10.6
2	2	Yes	389 to 694	14.5 to 22.7	8.7

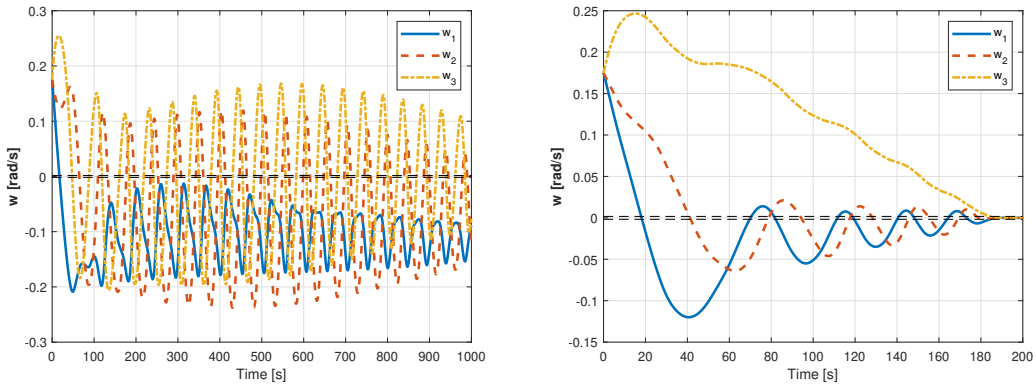
Table 4.5: Results summary for detumbling with 2014 Danyal Bustan *et al.* controller

4.3.3 2008 Wenchuan Cai *et al.*

As seen in subsection 3.3.4, the parameter β cannot have an effect for detumbling as it is multiplied by zero. Modifications on parameters k_0, σ_1 and σ_2 have been found to have almost no effect. The only parameter that affects the response is F_{max}.

No failure. Contrary to what could be though from the subsystem specifications, the value $F_{max} = 0.02$ results in no detumbling achieved, shown in Figure

4.16. Just like in subsection 4.3.2, higher values obtain better responses, although no improvement is made after $F_{max} = 0.2$. See table 4.6.



(a) $F_{max} = 0.02$ (b) $F_{max} = 0.2$

Figure 4.16: Angular velocity w_i results for detumbling manoeuvre with no failures: 2008 Wenchuan Cai *et al.* controller

1 RW failure. When there is a failure, both for $F_{max} = 0.2$ and $F_{max} = 2$ the detumbling is achieved in a degraded response as seen in Figure 4.17, although for the case of $F_{max} = 2$ there are some instances in which it takes a long time to settle inside the requirement region.

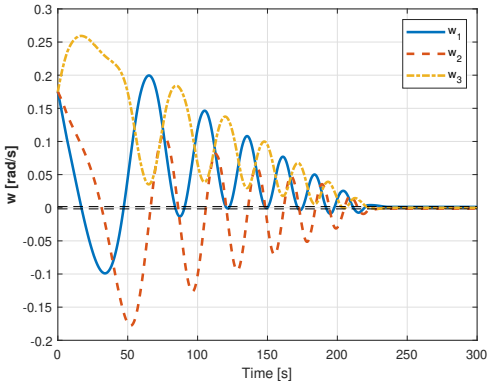
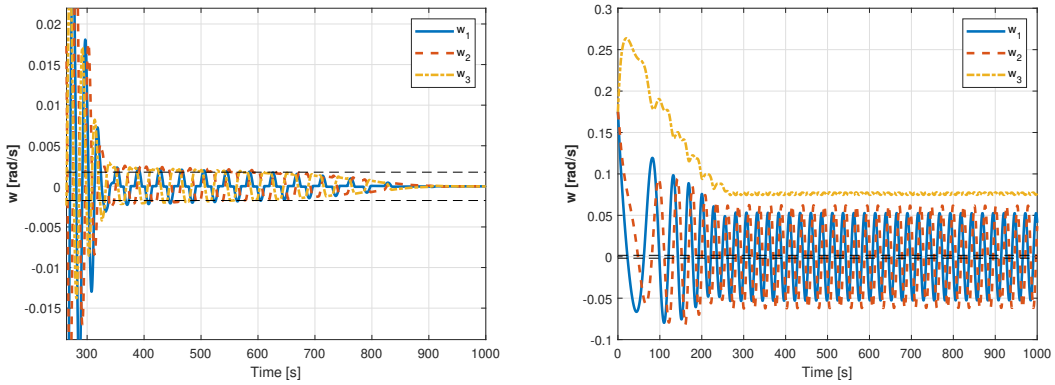


Figure 4.17: Angular velocity w_i results for detumbling manoeuvre with 1 RW failure: 2008 Wenchuan Cai *et al.* controller ($F_{max} = 2$)

2 RWs failures. When there are two failures it has been found no value of F_{max} that achieve the detumbling for all cases. Looking at table 4.6, the values 0.2, 2 and 20 all manage the detumbling in the majority of cases. However for specific

combinations of failures detumbling is not managed, like in Figure 4.18, and the combination of failures affected change for each value of F_{max}. This is the same behaviour than for 2018 Qiang Shen *et al.*.



(a) Failures in the 1st and 2nd RWs (b) Failures in the 2nd and 4th RWs

Figure 4.18: Angular velocity w_i results for detumbling manoeuvre with 2 RWs failures: 2008 Wenchuan Cai *et al.* controller ($F_{max} = 2$)

Failures	F_{max}	**Success**	**Time to target [s]**	**Total $\sum h_{r_i}$ used [Nms]**	**Final $\sum h_{r_i}$ [Nms]**
0	0.02	No, unstable	N/A	N/A	N/A
	0.2	Yes	184	14.1	12.0
	2	Yes	181	14.5	12.0
1	0.2	Yes	238 to 365	13.9 to 18.5	10.2 to 10.6
	2	Yes	214 to 649	12.9 to 24.9	10.0 to 11.5
2	0.2	Not always	492 to ∞	17.8 to ∞	8.7
	2	Not always	508 to ∞	11.4 to ∞	8.7 to 8.8
	20	Not always	686 to ∞	20.8 to ∞	8.7

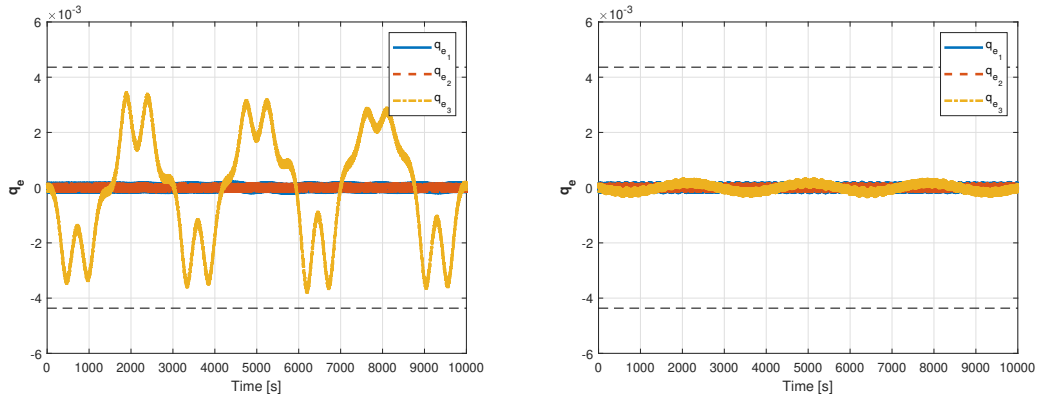
Table 4.6: Results summary for detumbling with 2008 Wenchuan Cai *et al.* controller

4.4 Third scenario: Long-term pointing

The same procedure for reporting results will be followed as in Chapter 3, performing the simulations for a time of 10000 seconds. Because the mission now is to observe cosmic gamma rays, an inertial pointing has been chosen as the desired attitude to track.

4.4.1 2018 Qiang Shen *et al.*

No failures. There is a balance between the value of parameter k and the magnitude of error quaternion reached. Decreasing the value of k below 0.01 does not have any effect on reducing the control effort used, only increases the errors. Higher values of k like 0.1 decrease the errors reached but increase the control used as shown in Figure 4.19 and in table 4.7.



(a) $k = 0.001$ (b) $k = 0.1$

Figure 4.19: Attitude error results $\mathbf{q}_e$ for long-term pointing with no failures: 2018 Qiang Shen *et al.* controller

1 RW failure. The response gets degraded but stays well inside the tolerance region, as shown in Figure 4.20.

2 RWs failures. Because the spacecraft is under-actuated, a drift in attitude slowly but inevitably develops, although long-term the controller manages to not be totally unstable, keeping the attitude close to the desired one.

4.4.2 2014 Danyal Bustan *et al.*

No failures. The parameter ρ has been found to have no effect, which is understandable as it is supposed to mediate the transient behaviour, and now there

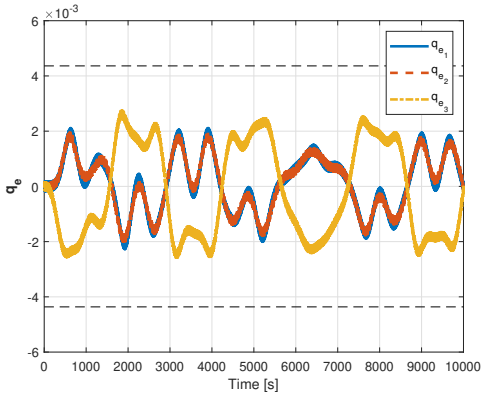
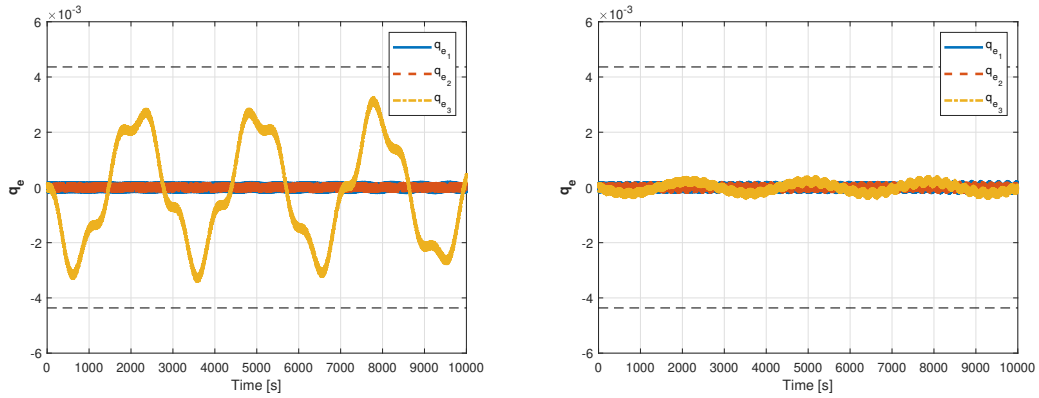


Figure 4.20: Attitude error results $\mathbf{q}_e$ for long-term pointing with 1 RW failure: 2018 Qiang Shen *et al.* controller ($k = 0.01$)

Failures	k	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-2}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-2}$
0	0.001	Yes	3.5	12.3	1.7
	0.01	Yes	1.9	12.7	1.8
	0.1	Yes	0.36	14.7	1.7
1	0.01	Yes	2.6 to 2.8	16.4	1.9 to 2.0
2	0.01	No, unstable	N/A	N/A	N/A

Table 4.7: Results summary for long-term pointing with 2018 Qiang Shen *et al.* controller

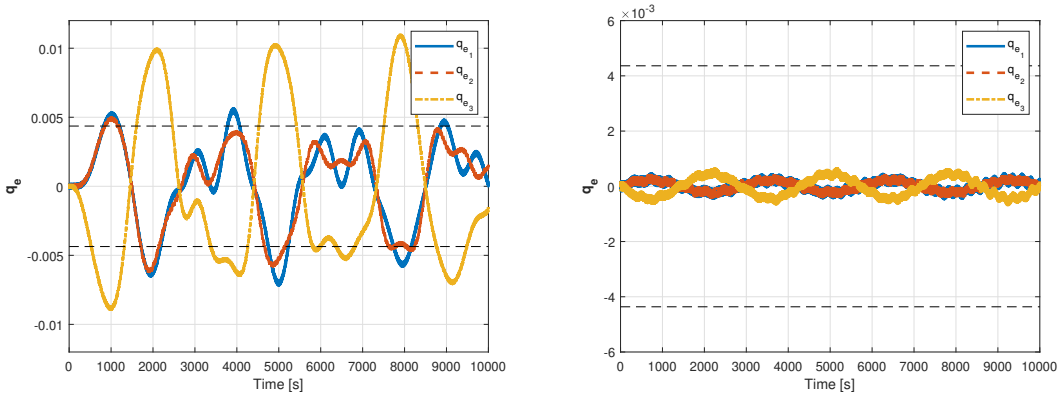
is no transient. The only parameter affecting the simulation is u_{max}, which for higher values decreases the attitude errors but increases the control used. See Figure 4.21 and table 4.8.



(a) $u_{max} = 2 \times 10^{-4}$ (b) $u_{max} = 2 \times 10^{-3}$

Figure 4.21: Attitude error results $\mathbf{q}_e$ for long-term pointing with no failures: 2014 Danyal Bustan *et al.* controller

1 RW failure. When there is a failure the controller with $u_{max} = 2 \times 10^{-3}$ is degraded but still remains well within compliance, while for $u_{max} = 2 \times 10^{-4}$ the attitude errors go outside the tolerance region.



(a) $u_{max} = 2 \times 10^{-4}$ (b) $u_{max} = 2 \times 10^{-3}$

Figure 4.22: Attitude error results $\mathbf{q}_e$ for long-term pointing with 1 RW failure: 2014 Danyal Bustan *et al.* controller

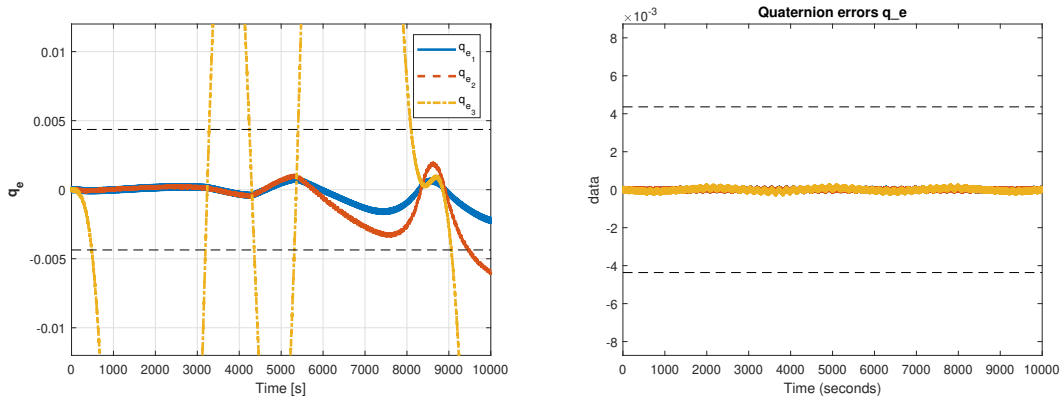
2 RWs failures. The same behaviour than for 2018 Qiang Shen *et al.* is developed, a drift in attitude but that keeps it close to the desired pointing.

Failures	ρ	u_{max} $\times 2$	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h}_{r_i}$ **used [Nms]** $\times 10^{-2}$	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-2}$
0	0	10^{-3}	Yes	0.34	16.4	1.7
	1	10^{-3}	Yes	0.34	16.4	1.7
	10	10^{-3}	Yes	0.34	16.4	1.7
	0	10^{-4}	Yes	3.3	12.9	1.8
1	0	10^{-3}	Yes	0.67 to 0.68	14.5 to 18.2	1.7
	0	10^{-4}	No, stable	10.5 to 11.0	18.2 to 19.2	1.7 to 1.9
2	0	10^{-3}	No, unstable	N/A	N/A	N/A

Table 4.8: Results summary for long-term pointing with 2014 Danyal Bustan *et al.* controller

4.4.3 2008 Wenchuan Cai *et al.*

No failures. This methods has been the hardest to adapt to the baseline. As seen in table 4.9, at the value of $F_{max} = 2 \times 10^{-3}$ the values of the attitude error reached a minimum or the same magnitude as the measurement errors, but for that set up the control used is enormous. Reducing F_{max} to 2×10^{-5} is the limit for the less control used by this method. When reducing F_{max} further to 2×10^{-6}, the attitude errors get outside of the tolerance region.



(a) $F_{max} = 2 \times 10^{-6}$ (b) $F_{max} = 2 \times 10^{-5}$

Figure 4.23: Attitude error results $\mathbf{q}_e$ for long-term pointing with no failures: 2008 Wenchuan Cai *et al.* controller

1 RW failure. The effect of having a failure is that the limit for which the attitude errors get outside of the accepted region increases to $F_{max} = 10^{-5}$.

2 RWs failures. The response compared to the other controllers is the same, a drift that keeps it close but out of the compliance attitude error region.

Failures	β	F_{max} $\times 2$	**Success**	**Maximum quaternion error** $\lvert q_{e_{i,max}} \rvert$ $\times 10^{-3}$	**Total** $\sum \dot{h_{r_i}}$ **used [Nms]**	**Final** $\sum h_{r_i}$ **[Nms]** $\times 10^{-2}$
0	2	10^{-2}	Yes	0.12	111	1.5
	2	10^{-3}	Yes	0.07	33	1.7
	2	10^{-4}	Yes	0.2	3.4	1.7
	2	10^{-5}	Yes	0.34	0.34	1.7
	2	10^{-6}	No, unstable	N/A	N/A	N/A
1	2	10^{-4}	Yes	0.23 to 0.25	1.82 to 2.78	1.7 to 1.8
	2	10^{-5}	No, unstable	N/A	N/A	N/A
2	2	10^{-5}	No, unstable	N/A	N/A	N/A

Table 4.9: Results summary for long-term pointing with 2008 Wenchuan Cai *et al.* controller

4.5 Chapter conclusions

In this chapter the AGILE mission baseline described in [3] has been simulated in the same three scenarios than in Chapter 3: slew manoeuvre, detumbling and long-term pointing. The discoveries are summarized in the following points:

- For the slew manoeuvre, the three methods can easily manage the manoeuvre with no failures and are robust to one failure. 2018 Qiang Shen *et al.* has the best response of the three, while 2008 Wenchuan Cai *et al.* offers the worst, more time-consuming response.
- With 2 failures, the slew manoeuvre is under-actuated and no method can reach the target, but in all cases their response is stable and settles after a short time.
- For detumbling, all three methods have been successful for the no-failure case and are robust to one failure. Among them, 2018 Qiang Shen *et al.* and 2008 Wenchuan Cai *et al.* show the best response, being very similar.
- When detumbling with 2 failures, the margin between angular momentum available and the required for the manoeuvre is very slim. 2014 Danyal

Bustan *et al.* has shown to consistently achieve detumbling in this case, while the other two methods fail with some combinations of RWs failures.

- The long term pointing with no failure has the best results for 2018 Qiang Shen *et al.*, followed closely by 2014 Danyal Bustan *et al.*. All three are robust to one failure, but 2008 Wenchuan Cai *et al.* uses a lot more control effort than the other two.

- The response for long term pointing with two failures is always the same, a slow drift in attitude due to the under-actuation of the spacecraft, although the three methods keep the pointing near the desired target.

This chapter has confirmed that the methods can be adapted to multiple missions with spacecrafts of different sizes, and their strengths of performance and robustness to failures have been reinforced with more evidence.

Among the methods, 2018 Qiang Shen *et al.* has strong performance across the board and 2008 Wenchuan Cai *et al.* is the most difficult to adapt and has the worst results in the slew manoeuvre and, specially, in the long-term pointing.

Conclusions

The present Thesis has been dedicated to the field of robust control over spacecraft's attitude. Particularly, it has been investigated the typology of failures that the ACS is more likely to suffer when using COTS components. Two general groups were identified based on the work in [5]: a high-end COTS group that suffers mainly from transient SEU failures, and a low-end COTS group that is at increased risk of complete component failure. It was decided to focus on the second, low-end group.

To tackle this type of failures, a list of innovative techniques that show potential to outperform the current "classical techniques" used has been identified. Based on their complexity, implementability, and personal interest of the author, the Adaptive control and Model identification have been selected as the techniques to move forward with further study.

A literature review has been conducted on both selected techniques, discussing a moderately large number of sources more in depth in Chapter 1 of this Thesis. A criteria to evaluate these sources has been created based on completeness of source explanation, easiness of implementation and original results obtained. The outcome of the analysis was summarized in appendix A. This has allowed to shortlist four Adaptive control methods for implementation and testing, that were referred to as 2018 Qiang Shen *et al.*[18], 2014 Danyal Bustan *et al.*[17], 2008 Wenchuan Cai *et al.*[23], and 2011 Qinglei Hu *et al.*[22] throughout the rest of the document. The Model identification technique was not continued because no candidate resulted among the most promising.

Chapter 2 presented exhaustively the mathematical layout of the shortlisted methods, and proceeded to implement them on a Simulink model environment to replicate the original results. Three of the methods obtained successful results, those being 2018 Qiang Shen *et al.*, for whom the replication was perfect, and 2014 Danyal Bustan *et al.* and 2008 Wenchuan Cai *et al.*, that had small differences between what was obtained and the originally reported, although this did not compromise their successful results. The method of 2011 Qinglei Hu *et al.* did not achieve a stable result, and a explanation was proposed based on analysing its mathematical equations.

In Chapter 3, the three successfully implemented methods have been tested onto the LUMIO CubeSat baseline [1] for three scenarios: slew manoeuvre, detumbling and long-term pointing.

- For the slew manoeuvre, the adaptive methods are shown to be better than the original controller. They are faster and can be tuned to respect the limitation on angular velocity. On top of that, they are robust to one failure, only facing small degradation on performance, while the original becomes unstable. Particularly, 2018 Qiang Shen *et al.* is the best, followed by 2014 Danyal Bustan *et al.*

- For detumbling, it has been shown in simulations and mathematically proven that the original results shown in [1] are only possible when the angular momentum capacity of the RWs is not limited. For this case, the adaptive controllers are faster, take less control effort, and are robust to one and two failures. The original is not robust to one failure and becomes unstable. When limiting the RWs angular momentum capacity, the original cannot achieve detumbling, while the three adaptive methods can perform it without problems.

- For long-term pointing, the original gets just to the attitude error limit, and with a failure goes out of the tolerance. The adaptive methods are well inside the attitude tolerance region and are minimally affected by one failure. 2008 Wenchuan Cai *et al.* is proven the hardest one to adapt.

In Chapter 4 the adaptive methods are applied to the AGILE mission baseline [3] for the same three scenarios.

- For the slew manoeuvre, the three can achieve it with one or no failures, with again 2018 Qiang Shen *et al.* being the best, followed by 2014 Danyal Bustan *et al.*

- For detumbling, all methods can manage it with zero and one failure, but only 2014 Danyal Bustan *et al.* can do it consistently with two failures.

- For long-term pointing, all three methods are able to deal with zero and one failures, with 2018 Qiang Shen *et al.* having the best results and 2008 Wenchuan Cai *et al.* being the hardest to adapt and worst-performing one.

As a summary, 2018 Qiang Shen *et al.* performs the best across the board, being very easy to adapt. 2014 Danyal Bustan *et al.* is the second easiest to adapt and has the unique advantage of detumbling with two failures. 2008 Wenchuan Cai *et al.* was the most difficult to adapt and showed consistently the worst results.

A future expansion of this work could be horizontally, doing the same process that has been done in this Thesis but for other innovative techniques from the list in section 1.1. However, the most interesting would be to continue on with the methods investigated and propose the adaptive techniques discussed to be applied to a real mission. The LUMIO CubeSat would be a potential candidate.

To successfully do that, two steps would have to be taken. First, a much finer optimisation of the methods' parameters for each of the spacecraft's life-cycle scenarios. This would require much more computational power and automation than what has been employed for this Thesis.

Secondly, the methods would have to be tested in a state-of-the-art simulator in the context of a real mission to validate their performance. The GAFE Simulator mentioned in [5] is a good example of what is being referred to here. Given the results obtained in the Thesis, it would be suggested to give priority to 2018 Qiang Shen *et al.*, followed by 2014 Danyal Bustan *et al.*, and lastly testing 2008 Wenchuan Cai *et al.*.

Bibliography

[1] B. Bosman, A. Paskeviciute, K. Woroniak, E. Bertels, A. Cervone, S. Speretta, A. Menicucci, D. Labate, G. Pilato, A. Kukharenka, A. Thorvaldsen, G. Merisio, C. Giordano, V. Franzese, M. Massari, and F. Topputo. LUMIO Phase A System Design Report (SDR). Provided by the Thesis mentor, 2021, Issue 1, Rev 2.

[2] Cal Poly. CubeSat Design Specification, Revision 13. `https://www.cubesat.org/cubesatinfo`, April 6th, 2015.

[3] L. De Rocco, M. Lovera. THE DESIGN OF THE ATTITUDE CONTROL SYSTEM FOR THE AGILE MISSION. Provided by the Thesis mentor.

[4] Nanosats Database. `https://www.nanosats.eu/`.

[5] Airbus Defence and Space. Proposal for Innovative Fault Management for AOCS COTS HW - Technical Part. Privately accessed *in confidence*, October 6th, 2020.

[6] Institute of Electrical and Electronics Engineers online library. https://ieeexplore.ieee.org/Xplore/home.jsp.

[7] American Institute of Aeronautics and Astronautics Journal of Guidance, Control, and Dynamics. https://arc.aiaa.org/journal/jgcd.

[8] ResearchGate online library. https://www.researchgate.net/.

[9] ScienceDirect online library. https://www.sciencedirect.com/.

[10] D. Pérez, R. Bevilacqua. Differential drag spacecraft rendezvous using an adaptive lyapunov control strategy. *Acta Astronautica*, 83:196–207, 2013.

[11] C. Riano-Rios, R. Bevilacqua, W. E. Dixon. Adaptive control for differential drag-based rendezvous maneuvers with an unknown target. *Acta Astronautica*, 2020.

[12] C. Riano-Rios, R. Bevilacqua, W. E. Dixon. Differential drag-based multiple spacecraft maneuvering and on-line parameter estimation using integral concurrent learning. *Acta Astronautica*, 174:189–203, 2020.

[13] S. Yin, B. Xiao, S. X. Ding, D. Zhou. A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 63(5), 2016.

[14] J. Ahmed, V. T. Coppola, D. S. Bernstein. Adaptive Asymptotic Tracking of Spacecraft Attitude Motion with Inertia Matrix Identification. *Journal of Guidance, Control, and Dynamics*, 21(5):684–691, 1998.

[15] Stanford University. Lecture 12 Basic Lyapunov theory. `https://stanford.edu/class/ee363/lectures/lyap.pdf`, 2008-09.

[16] C. Pittet, A. R. Luzi, D. Peaucelle, J.M. Biannic, J. Mignot. In-flight results of adaptive attitude control law for a microsatellite. *CEAS Space J*, 7:291–302, 2015.

[17] D. Bustan, S. K. Hosseini Sani, N. Pariz. Adaptive Fault-Tolerant Spacecraft Attitude Control Design With Transient Response Control. *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, 19(4):1404–1411, 2014.

[18] Q. Shen, C. Yue, C. Hiang Goh, B. Wu, D. Wang. Rigid-Body Attitude Tracking Control Under Actuator Faults and Angular Velocity Constraints. *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, 23(3):1338–1348, 2018.

[19] M. Li, M. Hou, C. Yin. Adaptive Attitude Stabilization Control Design for Spacecraft Under Physical Limitations. *Journal of Guidance, Control, and Dynamics*, 39(9):2176–2180, 2016.

[20] A. Zou, K. D. Kumar, Z. Hou. Quaternion-Based Adaptive Output Feedback Attitude Control of Spacecraft Using Chebyshev Neural Networks. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 21(9):1457–1471, 2010.

[21] A. Zou, K. D. Kumar. Adaptive fuzzy fault-tolerant attitude control of spacecraft. *Control Engineering Practice*, 19:10–21, 2011.

[22] Q. Hu, Y. Zhang, X. Huo, B. Xiao. Adaptive Integral-type Sliding Mode Control for Spacecraft Attitude Maneuvering Under Actuator Stuck Failures. *Chinese Journal of Aeronautics*, 24:32–45, 2011.

[23] W. Cai, X. H. Liao, Y. D. Song. Indirect Robust Adaptive Fault-Tolerant Control for Attitude Tracking of Spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1456–1463, 2008.

[24] Q. Hu, M. I. Friswell, D. J. Wagg, S. Neild. Adaptive Backstepping Fault-tolerant Control for Flexible Spacecraft with Bounded Unknown Disturbances. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, December 2009.

[25] Q. Shen, D. Wang, S. Zhu, E. K. Poh. Robust Control Allocation for Spacecraft Attitude Tracking Under Actuator Faults. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 25(3):1068–1075, 2017.

[26] A. T. Espinoza, W. Sanchez. On-board Parameter Learning Using a Model Reference Adaptive Position and Attitude Controller. In *2017 IEEE Aerospace Conference*, March 2017.

[27] A. T. Espinoza, D. Roascio. Concurrent Adaptive Control and Parameter Estimation through Composite Adaptation Using Model Reference Adaptive Control/Kalman Filter Methods. In *2017 IEEE Conference on Control Technology and Applications*, August 2017.

[28] E. Wilson, D. W. Shutter, R. W. Mah. MCRLS FOR ON-LINE SPACECRAFT MASS AND THRUSTER-PROPERTY IDENTIFICATION. In *Proceedings of the IASTED International Conference on Intelligent Systems and Control*, August 2004.

[29] D. S. Berkovitz. System Characterization and Online Mass Property Identification of the SPHERES Formation Flight Testbed. Master's thesis, Massachusetts Institute of Technology, February 2008.

[30] E. Wilson, D. W. Shutter, R. W. Mah. Motion-Based Thruster Fault Detection and Isolation. In *Proceedings of the 2005 AIAA Infotech@Aerospace Conference*, September 2005.

[31] C. Pittet, A. Falcoz, D. Henry. A Model-based diagnosis method for transient and multiple faults of AOCS thrusters. *IFAC-PapersOnLine*, 49(17):82–87, 2016.

[32] C. Nainer. In-Orbit Data Driven Parameter Estimation for Attitude Control of Satellites. Master's thesis, Université de Lorraine, May 2020.

[33] U. Lee, D. Besson, M. Mesbahi. Fast Inertia Property Estimation via Convex Optimization for the Asteroid Redirect Mission. In *53rd IEEE Conference on Decision and Control*, December 2014.

[34] Q. Hou, Y. Cheng, N. Lu, B. Jiang. Study on FDD and FTC of satellite attitude control system based on the effectiveness factor. In *2nd International Symposium on Systems and Control in Aerospace and Astronautics*, December 2008.

[35] H. Yoon, K. M. Riesing, K. Cahoy. Kalman Filtering for Attitude and Parameter Estimation of Nanosatellites Without Gyroscopes. *Journal of Guidance, Control, and Dynamics*, 40(9):2272–2288, 2017.

[36] Franco Bernelli Zazzera. *Course Notes Spacecraft Attitude Dynamics and Control*. Politecnico di Milano, 2020.

Appendix A

Sources classification table

Problem	**Year, Authors and Source**	**Method used**	**Completeness**	**Easiness**	**Results Quality**	**Overall**
Generic actuator loss of effectiveness	2018 Q. Shen *et. al.* [18]	Adaptive parameter update law	5	5	3	13
	2010 A. Zou *et. al.* [21]	Adaptation law with virtual error dynamics and fuzzy logic rules to estimate nonlinear term	4	3	3	10
	2014 D. Bustan *et. al.* [17]	Adaptive law with extra parameter for transient control	4	5	4	13
Generic actuator bias	2018 Q. Shen *et. al.* [18]	Adaptive parameter update law	5	5	3	13
Generic actuator saturation	2016 M. Li *et. al.* [19]	Neural network to estimate nonlinear saturation term	3	2	2	7
	2014 D. Bustan *et. al.* [17]	Explicitly includes "Umax" in the control law	4	5	4	13
Generic actuator sign reversal	2010 A. Zou *et. al.* [21]	Adaptation law with virtual error dynamic and fuzzy logic rules to estimate nonlinear term	4	3	3	10

Problem	Year, Authors and Source	Method used	Completeness	Easiness	Results Quality	Overall
Limitation on allowable angular velocity	2018 Q. Shen *et. al.* [18]	Command filter on angular velocity based on hyperbolic tangent	5	5	5	15
	2016 M. Li *et. al.* [19]	Command filter on angular velocity based on hyperbolic tangent	5	5	5	15
Thruster outage	2011 Q. Hu *et. al.* [22]	Adaptive law with sliding surface integral term	5	4	4	13
	2005 E. Wilson *et. al.* [30]	Maximum likelihood estimation for FDI of predetermined failure modes	3	3	4	10
	2016 C. Pittet *et. al.* [31]	Two methods for fault identification. FDI based on total moment of inertia, and Unknown Input Observer	2	3	2	8
	2008 W. Cai *et. al.* [23]	Adaptive law with combined filtered error variable "s"	5	5	5	15
Thruster stuck	2011 Q. Hu *et. al.* [22]	Adaptive law with sliding surface integral term	5	4	4	13
Thruster loss of effectiveness	2008 W. Cai *et. al.* [23]	Adaptive law with combined filtered error variable "s"	5	5	5	15
	2005 E. Wilson *et. al.* [30]	Maximum likelihood estimation for FDI of predetermined failure modes	3	3	4	10
Thruster bias	2004 E. Wilson *et. al.* [28]	Online/offline Multiple concurrent recursive LS for direct and inverse matrices	4	5	3	12
Thruster misalignment	2020 C. Nainer [32]	Instrumental Variable method for non-normal Gaussian noise for offline estimation	5	2	3	10
Thruster limitation	2008 W. Cai *et. al.* [23]	Adaptive law with combined filtered error variable "s"	5	4	5	14

Problem	Year, Authors and Source	Method used	Completeness	Easiness	Results Quality	Overall
Reaction wheels loss of effectiveness	2009 Q. Hu *et. al.* [24]	Sliding surface design and adaptation law	2	3	2	7
	2008 Q. Hou *et. al.* [34]	Two-stage KF, actuator effectiveness factor included in the filter for estimation	3	2	2	7
Reaction wheels bias	2009 Q. Hu *et. al.* [24]	Sliding surface design and adaptation law	2	3	2	7
Reaction wheels failure	2009 Q. Hu *et. al.* [24]	Sliding surface design and adaptation law	2	3	2	7
Reaction wheels misalignment	2017 H. Yoon *et. al.* [35]	Augmented KF	4	4	3	12
Reaction wheels saturation	2015 C. Pittet *et. al.* [16]	Structured adaptive parameters resembling original PD controller	4	2	4	10
Uncertain inertia	2011 Q. Hu *et. al.* [22]	Control law independent from Inertia terms	5	4	4	13
	2017 H. Yoon *et. al.* [35]	Augmented KF	4	4	3	12
	2020 C. Nainer [32]	Instrumental Variable method for non-normal Gaussian noise for offline estimation	5	2	3	10
	2014 U. Lee *et. al.* [33]	Add constrains to the problem and formulate it as a convex optimization, use MATLAB CVX or other to solve it	3	2	2	7
	2004 E. Wilson *et. al.* [28]	Online/offline Multiple concurrent recursive LS for direct and inverse matrices	4	5	3	12
	2008 D. S. Berkovitz [29]	Online recursive LS estimation or offline batch LS of inverse inertia matrix	4	5	4	13
	1998 J. Ahmed *et. al.* [14]	Adaptive law with inertia components as estimation parameters	5	5	2	12

Problem	Year, Authors and Source	Method used	Completeness	Easiness	Results Quality	Overall
Uncertain inertia	2017 A. T. Espinoza and W. Sanchez [26]	MRAC to estimate the inertia matrix components	5	3	2	10
	2009 Q. Hu *et. al.* [24]	Adaptation parameter to estimate is the norm of the inertia matrix	2	3	2	7
	2014 D. Bustan *et. al.* [17]	Explicitly includes "Umax" in the adaptive control law	4	5	1	10
	2008 W. Cai *et. al.* [23]	Control law independent from Inertia terms	5	5	5	15
Uncertain Centre of Mass	2008 D. S. Berkovitz [29]	Online recursive LS estimation or offline batch LS	4	5	4	13
	2014 U. Lee *et. al.* [33]	Add constrains to the problem and formulate it as a convex optimization, use MATLAB CVX or other to solve it	3	2	2	7
	2004 E. Wilson *et. al.* [28]	Online/offline Multiple concurrent recursive LS for direct and inverse matrices	4	5	3	12
Uncertain total mass	2004 E. Wilson *et. al.* [28]	Online/offline Multiple concurrent recursive LS for direct and inverse matrices	4	5	1	10
	2017 A. T. Espinoza and D. Roascio [27]	Combines Extended KF with MRAC into a composite adaptation technique	2	2	2	6
Gyro sensor bias	2008 Q. Hou *et. al.* [34]	Two-stage KF, actuator effectiveness factor included in the filter for estimation	3	2	2	7
	2020 C. Nainer [32]	State-Variable low pass filter to reduce effect of noise and bias for offline estimation	5	2	3	10

Problem	Year, Authors and Source	Method used	Completeness	Easiness	Results Quality	Overall
Gyro sensor outage or absent	2016 C. Pittet *et. al.* [31]	Two methods for fault identification. FDI based on total moment of inertia, and Unknown Input Observer	2	3	2	8
	2017 H. Yoon *et. al.* [35]	Augmented KF	4	4	3	12
	2020 C. Nainer [32]	Second-order Butterworth filter and central differentiation from position measurements for offline estimation	3	2	3	8
Residual magnetic dipole	2017 H. Yoon *et. al.* [35]	Augmented KF	4	4	3	12