POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Preliminary Design of an Attitude Determination Module for Small Spacecraft in LEO

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Lucas Reino Diez**

Student ID: 986110
Advisor: Prof. Franco Bernelli
Co-advisor: Manuel Rodríguez Halcón
Academic Year: 2022-23

Abstract

In this work, the preliminary design of an attitude determination module for small spacecraft in LEO is carried out. Even though several standalone attitude determination and control system (ADCS) units are currently available, very few determination-only (ADS) solutions exist. This project aims to fill that gap in the market, whilst integrating multiple sensors from Solar MEMS Technologies' catalog. A versatile sensor suite is selected, together with appropriate mathematical models. Moreover, several variants of the multiplicative extended Kalman filter (MEKF) are studied, where two different MEMS gyroscopes are tested in dynamic replacement mode. Additionally, a novel bias clipping technique to improve convergence is introduced, which consists in adding artificial bounds to the bias estimate within the filter. This better captures the real behavior of the gyroscope, and shows promising results. The contents are presented in a didactic manner, and allow for a deep understanding of the multiple factors involved in the design process. A comprehensive simulation strategy, including three representative scenarios with different conditions, is developed to test the performance of the designed attitude determination module. Extensive Monte-Carlo simulations demonstrate that a combination of Sun sensor, three-axis magnetometer, star tracker and three-axis gyroscope measurements provide quasi-continuous attitude and angular velocity estimates with enough accuracy for most small spacecraft in LEO. The results present an in-depth comparison of current attitude estimation methods, and yield a sound starting point for future developments.

Keywords: spacecraft attitude, small spacecraft, estimation, Kalman filter

Sommario

In questo lavoro, viene effettuato il progetto preliminare di un modulo di determinazione dell'assetto per piccoli satelliti in orbita terrestre bassa (LEO). Anche se attualmente esistono diverse soluzioni autonome di determinazione e controllo dell'assetto (ADCS), esistono poche soluzioni solo di determinazione (ADS). Questo progetto mira a colmare questa lacuna nel mercato, integrando contemporaneamente più sensori dal catalogo di Solar MEMS Technologies. Viene selezionata una serie di sensori versatile, insieme a modelli matematici appropriati. Inoltre, vengono studiate diverse varianti del filtro di Kalman esteso moltiplicativo (MEKF), dove vengono inseriti due diversi giroscopi MEMS in modalità di sostituzione dinamica. Inoltre, viene introdotta una nuova tecnica di limitazione del bias dei sensori per migliorare la convergenza, che consiste nell'aggiungere limiti artificiali alla stima del bias all'interno del filtro. Questo rispetta meglio il comportamento reale del giroscopio e mostra risultati promettenti. I contenuti sono presentati in modo didattico e consentono una comprensione approfondita dei molteplici fattori coinvolti nel processo di progettazione. Viene sviluppata una strategia di simulazione completa, che include tre scenari rappresentativi con condizioni diverse, per verificare le prestazioni del modulo di determinazione dell'assetto progettato. Estese simulazioni con metodo Monte Carlo dimostrano che una combinazione di sensore solare, magnetometro a tre assi, star tracker e giroscopio a tre assi forniscono stime quasi continue dell'assetto e della velocità angolare con sufficiente precisione per la maggior parte dei piccoli satelliti in LEO. I risultati presentano un confronto approfondito dei metodi attuali di stima dell'assetto e costituiscono un solido punto di partenza per futuri sviluppi.

Parole chiave: assetto del satellite, piccoli satelliti, stima, filtro di Kalman

Contents

List of Figures

List of Tables

1 Introduction

1.1. Context

The *attitude* of a spacecraft is defined as its orientation in space, with respect to a suitable reference frame. Most modern-day spacecraft missions have specific pointing requirements, which drive the need to accurately estimate the spacecraft's attitude and then effectively control it. Typically, one talks about an *attitude determination and control system* (ADCS) of a spacecraft. The estimation and control problems are coupled but, nonetheless, can be considered separately to some extent. This approach has been adopted by most space missions, as it greatly simplifies the design process and in most cases provides satisfactory results.

This work has been carried out at Solar MEMS Technologies, which is a company specialized in micro-electromechanical systems technology (MEMS) and its application to the design and development of high accuracy Sun sensors for the aerospace industry. In addition to Sun sensors, they offer a horizon sensor and are currently developing a star tracker for nano satellites. Given that the company does not produce any kind of actuator to carry out the attitude control of a spacecraft, it is reasonable to limit the scope of this project to attitude determination. The objective of the thesis is therefore to perform the preliminary design of an attitude determination module by integrating multiple sensors from Solar MEMS Technologies' catalog.

When designing an ADCS from scratch, sensors must be installed in precise locations and fed by dedicated power and data lines. Additionally, attitude estimation algorithms need to be developed. The efficient integration of all elements is not a simple task, and requires specialized engineers. By designing a self contained attitude determination module, spacecraft designers would only need to integrate a single, compact unit with all the necessary components. This considerably facilitates implementation of the ADCS, and ensures performance in a variety of scenarios.

1.2. Scope of the Project

The field of spacecraft attitude determination was born in the early 1960s, coinciding with the beginning of the Space Age. Since the first algorithm for determining the attitude of a spacecraft was published in 1964, a great number of developments have been carried out. Today, many different approaches to the attitude determination problem exist, and investigation in the field is still ongoing.

The objective of this project is not to perform an comprehensive review of all the different algorithms available, nor present a novel approach (although some original techniques are proposed). Instead, it will focus on selecting an appropriate suite of sensors, implementing currently available algorithms, and conducting simulations to evaluate the performance of the attitude determination module in various scenarios. A more practical approach has therefore been adopted, although no physical experiments will be carried out.

The scope is limited to a preliminary assessment of the module's components and algorithm using a high-level programming language. Other aspects such as hardware selection (choosing a suitable microcontroller), the module's structural design (housing, component location, etc.) or its interfaces with other subsystems of the spacecraft will not be considered, and are left for future work.

1.3. Objectives

The main goal of this project is to perform the preliminary design of an attitude determination module suitable for small spacecraft in low Earth orbit (LEO). This may be broken down into a number of smaller objectives:

- Analyze other existing ADCS modules on the market and extract their main characteristics.
- Explore different sensor configurations, suited for a representative set of realistic scenarios.
- Study at least two modern algorithms for attitude estimation, and determine which one provides the better performance in the scenarios considered.
- Assess the precision in attitude knowledge that can be achieved using Solar MEMS Technologies' sensors.

1.4. Structure of the Document

The thesis will be organized in three main blocks: preliminary studies, design, and simulation. Starting with the first block, in Chapter 2, the current small spacecraft market is analyzed, and present-day commercial ADS/ADCS solutions are reviewed. It must be noted that most available units combine both attitude determination and control, yet they are still relevant for the topic of this thesis. Additionally, a brief review of historical and modern attitude determination algorithms is included. In Chapter 3, the theoretical framework required to tackle the attitude determination problem is provided, covering attitude representations, attitude kinematics and some relevant concepts from statistics.

Chapter 4 addresses the sensor selection and modeling. The usefulness of various sensors in different scenarios is discussed, and some measurement models are presented. The design block continues with Chapter 5, in which several variations of the extended Kalman filter are described, commenting some of their advantages and limitations. Additionally, an original technique to improve filter convergence is proposed.

In order to test and compare the previous attitude estimation methods, in Chapter 6, the complete ADS is evaluated through multiple simulations in a variety of scenarios. Finally, in Chapter 7, the final conclusions are drawn and future developments suggested.

2 State of the Art

The purpose of this chapter is to reflect the current state of small spacecraft systems and ADCS technologies. Only publicly available information will be reviewed. The intent is to provide motivation for the development of an attitude determination module that will fit into the current market, justifying the topic of this thesis.

2.1. Small Spacecraft Systems and Current Market

A small satellite, or smallsat, is a satellite with a mass less than 180 kg and reduced dimensions [5]. They can additionally be divided into the following categories:

1. Minisatellite 100-180 kg
2. Microsatellite 10-100 kg
3. Nanosatellite 1-10 kg
4. Picosatellite < 1 kg

Perhaps the most popular kind of small satellites, CubeSats, fall both under the microsatellite and nanosatellite clasifications. They use a standard size and form factor, consisting of small $10 \times 10 \times 10$ cm cubes which can be arranged in different configurations (1U, 2U, 3U, etc.). The historical distribution of form factors is shown in Figure 2.1, and it is clear that 3U CubeSats continue the most popular size. 0.25U has increased rapidly in the last years thanks to Swarm, and 6U is being used by multiple newer constellations [1]. Other platforms such as PocketQube, TubeSat of ThinSat, although much less popular, have also been included in this comparison.

A comprehensive review of the current state of nanosatellites, including launch forecasts and other predictions, can be found in [6]. Figure 2.2 shows the yearly number of nanosatellite launches together with historical and current predictions. The bottom line is that, even though most predictions have been overestimated, with only four interplanetary CubeSats in space, expanding launch options, and numerous technologies yet to be developed, the times of nanosatellites are still ahead.

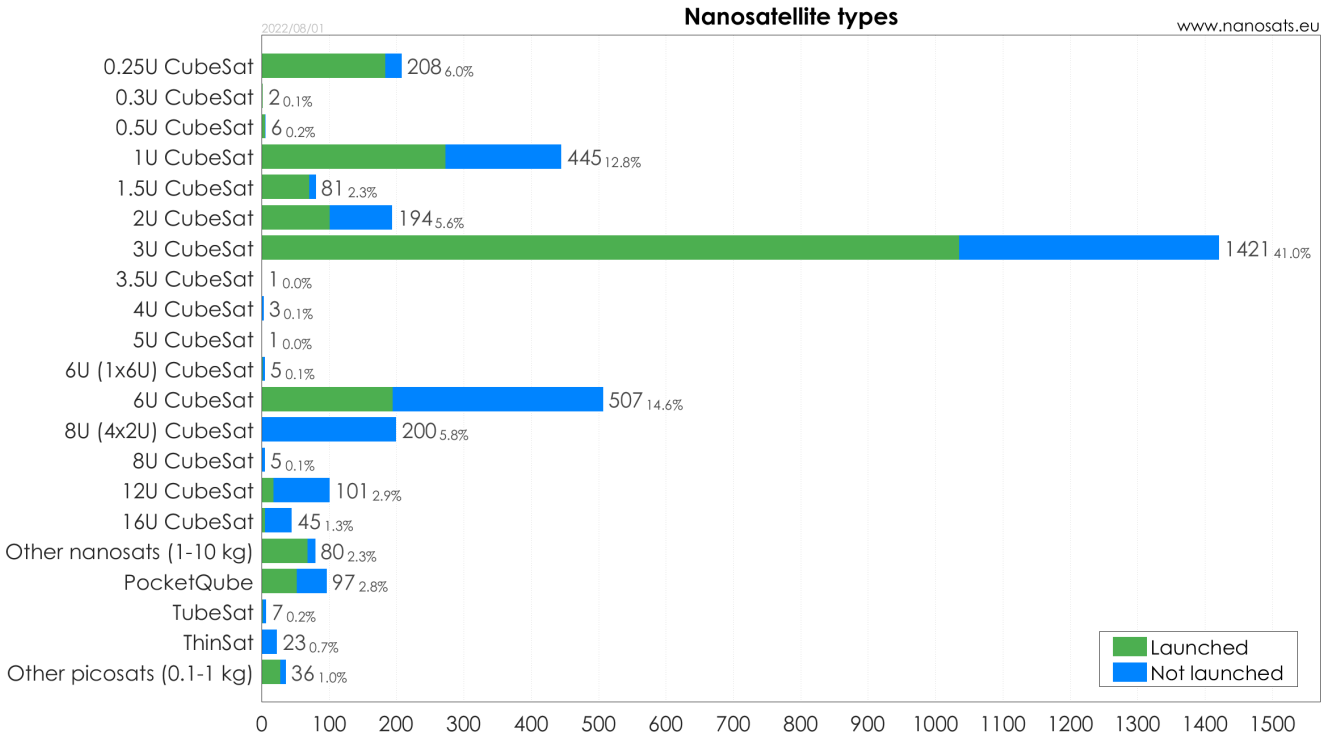


Figure 2.1: Nanosatellite form factors. Credit: Kulu [1].

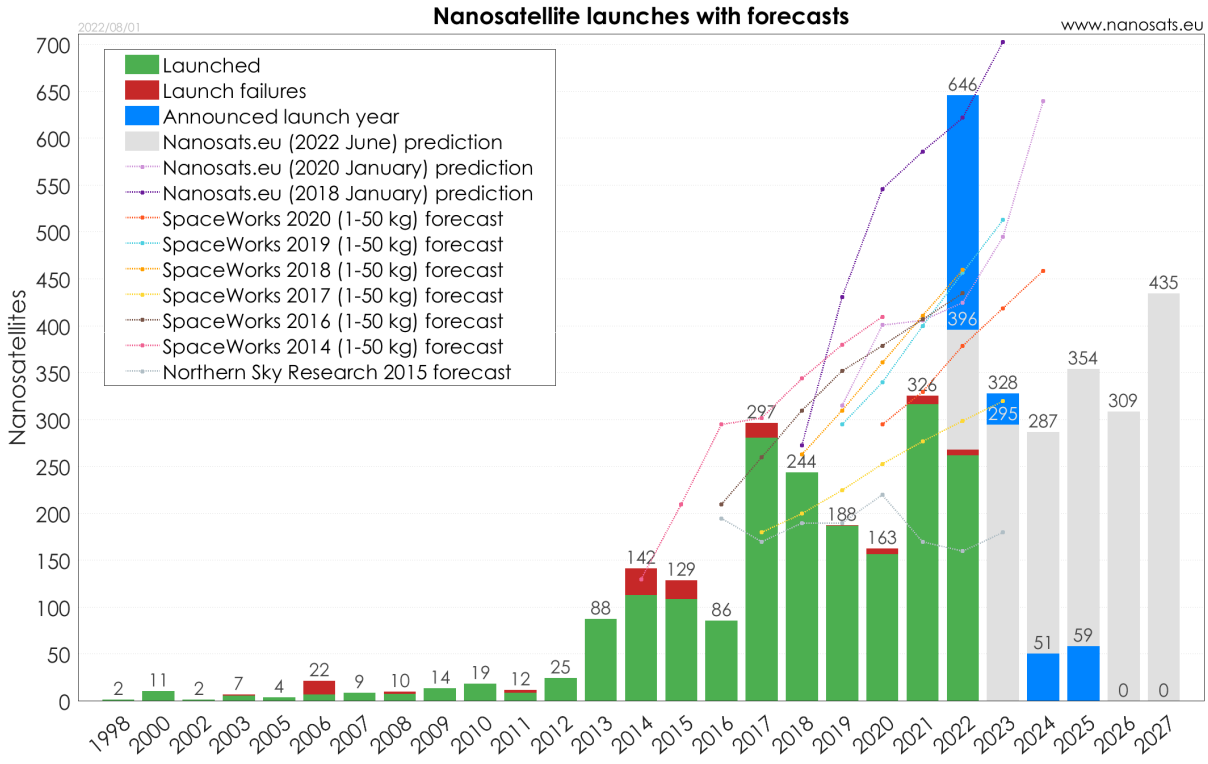


Figure 2.2: Launch Forecast showing previous predictions. Credit: Kulu [1].

Another interesting conclusion that can be extracted from the data is that sizes are both getting larger and smaller. On one hand, educational missions could move into smaller form factors. On the other hand, commercial satellites are moving towards larger sizes to save development costs on miniaturization, spurred by decreasing launch costs.

Furthermore, the number of launches beyond LEO will increase significantly in the following years. These include MEO, GTO, GEO, lunar and interplanetary or deep space missions. According to [6], only eight of such CubeSats had been launched by 2021. In contrast, 60 launches are scheduled up to 2025, with 15 already performed in 2022. In this context, there is a slight skew towards larger form factors, being 6U and 12U the most popular choices for current and planned missions.

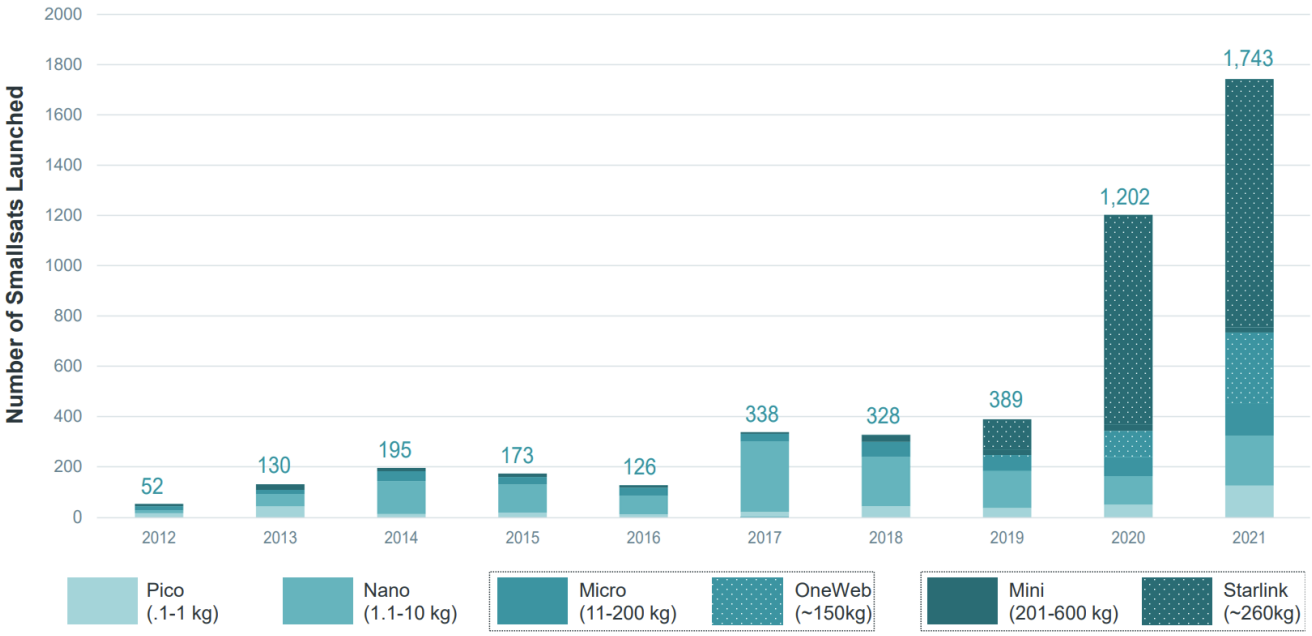


Figure 2.3: Smallsat market evolution since 2012 by mass class. Credit: Bryce Space and Technology [2].

All things considered, in 2020 and 2021 nanosatellites make up only about 10% of the total number of smallsats launched [2]. These numbers, however, are heavily influenced by the recent Starlink and OneWeb constellations. Excluding them, the nanosatellite class has constituted the largest share of smallsats since 2013. In 2021, they account for 42% of the total share, while picosatellites and microsatellites are tied at 27% and minisatellites come last at only 4%. Including the Starlink and OneWeb constellations, mainly constituted of larger minisatellites (Starlink) and microsatellites (OneWeb), these classes' shares increase to 58% (mini) and 24% (micro) respectively, with a combined >1000 launches (see Figure 2.3). As has been shown, a single company such as SpaceX

or Planet can heavily skew the results. Therefore it is not easy to predict how will the smallsat market drift in the years to come.

What is clear is that smallsats have become the main focus of the space industry, making up to 94% of the spacecraft launched in 2021. This is a very recent trend, since 69% of all smallsats in the last 10 years have been launched just in 2020 and 2021. The great majority of these are for communication purposes, followed to a lesser extent by remote sensing and technology development [2].

The main operators have become SpaceX (1944 smallsats in orbit), Planet (485), OneWeb (394), Spire Global (147) and Swarm Technologies (121), recently acquired by SpaceX. That is, 70% of the smallsat market is controlled by US operators, followed by the UK and China with only a 9% and 6% share respectively [2]. The extent to which European companies will be able to keep up with this pace is hard to predict.

One important bottleneck that the small satellite sector is facing worldwide is the lack of convenient launch systems, which is hindering smallsat projects and putting pressure on launch services [7]. In Europe, fourteen organizations have joined forces in a EU Horizon 2020 project called SMall Innovative Launcher for Europe (SMILE) [8]. In words of the project coordinator Leo Timmermans, "Over 500 small satellites are currently operational, most of which are 1-5 kg CubeSats. A trend towards more mature and better performing satellites (10-50 kg) is on-going. About 50-100 companies are currently deploying small satellites, and many more will follow."

This information is vital to address the feasibility of a new ADCS development. As of now, most commercially available options (all focused on CubeSats) are based outside of Europe. This implies hefty import fees for European companies that may want to use these products. Additionally, many EU funded projects require made in Europe goods to be used. Another aspect to take into account is the fact that mega constellation projects tend to implement in-house ADCS solutions. In 2021, 73% out of 1743 smallsats launched were destined towards Starlink and OneWeb's constellations. Therefore, the real market for a commercial turnkey ADCS is much smaller than what the numbers may imply.

2.2. Present-day Integrated ADCS Solutions

In this section, currently available commercial ADCS solutions will be analyzed. Almost all of them are geared towards CubeSats which, as discussed in the previous section, constitute the main target within the smallsat market.

Table 2.1: General information of currently available integrated systems.

Manufacturer	Model	Platform	Actuators	Sensors	Price
CubeSpace	*Y-Momentum Series*	2-12U	1 RW 3 MTQ	10 coarse SS 3 gyros 1 TAM	$22-27k
	3-Axis Series	2-12U	3 RW 3 MTQ (= •)	10 coarse SS 1 fine SS, 1 ES 3 gyros, 1 TAM (Optional STR)	$34-58k
Blue Canyon Technologies	*XACT Series*	3-12U	•	1 STR, 1 TAM	∼ $140k
	FlexCore	Microsats	•	1 STR, 1 TAM	N/A
AAC Clyde Space & Berlin Space Technologies	*iADCS Series*	1-12U	•	3 gyros 1 STR (Optional TAM and precision gyro)	∼ €140k
KU Leuven	*ARCUS*	<12U	•	3 gyros 1 STR, 1 TAM 6 photodiodes	N/A
SatRev S.A.	*ADCS*	∼12U*	•	1 2D SS 3 gyros, 3 ACC 1 TAM, 1 GPS	N/A
NewSpace Systems	*CubeSat ACS*	∼12U*	•	3 fine SS, 3 gyros 1 stellar gyro 1 TAM, 1 GPS	N/A
Tensor Tech	*ADCS10*	1-3U	3 MTQ	1 fine SS 1 TAM	N/A
	ADCS100-400	1.5-16U	3 MTQ 1-4 reaction spheres	6 fine SS 3 gyros 1 TAM	∼ $70k
D-Orbit	*ADCS Suite*	Minisats	•	Configurable	N/A
Serenum Space	*VAC02 ADCS*	1-6U	•	6 SS, 3 gyros 2 TAM	N/A
Adcole Maryland Aerospace	*MAI-25*	1-3U*	1 RW 3 MTQ	1 TAM	N/A
	MAI-400	3-6U*	•	2 ES, 1 TAM	N/A
	MAI-500	3-6U*	•	2 STR, 1 TAM	N/A

* These target platforms are not explicitly stated by the manufacturer, but have been derived from the maximum torque values.

As can be seen in Table 2.1, a number of companies from all around the globe are involved in the manufacturing of integrated systems. These combine the data from multiple sensors to determine the spacecraft's attitude, and provide actuators to control it. They also include processors and software with attitude determination and control capabilities. Typical components are reaction wheels (RW), magnetorquers (MTQ), three-axis magnetometers (TAM), star trackers (STR), Sun sensors (SS) and Earth sensors (ES).

This list is quite comprehensive, compiling information from multiple reliable sources such as Satsearch [9], CubeSatShop [10] and NASA's *State-of-the-Art of Small Spacecraft Technology* report [11], besides the actual manufacturer's websites. Of course, the research is limited to publicly available information, and any ongoing development which hasn't been disclosed yet will not be reflected. Some systems might have been left out as well.

Something to keep in mind is the fact that not all the presented models are up to the same NASA standard Technology Readiness Level (TRL) as defined by the 2020 NASA Engineering Handbook, found in NASA NPR 7123.1C *NASA Systems Engineering Processes and Requirements*. In some cases, flight heritage is claimed (CubeSpace, Blue Canyon Technologies, Tensor Tech, AAC Clyde Space & Berlin Space Technologies, NewSpace Systems, Adcole Maryland Aerospace), but most manufacturers do not specify a specific TRL. Others mention qualification and acceptance testing (Serenum Space, KU Leuven), and overall a TRL of at least 5 is expected from the analyzed systems.

Except for Blue Canyon Technologies' FlexCore and D-Orbit's ADCS suite, which are oriented towards microsats and minisats respectively, the rest have been designed for CubeSats ranging from 1U to 16U. The actuator choice is quite uniform across the board, with most systems opting for a combination of three reaction wheels and three magnetorquers to achieve three-axis control. For less demanding applications (detumbling and Earth or Sun pointing), one or no reaction wheels can be used. An exception is the novel *reaction sphere* concept from Tensor Tech, which consists of a variable-speed, single-gimbal control moment gyroscope (CMG) driven by spherical motor technology. Traditional CMGs are rarely used since reaction wheels provide sufficient torque for the majority of applications in the smallsat segment, without the added complexity. Perhaps future missions with high-agility requirements could benefit from using them, but currently there aren't many CMGs suited for this spacecraft class available.

There is more variety regarding the sensor selection. All models implement a magnetometer, often along sun sensors and MEMS gyroscopes. Some rely more heavily on the use of a star tracker, and others include Earth sensors or GPS receivers. Several manufacturers provide various levels of customization, allowing the client to choose the sensor combination that better suits its mission (turn-key solutions).

In Table 2.2, some performance parameters of the previous systems are analyzed. Regarding the volume, most models have a box shape with a base of approximately 10×10 cm and varying height. If, for example, the height is of 3 cm, then the volume is typically said to have a 0.3U form factor (as it occupies 30% of a CubeSat unit).

As to the radiation tolerance, some manufacturers provide a numerical value in krad, and others simply state the system's lifetime in a LEO orbit in years (which correlates with a certain amount of radiation).

Table 2.2: Performance characteristics of currently available integrated systems.

Manufacturer	Model	Mass [g]	Volume	Pointing Accuracy†	3-Axis Control	Radiation Tolerance
CubeSpace	*Y-Momentum Series**	300-600	0.3-0.5U	5°	—	20 krad
	*3-Axis Series**	530-1150	0.3-0.8U	1°	•	20 krad
Blue Canyon Technologies	*XACT Series**	885-1810	0.5-0.75U	0.02°	•	5 years
	*FlexCore**	Config. dependent	Config. dependent	0.02°	•	5 years
AAC Clyde Space & Berlin Space Technologies	*iADCS Series*	345-1500	0.3-0.7U	1°	•	2 years
KU Leuven	*ARCUS*	715	0.5U	0.3°	•	3 years
SatRev S.A.	*ADCS**	290	0.2U	N/A	•	N/A
NewSpace Systems	*CubeSat ACS**	200	0.2U	1°	•	10 krad
Tensor Tech	*ADCS10*	140	0.2U	N/A	—	3 years
	*ADCS100-400**	300-1200	0.2-0.8U	0.2°	•	3 years
D-Orbit	*ADCS Suite**	Config. dependent	Config. dependent	0.3°	•	N/A
Serenum Space	*VAC02 ADCS*	424	0.3U	1.3°	•	36 krad / 3 years
Adcole Maryland Aerospace	*MAI-25*	250	0.25U	3°	—	N/A
	MAI-400	700	0.5U	N/A	•	N/A
	MAI-500	1050	0.6U	0.1°	•	N/A

* These models may not be fully integrated, that is, there can be elements (usually reaction wheels) which are not included in the mass and volume calculations.
† Expressed in terms of 3σ limits (99.7% confidence interval).

Only Blue Canyon Technologies produces systems with high pointing accuracy (0.02° or 100 arcsec). The rest drop by one or two orders of magnitude, to around 0.1° and 1° respectively. These values are still acceptable, and depending on the application could be more than enough. It must be noted that in this context, pointing accuracy refers to attitude control accuracy. Alternatively, attitude determination accuracy can be considered. Some manufacturers explicitly state both, and in general the latter is approximately up to one order of magnitude higher.

(a) CubeSpace 3-Axis ADCS. (b) Blue Canyon Technologies XACT 100.

Figure 2.4: Fully integrated vs non fully integrated ADCS units.

Another factor to take into account is that not all units are fully integrated. Most of them consist of a single block with all the components inside, but some leave out the reaction wheels or other components (see Figure 2.4). This is primarily the case for models aimed towards bigger satellites, where removing the RWs from the main package allows for better space management. NewSpace Systems, in collaboration with CubeSpace, have taken this idea to the extreme, designing a distributed system where most components can be placed arbitrarily within the spacecraft (see Figure 2.5). Aimed towards microsatellites, not much information about the project (probably still under development) is available at the moment.

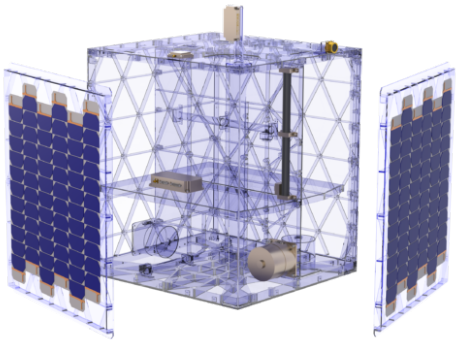


Figure 2.5: Distributed ADCS concept from NewSpace Systems.

Other technical considerations include the number of external interfaces available to incorporate additional components (for instance, AAC Clyde Space's iACDS200 can connect to 6 or more additional sun sensors, an external magnetometer or GNSS receivers), and long-term performance. Given the relatively poor reliability of many existing ADCS components, especially actuators, this is a concern for CubeSat applications. A component can become unserviceable due to radiation damage and/or other sources of degradation, in which case there needs to be some kind of backup plan to compensate for the loss. If the critical element is a sensor, the attitude determination algorithm must be able to autonomously correct for it. On the other hand, if it is an actuator, the control logic must vary accordingly (e.g. considering underactuated control). In general, information regarding these aspects is not publicly available. Some manufacturers allow for some degree of redundancy in order to avoid these issues (e.g. including four reaction wheels instead of three).

Perhaps just as important as technical specifications are the user interface and technical support provided by the manufacturer, as well as various geopolitical interests. As shown in Table 2.3, out of the 11 companies analyzed six are based inside the EU. Some manufacturers allow the user to program its own control loops, and offer programming support as an extra (e.g. Berlin Space Technologies for €12k per month).

Table 2.3: Manufacturer's countries of origin.

Manufacturer	Country
CubeSpace	South Africa
Blue Canyon Technologies	USA
AAC Clyde Space	Sweden/Scotland
Berlin Space Technologies	Germany
KU Leuven	Belgium
SatRev S.A.	Poland
NewSpace Systems	South Africa
Tensor Tech	Taiwan
D-Orbit	Italy
Serenum Space	Czech Republic
Adcole Maryland Aerospace	USA

2.2.1. ADS-Only Solutions

As of now, complete ADCS units have been considered. The market for these units is quite saturated, as multiple, already established products, offer solutions for a wide range of situations. Alternatively, attitude determination and control can be treated separately.

Since this thesis is being developed under the supervision of Solar MEMS Technologies S.L., who manufacture several sensors but no actuators, it only makes sense to consider developing an attitude determination module. An additional advantage is the fact that the ADS does not (always) depend on the satellite size, but just on the type of orbit/mission.

Currently, only D-Orbit advertises a standalone ADS unit: D-Sense. It includes several Sun sensors and Earth sensors, a magnetometer, a three-axis gyroscope and a camera "for photos, videos and star tracking". Everything is included into a single, compact box (200 g and $7 \times 7 \times 5$ cm), equipped with its own microcontroller, and fed by a single pair of power and data line (using a CAN bus interface). The power consumption is of 300 mW in standard operative mode and 500 mW in camera mode, and the unit is able to withstand temperatures ranging from $-30°C$ to $70°C$.

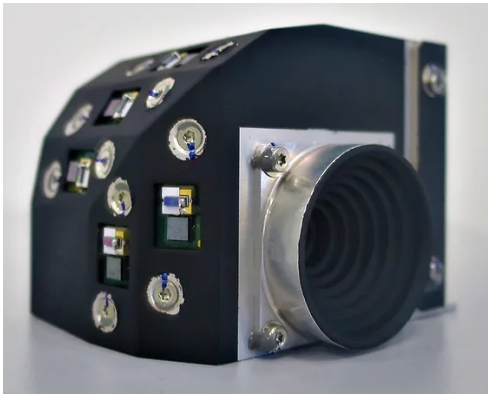


Figure 2.6: D-Sense multi-sensor ADS module.

2.3. Attitude Determination Algorithms

Besides physical and functional characteristics, the algorithms used to solve the attitude determination problem are essential to guarantee performance and robustness. Unfortunately, most manufacturers do not publish any details regarding the inner workings of their systems. A couple state the use of an extended Kalman filter (EKF) to compute the attitude vector, and CubeSpace additionally mentions using various rate filters and the TRIAD method. Usually, multiple estimation modes are used depending on the mission's phase (detumbling, target tracking, etc.), orbit type or sun availability among other factors.

In this next section, a brief review of the state of the art in attitude determination algorithms will be carried out. This will set the stage for the consequent ADS design.

Attitude determination methods can be divided into several categories. One useful division is to distinguish between static and filtering approaches. Static determination methods use measurements taken at the same moment, or close enough in time that the spacecraft motion between the measurements can be ignored or easily compensated for. One important requirement is that enough information to compute the attitude is available at each time. This implies at least one measured vector (e.g. the direction to a star in the spacecraft body frame) and one angle (e.g. of the Sun relative to the line-of-sight of a single-axis Sun sensor). In practice, however, over-determined approaches which use two vectors or more are typically employed. These include the TRIAD method, originally published in 1964, solutions to Wahba's problem, or maximum likelihood estimation (MLE).

The term *filtering* traditionally refers to a class of methods that can be used for estimating the state of a time-varying system which is directly or indirectly observed through noisy measurements. In the context of attitude estimation, filtering approaches make explicit use of the equations of motion of the spacecraft to accumulate a "memory" of past measurements. Typically, recursive filtering is chosen over batch filtering, as a real-time estimate of the attitude is desired.

The mathematics behind attitude determination can be broadly characterized into approaches that account for stochastic variables and approaches that do not. The term "attitude estimation" is in principle restricted to stochastic methods, while "attitude determination" refers to the latter approach. However, this distinction is often blurred, and the terms used interchangeably. Stochastic methods include Kalman filtering and MLE. On the other hand, the TRIAD method or Wahba's problem do not show explicit relations to stochastic errors in their original formulation.

Static approaches offer great simplicity and low computational cost. They have been extensively used in historical space missions and still have a place in spacecraft contingency design (in case of anomalies) or as simple tools for analysis purposes. Most modern spacecraft, however, implement filtering methods due to their versatility and the fact that they generally produce more accurate estimates.

Early applications in the late 70s and early 80s relied mostly on the extended Kalman filter (EKF) for attitude estimation. Since then, many new approaches have been developed which, in some aspects, have proven to be superior to the EKF. The vast majority fall under the category of optimal recursive estimation in the minimum mean-square error (MMSE) sense. Several of these algorithms still maintain the basic EKF structure but employ various modifications, mainly to improve convergence. This is the case for the filter

QUEST [12] or the iterated Kalman filter [13]. Others, such as the unscented Kalman filter [14, 15], transform the so called sigma-points instead of linearizing the dynamic model of the system. The previous algorithms are based on the Gaussian assumption that the probability density function is adequately described by its mean and covariance. Other approaches which do not require this assumption exist, but have not found widespread application. More details about modern attitude estimation algorithms through 2007 can be found in the comprehensive and excellent survey by Crassidis et al. [16].

In spite of the many newer alternatives, which have shown to have some advantages, Crassidis et al. [16] apply the old saying "If it ain't broke don't fix it", and argue that the standard EKF still remains the workhorse of real-time spacecraft attitude estimation due to its relatively simple formulation and flexible characteristics.

3 Theoretical Framework

In this chapter, a theoretical framework is provided, dealing with some fundamental concepts and historical attitude determination algorithms. It is assumed, however, that the reader has some familiarity with this material. For a comprehensive, from the ground up review of attitude determination theory, the reader is referred to the classic textbook edited by Wertz [17] or the more recent version by Crassidis & Markley [3].

First, some useful reference frames will be reviewed. Next, a few attitude representations, which characterize the rotation from one reference frame to another, are introduced. In Section 3.2.7, the focus will be on attitude error representations, which are used for small rotations and become essential in the Kalman filter formulation.

The relationship between the attitude dynamics of a spacecraft and a certain attitude representation is given by the attitude kinematics equations, which are stated in Section 3.3. Finally, Section 3.4 provides a brief refresher of some fundamental statistical concepts.

3.1. Useful Reference Frames

3.1.1. Spacecraft Body Frame

The spacecraft body frame is defined by an origin, specified somewhere within the spacecraft's body, and three right-handed Cartesian axes. A spacecraft is made up of multiple components, some of which may shift due to the extreme forces experienced during launch and/or thermal deformations, or moved deliberately (such as solar panels to be oriented towards the Sun, or the deployment of boom/antennas). Typically the body axes are fixed to some sufficiently rigid *navigation base*, where most sensors, actuators and other payload instruments are attached. The orientation of these components with respect to the navigation base, and therefore with respect to the body axes, is known. The purpose of attitude determination and control is to estimate and to control the orientation of the body frame with respect to some external reference frame, commonly an inertial frame such as those described in the next section.

3.1.2. Inertial Reference Frames

An inertial reference frame is a frame in which Newton's equations are valid. A frame moving at a constant velocity and with no rotation with respect to an inertial frame will also be inertial. In our universe everything is in constant accelerated motion, and therefore the concept of inertial frame is in theory only ideal. In practice, however, cosmic motion is extremely small compared to the dynamics of most systems, including spacecraft attitude and orbital dynamics. Celestial frames with their axes pointing to very distant "fixed" stars are the best realizations of inertial frames. The standard is the *International Celestial Reference System* (ICRF), with its z axis aligned with the Earth's North pole and its x axis with the *vernal equinox*, the intersection of the Earth's equatorial plane with the plane of the Earth's orbit around the sun (the *ecliptic*). The y axis is simply the one that completes the right-handed Cartesian coordinate system. Unfortunately neither the North pole nor the ecliptic are inertial, since both vary in time. The ICRF axes are defined to be the "mean" (smoothing out short-period dynamics) orientations of the pole and vernal equinox at a some fixed epoch.

The ICRF frame is centered at the Solar System's center of mass. Inertial frames whose center lies at Earth's center of mass are denoted *Earth Centered Inertial* (ECI) coordinate frames. These frames have a linear acceleration due to the rotation of the Earth around the Sun, but for attitude analysis it is unimportant. A commonly used ECI frame is the J2000 frame, whose definition is based on the Earth's equator and equinox, determined from observations of planetary motions, plus other data. The J2000 frame essentially coincides with the ICRF frame taken at 12:00 Terrestrial Time on 1 January 2000 (with its origin displaced). Note that the J2000 frame came first, and the realization of ICRF was made to coincide almost exactly with it. The difference between both is less than 0.1 arcsec, and therefore are mostly treated as equivalent. Other useful ECI frame is the TEME frame, used for the NORAD two-line elements.

Throughout this project, the J2000 frame will be considered as the default inertial reference frame (although this is not especially relevant).

3.2. Attitude Representations

The discipline of attitude determination is concerned with the estimation of the proper orthogonal matrix that transforms vectors from an inertial reference frame to a body frame. These will be referred to as rotation matrices or *attitude* matrices and will be denoted by the letter A. A transformation given by A can be interpreted in an active

(also known as *alibi*) or passive (*alias*) sense. Consider a vector $\mathbf{x}$ which is transformed into $\mathbf{x}'$. This transformation can be viewed as a direct rotation of the vector with no change in the basis (active interpretation), or as a rotation in the basis with no change in the vector (passive interpretation). The components of $\mathbf{x}'$ will be the same either way ($\mathbf{x}' = A\mathbf{x}$), but when comparing both interpretations, the vector and basis would have been rotated in opposite directions. To illustrate these ideas, a simple two-dimensional example is provided in Figure 3.1. If a given reference frame F' is rotated an angle θ around the z axis with respect to F, then the corresponding transformation A will rotate $\mathbf{x}$ an angle $-\theta$ around said axis.

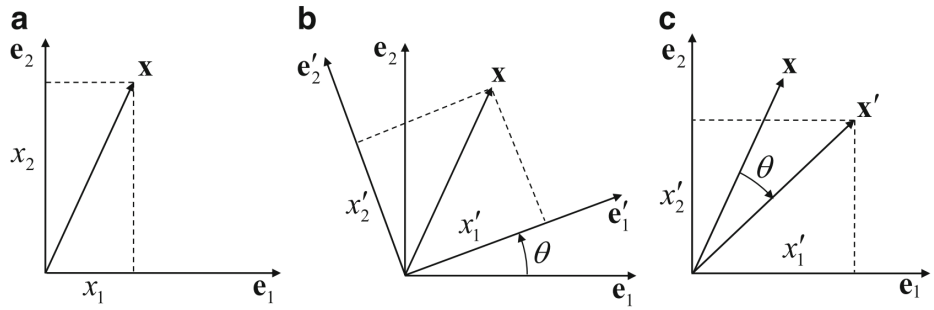


Figure 3.1: Active and passive interpretations of a transformation. (**a**) Initial vector. (**b**) Passive interpretation. (**c**) Active interpretation. Credit: Crassidis & Markley [3].

Unless specified otherwise, all the transformations considered throughout this document are to be interpreted as passive transformations. Note that in typical linear algebra applications, the transformation is generally interpreted in an active sense, and therefore care must be taken regarding the sign of the angles.

Successive rotations can be accomplished by simple matrix multiplication of the respective transformation matrices. For instance, consider that we have three reference frames, F, G and H, and we transform a vector from F to G and then from G to H. This can be effectively represented by

$$\mathbf{x}_H = A_{HG}\,\mathbf{x}_G = A_{HG}\,(A_{GF}\,\mathbf{x}_F) = (A_{HG}\,A_{GF})\,\mathbf{x}_F \tag{3.1}$$

A direct transformation from F to H can therefore be accomplished through the transformation $A_{HF} = A_{HG}\,A_{GF}$. In a passive sense, A_{GF} would represent the rotation from F to G, A_{HF} the rotation from F to H and so on. The composition of rotations will become relevant when considering attitude error.

3.2.1. Euler Angle/Axis Representation

Euler's Theorem states that any rotation is a rotation about a fixed axis. If the generic transformation from reference frame F to F' is denoted by A, where $A\mathbf{x} = \mathbf{x}'$, then Euler's Theorem can be stated algebraically as

$$A\mathbf{e} = \mathbf{e} \tag{3.2}$$

Given an axis $\mathbf{e}$ and angle ϑ, a rotation matrix $A(\mathbf{e}, \vartheta)$ can be constructed as:

$$A(\mathbf{e}, \vartheta) = I_3 - \sin\vartheta\,[\mathbf{e}\times] + (1 - \cos\vartheta)\,[\mathbf{e}\times]^2 \tag{3.3}$$

where I_3 is the identity matrix of order three and $[\mathbf{e}\times]$ is the *cross product matrix* of $\mathbf{e}$, defined by

$$[\mathbf{e}\times] = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{3.4}$$

The concept of cross product matrix will be used extensively throughout this text. Note that it is just a matrix representation of the regular cross product or vector product between two vectors: $\mathbf{x} \times \mathbf{y} = [\mathbf{x}\times]\,\mathbf{y}$.

The attitude matrix appears to depend on four parameters, ϑ and the three components of $\mathbf{e}$, but only three of them are independent owing to the constraint $\|\mathbf{e}\| = 1$. The nine-component attitude matrix has only three independent parameters as well, due to the six scalar constraints derived from the orthogonality constraint $A^TA = I$ (where A^T is the transpose of A).

3.2.2. Rotation Vector Representation

The Euler axis and angle can be combined into the three-component *rotation vector*

$$\boldsymbol{\vartheta} = \vartheta\,\mathbf{e} \tag{3.5}$$

To express the attitude matrix in terms of the rotation vector, the series expansions for $\sin\vartheta$ and $\cos\vartheta$ can be substituted in Equation (3.3), which after some manipulation leads to

$$A(\boldsymbol{\vartheta}) = \exp(-[\boldsymbol{\vartheta}\times]) \tag{3.6}$$

where exp is the matrix exponential.

This three-component attitude representation is useful for small rotations but not so much for large rotations, as it obscures the periodicity of the attitude matrix as a function of ϑ. In particular, in Equation (3.6) it is not obvious that for $\vartheta = 2\pi$, A returns to the identity transformation. The rotation angle can be restricted to $0 \leq \vartheta \leq \pi$, which avoids this problem and gives a 1:1 mapping of rotations. However, when $\vartheta = \pi$ both $\boldsymbol{\vartheta}$ and $-\boldsymbol{\vartheta}$ represent the same rotation, and therefore the rotation vector parametrization of a smoothly-varying attitude can experience a discontinuous jump. Unfortunately, no 1:1 global, three-component attitude representation exists without singular points [18]. This discontinuity can be avoided by giving up the 1:1 mapping by allowing rotation angles greater than π, but this causes other difficulties. Additionally the expense of computing the matrix exponential renders the rotation vector parametrization impractical for numerical computations.

3.2.3. Quaternion Representation

Quaternions were introduced by Hamilton in 1843, and are used nowadays primarily due their utility in describing spatial rotations. A quaternion can be considered as a four component vector with some additional operations defined on it. In this text, a quaternion $\mathbf{q}$ will have a three-vector part $\mathbf{q}_{1:3}$ and a scalar part q_4:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \quad \text{where} \quad \mathbf{q}_{1:3} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.7}$$

following the convention used by Crassidis & Markley [3]. Some authors consider the scalar component to be q_0 and place it on top of the vector part, since it is more historically accurate, but both approaches are valid as long as the quaternion operations are defined accordingly.

The quaternion product will be defined as

$$\bar{\mathbf{q}} \otimes \mathbf{q} = \begin{bmatrix} q_4\, \bar{\mathbf{q}}_{1:3} + \bar{q}_4\, \mathbf{q}_{1:3} - \bar{\mathbf{q}} \times \mathbf{q} \\ \bar{q}_4\, q_4 + \bar{\mathbf{q}} \cdot \mathbf{q} \end{bmatrix} \tag{3.8}$$

and can be represented by matrix multiplication, very much like the vector product:

$$[\mathbf{q}\otimes] = \begin{bmatrix} q_4\, I_3 - [\mathbf{q}_{1:3}\times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} = \begin{bmatrix} \Psi(\mathbf{q}) & \mathbf{q} \end{bmatrix} \tag{3.9}$$

Quaternion multiplication is associative, $\mathbf{q} \otimes (\bar{\mathbf{q}} \otimes \bar{\bar{\mathbf{q}}}) = (\mathbf{q} \otimes \bar{\mathbf{q}}) \otimes \bar{\bar{\mathbf{q}}}$, and distributive, $\mathbf{q} \otimes (\bar{\mathbf{q}} + \bar{\bar{\mathbf{q}}}) = \mathbf{q} \otimes \bar{\mathbf{q}} + \mathbf{q} \otimes \bar{\bar{\mathbf{q}}}$, but not commutative in general, $\mathbf{q} \otimes \bar{\mathbf{q}} \neq \bar{\mathbf{q}} \otimes \mathbf{q}$.

The identity quaternion is defined as

$$\mathbf{I}_q = \begin{bmatrix} \mathbf{0}_3 \\ 1 \end{bmatrix} \tag{3.10}$$

which obeys $\mathbf{I}_q \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{I}_q = \mathbf{q}$.

Additionally, the conjugate $\mathbf{q}^*$ of a quaternion can be obtained by changing the sign of its vector part:

$$\mathbf{q}^* = \begin{bmatrix} -\mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \tag{3.11}$$

The product of a quaternion by its conjugate gives $\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = \|\mathbf{q}\|^2 \mathbf{I}_q$, and it is not difficult to see that $[\mathbf{q}^* \otimes] = [\mathbf{q} \otimes]^T$.

The quaternion product notation may be overloaded to allow for the multiplication of a three-component vector $\mathbf{x}$ and a quaternion, using the following definition:

$$\mathbf{x} \otimes \mathbf{q} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \otimes \mathbf{q} = [\mathbf{x} \otimes] \, \mathbf{q} = \begin{bmatrix} -[\mathbf{x}\times] & \mathbf{x} \\ -\mathbf{x} & 0 \end{bmatrix} \otimes \mathbf{q} \tag{3.12}$$

Here, only the quaternion properties relevant to this work have been introduced. Many more can be found in Section 2.7 from Crassidis & Markley [3] or any other text devoted to quaternion algebra.

For attitude parametrization, only *unit quaternions*, i.e. quaternions with unit norm, are considered. Substituting $\sin\vartheta = 2\sin(\vartheta/2)\cos(\vartheta/2)$ and $\cos\vartheta = \cos(\vartheta/2)^2 - \sin(\vartheta/2)^2$ into Equation (3.3) and defining the quaternion

$$\mathbf{q}(\mathbf{e}, \vartheta) = \begin{bmatrix} \mathbf{e}\sin(\vartheta/2) \\ \cos\vartheta/2 \end{bmatrix} \tag{3.13}$$

gives the quaternion representation of the attitude matrix

$$A(\mathbf{q}) = (q_4^2 - \|\mathbf{q}_{1:3}\|^2)\, I_3 - 2\, q_4\, [\mathbf{q}_{1:3}\times] + 2\, \mathbf{q}_{1:3}\, \mathbf{q}_{1:3}^T \tag{3.14}$$

The rotation of a three-component vector $\mathbf{x}$ can be implemented through quaternion

multiplication:

$$\mathbf{q} \otimes \mathbf{x} \otimes \mathbf{q}^* = \begin{bmatrix} A(\mathbf{q}) \\ 0 \end{bmatrix} \tag{3.15}$$

This can be used to derive the rule for performing successive transformations using quaternions:

$$A(\bar{\mathbf{q}} \otimes \mathbf{q}) = A(\bar{\mathbf{q}})\, A(\mathbf{q}) \tag{3.16}$$

which is just the product of the respective quaternions. Using the quaternion product $\otimes$ defined previously, the order of quaternion multiplication is identical to the order of matrix multiplication. Note that the order may be reversed if other, more classical definitions, are used.

One of the multiple advantages of the quaternion representation is that it expresses the attitude matrix as a homogenous quadratic function of the elements of the quaternion, requiring no trigonometric or other transcendental function evaluations. A consequence of this is that $\mathbf{q}$ and $-\mathbf{q}$ represent the same attitude. This 2:1 mapping of quaternions is a minor annoyance that cannot be solved without introducing discontinuities such as the one described for the rotation vector representation.

In practice, the 2:1 mapping does not represent any major issue and the quaternion representation has been widely adopted as a global attitude representation due to its lack of singular points and the fact that it is much more compact (four components) than the attitude matrix (nine components). Additionally, it is much easier to generate a quaternion from Euler's angle and axis and extract them from it.

3.2.4. Rodrigues Parameters Representation

This attitude representation was first introduced by Olinde Rodrigues in 1840, and later analyzed by J. Willard Gibbs. For this reason the vector of Rodrigues parameters is often called *Gibbs vector* and denoted by $\mathbf{g}$. It is related to the quaternion through

$$\mathbf{g} = \frac{\mathbf{q}_{1:3}}{q_4} \tag{3.17}$$

which has the inverse

$$\mathbf{q} = \frac{\pm 1}{\sqrt{1 + \|\mathbf{g}\|^2}} \begin{bmatrix} \mathbf{g} \\ 1 \end{bmatrix} \tag{3.18}$$

Additionally, $\mathbf{g}$ can be expressed in terms of the Euler axis and angle:

$$\mathbf{g}(\mathbf{e}, \vartheta) = \mathbf{e} \tan(\vartheta/2) \tag{3.19}$$

Substituting Equation (3.18) into Equation (3.14) gives the Rodrigues parameter representation of the attitude matrix:

$$A(\mathbf{g}) = I_3 + 2\,\frac{[\mathbf{g}\times]^2 - [\mathbf{g}\times]}{1 + \|\mathbf{g}\|^2} \tag{3.20}$$

It is clear from Equation 3.17 that $\mathbf{q}$ and $-\mathbf{q}$ map to the same Gibs vector, and therefore the Rodrigues parameters provide a 1:1 mapping of rotations. The singularity in this case is that for 180° rotations, g becomes infinite. For small rotations, however, the Rodrigues parameters provide an excellent representation and its relationship with the attitude quaternion does not involve trigonometric functions.

3.2.5. Modified Rodrigues Parameters

The *modified Rodrigues parameters* (MRPs) are the newest of the commonly employed attitude representations, and slightly differ from the original Rodrigues parameters in that they are related to the quaternion by

$$\mathbf{p} = \frac{\mathbf{q}_{1:3}}{1 + q_4} \tag{3.21}$$

The MRP representation is 2:1 just like the quaternion representation, and has the inverse

$$\mathbf{q} = \frac{1}{1 + \|\mathbf{p}\|^2}\begin{bmatrix} 2\,\mathbf{p} \\ 1 - \|\mathbf{p}\|^2 \end{bmatrix} \tag{3.22}$$

In terms of the Euler axis and angle:

$$\mathbf{p}(\mathbf{e}, \vartheta) = \mathbf{e}\tan(\vartheta/4) \tag{3.23}$$

and the associated attitude matrix is

$$A(\mathbf{p}) = I_3 + \frac{8\,[\mathbf{p}\times]^2 - 4\,(1 - \|\mathbf{p}\|^2)[\mathbf{p}\times]}{(1 + \|\mathbf{p}\|^2)^2} \tag{3.24}$$

3.2.6. Euler Angles Representation

The *Euler angles* representation is used extensively in many fields, and the reader may already be familiarized with some form of it such as the *roll, pitch, yaw* sequence typically used for aircraft. A sequence in which no rotation axis is repeated receives the

name of *asymmetric sequence*. Even though Euler angles present singularities (e.g. gimbal lock), they provide a very intuitive representation of a rotation.

Since the Euler angles representation will not be utilized for this project, their full description and properties will not be explored. The small-angle, first order approximation of the attitude matrix for asymmetric sequences, however, helps to intuitively interpret attitude error representations. For a given sequence $\{i, j, k\}$, with rotation axes $\mathbf{e}_i$, $\mathbf{e}_j$, $\mathbf{e}_k$, and the respective small-angles $\delta\theta_i$, $\delta\theta_j$, $\delta\theta_k$,

$$A_{ijk}(\delta\theta_i, \delta\theta_j, \delta\theta_k) \approx I_3 - [\boldsymbol{\delta\theta}\times] \tag{3.25}$$

where

$$\boldsymbol{\delta\theta} = \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \delta\theta_3 \end{bmatrix} \tag{3.26}$$

3.2.7. Attitude Error Representations

The attitude matrix represents the rotation A_{BR} from some reference frame R to the spacecraft body frame B. Attitude estimation errors are commonly represented as a small rotation $A_{B\hat{B}}$ between B and an estimated body frame $\hat{B}$:

$$A_{BR} = A_{B\hat{B}} A_{\hat{B}R} \tag{3.27}$$

where the estimated attitude is represented by $A_{\hat{B}R}$. Since the error is typically small, $A_{B\hat{B}}$ is expected to be close to the identity matrix.

Attitude errors are most naturally represented in terms of the rotation vector and its small-angles approximation

$$A(\boldsymbol{\delta\vartheta}) = \exp(-[\boldsymbol{\delta\vartheta}\times]) \approx I_3 - [\boldsymbol{\delta\vartheta}\times] + \frac{1}{2}[\boldsymbol{\delta\vartheta}\times]^2 \tag{3.28}$$

Other attitude parametrizations can be used to represent attitude errors, such as the quaternion:

$$A(\boldsymbol{\delta q}) \approx I_3 - 2\,[\boldsymbol{\delta q}_{1:3}\times] + 2\,[\boldsymbol{\delta q}_{1:3}\times]^2 \tag{3.29}$$

the Gibbs vector:

$$A(\boldsymbol{\delta g}) \approx I_3 - 2\,[\boldsymbol{\delta g}\times] + 2\,[\boldsymbol{\delta g}\times]^2 \tag{3.30}$$

or the MRPs:

$$A(\boldsymbol{\delta p}) \approx I_3 - 4\,[\boldsymbol{\delta p}\times] + 8\,[\boldsymbol{\delta p}\times]^2 \tag{3.31}$$

Note that Equation (3.27) may be written in terms of the quaternion as $\mathbf{q} = \boldsymbol{\delta q} \otimes \hat{\mathbf{q}}$, where $\boldsymbol{\delta q}$ is expected to be close to the identity quaternion. The error quaternion can be expressed in terms of other attitude parametrizations as:

$$\boldsymbol{\delta q}(\boldsymbol{\delta\vartheta}) \approx \begin{bmatrix} \boldsymbol{\delta\vartheta}/2 \\ 1 - \|\boldsymbol{\delta\vartheta}\|^2/8 \end{bmatrix} \tag{3.32}$$

or

$$\boldsymbol{\delta q}(\boldsymbol{\delta g}) \approx \begin{bmatrix} \boldsymbol{\delta g} \\ 1 - \|\boldsymbol{\delta g}\|^2/4 \end{bmatrix} \tag{3.33}$$

or

$$\boldsymbol{\delta q}(\boldsymbol{\delta p}) \approx \begin{bmatrix} 2\,\boldsymbol{\delta p} \\ 1 - 2\,\|\boldsymbol{\delta p}\|^2 \end{bmatrix} \tag{3.34}$$

All of these error representations are equivalent up to second order if the following identification is made:

$$\boldsymbol{\delta\vartheta} = 2\,\boldsymbol{\delta q}_{1:3} = 2\,\boldsymbol{\delta g} = 4\,\boldsymbol{\delta p} \tag{3.35}$$

and agree to first order with the Euler angles' small-angles approximation in Equation (3.25). Since for most applications only the first order term is relevant, these error representations share an exceptionally intuitive interpretation given by the Euler angles. For instance, the first component will represent a rotation around the x axis, the second component a rotation around the y axis, and the third component a rotation around the z axis. Of course, the rotation angles will need to be interpreted in radians.

3.3. Attitude Kinematics

The kinematics equations for every attitude representation reviewed in the previous section can be derived. Here, the focus will be on quaternion kinematics, since the quaternion is the global attitude representation of choice for most current Kalman filtering applications.

Kinematic equations are differential equations that relate a certain attitude representation with the *angular velocity* vector $\boldsymbol{\omega}(t)$, which is defined as the time rate at which an object rotates around an instantaneous axis. A more formal definition can be stated

in terms of the rotation vector

$$\boldsymbol{\omega}(t) = \lim_{\Delta t \to 0} \frac{\boldsymbol{\vartheta}(t + \Delta t) - \boldsymbol{\vartheta}(t)}{\Delta t} \tag{3.36}$$

where $\boldsymbol{\omega}(t)$ represents the instantaneous rotation at time t of the body frame around the reference frame (the frame-specifying subscripts have been omitted for simplicity). Note that direction of $\boldsymbol{\omega}(t)$ is perpendicular to the instantaneous plane of rotation, and since the rotation vector is specified in radians, its units will be rad/s (assuming time is measured in seconds).

The kinematics equation for the quaternion representation, which will not be derived here, is

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \left[\boldsymbol{\omega}(t) \otimes\right] \mathbf{q}(t) \tag{3.37}$$

The fact that $[\boldsymbol{\omega}(t) \otimes]$ is a skew-symmetric matrix ($[\boldsymbol{\omega}(t) \otimes]^T = -[\boldsymbol{\omega}(t) \otimes]$) ensures that Equation (3.37) preserves the quaternion unit norm.

3.3.1. Integration methods

Once the appropriate differential equations for attitude propagation have been established, it is necessary to determine a method for solving them. An exact, closed-form solution will not be available in most cases, and therefore two methods are discussed in this section: an approximate closed-form solution assuming constant angular velocity, and direct numerical integration.

Approximate Closed-Form Solution for the kinematics equation

Equation (3.37) is an homogeneous matrix differential equation. The standard approach for solving equations of these form is to determine the *state transition matrix* $\Phi(t, t_0)$ which "maps" an initial state $\mathbf{q}(t_0)$ into the current state as [19]

$$\mathbf{q}(t) = \Phi(t, t_0)\, \mathbf{q}(t_0) \tag{3.38}$$

a differential equation for determining $\Phi(t, t_0)$ can be developed by substituting Equation (3.38) and its derivative into Equation (3.37), which gives

$$\dot{\Phi}(t, t_0) = \frac{1}{2} \left[\boldsymbol{\omega}(t) \otimes\right] \Phi(t, t_0) \tag{3.39}$$

with the initial condition $\Phi(t_0, t_0) = I_4$. Only when the angular velocity is constant ($\boldsymbol{\omega}(t) = \boldsymbol{\omega} = \text{constant}$) can a practical closed-form solution of Equation (3.39) be obtained:

$$\Phi(t, t_0) = \exp\left(\frac{1}{2}\left[\boldsymbol{\omega}\otimes\right]\Delta t\right) \tag{3.40}$$

where $\Delta t = t - t_0$. After some manipulation of the series expansion for the matrix exponential (and introducing the series expansions for trigonometric functions), the previous equation can be rewritten in a more convenient form for numerical computation [17]:

$$\Phi(t, t_0) = \cos\left(\frac{1}{2}\|\boldsymbol{\omega}\|\Delta t\right) I_4 + \frac{1}{\|\boldsymbol{\omega}\|}\sin\left(\frac{1}{2}\|\boldsymbol{\omega}\|\Delta t\right)\left[\boldsymbol{\omega}\otimes\right] \tag{3.41}$$

which avoids the computation of an infinite series. This expression corresponds to an orthogonal rotation, which retains the normalization of the propagated attitude quaternion. Because of its simplicity, this closed-form solution has been widely used as the kinematic integrator for onboard computers.

In general, the angular velocity vector $\boldsymbol{\omega}(t)$ can be measured on board using a gyroscope. It is assumed that the gyro data is sampled at a fixed rate and that $\boldsymbol{\omega}(t)$ is constant over the sampling interval. In most cases this is a reasonable approximation, since modern spacecraft gyros use sampling rates ranging from 10 to 1000 Hz. Then, considering a time interval $[t_k,\, t_{k+1}]$ which is split into N samples (including one at t_k but not at t_{k+1}),

$$\mathbf{q}(t_{k+1}) = \Phi(\tau_{N+1}, \tau_N)\cdots\Phi(\tau_{j+1}, \tau_j)\cdots\Phi(\tau_2, \tau_1)\,\mathbf{q}(t_k) \tag{3.42}$$

where

$$\tau_j = t_k + (j-1)\,\frac{t_{k+1} - t_k}{N}, \quad j = 1, 2, ..., N \tag{3.43}$$

Additionally,

$$\Phi(\tau_{j+1}, \tau_j) = \cos\left(\frac{1}{2}\|\boldsymbol{\omega}_j\|\Delta\tau\right) I_4 + \frac{1}{\|\boldsymbol{\omega}_j\|}\sin\left(\frac{1}{2}\|\boldsymbol{\omega}_j\|\Delta\tau\right)\left[\boldsymbol{\omega}_j\otimes\right] \tag{3.44}$$

where $\Delta\tau = \tau_{j+1} - \tau_j$ corresponds to the sampling time interval of the gyro and $\boldsymbol{\omega}_j = \boldsymbol{\omega}(\tau_j)$ is the sampled angular velocity vector at time τ_j ($\boldsymbol{\omega}(t) = \boldsymbol{\omega}_j\ \forall\, t \in [\tau_j,\, \tau_{j+1})$).

Gyro data is typically sampled with a zero-order hold, which corresponds to the above description. Thus, it is intrinsically discrete, and even though some information about the derivatives of the angular velocity may be extracted, the integration method described in this section is perfectly appropriate.

Direct Integration

For more precise results, one can resort to numerical algorithms to solve the kinematics equation. Since it is a system of nonstiff ordinary differential equations, explicit predictor-corrector methods such as those in the Runge-Kutta family are well suited. Multi-step methods in the Adams-Bashford or Adams-Moulton families can also be used.

Numerical methods in some way "estimate" the derivatives of $\Phi(t, t_0)$, and generally are capable of producing extremely precise solutions. However, their precision and convergence heavily depend on the *step size* of the numerical integration, which in this case is fixed to be equal or greater than the sampling time. For typical gyro sampling rates this is not an issue, but it is something to take into account as it could potentially lead to the divergence of the attitude solution.

Another disadvantage of numerical algorithms is that they don't retain the normalization of the quaternion, and therefore it needs to be normalized by brute force at every step. This, along with the rest of operations, incur in computation times which exceed that of the approximate method described in the previous section. Furthermore, the introduction of noise in the measurement of $\boldsymbol{\omega}(t)$ leads to a significant deterioration of the precision, to the point where the computed attitude solution is indistinguishable from the one obtained using Equation (3.42).

For onboard applications where reliability and computational efficiency are essential, the approximate closed-form solution has for a long time been the integration method of choice. Nonetheless, direct integration has its place in this project, as it can provide a more precise "true" trajectory against which different attitude determination methods can be compared in a simulated environment. This "true" trajectory is obtained by integrating a given $\boldsymbol{\omega}(t)$ which can then also be used to generate the gyro measurements (covered in detail in Chapter 4).

3.3.2. Attitude error kinematics

Usually, only an estimation of the angular velocity vector is available, which will be denoted by $\hat{\boldsymbol{\omega}}(t)$. Note that the estimated value of the angular velocity is not necessarily the one measured by the gyro[1]. The error angular velocity $\boldsymbol{\delta\omega}$ can then be defined as

$$\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \tag{3.45}$$

[1]Using certain attitude determination methods, such as the Kalman Filter, the gyro bias can also be estimated and then subtracted from the measured angular velocity.

where the time dependency has been omitted for simplicity.

The exact analytical attitude error kinematics equations for most of the known attitude representations can be found in [20]. However, in the context of this work, only a first order approximation is required (which will become the basis of the Extended Kalman Filter). If we denote any of the attitude error representations in Equation (3.35) by $\mathbf{a}$, all of them agree to first order with

$$\dot{\mathbf{a}} = -[\hat{\boldsymbol{\omega}}\times]\,\mathbf{a} + \boldsymbol{\delta\omega} \tag{3.46}$$

3.4. Statistics Refresher

Many attitude estimation methods, including the Kalman Filter, deal with random noise of known statistics. Additionally, propagating an estimated (uncertain) state requires applying a nonlinear transformation to a certain probability distribution. Here, a short statistics refresher covering these concepts will be provided. The main references for this section are [21–23].

3.4.1. Random Variables and the Gaussian Distribution

Let X be a scalar, continuous random variable. Its *cumulative distribution function* $F(x)$ represents the probability that $X \leq x$, which may be written as $F(x) = P\{X \leq x\}$. $F(x)$ can be computed from

$$F(x) = \int_{-\infty}^{x} f(u)du \tag{3.47}$$

where $f(x)$ is the *probability density function* (PDF) of X.

Now, let $g(X)$ be a function of the random variable X. The *expected value* of $g(X)$ is defined by

$$E\{g(X)\} = \int_{-\infty}^{\infty} g(x)f(x)dx \tag{3.48}$$

In particular, the expected value or *mean* of X is

$$\mu_X = E\{X\} = \int_{-\infty}^{\infty} xf(x)dx \tag{3.49}$$

and its *variance* is defined by

$$var(X) = E\{(X-\mu_X)^2\} = \int_{-\infty}^{\infty} (x-\mu_X)^2 f(x)dx \tag{3.50}$$

The variance of a random variable quantifies its variation relative to the expected value, and it is related to the *standard deviation* σ_X in that $\sigma_X^2 = var(X)$. Note that the standard deviation has the same units as the mean. In cases where the meaning is clear, the subscript indicating the random variable may be dropped.

The kth *moment* of X about a value c is defined by

$$E\{(X-c)^k\} = \int_{-\infty}^{\infty} (x-c)^k f(x) dx \tag{3.51}$$

The mean of X then corresponds to its first *raw moment* (i.e moment about zero) and the variance to its second *central moment* (i.e. moment around the mean).

The Gaussian distribution or normal distribution[1] is the most commonly used distribution in statistics. This stems from the *central limit theorem*, which shows that the sum of many independent and identically distributed random variables tends to a normal distribution. In general, sensor noise can be considered to be Gaussian distributed, as it is the superposition of many small random effects.

The notation $X \sim N(\mu, \sigma^2)$ is used to indicate that a random variable X has the Gaussian PDF described by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{3.52}$$

Thus, a Gaussian random variable is completely described by two parameters μ and σ, and it can be shown that they correspond to its mean and standard deviation respectively. Additionally, since the PDF in Equation (3.52) is symmetric, its odd central moments are zero.

A convenient way to represent the uncertainty of a random variable is through a *confidence interval*, which is the range of values the variable is expected to lie in with a certain *confidence level*. The confidence level represents the probability of said range containing the true value. For example, a 90% confidence interval would imply that the true value of the variable is within the specified range with a probability (given as a percentage) of 90%. In scientific applications, confidence intervals are typically defined using multiples of the the standard deviation. For instance, the most commonly used confidence intervals of a scalar Gaussian random variable correspond to

- 1σ interval: $P\{X \in [\mu-\sigma, \mu+\sigma]\} = 68.3\%$

[1] These terms will be used interchangeably.

- 2σ interval: $P\{X \in [\mu - 2\sigma, \mu + 2\sigma]\} = 95.4\%$
- 3σ interval: $P\{X \in [\mu - 3\sigma, \mu + 3\sigma]\} = 99.7\%$

Empirically, 99.7% probability is treated as near certainty and the 3σ interval will be used extensively throughout this project.

The previous concepts can be extended to multivariate continuous random variables. For a random vector $\mathbf{X} = [X, Y]^T$, the *joint cumulative distribution function*, is defined as

$$F(x, y) = P\{X \leq x \text{ and } Y \leq y\} = \int_{-\infty}^{x} \int_{-\infty}^{y} f(u, v) du dv \tag{3.53}$$

where $f(x, y)$ is the *joint probability density function*. The expected value of $g(X, Y)$ may therefore be computed as

$$E\{g(x, y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) f(x, y) dx dy = \int_{\mathbb{R}^2} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{3.54}$$

The mean of a random vector $\mathbf{X} = [X_1, X_2, ..., X_n]^T$ is defined by

$$\boldsymbol{\mu} = E\{\mathbf{X}\} = \int_{\mathbb{R}^n} \mathbf{x} f(\mathbf{x}) d\mathbf{x} \tag{3.55}$$

but the analogous to Equation (3.50) becomes

$$P = E\{(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T\} = \int_{\mathbb{R}^n} (\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T f(\mathbf{x}) d\mathbf{x} \tag{3.56}$$

which does not exactly correspond to the variance, and is instead called the *covariance matrix*[1] of $\mathbf{X}$. In probability theory and statistics, covariance is originally defined as $cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\}$, and measures the joint variability of two scalar random variables. Note that $cov(X, X) = var(X)$. The elements of P can therefore be computed using

$$P_{ij} = cov(X_i, X_j), \quad i, j = 1, 2, ..., n \tag{3.57}$$

where P_{ij} represents the (i, j)-th element of P. In particular, when $i = j$ (that is, elements on the main diagonal), $P_{ii} = \sigma_i^2$.

The notation $\mathbf{X} \sim N(\boldsymbol{\mu}, P)$ is used to indicate that $\mathbf{X}$ has the multivariate Gaussian

[1] Also known as the auto-covariance matrix, variance-covariance matrix, or simply covariance.

or normal density function described by

$$f(\mathbf{X}) = \frac{1}{\sqrt{(2\pi)^n \det(P)}} \exp\left(-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T P^{-1}(\mathbf{X}-\boldsymbol{\mu})\right) \tag{3.58}$$

where $\det(P)$ is the determinant of P. Again, the Gaussian random vector $\mathbf{X}$ is completely defined by its expected value $\boldsymbol{\mu} \in \mathbb{R}^n$ and its covariance matrix $P \in \mathbb{R}^{n\times n}$.

Confidence intervals become confidence *regions* in $\mathbb{R}^n$, defined by $P\{\mathbf{X} \in \Omega\} = P_\Omega$. The shape of these regions is a multidimensional ellipsoid described by

$$(\mathbf{X}-\boldsymbol{\mu})^T P^{-1}(\mathbf{X}-\boldsymbol{\mu}) = d^2 \tag{3.59}$$

where d is known as the *Mahalanobis distance* and depends on P_Ω. The eigenvalues of P determine the size of the ellipsoid, whereas the eigenvectors provide the direction of its axes. Usually, only projections of the ellipsoid into two-dimensional ellipses are represented. If we denote the ellipse as $\Sigma = \Sigma(d)$, then the confidence regions are

- 1σ region: $P\{\mathbf{X} \in \Sigma(\sigma)\} = 39.4\%$
- 2σ region: $P\{\mathbf{X} \in \Sigma(2\sigma)\} = 86.5\%$
- 3σ region: $P\{\mathbf{X} \in \Sigma(3\sigma)\} = 98.9\%$

3.4.2. Propagating Means and Covariances Through Nonlinear Transformations

Consider the following problem. A random vector $\mathbf{x}$ has mean $\bar{\mathbf{x}}$ and covariance P_x, and a second random vector, $\mathbf{y}$, is related to $\mathbf{x}$ through the nonlinear transformation

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{3.60}$$

where $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^n$. The problem is to estimate $\bar{\mathbf{y}}$ and P_y. Note that only partial distribution information of $\mathbf{x}$ (in the form of mean and covariance) is being considered, and no assumptions about its complete PDF are being made.

A first idea is to generate the Taylor series expansion of Equation (3.60). Let $\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\delta}\mathbf{x}$ where $\boldsymbol{\delta}\mathbf{x}$ is a zero mean random variable with covariance P_x. Expanding $\mathbf{h}$ about $\bar{\mathbf{x}}$,

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\bar{\mathbf{x}} + \boldsymbol{\delta}\mathbf{x}) = \mathbf{h}(\bar{\mathbf{x}}) + \nabla\mathbf{h}\boldsymbol{\delta}\mathbf{x} + \frac{1}{2}\nabla^2\mathbf{h}\boldsymbol{\delta}\mathbf{x}^2 + \frac{1}{3!}\nabla^3\mathbf{h}\boldsymbol{\delta}\mathbf{x}^3 + \cdots \tag{3.61}$$

where, for the sake of simplicity, $\nabla^i \mathbf{h}\boldsymbol{\delta}\mathbf{x}^i$ represents the ith order term in the multidimensional Taylor series (for instance, $\nabla\mathbf{h}$ is the Jacobian of $\mathbf{h}$ evaluated at $\bar{\mathbf{x}}$). Taking expectations, it can be shown that

$$\bar{\mathbf{y}} = E\{\mathbf{y}\} = \mathbf{h}(\bar{\mathbf{x}}) + \frac{1}{2}\nabla^2\mathbf{h}P_x + \frac{1}{6}\nabla^3\mathbf{h}E\{\boldsymbol{\delta}\mathbf{x}^3\} + \cdots \tag{3.62}$$

$$\begin{aligned} P_y &= E\{(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^T\} \\ &= \nabla\mathbf{h}P_x(\nabla\mathbf{h})^T + \frac{1}{4}\nabla^2\mathbf{h}E\{\boldsymbol{\delta}\mathbf{x}^3\}(\nabla\mathbf{h})^T + \frac{1}{2}\nabla\mathbf{h}E\{\boldsymbol{\delta}\mathbf{x}^3\}(\nabla^2\mathbf{h})^T + \cdots \end{aligned} \tag{3.63}$$

Next, a few strategies to approximate these values are discussed.

Linear Approximation

If only the first order elements of Equations (3.62-3.63) are considered (effectively linearizing $\mathbf{h}$), then

$$\bar{\mathbf{y}} \approx \mathbf{h}(\bar{\mathbf{x}}) \tag{3.64}$$

$$P_y \approx \nabla\mathbf{h}P_x(\nabla\mathbf{h})^T \tag{3.65}$$

In the special case that $\mathbf{x}$ is Gaussian distributed, $E\{\boldsymbol{\delta}\mathbf{x}^i\} = 0$ if i is odd (due to the symmetry of the distribution). This implies that the linear approximation predicts the mean correctly to first order and the covariance to third order.

The Unscented Transform

In 1994, Jeffrey Uhlmann noted that *with a fixed number of parameters, it should be easier to approximate a given distribution than it is to approximate an arbitrary nonlinear function* [24]. That is, he suggested that a better approach than the previous linearization method would be to use the exact nonlinear transformation applied to an approximate probability distribution. This can be accomplished by deterministically choosing a set of weighted *sigma points*, so that certain properties of these points (such as their mean and covariance) match those of $\mathbf{x}$. Each point undergoes the nonlinear transformation, and the estimated mean and covariance of $\mathbf{y}$ will then correspond to those of the transformed set of points. This idea is referred to as an *unscented transform* (UT).

In general, any set of sigma points that matches the available information of the initial distribution is as good as any. However, there are some considerations that may give preference to some sigma sets over others. Suppose that only mean and covariance information is available for some n-dimensional random vector. This information can be captured in a *simplex* set of $n+1$ sigma points, which utilizes the minimum number

of points required to match the mean and covariance but generally introduces nonzero higher order moments. These higher order moments may fortuitously align with those of the true distribution, or on the contrary be in the opposite direction potentially harming the results.

In the original paper [22], Uhlmann proposes a symmetric set of $2n+1$ points. If the true distribution is also symmetric (e.g. Gaussian), then this set will be more accurate than the simplex set. If not, the simplex set is still likely to produce the poorest results in the worst case (when higher order moments are in the opposite direction). Following this logic, "neutral" higher order moments (i.e. close to zero) are preferred.

The *scaled unscented transform* [23] takes the previous idea one step further by providing a scaling parameter α, which can be made arbitrarily small in order to minimize higher order moments (while maintaining the true mean and covariance). Additionally, it provides a mechanism to minimize the error in the fourth order term of Equation (3.63) and guarantees that the estimate of P_y remains positive semidefinite, which is essential in filtering applications.

The point selection algorithm for the scaled UT is given as follows. First, compute the following symmetric set of $2n+1$ sigma points with associated weights:

$$\begin{aligned}
\mathcal{X}_0 &= \bar{\mathbf{x}} \\
\mathcal{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{(n+\lambda)P_x}\right)_i, \quad i = 1, 2, ..., n \\
\mathcal{X}_i &= \bar{\mathbf{x}} - \left(\sqrt{(n+\lambda)P_x}\right)_{i-n}, \quad i = n+1, n+2, ..., 2n \\
w_0 &= \frac{\lambda}{n+\lambda} \\
w_i &= \frac{1}{2(n+\lambda)}, \quad i = 1, 2, ..., 2n
\end{aligned} \tag{3.66}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter. α determines the spread of the sigma points around $\bar{\mathbf{x}}$ and is usually set to a small positive value (e.g. 10^{-3}). κ is a secondary scaling parameter which is usually set to 0. $\left(\sqrt{(n+\lambda)P_x}\right)_i$ is the ith row of the matrix square root, which can be computed using the Cholesky decomposition. Then, propagate the sigma points through the nonlinear function,

$$\mathcal{Y}_i = \mathbf{h}(\mathcal{X}_i), \quad i = 0, 2, ..., 2n \tag{3.67}$$

The mean and covariance of $\mathbf{y}$ are approximated using a weighted sample mean and

covariance (the latter with an added term) for the propagated sigma points,

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2n} w_i \mathcal{Y}_i \tag{3.68a}$$

$$P_y \approx \sum_{i=0}^{2n} w_i (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T + (1 - \alpha^2 + \beta)(\mathcal{Y}_0 - \bar{\mathbf{y}})(\mathcal{Y}_0 - \bar{\mathbf{y}})^T \tag{3.68b}$$

where β is used to incorporate prior knowledge of the distribution of $\mathbf{x}$, in order to minimize the error in the fourth order term of Equation (3.63) (for Gaussian distributions, $\beta = 2$ is optimal).

It is often stated that for non-Gaussian inputs, these approximations are accurate to at least second order[1]. In the special case that $\mathbf{x}$ is Gaussian distributed, both the mean and covariance would then be accurate to third order. Typically, the UT predicts the mean more accurately than a simple linearization of $\mathbf{h}$, and the covariance at least as accurately. An appropriate choice of α and β can further minimize the error in fourth order terms.

Monte-Carlo Sampling

Monte-Carlo simulations are computational methods that rely on the use of random sampling and statistics. They give approximate (numerical) solutions to problems which are hard to solve using deterministic mathematical approaches. In the context of applying a nonlinear transformation to a probability distribution, performing a Monte-Carlo simulation consists on the following steps. First, a large amount of points are randomly sampled from the prior distribution (so that its properties are matched as accurately as possible). These points are then propagated through the nonlinear function, and their sample mean and covariance approximate the mean and covariance of the transformed distribution.

However, there are a some drawbacks to this method. First, the relationship between the number of points and the accuracy of the resulting approximation is not clearly defined and depends on both the prior distribution and the nature of the nonlinear function. Secondly, propagating a large number of points (typically being in the order of tens of thousands to a million) comes with an increased computational burden. Typically, a compromise between accuracy and computational cost is achieved.

Nevertheless, the resulting approximation tends to be much more accurate than

[1] This, however, has been shown to not always be the case [25].

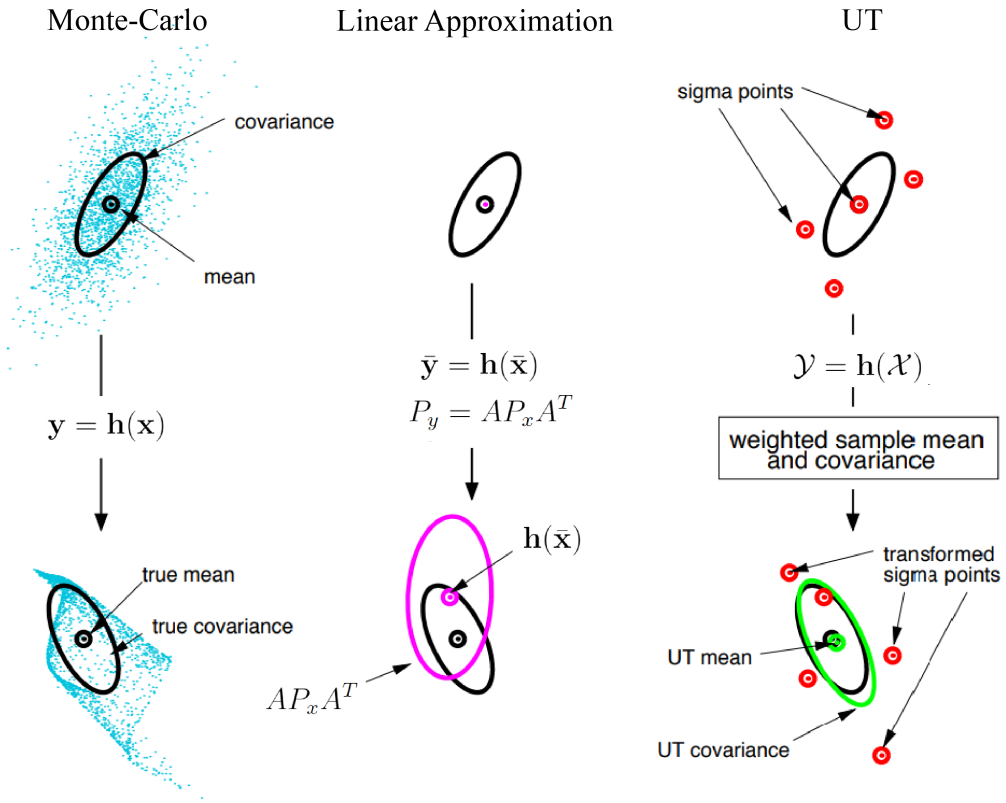


Figure 3.2: Mean and covariance propagation using Monte-Carlo sampling, a linear approximation and the UT. Adapted from Wan & van der Merwe [4].

those obtained through linearization or the UT[1], as higher order moments of the prior distribution are correctly captured. Therefore, even though Monte-Carlo sampling requires increased computational times, it can be used to compute the "true" propagated mean and covariance to which other approximations may be compared. In Figure 3.2, a graphical representation of the previous approximation methods is presented.

3.4.3. Stochastic Processes

A stochastic process, also called a random process, is simply a random variable that changes with some parameter, generally time. The *autocorrelation function* of a continuous-time process $\boldsymbol{\nu}(t)$ (i.e. the random variable $\boldsymbol{\nu}$ at each time is continuous) is defined by

$$R_v(t_1, t_2) = E\{\boldsymbol{\nu}(t_1)\boldsymbol{\nu}^T(t_2)\} \tag{3.69}$$

[1]Although the UT may superficially resemble a Monte-Carlo method, no random sampling is used, and in consequence only a small number of points ($2n+1$ for an nth dimensional space) are required.

and quantifies the similarity of the random process to itself at two different times.

A random process is *stationary* if its distribution is independent of time. If a random process is stationary, its expected value will be time invariant and its autocorrelation function will only depend on the time difference $\tau = t_2 - t_1$ (i.e. $R_v(t_1, t_2) = R_v(\tau)$). The *power spectral density* (PSD) of a stationary random process $\boldsymbol{\nu}(t)$ is the Fourier transformation of its autocorrelation function:

$$S_v(j\omega) = \int_{-\infty}^{\infty} R_v(\tau) e^{-j\omega\tau} d\tau \tag{3.70}$$

One stochastic process that is particularly useful for modeling purposes is *white noise*. A continuous-time stochastic process $\boldsymbol{\nu}(t) \in \mathbb{R}^n$ is referred to as white noise if its PSD is constant:

$$S_v(j\omega) = \sigma_v^2 I_n \tag{3.71}$$

Then, by Equation (3.70), its correlation function is

$$R_v(t) = \sigma_v^2 \delta(\tau) I_n \tag{3.72}$$

where $\delta(\tau)$ is the Dirac delta function defined as

$$\delta(\tau) = 0 \quad \text{if } \tau \neq 0, \qquad \int_{-\infty}^{\infty} \delta(\tau) d\tau = 1 \tag{3.73}$$

and has units of Hz[1]. An implication of Equation (3.72) is that white noise has zero mean. Additionally, it reflects that $\boldsymbol{\nu}(t_1)$ is independent from $\boldsymbol{\nu}(t_2)$ for all $t_1 \neq t_2$. If this is not the case (the PSD is not constant), the process is referred to as *colored* noise.

A white noise process can have any probability distribution; however, the Gaussian distribution is often assumed (recall the central limit theorem). The notation $\boldsymbol{\nu} \sim N(0, S)$ will sometimes be used to describe a continuous-time, Gaussian, white noise process $\boldsymbol{\nu}$. In this case, S is to be interpreted as the PSD and not the covariance matrix.

[1]Reasoning through the units of σ_v^2 often causes confusion. From the PSD perspective, σ_v^2 represents the power per unit frequency interval. For example, if $\boldsymbol{\nu}$ is an angular rate in rad/s, then σ_v^2 has units of $(\text{rad}^2/\text{s}^2)/\text{Hz} = \text{rad}^2/\text{s}$. The units of σ_v are then $\text{rad}/\sqrt{\text{s}}$ or $\text{rad}/\text{s}/\sqrt{\text{Hz}}$.

4 Sensor Selection and Modeling

The selection of an appropriate sensor suite is essential for achieving accurate and reliable attitude estimation. In the case of small spacecraft, this task is particularly challenging due to the stringent constraints imposed by the satellite's size, weight, power, and economic budget. These limitations will be implicitly taken into consideration by restricting the sensor selection to components specifically designed for small spacecraft.

In this chapter, sensor selection and modeling will be covered. First, a versatile sensor suite is proposed, integrating multiple sensors from Solar MEMS Technologies' catalog. Then, a brief section on measurement error sources is included, providing some insight on their nature and assessing some practical considerations. Next, based on this information, several measurement models for the different sensors are presented. These models are essential for the integration of sensor data in the attitude estimation algorithms and, additionally, may be used to perform the simulations. The chapter ends with an overview of the technical specifications of the selected sensor suite.

4.1. Sensor Suite Selection

The sensor selection depends on several mission requirements, which typically include:

- **Attitude determination accuracy:** The required accuracy of the attitude determination depends on the mission objectives. Some missions require high-precision attitude determination, while others may be able to tolerate lower accuracy.
- **Attitude determination frequency:** The frequency at which the attitude determination needs to be updated. For some missions, frequent updates are required, while for others, less frequent updates may be sufficient.
- **Operational environment:** The operational environment of the spacecraft can impact the selection of the sensor suite. For example, magnetic fields, radiation, and thermal conditions can affect the performance of certain sensors, such as magnetometers and gyroscopes.

- **Cost:** The cost of the sensor suite can be a limiting factor in some missions, particularly those with smaller budgets.
- **Size, weight, and power constraints:** The sensor suite must be designed to fit within the size, weight, and power constraints of the spacecraft. Small satellites, such as CubeSats, may have more limited resources in terms of size, weight, and power, which can impact the selection of the sensor suite.
- **Redundancy:** Some missions may require redundant sensor suites to ensure the reliable operation of the attitude determination system in the event of a sensor failure.
- **Integration with other spacecraft systems:** The sensor suite must be compatible with other spacecraft systems, such as the attitude control system and communication system.

For this project, however, the design flow shall be reversed. Instead of designing an ADS to satisfy a given set of mission requirements, an attitude determination module integrating multiple sensors from Solar MEMS Technologies' catalog is proposed. The goal is then to analyze the achievable performance, and will be up to the spacecraft systems engineer to decide whether it fulfills their mission requirements. Nonetheless, the sensor selection will be carried out keeping the previous points in mind, with the intent of designing a versatile module suitable for a wide range of missions. This has proved to be a reasonable strategy, given the large number of modular ADCS solutions currently available on the market (see Chapter 2).

There are a variety of sensors that can be used for small spacecraft in LEO. Next, some of their respective advantages and disadvantages are briefly discussed:

- **Gyroscopes:** Modern gyros are almost exclusively used in strap-down mode (solidly attached to the spacecraft). They are reliable and accurate, but can be expensive and may require frequent calibration.
- **Sun sensors:** Sun sensors determine the spacecraft's orientation with respect to the sun. They are low cost and can provide accurate measurements, but they require sunlight and can only be used when the sun is visible.
- **Star trackers:** Star trackers compute the spacecraft's attitude with respect to a celestial reference frame. They provide highly accurate measurements and can operate in a nearly continuous fashion, but they can be expensive and require significant processing power.

- **Magnetometers:** Magnetometers measure the strength and direction of the Earth's magnetic field, and are relatively small, lightweight, rugged, and inexpensive. In LEO they can provide a continuous, reliable reference for attitude determination, but they are susceptible to interference from other magnetic sources (current loops in solar arrays, electric motors, payload instruments, magnetorquers, etc.).
- **Earth horizon sensors:** Earth horizon sensors compute the nadir vector by detecting points on the Earth's horizon, and therefore are especially suited for Earth-pointing spacecraft. They are low cost and can provide relatively accurate measurements, but they require a clear view of the horizon and may be affected by atmospheric conditions.

Other sensors such as GPS receivers and accelerometers may also be used to be used to complement the previous options, but they present serious practical limitations and have not experienced widespread use.

A combination of multiple sensors is typically used to provide redundancy and improve the accuracy and reliability of the attitude determination system. Historically, magnetometers, Sun sensors and Earth horizon sensors have been used in conjunction with precise gyroscopes. Modern star trackers, able to track multiple stars simultaneously, appeared around 1990 and have since then been extensively used in missions with high-precision pointing requirements. Solar MEMS Technologies manufacture a couple of sensors oriented towards nanosatellite platforms such as CubeSats. Namely, a moderately fine Sun sensor (NanoSSOC) and an Earth horizon sensor (HSNS). Additionally, a star tracker (STNS) is currently under development.

To gain some insight into which sensor combinations are better suited for this project, Table 2.1 (showing the sensors utilized by the different commercial ADCS modules) may be analyzed. Out of the 16 models (including the ADS-only solution), 15 implement a three-axis magnetometer (TAM), 9 include gyros, 9 Solar sensors, 6 star trackers, 3 Earth horizon sensors, 2 GPS receivers and only one accelerometers. It is clear that the TAM is an indispensable element in any attitude determination sensor suite, given its reliability, relative low cost and almost insignificant power demand. The next most popular sensors are gyros and Solar sensors, followed closely by star trackers for high-precision applications. Moreover, Earth horizon sensors, GPS receivers and accelerometers are seldom used.

Following the previous discussion, a sensor suite composed of Solar MEMS' NanoSSOC and future STNS, in conjunction with commercial-off-the-shelf (COTS) gyros and TAM, has the potential to provide versatility, redundancy and high performance. The

HSNS will be omitted since it does not offer any additional advantages to the previous selection and is not as flexible (it would require special customization depending on the orbit altitude). Regarding the gyro, while it is true that *gyroless* configurations exist, they require precise knowledge about the spacecraft mass distribution in order to successfully implement a Kalman filter and carry out the attitude estimation. In order to follow a more flexible approach, with potentially no drawbacks, gyros will be considered within the sensor suite.

The TAM selection is relatively straightforward, since its precision does not depend as much on the specific technology used, but rather on the Earth's magnetic field modeling error, orbit uncertainty, magnetic interference, or orthogonality and alignment (among other error sources, which will be covered in more detail in Section 4.2). The MM200 model from AAC Clyde Space will be selected as a low cost option, specifically designed for nanosatellite platforms. On the other hand, gyro performance is highly dependent on the used technology. In the following section, several options are studied.

4.1.1. Gyro Selection

Gyros can be classified by the physical mechanisms they use: spinning-mass gyros, optical gyros (FOGs and RLGs), or Coriolis vibratory gyros (CVGs). In the early spacecraft, spinning-mass gyroscopes were commonly used, which relied on the conservation of angular momentum to maintain the spacecraft's orientation. Later, fiber optic gyroscopes and ring laser gyroscopes were also developed, which use the principles of interferometry to measure rotation rates with high accuracy. Today, CVGs manufactured with micro-electromechanical system (MEMS) are commonly used in small spacecraft due to their small size, low power consumption, and low cost (albeit with low performance and short lifetime compared to other gyros).

MEMS gyros have been successfully flown in numerous space missions, making them a trustworthy and established option for small spacecraft. FOGs have also been occasionally employed, usually providing superior performance at a mass and cost penalty. Even though both options may be suitable for this project, MEMS gyroscopes offer more versatility and a greater number of COTS models are available on the current market. Therefore, they will be the technology of choice for the attitude determination module.

In [11], a comprehensive review of currently available gyroscopes (and other sensors) for small spacecraft is available. Inertial sensors can be packaged in a variety of ways, ranging from single-axis devices such as individual gyroscopes or accelerometers, to Inertial Measurement Units (IMUs), which include multiple axes of both gyroscopes and

accelerometers to enable six degrees of freedom inertial propagation. Some vendors also provide packages that include additional sensors like magnetometers and barometers.

For this project, the selection will be limited to standalone MEMS gyros, which additionally come in a variety of performance levels. In order to carry out a thorough assessment, two representative models have been selected. On one hand, the PinPoint model (CRM100, CRM200) from Silicon Sensing is selected as low-cost option ($\sim$ €100) with decreased performance. On the other hand, the STIM202 model from Safran Sensing Technologies will be considered as a high-performance alternative, along with a hefty increase in price ($\sim$ €3000).

4.2. Measurement Error Sources

The assessment of measurement error sources is a critical step in the development of a sensor model. It involves identifying and quantifying the sources of error that affect the measurements made by the sensors, including environmental factors (e.g. temperature, vibrations, etc.), inherent sensor noise, and any errors introduced by the sensor electronics or processing algorithms.

Once the sources of error have been identified, mathematical models can be developed to characterize their effects on the sensor measurements. These error models are typically expressed as statistical distributions, such as Gaussian or uniform distributions. The Gaussian distribution offers several useful properties, including its ability to accurately model many natural and man-made phenomena, its well-known mathematical properties, and its ability to be easily incorporated into statistical methods such as the Kalman filter. While it is true that the Gaussian distribution does not have finite bounds, this is typically not a concern for practical applications.

By incorporating measurement error sources into the sensor model, it becomes possible to accurately simulate the behavior of the sensor under a wide range of operating conditions, and to estimate the uncertainty associated with the measurements made by the sensor.

4.2.1. Sun Sensor

Sun sensors have a relatively simple geometry, which makes them easier to model compared to more complex sensors. Typical measurement error sources include misalignment of the sensor with respect to the spacecraft body frame, sensor noise, atmospheric attenuation, and non-uniformity of the sun sensor's response to incident light. Addition-

ally, thermal effects and aging of the sensor can also affect the accuracy.

Systematic errors, such as those due to misalignment or non-uniformity, are accounted for in the calibration step. This is typically done during the manufacturing process or before launch, and may need to be updated periodically throughout the mission to ensure accuracy of the measurements.

The remaining errors can generally be approximated by a normal distribution with a mean of zero and a standard deviation that depends on the specific sensor model and its operating conditions. Typically, its value is stated in the sensor's datasheet and expressed directly in angular units.

4.2.2. Star Tracker

Star trackers are considerably more complex, and are subject to a wider range of errors. Some common error sources include:

- **Star catalog errors:** Inaccuracies in the star catalog used by the star tracker (unmodeled parallax effects, errors on star propagation due to proper motion, etc.) can result in errors in attitude determination.
- **Stellar aberration:** Stellar aberration is an apparent shift in the positions of stars caused by the motion of the observer. It can cause errors in the measurements obtained by star trackers, as the apparent positions of stars may be slightly shifted from their actual positions.
- **Sensor Noise:** Star tracker measurements are subject to noise, which can be caused by a variety of factors, such as thermal fluctuations, dark current, fixed-pattern noise, readout noise and quantization errors.
- **Optical distortion and aberrations:** Imperfections in the optics of the star tracker can introduce errors in the measurement of the position of stars.
- **Thermal effects:** Thermal variations can cause mechanical distortions in the star tracker, leading to measurement errors.
- **Centroiding errors:** Centroiding errors can be caused by various factors such as noise in the image, pixel size, pixel sensitivity, distortion, and optical aberrations. These errors can cause a shift in the calculated centroid position and thus introduce errors in the star identification process.

Linked to some of the previous effects, star trackers typically present time-correlated noise, with correlation times proportional to the spacecraft's angular velocity. Its two

major components are high spatial frequency error (HSFE), related to the time a star needs to cross one pixel, and low spatial frequency error (LSFE), associated to the time a star needs to cross the entire sensor field-of-view. Nevertheless, these errors are relatively small and are often lumped into Gaussian noise together with other error sources. For the purposes of this work, this will constitute a valid simplification.

Furthermore, systematic errors (stellar aberration, optical distortion, thermal effects, etc.) can be eliminated to a sufficient degree through calibration. Usually, star trackers are able to provide a full attitude estimate and their precision is specified in terms of across- and around-boresight attitude errors (their standard deviation). These, however, may be traced back to an unique Gaussian error affecting each individual star direction.

4.2.3. Magnetometer

Magnetometers are also subject to some of the instrumental error sources affecting the previous sensors, including sensor noise, thermal effects, quantization errors, scaling factors and non-orthogonality. Additionally, the measurements can be affected by nearby magnetic or large electric fields originating from other spacecraft components. Fortunately, most of these can be calibrated, and the rest may be approximated by a Gaussian distribution whose standard deviation is typically indicated in sensor datasheets.

However, inaccurate modeling of the Earth's magnetic field and orbit uncertainty also introduce errors relevant for attitude determination. The geomagnetic field, denoted by $\mathbf{B}$, is related to the geomagnetic potential V through $\mathbf{B} = -\nabla V$. The latter can be approximated by a series expansion of spherical harmonics [26]:

$$V(r,\theta,\phi) = R_E \sum_{n=1}^{\infty} \sum_{m=0}^{n} \left(\frac{R_E}{r}\right)^{n+1} (g_n^m \cos m\phi + h_n^m \sin m\phi) P_n^m(\cos\theta) \tag{4.1}$$

Here, r, θ, ϕ refer to coordinates in a geocentric spherical coordinate system, R_E is the mean radius of the Earth, 6371 km, and P_n^m are Schmidt semi-normalized associated Legendre functions of degree n and order m. The spherical harmonic coefficients g_n^m and h_n^m may be determined experimentally by combining earth-based and satellite measurements of Earth's magnetic field. This series is typically truncated at $n = 13$, and the coefficients are regularly updated by the International Association for Geomagnetism and Aeronomy (IAGA) producing what is known as the International Geomagnetic Reference Field (IGRF).

Although modeling errors are sometimes considered as white noise [27, 28], they are highly time-correlated and therefore a more special treatment is recommended [29].

To simulate this autocorrelation, two IGRF models may be used, one with coefficients offset from the truth epoch by three years. For a selected time interval, this results in time-correlated noise with a root-mean-square error (RMSE) of 184 nT, in line with most reported modeling error estimations [30, 31]. In Figure 4.1, the angle between the true and offset magnetic field vectors is shown, with an RMSE close to 0.2°.

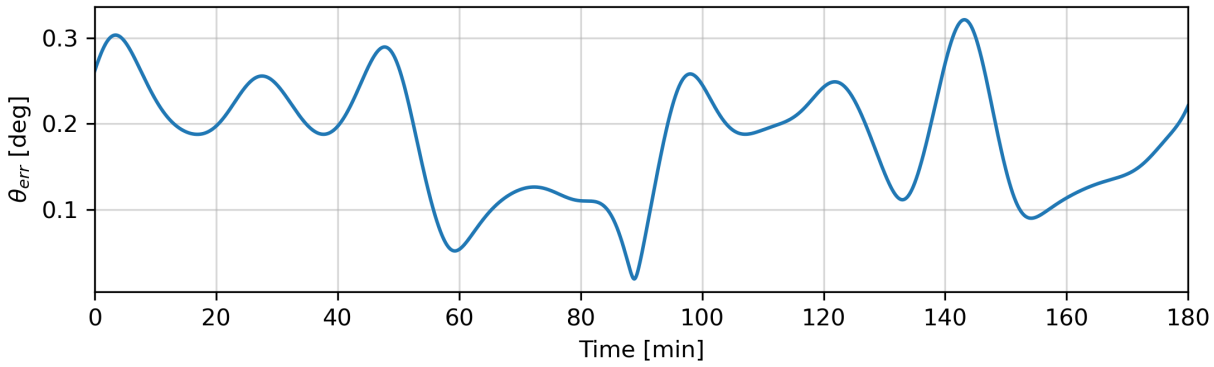


Figure 4.1: Typical geomagnetic field model error using a three year offset estimation model.

Assuming that the spacecraft stores a full order IGRF model to perform its computations, the noise in Figure 4.1 is an accurate representation of what the error between the estimated and real geomagnetic field may look like, including possible orbit uncertainty. The Kalman filter deals much less effectively with non-Gaussian error sources, and simulating time-correlated noise augments the realism of the analysis. In cases where a Gaussian approximation is required, the RMSE may be interpreted as its standard deviation.

4.2.4. Gyroscope

As with the previously analyzed sensors, gyroscope measurements may be corrupted by deterministic and stochastic errors. Deterministic errors typically include scale factor error, misalignment, non-linearity, and turn-on bias. Except for turn-on bias, which requires real-time estimation via state augmentation, in general these errors show insignificant variation throughout the instrument's lifetime and can be accounted for through factory calibration.

Stochastic errors, on the other hand, are difficult to predict. They can be caused by temperature variations, vibration, and electronic noise among other factors, and are typically characterized by their power spectral density. Its dominant components are known as *angle random walk* (ARW), *bias instability* (BI) and *rate random walk* (RRW),

which may be extracted from what is known as an Allan Standard Deviation (ASD) plot [32]. The manufacture usually provides the ASD plot or parameters extracted from it.

In general terms, ARW corresponds to white noise and RRW represents an actual random walk of the measurement (i.e. its derivative is Gaussian). BI, on the other hand, is slightly more complex and requires at least first-order Gauss-Markov models, which will not be covered here. Ultimately, the non-Gaussian noise components result in a time-varying bias which needs to be continuously estimated on-board to avoid significant errors.

4.3. Sensor Models

Sensor models are used to describe the relationship between the measurements obtained by a sensor and the physical parameters being measured. These models are used for a variety of purposes, such as calibrating the sensor, simulating its behavior in different scenarios, and optimizing the sensor system's performance. In the context of this work, they allow for measurement simulation and data processing within the filtering framework.

A sensor model can be highly complex and depend on a variety of factors, including the physical principles governing the sensor's operation, environmental conditions, and measurement noise sources. Typically, simplified models can provide sufficient accuracy while being easier to develop, interpret, and implement. In this section, several widely used models for the different types of sensor are introduced.

4.3.1. Unit Vector Observation Model

In many attitude systems, the observations are naturally represented as vectors. Typical measurements include the direction to the Earth, the Sun, a star, or the Earth's magnetic field. Unit vectors are preferred because they are independent of the magnitude of the measured quantity and are mathematically simpler to work with. The unit vector observation model assumes that the measured vector is simply a linear combination of the true unit vector and measurement noise. As argued in Section 4.2, in most cases this is a reasonable approximation. Discrete-time observations for a single sensor may therefore be given by

$$\widetilde{\mathbf{b}} = A\mathbf{r} + \boldsymbol{\nu} \tag{4.2}$$

where $\widetilde{\mathbf{b}}$ denotes the measurement vector in the body frame, $\mathbf{r}$ is the known reference vector in an inertial reference frame, A is the attitude matrix, and $\boldsymbol{\nu}$ represents zero-

mean Gaussian measurement error vector with covariance $\sigma^2 I_3$.

Note that the simplifying assumption of isotropic measurement error is being made, which is clearly not true; if the error-corrupted vectors are thought of as arrows sharing a common base, their points should all lie on a unit sphere. Nevertheless, Malcolm Shuster demonstrated that in the context of Kalman filtering, considering isotropic measurement error produces equivalent results to considering its projection onto the unit sphere [33]. A simple illustrative diagram is provided in Figure 4.2, where an *uncertainty cone* bounds (in a statistical sense) all the possible directions of the real reference vector in the spacecraft's body frame.

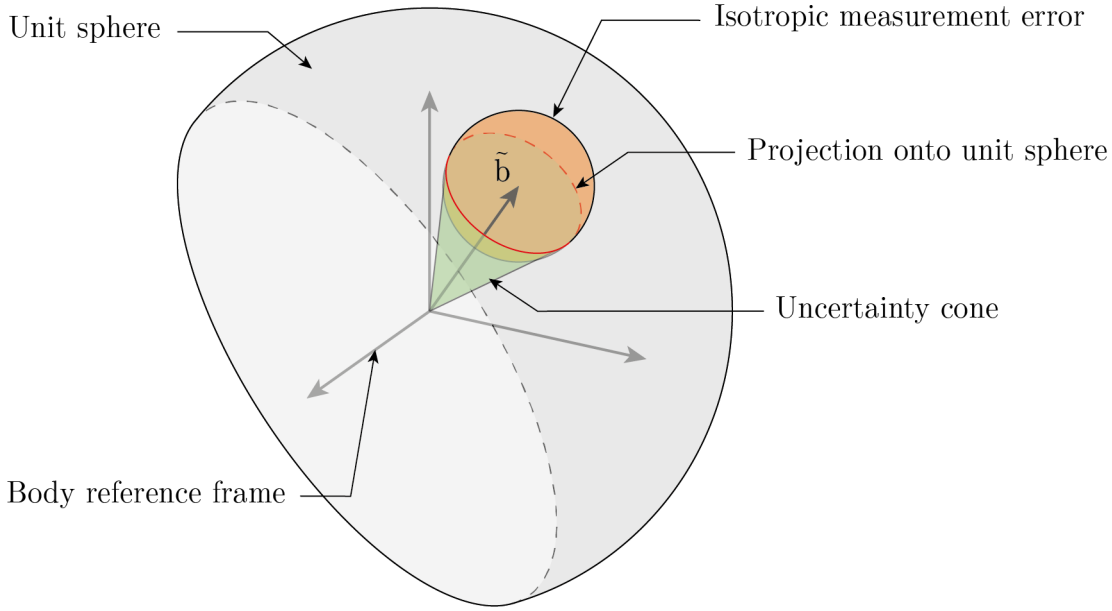


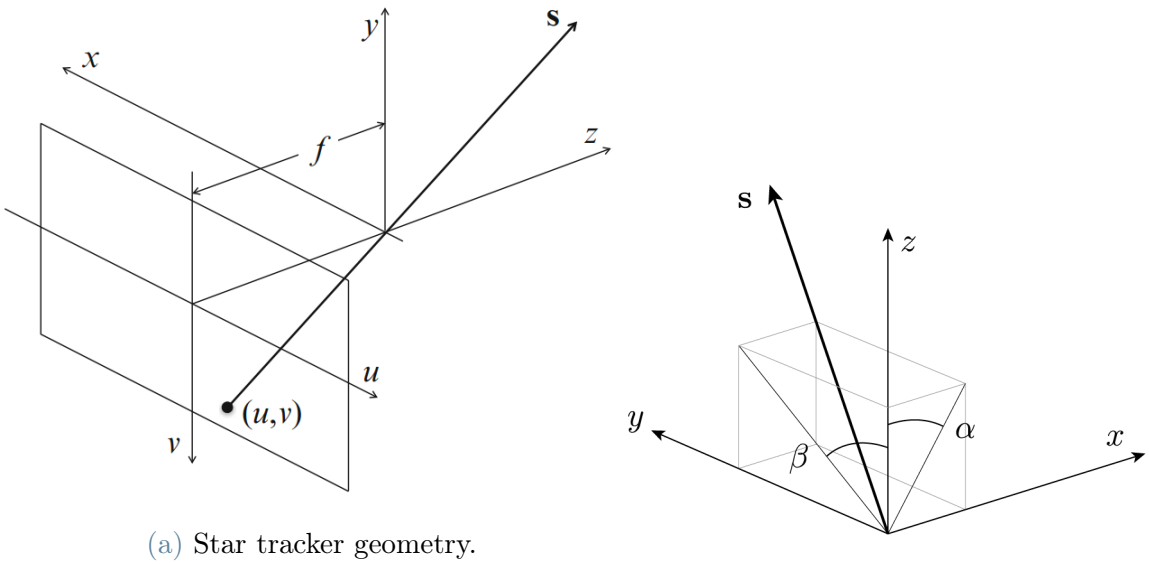
Figure 4.2: Diagram of an exaggerated measurement error distribution (3σ limits).

While unit vector measurements are the norm in static methods such as TRIAD and Wahba's problem, some non-linear filtering methods (for instance the UKF) allow more flexibility. Specifically, two quantity observation models may be employed (e.g. the two angles measured by a two-axis Sun sensor).

4.3.2. Two quantity Observation Models

In this section, two different models are presented, valid for star trackers and two-axis Sun sensors such as Solar MEMS' NanoSSOC. Figure 4.3a shows the simplified geometry of a star tracker, similar to that of a pinhole camera. The x, y and z axis constitute a right-handed Cartesian coordinate system, with its origin at the vertex of the optical

lens and its z axis along the optical axis, the star tracker's boresight. The focal plane is located at a distance f behind the vertex (focal distance) and, at its center, the origin of the u, v coordinate system is defined.



(a) Star tracker geometry. Adapted from Crassidis & Markley [3].

(b) Two angles model.

Figure 4.3: Typical two quantity observation models for star trackers and Sun sensors.

The unit vector $\mathbf{s}$ from the spacecraft to a star can be computed from the focal plane coordinates as:

$$\mathbf{s} = \frac{1}{\sqrt{f^2 + u^2 + v^2}} \begin{bmatrix} u \\ v \\ f \end{bmatrix} \tag{4.3}$$

and it is conventional to define $\alpha = \arctan(u/f)$ and $\beta = \arctan(v/f)$, leading to the two angles model in Figure 4.3b where

$$\mathbf{s} = \frac{1}{\sqrt{1 + \tan^2\alpha + \tan^2\beta}} \begin{bmatrix} \tan\alpha \\ \tan\beta \\ 1 \end{bmatrix} \tag{4.4}$$

The previous definitions allow for a more accurate characterization of the measurement errors. Using the model in Figure 4.3a, errors may directly represent uncertainty in the imaged stars centroids and be expressed in terms of pixels or other units of length. Alternatively, they may represent uncertainty in the angles α and β and be expressed in angular units.

Note that isotropic error in either (u, v) or (α, β) does not translate into a circular

uncertainty cone such as the one in Figure 4.2 (except when $\mathbf{s}$ is aligned with the z axis). Nevertheless, for small errors it is generally a good approximation. If the initial measurements are subject to a Gaussian error with covariance $\sigma_{uv}^2 I_2$ or $\sigma_{\alpha\beta}^2 I_2$ (where $\sigma_{\alpha\beta}$ is expressed in radians), the covariance of the corresponding Gaussian error vector in Equation (4.2) can be approximated by $(\sigma_{uv}/f)^2 I_3$ or $\sigma_{\alpha\beta}^2 I_3$ respectively.

As stated in Section 4.2, Sun sensor error is typically provided by the manufacturer as uncertainty in (α, β). For star trackers, since the attitude determination is performed internally, the error is generally specified in terms of across- and around-boresight attitude errors. These, however, can be easily translated into (α, β) error through a simple procedure. First, simulate n stars in the star tracker's FOV (a mean number of 6-8 identified stars may be assumed) and compute their corresponding (α, β). Then, corrupt each angle with Gaussian error of covariance $\sigma_{\alpha\beta}^2 I_2$, solve Wahba's problem to obtain an attitude estimation, and calculate the across- and around-boresight errors. To find the correct value of $\sigma_{\alpha\beta}$, repeat the previous procedure until the computed attitude accuracy matches with that provided by the manufacturer to a sufficient degree.

4.3.3. Gyro Measurement Model

A widely used continuous-time model for three-axis gyroscope measurements is:

$$\widetilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) + \boldsymbol{\eta}_v(t) \tag{4.5a}$$

$$\dot{\boldsymbol{\beta}}(t) = \boldsymbol{\eta}_u(t) \tag{4.5b}$$

where $\boldsymbol{\omega}(t)$ is the true angular velocity, $\widetilde{\boldsymbol{\omega}}(t)$ is the measured angular velocity, $\boldsymbol{\beta}(t)$ is the true bias, and $\boldsymbol{\eta}_v(t)$ and $\boldsymbol{\eta}_u(t)$ are independent white noise process (Gaussian, zero-mean) with power spectral densities $\sigma_v^2 I_3$ and $\sigma_u^2 I_3$ respectively. In this model, $\boldsymbol{\eta}_v(t)$ corresponds to ARW and $\boldsymbol{\eta}_u(t)$ to RRW.

For simulation purposes, a discrete-time model is usually required. With a gyro sampling interval Δt, the discrete-time angular velocity and bias equations are:

$$\widetilde{\boldsymbol{\omega}}_{k+1} = \boldsymbol{\omega}_k + \frac{1}{2}(\boldsymbol{\beta}_{k+1} + \boldsymbol{\beta}_k) + \left(\frac{\sigma_v^2}{\Delta t} + \frac{1}{12}\sigma_u^2 \Delta t\right)^{1/2} \mathbf{N}_v \tag{4.6a}$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \sigma_u^2 \, \Delta t^{1/2} \, \mathbf{N}_u \tag{4.6b}$$

where the subscript k denotes the k-th step, and $\mathbf{N}_v$ and $\mathbf{N}_u$ are Gaussian variables with covariance each given by the identity matrix. A complete derivation of the previous equations can be found in [3].

Equations (4.5b) and (4.6b) imply that the bias is unbounded, since the standard deviation of $\beta(t)$ grows indefinitely over time (a well known property of Gaussian random walk). While bias may drift due to environmental factors such as temperature changes and mechanical stresses, in practice it is limited by the physical properties of the gyro (consider the finite turn-on bias). To avoid excessively large values, an artificial limit may be imposed. In the context of measurement simulation, this can be easily achieved by forcing the sign of $\mathbf{N}_u$ to be negative whenever $\beta_k(i) > \beta_{max}$ or positive if $\beta_k(i) < -\beta_{max}$, where β_{max} is the bias limit and $\beta(i)$ denotes any component of $\boldsymbol{\beta}_k$ $(i = 1, 2, 3)$.

4.4. Technical Specifications

This section will provide an overview of the key technical specifications of the selected sensors. First, the specifications given by each manufacturer are presented in Table 4.1.

Table 4.1: Technical specifications of the selected sensor suite.

Manufacturer	Model	Technical Specifications
Solar MEMS	*STNS*	• Max. sampling rate: 5 Hz • Across-boresight error: 30 arcsec (3σ) • Along-boresight error: 200 arcsec (3σ) • Field of view (FOV): 14° • Max. slew rate: 2 deg/s
Solar MEMS	*NanoSSOC*	• Max. sampling rate: 50 Hz • Accuracy: 0.5° (3σ)
AAC Clyde Space	*MM200*	• Max. sampling rate: 500 Hz • Noise spectral density: 1.18 nT/$\sqrt{\text{Hz}}$
Silicon Sensing	*CRM100*	• Max. sampling rate: 160 Hz • Turn-on bias: 0.42 deg/s (3σ) • Max. bias variation with temperature: ±3 deg/s • ASD plot in Figure 4.4
Safran	*STIM202*	• Max. sampling rate: 1000 Hz • Turn-on bias: 0.01 deg/s (3σ) • Typical bias drift over temperature in Figure 4.6 • ASD plot in Figure 4.5

First, note that the sampling rate is generally configurable. For a fair comparison, both gyroscopes will be simulated using identical sampling rate. Additionally, the TAM sampling rate will be selected so as to match that of the Sun sensor. These can be adjusted depending on the scenario and pointing requirements. Spacecraft typically rotate very

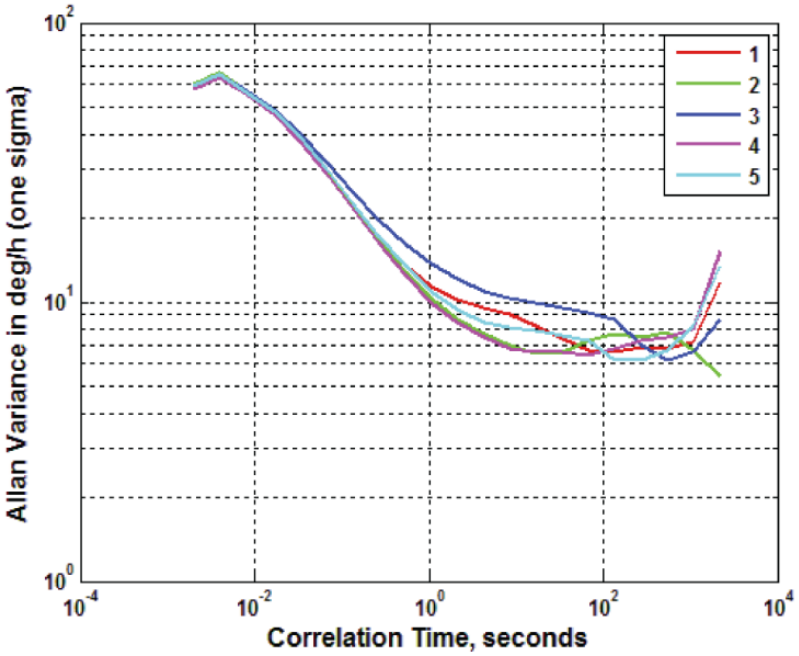


Figure 4.4: Allan standard deviation (and not variance, as indicated on the vertical axis of the original figure) plot of the CRM100 gyroscope.

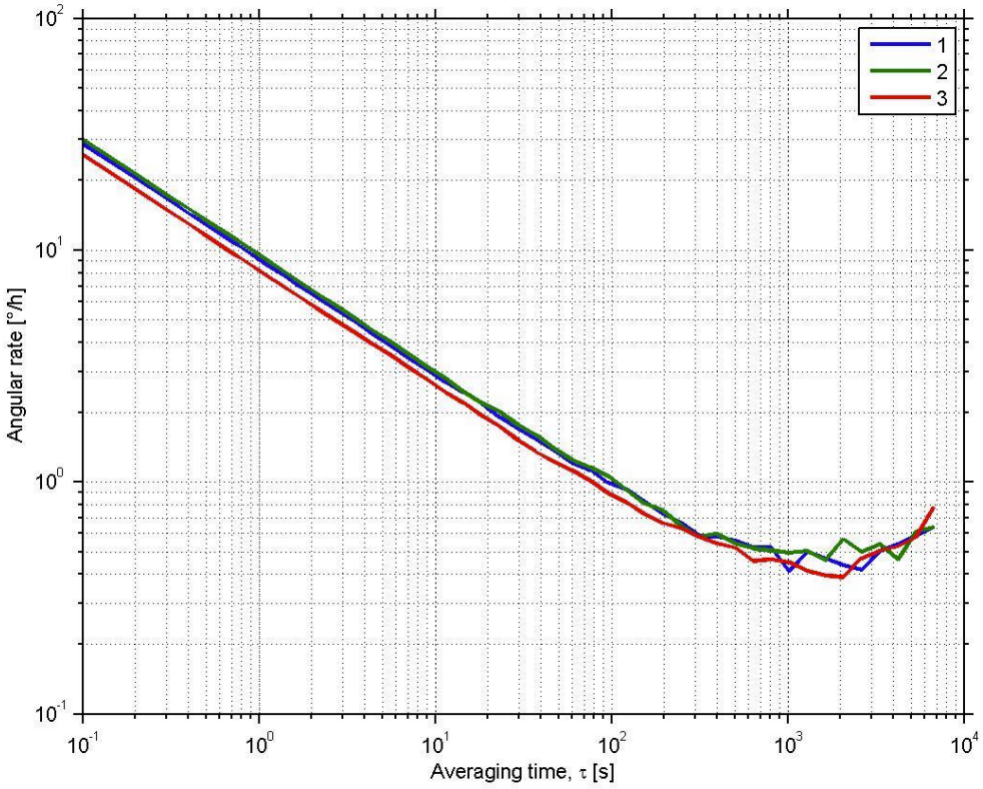


Figure 4.5: Allan standard deviation plot of the STIM202 gyroscope.

slowly, and changes in the measured quantities (magnetic field, Sun or star directions, etc.) are also slow. For most applications, sampling rates ranging from 1 to 10 Hz tend to provide satisfactory results.

As for the star track tracker and Sun sensor, translating the previous accuracy values into useful properties for the models in Equations (4.2) and (4.4) is straightforward. On one hand, the Sun sensor accuracy is already specified in (α, β), and just a change in units (to radians) is required for use in the unit vector observation model. Regarding the star tracker, its across- and along-boresight error can be expressed in terms of (α, β) accuracy following the procedure presented in Section 4.3.2. Assuming an average number of 6 stars in the specified FOV, the resulting angular error in the direction of each star has a 3σ value of 55°.

Dealing with the magnetometer is slightly more complex. The value indicated in the sensor's datasheet corresponds to white noise corrupting the measurement, which may be converted into discrete measurement error by multiplying the spectral density by the square root of the sampling rate. At 50 Hz, this results in a Gaussian error with covariance $\sigma_{TAM}^2 I_3$, where $\sigma_{TAM} \simeq 8.4$ nT. Since the minimum magnetic field encountered in LEO orbits has an intensity of 24000 nT, this results in a worst-case (α, β) error with standard deviation $\sim 0.02°$. In addition, modeling noise needs to be considered. As argued in Section 4.2.3, autocorrelated noise with an RMSE (standard deviation) of 0.2° may be used.

Table 4.2: Gyro parameters.

Model	σ_v [°/$\sqrt{\text{h}}$]	σ_u [°/h$^{3/2}$]	β_{max} [deg/s]
CRM100	0.2	200	4
STIM202	0.2	1	0.15

The parameters σ_v and σ_u used in the continuous gyro measurement model of Equation (4.5) may be extracted from the ASD plots in Figures 4.4 and 4.5 following the procedure in [32]. However, these plots are computed at a constant temperature, which is not realistic in the context of a low Earth orbit. Generally, due to periodic eclipses, the spacecraft is subject to extreme changes in temperature which introduce significant drift in the gyro bias. To include this effect into the ARW and RRW parameters, a trial-and-error process has been carried out; starting from the values extracted from the ASD plots, modify them until the simulated gyro response at rest matches with the expected bias drift (e.g. the evolution in Figure 4.6). The resulting values are presented in Table 4.2, along with indicative values of the bias limit β_{max}.

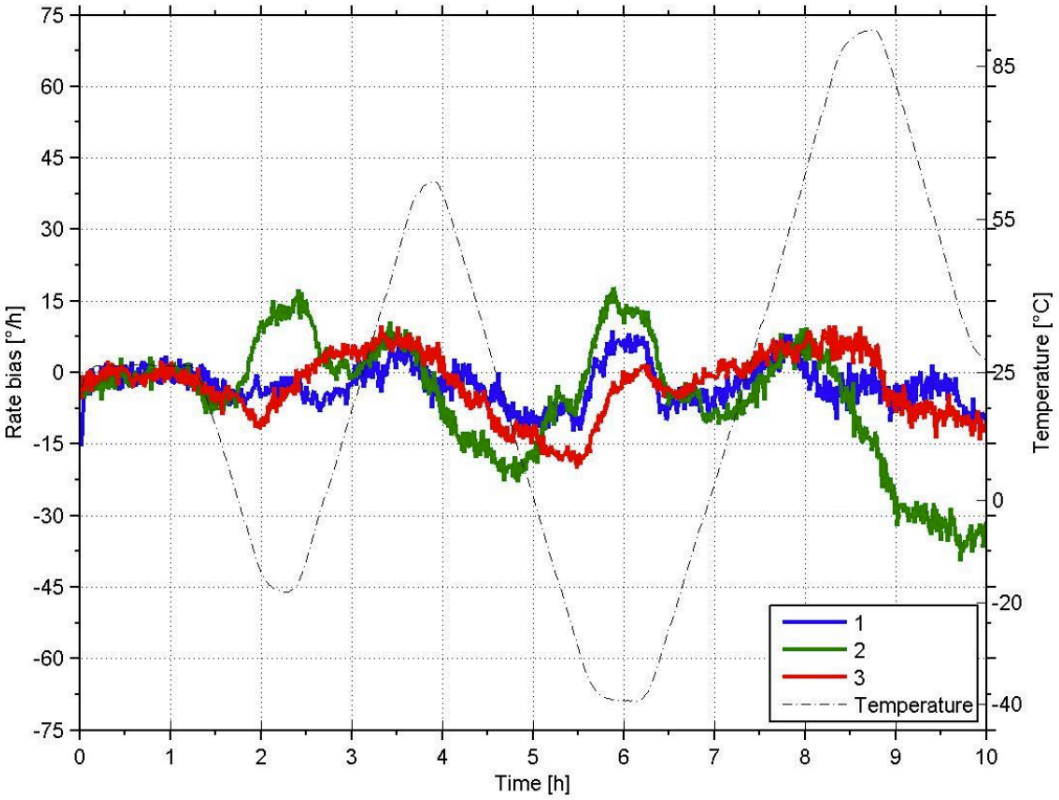


Figure 4.6: Typical bias drift over temperature of the STIM202 gyroscope.

5 Filter Design

This chapter explores the design of Kalman filters for spacecraft attitude estimation, which is an essential aspect of the attitude determination module. The Kalman filter is a widely-used state estimation method that combines system dynamics and measurements to provide an accurate estimate of the state. After a brief introduction, three specific Kalman filters will be covered: the multiplicative extended Kalman filter (MEKF), an unscented version based on Crassidis & Markley's USQUE [14], and an iterated MEKF (IMEKF) based on the recent work by Chang [13]. Additionally a novel technique for improving convergence (bias clipping) will be introduced.

The MEKF is widely regarded as the workhorse of real-time spacecraft attitude estimation, and represents the attitude as the product of an estimated quaternion and a deviation from that estimate. Since it is based on the classical EKF, it also uses linearization to approximate the system and measurement equations. The USQUE, on the other hand, uses the unscented transform to propagate the state and its uncertainty through the system dynamics and perform the measurement update. The IMEKF is a modified version of the MEKF that improves the filter convergence.

Overall, this chapter provides an in-depth overview of the design and implementation of various Kalman filters for spacecraft attitude estimation, addressing relevant practical aspects and the bias clipping technique.

5.1. Introduction to Kalman Filtering

The Kalman filter, originally developed by Rudolf Kalman in the 1960s, is a recursive algorithm that uses measurements of a system and a mathematical model of its dynamics to estimate the system's state (e.g. the spacecraft's attitude). The filter works by combining these measurements with the model predictions to form an optimal estimate.

Kalman filtering is particularly useful for systems that are subject to noise and uncertainties, as it can handle these factors and provide accurate estimates of the system state. The filter is also able to adapt to changes in the system dynamics over time and can make real-time predictions based on the available measurements and the system model.

The design of a Kalman filter involves several key steps, including defining the system state, selecting dynamic and measurement models, and determining the process and measurement noise covariances. The successful implementation of a Kalman filter requires careful consideration of the previous steps and a thorough understanding of the underlying system dynamics. In the following sections, these aspects will be covered for the specific case of spacecraft attitude estimation.

5.2. Multiplicative Quaternion Representation

Section 3.2 presented several different attitude representations, some of which are more appropriate for attitude estimation than others. For most filtering applications, the quaternion has become the representation of choice, mainly due to its compact parameterization and lack of singularities. Quaternions are also very computationally efficient and easy to normalize, making them convenient to work with in numerical algorithms.

As argued in Section 3.2.7, the most natural way of representing attitude errors is using a multiplicative approach:

$$\mathbf{q} = \boldsymbol{\delta}\mathbf{q} \otimes \hat{\mathbf{q}} \tag{5.1}$$

where $\mathbf{q}$ represents the true quaternion, $\hat{\mathbf{q}}$ the estimated quaternion, and $\boldsymbol{\delta}\mathbf{q}$ the error quaternion defined in the spacecraft body frame. Note that they are all properly normalized unit quaternions.

An additive error representation ($\mathbf{q} = \hat{\mathbf{q}} + \Delta\mathbf{q}$) is simpler to implement and may be preferred in certain applications, but presents some problems which have not yet been solved in a satisfactory manner. For instance, enforcing the unit norm constraint eventually produces an ill-conditioned covariance matrix, since only three degrees of freedom exist [3].

5.3. MEKF

The fundamental idea behind the multiplicative EKF (and its variants) is to utilize the quaternion as a "global" attitude representation, with a three-component state vector $\mathbf{a}$ serving as the "local" representation of attitude errors. This approach present several advantages; on one hand, the multiplicative formulation guarantees that the normalization of $\hat{\mathbf{q}}$ is maintained. Additionally, since the four-component error quaternion has been effectively replaced by a three-component error vector, the covariance matrix has minimum dimensionality and therefore does not show any singularities.

Typical choices for $\mathbf{a}$ include the rotation vector, two times the vector part of the quaternion, two times the vector of Rodrigues parameters, or four times the vector of MRPs. As discussed in 3.2.7, all of these error representations are equivalent up to second order, and share the following first order error quaternion parametrization:

$$\boldsymbol{\delta}\mathbf{q}(\mathbf{a}) \approx \begin{bmatrix} \mathbf{a}/2 \\ 1 \end{bmatrix} \tag{5.2}$$

Based on the study in [14], four times the vector of MRPs ($\mathbf{a} = 4\,\boldsymbol{\delta}\mathbf{p}$) has been chosen as the local attitude error representation for the current filters.

The MEKF algorithm is presented in Table 5.1. Even though a mathematical derivation is readily available in any dedicated textbook, a brief review is presented next for convenience, and to point out certain aspects which differentiate it from the other considered filters. Additionally, an integration technique to process n gyro samples between successive attitude measurements (which is not addressed in any of the referenced sources) is presented.

Table 5.1: MEKF.

Initialization	Set $\hat{\mathbf{q}}_0$, $\hat{\boldsymbol{\beta}}_0$ and P_0
Propagation	$\hat{\mathbf{q}}_{k-1}^{+}$, $P_{k-1}^{+} \longrightarrow \hat{\mathbf{q}}_k^{-}$, P_k^{-} using $\hat{\boldsymbol{\omega}} = \widetilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{k-1}^{+}$ $\dot{\hat{\mathbf{q}}} = \frac{1}{2}\left[\hat{\boldsymbol{\omega}} \otimes\right] \hat{\mathbf{q}}$ $\dot{P} = FP + PF^T + GQG^T$
Local update	$\Delta\hat{\mathbf{x}}_k^{+} = K_k(\widetilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{q}}_k^{-}))$ $P_k^{+} = P_k^{-} - K_k H_k P_k^{-}$ where $K_k = P_k^{-} H_k^T (H_k P_k^{-} H_k^T + R_k)^{-1}$ $\mathbf{h}(\hat{\mathbf{q}}_k^{-}) = \begin{bmatrix} A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_1 \\ A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_m \end{bmatrix}$, $H_k = \begin{bmatrix} [A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_1\times] & 0_3 \\ [A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_2\times] & 0_3 \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}_k^{-})\mathbf{r}_m\times] & 0_3 \end{bmatrix}$
Global update	$\Delta\hat{\mathbf{x}}_k^{+} = \begin{bmatrix} \hat{\mathbf{a}}_k^{+} \\ \Delta\hat{\boldsymbol{\beta}}_k^{+} \end{bmatrix}$ $\hat{\mathbf{q}}_k^{+} = \boldsymbol{\delta}\mathbf{q}(\hat{\mathbf{a}}_k^{+}) \otimes \hat{\mathbf{q}}_k^{-}$ $\hat{\boldsymbol{\beta}}_k^{+} = \hat{\boldsymbol{\beta}}_k^{-} + \Delta\hat{\boldsymbol{\beta}}_k^{+}$

The MEKF proceeds by iteration of three steps: propagation to the next measurement time, measurement update, and state vector reset. While the first two are general steps in any Kalman filter, the third is a particular feature of the MEKF. The reset moves the updated information from the error state to the global attitude representation, and resets the components of the error state to zero. Using this strategy, $\mathbf{a}$ always remains small and therefore never approaches a singularity.

5.3.1. Step 1: Propagation

There are two ways of approaching the propagation step. The first one, more in line with classical Kalman filter applications, is to use a dynamic model which incorporates attitude kinematics and attitude dynamics, and therefore considers both the attitude and angular velocity as state variables. Gyro data is then included as conventional measurements in the update step. The alternative is to incorporate gyro information as part of dynamic model, excluding attitude dynamics equations. This is often referred to as using gyros for dynamic model replacement, and presents two significant advantages; first, gyro information is typically more accurate than the available models of rotational dynamics and torques, and secondly, requires much less computation. For these reasons, it will be the method of choice for the current application.

As stated in Section 3.3, the kinematics equations for the global quaternion estimate and the local error vector are

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\left[\hat{\boldsymbol{\omega}}\otimes\right]\hat{\mathbf{q}} \tag{5.3}$$

and

$$\dot{\mathbf{a}} = -[\hat{\boldsymbol{\omega}}\times]\,\mathbf{a} + \boldsymbol{\delta\omega} \tag{5.4}$$

respectively, where $\hat{\boldsymbol{\omega}}$ is the estimated angular velocity vector and $\boldsymbol{\delta\omega}$ the error angular velocity ($\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$). Note that the time dependency of all variables has been omitted for simplicity.

For general on-board usage, the MEKF estimates gyro biases in addition to the attitude. Therefore, it updates the error state

$$\Delta\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \Delta\boldsymbol{\beta} \end{bmatrix} \tag{5.5}$$

where $\Delta\boldsymbol{\beta}$ is the error in gyro biases, defined as the difference between their true and estimated values ($\Delta\boldsymbol{\beta} = \boldsymbol{\beta} - \hat{\boldsymbol{\beta}}$). It is assumed that the relatively fixed gyro error sources (e.g. scale factor, misalignment) have already been calibrated by an appropriate spacecraft

maneuver profile. By substituting the continuous gyro model from Equation (4.5) into Equation (5.4), the kinematics of the error state can be obtained:

$$\dot{\mathbf{a}} = -[\hat{\boldsymbol{\omega}}\times]\,\mathbf{a} - \Delta\boldsymbol{\beta} - \boldsymbol{\eta}_v \tag{5.6a}$$

$$\Delta\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_u \tag{5.6b}$$

where the following considerations have been made:

$$\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} = \widetilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v = -\Delta\boldsymbol{\beta} - \boldsymbol{\eta}_v \tag{5.7}$$

$$\dot{\hat{\boldsymbol{\beta}}} = 0 \tag{5.8}$$

The previous system of equations may be expressed as:

$$\Delta\dot{\mathbf{x}} = F\Delta\mathbf{x} + G\mathbf{w} \tag{5.9a}$$

$$\left(\dot{P} = FP + PF^T + GQG^T\right) \tag{5.9b}$$

where P is the state covariance,

$$F = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}\times] & -I_3 \\ 0_3 & 0_3 \end{bmatrix}, \quad G = \begin{bmatrix} -I_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} \tag{5.10}$$

and $\mathbf{w}$ is a process noise vector with power spectral density

$$Q = \begin{bmatrix} \sigma_v^2 I_3 & 0_3 \\ 0_3 & \sigma_u^2 I_3 \end{bmatrix} \tag{5.11}$$

Note that 0_3 represents a 3×3 null matrix.

For practical applications, a discrete-time version of Equations (5.3) and (5.6) is required. Recall that within the MEKF framework, the error state is always reset to zero after the measurement update, and remains identically zero over the propagation step. Therefore, only its covariance, which will be denoted by $P(t)$, needs to be propagated. Consider the time interval $[t_k,\, t_{k+1}]$ between successive measurements, which is split into n gyro samples (including one at t_k but not at t_{k+1}). Recall that the quaternion estimate may be propagated using

$$\hat{\mathbf{q}}(t_{k+1}) = \Omega(\tau_{n+1}, \tau_n) \cdots \Omega(\tau_{j+1}, \tau_j) \cdots \Omega(\tau_2, \tau_1)\, \hat{\mathbf{q}}(t_k) \tag{5.12}$$

with

$$\tau_j = t_k + (j-1)\,\frac{t_{k+1}-t_k}{n}, \quad j = 1, 2, ..., n \tag{5.13}$$

$$\Omega(\tau_{j+1}, \tau_j) = \cos\left(\frac{1}{2}\,\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau\right) I_4 + \frac{1}{\|\hat{\boldsymbol{\omega}}_j\|}\sin\left(\frac{1}{2}\,\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau\right)[\hat{\boldsymbol{\omega}}_j \otimes] \tag{5.14}$$

where $\Delta\tau = \tau_{j+1} - \tau_j$ corresponds to the sampling time interval of the gyro and $\hat{\boldsymbol{\omega}}_j = \hat{\boldsymbol{\omega}}(\tau_j)$ is the estimated angular velocity vector at time τ_j (it is being assumed that $\hat{\boldsymbol{\omega}}(t) = \hat{\boldsymbol{\omega}}_j \;\forall\, t \in [\tau_j,\, \tau_{j+1})$). Note that $\hat{\boldsymbol{\omega}}(t) = \widetilde{\boldsymbol{\omega}}(t) - \hat{\boldsymbol{\beta}}(t)$.

The solution to Equation (5.6) is slightly more complicated, since it involves an external variable, namely process noise. Nevertheless, assuming constant angular velocity, the following closed-form solution can be obtained [19]:

$$\Delta\mathbf{x}(t) = \Phi(t, t_0)\Delta\mathbf{x}(t) + \int_{t_0}^{t} \Phi(\tau, t_0)\, G(\tau)\, \mathbf{w}(\tau)\, d\tau \tag{5.15}$$

Following the previous strategy, the time interval between measurements $[t_k,\, t_{k+1}]$ may be split into n gyro samples, where at each individual time step τ_j the state is propagated until τ_{j+1} using

$$\Delta\mathbf{x}(\tau_{j+1}) = \Phi(\tau_{j+1}, \tau_j)\Delta\mathbf{x}(\tau_j) + \int_{\tau_j}^{\tau_{j+1}} \Phi(t, \tau_j)\, G(t)\, \mathbf{w}(t)\, dt \tag{5.16}$$

where

$$\Phi(\tau_{j+1}, \tau_j) = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{5.17a}$$

$$\Phi_{11} = I_3 - [\hat{\boldsymbol{\omega}}_j\times]\,\frac{\sin(\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau)}{\|\hat{\boldsymbol{\omega}}_j\|} + [\hat{\boldsymbol{\omega}}_j\times]^2\,\frac{1-\cos(\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau)}{\|\hat{\boldsymbol{\omega}}_j\|^2} \tag{5.17b}$$

$$\Phi_{12} = [\hat{\boldsymbol{\omega}}_j\times]\,\frac{1-\cos(\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau)}{\|\hat{\boldsymbol{\omega}}_j\|^2} - \Delta\tau I_3 - [\hat{\boldsymbol{\omega}}_j\times]^2\,\frac{\hat{\boldsymbol{\omega}}_j\Delta\tau - \sin(\|\hat{\boldsymbol{\omega}}_j\|\Delta\tau)}{\|\hat{\boldsymbol{\omega}}_j\|^3} \tag{5.17c}$$

$$\Phi_{21} = 0_3, \quad \Phi_{22} = I_3 \tag{5.17d}$$

with τ_j, $\hat{\boldsymbol{\omega}}_j$ and $\Delta\tau$ defined previously.

At each time step τ_j, the state covariance $P(t)$ may be propagated until τ_{j+1} using

$$P(\tau_{j+1}) = \Phi(\tau_{j+1}, \tau_j)P(\tau_j)\Phi^T(\tau_{j+1}, \tau_j) + \int_{\tau_j}^{\tau_{j+1}} \Phi(t, \tau_j)\, G(t)\, Q(t)\, G^T(t)\, \Phi^T(\tau_{j+1}, \tau_j)\, dt \tag{5.18}$$

where the integral corresponds to the discrete process noise covariance $Q_{\Delta\tau}$ and can be approximated by

$$Q_{\Delta\tau} \approx \begin{bmatrix} \left(\sigma_v^2 \Delta\tau + \frac{1}{3}\sigma_u^2 \Delta\tau^3\right) I_3 & -\left(\frac{1}{2}\sigma_u^2 \Delta\tau^2\right) I_3 \\ -\left(\frac{1}{2}\sigma_u^2 \Delta\tau^2\right) I_3 & (\sigma_u^2 \Delta\tau) I_3 \end{bmatrix} \tag{5.19}$$

Equation (5.18) is then applied recursively in order to propagate P from t_k to t_{k+1}.

5.3.2. Step 2: Measurement Update

The MEKF typically makes use of the unit vector observation model presented in Section 4.3. Multiple (m) vector observations can be concatenated to form

$$\widetilde{\mathbf{y}} = \begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_2 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_m \end{bmatrix} + \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \\ \boldsymbol{\nu}_m \end{bmatrix} = \mathbf{h}(\mathbf{q}) + \mathbf{v} \tag{5.20}$$

where the power spectral density of $\mathbf{v}$ is

$$R = \begin{bmatrix} \sigma_1^2 I_3 & & & \\ & \sigma_2^2 I_3 & & \\ & & \ddots & \\ & & & \sigma_m^2 I_3 \end{bmatrix} \tag{5.21}$$

Since the MEKF updates the error state, the relationship between $\widetilde{\mathbf{y}}$ and $\Delta\mathbf{x}$ needs to be found. This can be achieved by linearizing Equation (5.20) around $\hat{\mathbf{q}}$:

$$\widetilde{\mathbf{y}} = \mathbf{h}(\hat{\mathbf{q}}) + H(\hat{\mathbf{q}})\Delta\mathbf{x} + \mathbf{v} \tag{5.22}$$

where $H(\mathbf{q})$ is the measurement sensitivity matrix, defined as

$$H(\mathbf{q}) = \frac{\partial \mathbf{h}}{\partial \Delta\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{a}} & \frac{\partial \mathbf{h}}{\partial \Delta\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} H_{\mathbf{a}} & H_{\Delta\boldsymbol{\beta}} \end{bmatrix} \tag{5.23}$$

Now, for a single observation, the true and estimated body vectors are given by

$$\mathbf{b} = A(\mathbf{q})\mathbf{r}, \quad \hat{\mathbf{b}} = A(\hat{\mathbf{q}})\mathbf{r} \tag{5.24}$$

respectively, and therefore Equation (5.20) may be rewritten as:

$$\widetilde{\mathbf{y}} = \hat{\mathbf{b}} + \Delta\mathbf{b} + \boldsymbol{\nu} \tag{5.25}$$

where $\Delta\mathbf{b} = \mathbf{b} - \hat{\mathbf{b}}$. Applying the quaternion composition property in Equation (3.16) and using a first order approximation of the attitude matrix $A(\boldsymbol{\delta}\mathbf{q})$:

$$\mathbf{b} = A(\boldsymbol{\delta}\mathbf{q})A(\hat{\mathbf{q}})\mathbf{r} \approx (I_3 - [\mathbf{a}\times])A(\hat{\mathbf{q}})\mathbf{r} \tag{5.26}$$

Thus

$$\Delta\mathbf{b} = -[\mathbf{a}\times]A(\hat{\mathbf{q}})\mathbf{r} = -[\mathbf{a}\times]\,\hat{\mathbf{b}} = [\hat{\mathbf{b}}\times]\,\mathbf{a} \tag{5.27}$$

and then

$$\widetilde{\mathbf{y}} = \hat{\mathbf{b}} + [\hat{\mathbf{b}}\times]\,\mathbf{a} + \boldsymbol{\nu} \tag{5.28}$$

Comparing Equations (5.25) and (5.28), the sensitivity matrix for m measurements is therefore given by

$$H(\hat{\mathbf{q}}) = \begin{bmatrix} [A(\hat{\mathbf{q}})\mathbf{r}_1\times] & 0_3 \\ [A(\hat{\mathbf{q}})\mathbf{r}_2\times] & 0_3 \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}})\mathbf{r}_m\times] & 0_3 \end{bmatrix} \tag{5.29}$$

Note that H is to be computed using the quaternion estimate right after the propagation step and prior to the measurement update. Typically, *a-priori* values are denoted by $\hat{\mathbf{q}}^-$, while *a-posteriori* values (after the measurement update) are denoted by $\hat{\mathbf{q}}^+$. At time t_k, the standard Kalman update equations are:

$$\Delta\hat{\mathbf{x}}_k^+ = \Delta\hat{\mathbf{x}}_k^- + K_k\,\mathbf{z}_k \tag{5.30a}$$

$$P_k^+ = P_k^- - K_k H_k P_k^- \tag{5.30b}$$

where the *innovation* $\mathbf{z}_k$ is given by

$$\mathbf{z}_k = \widetilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \widetilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{q}}_k^-) - H_k\Delta\hat{\mathbf{x}}_k^- \tag{5.31}$$

and the Kalman gain is computed as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5.32}$$

Note that since $\Delta\mathbf{x}_k^- = 0$,

$$\Delta\hat{\mathbf{x}}_k^+ = K_k\,\mathbf{z}_k \tag{5.33}$$

$$\mathbf{z}_k = \widetilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{q}}_k^-) \tag{5.34}$$

5.3.3. Step 3: Reset

A reset procedure is used to move information from the updated error state $\Delta\hat{\mathbf{x}}^+$ to the global estimate vector $\hat{\mathbf{q}}^+$ and $\hat{\boldsymbol{\beta}}^+$. This is achieved through

$$\hat{\mathbf{q}}^+ = \boldsymbol{\delta}\mathbf{q}(\hat{\mathbf{a}}^+) \otimes \hat{\mathbf{q}}^- \tag{5.35a}$$

$$\hat{\boldsymbol{\beta}}^+ = \hat{\boldsymbol{\beta}}^- + \Delta\hat{\boldsymbol{\beta}}^+ \tag{5.35b}$$

after which the error state is reset to zero. This operation has to preserve the normalization of the quaternion, and therefore the exact quaternion parametrization $\boldsymbol{\delta}\mathbf{q}(\mathbf{a})$ must be used (the linearized expression in Equation (5.2) or the second order approximation from Section 3.2.7 are not valid).

5.4. IMEKF

As argued in [13] or [34], for spacecraft attitude estimation, the nonlinearity mainly lies in the measurement model. In the MEKF, at time t_k, the measurement sensitivity matrix is computed by linearizing Equation (5.20) around $\hat{\mathbf{q}}_k^-$, which may not be adequate for large attitude errors. The basic idea of the IMEKF is to relinarize the measurement model around the more accurate a posteriori estimate, and the perform once again the measurement update. This can be done a number of times, but according to the results in [13], one iteration strikes a good balance between performance improvement and added computational cost. The IMEKF algorithm is presented in Table 5.2.

5.5. USQUE/MUKF

The unscented version of the MEKF uses the UT to propagate the quaternion and error state covariance, as well as to obtain the predicted measurements $\hat{\mathbf{y}}$. In this section, a formulation based on Crassidis & Markley's USQUE [14] will be presented. In addition to the original USQUE, a modified version using the scaled UT is considered, which will be referred to as MUKF.

Table 5.2: IMEKF.

Initialization

Set $\hat{\mathbf{q}}_0$, $\hat{\boldsymbol{\beta}}_0$ and P_0

Propagation

$\hat{\mathbf{q}}_{k-1}^{+}$, $P_{k-1}^{+} \longrightarrow \hat{\mathbf{q}}_k^{-}$, P_k^{-} using

$\hat{\boldsymbol{\omega}} = \widetilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{k-1}^{+}$

$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\left[\hat{\boldsymbol{\omega}} \otimes\right] \hat{\mathbf{q}}$

$\dot{P} = FP + PF^T + GQG^T$

Update

$\hat{\mathbf{q}}_{k,1} = \hat{\mathbf{q}}_k^{-}$, $\hat{\boldsymbol{\beta}}_{k,1} = \hat{\boldsymbol{\beta}}_k^{-}$, $P_{k,1} = P_k^{-}$

for $i = 1, 2$:

Local update:

$\Delta\hat{\mathbf{x}}_{k,i} = K_{k,i}(\widetilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{q}}_{k,i}))$

$P_{k,i+1} = P_{k,i} - K_{k,i}H_{k,i}P_{k,i}$

where

$K_{k,i} = P_{k,i}H_{k,i}^T(H_{k,i}P_{k,i}H_{k,i}^T + R_{k,i})^{-1}$

$$\mathbf{h}(\hat{\mathbf{q}}_{k,i}) = \begin{bmatrix} A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_1 \\ A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_m \end{bmatrix}, \quad H_{k,i} = \begin{bmatrix} [A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_1\times] & 0_3 \\ [A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_2\times] & 0_3 \\ \vdots & \vdots \\ [A(\hat{\mathbf{q}}_{k,i})\mathbf{r}_m\times] & 0_3 \end{bmatrix}$$

Global update:

$$\Delta\hat{\mathbf{x}}_{k,i} = \begin{bmatrix} \hat{\mathbf{a}}_{k,i} \\ \Delta\hat{\boldsymbol{\beta}}_{k,i} \end{bmatrix}$$

$\hat{\mathbf{q}}_{k,i}^{+} = \boldsymbol{\delta}\mathbf{q}(\hat{\mathbf{a}}_{k,i}^{+}) \otimes \hat{\mathbf{q}}_{k,i}$

$\hat{\boldsymbol{\beta}}_{k,i+1} = \hat{\boldsymbol{\beta}}_{k,i} + \Delta\hat{\boldsymbol{\beta}}_{k,i+1}$

end

$\hat{\mathbf{q}}_k^{+} = \hat{\mathbf{q}}_{k,2}$, $\hat{\boldsymbol{\beta}}_k^{+} = \hat{\boldsymbol{\beta}}_{k,2}$, $P_k^{+} = P_{k,2}$

Table 5.3: USQUE/MUKF.

Initialization	Set $\hat{\mathbf{q}}_0$, $\hat{\boldsymbol{\beta}}_0$ and P_0
Sigma point generation	**USQUE**: $\alpha = 1$, $\kappa = 1$, $\gamma = 0$ **MUKF**: $\alpha = 10^{-3}$, $\kappa = 0$, $\gamma = 2$ $\hat{\mathbf{x}}_{k-1}^{+} = \begin{bmatrix} \hat{\mathbf{a}}_{k-1}^{+} = \mathbf{0}_3 \\ \hat{\boldsymbol{\beta}}_{k-1}^{+} \end{bmatrix}$ $\mathcal{X}_{k-1}(0) = \hat{\mathbf{x}}_{k-1}^{+}$ $\mathcal{X}_{k-1}(i) = \hat{\mathbf{x}}_{k-1}^{+} + \left(\sqrt{(6+\lambda)(P_{k-1}^{+} + Q')}\right)_i, \quad i = 1, ..., 6$ $\mathcal{X}_{k-1}(i) = \hat{\mathbf{x}}_{k-1}^{+} - \left(\sqrt{(6+\lambda)(P_{k-1}^{+} + Q')}\right)_{i-n}, \quad i = 7, ..., 12$ $w_0 = \lambda/(n+\lambda)$ $w_i = 1/(2(n+\lambda)), \quad i = 1, ..., 12$
Propagation	$\hat{\mathbf{q}}_{k-1}^{+}, \hat{\mathbf{x}}_{k-1}^{+}, P_{k-1}^{+} \longrightarrow \hat{\mathbf{q}}_k^{-}, \hat{\mathbf{x}}_k^{-}, P_k^{-}$ using $\hat{\mathbf{q}}_{k-1}(i) = \boldsymbol{\delta}\mathbf{q}_{k-1}(i) \otimes \hat{\mathbf{q}}_{k-1}^{+}, \quad i = 0, 1, ..., 12$ $\hat{\boldsymbol{\omega}}(i) = \widetilde{\boldsymbol{\omega}} - \mathcal{X}_{k-1}^{\beta}(i)$ $\dot{\hat{\mathbf{q}}}(i) = \frac{1}{2}\left[\hat{\boldsymbol{\omega}}(i)\otimes\right]\hat{\mathbf{q}}(i)$ $\mathcal{X}_k^a(i) \longleftarrow \boldsymbol{\delta}\mathbf{q}_k(i) = \hat{\mathbf{q}}_k(i) \otimes \hat{\mathbf{q}}_k^{-1}(0)$ $\hat{\mathbf{q}}_k^{-} = \hat{\mathbf{q}}_k(0)$ $\hat{\mathbf{x}}_k^{-} \longleftarrow$ UT mean of $\mathcal{X}_k^{\beta}(i)$ $P_k^{-} \longleftarrow$ UT covariance of $\mathcal{X}_k^{\beta}(i)$
Local update	$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + K_k\,\mathbf{z}_k$ $P_k^{+} = P_k^{-} - K_k P_k^{z} K_k^{T}$ where $\mathbf{z}_k = \widetilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^{-}$ $K_k = P_k^{xy}(P_k^{z})^{-1}$ $\mathbf{h}(\hat{\mathbf{q}}_k(i)) = \begin{bmatrix} A(\hat{\mathbf{q}}_k(i))\mathbf{r}_1 \\ A(\hat{\mathbf{q}}_k(i))\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}_k(i))\mathbf{r}_m \end{bmatrix}$
Global update	$\hat{\mathbf{q}}_k^{+} = \boldsymbol{\delta}\mathbf{q}(\hat{\mathbf{a}}_k^{+}) \otimes \hat{\mathbf{q}}_k^{-}$

The USQUE/MUKF algorithm is presented in Table 5.3. In order to formulate it, a different, more general approach than the one used for the MEKF is required. Using a Bayesian approach[1], and assuming that all variables are Gaussian distributed, the following general equations for optimal recursive estimation can be obtained [35]

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k\,\mathbf{z}_k \tag{5.36a}$$
$$P_k^+ = P_k^- - K_k P_k^z K_k^T \tag{5.36b}$$

where $\mathbf{z}_k = \widetilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^-$ is known as the innovation, and P_k^z corresponds to its covariance. For the system model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \tag{5.37a}$$
$$\widetilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{5.37b}$$

the optimal terms in this recursion are given by

$$\hat{\mathbf{x}}_k^- = E\{\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1})\} \tag{5.38a}$$
$$K_k = P_k^{xy}(P_k^z)^{-1} \tag{5.38b}$$
$$\hat{\mathbf{y}}_k^- = E\{\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)\} \tag{5.38c}$$

where P_k^{xy} is the cross-correlation matrix between $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{y}}_k^-$. The original Kalman filter calculates all terms in these equations exactly in the linear case. For nonlinear models, however, the EKF approximates the optimal terms as:

$$\hat{\mathbf{x}}_k^- \approx \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \bar{\mathbf{w}}_{k-1}) \tag{5.39a}$$
$$K_k \approx \hat{P}_k^{xy}(\hat{P}_k^z)^{-1} \tag{5.39b}$$
$$\hat{\mathbf{y}}_k^- \approx \mathbf{h}(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k)) \tag{5.39c}$$

where the predictions are approximated simply as the function of the prior *mean* values[2], which for the state correspond to the prior estimates. In the MEKF, $\hat{\mathbf{x}}$ would correspond to the error state. First, $\hat{\mathbf{a}}$ is assumed to remain zero after the propagation (instead of propagating its distribution and then computing its mean), and the attitude error covariance P_{k-1} is propagated using a first order linearization of $\mathbf{f}$ around $\hat{\mathbf{a}}_{k-1} = \mathbf{0}_3$ (always zero after the reset).

[1] The Kalman filter may be derived using various approaches. It was originally formulated through the minimization of the mean-square error (MMSE) between predicted and actual measurements, but Bayesian inference provides a more general framework.

[2] The noise means are denoted by $\bar{\mathbf{w}} = E\{\mathbf{w}\}$ and $\bar{\mathbf{v}} = E\{\mathbf{v}\}$ are are usually assumed to equal zero.

Similarly, just the estimated state is used to compute $\hat{\mathbf{y}}_k^-$ (instead of transforming the complete state distribution and then computing its mean), and a linear approximation of $\mathbf{h}$ is used to compute P_k^{xy} and P_k^z ($\hat{P}_k^{xy} = P_k^- H_k^T$ and $\hat{P}_k^z = H_k P_k^- H_k^T + R_k$).

On the other hand, the unscented version of the MEKF uses the UT to propagate/transform the mean and covariance of the state variables. First, the state is redefined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\beta} \end{bmatrix} \tag{5.40}$$

where the bias, and not the bias error as in the MEKF, is being considered. At time t_k, the sigma points and associated weights are computed using the expressions in Section 3.4.2:

$$\begin{aligned}
\mathcal{X}_k(0) &= \hat{\mathbf{x}}_k \\
\mathcal{X}_k(i) &= \hat{\mathbf{x}}_k + \left(\sqrt{(n+\lambda)(P_k + Q')}\right)_i, \quad i = 1, 2, ..., n \\
\mathcal{X}_k(i) &= \hat{\mathbf{x}}_k - \left(\sqrt{(n+\lambda)(P_k + Q')}\right)_{i-n}, \quad i = n+1, n+2, ..., 2n \\
w_0 &= \frac{\lambda}{n+\lambda} \\
w_i &= \frac{1}{2(n+\lambda)}, \quad i = 1, 2, ..., 2n
\end{aligned} \tag{5.41}$$

where $n = 6$ (state dimension), $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter, and Q' is related to the process noise covariance, which will be discussed shortly. $\left(\sqrt{(n+\lambda)(P_k + Q')}\right)_i$ is the ith row of the matrix square root, which can be computed using the Cholesky decomposition. Each vector $\mathcal{X}_k(i)$ may be partitioned into

$$\mathcal{X}_k(i) = \begin{bmatrix} \mathcal{X}_k^a(i) \\ \mathcal{X}_k^\beta(i) \end{bmatrix}, \quad i = 0, 1, ..., 12 \tag{5.42}$$

In order to propagate the quaternion distribution using the UT, a "sigma set" of quaternions is generated:

$$\hat{\mathbf{q}}_k(i) = \boldsymbol{\delta}\mathbf{q}_k(i) \otimes \hat{\mathbf{q}}_k, \quad i = 0, 1, ..., 12 \tag{5.43}$$

where $\boldsymbol{\delta}\mathbf{q}_k(i) = \boldsymbol{\delta}\mathbf{q}(\mathcal{X}_k^a(i))$ shall be computed using the exact quaternion parametrization in order to preserve normalization. Note that $\hat{\mathbf{q}}_k(0) = \hat{\mathbf{q}}_k$, since the attitude error vector is reset before the propagation. The updated quaternions are propagated forward using Equation (5.12), each with an estimated angular velocity given by

$$\hat{\omega}(t) = \widetilde{\omega}(t) - \mathcal{X}_k^\beta(i) \quad \forall t \in [t_k, t_{k+1}], \quad i = 0, 1, ..., 12 \tag{5.44}$$

At t_{k+1}, the propagated sigma points $\mathcal{X}^a_{k+1}(i)$ can be computed using

$$\boldsymbol{\delta}\mathbf{q}_{k+1}(i) = \hat{\mathbf{q}}_{k+1}(i) \otimes \hat{\mathbf{q}}^{-1}_{k+1}(0), \quad i = 0, 1, ..., 12 \tag{5.45}$$

and the expressions in Section 3.2 relating the quaternion to other attitude representations (for instance, if $\mathbf{a} = 4\,\boldsymbol{\delta}\mathbf{p}$, then $\mathcal{X}^a(i) = 4\,\boldsymbol{\delta}\mathbf{q}_{1:3}(i)/(1 + \delta q_4(i))$).

The predicted mean and covariance can now be computed using

$$\hat{\mathbf{x}}_{k+1} \approx \sum_{i=0}^{2n} w_i \mathcal{X}_{k+1}(i) \tag{5.46a}$$

$$\begin{aligned} P_{k+1} \approx & \sum_{i=0}^{2n} w_i (\mathcal{X}_{k+1}(i) - \hat{\mathbf{x}}_{k+1})(\mathcal{X}_{k+1}(i) - \hat{\mathbf{x}}_{k+1})^T \\ & + (1 - \alpha^2 + \gamma)(\mathcal{X}_{k+1}(0) - \hat{\mathbf{x}}_{k+1})(\mathcal{X}_{k+1}(0) - \hat{\mathbf{x}}_{k+1})^T + Q' \end{aligned} \tag{5.46b}$$

where γ is an auxiliary parameter of the scaled UT.

Recall that in the formulation of the MEKF, the error state covariance was propagated using the linearized kinematics in Equation (5.9) and a recursive integration technique based on n gyro measurements between t_k and t_{k+1}. With this approach, process noise was seamlessly incorporated into the propagated value. Unfortunately, adopting a similar strategy here would imply computing $\hat{\mathbf{a}}$ and P, adding $Q_{\Delta\tau}$ (with $\Delta\tau$ equal to the gyro sampling interval), and regenerating all sigma points at every gyro sampling time. For obvious reasons, this is very impractical and computationally inefficient. One solution is to consider $Q_{\Delta t}$, with $\Delta t = t_{k+1} - t_k$, and add it to the propagated covariance in Equation (5.46b). Crassidis & Markley propose a more sophisticated approach, approximating the integration of process noise by adding the matrix

$$Q' = \frac{\Delta t}{2} \begin{bmatrix} (\sigma_v^2 - \frac{1}{6}\sigma_u^2 \Delta t^2) I_3 & O_3 \\ 0_3 & \sigma_u^2 I_3 \end{bmatrix} \tag{5.47}$$

before and after the propagation, akin to a trapezoidal rule for integration. This is reflected in the previous formulation.

Proceeding with the measurement update, the mean observation and output covariance are given by

$$\hat{\mathbf{y}}_{k+1} \approx \sum_{i=0}^{2n} w_i \mathcal{Y}_{k+1}(i) \tag{5.48a}$$

and

$$P^y_{k+1} \approx \sum_{i=0}^{2n} w_i(\mathcal{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1})(\mathcal{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1})^T + (1 - \alpha^2 + \gamma)(\mathcal{Y}_{k+1}(0) - \hat{\mathbf{y}}_{k+1})(\mathcal{Y}_{k+1}(0) - \hat{\mathbf{y}}_{k+1})^T \tag{5.48b}$$

where

$$\mathcal{Y}_{k+1}(i) = \mathbf{h}(\mathcal{X}_{k+1}(i)), \quad i = 0, 2, ..., 12 \tag{5.49}$$

Given that the Jacobian matrix of $\mathbf{h}$ does not need to be derived, any measurement function may be used, including the two quantity observation models in Section 4.3.2. While Crassidis & Markley implement an unit vector observation model in [14], they recommend that two quantity observation models are used instead, since Equation (5.48b) may not produce a mean observation with unit norm. For this project both alternatives have been tested; with the selected sensor suite, the resulting attitude errors agree to within 1 arcsec. While the results using a two quantity observation model are indeed more precise, such a small improvement does not justify the added computational cost. For this reason, the vector observation model will be kept:

$$\mathbf{h}(\mathcal{X}_{k+1}(i)) = \begin{bmatrix} A(\hat{\mathbf{q}}_{k+1}(i))\mathbf{r}_1 \\ A(\hat{\mathbf{q}}_{k+1}(i))\mathbf{r}_2 \\ \vdots \\ A(\hat{\mathbf{q}}_{k+1}(i))\mathbf{r}_m \end{bmatrix} \tag{5.50}$$

The innovation covariance is simply given by

$$P^z_{k+1} = P^y_{k+1} + R_{k+1} \tag{5.51}$$

and the cross-correlation matrix is computed as

$$P^{xy}_{k+1} \approx \sum_{i=0}^{2n} w_i(\mathcal{X}_{k+1}(i) - \hat{\mathbf{x}}_{k+1})(\mathcal{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1})^T + (1 - \alpha^2 + \gamma)(\mathcal{X}_{k+1}(0) - \hat{\mathbf{x}}_{k+1})(\mathcal{Y}_{k+1}(0) - \hat{\mathbf{y}}_{k+1})^T \tag{5.52}$$

After adding a "$-$" superscript to the previous a priori values, the update may be performed using

$$\hat{\mathbf{x}}^+_{k+1} = \hat{\mathbf{x}}^-_{k+1} + K_{k+1}\,\mathbf{z}_{k+1} \tag{5.53a}$$

$$P^+_{k+1} = P^-_{k+1} - K_{k+1}P^z_{k+1}K^T_{k+1} \tag{5.53b}$$

with

$$\mathbf{z}_{k+1} = \widetilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}^{-}_{k+1} \tag{5.54}$$

$$K_{k+1} = P^{xy}_{k+1}(P^{z}_{k+1})^{-1} \tag{5.55}$$

after which $\hat{\mathbf{a}}^{+}_{k+1}$ is reset to zero.

The scaled UT used in the current formulation differs from the older version used in [14]. However, the same results[1] may be obtained using $\alpha = 1$, $\kappa = 1$ and $\gamma = 0$. A modern version based on the typical scaled UT parameters ($\alpha = 10^{-3}$, $\kappa = 0$, $\gamma = 2$) and receiving the generic name MUKF (multiplicative unscented Kalman filter) will also be tested. Note that the MUKF parameters are designed for Gaussian distributions, minimizing higher order moments. On the other hand, those of the USQUE can introduce significant higher order moments, which may produce better results if they are aligned with those of the true distribution (or poorer results if they are in the opposite direction).

5.6. Bias Clipping

The bias clipping technique basically consists on artificially limiting (or constraining) the bias estimation within the filter. Constrained Kalman filtering is not a new idea [36], [37], and has been used extensively in fields such as chemical engineering, where e.g. concentrations of a solution may be forced to take positive values. Ultimately, adding constraints to the estimated values is a way of incorporating additional knowledge about certain variables. To the knowledge of the author, however, there has not been investigation of a constrained Kalman filter for spacecraft attitude estimation.

For large initialization errors, the filtering performance of the MEKF may be significantly degraded due to linearization errors. This is a well documented limitation of extended Kalman filters, and has motivated the development of alternatives such as the IMEKF and USQUE. During the convergence phase, the state variables are typically subject to large oscillations before settling down, which negatively impacts the speed of convergence. Regarding the unconstrained attitude error vector, the values reached during these oscillations do not present any conceptual issues. However, gyro biases may reach values which are physically unattainable. As stated in Section 4.3.3, gyro bias is not truly unbounded as the used model may imply, but rather limited by the physical properties of the gyro (consider the finite turn-on bias).

In addition, the probability density function of the bias estimate is not unimodal (more than one peak exist), which can cause the filter to converge to the wrong value.

[1] In [14], different values of λ are studied, concluding that the USQUE performs best using $\lambda = 1$.

This is easy to visualize; imagine that the spacecraft is spinning around a certain axis, and multiple sensors are providing measurements at fixed time intervals. In order to correctly capture the spacecraft's motion, measurements are sampled at a frequency which is multiple times greater than the rotational frequency of the spacecraft (obeying the Nyquist sampling theorem). However, an unbounded gyro bias estimate may theoretically converge to a value that increases the estimated angular velocity over the Nyquist limit. Therefore, it is possible for the estimate to be erroneous while still agreeing with the measurements. This has been observed in simulations with a large angular velocity and relatively slow sensor sampling frequency (that is, with a reduced "Nyquist sampling margin").

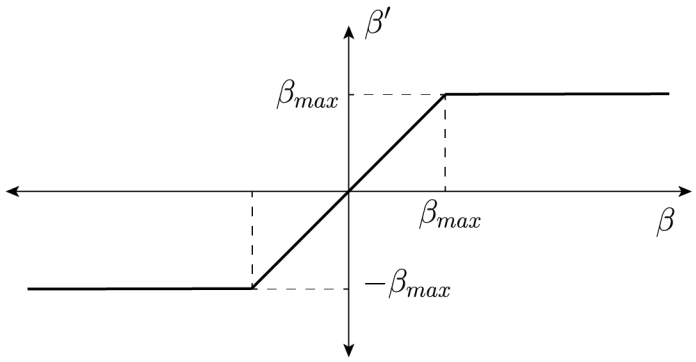


Figure 5.1: Saturation of the gyro bias estimate.

The issues stated in the previous discussion motivate the inclusion of an artificial gyro bias constraint. The *bias clipping* strategy achieves this by implementing a saturation effect on the estimated a posteriori value of the gyro bias. This effect is illustrated in Figure 5.1, where β' denotes the saturated bias. The value β_{max} must be selected in accordance with the gyro technical specifications. While the values of β_{max} indicated in Table 4.2 may be adopted directly, a safety factor of two will be taken into account (the bias estimate will be saturated at twice the indicated values).

Note that the results presented in this work are based on simulated measurements, and not on real sensor outputs. A practical application of the bias clipping technique must consider a previous characterization of β_{max} based on extensive gyro testing, and greater safety factors may be applied.

6 Simulation and Results

The objective of this Chapter is to assess the precision in attitude knowledge that can be achieved using the sensor suite selected in Chapter 4 and the various filters introduced in Chapter 5. Several performance comparisons between the MEKF, IMEKF and USQUE/MUKF are made through simulated examples using a realistic spacecraft model. The attitude estimation algorithms are implemented using Python. Overall, the results presented in this chapter provide valuable insights into the effectiveness of the various filtering methods and justify the sensor selection.

6.1. Simulation Strategy

6.1.1. General Considerations

The simulations are performed using a representative Earth-pointing spacecraft in a circular orbit at an altitude of 530 km (orbital period of $\sim$ 95.2 min). Additionally, the orbit is Sun synchronous with a corresponding inclination of 97.51°. This represents the most popular orbit for nanosatellites, with over 300 active spacecraft as of early 2023 [1]. Other hypothesis include the lack of perturbations altering the initial orbit or the spacecraft attitude, and perfect attitude control (the spacecraft will follow the specified angular velocity profile).

Recall that the sensor suite consists of a star tracker, fine sun sensors, a three-axis magnetometer (TAM) and a three-axis gyroscope. It is assumed that the Sun is within a Sun sensor's field of view at all times other than during eclipse. Additionally, the star tracker is pointing towards the z axis of the spacecraft body frame. The simulations will be carried out for the two selected gyro models (CRM100 and STIM202), in order to assess their performance and viability.

The sensors have the technical characteristics specified in Section 4.4, and are simulated using the various models introduced in Section 4.3. In particular, the two angles model is used to simulate star tracker and Sun sensor measurements, which are then fed to the filters as vector observations. Additionally, TAM measurements are generated using

an updated IGRF model (early 2023), while the reference model will be offset from the truth epoch by three years. This introduces a systematic error component which is added to the Gaussian measurement noise. Lastly, gyro measurements are simulated using the discrete-time model in Section 4.6. Note that the sampling frequency of each sensor will depend on the analyzed scenario, and is specified in the description of each particular case.

6.1.2. Case Studies

In this section, three different scenarios are presented. First, normal mode is studied, as it is the standard operating mode of a spacecraft's ADS and generally covers the majority of the mission's lifetime. The main objective is to assess steady state accuracy of the attitude estimation, comparing the performance of the different filters.

Next, a slew maneuver is analyzed. A change in the pointing direction of a satellite may be performed for various reasons, which include:

- Satellite commissioning and decommissioning.
- Orbital station keeping.
- Calibrating and testing the attitude control system.
- Pointing the solar panels towards the Sun to recharge its batteries or optimize power generation.
- Changing the orbital plane or altitude of the satellite.

With a time-varying rotational speed, the multiple assumptions of constant angular velocity in the filtering algorithms are tested.

Lastly, a scenario of complete loss of attitude knowledge is studied. This may happen during a spacecraft's lifetime due to sensor failure, radiation effects, actuator malfunction, or software or communication errors, and can be a critical issue. Typically, when the spacecraft experiences an anomaly or malfunction that may threaten its operational status or safety, it enters the so called *safe* mode. Most systems are powered down or switched to low-power mode, and only essential components are maintained. In this scenario, the limits of the attitude determination module will be pushed, finding the minimum set of sensors required to recover attitude knowledge to an acceptable degree of accuracy.

The previous scenarios represent typical phases or modes throughout the average mission's lifetime, and allow to test the module design in a wide range of conditions. Other common phases/modes such as detumbling, Sun-pointing mode or target-pointing

mode will not be considered, as they become redundant and do not add much value to the analysis.

Case 1: Nominal Earth-Pointing Conditions

In normal mode, the spacecraft is continuously pointing towards the Earth, and thus completing a full rotation each orbit (resulting in an angular velocity of 0.063 deg/s in the direction perpendicular to the orbital plane). The complete sensor suite will be used in order to assess whether there is any performance improvement over star tracker-only attitude estimation. Additionally, several combinations of sensor sampling rates will be analyzed to quantify the advantage (if any) in using higher rates. Other relevant aspects of the simulation are outlined below:

- Total simulation time of 10 minutes.
- Eclipse conditions during the second half of the simulation.
- Complete availability of all the sensors except for the Sun sensor during eclipse. This assumes that no Sun intrusions in the star tracker's FOV are present.
- Constant average number of six identified stars by the star tracker. In reality, this number typically oscillates between one and about ten stars..
- Initial attitude-estimation error: $\boldsymbol{\delta\vartheta}_0 \sim N(\boldsymbol{\mu}_{\delta\theta}, \sigma^2_{\delta\vartheta} I_3)$, with $\boldsymbol{\mu}_{\delta\theta} = [0°, 0°, 0°]^T$ and $\sigma_{\delta\vartheta} = 0.1°$.
- Initial bias-estimation error: $\Delta\boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu}_{\Delta\beta}, \sigma^2_{\Delta\beta} I_3)$, with $\boldsymbol{\mu}_{\Delta\beta} = [0, 0, 0]^T$ deg/s and $\sigma_{\Delta\beta}$ equal to the standard deviation of the corresponding turn-on bias.

The initial conditions mimic a typical situation, where normal operation begins after an initial attitude acquisition phase using Sun sensor and TAM measurements only. A relatively accurate attitude estimation is already available, and therefore there are no filter convergence issues. On the other hand, no previous gyro bias estimate exists and it is set to zero. The fact that the true bias is initially distributed according to the turn-on bias specification leads to the indicated distribution of $\Delta\boldsymbol{\beta}_0$.

Note that the initial attitude-estimation error is being expressed in terms of the rotation vector, but the filters employ a different error representation, namely $\mathbf{a} = 4\,\boldsymbol{\delta}\mathbf{p}$. However, since the error is small, both representations are virtually equivalent and therefore just a simple change in units (from degrees to radians) is required.

Also note that multiple simulations (i.e. Monte-Carlo runs) will be carried out to av-

erage the results. In each simulation, the true quaternion[1] and gyro bias will be generated using a different realization of the previous initial error distributions.

Case 2: Slew Maneuver

The angular velocity profile of a slew maneuver can vary depending on the specific requirements of the mission and the design and size of the satellite. However, in general, a typical slew maneuver involves an initial acceleration phase, followed by a constant velocity phase, and then a deceleration phase as the satellite approaches its target attitude. A representative angular velocity profile for small satellites can be found in [38], showing a maximum magnitude of approximately 2 deg/s and a total duration of 90 s. To simplify computations, a slightly simpler profile will be employed, using "rest-to-rest" motion (the maneuver endpoints are inertially fixed). The magnitude of the angular velocity $\omega(t) = \|\boldsymbol{\omega}(t)\|$ shall verify:

- Start and finish at rest: $\omega(0) = \omega(T) = 0$, where $T = 90$ s.
- Start and finish with zero torque: $\dot{\omega}(0) = \dot{\omega}(T) = 0$.
- Reach a maximum magnitude[2] $\omega_{max} \simeq 2$ deg/s.

The polynomial $\omega(t) = Ct^2(T-t)^2$ automatically fulfills the first two requirements, and choosing $C = 5 \cdot 10^{-7}$ deg/s^5 achieves $\omega_{max} = 2.05$ deg/s. The resulting angular velocity profile is displayed in Figure 6.1. Integrating the previous expression and computing its derivative gives a total rotation of 98.4° and a maximum angular acceleration of magnitude $\dot{\omega}_{max} = 0.07$ deg/s^2 respectively.

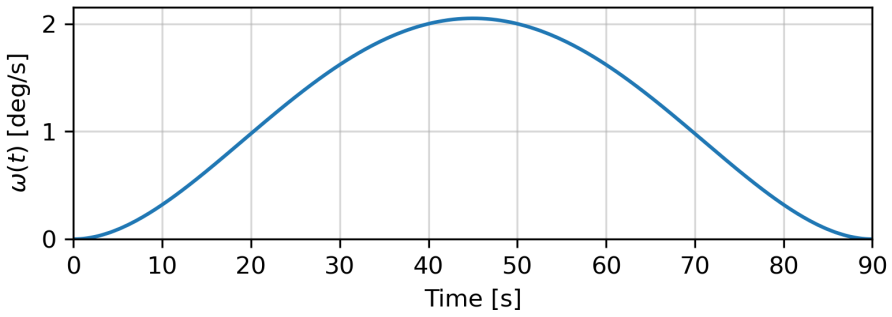


Figure 6.1: Angular velocity profile used for the simulations.

The main objective is to test whether nominal accuracy is maintained during a slew

[1] Adding the error to the true quaternion in order to generate the estimate or doing it the other way around are equivalent.

[2] Note that ω_{max} may be limited by the star tracker's maximum slew rate. Coincidentally, 2 deg/s matches that of the selected model.

maneuver. This will be assessed using various gyro sampling rates, which are directly related to the precision in attitude propagation. Exact initial attitude and gyro bias knowledge is assumed ($\sigma_{\delta\vartheta} = \sigma_{\Delta\beta} \simeq 0$), as well as complete availability of all the sensors. For the purposes of this analysis, the axis of rotation does not have much influence. Nevertheless, an axis which is not aligned with the Sun nor the star tracker's boresight will be chosen.

Case 3: Lost In Space

Here, the initialization problem is addressed. Complete loss of attitude knowledge will be simulated, testing the different filters' ability to converge to a correct estimate. A worst-case scenario, where only the gyroscope and either the Sun sensor or TAM are available due to a failure in the star tracker, is studied. The goal is to analyze which combinations of gyro model (CRM100 vs. STIM202) and attitude sensor allow the filter to converge. Additionally, the case where both attitude sensors are used will be considered, simulating regular eclipse cycles.

One of the main goals of this simulation is to assess the speed of convergence of the different algorithms, as well as the achievable accuracy using Sun sensor and magnetometer measurements only. Total attitude and bias uncertainty are assumed, which lead to the following initial conditions:

- Initial attitude-estimation error: $\boldsymbol{\delta\vartheta}_0 \sim N(\boldsymbol{\mu}_{\delta\vartheta}, \sigma^2_{\delta\vartheta} I_3)$, with $\boldsymbol{\mu}_{\delta\vartheta} = [0°, 0°, 0°]^T$ and $\sigma_{\delta\vartheta} = 60°$.
- Initial bias-estimation error: $\Delta\boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu}_{\Delta\beta}, \sigma^2_{\Delta\beta} I_3)$, with $\boldsymbol{\mu}_{\Delta\beta} = [0, 0, 0]^T$ deg/s and $\sigma_{\Delta\beta}$ equal to the standard deviation of the corresponding turn-on bias.

Even if the value of $\sigma_{\delta\vartheta}$ is quite large, it still can be directly translated into MRP error ($4\tan^- 1[(60 \times \pi)/(4\times)] = 58.68°$). In this context, it represents a complete lack of confidence in the provided attitude estimate.

The same attitude error distribution can be used to generate the true quaternion, which in terms of Euler angle results in an approximate mean error of $\sqrt{3} \times 60° \simeq 104°$. However, in order to simulate a worst-case scenario, each error component will instead be uniformly distributed, ranging from $-180°$ to $180°$. Using this strategy, a Monte-Carlo analysis with 10000 samples indicates an expected Euler angle error of about $150°$. Note that in order to perform a realistic analysis and truly capture the average characteristics of the various filters, it is important to provide several different initial attitude errors. As the simulations will show, using a fixed value may lead to wrong conclusions.

6.2. Simulation Results

In order to quantify the simulation results and assess filter performance, several parameters are analyzed:

- **Pointing error**: Characterized by the components of the local error vector $\mathbf{a}(t)$ or the error Euler angle $\delta\vartheta(t)$ (from the Euler angle/axis representation).
- **Computation time**: Run time for each filter, expressed as a ratio to the MEKF run time (i.e. MUKF run time = $ratio \times$ MEKF run time).
- **Convergence time**: Time it takes for the filter to converge within its 3σ bounds.

These will be averaged through 20 Monte-Carlo runs, which in the author's experience are representative enough. The time average of the resulting averaged error Euler angle, referred to as the root-mean-square error (RMSE), will also be considered. It represents the mean error that can be expected at any particular time, and provides a scalar performance indicator which eases the filter comparison. If the Monte-Carlo average of the error Euler angle is denoted by $\delta\vartheta_{MC}(t)$, then the RMSE is simply defined as

$$\text{RMSE} = \frac{1}{N}\sum_{i=1}^{N}\delta\vartheta_{MC}(t_i) \tag{6.1}$$

where N is the total amount of attitude estimates during the simulation. Note that when the gyro sampling rate is higher than any of the attitude sensor's sampling rates, an attitude estimate is generated at each gyro sampling time. Otherwise, N will correspond to the total number of measurement updates.

Pointing errors will be computed with respect to a "true" quaternion trajectory $\mathbf{q}(t)$, which has been propagated using direct numerical integration of a "true" angular velocity $\boldsymbol{\omega}(t)$. Note that these true parameters are also used to generate the attitude and gyro measurements. Regarding the chosen error characterizations, recall that any of the local error vector representations for $\mathbf{a}(t)$ agree to first order with the Euler angles, providing an intuitive interpretation. For instance, the first component will represent a rotation around the x axis of the spacecraft body frame, the second component a rotation around the y axis, and the third component a rotation around the z axis. Both the components of the local error vector $\mathbf{a}(t)$ and the error Euler angle $\delta\vartheta(t)$ correspond to the ESA recommended pointing error specifications [39].

6.2.1. Case 1: Nominal Earth-Pointing Conditions

A number of different configurations are analyzed in order to assess the filtering performance of the various algorithms using the selected gyro models, and determine the optimal sampling rate of each sensor. First, the complete sensor suite is considered, at sampling rates of 1 Hz, 5 Hz and 50 Hz (except for the star tracker, which is limited to 5 Hz). Additionally, a simulation using star tracker and gyro measurements only is studied. Results for the RMSE (expressed in arcseconds) are presented in Table 6.1, where the best performing configurations have been highlighted.

Table 6.1: Case 1: RMSE [arcsec] for the different configurations.

Gyro Model	Sensor Configuration	MEKF	IMEKF	MUKF	USQUE
CRM100	All @ 1 Hz	53.15	54.72	52.68	52.68
	All @ 5 Hz	33.08	33.71	33.71	33.71
	STR/gyro @ 5 Hz	**32.84**	33.56	**32.84**	**32.84**
	All @ 50 Hz (STR @ 5 Hz)	35.15	35.85	35.15	35.15
STIM202	All @ 1 Hz	28.34	29.03	28.34	28.34
	All @ 5 Hz	18.69	19.20	18.69	18.69
	STR/gyro @ 5 Hz	**18.61**	19.10	**18.61**	**18.61**
	All @ 50 Hz (STR @ 5 Hz)	19.49	19.96	19.49	19.49

From these values, several important conclusions can be extracted. First, it is clear that all filters offer very close performance. For small initial errors, the unscented versions generally produce results which are practically identical to those of the MEKF, which indicate that the linearity assumptions are valid. The IMEKF exhibits very similar performance as well, but consistently generates slightly worse estimates. This agrees with the results presented in [13].

Secondly, the contrast in accuracy using the CRM100 and STIM202 gyros is quite obvious. By using the STIM202, a $\sim$ 46% decrease in attitude-estimation error can be obtained. Keep in mind that the difference is still in the order of tens of arcseconds, which may be significant or not depending on the application. It must be noted, however, that a precision of even a few hundreds of arcseconds will satisfy the pointing requirements of any typical mission for small spacecraft.

Regarding the sensor configuration, several distinct effects can be observed. On one hand, the results show that increasing the sensor sampling rates from 1 Hz to 5 Hz leads to a $\sim$ 36% decrease in attitude-estimation error. Again, whether this improvement is significant or not will depend on the specific application. A more interesting observation,

however, is that Sun sensor and TAM measurements do not improve accuracy, and may even slightly degrade filter performance. Finally, at 50 Hz, there is a noticeable increase in attitude error with respect to the 5 Hz case. Rather than being due to SS/TAM measurements, however, it is largely due to the higher gyro sampling rate (a simulation with star tracker measurements only has been carried out, producing very similar results). As with most inertial measurements, decreasing the gyro sampling rate implies an averaging process which reduces measurement noise. Thus, it appears that fewer, less noisy gyro measurements are preferred. This gyro oversampling effect will be subject to further analysis in Case 2.

In terms of computation time, the IMEKF takes on average $\sim 70\%$ longer than the MEKF to perform each simulation (IMEKF run time $\sim 1.7\times$ MEKF run time), and the MUKF/USQUE take on average $\sim 750\%$ longer (MUKF/USQUE run time $\sim 8.5\times$ MEKF run time). Although the total run time does change depending on the sensor sampling rates (at 5 Hz the simulations take approximately five times more, and at 50 Hz approximately 50 times more), these ratios remain relatively fixed. While the IMEKF does not add excessive run time, the unscented filters come with a substantial computational cost, which may prohibit its use for onboard implementations.

In nominal filter operation, it is clear that the best results are obtained using the MEKF to filter star tracker measurements only. One one hand, the MEKF performs significantly faster than any other filter. On the other hand, adding SS/TAM measurements has proven to degrade attitude-estimation accuracy. Whether the sampling rate shall be set to 1 Hz, 5 Hz, or any other value, will depend on the pointing requirements of the spacecraft (a higher sampling rate implies a higher precision but also higher computational cost). The gyro selection, in contrast to the sampling rate, cannot be modified onboard and therefore will be considered after reviewing the results for cases 2 and 3.

Next, some graphical results using the MEKF and star tracker measurements only at 5 Hz and 1 Hz are presented. First, the error Euler angle with its respective 3σ bound is shown in Figure 6.2. The time-averaged 3σ bounds of the various configurations accurately match three times the RMSE values in Table 6.1, which indicates that the filter is well tuned (the process noise has been correctly modeled). Notice that the error is quite noisy, in contrast to typical results that may be seen in the open literature. This is due to the low quality gyros used, which greatly decrease the confidence in a priori values and therefore transfer star tracker noise directly into the estimate. Another remarkable aspect is the fact that the filter converges almost instantaneously. Once again, this is caused by the continuous lack of confidence in a priori values, which lead the filter to quickly adopt the attitude derived from star tracker measurements.

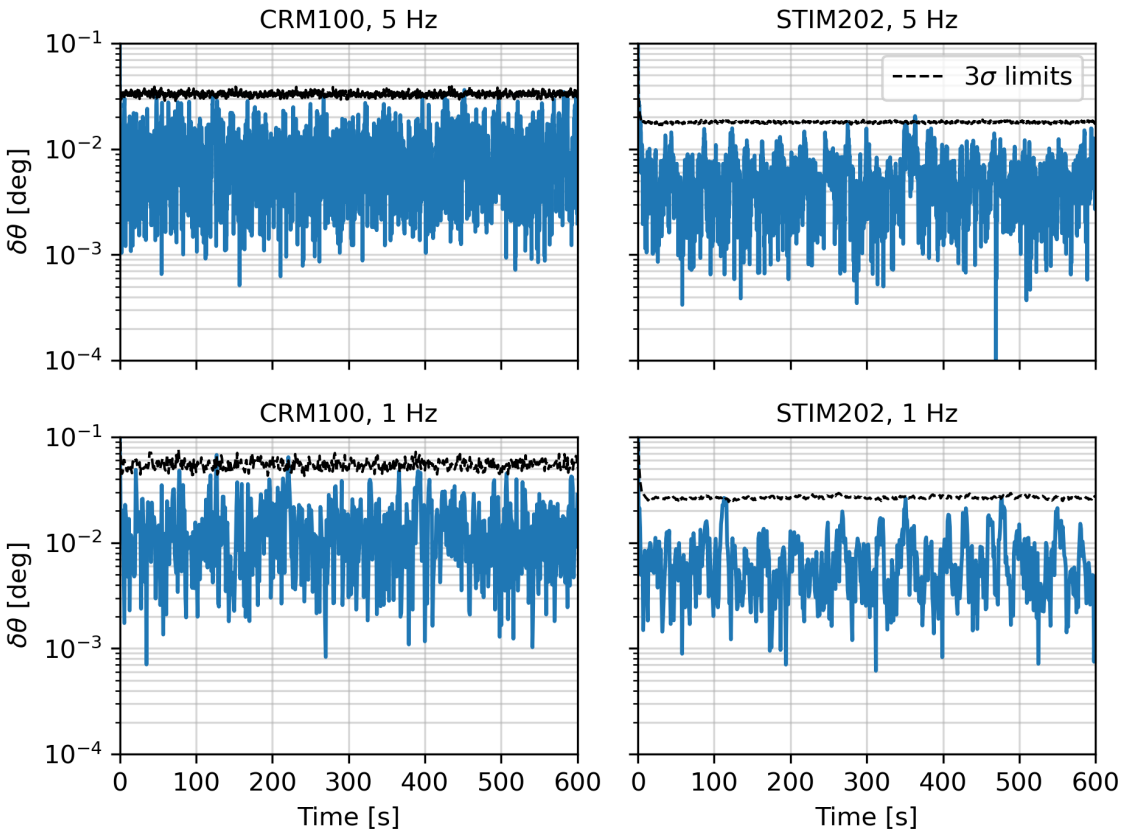


Figure 6.2: Case 1: Error Euler angle and 3σ bound.

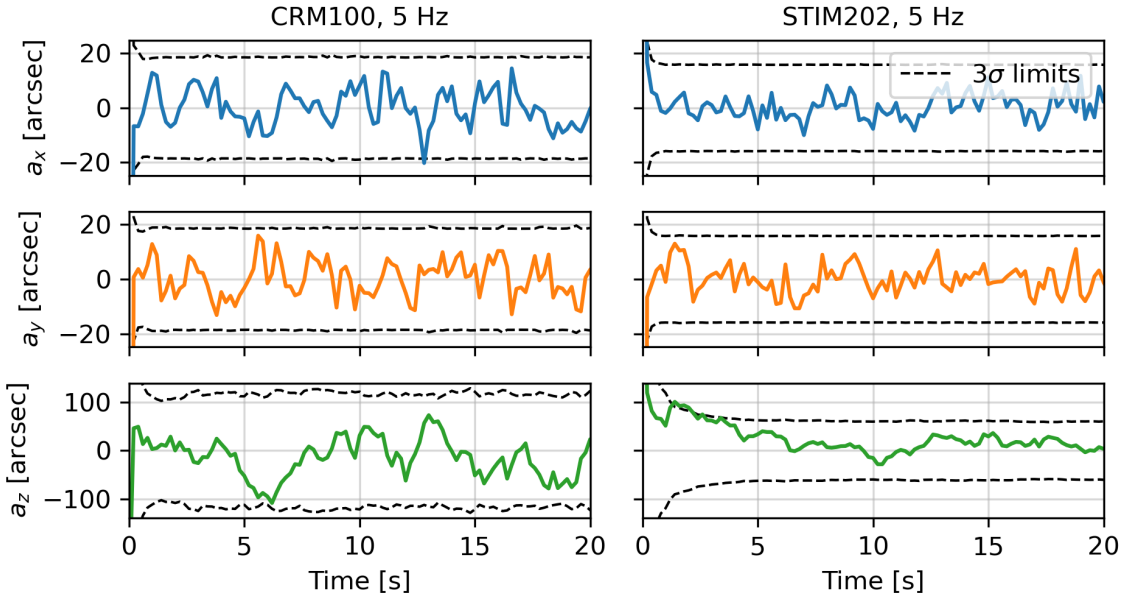


Figure 6.3: Case 1: Attitude error vector and 3σ bounds at a sampling rate of 5 Hz.

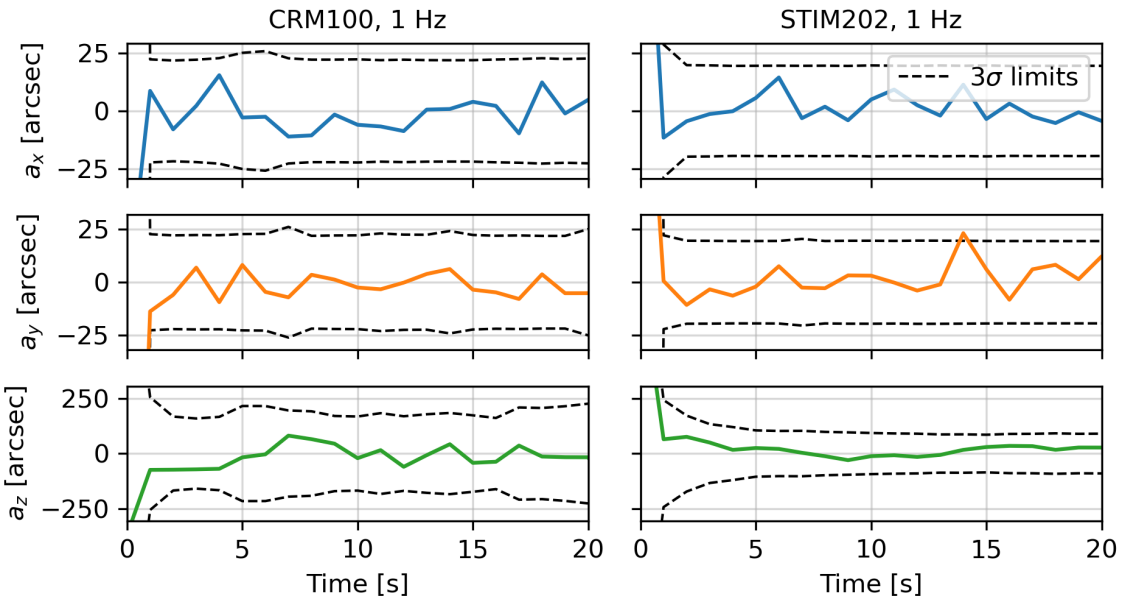


Figure 6.4: Case 1: Attitude error vector and 3σ bounds at a sampling rate of 1 Hz.

In Figures 6.3 and 6.4, the components of the local error vector in the spacecraft body frame and their respective 3σ bounds are shown for the first 20 seconds of the simulation. Recall that the star tracker boresight is aligned with the z axis of the spacecraft, and the specifications in Table 4.1 indicate an across-boresight error of 30 arcsec (3σ) and an along-boresight error of 200 arcsec (3σ), which correspond to those obtained using static attitude determination methods. As the results show, these can be improved using a filtering approach. On one hand, across-boresight errors (a_x, a_y) are reduced in approximately 25% to 50% using the several gyro and sampling rate combinations. A more significant effect can be observed in the along-boresight component; with a sampling rate of 5Hz and using the CRM100 gyro, the 3σ bounds stabilize at around 120 arcsec. Using the STIM202 gyro instead, they stabilize at $\sim$ 60 arcsec. With a sampling rate of 1 Hz, these values increase to 200 arcsec and 90 arcsec respectively. Based on these results, it can be concluded that the 1 Hz, CRM100 configuration offers barely any improvement over static star tracker performance, while using the STIM202 at 5 Hz can reduce the along-boresight error to approximately one third of the original value. With the other configurations, intermediate values are obtained.

Regarding the bias estimation, some results are presented next. In Figure 6.5, the evolution of the estimated and true bias components is shown for both gyroscopes at a sampling rate of 5 Hz. Note that the bias is expressed in an inertial reference frame, whose axes will be denoted by 1, 2 and 3. It is interesting to observe the nature of bias drift, and it is clear that the filter manages to produce a good estimate. While it may appear that

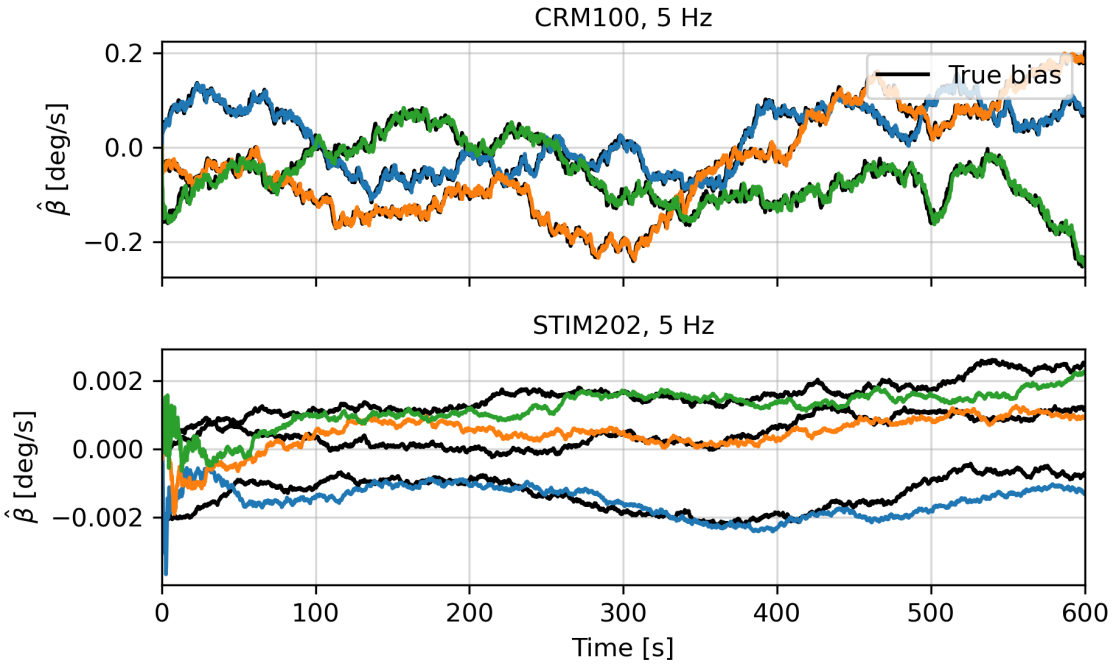


Figure 6.5: Case 1: Estimated and true bias evolution.

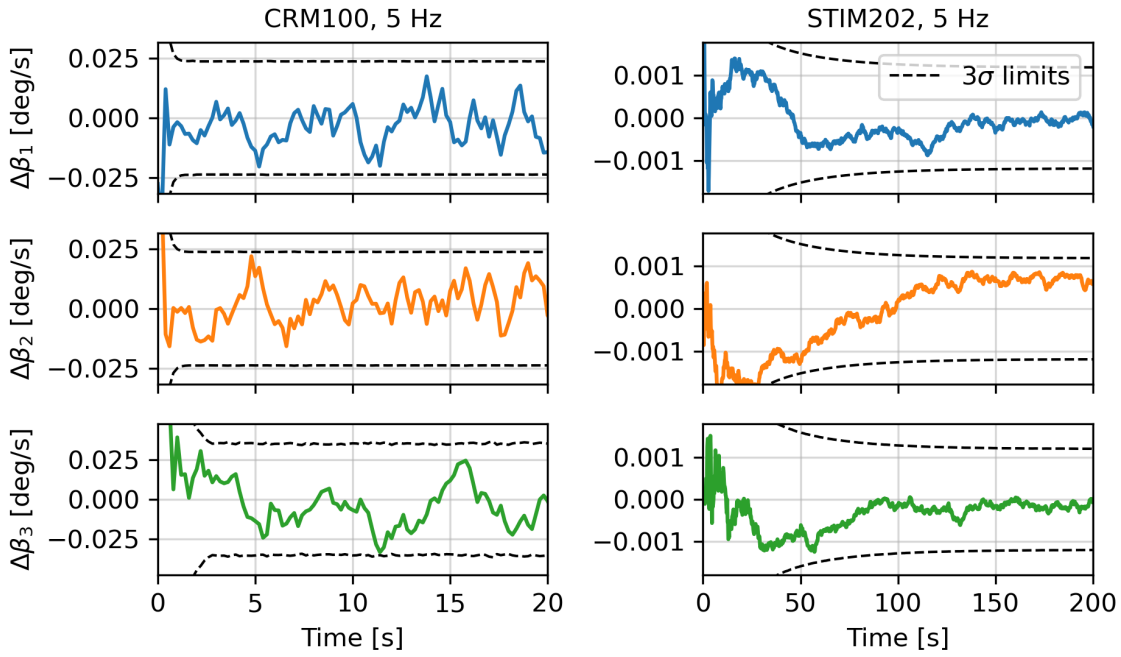


Figure 6.6: Case 1: Bias errors and 3σ bounds.

the estimated bias more closely follows the true bias using the CRM100 gyro, notice the difference in magnitude. In Figure 6.6, the components of bias error ($\Delta\boldsymbol{\beta} = \boldsymbol{\beta} - \hat{\boldsymbol{\beta}}$) along with its 3σ bounds are shown in order to better assess filter performance. Here, the time

window has been adapted to illustrate the relevant characteristics of each configuration. With the CRM100, convergence takes a couple of seconds and a precision of about 0.025 deg/s (3σ) in each component is achieved. This represents a significant improvement over non-calibrated gyro measurements, which may reach errors in the order of degrees per second. On the other hand, the results obtained using the STIM202 gyro show a much larger convergence time, possibly due to the increased confidence in gyro measurements. The final precision in angular velocity estimation is of around 0.001 deg/s (3σ) in each component. In contrast to attitude errors, bias errors are fairly isotropic. With a sampling rate of 1 Hz the results are very similar, and therefore will be omitted.

6.2.2. Case 2: Slew Maneuver

The main goal of this analysis is to test the constant angular velocity assumptions made in the filters' formulation, and assess whether nominal accuracy is maintained during a slew maneuver. In accordance to the previous results, only star tracker and gyro measurements will be considered, and the four filters are analyzed with sampling rates of 1 Hz and 5 Hz. Additionally, a case where the gyro sampling rate is increased to 160 Hz is studied. Results for the RMSE (expressed in arcseconds) are presented in Table 6.2.

Table 6.2: Case 2: RMSE [arcsec] for the different configurations.

Gyro Model	Sampling Rate	MEKF	IMEKF	MUKF	USQUE
CRM100	Both @ 1 Hz	53.57	54.42	53.21	53.21
	Both @ 5 Hz	33.32	33.84	33.34	33.34
	STR @ 5 Hz, gyro @ 160 Hz	36.45	37.23	36.45	36.45
STIM202	Both @ 1 Hz	253.74	187.36	253.73	253.73
	Both @ 5 Hz	29.30	24.77	29.30	29.30
	STR @ 5 Hz, gyro @ 160 Hz	20.15	20.72	20.15	20.15

Upon first glance, these results are quite surprising. On one hand, those obtained using the CRM100 gyro do not differ much from the nominal values in Table 6.1. All filters offer almost identical performance, and a higher star tracker sampling rate improves attitude precision. In this case, the filter is heavily relying on star tracker data to generate the estimate. Notice, however, that increasing the gyro sampling rate to 160 Hz degrades filter performance to some degree. At first this may seem counter-intuitive, since a higher number of angular velocity samples should result in a more precise time propagation. Nonetheless, as argued in the previous section, it can be justified by the consequent increase in gyro measurement noise.

The results obtained using the STIM202, on the other hand, present a significant degradation, especially at 1 Hz. This is a clear case of improper filter tuning; the process noise, which is entirely based on the mathematical gyro model, does not adequately capture the imprecision in attitude and covariance propagation caused by the time-varying angular velocity, therefore causing the filter to diverge. When using the CRM100 gyro this is not as problematic; in general, a larger process noise will increase the robustness of the filter against possible modeling inaccuracies. Fortunately, by increasing the gyro sampling rate to 160 Hz, the propagation accuracy is recovered to some extent. As can be seen, the values in the last row of Table 6.2 more closely match those obtained for the nominal case.

The main conclusion that can be extracted from these results is that special care must be taken in scenarios with time varying angular velocity. Nevertheless, a comparable performance to the nominal case may be achieved with both gyros.

6.2.3. Case 3: Lost In Space

a) Using one attitude sensor only

First, the most critical scenario is addressed, where attitude knowledge must be recovered using a gyro and measurements from either the Sun sensor or TAM (not both at the same time). Additionally, the spacecraft is assumed to be Earth-pointing and therefore rotating with very slow angular velocity. At first, it might seem impossible to have three-axis attitude determination using just one measured direction; indeed, it cannot be done using static methods. Intuitively, however, one measurement could be propagated forward in time until a second measurement is available. Then, both could be combined to produce an full attitude estimate. This interpretation closely describes what a filtering algorithm may be able to achieve in the right conditions. For this to work, the time propagation shall be accurate enough, and the successive measurements need to be independent from each other (if the sensor measures the same direction, no new information is being added).

After some experimentation, it has been determined that the only viable option is to use the higher grade STIM202 gyro along with TAM measurements (otherwise no filter manages to converge). On one hand, time propagation with the CRM100 is not precise enough. On the other hand, the Sun vector remains virtually fixed in the inertial reference frame, which together with the slowly rotating spacecraft provides very weak observability (two successive measurements are almost identical). Additionally, the experiments have concluded that a sampling rate of 1 Hz (for both the gyro and TAM) is sufficient. Since

the magnetic field direction changes slowly, further increasing the sampling rate does not provide any noticeable improvement in convergence speed, and only adds computational cost.

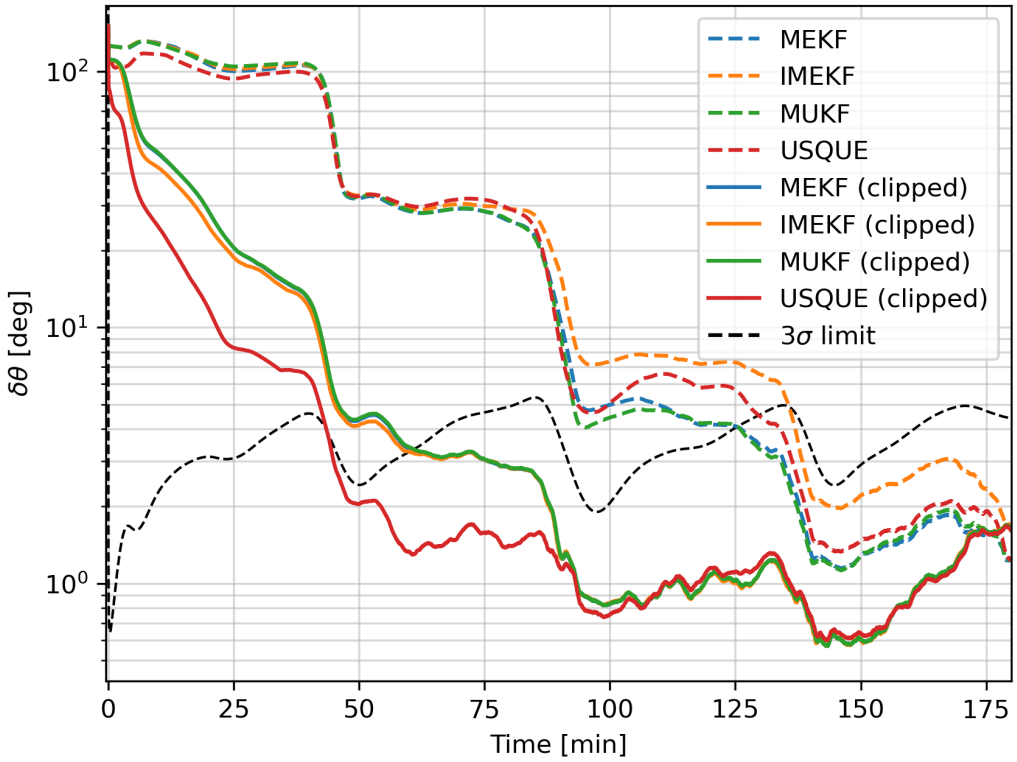


Figure 6.7: Case 3: Average error Euler angle and 3σ bound using TAM measurements.

In order to assess the effect of the bias clipping technique, two different versions of each filter are considered: one implementing bias clipping and other with no modifications. In Figure 6.7, the average error Euler angle evolution is shown (over 20 simulations). Each filter produces very similar 3σ bounds, and that of the MEKF has also been included to add some perspective. The first thing to note is that all filters converge to an error of a few degrees in less than two orbits, regardless of the algorithm used. While the MEKF has been known to present convergence problems, this is typically due to the reduced observability that may arise in orbits at a higher altitude and/or with less inclination, which cause the magnetic field to vary more slowly. This is patently not the case for polar orbits.

Furthermore, it is clear that bias clipping has a significant effect on the convergence speed, where the "clipped" filters take less than half of the time to converge. While the difference in filtering performance is less pronounced between the various algorithms, some observations can be made. First, the results show that the MEKF and MUKF offer

almost identical performance, indicating that the scaled unscented transform (using the recommended parameters) closely resembles a plain linearization. The same applies to the IMEKF, where the additional iteration does not seem to help much. The USQUE, on the other hand, appears to produce noticeably different results; together with bias clipping, it offers the best performance. The convergence times of the various filters, computed as the times of crossing the respective 3σ bound, are indicated in Table 6.3.

Table 6.3: Case 3: Convergence time (in minutes) of each filter.

	MEKF	IMEKF	MUKF	USQUE
With no bias clipping	125.3	135.7	125.3	129.6
With bias clipping	60.7	60.1	43.6	60.9

Bear in mind that Figure 6.7 presents averaged results, and a single simulation may reveal very different behavior. In Figure 6.8, two unique realizations are shown. They have been initialized using different attitude and bias errors, and both implement the clipping technique. While the MEKF, IMEKF and MUKF generally produce consistent outputs (not much changes from simulation to simulation), the USQUE is less stable. Sometimes it may naturally converge very rapidly, but other times it performs worse than the other algorithms. Nevertheless, as shown in Figure 6.7, on average it provides faster convergence.

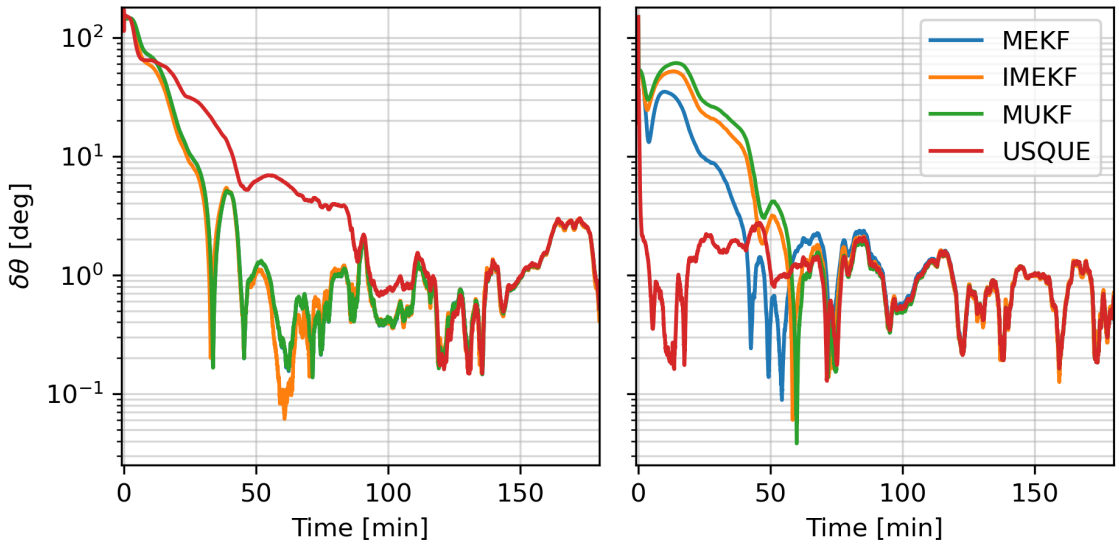


Figure 6.8: Case 3: Filter performance in two different realizations.

In conclusion, the results show that the bias clipping strategy generates the most significant performance improvements. An illustrative example to see the clipping effect

in action is presented in Figure 6.9. Moreover, while the USQUE converges 30% faster than the other alternatives, its computational cost is about eight times higher than that of the MEKF. Since convergence speed is typically not a major concern (as long as the filter converges), the MEKF proves to be the overall better option.

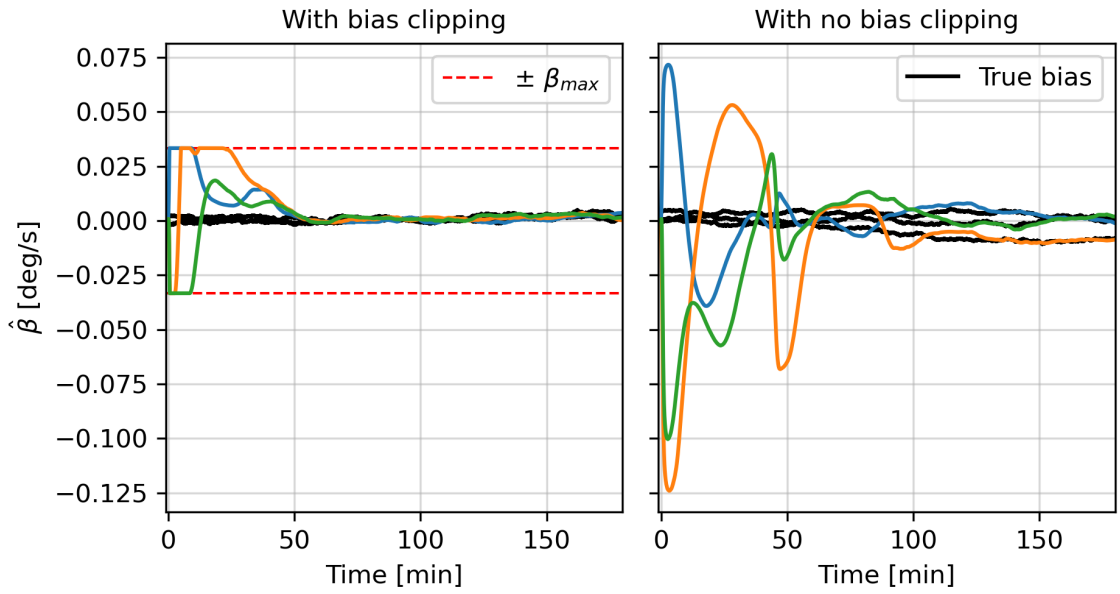


Figure 6.9: Case 3: Effect of clipping on the bias estimate.

b) Using Sun sensor and TAM measurements

Lastly, attitude knowledge recovery using Sun sensor and TAM measurements is studied, simulating regular eclipse cycles. The sampling rate of all sensors is set to 1 Hz (for the previously mentioned reasons), and the spacecraft will be spinning on an axis perpendicular to the direction of the Sun (the eclipse covers approximately half of the orbit). In Figure 6.10, the error Euler angle obtained using both gyros is shown. In this case, all filtering algorithms produce very similar results (including the 3σ bounds), and therefore only those obtained through the MEKF are presented. Since two independent directions are available, convergence is much faster than in the previous case, and bias clipping is generally not required. Using the CRM100, a RMSE of about 0.16° can be achieved during the illuminated portion of the orbit. With the STIM202, it drops to 0.08°. Again, these results agree with the computed 3σ bounds. Moreover, entering the shadow region results in complete loss of attitude knowledge when using the CRM100 gyro, while some precision can be maintained with the STIM202.

Overall, the results show that attitude knowledge may be recovered with a multitude of sensor configurations and filtering algorithms. While convergence takes just a few

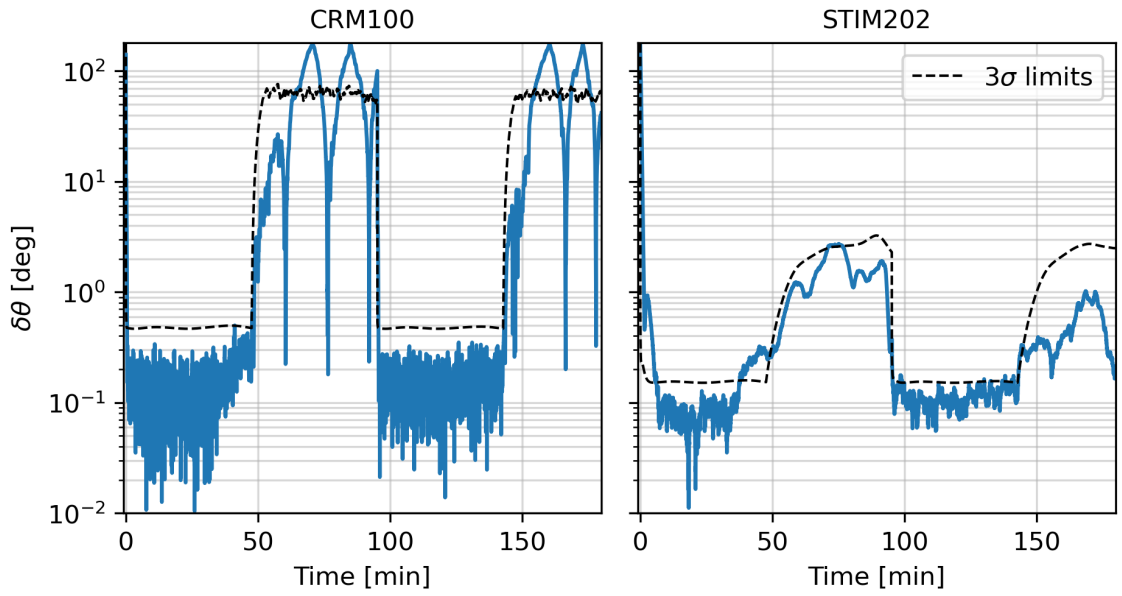


Figure 6.10: Case 3: Average error Euler angle and 3σ bound using TAM and Sun sensor measurements.

minutes when using two independent directions or more, it slows down considerably if just TAM measurements are available, additionally requiring the better STIM202 gyro. The precision obtained using Sun sensor and TAM measurements may be sufficient for some missions, specially when eclipses are not common (e.g. for a dawn-dusk Sun synchronous orbit). Therefore, the ADS module could be tailored to include a star tracker or not depending on the specific mission requirements.

Note that in the open literature, for instance in [13], the various filters are shown to converge quite slower than what the current results indicate. However, this is strongly influenced by the precision of the gyroscope in relation to that of the attitude sensors. In this work, relatively poor performing gyros have been selected, which cause the filters to converge very rapidly when two or more directions are available.

7 Conclusions and Future Work

In this project, the preliminary design of an attitude determination module for small spacecraft in LEO has been successfully carried out. After analyzing the current state of the art of integrated ADCS/ADS solutions and attitude determination algorithms, an appropriate sensor suite and several variants of the multiplicative extended Kalman filter (MEKF) were selected as the basis of the module

Three scenarios have been proposed to test the design, simulating different conditions that a spacecraft may experience during its lifetime. First, nominal Earth-pointing operation was analyzed, as it is the standard operating mode of a spacecraft's ADS and generally takes up the majority of the mission's lifetime. Secondly, a typical slew maneuver was considered in order to cover other common spacecraft modes. Lastly, a more critical scenario with complete loss of attitude knowledge was studied to test the limits of the design.

Multiple simulations have been performed to find the best configuration in each scenario, evaluating:

- Two MEMS gyroscopes with different technical specifications.
- Four Kalman filters; the classical MEKF, an iterated version (IMEKF), and two unscented variants (USQUE and MUKF).
- A novel bias clipping technique.

In addition, various combinations of attitude sensors at different sampling frequencies have been tested.

The results show that the selected sensor suite, integrating Sun sensors and a star tracker from Solar MEMS Technologies' catalog, provides high pointing accuracy in any of the analyzed scenarios. In nominal operation, only a star tracker shall be used in conjunction with gyro data, as the noisier Sun sensor and magnetometer measurements may degrade filter performance. Moreover, a gyro sampling rate matching that of the star tracker is recommended. All filters provide almost identical performance, and therefore the MEKF is preferred due to its reduced computational cost. While a lower grade gyroscope

may be enough for most applications, producing maximum pointing and angular velocity errors of approximately 25/100 arcsec (across/along the star tracker boresight) and 0.025 deg/s respectively (3σ limits), these can be reduced to 20/60 arcsec and 0.001 deg/s using a higher grade alternative.

In the context of slew maneuvers, a performance comparable to that in the nominal case may be achieved. However, special care must be taken when using the higher grade gyro, as the time-varying angular velocity can result in integration errors which are not adequately captured by the established process noise (that is, the filters become badly tuned). The simulations show that increasing the gyro sampling rate (thus producing a better approximation of the real angular velocity profile) solves this issue. Alternatively, artificial process noise may be added to the model.

Lastly, the results for the lost-in-space scenario indicate that attitude knowledge can be recovered in polar orbits using a higher grade gyro and magnetometer measurements only. Note that in orbits with lower inclination this may not always be possible, although these are much less common for small spacecraft. Additionally, the bias clipping technique proves to be a very effective tool for filter initialization, halving the time of convergence. Conversely, the difference in filtering performance is less pronounced between the various algorithms. Although the USQUE may offer slightly better average performance, the MEKF produces more consistent results and requires much less computation, which again make it the preferred option.

Based on the previous results, it is clear that an attitude determination module consisting of several Sun sensors, a three-axis magnetometer, one star tracker, and an affordable three-axis MEMS gyroscope is capable of producing quasi-continuous attitude and angular velocity estimates in various scenarios, with enough accuracy for most small spacecraft missions in LEO. The use of a higher grade gyro may be justified in missions with stringent pointing requirements, owing to a significant increase in attitude accuracy and more flexibility in terms of attitude recovery. For less demanding applications, however, a lower grade gyro will generally provide adequate performance. Furthermore, the use of a star tracker may be optional if a pointing accuracy of no more than 1° is required, specially in polar orbits where eclipses are not common (e.g. for a dawn-dusk Sun synchronous orbit). Otherwise, a higher grade gyro is generally needed to maintain attitude knowledge during an eclipse.

Regarding the filter selection, the MEKF has proven to be the best option for most applications, given its much lower computational cost compared to the unscented filters, and similar performance in all of the analyzed scenarios. Only under very specific con-

ditions, where the filter must be initialized with poor observability, may the USQUE present an advantage over the MEKF. Therefore, careful consideration of the mission requirements and computational capacity budget is needed in order to assess whether implementing the USQUE is necessary and/or feasible.

Finally, it is opportune to recall that only a preliminary design of the attitude determination module has been carried out, and further experimentation will be required at a more advanced stage. For instance, although the most common orbit for CubeSats has been analyzed, other orbits with lower inclination shall also be studied (e.g. to check if attitude determination is still possible using magnetometer measurements only). Additionally, more realistic conditions may be simulated, including attitude dynamics with perturbation torques and non-perfect control. The integration with an appropriate control system also needs to be assessed, as there may be some unwanted interaction between control and estimation that ought to be avoided. Lastly, a *gyroless* configuration could also be considered; although it requires a slightly more complex formulation, it may produce comparable results to those using a lower grade gyro.

Bibliography

[1] E. Kulu, "Nanosats database," 2022. [Online] *https://www.nanosats.eu*. Accessed: September 28, 2022.

[2] Bryce Space and Technology, "Smallsats by the numbers 2022," 2022. [Online] *https://brycetech.com/reports/report-documents/Bryce_Smallsats_2022.pdf*. Accessed: September 28, 2022.

[3] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, New York, 2014.

[4] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, October 2000.

[5] NASA, "What are smallsats and cubesats?," 2017. [Online] *https://www.nasa.gov/content/what-are-smallsats-and-cubesats*. Accessed: September 28, 2022.

[6] E. Kulu, "Nanosatellite launch forecasts 2022: Track record and latest prediction," 36th Annual Small Satellite Conference, August 2022.

[7] M. Hader, "How europe can develop and secure a small-satellite launcher solution," 2021. [Online] *https://www.rolandberger.com/en/Insights/Publications/Supporting-Europe%27s-microlauncher-lift-off.html*. Accessed: September 29, 2022.

[8] European Commision, "Mini satellites make europe more competitive in the new space race," 2019. [Online] *https://cordis.europa.eu/article/id/360991-mini-satellites-make-europe-more-competitive-in-the-new-space-race*. Accessed: September 28, 2022.

[9] Satsearch B.V., 2022. [Online] *https://satsearch.co*. Accessed: October 5, 2022.

[10] CubeSatShop, 2022. [Online] *https://www.cubesatshop.com*. Accessed: October 5, 2022.

[11] NASA, "State-of-the-art of small spacecraft technology," tech. rep., Ames Research Center, Moffett Field, California, October 2021.

[12] M. D. Shuster, "A simple Kalman filter and smoother for spacecraft attitude," *The Journal of the Astronautical Sciences*, vol. 37, no. 1, pp. 89-106, January 1989.

[13] L. Chang, B. Hu, and K. Li, "Iterated multiplicative extended Kalman filter for attitude estimation using vector observations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4 pp. 2053-2060, August 2016.

[14] F. L. Markley and J. L. Crassidis, "Unscented filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 536-542, July-August 2003.

[15] R. Zanetti, K. J. DeMars, and D. Mortari, "Novel multiplicative unscented Kalman filter for attitude estimation," *AAS/AIAA Space Flight Mechanics Meeting - Advances in the Astronautical Sciences Series*, vol. 143, pp. 337-348, January 2012.

[16] J. L. Crassidis, F. L. Markley, and Y. Chen, "A survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12-28, 2007.

[17] J. R. Wertz, *Spacecraft Attitude Determination and Control.* Kluwer Academic, The Netherlands, 1978.

[18] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM Review*, vol. 6, no. 4, pp. 422–430, October 1964.

[19] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems.* Chapman and Hall/CRC, New York, 2004.

[20] A. B. Younes, D. Mortari, J. D. Turner, and J. L. Junkins, "Attitude error kinematics," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 330-335, January-February 2014.

[21] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors.* McGraw-Hill, New York, 2008.

[22] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," *Proceedings of the American Control Conference*, p. 1628–1632, June 1995.

[23] S. J. Julier, "The scaled unscented transformation," *Proceedings of the American Control Conference, pp. 1108–1114*, May 2002.

[24] J. K. Uhlmann, *Dynamic Map Building and Localization: New Theoretical Foundations*. Ph.D. thesis, University of Oxford, 1995.

[25] F. Gustafsson and G. Hendeby, "Some relations between extended and unscented Kalman filters," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545-555, February 2012.

[26] D. E. Winch, D. J. Ivers, J. P. R. Turner, and R. J. Stening, "Geomagnetism and schmidt quasi-normalization," *Geophysical Journal International*, vol. 160, no. 2, pp. 487–504, 2005.

[27] M. Wiegand, "Autonomous satellite navigation via Kalman filtering of magnetometer data," *Acta Astronautica*, vol. 38. no. 4-8, pp. 395-403, 1996.

[28] D. Cilden-Guler, Z. Kaymaz, and C. Hajiyev, "Evaluation of geomagnetic field models using magnetometer measurements for satellite attitude determination system at low earth orbits: Case studies," *Advances in Space Research*, vol. 61, no. 1, pp. 513-529, 2018.

[29] T. E. Humphreys, "Attitude determination for small satellites with modest pointing constraints," 2002.

[30] N. A. Matteo and Y. T. Morton, "Ionosphere geomagnetic field: Comparison of IGRF model prediction and satellite measurements 1991–2010," *Radio Science*, vol. 46, no. 4, pp. 1-10, 2011.

[31] F. J. Lowes, "IGRF health warning, errors, and limitations," 2022. [Online] *https://www.ncei.noaa.gov/products/international-geomagnetic-reference-field/health-warning*. Accessed: March 2, 2023.

[32] J. A. Farrell, F. O. Silva, F. Rahman, and J. Wendel, "IMU error modeling tutorial: INS state estimation with real-time sensor calibration," *IEEE Control Systems Magazine*, vol. 42, no. 6, pp 40-66, December 2022.

[33] M. D. Shuster, "Kalman filtering of spacecraft attitude and the quest model," *The Journal of the Astronautical Sciences*, vol. 38, no. 3, pp. 377-393, 1990.

[34] L. Chang, "Multiplicative extended Kalman filter ignoring initial conditions for attitude estimation using vector observations," *The Journal of Navigation*, vol. 1-15, doi: 10.1017/S0373463322000443, 2023.

[35] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering*

and Neural Networks (S. Haykin, ed.), ch. 7, pp. 221–280, New Jersey: John Wiley & Sons, 2001.

[36] S. Kolås, B. Foss, and T. Schei, "Constrained nonlinear state estimation based on the UKF approach," *Computers & Chemical Engineering*, vol. 33, no.8, pp. 1386-1401, 2009.

[37] E. L. Hasseltine and J. B. Rawlings, "A critical evaluation of extended Kalman filtering and moving horizon estimation," *TWMCC*, Tech. Rep. 2002-03, 2003.

[38] R. Aucoin, "Real-time optimal slew maneuver planning for small satellites," 33th Annual Small Satellite Conference, August 2019.

[39] ESA, "ESA pointing error engineering handbook," *ESSB-HB-E-003 Working Group*, 2011.