



**Politecnico
di Torino**

Politecnico di Torino

Master Degree in Computer Engineering

A.a. 2024/2025

March/April Graduation Session

**Leveraging light-curve inversion for
real-time kinematic state estimation
of uncooperative targets**

Supervisors:

Dr. Andrew Lawrence Price

Dr. Mathieu Salzmann

Prof. Manuela Battipede

Candidate:

Francesco Renis

Acknowledgements

I would like to express my gratitude to Dr. Andrew Price for proposing this project and to EPFL CVLab for funding my participation in the ESA 9th European Conference on Space Debris, held in Bonn, Germany. A scientific paper was published as part of the conference proceedings.

I would like to thank Luís Gonçalves for his invaluable collaboration in sharing his Light-curve Simulator under development; I wish him success in completing his PhD. Additionally, I want to acknowledge the support of the industry partners ClearSpace, for providing access to their high-fidelity data, and PAVE Space, for their valuable feedback on the practical relevance of this work.

Merci beaucoup to the EPFL Space Center for hosting me during my Master Thesis project in Lausanne.

My deepest thanks go to my family for giving me the opportunity to study and live abroad, and most of all, for their unconditional support.

Grazie.

Abstract

The growing number of space objects threatens mission sustainability, making precise real-time tracking essential for Active Debris Removal (ADR) and In-Orbit Servicing (IOS) missions. For an uncooperative target, a Vision Based Navigation (VBN) relative pose (attitude and position) estimation system coupled with a state estimator are likely required. Missions can further be supported by commissioning unresolved observations of the target to produce light curves, which can then be used to extract rotation rates and axes of rotation. This work performs the novel task of exploiting the light curves as kinematic priors to enhance the performance of the state estimator.

An Extended Kalman Filter (EKF) and an Unscented Kalman Filter (UKF) are implemented. Light curve motion prior extraction and VBN pose estimation are simulated based on state-of-the-art results. Three independent studies are performed exploiting the motion priors: (1) Optimizing the Kalman filter tuning for specific kinematic scenarios; (2) Injecting the priors as an initial condition to improve convergence and steady state errors; and (3) Enhancing an outlier rejection function with supplementary proxy measurements from the priors. Performance is evaluated on a custom synthetic light curve dataset based on the Atlas Centaur rocket body, and a private commercial dataset based on the Vega Secondary Payload Adapter from commercial collaborator, ClearSpace.

By exploiting kinematic priors, convergence time and steady state error reductions of $3\times$ or more are exhibited for certain state components, dependent on the kinematic scenario and filter tuning. In general, several trade-offs are observed with kinematic priors providing the opportunity for the lowest steady state errors. This method has the potential to improve the pose estimation accuracy for proximity operations of uncooperative tumbling objects, supporting ADR and IOS missions, especially considering the mild assumptions required.

Contents

List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 The Space Debris problem	2
1.2 Current solutions background	3
1.3 Objectives and structure of this work	5
2 Theoretical background	7
2.1 Spacecraft dynamics	7
2.1.1 Attitude dynamics	8
2.1.2 Orbital dynamics	15
2.1.3 Orbit geometry and orbital elements	19
2.1.4 Relative motion	24
2.2 Kalman filtering	29
2.2.1 Linear Kalman Filter	32
2.2.2 Extended Kalman Filter	33
2.2.3 Unscented Kalman Filter	35
2.2.4 Outlier rejection	38

3	Light curves	41
3.1	Definition and analysis techniques	41
3.2	Light-curves Simulator	43
3.3	One-axis tumbling analysis	43
3.3.1	Synthetic data	44
3.3.2	Commercial data	46
3.4	Motion prior extraction assumption	46
4	State estimation model	49
4.1	State modeling	49
4.2	Real-time measurements	54
4.3	Software framework	56
5	Kinematic priors injection studies	57
5.1	Kalman Filter parameter tuning	60
5.2	Inertia matrix estimation improvement	65
5.3	Enhanced outlier rejection	70
6	Conclusions	74
6.1	Future work	75
	References	76

List of Figures

1.1	Evolution of number of objects in all orbits	3
1.2	Block diagram of the framework	6
2.1	Gimbal lock	12
2.2	Two-body problem framework	16
2.3	The ellipse and its nomenclature	20
2.4	Classical orbital elements	22
2.5	Chi-squared distribution and confidence values	40
3.1	Rocket body synthetic light curve	44
3.2	Spectrum of a rocket body synthetic light curve	45
3.3	Spectrum of a VESPA light curve	47
5.1	Convergence metrics visualized	59
5.2	Study 2 - Relative distance components estimation error	68
5.3	Study 2 - Relative attitude and target inertia matrix components estimation error	69
5.4	Study 3 - Relative attitude components estimation error	71
5.5	Study 2 - Relative distance components estimation error	72

List of Tables

2.1	Classification of orbits	23
2.2	Typical missions per orbit type	23
2.3	Two-Line Element (TLE) format, line 1	24
2.4	Two-Line Element (TLE) format, line 2	25
5.1	Tumbling scenarios definition	60
5.2	Optimal parameters for each scenario	62
5.3	Study 1 - UKF filter convergence times	63
5.4	Study 1 - UKF filter convergence RMSEs	63
5.5	Study 1 - EKF state convergence times	63
5.6	Study 1 - EKF state convergence RMSEs	64
5.7	Study 2 - State value convergence times	66
5.8	Study 2 - State value convergence RMSEs	66
5.9	Study 2 - Filter convergence times	67
5.10	Study 2 - Filter convergence RMSEs	67
5.11	Study 3 - State value RMSE during fault window	72

Chapter 1

Introduction

Space exploration has always captured the interest of people worldwide. Collective imagination has been widely influenced by movies, literature, paintings, music, which have always portrayed space as covered by a fascinating veil of mystery. At least once, every human being has dreamed of exploring other planets, finding new forms of life or being able to travel across the galaxy. As many fields, space is indeed marked by fascinating yet scientifically challenging aspects. Since scientific research tends to question reality on increasingly broader orders of magnitude, whatever goes towards extremely small or extremely large scales becomes the frontier of human knowledge.

From the very first steps taken towards reaching space, this domain has always been very appealing for the biggest governments of the world, not only for the high investment pay-off chance, but unfortunately also for applications in the military domain. The first rocket to reach space (more than 100 Km from Earth's surface) was in fact developed as a weapon by Nazi Germany built during World War II. And it is significant that the first satellite to be sent into Earth orbit was launched by the Union of Soviet Socialist Republics (USSR) in 1957 in the context of the Space Race against the USA during the Cold War.

Despite being initially born as side effects of military applications, discoveries in the space domain have led to technologies that are now integrated in our daily lives. Telecommunications, the Internet, GPS navigation, and weather forecasting are only a few examples of such technologies that we now take for granted. As time passed, space research has evolved to mainly focus on purely civil and scientific

scopes. From distant observation of celestial bodies, with Space Telescopes like Hubble and the James Webb, to understand the origin of the universe, to planetary science mission and Earth observation satellites, contributing to weather forecasting and climate monitoring.

Extending far beyond, though not entirely, from its initial military origins, the space research domain has enabled peaceful international collaborations that have produced innovations benefiting all of humanity.

1.1 The Space Debris problem

Since the beginning of the space age, thousands of spacecrafts, rockets and instruments have been sent into Earth orbit. In the early days of spaceflight these systems were exactly planned to fulfill their task, with very little or no thought about what would happen when they reached the end of their lifetime (End Of Life, EOF). Over time, those have accumulated, and, as often happens when technological advancements are strongly pursued, the number of nonfunctional old units (Resident Space Objects) now populating Earth's orbits poses a significant threat to future space missions and to Earth's surface as well. Such objects can in fact collide generating spread out fragments that accelerate and become thus more dangerous than the original bodies individually. For reference, a collision with a 1 cm particle travelling 10 km/s (of which there are about a million in orbit) releases the same energy as a small car crashing at 40 km/h. Consequently, the generated fragments can collide themselves, rising a cascaded exponential effect.

Space debris is defined by European Space Agency (ESA) as all non-functional, man-made objects, including fragments and elements thereof, in Earth orbit or re-entering Earth's atmosphere. As of the July 2024 Annual Space Environment Report, ESA is tracking more than 20 thousand objects in Low Earth Orbit (LEO), orbits closer to Earth's surface, estimating millions more untracked ones [17], Fig. 1.1.

The international community is well aware of the Space Debris problem. Since the very first Inter-Agency Space Debris Mitigation Guidelines defined in 2002, efforts have focused into defining regulations to prevent the situation from worsening. One significant example is represented by ESA's Zero Debris Charter [3], a

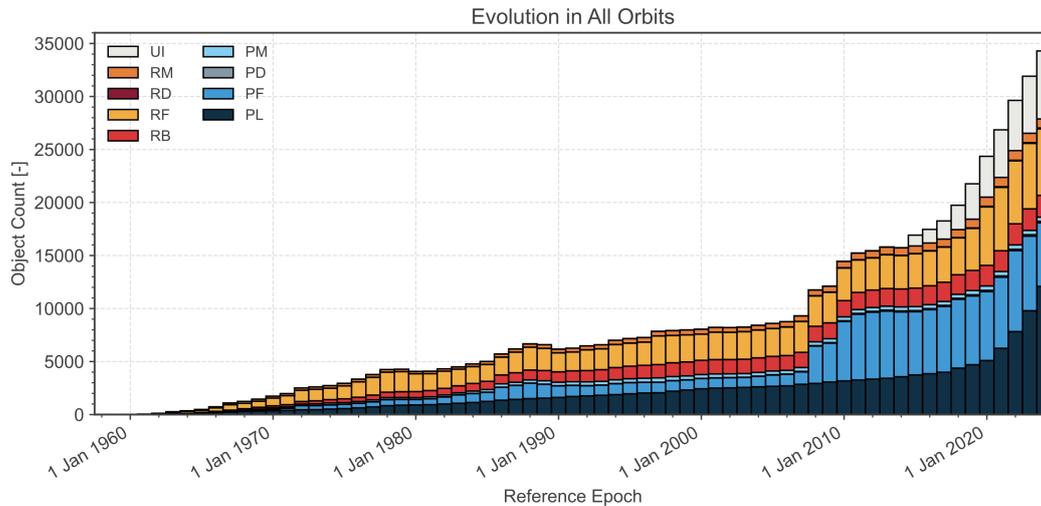


Fig. 1.1 Evolution of number of objects in all orbits. UI is Unidentified, RX are Rocket parts and PX are payloads. For the full legend and additional plots, refer to ESA 2024 Space Environment Report in [17]

global collaborative initiative facilitated by ESA that aims at providing guidelines to significantly reduce the creation of new space debris by year 2030.

1.2 Current solutions background

Examples of debris prevention and mitigation solutions include Life Extension missions, through In-Orbit Servicing (IOS) aimed at repairing, refueling or upgrading satellites, Controlled Reentry operations or Active Debris Removal (ADR) missions, to actively capture and remove Resident Space Objects (RSOs). However, significant engineering challenges remain. Currently, various programs are underway to support and develop the technologies necessary for Active Debris Removal. In 2018, the RemoveDEBRIS mission [8] successfully demonstrated capture of an uncooperative CubeSat, a miniaturized satellite with 10 cm edges used for technological demonstrations or educational purposes. In the second half of 2026, the ClearSpace-1 mission will mark the first large scale ADR operation, capturing the VESPA (VEGA Secondary Payload Adapter), the upper part of a VEGA Rocket, left in LEO after a launch in 2013.

For a non-functioning target satellite or piece of debris, a Servicer satellite sent in a proximity mission towards it will not have access to cooperative relative navigation

techniques such as relative GNSS (Global Navigation Satellite System), Target telemetry or Target beacons emitting signals. For relative attitude determination in such uncooperative circumstances, an optical sensing technique coupled with a state estimator is required.

State estimation, or state observation, refers to the process of providing an estimate of the internal states of a system, as the name suggest.

With the appropriate level of abstraction, any real physical system can be seen as an object that receives an input and generates an output based on its current condition. Think of a light switch: it receives an input, the flip of the button, and depending on its current status, a light bulb will be either turned on or off. The same happens, for example, with a car: for the same pedal push, the resulting change on the speed will depend on the speed the car had before. Or the thermal control of a room: producing the same amount of heat will rise a different change in the temperature depending on the current temperature of the room. Countless examples exist. Intuitively, the state of a system depends thus on his history, on the past inputs it received or on its initial condition before its spontaneous evolution. The history of a car is the engine torque that was applied and made it reach its speed, and the state of a thermal control system is the room temperature value when it gets turned on.

State estimation theory says that one can reconstruct the history of a system, i.e, recover the value of its state, by combining a time series of output values the system produced, measured by sensors, possibly knowing the inputs that were applied. Consequently, a state estimator is a device that knows the mathematical model of a system, and enables this processing. Such kind of computations can be either developed offline, working with a set of data gathered at a given time in the past, or in real-time, processing new measurements coming from available sensor as they are produced. In the context of space debris removal missions the latter is the case.

In real world dynamic systems, uncertainties, namely *noise*, can arise from measurements, from inaccurate mathematical modeling or from unpredictable environmental conditions. The ability to recover the true state of a system even in presence of noise or incomplete data is thus essential. This is specifically referred to as *filtering*. For this reason, as one can easily imagine, real-time state estimation and filtering and fundamentally overlapping concepts.

As mentioned, proximity operations with uncooperative targets, such as the ones occurring during Active Debris Removal or In-Orbit Servicing mission, require the

usage of Vision-Based Navigation (VBN) measurement models. The most basic technique involves providing the Servicer spacecraft with an optical camera that takes images of the approached Target at regular time intervals and feeds them to a Neural Network. This Neural Network is a device that has been programmed (or *trained*) to recognize the distance and the rotation of the body present in a picture, solving the problem called *Pose Estimation*. Such Pose Estimation Neural Network is the device that provides the measurements to be fed to the real-time state estimator. If the rotation, or pose, and the relative distance of the target object are the measured output values, the internal state of the target are its rotational speed, i.e. angular velocity, and its relative translational acceleration. Moreover, since the target is not under control, it doesn't receive any input.

Additionally, in advance of a servicing mission, observations of the client satellite can be made using ground-based and in-orbit Space Situational Awareness (SSA) services. In this context, a powerful element is represented by *light curves*, which are series of brightness intensity values of objects observed in space. The changes of an object's brightness over a given time window are directly related to the rotational motion it exhibits. In particular, if the Target's shape and reflective properties are known, information about its dynamics can be inferred, extracting what will be referred to as the *kinematic (or motion) prior*.

1.3 Objectives and structure of this work

In the context of a general Active Debris Removal or In-Orbit Servicing mission, this work explores the possibility of improving the performance of a real-time *state estimation model*, fed with measurements from a *pose estimation neural network* working on proximity images, by leveraging *kinematic prior* information coming from light-curves analysis.

This thesis is organized as follows. Chapter 2 reviews the theoretical background needed to understand this work. The mathematical description of the dynamics of a spacecraft and the theory of Kalman Filtering, a widely-known state estimation model, is presented. Chapter 3 introduces the concept of light-curves and the tools for their analysis. Chapter 4 outlines the complete state estimation model built, as depicted in the block diagram in Fig. 1.2. It describes how the different building blocks of this work are combined, presenting the model to propagate the Servicer-

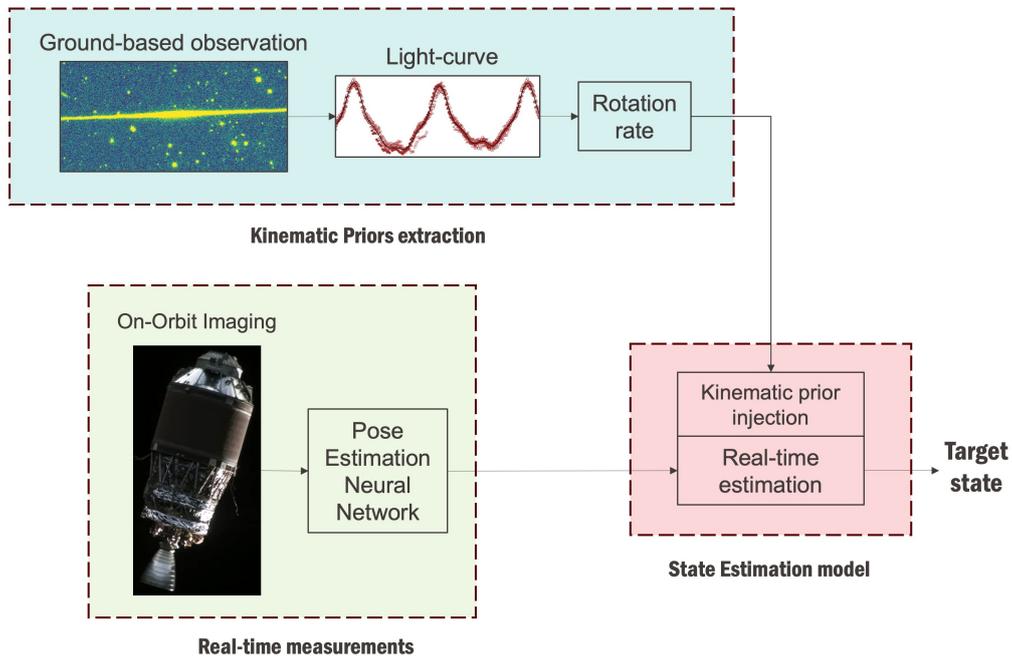


Fig. 1.2 Block diagram of the framework

Target proximity scenario and simulate the measurements as if they were from a pose estimation neural network. Chapter 5 introduces three different kinematic prior injection studies that have been explored and discusses the results obtained. Chapter 6 concludes the discussion and presents directions for future work.

A scientific paper based on this work was published as part of the ESA 9th European Conference on Space Debris proceedings.

Chapter 2

Theoretical background

This chapter presents the theoretical and mathematical background needed for the understanding of this work. The aim is to provide the reader with precise and detailed descriptions, without diving into all the derivations, possibly merging concepts from different sources to present a unified and clearer explanation.

First, a spacecraft's attitude and orbital dynamics equations and relative derivations are presented, from the most general representation to the formulation of the relative motion problem, that models the environment in which this work applies. Next, the Kalman filtering theory is presented, focusing on its extensions for nonlinear systems. A technique for detecting outlier measurements is also presented.

2.1 Spacecraft dynamics

Solving any scientific problem requires first the definition of a mathematical description that models the real physical behavior of the system under inspection. The same physical system can be represented by different mathematical models, each with specific advantages, limitations and working under different assumptions. For instance, a spacecraft is often described as a rigid body moving with respect to a given inertial frame. A rigid body motion is the combination of a translation of its Center of Mass (CoM) with a rotation about an axis passing through the CoM. In this context, a spacecraft position evolves according to orbital dynamics, while its orientation is governed by attitude dynamics.

This chapter provides first the derivation of the two mentioned dynamic models for a single spacecraft, along with associated definitions. The modeling of the relative motion between two spacecrafts is presented afterwards.

2.1.1 Attitude dynamics

The study of the attitude dynamics of a spacecraft refers to the analysis of the rotational motion of a rigid body about an axis passing through its CoM. In particular, the torques applied to a rigid body have the effect of changing its angular speed (dynamic equations), which in turn determines the evolution of the body's orientation (kinematic equations).

Consider a rigid body rotating with respect to an inertial observer reference frame. An inertial reference frame is a fixed frame of reference, not accelerating, with a given origin and axes $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, whose directions are denoted by unit vectors $\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3$. A non-inertial rotating reference frame is fixed to the rotating body, with origin in its CoM, and axes $\mathbf{x}, \mathbf{y}, \mathbf{z}$, with directions $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$. The angular velocity of the body with respect to the inertial frame, represented in its body frame, is thus defined as $\boldsymbol{\omega} = \omega_1 \mathbf{b}_1 + \omega_2 \mathbf{b}_2 + \omega_3 \mathbf{b}_3$. A bold symbol denotes a vector or a matrix.

From classical physics, the angular momentum of a particle i of the body is defined as $\mathbf{H}_i = \mathbf{r}_i \times m_i \dot{\mathbf{R}}_i$, where \mathbf{r}_i is the position of the particle in the body frame, \mathbf{R}_i is the position of the particle in the inertial frame, $\dot{\mathbf{R}}_i$ is its time derivative and

$$\mathbf{q} \times \mathbf{p} = \begin{bmatrix} q_2 p_3 - q_3 p_2 \\ q_3 p_1 - q_1 p_3 \\ q_1 p_2 - q_2 p_1 \end{bmatrix} \quad (2.1)$$

is the cross product between two generic \mathbf{q} and \mathbf{p} vectors. By expressing the angular momentum with respect to the CoM, assuming the body rigid, the resulting total angular momentum is independent of the translational motion, and is defined as

$$\begin{aligned} \mathbf{H} &= \sum_i \mathbf{r}_i \times (\boldsymbol{\omega} \times \mathbf{r}_i) = \\ &= \int_B \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm, \quad m_i \rightarrow \infty \end{aligned} \quad (2.2)$$

The term $\boldsymbol{\omega} \times \mathbf{r}$ accounts for the fact that the body frame is rotating with respect to the inertial frame and comes from the notion of vector derivative. The derivative of \mathbf{r}_i is in fact expressed as $\dot{\mathbf{r}}_i = \dot{x}\mathbf{b}_1 + \dot{y}\mathbf{b}_2 + \dot{z}\mathbf{b}_3 + x\dot{\mathbf{b}}_1 + y\dot{\mathbf{b}}_2 + z\dot{\mathbf{b}}_3$. Considering the rotation $\delta\mathbf{b}_i = \delta\boldsymbol{\theta} \times \mathbf{b}_i$, where $\delta\boldsymbol{\theta} = \boldsymbol{\omega}\delta t$ and $\delta t \rightarrow \infty$, it yields

$$\dot{\mathbf{b}}_i = \boldsymbol{\omega} \times \mathbf{b}_i \quad (2.3)$$

which results in

$$\dot{\mathbf{r}}_i = \dot{r}_B + \boldsymbol{\omega} \times \mathbf{r} \quad (2.4)$$

Developing Eq. (2.2) and compacting the result in matrix notation, one can obtain

$$\mathbf{H} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \mathbf{J}\boldsymbol{\omega} \quad (2.5)$$

where

$$\begin{aligned} J_{11} &= \int_B (y^2 + z^2) dm \\ J_{22} &= \int_B (x^2 + z^2) dm \\ J_{33} &= \int_B (x^2 + y^2) dm \end{aligned} \quad (2.6)$$

are called moments of inertia, and

$$\begin{aligned} J_{12} = J_{21} &= - \int_B xy dm \\ J_{13} = J_{31} &= - \int_B xz dm \\ J_{23} = J_{32} &= - \int_B yz dm \end{aligned} \quad (2.7)$$

are called products of inertia.

Matrix \mathbf{J} is in fact called inertia matrix of the object. It embeds knowledge on the mass distribution of the object, and qualitatively tells how hard is to rotate an object around a given axis. The further the mass is distributed with respect to the considered axis, the higher the inertia value. The inertia matrix is always symmetric, but in general not diagonal. A reference frame for which \mathbf{J} is diagonal can always be found by means of a rotation, but that is not guaranteed to be the most useful in practical applications. For objects with uneven and irregular mass distributions, the

principal axes of inertia (axes for which \mathbf{J} is diagonal) are very likely not aligned with the more intuitive and simpler body reference frame. In particular, in most practical cases dealing with orbital debris, this happens to be the case. However, if little knowledge on the mass distribution is available, assuming a diagonal inertia matrix might still work in most applications.

Suppose now that a moment $\mathbf{M} = M_1\mathbf{b}_1 + M_2\mathbf{b}_2 + M_3\mathbf{b}_3$ is acting on the body. The II law of dynamics for rotational motion says

$$\dot{\mathbf{H}} = \mathbf{M} \quad (2.8)$$

Since $\dot{\mathbf{H}} = \dot{\mathbf{H}}_B + \boldsymbol{\omega} \times \mathbf{H}$, with $\mathbf{H} = \mathbf{J}\boldsymbol{\omega}$ and $\dot{\mathbf{H}} = \mathbf{J}\dot{\boldsymbol{\omega}}$, then

$$J\boldsymbol{\omega} + \boldsymbol{\omega} \times J\boldsymbol{\omega} = \mathbf{M} \quad (2.9)$$

is called Euler moment equation. It is a nonlinear equation and in general it has no analytical solution.

Assuming to consider a body reference frame for which $\mathbf{J} = \text{diag}(J_1, J_2, J_3)$ is diagonal, the Eq. (2.9) can be rewritten as

$$\begin{aligned} J_1 \dot{\omega}_1 + \omega_2 \omega_3 (J_3 - J_2) &= M_1 \\ J_2 \dot{\omega}_2 + \omega_1 \omega_3 (J_1 - J_3) &= M_2 \\ J_3 \dot{\omega}_3 + \omega_1 \omega_2 (J_2 - J_1) &= M_3 \end{aligned} \quad (2.10)$$

which in matrix form becomes

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{(J_2 - J_3)}{J_1} \omega_3 & 0 \\ \frac{(J_3 - J_1)}{J_2} \omega_3 & 0 & 0 \\ \frac{(J_1 - J_2)}{J_3} \omega_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} M_1/J_1 \\ M_2/J_2 \\ M_3/J_3 \end{bmatrix} \quad (2.11)$$

Fixing $\mathbf{M} = 0$ one can inspect the free response (free motion) of the body. In this case, in specific scenarios with additional assumptions (symmetric body, constant angular speed components), analytical solutions can be found, and are very useful to inspect the stability properties of the differential equations. For detailed derivations and perturbation analyses, the reader might refer to the book in [21].

Without diving into the analytical description, results that link the shape of the inertia matrix to the expected dynamics of $\boldsymbol{\omega}$ are derived. In particular, for a

generic asymmetric body, when in free motion, including an energy dissipation term accounting for a semi-rigid body assumption (flexible elements, possible fluid slosh), the following holds:

- ω is a bounded harmonic signal (composition of sinusoidal waves that keeps oscillating without damping to zero)
- The motion about the major principal axis is asymptotically stable
- The motion about the minor and intermediate principal axes is unstable

The description of the evolution of the orientation of the body, namely attitude if talking about a spacecraft, can be now introduced.

The orientation of a body in a three-dimensional space depends on 3 degrees of freedom, which can be generally represented as the composition 3 consequent rotations with respect to the three axes of a reference frame. In particular, the elementary rotation matrices can be defined as:

$$\mathbf{T}_1(\phi) \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.12)$$

$$\mathbf{T}_2(\theta) \doteq \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.13)$$

$$\mathbf{T}_3(\psi) \doteq \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

describing a rotation about, respectively, the \mathbf{X} , \mathbf{Y} , \mathbf{Z} axes of angles ϕ , θ , ψ , called Euler angles. Any rotation can be expressed by means of a product between the elementary rotation matrices, with a specific order. Each matrix multiplication rotates the axes of the body, a following rotation is defined with respect to the rotated object, which thus makes the rotation matrix product non-commutative.

Take as example the Tait-Bryan 123 rotation, where 123 is the rotation order and Tait-Bryan is a name to group the rotation compositions that do not repeat indices (123, 321 with respect to 313, 323), of any angles ϕ and ψ , with $\theta = \pi/2$. The

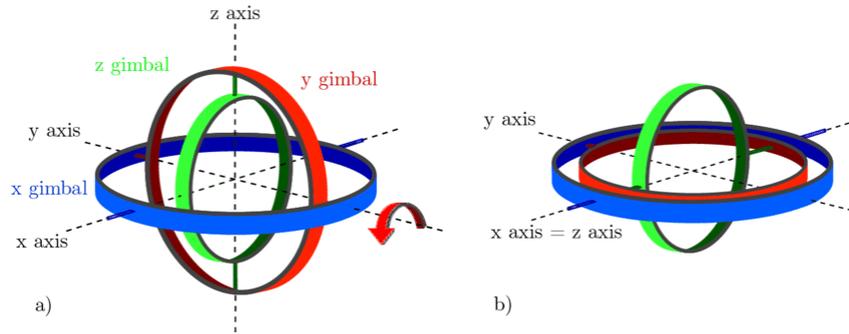


Fig. 2.1 Gimbal lock

composite rotation matrix is

$$\begin{aligned}
 T_{123} \left(\phi, \frac{\pi}{2}, \psi \right) &= T_1(\phi) \cdot T_2(\pi/2) \cdot T_3(\psi) \\
 &= \begin{bmatrix} 0 & 0 & 1 \\ \cos \phi \sin \psi + \cos \psi \sin \phi & \cos \phi \cos \psi - \sin \psi \sin \phi & 0 \\ \sin \psi \sin \phi - \cos \phi \cos \psi & \cos \phi \sin \psi + \cos \psi \sin \phi & 0 \end{bmatrix} \quad (2.15) \\
 &= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\phi + \psi) & \cos(\phi + \psi) & 0 \\ -\cos(\phi + \psi) & \sin(\phi + \psi) & 0 \end{bmatrix}
 \end{aligned}$$

What results is that only the sum of ϕ and ψ angles can be determined from the obtained rotation matrix: this phenomenon is called gimbal lock. Its consequence is a loss of a degree of freedom, which is also named singularity. It originates when one applies a rotation that makes one of the three axes align with one of the others, restraining the rotation space to be two-dimensional, as seen in Fig. 2.1.

Different standards for rotation representations exist (Roll Pitch Yaw (RPY) angles, for example). Among all, particularly interesting and useful are the ones that do not suffer from singularities. The best example in this sense is the quaternion, one of the most used representations.

A quaternion is generally represented in the form

$$\begin{aligned}
 \mathbf{q} &= q_0 + \mathbf{q} \\
 &= q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} \\
 &= \left(\cos \frac{\beta}{2}, u_1 \sin \frac{\beta}{2}, u_2 \sin \frac{\beta}{2}, u_3 \sin \frac{\beta}{2} \right) \\
 &= (q_0, q_1, q_2, q_3) \\
 &= (q_0, \mathbf{q}) = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix}
 \end{aligned} \tag{2.16}$$

where q_0 is called real part, \mathbf{q} is the vector (or imaginary) part, and $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ is the 4D linear vector space basis, which follows the rule $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. A quaternion with null real part is said pure, and a quaternion with null vector part and unitary real part is called identity. In the quaternions space the following expressions hold:

- Quaternion norm

$$|\mathbf{q}| = \|\mathbf{q}\| = \|\mathbf{q}\|_2 = \sqrt{\sum_{i=0}^3 q_i^2} \tag{2.17}$$

- Quaternion complex conjugate

$$q^* = (q_0, -\mathbf{q}) = \cos \frac{\beta}{2} - \mathbf{u} \sin \frac{\beta}{2} \tag{2.18}$$

- Quaternion product

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 p_0 - \mathbf{q} \cdot \mathbf{p} \\ q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p} \end{bmatrix} \tag{2.19}$$

where

$$\mathbf{q} \cdot \mathbf{p} = q_1 p_1 + q_2 p_2 + q_3 p_3 \tag{2.20}$$

is the dot product, and $\mathbf{q} \times \mathbf{p}$ is the cross product.

Considering a rotation of a vector \mathbf{r} about an axis $\mathbf{u} = (u_1, u_2, u_3)$ of an angle β , the corresponding quaternion can be computed as

$$\mathbf{q} = \left(\cos \frac{\beta}{2}, u_1 \sin \frac{\beta}{2}, u_2 \sin \frac{\beta}{2}, u_3 \sin \frac{\beta}{2} \right) \quad (2.21)$$

and the equivalent rotated vector becomes

$$(0, \mathbf{r}') = \mathbf{q} \otimes (0, \mathbf{r}) \otimes \mathbf{q}^* \quad (2.22)$$

The inverse of the rotation is represented by

$$\mathbf{q}^{-1} = \mathbf{q}^* \quad (2.23)$$

A quaternion representing a true rotation must have a unitary norm. A four dimensional quaternion needs in fact to be traced back to the three-dimensional space, interpreting it as 3 degrees of freedom plus a unitary norm constraint. The reader might find an interesting tool for better quaternion visualization at [2].

Once an appropriate representation is chosen, the complete description of the attitude dynamics is concluded by introducing the kinematic equations, that describe the evolution of the orientation of a body that is rotating at a given angular speed. The quaternion kinematics equation, in particular, can be written in the following different equivalent forms:

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{1}{2} \mathbf{q} \otimes (0, \boldsymbol{\omega}) \\ &= \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \mathbf{q} \\ &= \frac{1}{2} \mathbf{Q}(\mathbf{q}) \boldsymbol{\omega} \end{aligned} \quad (2.24)$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) \doteq \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \quad (2.25)$$

and

$$\mathbf{Q}(\mathbf{q}) \doteq \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \quad (2.26)$$

The complete set of dynamic-kinematic equations is thus obtained. The provided mathematical description allows to represent the trajectory of the attitude of a rigid body, such as a spacecraft, knowing the torque applied to it, if any, its inertia matrix and the initial conditions for the angular velocity vector and the attitude quaternion.

The complete equations can thus be converted in state-space representation, to enable more straightforward usage for numerical integration. The state equations yield

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\mathbf{u} \quad (2.27)$$

where

$$\begin{aligned} \mathbf{x} &= [q_0 \quad q_1 \quad q_2 \quad q_3 \quad \omega_1 \quad \omega_2 \quad \omega_3]^T \\ \mathbf{u} &= [M_1 \quad M_2 \quad M_3]^T \end{aligned} \quad (2.28)$$

And $\mathbf{A}(\mathbf{x})$ and \mathbf{B} matrices components are derived from Eq. (2.11) and Eq. (2.24).

2.1.2 Orbital dynamics

Derived from the most general celestial mechanics equations, which describe the motion of any celestial body under the influence of gravity, the orbital dynamics specifically focuses on the trajectories of spacecrafts.

The first result, for a body in an unperturbed orbit around a planet, was obtained by Johannes Kepler in 1605. His empirical laws of planetary motion state the following:

1. The orbit of a planet is an ellipse with the sun located at one focus
2. The radius vector joining the sun and a planet sweeps out equal areas in equal intervals of time
3. The square of orbital period of a planet is proportional to the cube of semimajor axis

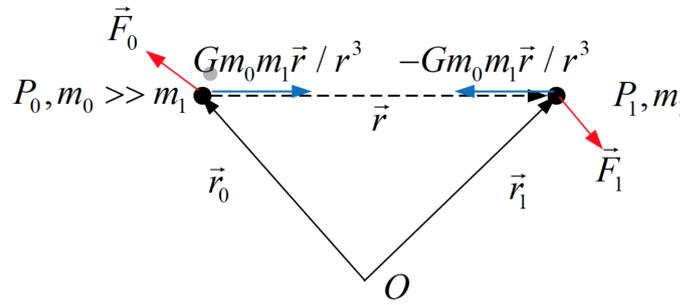


Fig. 2.2 Two-body problem framework

After some time, in 1687 Isaac Newton published his famous book *Philosophiae Naturalis Principia Mathematica*. It built on other recent discoveries of his time, and it is recognized as one of the most important scientific works in history. In the book he describes the fundamental laws of astrodynamics, the laws of motion and the law of universal gravitation.

The latter states that any two particles attract each other with a force

$$\mathbf{F} = G \frac{m_1 m_2}{r^2} = G \frac{m_1 m_2}{r^3} \mathbf{r} \quad (2.29)$$

where m_1 , m_2 are the particles masses, $r = |\mathbf{r}|$ is the vector connecting the two particles and $G = 6.67e^{-11} \frac{Nm^2}{Kg^2}$ is the Universal gravitational constant. Kepler's empirical laws of planetary motion may be derived from Newton's physical laws, under the assumption that the orbiting body is subject only to the gravitational force of the central attractor.

To the purpose of obtaining a representation of the motion of a body in an orbit, a two-body framework is defined using these laws. In classical mechanics, the two-body problem for a gravitational case is therefore to describe and predict the motion of two bodies reciprocally orbiting in space.

Consider two point masses m_0 and m_1 , located at points \mathbf{P}_0 and \mathbf{P}_1 in an inertial frame with origin O , as shown in Fig. 2.2. In this setting, \mathbf{r}_0 and \mathbf{r}_1 are the position vectors of the masses, r is their relative position, v_0 and v_1 are the velocities and F_0 and F_1 are external non-gravitational external forces.

Using Newton's II law and the universal gravitation law, the following holds:

$$\begin{aligned}\dot{\mathbf{v}}_0 &= G \frac{m_1}{r^3} \mathbf{r} + \frac{1}{m_0} \mathbf{F}_0 \\ \dot{\mathbf{v}}_1 &= -G \frac{m_0}{r^3} \mathbf{r} + \frac{1}{m_1} \mathbf{F}_0\end{aligned}\quad (2.30)$$

The expression of the relative velocity $\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_0$ therefore becomes

$$\dot{\mathbf{v}} = -\frac{G(m_0 + m_1)}{r^3} \mathbf{r} + \frac{1}{m_1} \left(\mathbf{F}_1 - \frac{m_1}{m_0} \mathbf{F}_0 \right) \quad (2.31)$$

At the same time, using the transformations

$$\begin{aligned}\mathbf{r}_c &= \frac{m_0}{m_0 + m_1} \mathbf{r}_0 + \frac{m_1}{m_0 + m_1} \mathbf{r}_1 \\ \mathbf{v}_c &= \frac{m_0}{m_0 + m_1} \mathbf{v}_0 + \frac{m_1}{m_0 + m_1} \mathbf{v}_1\end{aligned}\quad (2.32)$$

the expression of the motion of the CoM of the system composed by the two bodies is found:

$$\dot{\mathbf{v}}_c = \frac{1}{m_1} \frac{\mathbf{F}_1 + \mathbf{F}_0}{1 + m_0/m_1} \quad (2.33)$$

The key assumption for the derivation of the orbital dynamics equation follows.

Assuming $m_0 \gg m_1$, meaning that one of the two objects in the relative setting exhibits negligible gravitational attraction on the other, the relative motion equation becomes

$$\dot{\mathbf{v}} + \mu \frac{\mathbf{r}}{r^3} = \frac{1}{m_1} \mathbf{F}_1 \quad (2.34)$$

where $\mu = Gm_0$ is called standard gravitational parameter of a celestial body, equal to $\mu_{Earth} = 3.986 \cdot 10^{14} m^3/s^{-2}$ for the Earth. The expression is called the restricted two-body problem. If additionally no acting force on the smaller body is assumed, i.e. $\mathbf{F}_1 = 0$, the equation yielding

$$\dot{\mathbf{v}} + \mu \frac{\mathbf{r}}{r^3} = 0 \quad (2.35)$$

is called the free restricted two-body equation.

Under the same assumption, the solution of Eq. (2.33) gives $\dot{\mathbf{v}}_c = 0$, showing that the CoM of a restricted two-body problem can be chosen as the origin of any given inertial frame.

For Eq. (2.35) the following holds:

1. Total mechanical energy conservation

$$\dot{\varepsilon} = \frac{d}{dt} \left(\frac{v^2}{2} - \frac{\mu}{r} \right) = 0, \quad \varepsilon = \text{const} \quad (2.36)$$

where ε is the total mechanical energy per unit of mass. This allows to retrieve, for a given constant total energy ε , the corresponding orbital velocity of the object, computed as $v = \sqrt{2\mu/r + 2\varepsilon}$.

2. Angular momentum conservation

$$\dot{\mathbf{h}} = \frac{d}{dt} (\mathbf{r} \times \mathbf{v}) = 0, \quad \mathbf{h} = \text{const} \quad (2.37)$$

where \mathbf{h} is the specific angular momentum vector per unit of mass. It is, by definition, perpendicular to the plane of \mathbf{r} and \mathbf{v} . With \mathbf{h} constant, \mathbf{r} and \mathbf{v} will always remain in the same plane, called orbital plane, which contains the two masses.

Taking the cross product of Eq. (2.35) with h , through simple operations, one can obtain

$$\mathbf{v} \times \mathbf{h} - \frac{\mu}{r} \mathbf{r} = \text{const} \doteq \mu \mathbf{e} \quad (2.38)$$

where \mathbf{e} is called eccentricity vector and $e = |\mathbf{e}|$ is the scalar eccentricity, and

$$r = \frac{p}{1 + e \cos \theta} \quad (2.39)$$

where $p = h^2/\mu$ is called semilatus rectum and θ is the angle between \mathbf{r} and \mathbf{e} , called true anomaly. A detailed discussion on Eq. (2.39) and its parameters will follow.

Taking a step back to the scope of this section, employing the differential equation in Eq. (2.35) one can fully derive the state equations of a spacecraft in orbit:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\mu \frac{\mathbf{r}}{r^3} \\ \dot{x} &= [\dot{\mathbf{r}}, \dot{\mathbf{v}}] \end{aligned} \quad (2.40)$$

In fact, integrating Eq. (2.35) with any given initial conditions the position trajectory generated will follow Eq. (2.39), using the change of representation from x, y, z cartesian coordinates to r, θ polar coordinates.

Higher dimensionality problems of this kind, namely n -body problems, have been widely studied in scientific history. Such problems however, differently than the two-body problem, do not have in general an analytical solution. Other than specific constrained settings, n -body problems with $n \geq 3$ are referred to as chaotic systems, with high sensitivity to initial conditions and whose differential equations can only be numerically integrated, as no closed form solution exists.

A fundamental reminder is that all the description works under the assumption that the bodies in orbit are only subject to the gravitational force of their attractor. In the real world, the orbits are said to be non-Keplerian, meaning they are subjected to perturbations. The most impactful ones are, for instance, aerodynamic drag due to residual atmosphere and irregular mass distributions of planets (i.e. Earth oblateness at the poles) that generate gravity potential harmonics. Among the less significant ones there are solar radiation effects and third-body forces. The incidence of the perturbations depends highly on the time window considered and on the required degree of accuracy of the solutions. The Keplerian orbit assumption works for the scope of this work.

2.1.3 Orbit geometry and orbital elements

Using the results discussed above, Eq. (2.39) appears to be the one of a conic section. For $\theta \in [0, 2\pi]$, using polar coordinates, \mathbf{r} describes a conic.

A conic section (or simply conic) is the curve that is obtained by intersecting a cone and a plane, with given inclination and distance from the vertex of the cone governed by the parameters of the equation. In particular:

- The eccentricity e determines the shape. The conic section can be:
 - ellipse, for $0 \leq e < 1$ (if $e = 0$ it degenerates into a circle)
 - parabola, for $e = 1$
 - hyperbola, for $e > 1$
- The semilatus rectum p determines the size

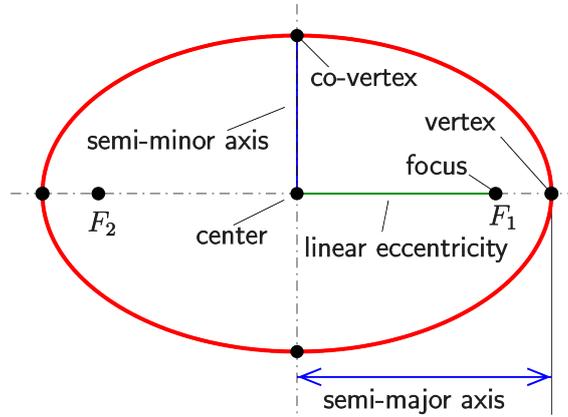


Fig. 2.3 The ellipse and its nomenclature

and θ is calculated from the point of the conic closest to the focus, its origin. The ellipse, particularly relevant in the context of orbital dynamics, is the locus of points whose sum of the distances from two fixed points (focus) is a constant value, equal to $2a$, with $a = p/(1 - e^2)$ being the semi-major axis.

If interpreted as an orbit, the two vertices of the ellipse are called apsides. The periapsis (perigee for an orbit around the Earth, perihelion around the Sun) is the closest apsis to the central planet, it corresponds to $\theta = 0$ and its distance from the main focus is $r_p = p/(1 + e)$. The apoapsis (apogee for an orbit around the Earth, aphelion around the Sun) is the furthest, it corresponds to $\theta = \pi$ and its distance from the main focus is $r_p = p/(1 - e)$. Most of the objects in the Solar System reside in elliptical orbits around the sun.

From the results of Eq. (2.36) one can derive the vis-viva equation:

$$v = \sqrt{2\frac{\mu}{r} - \frac{\mu}{a}} \quad (2.41)$$

which expresses the univocally determined velocity of an object in a given orbit. By computing it for a parabolic orbit, which is the first open conic case, that is with $a \rightarrow \infty$, one can compute the velocity needed for an object to leave a given closed orbit. For any orbital position with radius r it is called escape velocity and is computed as $v = \sqrt{2\mu/r}$.

Assumed that the central planet occupies one of the two foci of a given elliptical orbit, the essential practice is to define a reference frame. The current standard celestial reference system is the International Celestial Reference System (ICRF) used by the International Astronomical Unit. It is centered in the Solar System's center of mass, and its axis are fixed pointing towards distant very active and luminous galaxy nuclei, called quasars, which are assumed to be stationary given their extreme distances. This choice ensures stability and no rotational motion effects.

An easier and sufficiently accurate definition for scopes restricted to Earth orbits is the Earth Centered Inertial (ECI) frame. It is defined as follows:

1. Z-axis (**K**) pointing towards the celestial north pole.
2. X-axis (**I**) pointing towards the vernal equinox. The equinoxes are the two points in which Earth's orbital plane (ecliptic plane) intersects Earth's equatorial plane (celestial equator). The vernal (or March) equinox, for the northern hemisphere, is the point crossed when the Sun is heading northwards, as seen from the Earth.
3. Y-axis (**J**) completes the right-handed triad on the equatorial plane, $\mathbf{J} = \mathbf{K} \times \mathbf{I}$.

Contrarily to the ICRS, the ECI frame does not adjust for small changes in Earth's orientation due to precession, nutation and polar changes, but it is however a sufficiently accurate choice for some applications. For this reason, it needs to be defined as the ICRS for a given epoch of time. Common practice is to consider the ECI equivalent with the ICRS at epoch J2000, i.e. 1st January 2000 of the Julian calendar. A wide variety of possible reference frames is available. The choice is made depending on the requirements of the problem, on the point of view to be considered and on what simplifies calculations. For a thorough description of other reference frames used in space applications, the reader might refer once again to the book in [21].

Defined the reference frame, there is need for a unique definition of an orbit. Among the different ways to mathematically describe one, the most widely used representation are the classical orbital elements. Since the solution of Eq. (2.35) has six degrees of freedom (seven with μ , which is assumed defined), the orbital elements are a set of six independent parameters that uniquely define an orbit, as follows:

1. a , semimajor axis

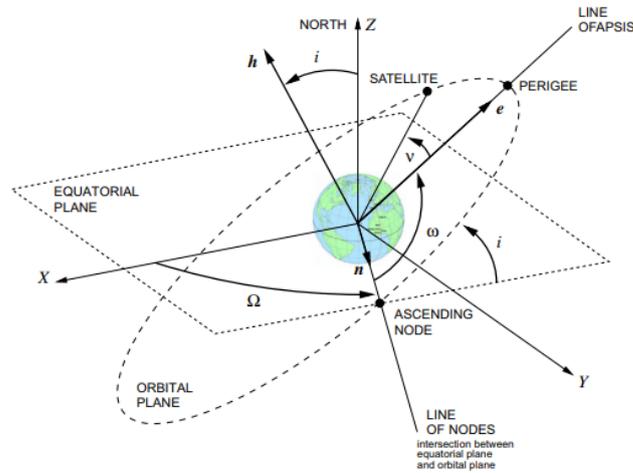


Fig. 2.4 A visualization of the classical orbital elements

2. e , eccentricity
3. i , inclination: the angle between the orbital plane and the fundamental plane of the central planet (reference plane, \mathbf{K})
4. Ω , right ascension of the ascending node: the angle from \mathbf{I} to the ascending node. The ascending node is the point of intersection between the orbit and the reference plane, when the body in orbit crosses from below to above
5. ω , argument of perige: the angle from the ascending node to the periapsis
6. θ (or v), true anomaly: angle from periapsis to the orbiting body position

The choice of such parameters and their notation is not unique in scientific literature.

For easier visualization, the reader might refer to Fig. 2.4 or to the resource in [1], which also provides the standard formulas for representation changes between orbital elements and Euclidean vectors.

Using orbital elements, an astronomical classification of orbits can be defined, with respect to the strictly analytical one, see Tables 2.1 and 2.2.

Rather trivially, depending on the specific setting, some values may be undefined and therefore irrelevant. Moreover, orbit integration models, even the most sophisticated, suffer from propagation errors, which requires periodic updates of the orbital information of objects. This is why better orbit description standards have been developed, with the Two-Line Element (TLE) being the most common. A TLE

Table 2.1 Classification of orbits

Orbit Type	Altitude (km)	Period	Inclination	Eccentricity
Low Earth Orbit (LEO)	160 – 2,000	90–120 min	0°–90° (varies)	≈ 0
Medium Earth Orbit (MEO)	2,000 – 35,786	2–12 hours	Varies (e.g., 55° GPS)	≈ 0
Geostationary Orbit (GEO)	35,786	24 hours	≈ 0°	≈ 0
High Earth Orbit (HEO)	300 – 40,000	12–24 hours	≈ 63.4° (Molniya)	≈ 0.7
Sun-Synchronous Orbit (SSO)	600 – 800	90–100 min	≈ 98° (polar)	≈ 0

Table 2.2 Typical missions per orbit type

Orbit Type	Usual Missions
LEO	Earth observation, ISS, scientific missions
MEO	Navigation (GPS, Galileo), communications
GEO	Communications, weather monitoring, broadcasting
HEO	High-latitude communications, reconnaissance
SSO	Earth observation, remote sensing

Table 2.3 Two-Line Element (TLE) format, line 1

Column	Description
01	Line Number of Element Data
03 – 07	Satellite Number
08	Classification (U=Unclassified)
10 – 11	International Designator (Last two digits of launch year)
12 – 14	International Designator (Launch number of the year)
15 – 17	International Designator (Piece of the launch)
19 – 20	Epoch Year (Last two digits of year)
21 – 32	Epoch (Day of the year and fractional portion of the day)
34 – 43	First Time Derivative of the Mean Motion
45 – 52	Second Time Derivative of Mean Motion (Leading decimal point)
54 – 61	BSTAR drag term (Leading decimal point)
63	Ephemeris type
65 – 68	Element number
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs =0; minus signs =1)

orbit description is provided in a text file whose first line is a 24-character name, and the following lines are 69 characters long and have the information as described in Tables 2.3 and 2.4. Orbit propagation errors have a different impact depending on the type of the orbit, affecting the TLEs update frequency required to maintain their accuracy. Orbits in LEO require almost daily updates, while TLEs for orbits in GEO can be maintained up to weeks.

2.1.4 Relative motion

Extending from the outlined equations presenting an individual spacecraft's dynamics, a model for the description of the spacecraft relative motion is presented. Considering two rigid-body spacecraft orbiting around the earth. Label one spacecraft as the Target and the other as the Servicer. In this setting, the relative motion equations, rather intuitively, aim at providing a description of the relative attitude and relative translation dynamics of the two spacecrafts.

Among several spacecraft relative dynamics models, the one from [5] is used. Such model is particularly suited for usage in a scenario where the Target is assumed to be an uncooperative object, which applies to the scope of this work. The cited

Table 2.4 Two-Line Element (TLE) format, line 2

Column	Description
01	Line Number of Element Data
03 – 07	Satellite Number
09 – 16	Inclination [Degrees]
18 – 25	Right Ascension of the Ascending Node [Degrees]
27 – 33	Eccentricity (Leading decimal point assumed)
35 – 42	Argument of Perigee [Degrees]
44 – 51	Mean Anomaly [Degrees]
53 – 63	Mean Motion [Revs per day]
64 – 68	Revolution number at epoch [Revs]
69	Checksum (Modulo 10)

source references the model in [9], but re-derives the equations expressing them with respect to the Target spacecraft's frame. The dynamics works under the assumption of Keplerian orbits, thus with no perturbations included.

For the chosen representation, the relative translational dynamics is dependent on the relative attitude dynamics, which will be presented first. This formulation shows how to build the equation for the relative dynamics assuming single integration of the two individual attitude and translational dynamics.

Let $q_{TS} = q_S^{-1} * q_T$ be the quaternion error between the attitude of the Target q_T and the attitude of the Servicer q_S , representing the relative orientation between the two. The equivalent rotation matrix $\mathbf{R}_S^T = \mathbf{R}(q_{TS})$ transforms a vector expressed in the Servicer frame to the Target frame with an extrinsic rotation. For the sake of clarity, the trivial representation change from quaternion to rotation matrix is represented as

$$\mathbf{R}(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.42)$$

The relative angular velocity expressed in the Target frame is given as

$$\omega_{ST}^T = \omega_S^T - \omega_T^T = \mathbf{R}_S^T \omega_S^S - \omega_T^T \quad (2.43)$$

knowing that $\omega_{ST}^T = -\omega_{TS}^T$. Relative attitude and relative angular velocity follow the quaternion kinematics equation as in (2.24).

The relative angular acceleration is expressed as

$$\begin{aligned}\alpha_{ST}^T &= \left(\frac{d\omega_{ST}}{dt} \Big|_T \right)^T \\ &= \mathbf{R}_I^T \left(\frac{d\omega_{ST}}{dt} \Big|_I \right)^I - \omega_T^T \times \omega_{ST}^T\end{aligned}\quad (2.44)$$

where the subscripts after the vertical bar represent in which reference frame the differentiation is done, I is the Inertial ECI frame as introduced in 2.1.2, and \mathbf{R}_I^T is the rotation matrix that transforms a vector from the ECI frame into the Target frame, represented by the Target's attitude.

The expression of the relative angular acceleration with respect to the ECI frame, differentiated in the ECI frame, is computed as

$$\left(\frac{d\omega_{ST}}{dt} \Big|_I \right)^I = \left(\frac{d\omega_S}{dt} \Big|_I \right)^I - \left(\frac{d\omega_T}{dt} \Big|_I \right)^I \quad (2.45)$$

Knowing that

$$\begin{aligned}\mathbf{R}_I^T \left(\frac{d\omega_T}{dt} \Big|_I \right)^I &= \left(\frac{d\omega_T}{dt} \Big|_T \right)^T \\ &= \mathbf{J}_T^{-1} (\mathbf{m}_T - \omega_T^T \times \mathbf{J}_T \omega_T^T) \\ \mathbf{R}_I^T \left(\frac{d\omega_S}{dt} \Big|_I \right)^I &= \mathbf{R}_S^T \left(\frac{d\omega_S}{dt} \Big|_S \right)^S \\ &= \mathbf{R}_S^T \left[\mathbf{J}_C^{-1} (\mathbf{m}_C - \omega_S^S \times \mathbf{J}_S \omega_S^S) \right]\end{aligned}\quad (2.46)$$

where \mathbf{J}_T and \mathbf{J}_S are respectively the inertia matrices of the Target and of the Servicer, and \mathbf{m}_T and \mathbf{m}_S are respectively external moments applied to the Target and to the Servicer, the final expression for the relative angular acceleration then becomes

$$\begin{aligned}\alpha_{ST}^T &= \mathbf{R}_S^T \left[\mathbf{J}_S^{-1} (\mathbf{m}_S - \omega_S^S \times \mathbf{J}_C \omega_S^S) \right] \\ &\quad - \mathbf{J}_T^{-1} (\mathbf{m}_T - \omega_T^T \times \mathbf{J}_T \omega_T^T) - \omega_T^T \times \omega_{ST}^T\end{aligned}\quad (2.47)$$

The relative dynamics description is instead computing starting from the expressions of \mathbf{r}_T^T and \mathbf{r}_S^S , which are, respectively, the Target's and Servicer's position expressed in their respective body frames. The relative position is defined as

$$\mathbf{r}_{ST}^T = \mathbf{r}_S^T - \mathbf{r}_T^T = \mathbf{R}_S^T \mathbf{r}_S^C - \mathbf{r}_T^T \quad (2.48)$$

The relative velocity expressed in the target frame is obtained as:

$$\mathbf{v}_{ST}^T = \left(\frac{d\mathbf{r}_{ST}}{dt} \Big|_T \right)^T = \left(\frac{d\mathbf{r}_{ST}}{dt} \Big|_I \right)^T - \boldsymbol{\omega}_T^T \times \mathbf{r}_{ST}^T, \quad (2.49)$$

$$= \mathbf{R}_S^T \mathbf{R}_I^S \left(\frac{d\mathbf{r}_{ST}}{dt} \Big|_I \right)^I - \left(\mathbf{R}_S^T \boldsymbol{\omega}_S^S - \boldsymbol{\omega}_T^T \right) \times \mathbf{r}_{ST}^T. \quad (2.50)$$

developed using Eq. (2.43).

The relative acceleration in the Target frame is computed as

$$\begin{aligned} \mathbf{a}_{ST}^T &= \left(\frac{d^2\mathbf{r}_{ST}}{dt^2} \Big|_T \right)^T \\ &= \left(\frac{d^2\mathbf{r}_{ST}}{dt^2} \Big|_I \right)^T - 2\boldsymbol{\omega}_T^T \times \mathbf{v}_{ST}^T \\ &\quad + \left(\frac{d\boldsymbol{\omega}_T}{dt} \Big|_T \right)^T \times \mathbf{r}_{ST}^T + \boldsymbol{\omega}_T^T \times (\boldsymbol{\omega}_T^T \times \mathbf{r}_{ST}^T) \end{aligned} \quad (2.51)$$

The relative acceleration term expressed in the Target frame, differentiated with respect to the ECI frame, is simply corresponding to

$$\left(\frac{d^2\mathbf{r}_{ST}}{dt^2} \Big|_I \right)^T = \left(\frac{d^2\mathbf{r}_S}{dt^2} \Big|_I \right)^T - \left(\frac{d^2\mathbf{r}_T}{dt^2} \Big|_I \right)^T \quad (2.52)$$

The absolute accelerations of the Target and the Servicer with respect to the ECI frame are

$$\left(\frac{d^2 \mathbf{r}_S}{dt^2} \Big|_I \right)^T = -\frac{\mu}{r_S^3} \mathbf{R}_S^T \mathbf{r}_S^S \quad (2.53)$$

$$\begin{aligned} \left(\frac{d^2 \mathbf{r}_T}{dt^2} \Big|_I \right)^T &= -\frac{\mu}{r_T^3} \mathbf{r}_T^T \\ &= -\frac{\mu}{\|\mathbf{R}_S^T \mathbf{r}_S^S - \mathbf{r}_{ST}^T\|^3} (\mathbf{R}_S^T \mathbf{r}_S^S - \mathbf{r}_{ST}^T). \end{aligned} \quad (2.54)$$

which correspond to the standard Keplerian free restricted two body equation in Eq. (2.35). By substituting Eqs. (2.52), (2.53) and (2.54) into Eq. (2.51), the complete expression of the relative translational dynamics expressed in the Target's body frame yields

$$\begin{aligned} \mathbf{a}_{ST}^T &= -\frac{\mu}{r_S^3} \mathbf{R}_S^T \mathbf{r}_S^S \\ &+ \frac{\mu}{\|\mathbf{R}_S^T \mathbf{r}_S^S - \mathbf{r}_{ST}^T\|^3} (\mathbf{R}_S^T \mathbf{r}_S^S - \mathbf{r}_{ST}^T) \\ &+ \mathbf{J}_S^{-1} (\mathbf{m}_S - \boldsymbol{\omega}_S^T \times \mathbf{J}_S \boldsymbol{\omega}_S^T) \times \mathbf{r}_{ST}^T \\ &- 2\boldsymbol{\omega}_S^T \times \mathbf{v}_{ST}^T + \boldsymbol{\omega}_S^T \times (\boldsymbol{\omega}_S^T \times \mathbf{r}_{ST}^T) \end{aligned} \quad (2.55)$$

The complete dynamics of the spacecraft relative motion problem, expressed in state-space representation, thus propagates as follows:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} \Big|_T = \frac{d}{dt} \begin{bmatrix} \mathbf{q}_{TS} \\ \boldsymbol{\omega}_{ST}^T \\ \mathbf{r}_{ST}^T \\ \mathbf{v}_{ST}^T \end{bmatrix} \Big|_T = \begin{bmatrix} \frac{1}{2} \mathbf{Q}(\mathbf{q}) \boldsymbol{\omega}_{TS}^T \\ \boldsymbol{\alpha}_{ST}^T \\ \mathbf{v}_{ST}^T \\ \mathbf{a}_{ST}^T \end{bmatrix} \quad (2.56)$$

As stated in the introduction, all the description is presented in its most general form. At the same time, it provides also the expressions that allow to derive a representation solely dependent on relative states and absolute states of the Servicer, which will be needed later during the discussion.

2.2 Kalman filtering

The Kalman Filter is a well established estimation algorithm introduced by Rudolf E. Kalman in 1960 [12]. It finds applications in a wide variety of fields, ranging from economics and medicine to robotics and aerospace, while being also particularly well suited for real-time applications in general. It in fact references an internal representation of the system, and can take into account prior modeled knowledge of the environment. During the online processing, at a reasonable computational cost, the filter uses new measurements as they arrive for continuously refining the estimates. The Kalman filter is in fact defined as a recursive estimator. This means that, in contrast to batch estimation techniques, only the estimation at the previous time step and the current estimate are needed, and no history of the observations is required.

A general description will follow, without going into too much mathematical detail. Different formulations, which are derived on the most general one, will then be presented. The representation in discrete time is chosen, as relatively simpler and more suitable for direct online implementation. The reader is expected to intuitively interpret vector expressions, with bold notation reserved for vectors with physical meaning, as in the previous section.

Consider a generic discrete-time Linear Time Varying (LTV) system, described by the standard state equations

$$\begin{aligned}x_{k+1} &= F_k x_k + G_k u_k + d_k \\ y_k &= H_k x_k + d_k^y\end{aligned}\tag{2.57}$$

where $k \in \mathbb{Z}$ is the time index, $x_k \in \mathbb{R}^n$ is the state, u_k is the input, $y_k \in \mathbb{R}^m$ is the output, d_k is the process noise and d_k^y is the measurement noise. In this context, suppose that x_k, d_k, d_k^y are unknown, not measured, and y_k, u_k are measured.

The goal of a Kalman Filter is then to obtain a possibly accurate estimate \hat{x}_k of x_k , from current and past measurements of y_k and u_k .

First, key assumptions have to be stated. Assume that both process noise d_k and measurement noise d_k^y are independent and identically distributed and white noises. This mean they have zero mean ($\mathbb{E}[d_k] = \mathbb{E}[d_k^y] = 0$), bounded variance and

auto-correlation (which is auto-covariance for a zero-mean process) equal to

$$\begin{aligned}\mathbb{E}[d_k d_i^T] &= \delta_{ki} Q_d, & i \in \mathbb{R}^n \\ \mathbb{E}[d_k^y (d_i^y)^T] &= \delta_{ki} R_d, & i \in \mathbb{R}^q\end{aligned}\quad (2.58)$$

where $\delta_{kk} = 1$, $\delta_{ki} = 0$ with $i \neq k$ is the Kronecker delta. Following this representation, two matrices arise, Q_d and R_d , respectively the process noise covariance matrix and the measurement noise covariance matrix, whose form and effects will be discussed later. Moreover, those disturbances are uncorrelated mutually and with respect to the inputs, which means, respectively, $\mathbb{E}[d_k (d_i^y)^T] = 0$ and $\mathbb{E}[d_k u_i^T] = \mathbb{E}[d_i^y u_i^T] = 0$. This essentially models noise in a rather simple and mathematically convenient way, following a common assumption on engineering applications that nonetheless reflects the majority of real world disturbances.

Like many other state observer algorithms, the Kalman Filter (KF) is based on two fundamental operations: a prediction step, followed by an update step, which are presented below.

1. Prediction

Knowing the estimated state of the system at time $k - 1$, the filter predicts the future state estimate \hat{x}_k^- using the system model

$$\hat{x}_k^- = F_{k-1} \hat{x}_{k-1} + G_{k-1} u_{k-1} \quad (2.59)$$

At the same time, a prediction of the error covariance matrix P_k is computed:

$$P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q_d \quad (2.60)$$

2. Update

The filter then corrects the prediction \hat{x}_k^- using the current output measurement y_k , producing the updated estimate

$$\hat{x}_k = \hat{x}_k^- + K_k \delta y_k \quad (2.61)$$

$$\delta y_k = y_k - H_k \hat{x}_k^- \quad (2.62)$$

where δy_k is called measurement innovation term, and the Kalman gain matrix K_k is computed as

$$S_k = H_k P_k^- H_k^T + R_d \quad (2.63)$$

$$K_k = P_k^- H_k^T S_k^{-1} \quad (2.64)$$

The error covariance matrix is also updated at this step, evolving according to the Difference Riccati Equation (DRE)

$$P_k = (I - K_k H_k) P_k^- \quad (2.65)$$

As clearly visible, a crucial role is played by the Kalman gain matrix K_k during the process. It in fact, on a high level, weights the innovation term δy_k in the update step, and determines how much the predicted estimates and the new sensor measurements should be weighted to ensure that the estimation error is minimized. From a rigorous point of view, it can be demonstrated that, under the aforementioned assumptions, in a globally observable system, a Linear Kalman Filter guarantees an estimation error with zero mean and minimum variance. This means that the Kalman gain computed in (2.64) is the optimal solution in terms of the Least Square problem

$$K_k = \arg \min_{\mathcal{K}} \mathbb{E} [\|x_k - \hat{x}_k\|_2^2], \quad \forall k \quad (2.66)$$

The detailed analysis of the algebraic derivation [12] is beyond the scope of this thesis and would be redundant.

As very common in a wide variety of engineering solutions, the implementation of a Kalman Filter requires some design phases and possibly some previous knowledge of the environment under analysis. First rather evident subjects are the Q_d and R_d matrices, respectively the covariance matrix of the process noise d_k and of the measurement noise d_k^y . The values of these matrices basically embed the "trust" that is given on the model of the process and on the quality of the measurements (lower value, higher confidence) and have an impact on the convergence time and the amplitude of the oscillations at steady state (lower value, fewer oscillations but slower convergence). Typically, they are chosen as diagonal matrices with the variances of d_k and d_k^y on the diagonal. Such values, as explained, can be modeled from prior understanding of the environment, and usually require trial and error tuning procedures. Another crucial aspect in the implementation of a Kalman Filter

is its sampling frequency. There is usually little room for design of such value, as it is forcedly constrained by the one of slowest (in processing frequency sense) processing unit in the real pipeline implementation. The value of the sampling rate, quite intuitively, has quite a big impact on the performance of the estimation process, both in terms of accuracy and convergence time.

In the end, a relevant role in the process is played by the chosen initializations of the estimated variables, \hat{x}_0 and P_0 , which are typically set to 0 and I respectively. Once again, however, difference choices can be made if any a priori information on the system is available. If any, the initial value of P_0 shall reflect the confidence that is given on those (lower value, higher confidence). Accurate initialization values for the Kalman Filter are of extreme value to achieve faster convergence and better performance, and can be one of the most impactful design choices. This is why this aspect is in fact is a huge foundation for the scope of this work.

As presented earlier, different extensions and applications of the Kalman Filter have been developed starting from this basic formulation [15]. Continuing the chapter, three versions of this model are presented: the Linear Time Invariant (LTI) Kalman Filter, simplified for LTI systems, the Extended Kalman Filter, an extension for nonlinear systems, and the Unscented Kalman Filter, built for nonlinear systems with an innovative procedure for generating state predictions.

2.2.1 Linear Kalman Filter

The Linear Time Invariant (LTI) Kalman Filter represents a simplification of the most general formulation of the Linear Kalman Filter appropriate for, as the name suggests, LTI systems.

Once again, the formulation will be directly presented in discrete-time for simplicity. Consider the discrete-time LTI system described by

$$\begin{aligned} x_{k+1} &= Fx_k + Gu_k + d_k \\ y_k &= Hx_k + d_k^y \end{aligned} \tag{2.67}$$

The same assumptions on noise characterization as in Eq. (2.57) hold. For this kind of system, as very common result, it can be proved that, if $Q_d, R_d \succ 0$ and F, H are observable, then the error covariance matrix is guaranteed to converge to a

steady-state solution

$$\lim_{k \rightarrow \infty} P_k^- \rightarrow \bar{P} \quad (2.68)$$

Consequently, this implies that also P_k , S_k and K_k converge to given steady-state values P , S and K as $k \rightarrow \infty$. The equations of the Kalman Filter become then the following asymptotic equations:

$$\begin{aligned} \bar{P} &= FPF^T + Q_d \\ S &= H\bar{P}H^T + R_d \\ K &= \bar{P}H^T S^{-1} \\ P &= (I - KH)\bar{P} \end{aligned} \quad (2.69)$$

What in the general formulation was described as a Difference Riccati Equation, Eq. (2.65) here is instead an Algebraic Riccati Equation (ARE) of the form

$$\bar{P} = F(\bar{P} - \bar{P}H^T(H\bar{P}H^T + R_d)^{-1}H\bar{P})F^T + Q_d \quad (2.70)$$

from whose solution, through simple operations, the Kalman gain expression becomes

$$K = \bar{P}H^T(H\bar{P}H^T + R_d)^{-1} \quad (2.71)$$

The real-time estimation, according to the formulation described above, then follows the same steps prediction and update steps as in the general description.

The key property of the LTI Kalman Filter implementation is the fact that the solution of Eq. (2.70), and consequently the Kalman gain matrix, can be computed offline. Those matrices will then remain constant during the computation, thus drastically reducing the computational effort required. The LTI Kalman Filter works in steady-state and yet guarantees convergence to the optimal solution, with minimum variance estimation error.

2.2.2 Extended Kalman Filter

The Extended Kalman Filter (EKF), originally developed by the same scientist R.E. Kalman [13] constitutes the easiest and most straightforward extension of the Kalman Filter for nonlinear systems.

Consider the discrete-time nonlinear system

$$\begin{aligned}x_{k+1} &= f(x_k, u_k) + d_k \\ y_k &= h(x_k) + d_k^y\end{aligned}\tag{2.72}$$

The same assumptions on noise characterization as in Eq. (2.57) hold.

The main goal of EKF is to adapt such modeling to the framework of the most general LTV Kalman Filter. To achieve so, the easiest solution is therefore to linearize the nonlinear system around the estimated state trajectory. That is, at each time step, compute

$$\begin{aligned}F_k &= \frac{\partial f}{\partial x}(\hat{x}_k, u_k) \\ H_k &= \frac{\partial h}{\partial x}(\hat{x}_k)\end{aligned}\tag{2.73}$$

where F_k is the Jacobian of f evaluated at (\hat{x}_k, u_k) and H_k is the Jacobian of h evaluated at (\hat{x}_k) . Consequently, this allows to perform the standard estimation procedure, adapting the equations accordingly. For improved clarity, an estimation step of the EKF is made explicit below.

1. Prediction

$$\begin{aligned}x_k^- &= f(\hat{x}_{k-1}, u_{k-1}) \\ P_k^- &= F_{k-1}P_{k-1}F_{k-1}^T + Q_d\end{aligned}\tag{2.74}$$

2. Update

$$\begin{aligned}S_k &= H_k P_k^- H_k^T + R_d \\ K_k &= P_k^- H_k^T S_k^{-1} \\ \delta y_k &= y_k - h(x_k^-) \\ \hat{x}_k &= x_k^- + K_k \delta y_k \\ P_k &= (I - K_k H_k) P_k^-\end{aligned}\tag{2.75}$$

In such formulation, all assumptions made for the general structure still apply, as well as all the design choices required.

Fundamental characteristic of the EKF is that it is suboptimal: unlike its linear implementations, it does not guarantee the optimality of the solution. Its main advantage resides clearly in the relatively straightforward implementation with respect to the Linear case. This however comes at the cost of being potentially

seriously affected by model uncertainties or high linearization errors, making it impossible to guarantee convergence or stability in presence of highly nonlinear systems. Another relevant drawback is that the derivation of the Jacobian matrices can be rather nontrivial in most cases.

As a matter of fact, the EKF has demonstrated to be effective for applications where models have low nonlinearities. For instance, in the space domain, it is well-suited for pose estimation problems in relative navigation scenarios, as overviewed in [19], where a Multiplicative EKF (MEKF) has been developed and proven particularly successful.

2.2.3 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is another extension of the Kalman Filter for nonlinear systems. It was originally developed to provide a framework for nonlinear estimation that did not require any linearization [11]. The UKF employs in fact a transformation, namely Unscented Transformation, to handle the nonlinearities in the model. On a high level view, the core process behind the UKF is, at each step, to generate a set of points, called Sigma Points, that sample that distribution of the state estimate. Such points are then propagated individually through the nonlinear model, and then the distribution of such points is retrieved, thus retrieving the new state estimate. Its functioning will be clarified by the detailed mathematical description that follows.

Consider the discrete-time nonlinear system

$$\begin{aligned}x_{k+1} &= f(x_k, u_k) + d_k \\ y_k &= h(x_k) + d_k^y\end{aligned}\tag{2.76}$$

The same assumptions on noise characterization as in Eq. (2.57) hold. The first step involves the creation of the aforementioned sigma points as

$$\begin{aligned}\sigma_{k-1} &= \pm \sqrt{(n + \lambda)[P_k + Q_d]} \\ \chi_{k-1}(i) &= \sigma_{k-1}(i) + \hat{x}_{k-1} \\ \chi_{k-1}(0) &= \hat{x}_{k-1}\end{aligned}\tag{2.77}$$

where \sqrt{M} is notation for a matrix Z such that $M = ZZ^T$, namely matrix square root, $i \in (0, 2n)$ and λ is a parameter. The subscript k refers to the current variables and estimates, that is, to the one coming from the previous iteration.

The value of α controls the spread of distribution of the sigma points, and it is usually taken as a small number $0 < \alpha \leq 1$. The value of κ usually set to 0 provides a way to fine tune the higher order moments, and β weights the mean of the sigma points incorporating prior knowledge of the distribution, and is usually taken $\beta = 2$ when dealing with Gaussian assumptions. Diverse notations for those parameters can be found in the literature, but the reader shall know that all are equivalent and can be refactored in the one proposed.

By means of the covariance matrix P , the distribution of the estimation error is sampled and the mean of the points is shifted towards the last state estimate, whose unperturbed value is also kept inside the set of sigma points.

Each one of the sigma points is propagated forward using the nonlinear model in Eq. (2.76)

$$\chi_k(i) = f[\chi_{k-1}(i), k-1] \quad (2.78)$$

and the predicted mean is computed as

$$\hat{x}_k^- = \frac{1}{n+\lambda} \left[\lambda \chi_k(0) + \frac{1}{2} \sum_{i=1}^{2n} \chi_k(i) \right] \quad (2.79)$$

Consequently, the predicted covariance, not yet *corrected* via the measurement innovation term, is given by

$$P_k^- = \frac{1}{n+\lambda} \left[\lambda (\chi_k(0) - \hat{x}_k^-)(\chi_k(0) - \hat{x}_k^-)^T + \frac{1}{2} \sum_{i=1}^{2n} (\chi_k(i) - \hat{x}_k^-)(\chi_k(i) - \hat{x}_k^-)^T \right] + Q_d \quad (2.80)$$

To complete the prediction step, the mean observation and the output covariance are computed:

$$\hat{y}_k^- = \frac{1}{n + \lambda} \left[\lambda \gamma_k(0) + \frac{1}{2} \sum_{i=1}^{2n} \gamma_k(i) \right] \quad (2.81)$$

$$P_k^{yy} = \frac{1}{n + \lambda} \left[\lambda (\gamma_k(0) - \hat{y}_k^-)(\gamma_k(0) - \hat{y}_k^-)^T + \frac{1}{2} \sum_{i=1}^{2n} (\gamma_k(i) - \hat{y}_k^-)(\gamma_k(i) - \hat{y}_k^-)^T \right] + R_d \quad (2.82)$$

where

$$\gamma_k(i) = h[\chi_k(i), k - 1] \quad (2.83)$$

are the updated measurements, taken for each sigma point. As a last intermediate step, the cross-correlation matrix is calculated as

$$P_k^{xy} = \frac{1}{n + \lambda} \left[\lambda (\chi_k(0) - \hat{x}_k^-)(\gamma_k(0) - \hat{y}_k^-)^T + \frac{1}{2} \sum_{i=1}^{2n} (\chi_k(i) - \hat{x}_k^-)(\gamma_k(i) - \hat{y}_k^-)^T \right] \quad (2.84)$$

and the innovation covariance matrix results as

$$P_k^{vv} = P_k^{yy} + R_d \quad (2.85)$$

Finally, the update step follows the same computation as in Eq. (2.61) the standard Linear Kalman Filter formulation, with the computation of the Kalman gain, the measurement innovation and updated state estimate as follows:

$$\begin{aligned} K_k &= P_k^{xy} (P_k^{vv})^{-1} \\ P_k &= P_k^- - K_k P_k^{vv} K_k^T \\ v_k &\equiv y_k - \hat{y}_k^- = y_k - h(x_k^-, k) \\ \hat{x}_k &= \hat{x}_k^- + K_k v_k \end{aligned} \quad (2.86)$$

where y_k is the real measurement obtained at time k .

Among the advantages of the UKF with respect to the EKF, apart from avoiding the derivation of the Jacobians of the process model, as already mentioned, there are lower expected errors and, reasonably, validity for higher-order expansions. The principal drawbacks are usually higher computational costs and a more complex tuning process. Moreover, once again, as very common in nonlinear models, the optimality of the solution is never guaranteed.

Several extensions for the Unscented Kalman Filter have been proposed and developed in the literature, to adapt the device to specific applications and environments. In this sense, the most relevant for this thesis are the ones that try to cope with the unitary norm constraint of the quaternion, first among all the Unscented Quaternion Estimator (USQUE) [7], that will be later presented.

2.2.4 Outlier rejection

Filtering theory has been widely explored in scientific literature, and advanced techniques have been studied to overcome the main disadvantages of the Kalman Filter. Arguably, the most important assumption behind any Kalman Filter model is on the distribution of the noise sources, which need to be independent, identically distributed and white, as shown in Eq. (2.58). If the real noise behaves differently, the Kalman Filter would have no distribution to model and would thus diverge rather rapidly. This is why scientists and engineers have developed techniques to increase the robustness of the filter.

One of the main ideas to make a filter robust is to provide it with the capability of detecting and rejecting measurement outliers. An outlier is a value that deviates significantly from the rest of its set, in this case a measurement that deviate significantly from its distribution and is likely originated by a faulty sensor. Among the variety of different possible techniques [14], the one proposed in [18] has been chosen and used in this work. Broadly speaking, the key idea is to, at each step, compare the innovation term calculated using the latest measurement to a given confidence threshold, and consequently decide whether to compute the filter's update step or to discard the measurement thus propagating the predicted state.

In particular, a measurement validation gate is set around the innovation term computing the normalized estimation error squared

$$e_k^2 = v_k^T S_k^{-1} v_k \quad (2.87)$$

where v_k is the innovation term and S is the innovation covariance matrix (P_k^{vv} for a UKF). The innovation covariance matrix acts as a way to weight the validation gate with the trust that the filter would have given anyway, avoiding discarding measurement that could result in a high innovation term simply because, for example, the filter is still in convergence phase. The normalized estimation error squared e_k^2 varies as a Chi-Squared (χ^2) distribution with m is the dimension of the measurement vector y_k . By performing a Chi-squared test for a given confidence value it can be determined whether the measurement is to be kept or discarded. This is one of the most widely used approaches in statistics for hypothesis testing theory and construction of confidence intervals.

A Chi-squared distribution with k degrees of freedom (please note the temporary notation override) is the distribution of a sum of the squares of k independent standard normal random variables, by definition, always positive and not symmetric about its mean value, built as

$$Y = \sum_{i=1}^k \left(\frac{X_i - \mathbb{E}[X_i]}{\sigma} \right)^2 \quad (2.88)$$

where X_i is a random variable, $\mathbb{E}[X_i]$ is the expected value and σ is the standard deviation. The larger the value of Y , the more the observed data, i.e. the measurement, deviates from the expectations.

If you want to test your measurement against a α confidence, you have to find the bounding values that cover $(1 - \alpha)\%$ of the total area of the distribution. That is, the Kalman filter at each step verifies if the result of Eq. (2.87) falls inside the interval (χ_L^2, χ_U^2) , as shown in Fig. 2.5. Those two boundaries, lower and upper, are numerically computed as the inverse of the Chi-squared cumulative distribution function with m degrees of freedom at the values, respectively, $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$.

If a measurement is detected as faulty, the Kalman Filter discards it by not performing its update step, thus relying only on the predicted state value. In case the measurement vector is composed of values coming from more than one source, the outlier rejection mechanism is applied individually to each portion of the mea-

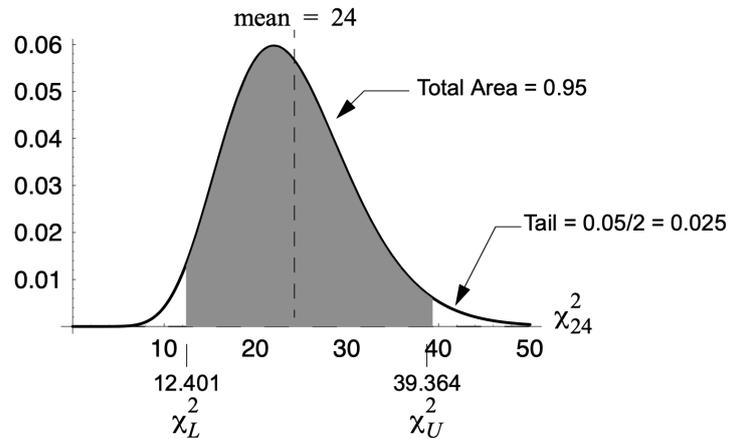


Fig. 2.5 Chi-squared distribution with 24 degrees of freedom. Critical values for 95% confidence, from [18]

surement vector, as possibly not the full measurement vector is to be considered as faulty. From the implementation perspective, having single measurement vector components that can fail independently requires defining individual update step execution flows. Each single filter update execution path, triggered by a potential fault of another measurement portion, keeps then track of a covariance matrix R restricted to the non-faulty measurement, which undergoes the standard update step of the Kalman Filter as in Eq. (2.61) (or the respective equivalents for the Kalman Filter extensions).

This technique is effectively able to detect measurement fault windows, as far as their duration is limited. Having no chance of updating the measurements would inhibit the update step, dropping the main concept behind the Kalman Filter algorithm. Faults occurring too frequently or lasting too long lead in fact the filter to diverge.

Chapter 3

Light curves

The goal of this section is to present what a light-curve is how can one be used to infer knowledge on the object that generated it.

The definition of light curve will be first presented, alongside with an overview of current light-curves analysis techniques. Some results for the chosen implementation are then presented to justify the assumed quality of the motion priors extraction.

3.1 Definition and analysis techniques

A light curve is a plot of the variation in brightness of an astronomical object over time. It is obtained by measuring the intensity of reflected light of an object from ground-based observation.

In astronomy, a light curve constitutes thus a powerful technique that can be used to extract valuable information about the object that generated it: its motion is in fact strictly related to the periodicity that the light curve exhibits. Light curve analysis has many applications, including determining rotation rates, shapes, taxonomic classification of solar system objects, characterizing exoplanets and extrasolar planetary systems and variable stars.

In the context of satellite and orbit debris tracking, the brightness variations are directly related to the rotation of the object and the observation geometry. Performing light curve frequency analysis allows to determine the attitude of the observed object and provides important information for ADR and helps to model effects that influence

the long-term dynamics of the object, such as the electromagnetic field, drag, solar radiation, fuel sloshing or outgassing. This process is generally referred to as light-curve inversion, or simply as light-curve analysis, when the focus is on extracting the object's rotation rate without inferring information about its mass distribution.

The most frequently used methods to extract the apparent (synodic) period from the light curves are Fourier-based techniques. A comprehensive review on the analysis techniques of optical measurements from space objects is given in [22].

In this work, we apply a method based on Fourier transform and CLEAN algorithm (WindowCLEAN). This method was developed to characterize the complex rotation of asteroids and comets in the solar system [16] and can reliably detect the dominant frequencies in light curves obtained from non-principal axis rotators. It was suggested that combining the periodicity analysis of light curves with WindowCLEAN with the shape inversion of radar observations may be a powerful tool to deduce non-principal axis spin states by significantly reducing the spin state parameter space that the shape inversion algorithm has to search. This is why this tool has been selected.

Extracting motion priors from an object in space rotating about a single axis is relatively straightforward. Since an object orbiting about a single body axis is characterized by a constant angular velocity, the period extracted by means of the frequency analysis directly corresponds to the synodic period and is interpreted as the rotation rate of the object that generated the light curve. The rotation rate extracted is what will be referred to as kinematic prior.

While challenging, light curves scientific literature shows that it is possible to extract rotation rates for object rotating about a nontrivial axis, resulting from superposition of motion about multiple body-fixed axes. As results from Eq. 2.6, an object orbiting about more than one of its body axes exhibits a tumbling motion, with the angular velocity being non-constant due to inertia coupling effects. For this reason, accounting for the axis precession period requires knowledge of the targets body geometry, which cannot be always assumed for an uncooperative target if no other shape determination methods are employed. Moreover, assuming a rotation rate can be extracted, enabling usage of such motion prior usage would require adjusting for phase, due to the time shift between the light curve observation and the actual proximity operation with the Target.

In the light-curves analysis context, the main issue lies in the scarce availability of light curves and related ground truth attitude data to effectively enable validation of the analyses. This is why, for the scope of this work, simulated light-curves with thus known related angular velocity value have been produced making use of a Light-curves Simulator.

3.2 Light-curves Simulator

Thanks to Luís Gonçalves, a PhD student from University of Coimbra, we had the opportunity to have access to a Light-curve Simulator under his development.

Given an object model and a specific orbit TLE (Tables 2.3 and 2.4), the model propagates the orbit and integrates the attitude motion. For a specified ground-based observer on the surface of the earth the reflection of the sunlight onto the object surface is computed with a given time step for a specified observation window. In particular, using the Cook-Torrance reflection model described in [6], a synthetic light curve is thus generated.

As the implementation currently stands, the tumbling integration only supports one axis rotation with constant speed scenarios, and no additional simulated noise is included.

3.3 One-axis tumbling analysis

This section describes the experiments carried out applying a light-curve analysis technique on one-axis tumbling light curves.

As outlined in Section 3.1, the WindowCLEAN frequency analysis method has been implemented. Given the brightness samples of a light curve, the algorithm produces a dirty spectrum of the periodicities latent in the observations through a Discrete time Fourier Transform that is then iteratively CLEANed from aliases and unpredictable spurious periodicities.

An example with a synthetic light curve generated with the Simulator outlined in Section 3.2 is shown first. After, a result on a high-fidelity simulated light curve accessed in collaboration with our industry partner ClearSpace is presented.

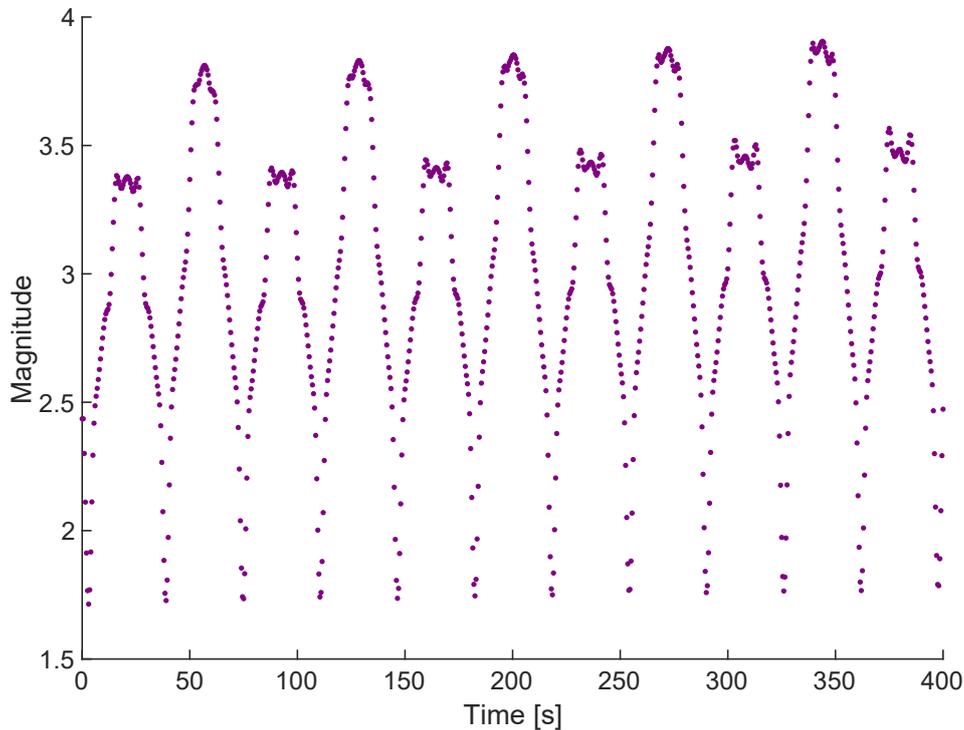


Fig. 3.1 Synthetic Light Curve of an Atlas Centaur R/B. One axis tumbling, 5 deg/s

3.3.1 Synthetic data

An example of a synthetically generated light curve, produced with the Simulator described in Section 3.2, is presented in Fig. 3.1. It is generated by the simulated tumbling motion of an Atlas Centaur Rocket Body model, similar to the one proposed in [4]. The body is rotating about its z axis, aligned with its shorter dimension, observed for a time window of 400 seconds with a sampling time of 0.5 seconds.

The WindowCLEAN tool has been fed with this light curve, and it was able to recover the clean spectrum shown in Fig. 3.2.

The highest component of the extracted spectrum appears at frequency 0.028 Hz, which corresponds to a period of around 35 seconds. Inspecting the nature of the light curve, exhibiting peaks at different magnitudes values, one can infer that the two different magnitude peaks are not generated by the same reflecting surface of the body.

In fact, while rotating, the rocket body periodically exposes the two ends of its cylindrical shape, which reflect with different intensities. The resulting frequency

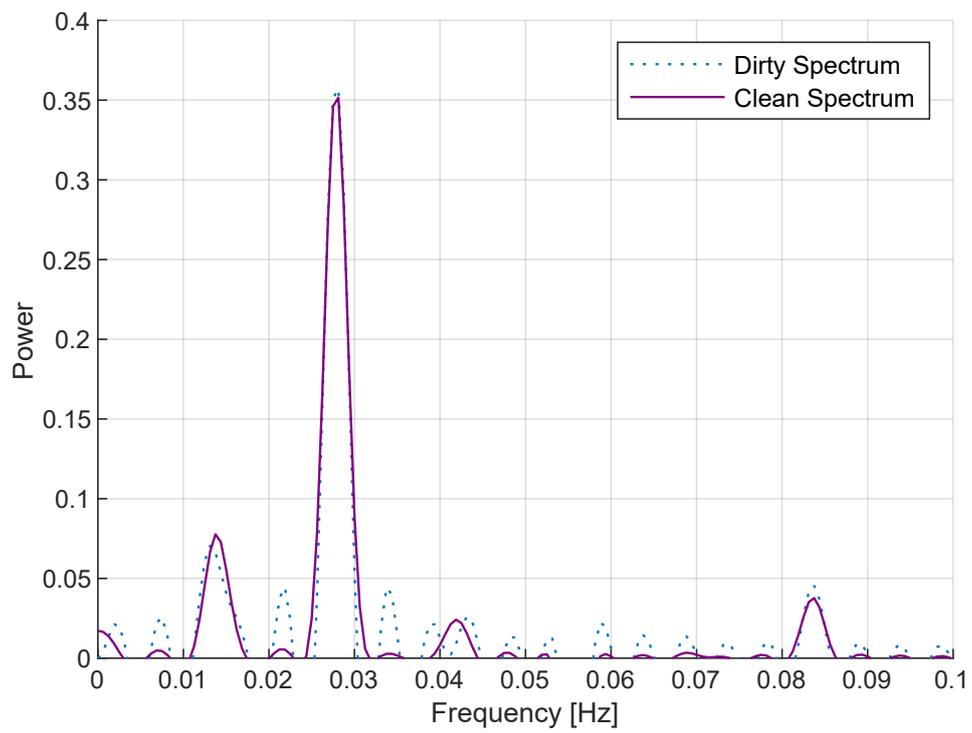


Fig. 3.2 Spectrum of a synthetic Light Curve of an Atlas Centaur R/B. One axis tumbling, 5 deg/s

corresponds thus half to half the turns of the rocket body, that produces two different brightness values for its two extremities.

The recovered rotation rate, associated to half the extracted period, thus corresponds to 5.04 deg/s. For a ground truth constant rotation rate of 5 deg/s, the obtained precision is 99%.

3.3.2 Commercial data

As introduced, we had the opportunity to validate our work on commercial data from industry partner ClearSpace. We had access to a generated light curve based on high fidelity synthetic ground based observations of a tumbling VEga Secondary Payload Adapter (VESPA), corresponding to the one currently left in Low Earth Orbit after a launch in 2013. The VESPA is characterized by a nearly conical shape, of approximately 1.6 meters in height and 2.3 meters in diameter, with variations depending on specific mission requirements. We are unfortunately unable to share the raw brightness data.

Once again, the frequency analysis using the WindowCLEAN tool has been conducted on the light curve. The results of the motion prior extraction process resulted in the recovered frequency spectrum shown in Fig. 3.3.

The commercial light curve analyzed is 550 seconds long, with a sampling time of 1 s. As results from Fig. 3.3, the highest and most relevant frequency component is detected at 0.0082 Hz, which leads to a rotation rate of around 2.9 deg/s. Given the available ground truth value of 3 deg/s, the resulting precision is around 97%.

3.4 Motion prior extraction assumption

The experiments conducted in 3.3 had the goal of identifying the workflow of a light-inversion process and characterizing the derived results. The next sections will make use of the results obtained, and, as anticipated, will analyze the integration of the extracted angular rates into a State estimation model. Therefore, this section enabled defining realistic assumptions on the quality of the kinematic motion priors resulting from a light-curve analysis process.

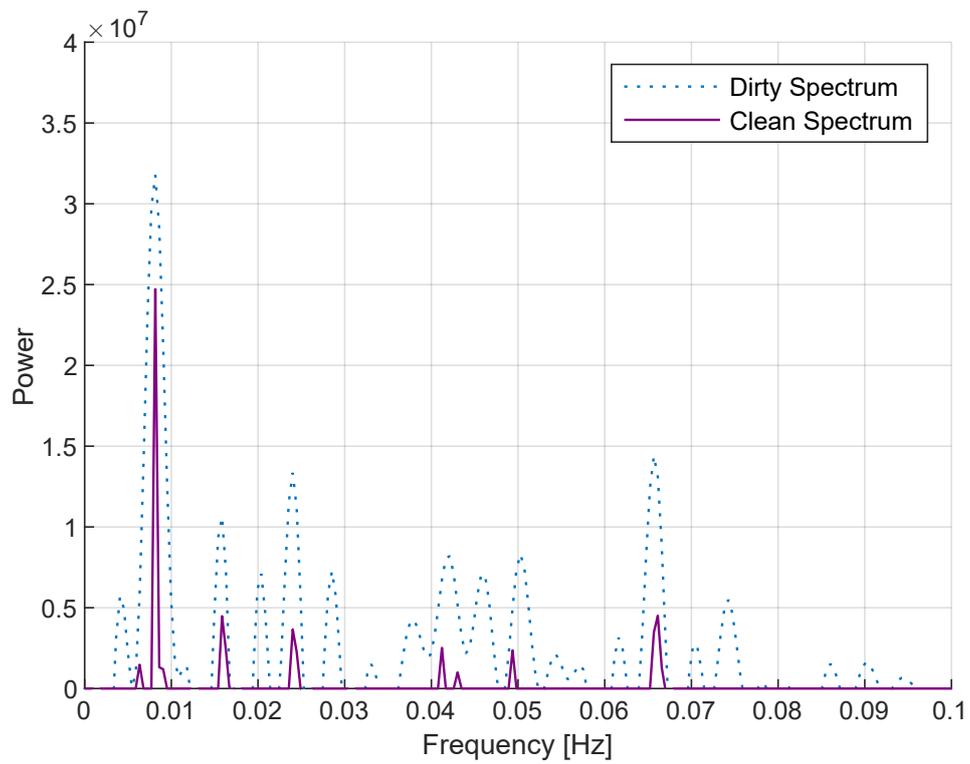


Fig. 3.3 Spectrum of a Light Curve of a VESPA commercial model. One axis tumbling, 3 deg/s

In particular, it is assumed that motion priors are obtainable for both single axis and multi-axes tumbling cases, being aware of the results that exist in the literature. Specifically, for the entirety of the work, we assume a conservative motion prior rotation rate extraction error of 5% relative to the ground truth rotation rates.

Chapter 4

State estimation model

This chapter introduces the primary simulation framework used in this study. The first section describes how the background mathematical theory is employed to build a state estimation model for a proximity rendezvous mission with an uncooperative target. Since a real-world application of the state estimation model would rely on real-time measurements provided by a pose estimation neural network, the process of simulating these real-time measurements is then described afterwards.

4.1 State modeling

In the simulation environment set up for this work, two Kalman Filters have been developed. As introduced in the previous chapters, the problem that has to be solved is to estimate the internal states, angular velocity most importantly, of a Target uncooperative object in orbit. Specifically, a Servicer spacecraft operates in proximity to the Target, utilizing real-time pose estimations provided by a neural network model.

In this work, an Extended Kalman Filter and an Unscented Kalman Filter implementation have been developed to serve as real-time state estimation models. Both filters internally reference the spacecraft relative dynamics model presented in 2.1.4.

In particular, the two models iteratively estimate the following state vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_{TS} \\ \boldsymbol{\omega}_{ST}^T \\ \mathbf{r}_{ST}^T \\ \mathbf{v}_{ST}^T \\ \mathbf{J}_T^k \end{bmatrix} \quad (4.1)$$

The first 4 components, for a total vector of dimensionality in \mathbb{R}^{13} , correspond exactly to the 4 relative attitude and relative translational dynamics components from Eq.(2.56). The last state component \mathbf{J}_T^k refer to a parametrization of the Target's inertia matrix, to possibly enable its estimation in some scenarios. In particular, as demonstrated in [20], in absence of external torques, as is the case with the considered relative dynamics model, only 2 of the 3 elements of a diagonal Target inertia matrix can be estimated. In the cited work, the following parametrization is proposed:

$$\mathbf{J}_T = \begin{bmatrix} J_{xx}/J_{yy} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & J_{zz}/J_{yy} \end{bmatrix} = \begin{bmatrix} e^{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-k_2} \end{bmatrix} \quad (4.2)$$

where J_{xx}, J_{yy}, J_{zz} are the elements on the diagonal and the two parameters, whose fixed value is:

$$\begin{aligned} k_1 &= \ln(J_{xx}/J_{yy}) \\ k_2 &= \ln(J_{yy}/J_{zz}) \end{aligned} \quad (4.3)$$

are estimated during the filtering process. The Target's inertia matrix can thus only be estimated up to a scale factor, which represents the third missing degree of freedom. The Euler equations that appear in the dynamics model, however, yield the same result regardless.

This estimated inertia matrix is used inside the integration process of the filter. Being constant, the inertia matrix parameters have no dynamics. Thus, with

$$\dot{\mathbf{j}}_T^k = \begin{bmatrix} \dot{k}_1 \\ \dot{k}_2 \end{bmatrix} = \mathbf{0} \quad (4.4)$$

the complete process model function used inside the Kalman Filters becomes:

$$\dot{\mathbf{x}} = \left. \frac{d\mathbf{c}}{dt} \right|_T = \left. \frac{d}{dt} \begin{bmatrix} \mathbf{q}_{TS} \\ \boldsymbol{\omega}_{ST}^T \\ \mathbf{r}_{ST}^T \\ \mathbf{v}_{ST}^T \\ \mathbf{J}_T^k \end{bmatrix} \right|_T = \begin{bmatrix} \frac{1}{2} \mathbf{Q}(\mathbf{q}) \boldsymbol{\omega}_{TS}^T \\ \boldsymbol{\alpha}_{ST}^T \\ \mathbf{v}_{ST}^T \\ \mathbf{a}_{ST}^T \\ \mathbf{0} \end{bmatrix} \quad (4.5)$$

for a total dimensionality in \mathbb{R}^{15} .

When dealing with a Kalman Filter working on attitude dynamics integration, dealing with the rotation representation becomes a delicate aspect. In particular, since the chosen model uses quaternions as a representation, attention must be brought into ensuring that its unitary norm constraint is kept during the estimation process. An estimated quaternion whose norm deviates from being unitary leads to incorrect rotation representation, loss of physical validity and drift in the attitude and angular velocity estimated values.

From the computational point of view, the quaternion unitary norm constraint is lost whenever the quaternion undergoes operations outside the quaternion space. That happens precisely when a quaternion gets summed or subtracted in the Cartesian space, that is by means of standard sum operations instead of multiplications. In a Kalman Filter, this is guaranteed to happen, as its fundamental update step, as discussed in Eq. (2.61) for its most general formulation, is meant to happen in Cartesian space. As the reader might trivially figure out, the amount of times the quaternions space operations are violated influences the magnitude of the normalization error accumulated. The direct consequence is that an Unscented Kalman Filter would suffer more from this with respect to the Extended Kalman Filter, and would this need a more sophisticated way of handling the issue. What in fact results from the literature is that a simple forcing of the quaternion unitary norm after each iteration is enough to prevent an EKF from being penalized. Instead, The way this has been handled in the UKF implementation follows a more advanced method, as presented in [7].

The basic idea is to use sigma points, for the attitude part of the state, built as attitude errors, instead of quaternions, and represented in another rotation representation. The propagation is still through the nonlinear model via quaternions, but the reconstruction of the predicted mean and covariance matrices is done using attitude errors

not represented as quaternions. Specifically, the chosen representation is a vector $p \in \mathbb{R}^3$ of Modified Rodrigues Parameters (MRP), defined by the transformation:

$$\mathbf{p} = f \frac{\mathbf{q}}{a + q_0} \quad (4.6)$$

to go from a quaternion $\mathbf{q} = [q_0, \mathbf{q}]$ to a MRP vector, and the transformation:

$$q_0 = \frac{-a\|\mathbf{p}\|^2 + f\sqrt{f^2 + (1 - a^2)\|\mathbf{p}\|^2}}{f^2 + \|\mathbf{p}\|^2} \quad (4.7)$$

$$\mathbf{q} = f^{-1}(a + q_0)\mathbf{p}$$

where a is a parameter, set to $a = 1$ and $f = 2 * (a + 1)$.

Going a little into the details, the UKF basically holds two sets of sigma points, as follows:

$$\chi_{k-1}^{\delta p}(i) = \begin{bmatrix} \delta p_{k-1}(i) \\ \boldsymbol{\omega}_{k-1}(i) \\ \mathbf{r}_{k-1}(i) \\ \mathbf{v}_{k-1}(i) \\ \mathbf{J}_{k-1}(i) \end{bmatrix}, \quad \chi_{k-1}^{\mathbf{q}}(i) = \begin{bmatrix} \mathbf{q}_{k-1}(i) \\ \boldsymbol{\omega}_{k-1}(i) \\ \mathbf{r}_{k-1}(i) \\ \mathbf{v}_{k-1}(i) \\ \mathbf{J}_{k-1}(i) \end{bmatrix} \quad (4.8)$$

where $\chi_{k-1}^{\mathbf{q}}(i)$ are the "natural" ones with the attitude portion represented as a quaternion and $\chi_{k-1}^{\delta p}(i)$ are with the attitude error value represented in MRP (subscripts and superscripts are avoided for clarity). The following description will only highlight the changes in the algorithm to handle the attitude partition of the sigma points, the rest of the state components remain unchanged.

At the beginning of each iteration, the UKF samples the P covariance matrix to create the set of sigma points, as shown in Eq. (2.77). In this case, those will correspond to the set of attitude errors in MRP. As shown in the same Eq. (2.77), the UKF formulation keeps the last state estimate as the 0-th sigma point, but since the two sets in (4.8) should have a direct correspondence, the error MRP sigma points corresponding to the latest state estimate needs to be equal to 0.

In particular, the quaternion error sigma points are obtained by adding in the quaternions space each error MRP sigma points converted in error quaternion to the estimated quaternion integrated at the previous filter step, as:

$$\mathbf{q}_{k-1}(i) = \delta \mathbf{q}_{k-1}(i) \oplus \hat{\mathbf{q}}_{k-1}(0) \quad (4.9)$$

where $\delta q_k(i)$ is the error MRP $\delta p_k(i)$ converted via Eq. (4.7). Following Eq. (2.78), each constructed quaternion sigma point is then propagated through the nonlinear model. Consequently, the "propagated" attitude errors are computed as the error between each propagated quaternion and the propagated 0-th quaternion from the last estimate, as:

$$\delta q_k^-(i) = \hat{q}_k^-(i) \oplus [\hat{q}_k^-(0)]^{-1} \quad (4.10)$$

Each error quaternion is converted to MRP via Eq. (4.6), creating the set of propagated sigma points $\chi_k^{\delta P}$. The direct consequence is that the "propagated" MRP error sigma point corresponding to the latest state estimate will once again be equal to 0. This set of sigma points then undergoes the standard UKF predicted mean and covariance computations in Eqs. (2.79), (2.80) and (2.84).

Since also update step should now be performed by means of MRP representation, all the UKF steps related to the measurement model have to be coherent with the MRP representation. First, the predicted measurement sigma points in Eq. (2.83) are created from the error MRP sigma points $\chi_k^{\delta P}$, and are naturally used for the steps in Eqs. (2.81) and (2.84). Secondly, the incoming measurements need to be handled accordingly as well. In particular, the quaternion error between the incoming measurement \tilde{y}_k^q and the propagated quaternion of the latest state estimate sigma point $\hat{q}_k^-(0)$ is computed. The resulting quaternion error is converted in MRP with Eq. (4.6) and goes inside the computation of the innovation term with the predicted measurement computed from Eq. (2.81), as an error value of attitude error values. If the state estimation error was exactly 0, the incoming quaternion measurement would be equal to the propagated quaternion from the previous state estimate, as would the predicted measurement. The innovation term $v_k^{\delta P}$ is in fact computed as:

$$\delta \tilde{p}_k = \delta \tilde{q}_k|_{MRP} = \left[\tilde{y}_k^q \otimes (\hat{q}_k^-(0))^{-1} \right]_{|MRP} \quad (4.11)$$

$$v_k^{\delta P} = \delta \tilde{p}_k - \hat{y}_k^{-, \delta P} \quad (4.12)$$

and then it follows the computations in the UKF update step as in (2.86).

At the end of each iteration, the post-update estimated quaternion is rebuilt as:

$$\begin{aligned} \hat{x}_k^{\delta q} &= \hat{x}_k^{\delta P}|_q \\ \hat{q}_k^+ &= \hat{x}_k^{-, \delta q} \otimes \hat{q}_k^-(0) \end{aligned} \quad (4.13)$$

where $\hat{x}_k^{\delta q}$ is post-update estimated attitude error, result of the UKF update step performed using the innovation term in Eq. (4.12), represented in quaternions.

Since the implemented UKF works, as shown, with an attitude error in Modified Rodrigues Parameters, thus dimensionality 3, all the covariance matrices P, Q and R should thus be set to have the same dimensionality 3 on the attitude component.

4.2 Real-time measurements

As mentioned in introductory chapters, both the implemented Kalman Filters have been developed to be fed by real-time measurements coming from a pose estimation model working on proximity images. Notably, the majority of in-orbit demonstrations have used conventional computer vision techniques such as edge detectors and template matching. However, the spacecraft pose estimation competitions hosted by ESA and the Stanford Space Rendezvous Laboratory yielded only machine learning based pose estimation algorithms as state-of-the-art. Companies such as Airbus and ClearSpace are planning on in-orbit demonstrations of machine learning Visual Based Navigation (VBN).

In this model's context, a simple measurement system consisting of an optical camera and a Convolutional Neural Network (CNN) pose estimation algorithm is proposed for this study. No additional sensor fusion is assumed to be performed here. In particular, simulated errors proportional to the error rate of a deployable CNN [10] are assumed.

The available measurements would thus be the relative orientation represented as a quaternion and the relative distance, yielded by the measurement model:

$$y_k = h(x_k) = \begin{bmatrix} \mathbf{q}_{TS} \\ \mathbf{r}_{ST}^T \end{bmatrix} \quad (4.14)$$

The defined relative dynamics process model directly solves for the position and orientation, which are in fact states in both the Kalman Filters. The evident result is that the measurement model of the filter is linear, which leads to improved filter robustness.

The remaining measurements needed for the integration of the process model are only the absolute states \mathbf{r}_S^S and $\boldsymbol{\omega}$ of the Servicer, which are available from its Attitude and Orbit Determination System. If instead the representation with respect to the Servicer was chosen, the Filters would require knowledge of the absolute states of the Target, which are not available since it is assumed uncooperative.

The environment in which this work has been developed is a simulated environment that individually integrates torque-free attitude dynamics and keplerian orbits of the Servicer and of the Target. The real-time measurements coming from the pose estimation model are thus simulated as the relative attitude and distance states of the Target and the Servicer with added noise, fed at a rate fixed to the defined sampling rate of the Kalman Filter.

The quaternion noise is modeled starting from a rotation vector of random amplitude α_n , Gaussian distributed with zero mean and variance $\sigma_y^q = 0.12rad$, and random axis \mathbf{r}_n uniformly distributed on the unitary sphere. The corresponding noisy quaternion measurement is thus computed as:

$$\mathbf{q}_n = \mathbf{q} \otimes \delta \mathbf{q}_n \quad (4.15)$$

where \mathbf{q} is the ground truth relative attitude and $\delta \mathbf{q}_n$ is computed from the random rotation vector as:

$$\delta \mathbf{q} = \left[\cos\left(\frac{\alpha_n}{2}\right) \quad \mathbf{r}_n \sin\left(\frac{\alpha_n}{2}\right) \right] \quad (4.16)$$

The variance used is inferred from real pose estimation model results. This noise modeling allows to truly represent a random quaternion of Gaussian distributed angle with given variance. A more simple and intuitive random 4 dimensional vector generator would have no room for specifying a known variance with a true physical meaning.

The relative distance noise is instead modeled as a simple 3 dimensional random vector whose norm follows a zero-mean Gaussian distribution with variance $\sigma_y^r = 0.37m$, inferred again from real pose estimation model results.

The outlier rejection method outlined in Section 2.2.4 is implemented. In particular, as will be explained thoroughly in the next sections, the two individual measurement source might fail independently and if no measurement source is available the sole predicted state value is kept. Faulty measurements are modeled as

extremely noisy values, to reflect the unstable behavior that can characterize a pose estimation model during fault windows.

In conclusion, in all the experiments carried out the Target object is assumed to have the shape of a rocket body, a cylinder with an elongated dimension on the y axis (ex. Atlas Centaur R/B). As shown in Eq. (2.6), since the mass is closer distributed to y axis, the relative inertia matrix component is the smallest. The inertia matrix used, assumed diagonal, represented according to the parametrization in (4.2), is equal to:

$$J_T = \begin{bmatrix} 4.33 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix} \quad (4.17)$$

corresponding to $k_1 = 1.466, k_2 = -1.439$.

The simulation length T_{sim} and the sampling time τ_s are defined before each case study. The state estimate initial condition for the Kalman Filter \hat{x}_0 is set to zero unless specific otherwise. The inertia matrix component of the initial state estimate $\hat{J}_{T,0}$ is always set to 80% of the true Target inertia matrix ground truth, to reflect assumed knowledge on the rough shape of the object (around 2 or 3 times more elongated along the y axis) but without precise values on the exact mass distribution ($\hat{J}_{x,0} = 3.16$ vs $J_{x,0} = 4.33$).

4.3 Software framework

The principal simulation environment has been designed in MATLAB/Simulink version R2019b. All the integrations are performed using the *ode4* integration method with fixed simulation sampling time of 0.01 seconds. The integrations inside the Kalman Filter exploit the same *ode4* integration method with the fixed filter sampling time defined before each case study.

Chapter 5

Kinematic priors injection studies

This section describes the experiments conducted on the model outlined in the previous chapters. Different ideas for injecting the kinematic priors extracted from light curves with the assumptions from Chapter 1 are explored. The results are analyzed by means of convergence metrics, which will be defined first. After an overview of the studies conducted, each one is described in detail.

The performance of the Kalman Filters is evaluated with measures of convergence time and Root Mean Squared Error (RMSE) at steady-state, i.e. after convergence is reached.

In this context, convergence can have a dual interpretation. The uncertainty that the filter associates to each component of the state estimate is embedded in the P matrix, and is decreased as part of the estimation process. Convergence from the filter's perspective is reached once the P matrix, for each one of its individual components paired with the state components in Eq. (4.1), has sufficiently reduced its oscillations. The P matrix convergence, t_{conv}^P and $RMSE^P$, is verified by setting a threshold on the variation of the mean of the P matrix diagonal components, per each state component.

The actual values of the state, however might still exhibit oscillations, still needing for new measurement updates to reduce the error with respect to the ground truth. The process has then converged by state value x perspective, t_{conv}^x and $RMSE^x$, once the error with respect to the ground truth goes below a certain threshold that is never exceeded again. The two convergence metrics thus report a different "meaning", and are both needed at the same time.

Convergence metric by filter P matrix is used to compare the performance between different filters (different Q_d and R_d parameter sets are considered as different filters) regardless of the actual state value those converge to. Consider the following example: two filters are tuned specifically one for a high frequency system (FH), for example a body with a high angular velocity, and one for a low frequency system (FL), a body with a low angular velocity. The Filter FH will likely never convergence to the same state value that can be reached by FL, as estimation on a faster system expects oscillations of higher amplitude. At the same time it will likely reach its lowest state estimate error value faster than FL, since allowing for more oscillations leads to faster exploration of the solution space and thus a faster convergence.

Measuring this performance difference by looking at the state value x convergence only would require to provide each convergence time measure in couple with the state value threshold it has been calculated on. The filter P convergence metric, instead, together with the associated RMSE, encapsulates this univocally, and is thus suitable for different filters comparison. The state value x convergence metric is yet still useful for defining the performance of a single Filter, as the time at which a precise state value is met is not exactly represented behind the filter P metric.

The state value x convergence metrics are computed on the state error with respect to the ground truth. The state error computes the difference between the L2-norm of the state and the L2-norm of the ground truth, per individual state component. The filter P metrics are computed on the L2-norm of each P matrix component. The resulting convergence metrics, considering the state vector in Eq. 4.1, are thus vectors in \mathbb{R}^5 .

A visualization of the difference between the two convergence perspectives can be visualized in Fig. 5.1

The initial estimated inertia parameters are set to 80% with respect to the ground truth (as discussed in 4.2), and the rest of the initial estimate vector \hat{x}_0 is set to zero.

The three studies conducted differ from the way the kinematic prior is injected inside the model and thus the effect that is obtained. In particular:

- Study 1 - Kalman Filter parameter tuning

The kinematic prior is injected as inferred knowledge on the system. This

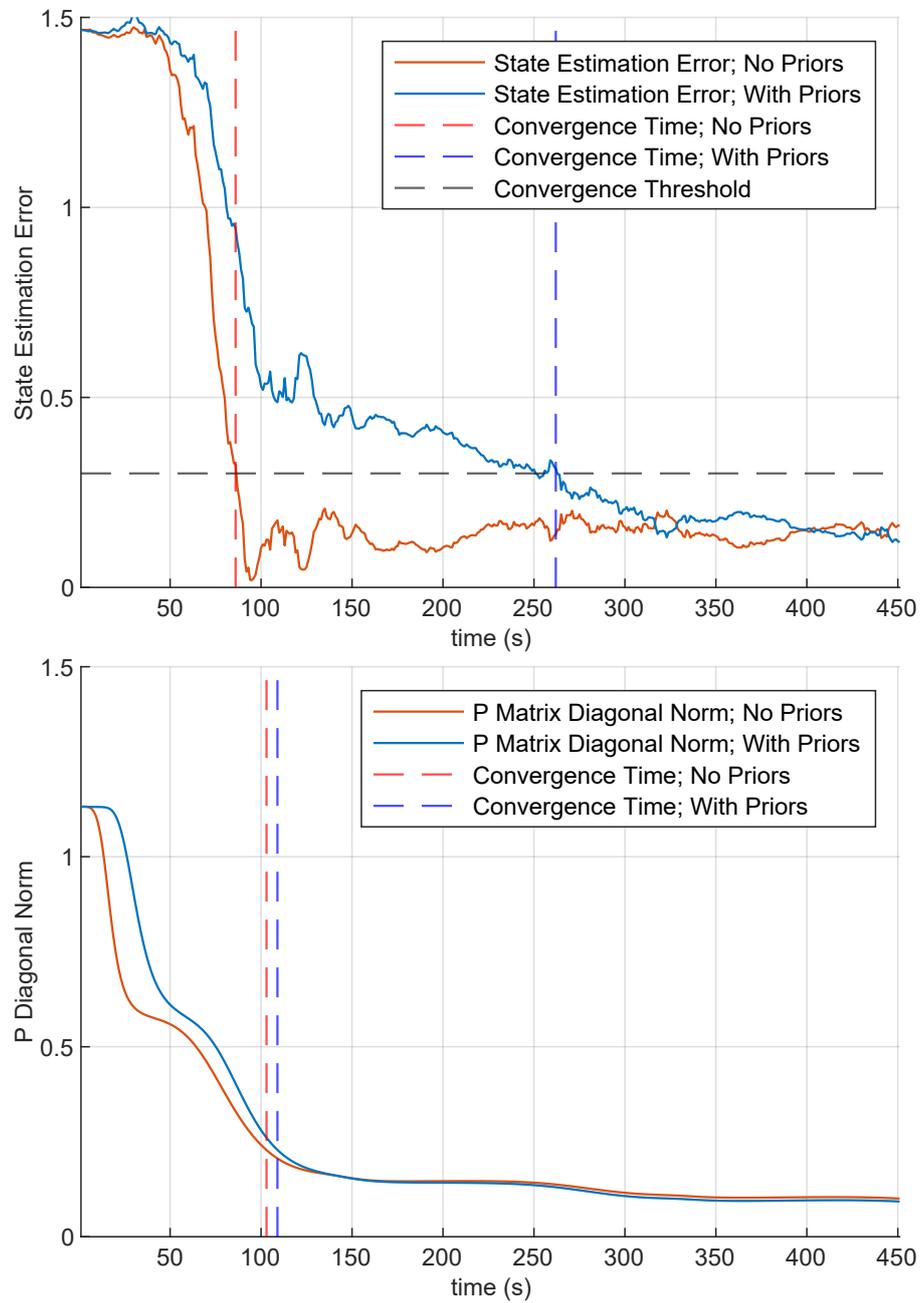


Fig. 5.1 Convergence metrics visualized

information can enable a scenario-aware parameter tuning, which performs better than a parameter set based on assumptions.

- Study 2 - Inertia matrix estimation improvement
The kinematic prior injected as an initial condition for the Kalman Filter. Setting a confidence of the initial angular velocity value by means of P_0 allows to shift the energy of the estimation process towards the inertia matrix component, that can converge faster.
- Study 3 - Enhanced outlier rejection
The kinematic prior injected as fictitious measurement during a pose fault. The kinematic prior is also injected as initial condition for the Kalman Filters at the same time. The additional measurement source gives the filter something to rely on to mitigate the divergence.

5.1 Kalman Filter parameter tuning

In this experiment, we argue that availability of information on the order of magnitude of the speed and on the tumbling axes of the Target can be beneficial for the parameters tuning process of the Kalman Filters. Four Target tumbling scenarios are defined, and different Kalman Filer Q_d parameter tunings are extracted for each scenario. A performance comparison between each fine-tuned parameter set and a general scenario-unaware parameter set is then discussed.

The four scenarios shown in Tab. 5.1 are defined. Each scenario sets a different Target initial angular velocity for its attitude dynamics integration, defining the ground truth state profile to be estimated. Values are chosen to span slow and fast

Table 5.1 Scenarios definition, respectively: *1 axis slow/fast, 2-axes slow/fast*

$\omega[\text{rad}/s]$	SC1	SC2	SC3	SC4
$\omega_{T,0}^x$	0	0	0.020	0.100
$\omega_{T,0}^y$	0	0	0	0
$\omega_{T,0}^z$	0.050	0.200	0.045	0.150

tumbling scenarios, with half an order of magnitude offsets, about 1 or 2 body axes.

The combination of the defined scenarios and the specified sampling rate are overall rather challenging, and are meant find possible improvements when the Kalman Filters are working under stressing conditions. Slower tumbling rates are generally less realistic, more trivial for the Kalman Filters, and in any case would not need further tuning with respect to the slow scenarios considered here (SC1 and SC3). Faster rates with the same relatively limited sampling time and measurement noise can lead to instabilities in the state estimation model. It can also be argued that designing proximity missions with uncooperative targets with such kind of complex tumbling motion would be less realistic anyhow.

The Q_d matrix tuning process has been carried out using iterative simulations. Given a pre-defined feasible range of potential Q_d matrix weights, the set of all the possible combinations of such values is created. An iterative optimization algorithm to find the best performing parameter set is then run. The objective function defined is sum of the per state component by filter convergence times and per state component relative by filter RMSE values at steady-state, as:

$$g(x) = \lambda_t t_{conv}^P + \lambda_{RMSE} RMSE^P \quad (5.1)$$

where each state component contribute is appropriately weighted, by means of $\lambda_t, \lambda_{RMSE} \in \mathbb{R}^5$, to match a common order of magnitude of 1. Since the goal is to compare different Filter implementations, the filter P matrix convergence metric, $t_{conv}^P, RMSE^P$, is chosen for the study.

The concept of best performing parameter set for each scenario, according to the objective function, is now defined, Table 5.2 shows the results. Matrix Q_d , as already mentioned, is defined diagonal. For each state component, each value shown weights equally the whole dimensionality of that given component. For example, the quaternion component for the EKF is weighted $10^{-z} \cdot \mathbf{1}_4$.

This study aims at demonstrating the effect of the Q_d matrix tuning on the performance of the Kalman Filters. Parameter tuning of a state estimation model is crucial to obtain the best possible results out of the estimation process, as already mentioned in Section 2.2. Parameter tuning is mainly referred to Q_d and R_d diagonal matrices values adjustment. The R_d matrix is modeled on the performance of the measurement devices, which, being under control, simplifies the operation. The Q_d matrix tuning, while still slightly conditioned by aspects under control, as the Filter's sampling time or the known assumptions made for the simplified process model used

Table 5.2 Optimal parameters for each scenario

	SC 1		SC 2		SC 3		SC 4	
	UKF	EKF	UKF	EKF	UKF	EKF	UKF	EKF
\mathbf{q}_{TS}	10^{-5}	10^{-7}	10^{-5}	10^{-5}	10^{-5}	10^{-6}	10^{-4}	10^{-6}
$\boldsymbol{\omega}_{ST}$	10^{-7}	10^{-7}	10^{-7}	10^{-6}	10^{-7}	10^{-6}	10^{-5}	10^{-5}
\mathbf{r}_{ST}	10^{-5}	10^{-7}	10^{-5}	10^{-7}	10^{-7}	10^{-6}	10^{-6}	10^{-6}
\mathbf{v}_{ST}	10^{-6}	10^{-5}	10^{-5}	10^{-5}	10^{-6}	10^{-6}	10^{-5}	10^{-4}
\mathbf{J}_{Th}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-2}	$5 \cdot 10^{-4}$	10^{-3}	10^{-4}

inside the Kalman Filter, is instead mainly dependent on the real physical behavior of the system. This is the reason why, in case no prior knowledge on the system is available in that sense, assumptions and trade off choices have to be made.

Scenario 4 represents a fast changing complex tumbling, as the rotation happens on 2 axes with a high angular rate. Conceptually, the best parameter tuning for such a scenario would need to allow for more oscillations, thus high Q_d values to let the filter adapt more quickly to the fast changing environment, and this is fact what results from the tuning process. Scenario 4 Q_d matrix is set to mimic a fixed assumption based parameter set. For each defined scenario, a simulation with the tuned scenario-aware parameters from Table 5.2 is run and compared. The duration of the simulations is $T_{sim} = 300$ seconds, with a sampling time $\tau_s = 0.5s$, corresponding to a pose estimation model acquiring images at 2 Frames Per Second (FPS) rate.

The comparison tables show, separately for the UKF and for the EKF, the metric value obtained with the scenario-aware parameters versus the metric value corresponding to the general assumption-based parameters between rounded brackets.

First, the results of this study are a good chance to visualize for the first time the performance difference between the two filters implemented. The general consideration is that the UKF performs overall worse than the EKF in almost all the considered scenarios. This could be explained by the fact that the UKF is in general meant to better exploit the incoming information from the measurements, if such information is available. The little amount of sources of measurements, the relative attitude and the relative distance only, and the few samples available for each measurement source, due to the relatively high sampling time, define a rather challenging scenario.

Table 5.3 UKF filter convergence times comparison. Format: scenario-aware parameters (assumption-based SC 4 parameters)

	SC1	SC2	SC3	SC4
\mathbf{q}_{TS}	73 (22.0)	36 (34.0)	35 (66.5)	- (41.5)
ω_{ST}	4.0 (4.0)	5.5 (4.5)	4.0 (4.0)	- (4.5)
\mathbf{r}_{ST}	6.0 (3.5)	6.0 (3.5)	9.5 (3.5)	- (3.5)
\mathbf{v}_{ST}	5.5 (4.0)	5.5 (4.0)	7.5 (4.0)	- (4.5)
\mathbf{J}_T	- (N/D)	- (N/D)	281.0 (108.5)	- (30.0)

Table 5.4 UKF filter convergence RMSE at steady-state comparison. Format: scenario-aware parameters (assumption-based SC 4 parameters)

	SC1	SC2	SC3	SC4
\mathbf{q}_{TS}	0.7228 (1.1234)	0.7705 (0.9957)	1.3926 (1.1982)	- (2.0174)
ω_{ST}	0.0031 (0.0035)	0.0047 (0.0054)	0.0046 (0.0043)	- (0.0132)
\mathbf{r}_{ST}	0.3233 (0.4152)	0.3456 (0.4218)	0.2833 (0.4157)	- (0.4244)
\mathbf{v}_{ST}	0.1318 (0.2818)	0.1780 (0.3032)	0.0905 (0.2820)	- (0.3164)
\mathbf{J}_T	- (N/D)	- (N/D)	1.7410 (1.7372)	- (1.3876)

Table 5.5 EKF state convergence times comparison. Format: scenario-aware parameters (assumption-based SC 4 parameters)

	SC 1	SC 2	SC 3	SC 4
\mathbf{q}_{TS}	56.5 (20.0)	84.5 (114.5)	37.0 (38.5)	- (17.5)
ω_{ST}	7.0 (7.0)	7.5 (7.5)	7.0 (7.0)	- (12.0)
\mathbf{r}_{ST}	10.5 (5.5)	6.0 (5.5)	10.5 (5.5)	- (5.5)
\mathbf{v}_{ST}	12.0 (9.0)	11.5 (8.5)	11.5 (9.0)	- (9.5)
\mathbf{J}_T	- (N/D)	- (N/D)	270 (224.5)	- (69.0)

Table 5.6 EKF state RMSE at steady-state comparison. Format: scenario-aware parameters (assumption-based SC 4 parameters)

	SC 1	SC 2	SC 3	SC 4
\mathbf{q}_{TS}	0.5170 (0.9437)	0.6190 (0.9432)	0.9602 (1.0321)	- (1.4355)
$\boldsymbol{\omega}_{ST}$	0.0017 (0.0020)	0.0028 (0.0030)	0.0023 (0.0022)	- (0.0059)
\mathbf{r}_{ST}	0.2596 (0.3228)	0.3310 (0.3303)	0.2515 (0.3244)	- (0.3555)
\mathbf{v}_{ST}	0.0573 (0.1259)	0.4587 (0.4599)	0.0642 (0.1265)	- (0.4477)
\mathbf{J}_T	- (N/D)	- (N/D)	0.1252 (1.2613)	- (0.3340)

This aspect, combined with the fact that the spacecraft relative dynamics problem is not a system with high nonlinearities, makes the local linearity assumption of the EKF work better than the involved sigma points nonlinear propagation of the UKF. Despite this, the quaternion error value almost in all cases is kept below 2 degrees, which can be considered a maximum acceptable error.

The main result is that using the assumption-based general parameters (Scenario 4 used for the purpose) on a scenario which is instead slower, with fewer oscillations, thus less challenging (as for example Scenario 1), results in a faster convergence time with respect to the scenario-aware performance, as shown in the first column of Tables 5.5 and 5.3. As already presented, this is to be expected, since allowing more oscillations lets a filter converge faster.

At the same time, a cost is paid on the RMSE at steady-state, which for every set of scenario-aware parameters is in fact lower. The faster the real system is, moving from Scenario 1 towards Scenario 4, the lower the difference in the convergence times becomes, and the difference in RMSE at steady-state, while more relevant than the convergence time, becomes less significant. This possibly reflects a real scenario: the more your assumptions are close to the real behavior of the system, which in this analysis means the more the considered scenario is close to fixed scenario 4, the less the performance difference between the assumption-based parameters and the scenario-aware ones is evident.

The choice of considering Scenario 4 as the fixed scenario for comparisons is done to mimic a possible assumption that would be defined in a real use-case. However, it is important to point out that any choice of a given parameter set can be made, and the cross comparisons still hold. Regardless of the choice made, that

parameter set will never meet the performance of a single scenario-aware one, which can be instead achieved leveraging kinematic priors.

Another relevant result is that the performance of the inertia matrix component estimation appears very sensible to the parameter tuning. On the same regard, another outcome of the study highlights how inertia matrix estimation is enabled only in two axes tumbling scenarios, as in 1 axes rotations no inertia matrix coupling effect arise, and the angular velocity remains in fact constant.

For this reason, in the presented tables, on the inertia matrix component for 1 axes tumbling scenarios, the Not Defined N/D label represents no estimation occurring. It corresponds to an error value equal to the error in the initial condition kept constant during the estimation, but its interpretation is as Non Defined.

It could be argued that in a real mission one would design an adaptive algorithm for Q_d and possibly R_d matrix real-time tuning, in order to be more conservative. What is also true is that adaptive algorithms are highly computationally intensive and real missions implementations tend to choose a trade off shifted towards being more risk-adverse, rather than conservative but with high computational and thus energetic requirements. What in any case still holds is that the knowledge on the scenario provided by the kinematic priors would enable a safe implementation with standard non-adaptive computational requirements.

5.2 Inertia matrix estimation improvement

As results from the previous study, without a possible additional direct measurement of the angular acceleration, inertia matrix estimation is enabled only in two axis scenarios. For this study, the simulation is thus set to scenario 3 from Table 5.1, which corresponds to a two axis tumbling with slow angular velocity. Simulations are run for $T_{sim} = 450$ seconds with sampling time $\tau_s = 1$ second, corresponding to a pose estimation model acquiring images at 1 Frames Per Second (FPS) rate.

The kinematic prior is injected as an initial condition for $\hat{\omega}_0$, angular velocity component of the initial state estimate \hat{x}_0 . At the same time the values of the initial diagonal P_0 matrix related to the angular velocity component are thus set to a lower value, to reflect the trust given to this initial knowledge.

Table 5.7 State value convergence times

	UKF	+ priors	EKF	+ priors
\mathbf{q}_{TS}	451	451	39	90
ω_{ST}	60	0	32	0
\mathbf{r}_{ST}	176	176	176	176
\mathbf{v}_{ST}	119	119	266	266
\mathbf{J}_{Th}	451	451	262	86

Table 5.8 State value convergence RMSE

	UKF	+ priors	EKF	+ priors
\mathbf{q}_{TS}	N/D	N/D	1.4659	1.2883
ω_{ST}	0.0038	0.0042	0.0014	0.0016
\mathbf{r}_{ST}	0.3916	0.3916	0.3907	0.3909
\mathbf{v}_{ST}	0.1012	0.0994	0.1076	0.1073
\mathbf{J}_{Th}	N/D	N/D	0.1838	0.1449

The results show how the trust in the kinematic prior initial condition is able to shift the energy of the estimation process towards the Target inertia matrix component. This reflects, both in the UKF and in the EKF, to a performance increase in the inertia matrix estimation.

Looking at the filter convergence results on the inertia matrix component, Table 5.9, the values show how the two filters, without and with kinematic priors, exhibit estimation convergence almost at the same time instant. However, the steady-state squared error at which the two converge, Table 5.10, is higher for the filter without priors. A different look on the same results is seen on the inertia matrix state component on Tables 5.7 and 5.8. The state value convergence, for practically the same RMSE, occurs almost 200 seconds earlier for the filter that exploits kinematic priors.

As expected, the "cost" of such performance increase on the inertia matrix state component is paid by a decrease in the performance on the relative quaternion component estimation. The result of shifting energy towards the inertia matrix estimation is having a filter that ends the quaternion component estimation earlier (in the filter convergence sense), leaving a high error on the state estimate, that needs

Table 5.9 Filter convergence times

	UKF	+ priors	EKF	+ priors
\mathbf{q}_{TS}	101	85	91	37
$\boldsymbol{\omega}_{ST}$	74	70	49	30
\mathbf{r}_{ST}	10	8	12	12
\mathbf{v}_{ST}	9	7	13	13
\mathbf{J}_{Th}	121	115	109	103

Table 5.10 Filter convergence RMSEs

	UKF	+ priors	EKF	+ priors
\mathbf{q}_{TS}	2.9499	2.5837	1.2749	1.9489
$\boldsymbol{\omega}_{ST}$	0.0037	0.0031	0.0012	0.0014
\mathbf{r}_{ST}	0.4011	0.3969	0.3950	0.3952
\mathbf{v}_{ST}	0.1186	0.1012	0.1158	0.1150
\mathbf{J}_{Th}	1.1810	0.7013	0.3102	0.1458

then more measurement corrections to adapt. For the quaternion component, the results show in fact a higher filter convergence RMSE (Table 5.10) and a higher state value convergence time for the filter with priors, for the same state value RMSE (Tables 5.7, 5.8).

Moreover, the angular velocity component $\boldsymbol{\omega}_{ST}$ on the filter working with priors sees state value convergence time $t_{conv} = 0$, as the forced initial value of the angular velocity is already below the threshold defined for state value convergence. The quaternion component estimation for the UKF never achieves state value convergence, since it is kept too high and never definitively crosses the threshold of 2 degrees. The same result is evident from the filter convergence RMSE of the quaternion component 5.10. The same happens for the inertia matrix component for the UKF. An overall conclusion is that the UKF is slightly more unstable and some of its results do not exactly follow the expected trend obtained with the EKF.

As a final result, the kinematic priors injection impact on the relative distance and relative velocity state components is irrelevant.

The discussed results also have a visual representation in Figs. 5.3 and 5.2.

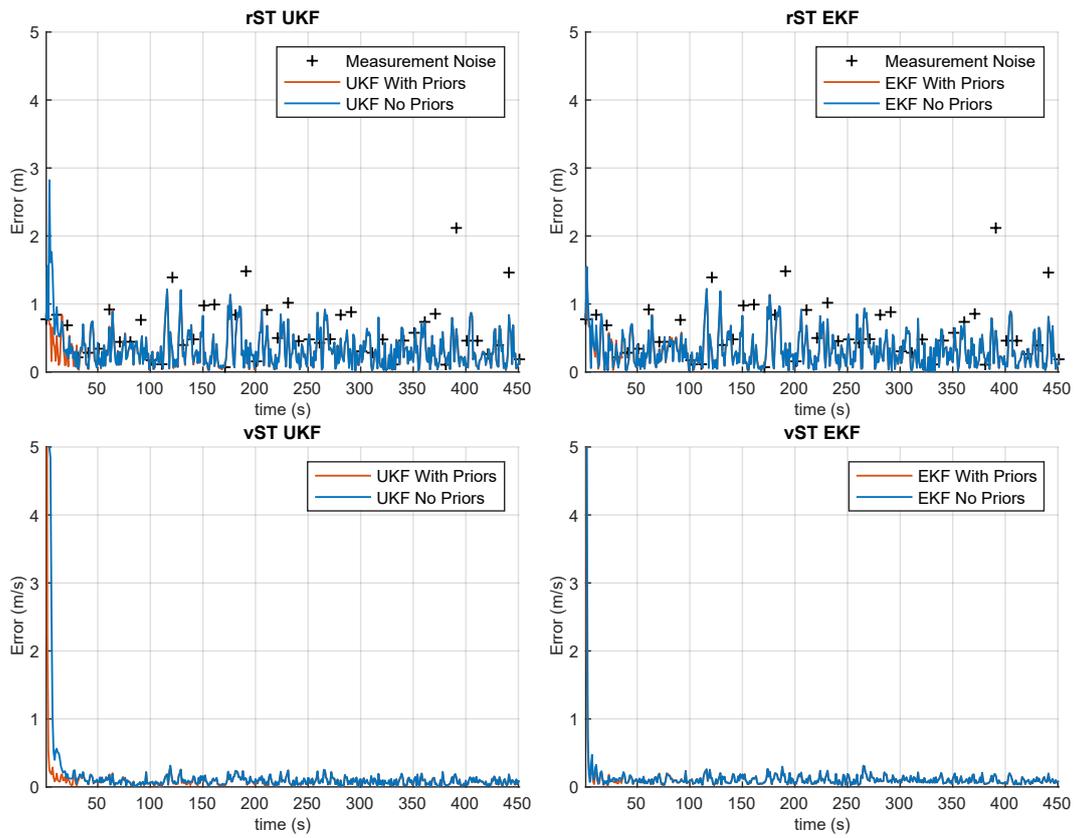


Fig. 5.2 Relative distance \mathbf{r}_{ST}^T , relative speed \mathbf{v}_{ST}^T state estimation errors

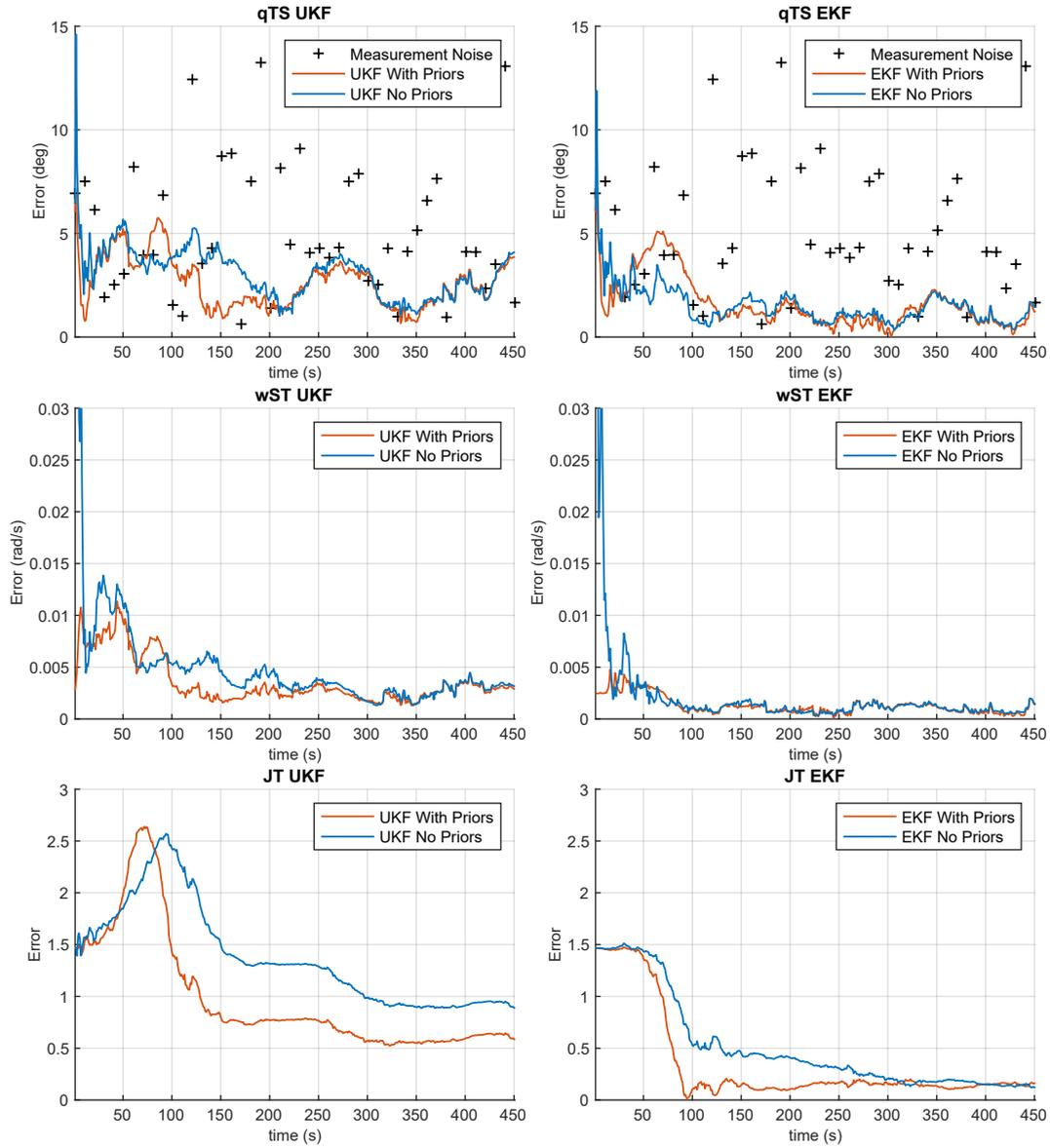


Fig. 5.3 Relative attitude \mathbf{q}_{TS} , relative angular velocity ω_{ST}^T and target inertia matrix parameters $k_1^{J^T}, k_2^{J^T}$ state estimation errors

5.3 Enhanced outlier rejection

The idea behind this study is to find a possible performance improvement by injecting the kinematic priors as an additional measurement source during faults in the real-time measurements. Also, the kinematic prior is provided as initial condition to the state estimation, as in Study 2, with the P_0 value set up accordingly.

As explained in Section 2.2.4, adding the support for dealing with individual sources of measurements failing requires defining individual execution paths triggered by the faults. For this study, each possible individual execution path (quaternion fault, relative distance fault, fault of both) is enhanced with an additional source of measurement, represented by the kinematic prior being "pushed" at each time step as a fixed fictitious measurement. Moreover, since each measurement needs to be weighted by a component on the R_d measurement noise matrix, an additional component $R_\omega = 10^{-3} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ is defined to weight the forced kinematic prior measurement inside the Kalman Filter computations.

The implementation of this work, as of now, supports kinematic prior measurement injection as a constant value, thus applicable only for 1 axis rotation. Multiple axis tumbling support would require accounting for phase during the injection. The simulated scenario is thus Scenario 1 from 5.1.

Table 5.11 compares a simulation without kinematic priors injection, thus only non-enhanced outlier rejection, and a simulation with the kinematic priors used as initial condition for the state estimate and used as a measurement during the pose fault. A fault window of 120 seconds of duration is simulated, with both relative quaternion and relative distance measurements failing at the same time. The results are analyzed by means of RMSE during the pose fault window, including 30 seconds of settling time after the fault is finished, to account for re-acquisition of convergence. Moreover, as results from 5.1, the Target inertia matrix component estimation is not enabled, thus not shown.

As expected, all the results exhibit diverging behavior. It is however significantly less impactful that it would be without the outlier rejection method, regardless if it is enhanced or not.

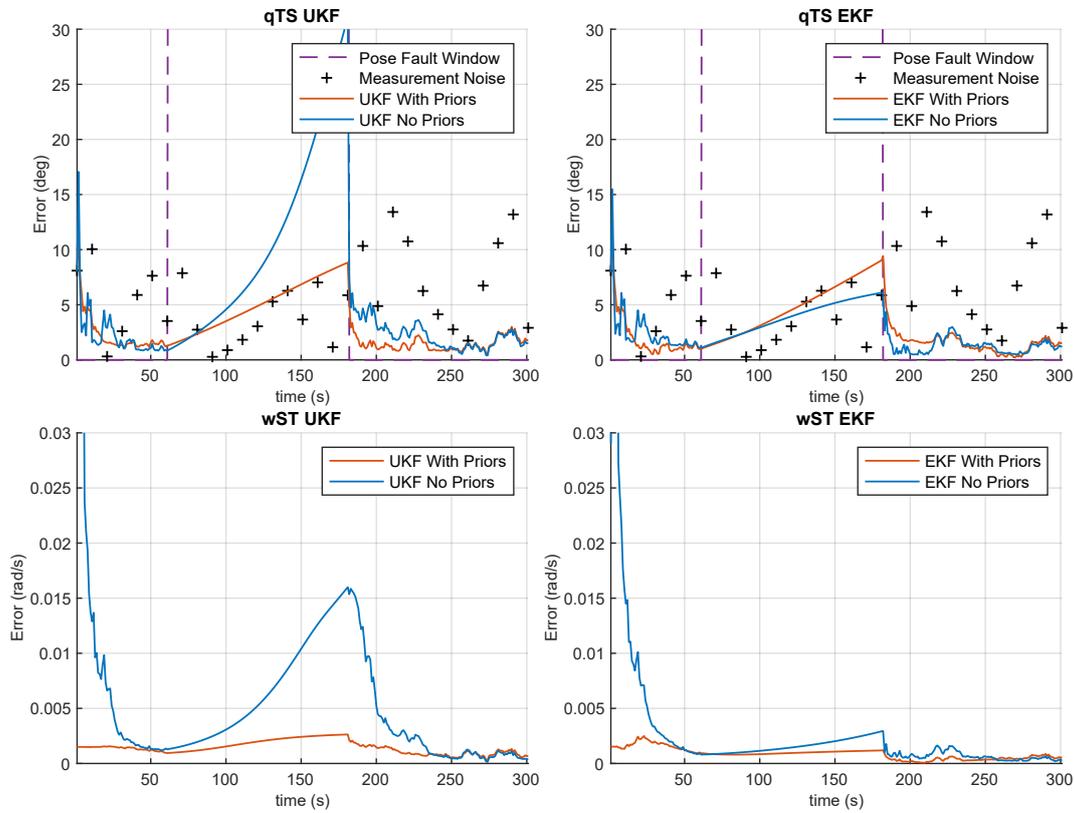


Fig. 5.4 Relative attitude \mathbf{q}_{TS} , relative angular velocity ω_{ST}^T state estimation errors

The effect of the enhancing the outlier rejection technique is very evident in the results of the angular velocity state component ω_{ST} , both of the EKF and the UKF, as its error is kept constant to the 5% assumed intrinsic error in the kinematic prior. Moreover, the table shows how the method is able to significantly prevent the quaternion state component \mathbf{q}_{TS} for the UKF from diverging. On the other hand, the EKF instead experiences a slightly slower divergence without the kinematic prior measurement injection.

Relative distance measurements faults raise a high offset error in the relative distance and speed estimation, without a significant diverging behavior for the time window considered. What is in fact also clear is that the kinematic prior injection has a sufficiently relevant impact only in the attitude and angular velocity components of the state. Even if the relative distance and speed estimates are coupled with the Target's angular velocity value, the diverging behavior of the latter, or its

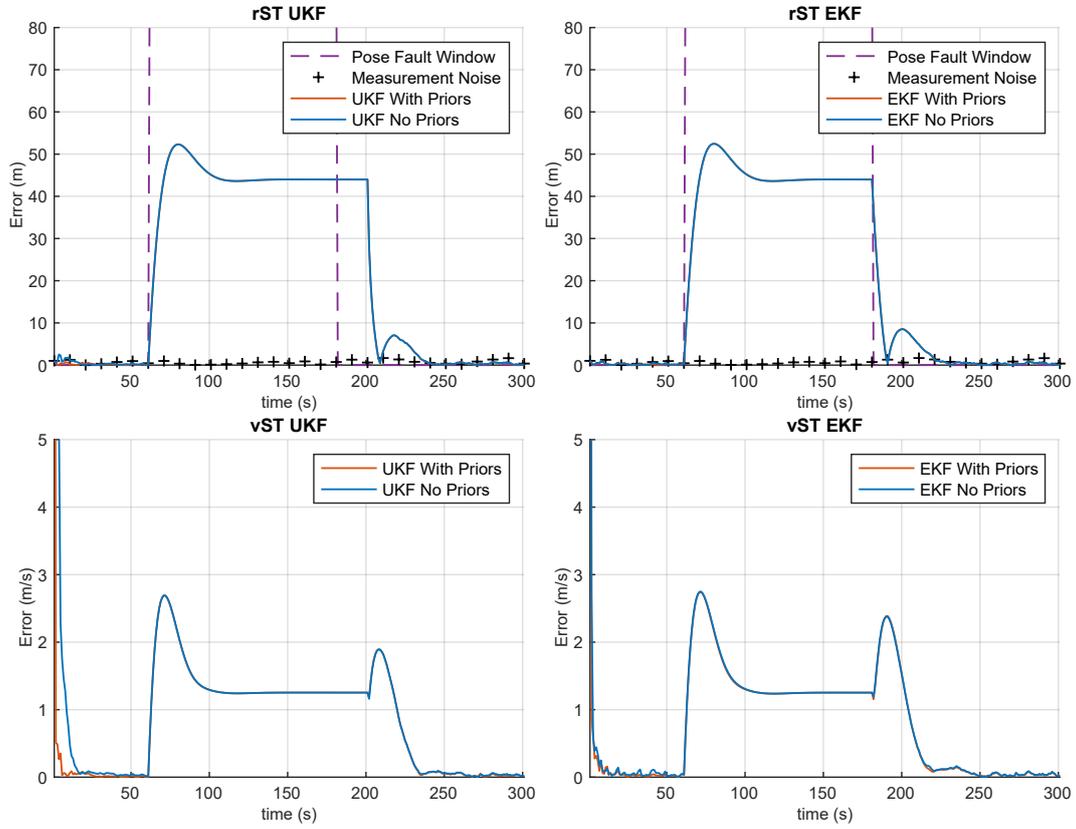


Fig. 5.5 Relative distance \mathbf{r}_{ST}^T , relative speed \mathbf{v}_{ST}^T state estimation errors

Table 5.11 State value RMSE during fault window and following 30 seconds of settling time.

	UKF	+ priors	EKF	+ priors
\mathbf{q}_{TS}	12.5006	4.9162	3.7476	4.8795
$\boldsymbol{\omega}_{ST}$	0.0085	0.0019	0.0016	0.0009
\mathbf{r}_{ST}	42.6851	42.6898	39.8952	39.8946
\mathbf{v}_{ST}	1.4836	1.4838	1.5518	1.5484

improvement with the enhanced outlier rejection, is not impactful on the first. The orders of magnitude difference in the values of ω_{ST} and r_{ST} could also be affecting the coupled impact.

A visual analysis can also be carried out by looking at Figs. 5.4 and 5.5.

Chapter 6

Conclusions

This work explored the enhancement of uncooperative spacecraft kinematic state estimation by leveraging kinematic priors extracted from light curves.

The simulation and state estimator's relative spacecraft dynamics model were first introduced. Two Kalman filter models enforcing the quaternion unitary norm constraint were then implemented; an Extended Kalman Filter and an Unscented Kalman Filter. A discussion on light curve analysis justified the source and fidelity of the kinematic motion priors. Kinematic motion prior extraction was demonstrated on synthetic and commercial 1-axis tumbling data. Three orthogonal studies were performed to demonstrate the impact of injecting information from kinematic priors into the Kalman filter.

By exploiting kinematic priors, convergence time and steady state error reductions of $3\times$ or more are exhibited for certain state components, dependent on the kinematic scenario and filter tuning. Overall, the EKF outperformed the UKF. Notably, the UKF was quite sensitive to tuning and also unstable in challenging low sample rate conditions. If additional model nonlinearities (e.g., orbit perturbations, kinematic damping or atmosphere-induced torque) were considered, the environment would become intrinsically more challenging. Under such conditions, an UKF could conceivably outperform an EKF due to its improved handling of high nonlinearities.

6.1 Future work

Future efforts should be focused on easing the current assumptions.

First, model validation on higher precision synthetic data could be pursued. To this extent, orbital perturbations (e.g., Earth oblateness J_2 , third-body etc.) and dissipation torques (e.g., LEO atmospheric drag) in the attitude dynamics can be incorporated. Additionally, a Servicer's thrust model could be added to simulate realistic rendezvous operations.

Second, improved support for 2-axes tumbling scenarios could be developed. State-of-the-art light curve analysis and inversion techniques would enable more accurate motion prior extraction characterization for complex body tumbling around non-trivial axes. Consequently, the enhanced outlier rejection method in 5.3 could be extended for supporting multi-axes angular velocity injection.

References

- [1] Classical Orbital Elements — Orbital Mechanics & Astrodynamics.
- [2] Quaternions - Visualisation.
- [3] ESA - The Zero Debris Charter, June 2023.
- [4] Laurence Blacketer. *Attitude characterisation of space objects using optical light curves*. phd, University of Southampton, March 2022.
- [5] Vincenzo Capuano, Kyunam Kim, Juliette Hu, Alexei Harvard, and Soon-Jo Chung. Monocular-Based Pose Determination of Uncooperative Known and Unknown Space Objects. *th International Astronautical Congress*, 2018.
- [6] R. L. Cook and K. E. Torrance. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982.
- [7] John L. Crassidis and F. Landis Markley. Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, July 2003.
- [8] Jason L. Forshaw, Guglielmo S. Aglietti, Thierry Salmon, Ingo Retat, Mark Roe, Christopher Burgess, Thomas Chabot, Aurélien Pisseloup, Andy Phipps, Cesar Bernal, François Chaumette, Alexandre Pollini, and Willem H. Steyn. Final payload test results for the removedebris active debris removal mission. *Acta Astronautica*, 138:326–342, 2017. The Fifth International Conference on Tethers in Space.
- [9] Pini Gurfil and Konstantin V. Kholshchevnikov. Manifolds and Metrics in the Relative Spacecraft Motion Problem. *Journal of Guidance, Control, and Dynamics*, 29(4):1004–1010, July 2006.
- [10] Yinlin Hu, Sebastien Speierer, Wenzel Jakob, Pascal Fua, and Mathieu Salzmann. Wide-Depth-Range 6D Object Pose Estimation in Space. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15865–15874, Nashville, TN, USA, June 2021. IEEE.
- [11] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193. SPIE, July 1997.

-
- [12] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.
- [13] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, March 1961.
- [14] Du Yong Kim, Sang-Goog Lee, and Moongu Jeon. Outlier Rejection Methods for Robust Kalman Filtering. In James J. Park, Laurence T. Yang, and Changhoon Lee, editors, *Future Information Technology*, pages 316–322, Berlin, Heidelberg, 2011. Springer.
- [15] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. Kalman Filter and Its Application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 74–77, Tianjin, China, November 2015. IEEE.
- [16] B. Mueller. The diagnosis of complex rotation in the lightcurve of 4179 toutatis and potential applications to other asteroids and bare cometary nuclei. *Icarus*, 158(2):305–311, 2002.
- [17] ESA Space Debris Office. ESA Space Environment Report 2024, July 2024.
- [18] Gregory Plett. ECE5550: Applied Kalman Filtering.
- [19] Brent E. Tweddle and Alvar Saenz-Otero. Relative Computer Vision-Based Navigation for Small Inspection Spacecraft. *Journal of Guidance, Control, and Dynamics*, 38(5):969–978, May 2015.
- [20] Brent Edward Tweddle. *Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations*. Thesis, Massachusetts Institute of Technology, 2013.
- [21] Yaguang Yang. *Spacecraft Modeling, Attitude Determination, and Control Quaternion-based Approach*. CRC Press, Boca Raton, FL : CRC Press, 2019. | “A science publishers book.”, 1 edition, February 2019.
- [22] J. Šilha. *Space Debris: Optical Measurements*. 2020.