



Full State Pose Estimation Using a Satellite Imager

by

D.P. Theron

*Dissertation presented for the degree of Masters of
(Electronic Engineering) in the Faculty of Engineering at
Stellenbosch University*

Supervisor: Prof. H.W. Jordaan
Dr. S. Busch

2025

To God, my Wife and my Mother

ACKNOWLEDGEMENTS

- * Prof. H.W. Jordaan, thank you for all your patience and guidance through this journey.
- * Pinkmatter for your financial support for taking the risk of investing in me and my future. I will be eternally grateful. Thank you for letting me feel like part of the company and taking such good care of me during the visits.
- * Clarissa, my wife, thank you for all your love and support during the difficult times.
- * My Mother, words cannot describe how much you meant to me during my masters degree.
- * My friend and colleagues in the ESL, especially Brandon Chetty, Dane Groves and Mark Msonko, who put up with my 'new productivity hacks' and 'crazy ideas', but also keeping me on track.
- * God, for all your guidance, support, love and faith in me, even though I didn't deserve it.

“ *Whether you think you can or you can't, you're right.* ”

– Mr. Henry Ford

DECLARATION

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

1 September 2025
Date:

Copyright © 2025 Stellenbosch University
All rights reserved.

ABSTRACT

Pose estimation on nanosatellites is still an on going topic of interest. It is important for satellite to know there position and attitude to do accurate target tracking. Traditional solutions to the pose estimation problem is mainly star trackers, which looks at the constalations of stars to determine the attitude and GPS to determine the position of the satellite along with other sensors like magnetometers and coarse sun sensors.

In this thesis, a sensor is developed that utilises the onboard satellite imager, to estimate the position and the attitude of the satellite. The sensor uses a camera model to take pictures of the Earth surface, a feature detector is ran on the image using scale invariant feature transform (SIFT) to identify and establish corospondance of features. A full state kinematic estimator using the extended Kalman Filter (EKF) based on the simultanous localisation and mapping (SLAM) approach. The filter makes used of feature vectors and feature discriptors detected on the image. This is used to estimate attitude and position of the satellite.

An simulation environment in MATLAB is developed to propagate a satellite and determine the ground truth pose. Several traditional sensors like the star tracker and magnetometer and GPS to be able to compare the Earth Tracker and create the possibility to fuse the sensors and determine the accuracy. Results show that the filter estimates the system states successfully. It is concluded that ...

UITTREKSEL

TABLE OF CONTENTS

Abstract	iii
Uittreksel	iv
Table of Contents	v
List of Figures	ix
List of Tables	xiv
Nomenclature	xvi
Variables and functions	xvi
Acronyms and abbreviations	xxi
1. Introduction	1
1.1. Problem Background	1
1.2. Proposed Solution: Visual Navigation for Pose Estimation	2
1.3. Contributions and Thesis Outline	3
2. Literature Study	5
2.1. Introduction	5
2.2. Industry Trends	5
2.3. Relative Pose Estimation	7
2.4. Absolute Pose Estimation	10
2.4.1. Traditional Pose Estimation	10
2.4.2. Visual Based Navigation	13
2.4.2.1. Image Processing	16
2.4.2.2. State Estimation	19
2.5. Conclusion	20
3. Modelling	22
3.1. Introduction	22
3.2. Rigid Body Mechanics	22

TABLE OF CONTENTS

3.2.1. Kinematics	22
3.2.2. Dynamics	25
3.2.2.1. Translational Dynamics	26
3.2.2.2. Rotational Dynamics	28
3.3. Reference Frame Transformations	30
3.3.1. Transformation Matrix	31
3.3.1.1. Rotation Matrix	31
3.3.1.2. Translation	31
3.3.1.3. Homogeneous Transformation Matrix	32
3.3.2. Latitude, Longitude, and Altitude	32
3.3.3. Earth Cenetered Earth Fixed	35
3.3.4. Earth Centered Inertial	36
3.3.5. Orbital Reference Frame	38
3.3.6. Body Referenece Frame	39
3.3.7. North - East - Down/ East - North - Up	40
3.4. Sensor Modelling	41
3.4.1. GPS Sensor Model	42
3.4.2. Gyroscope Sensor Model	43
3.4.3. Coarse Sun Sensor Model	45
3.4.4. Star Tracker Sensor Model	47
3.4.5. Magnetometer Sensor Model	49
3.5. Conclusion	51
4. Image Processing	53
4.1. Introduction	53
4.2. Satellite Image Characteristics	53
4.2.1. Imaging Geometry	53
4.2.2. Image Resolution	56
4.2.2.1. Spatial Resolution	56
4.2.2.2. Temporal Resolution	57
4.2.2.3. Radiometric resolution	58
4.2.3. Image Detectors	59
4.2.4. Environmental Noise	60
4.3. Camera Model	63
4.3.1. Pinhole Model	63
4.3.2. Intrinsic Camera Matrix	65
4.3.3. Extrinsic Camera Matrix	66
4.4. Lens Distortion	67
4.4.1. Radial Distortion	67

TABLE OF CONTENTS

4.4.2. Tangential Distortion	68
4.4.3. Chromatic Abberation	69
4.5. Earth Tracker	71
4.5.1. Image Generation	71
4.5.2. Feature Detection	73
4.5.3. Earth Tracker Algorithm	74
4.5.4. Catalogue Generation	75
4.6. Conclusion	77
5. State Estimation	78
5.1. Introduction	78
5.2. Recursive Estimation	78
5.3. The Extended Kalman Filter	80
5.3.1. The Prediction Step	81
5.3.2. The Update Step	82
5.3.3. Multi-Sensor Fusion Techniques	83
5.4. System Modelling	85
5.4.1. Motion Model	85
5.4.2. Sensor Measurement Model	87
5.4.2.1. Earth Tracker Measurement Model	88
5.4.2.2. GPS Measurement Model	88
5.4.2.3. Gyroscope Measurement Model	89
5.4.2.4. Coarse Sun Sensor Measurement Model	89
5.4.2.5. Magnetometer Measurement Model	90
5.4.2.6. Star Tracker Measurement Model	90
5.5. System Initialization	91
5.5.1. Position Initialisation	91
5.5.2. Velocity Initialisation	92
5.5.3. Attitude Initialisation	92
5.5.4. Angular Velocity Initialisation	93
5.6. System Integration	94
5.6.1. Satellite System	94
5.6.2. Earth Tracker System	95
5.6.3. Pose Estimation System	96
5.7. Conclusion	96
6. Experiments	98
6.1. Introduction	98
6.2. Test Configuration	98
6.3. Sensor Test	100

TABLE OF CONTENTS

6.3.1. Earth Tracker Alone	100
6.3.2. The ADCS Suite	102
6.3.3. Earth Tracker vs Star Tracker	104
6.4. Feature Test	106
6.5. TemporalTest	108
6.6. Conclusion	108
7. Conclusions and Future Work	109
7.1. Conclusion	109
7.2. Future Work	109
References	110
A. Appendix title goes here	115

LIST OF FIGURES

1.1.	Conceptual diagram of the proposed visual navigation problem, illustrating the key reference frames (ECI, Body) and the geometric relationship between the satellite's imager and known features on the Earth's surface.	2
2.1.	Annual number of CubeSat launches from 2002 to 2018, illustrating the rapid growth and adoption of small satellite platforms over the past two decades [1]. The trend highlights the increasing accessibility of space technologies driven by reduced launch costs, standardized CubeSat architectures, and expanding participation from commercial, educational, and research institutions. [1]. . .	6
2.2.	In this figure we can see how much space is taken up by the optical payload in the Pheonix CubeSat from the Arizona State University, which is about 1.5U in a 3U CubeSat. [2]	7
2.3.	Conceptual diagram of a chaser satellite performing relative pose estimation of a target satellite for an on-orbit servicing mission. [3]	8
2.4.	Comparison of sensor modalities: (a) Monocular image and (b) Stereo camera pair triangulation.	9
2.5.	NORAD Ground Tracks and Estimation for 4 x 10 min TLE data. [4] . . .	11
2.6.	A typical star tracker used for high-precision attitude determination.[5] . .	12
2.7.	A magnetometer for measuring the direction of the Earth's magnetic field.[6]	12
2.8.	A typical visual geo-localization pipeline, illustrating the use of geolocated features and a pre-rendered database. [7].	14
2.9.	Attitude estimation and control reference frames of the satellite [8].	15
2.10.	Carozza et al. vision-based attitude estimation framework, illustrating the use of feature matching and homography [9].	16
2.11.	Feature Mathcing using SURF and RANDSAC [10]	17
2.12.	Feature matching using the SIFT algorithm and calulating error vectors [11]	18
2.13.	Feature detection using ORB [12].	19
2.14.	Literature study framework and concepts learned out of each section.	20
3.1.	The Eular 3-2-1 rotation sequence (yaw-pitch-roll) describes a 3D orientation.	24
3.2.	Quaterion rotation represented by the rotation axis \bar{e} and rotation angle θ . .	24

3.3.	The real position and real velocity of the satellite at an altitude of 500km and an inclination of 48.8°, illustrating the change in position and velocity overtime governed by the gravitational kinematics.	27
3.4.	The J2 error introduced into system, illustrating its effects on the position and velocity.	28
3.5.	The Dzhanibekov effect to verify the dynamics.	29
3.6.	Illustration of the attitude coupling effect arising from differences in the spacecraft's principal moments of inertia. Variations in these inertial properties can cause rotations about one axis to induce secondary motions about the other axes, demonstrating the intrinsic coupling between rotational dynamics and the spacecraft's mass distribution.	30
3.7.	The WGS84 reference ellipsoid, a mathematically defined model of the Earth's shape used as the basis for the World Geodetic System 1984 (WGS84). It approximates the Earth as an oblate spheroid with a defined semi-major and semi-minor axis, providing a standard reference for geodetic measurements, satellite positioning, and global navigation systems.	33
3.8.	Comparison of different Earth models: the perfect sphere, the reference ellipsoid, and the geoid. The spherical model is a simple approximation with constant radius, the ellipsoid accounts for the Earth's flattening at the poles and bulging at the equator, and the geoid represents the irregular shape of the Earth's gravity field, closely following mean sea level. This comparison highlights the increasing accuracy and complexity of models used in geodesy, navigation, and Earth observation.	34
3.9.	A Digital Elevation Model (DEM) of the African continent, generated using USGS software. This DEM provides a detailed representation of the terrain's elevation, capturing variations in topography such as mountains, valleys, and plateaus, and is widely used in geospatial analysis, environmental modeling, and Earth observation studies.	35
3.10.	The Earth-Centered, Earth-Fixed (ECEF) reference frame, a Cartesian coordinate system fixed relative to the Earth, with its origin at the Earth's center of mass. The \bar{x} -axis points toward the intersection of the equator and the prime meridian, the \bar{y} -axis points 90° east of the X-axis along the equator, and the \bar{z} -axis aligns with the Earth's rotational axis toward the North Pole. This frame is commonly used for satellite navigation, geodesy, and Earth observation applications.	36

3.11.	The Earth-Centered Inertial (ECI) reference frame, an inertial Cartesian coordinate system with its origin at the Earth's center of mass. Unlike Earth-fixed frames, the ECI frame does not rotate with the Earth, making it suitable for describing the motion of satellites, spacecraft, and other celestial objects relative to the stars. It provides a consistent reference for orbital mechanics and space navigation.	37
3.12.	The Orbital Reference Frame, also known as the Local Vertical Local Horizontal (LVLH) frame, is a spacecraft-centered coordinate system used to describe positions and orientations relative to an orbiting body. Its origin is at the spacecraft's center of mass, the \bar{z} -axis points toward the center of the Earth (nadir), the \bar{y} -axis is aligned opposite to the orbital angular momentum vector (orbit normal), and the \bar{x} -axis completes the right-handed system, pointing along the velocity vector in the orbital plane. This frame is widely used in attitude control, guidance, and navigation for satellites.	39
3.13.	The Body Reference Frame, denoted by \mathcal{B} , is fixed to the satellite itself, with its origin at the satellite's center of mass. The axes are defined relative to the satellite's orientation: the \bar{x} -axis corresponds to roll, the \bar{y} -axis to pitch, and the \bar{z} -axis to yaw.	40
3.14.	The East-North-Up reference frame	41
3.15.	GPS measurement and noise model derived from the \mathbf{x}_{true} state in the \mathcal{L} reference frame.	43
3.16.	Groundtrack representation of the simulated GPS measurements, illustrating the apparent Earth rotation as the satellite completes an orbit.	43
3.17.	Simulated gyroscope measurement showing high-frequency noise and drift. .	45
3.18.	Orientation of the six coarse sun sensors on the CubeSat	46
3.19.	Simulated CSS measurement illustrating the magnitude of the direction of the sun and the noise.	47
3.20.	Representation of star vectors calculated out of the center of mass of the satellite, each star vector is a unit vector defined in the BRF.	48
3.21.	Simulated Star Tracker Measurements from the first 4 Stars, the figure shows how the unit vectors (direction) changes as the satellite orbits.	49
3.22.	The total magnetic field intensity given by the <i>worldmgn</i> function using IGRF. .	50
3.23.	Simulated Magnetometer measurements indicating the magnetic field strength in nT and which direction it is measured in according to the IGRF. .	51

4.1.	The relationship between sensor pitch, focal length, satellite altitude, and Ground Sampling Distance (GSD). The pixel pitch (p) determines the physical size of each sensor element, while the focal length (f) and satellite altitude (h) together define how this sensor geometry maps to the ground. A smaller pixel pitch or a longer focal length results in a finer GSD (higher image resolution), whereas increasing altitude enlarges the GSD, reducing spatial resolution.	54
4.2.	An illustration of the distortion of the <i>GSD</i> and <i>FOV</i> of an satellite image when taken Off-Nadir (at an oblique angle) [13]	55
4.3.	The effect off-nadir angle on satellite imagery.	56
4.4.	Illustration of the relationship between spatial resolution and Ground Sampling Distance (GSD). As GSD increases, each pixel represents a larger area on the ground, resulting in lower image resolution and less detail. Conversely, smaller GSD values correspond to higher resolution, capturing finer features in the scene.	57
4.5.	Illustration of temporal resolution in satellite imagery represented by a tree growing [14]. Temporal resolution refers to how frequently a satellite revisits the same location. Higher temporal resolution, achieved by shorter revisit intervals or larger satellite constellations, allows changes on the ground, such as vegetation growth or urban development, to be monitored more closely over time.	58
4.6.	Images are captured using different wave lengths and reconstructed to convey different information.	59
4.7.	Images are captured using different imager detectors.	60
4.8.	Difference in satellite imagery due to atmospheric absorption and scattering [15, 16]	61
4.9.	The effect of cloud cover changing the spectral signature of an image.	62
4.10.	The effect of seasonal and time differences.	62
4.11.	A satellite imager illustrated as an ideal pinhole model camera.	64
4.12.	The relationship between the camera reference frame and the pixel plane reference frame.	65
4.13.	Radial lens distortion applied to an satellite image of Cape Town stadium.	68
4.14.	Tangential lens distortion applied to a satellite image of Cape Town stadium.	69
4.15.	The effect of the lens on different wave lengths of light which will cause some wavelengths to be out of focus and enlarged. [17]	70
4.16.	Chromatic Abberation on a satellite image of Cape Town stadium	70
4.17.	Earth rendered in the simulation environment	71
4.18.	High resolution mosiac strip of Paris, France	72
4.19.	This is the high resolution strip placed on the surface of the simulated Earth.	72
4.20.	An generated satellite imager as seen from the satellite imager	73

4.21.	The french country side is used to compare the different feature points detected by SIFT, SURF and ORB	73
4.22.	An illustration of feature points being projected back to feature vectors.	75
4.23.	Catalogue generation using raycasting.	77
5.1.	Recursive estimator algorithm flowchart [18]	79
5.2.	The fully integrated Batch EKF architecture, which combines all raw sensor data into a single, complex measurement model $\mathbf{h}(\mathbf{x})$ for concurrent processing.	83
5.3.	The Multi-Filter EKF architecture, featuring independent local filters for each sensor and a master filter to fuse the local estimates into a global state.	84
5.4.	The Sequential Update EKF architecture, illustrating a single prediction step followed by multiple, sequential update steps for each individual sensor measurement.	84
5.5.	Satellite pose estimation states in the ECI reference frame	85
5.6.	Block diagram of the satellite simulation architecture. The 'true' motion model provides the ground truth state to two parallel pipelines: one generating distorted camera images and the other generating auxiliary sensor data.	94
5.7.	Block diagram of the Earth Tracker System's data processing pipeline. An input image undergoes feature detection, and the resulting feature pixels are processed through two concurrent streams: 2D-to-3D projection to generate the primary measurement vector, and catalogue generation to create geolocated features for pose estimation.	95
5.8.	Block diagram of the Pose Estimation System. A sequential update EKF fuses the Earth Tracker measurement, the geolocated catalogue vector, and auxiliary sensor data to produce an optimal estimate of the satellite's state. .	96
6.1.	The simera sense TriScape Camera	98
6.2.	Earth Tracker Measruement and measruemetn error of Feature 1	100
6.3.	The error graph between the true and estimated state generated by the Eart Tracker	101
6.4.	The Attitude Error of the Earth Tracker	101
6.5.	The error graph between the true and estimated state generated by the ADCS suite	103
6.6.	The Attitude Error of the ADCS Suite	103
6.7.	The error graph between the true and estimated state generated by the ADCS suite	104
6.8.	The Attitude Error of the ADCS Suite	105
6.9.	The error graph between the true and estimated state generated by the ADCS suite	105
6.10.	The Attitude Error of the ADCS Suite	106

LIST OF FIGURES

6.11. Earth Tracker Measurement with lens distortion	107
--	-----

LIST OF TABLES

2.1.	Sensor type, principle of operation, and typical accuracy.	12
2.2.	Sensor advantages and limitations.	12
6.1.	Sensor characteristics including sample rate, noise, and drift.	100
6.2.	Sensor ranking by filter order and justification.	102
6.3.	Sensor characteristics including sample rate, noise, and drift.	102
6.4.	Sensor characteristics including sample rate, noise, and drift.	104
6.5.	Comparison of position and attitude errors for different experiments. . . .	106
6.6.	Sensor characteristics including sample rate, noise, and drift.	106
6.7.	Performance of feature detectors for different numbers of features, showing position and attitude errors.	107
6.8.	Lens Distortion Parameters	107
6.9.	Comparison of position and attitude errors for different experiments with and without time delay.	108

NOMENCLATURE

VARIABLES AND FUNCTIONS

CONSTANTS

μ	Earth's gravitational parameter ($398600 \text{ km}^3/\text{s}^2$)
ω_e	Rotation speed of the Earth ($7.2921 \times 10^{-5} \text{ rad/s}$)
a	Semi-major axis of WGS84 ellipsoid (6378 km)
b	Semi-minor axis of WGS84 ellipsoid (6356 km)
I_{xx}	Moment of inertia about x-axis
I_{yy}	Moment of inertia about y-axis
I_{zz}	Moment of inertia about z-axis
J_2	Second zonal harmonic coefficient (1.082×10^{-3})
R_E	Mean radius of the Earth (6378 km)

FUNCTIONS

$\arctan 2(\cdot)$	Four-quadrant inverse tangent function
$\text{rotm2quat}(\cdot)$	Rotation matrix to quaternion conversion
$\text{wrldmagm}(\cdot)$	World magnetic model (IGRF) function
$f(\cdot)$	Process/motion model function
$h(\cdot)$	Measurement model function

MATRICES

\mathbf{A}	Direction cosine matrix (DCM) / rotation matrix
--------------	---

F	Jacobian of process model
G	Control matrix
H	Jacobian of measurement model
I	Inertia tensor / identity matrix
K	Intrinsic camera matrix
K	Kalman gain matrix
P	State covariance matrix
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
R	Rotation matrix
S	Innovation covariance matrix
T	Homogeneous transformation matrix

NOISE TERMS

η	Measurement noise vector
d	Drift/bias vector
q	Drift rate noise
v	Measurement noise
w	Process noise vector
σ^2	Variance

OPERATORS AND NOTATION

$(\cdot)^+$	Homogeneous coordinate representation
$(\cdot)^T$	Matrix/vector transpose
$(\cdot)^{-1}$	Matrix inverse
$\vec{\cdot}$	Unit vector
\cdot	Dot product

:	Second time derivative
:	Time derivative
$\hat{\cdot}$	Estimated value
\otimes	Quaternion multiplication
\times	Cross product

REFERENCE FRAME SUBSCRIPTS

B	Body reference frame
C	Camera reference frame
E	East-North-Up (ENU) reference frame
I	Inertial (ECI) reference frame
L	Geodetic (LLA) reference frame
M	Image plane (metric) reference frame
N	North-East-Down (NED) reference frame
O	Orbital (LVLH) reference frame
P	Pixel plane reference frame
R	Earth-Centered Earth-Fixed (ECEF) reference frame

VARIABLES

α	Skewing factor
ω	Angular velocity vector
Δt	Time step
$\hat{\mathbf{x}}$	Estimated state vector
$\hat{\mathbf{z}}$	Predicted measurement vector
λ	Longitude (geodetic)
\mathbf{a}	Acceleration vector
\mathbf{F}	Force vector

f	Feature vector
H	Angular momentum vector
p	Point vector
q	Quaternion vector
r	Position vector
S	Sun vector
s	Star vector
T	Torque vector
u	Control input vector
v	Velocity vector
x	State vector
x_{true}	True state vector
y	Innovation/residual vector
z	Measurement vector
ϕ	Roll angle (Euler angle)
ψ	Yaw angle (Euler angle)
θ	Pitch angle (Euler angle)
φ	Latitude (geodetic)
c_x	Horizontal principal point offset
c_y	Vertical principal point offset
f	Focal length
h	Altitude
k_1, k_2, k_3	Radial distortion coefficients
m	Mass
p	Pixel pitch

p_1, p_2	Tangential distortion coefficients
q_s	Quaternion scalar component
q_x	Quaternion x-component
q_y	Quaternion y-component
q_z	Quaternion z-component
r	Radial distance/position magnitude
s_x	Horizontal scale factor
s_y	Vertical scale factor
t	Time

ACRONYMS AND ABBREVIATIONS

ADCS	Attitude Determination and Control System
BRF	Body Reference Frame
CNN	Convolutional Neural Network
COTS	Commercial-Off-The-Shelf
CRF	Camera Reference Frame
CSS	Coarse Sun Sensor
DCM	Direction Cosine Matrix
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EKF	Extended Kalman Filter
ENU	East-North-Up
EO	Earth Observation
FOV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSD	Ground Sampling Distance
ICRF	International Celestial Reference Frame
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
J2000	Epoch J2000.0
LEO	Low Earth Orbit
LiDAR	Light Detection and Ranging
LKT	Lucas-Kanade Tracker
LLA	Latitude, Longitude, and Altitude Reference Frame
LVLH	Local Vertical Local Horizon
MSI	Multispectral Imagers
NED	North-East-Down
NIR	Near-Infrared

NORAD	North American Aerospace Defense Command
OCS	Orbit Control System
ONA	Off-Nadir Angle
ORB	Oriented FAST and Rotated BRIEF
PF	Particle Filter
PnL	Perspective-n-Lines Problem
PnP	Perspective-n-Point Problem
QGIS	Quantum Geographic Information System (GIS Software)
RANSAC	Random Sample Consensus
RUKF	Robust Unscented Kalman Filter
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localisation and Mapping
ST	Star Tracker
SURF	Speeded-Up Robust Features
SWaP-C	Size, Weight, Power, and Cost
SWIR	Shortwave Infrared
TLE	Two-Line Elements
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
USGS	U.S. Geological Survey
VO	Visual Odometry
WGS84	World Geodetic System 1984

CHAPTER 1

INTRODUCTION

1.1 PROBLEM BACKGROUND

In recent years, the satellite industry has undergone a significant paradigm shift toward smaller, more cost-effective platforms. This trend is largely enabled by the ongoing miniaturization of electronics and sensor technology, which has allowed for the development of highly capable small satellites, including nanosatellites and CubeSats. The primary drivers behind this shift are the substantially reduced development costs and accelerated timelines associated with these smaller platforms, which have made space accessible for academic, commercial, and governmental entities [1, 19].

A predominant application within this rapidly growing market is Earth Observation (EO), which involves using satellite-borne sensors to monitor the planet for a wide range of scientific and commercial purposes. A key requirement for many EO missions is the ability to capture high-resolution imagery (for a GSD in the range of 5-10m) of specific, pre-defined targets on the Earth's surface. This, in turn, necessitates a high degree of pointing accuracy (typically in the range of 0.01° or better) from the satellite's Attitude Determination and Control System (ADCS) [20–22].

This requirement introduces the critical technical challenge at the heart of this research. The low-cost philosophy of small satellites mandates the use of Commercial-Off-The-Shelf (COTS) components, creating a trade-off between cost and performance [1, 19]. A significant gap exists between the pointing accuracy required for meaningful remote sensing and what is achievable with standard, low-cost ADCS sensors. For instance, magnetometers provide robust, continuous attitude information but lack the necessary precision, while high-performance star trackers can deliver the required accuracy but are often prohibitively expensive [23]. This dilemma creates a pressing need for novel, cost-effective solutions that can bridge this performance gap.

1.2 PROPOSED SOLUTION: VISUAL NAVIGATION FOR POSE ESTIMATION

To address this challenge, this thesis investigates an autonomous visual navigation approach to determine the satellite's full six-degree-of-freedom (6-DOF) pose, encompassing both its three-dimensional position and attitude. Instead of relying on traditional external systems like GNSS, the satellite performs a type of "reverse GPS" by identifying known ground features in its own imagery to compute its state. The primary novelty of this work is the application of visual navigation techniques, commonly used in robotics, to Earth-orbiting satellites. Traditional satellite ADCS architectures rely on a suite of dedicated sensors, whereas this thesis demonstrates a "payload-as-a-sensor" approach, repurposing the primary science imager as the main navigation instrument. Figure 1.1 illustrates the core mechanics of this problem, showing the relationship between the satellite and the Earth in the Earth-Centered Inertial (ECI) reference frame and how it relates to the 6-DOF navigation challenge.

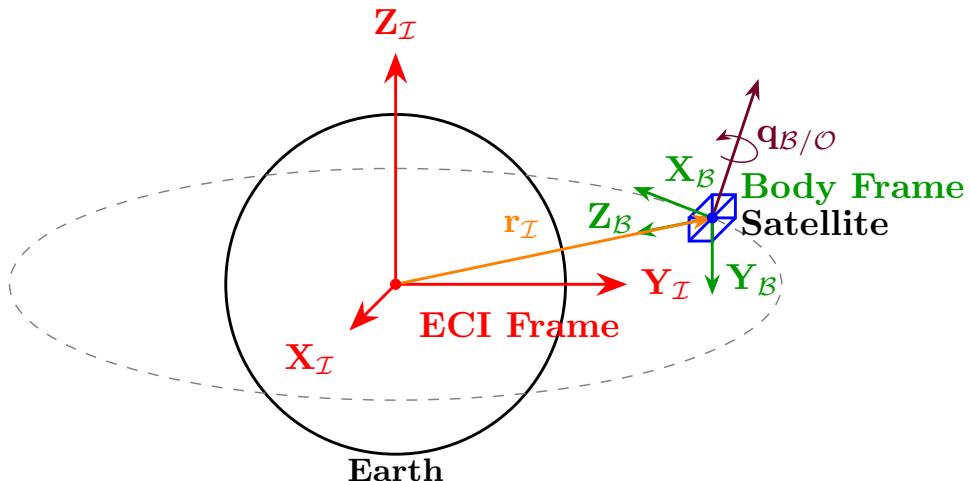


Figure 1.1: Conceptual diagram of the proposed visual navigation problem, illustrating the key reference frames (ECI, Body) and the geometric relationship between the satellite's imager and known features on the Earth's surface.

This "payload-as-a-sensor" methodology presents several significant advantages. It offers a potential reduction in system cost and complexity by eliminating the need for expensive, high-fidelity sensors. It also enhances system efficiency by enabling the dual-use of the primary imager during periods when it might otherwise be idle. Finally, it provides a direct, relative measurement of the satellite's pointing with respect to the Earth, which is fundamentally what is required for high-accuracy EO missions.

The core technical challenge lies in transforming raw imagery into a precise pose estimate.

Although conceptually related to Simultaneous Localization and Mapping (SLAM), the problem considered here involves only the localization component. Unlike full SLAM, where both the environment and the pose are estimated simultaneously, the "map", a catalogue of georeferenced ground features, is assumed to be known *a priori*. The focus is therefore on the localization task, which entails addressing three key challenges:

- * **Geometric Inversion:** The complex inverse problem of determining a 6-DOF pose from 2D image projections of known 3D landmarks.
- * **Uncertainty Management:** The real-time handling of measurement noise and dynamic orbital motion within a recursive filter.
- * **Sensor Fusion Integration:** Incorporating visual measurements into the traditional Attitude Determination and Control System (ADCS) through sensor fusion techniques, enabling synergistic use of camera, gyroscope, magnetometer, and other sensor data.

To maintain a clear research focus, this thesis presumes that the feature matching problem, where the correct association of image features to catalogue entries, is solved. This necessary simplification allows the research to concentrate on its core contribution: developing a state estimation framework that optimally leverages the geometric data from these established feature correspondences.

1.3 CONTRIBUTIONS AND THESIS OUTLINE

The main contributions of this thesis are:

- * The formulation of a novel satellite state estimation framework that adapts visual odometry principles from robotics for Earth-orbiting applications.
- * The development of a "payload-as-a-sensor" model that integrates the primary imager as a navigation sensor within an Extended Kalman Filter.
- * The design and implementation of a complete simulation environment to validate the performance of the proposed visual navigation system under various operational conditions.

The remainder of this document is structured as follows:

Chapter 2 presents a comprehensive literature review, examining both relative and absolute pose estimation methods. It then explores traditional pose estimation and visual-based navigation approaches, followed by an investigation of contemporary image

processing techniques, state estimation methods, and sensor fusion strategies involving traditional sensors.

Chapter 3 establishes the mathematical framework for the simulation, detailing rigid body mechanics, reference frame transformations, and the models for auxiliary sensors.

Chapter 4 details the development of the vision-based Earth Tracker, including the camera model, lens distortion simulation, and the algorithms for generating measurement vectors and feature catalogues.

Chapter 5 focuses on the design of the state estimation algorithm, presenting the theoretical background of recursive estimation and the detailed derivation of the Extended Kalman Filter (EKF) for this specific application.

Chapter 6 presents the simulation results and a thorough performance analysis of the system under various test conditions, including sensor noise, feature availability, and distortion effects.

Chapter 7 concludes the thesis by summarizing the key findings, discussing the limitations of the work, and proposing avenues for future research.

CHAPTER 2

LITERATURE STUDY

2.1 INTRODUCTION

This chapter provides the contextual and technological background underlying the development of a novel visual navigation system for nanosatellite pose estimation. It begins by addressing the economic motivation and rationale behind the problem, as identified in existing literature.

The chapter then follows a cascading structure. It first distinguishes between relative and absolute pose estimation, establishing the foundational concepts for attitude determination. Next, it compares traditional pose estimation sensors with emerging vision-based navigation techniques, highlighting the shift toward optical solutions in modern nanosatellite missions. The discussion then situates vision-based navigation within the context of space applications, before exploring its two core components in detail: image processing and state estimation through sensor fusion.

2.2 INDUSTRY TRENDS

The landscape of the space industry has undergone a radical transformation, moving from a highly exclusive domain to a commercially accessible one. Historically, the greatest barriers to entry were the large costs associated with developing specialised satellite components and the prohibitive expenses of space launch campaigns, which restricted missions primarily to government or military funding [24, 25]. This dynamic has fundamentally changed due to two synergistic global trends: the relentless miniaturisation of electronics and the competitive reduction in launch costs by new commercial providers. This reduction in launch costs has significantly lowered the price per kilogram to Low Earth Orbit (LEO) [26], thereby democratizing access to space for commercial and educational sectors. As

illustrated in Figure 2.1, the number of CubeSat launches is projected to increase rapidly in the coming years [1, 19, 27].

This shift necessitated a new development paradigm based on standardisation and an iterative approach, culminating in the widespread adoption of the CubeSat platform [1, 28]. The CubeSat standard is defined by its strict SWaP-C (Size, Weight, Power, and Cost) constraints [29, 30]. The necessity of low cost means that CubeSats heavily rely on Commercial Off-the-Shelf (COTS) components. While this maintains affordability, COTS-based sensors are inherently less accurate and reliable than their expensive, space-rated counterparts [31].

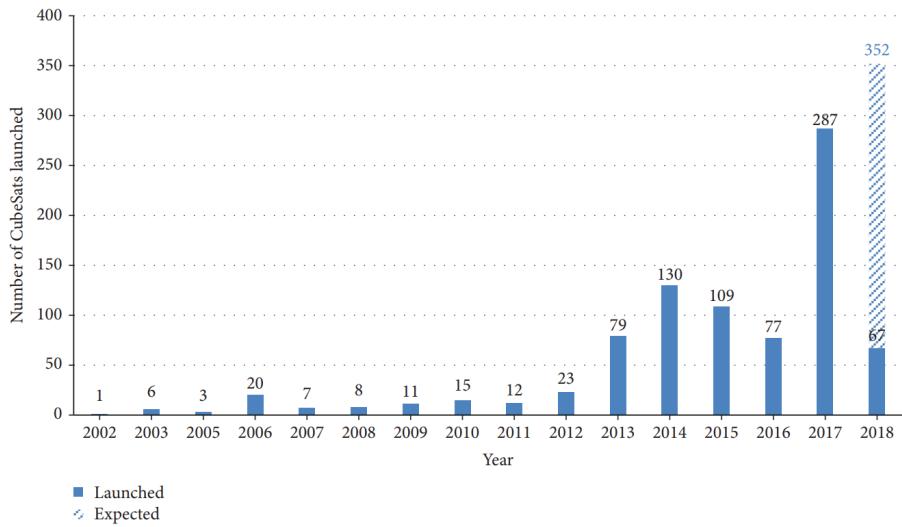


Figure 2.1: Annual number of CubeSat launches from 2002 to 2018, illustrating the rapid growth and adoption of small satellite platforms over the past two decades [1]. The trend highlights the increasing accessibility of space technologies driven by reduced launch costs, standardized CubeSat architectures, and expanding participation from commercial, educational, and research institutions. [1].

The low cost of CubeSats has driven a surge in lucrative Earth Observation (EO) missions, where high-resolution cameras function as the primary payload [27, 32]. EO applications demand highly accurate target tracking and precise pose (position and attitude) estimation [32]. Crucially, the size and power requirements of the EO payload severely limit the allocation of resources to the Attitude Determination and Control Subsystem (ADCS) [31] as illustrated in Figure 2.2 in the Arizona State University Pheonix CubeSat [2]. This constraint often forces mission designers to omit the highly accurate, but large and expensive, star tracker, relying instead on coarse sensors like the Magnetometer and Coarse Sun Sensors (CSS) [20, 29].

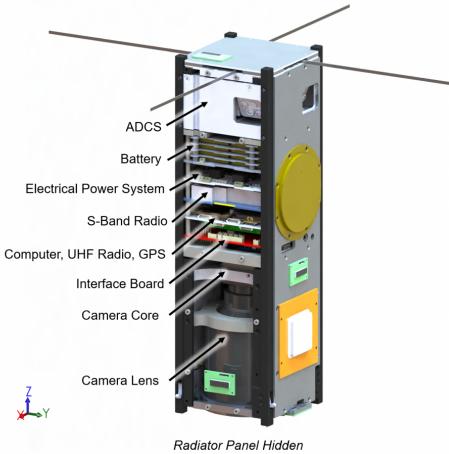


Figure 2.2: In this figure we can see how much space is taken up by the optical payload in the Pheonix CubeSat from the Arizona State University, which is about 1.5U in a 3U CubeSat. [2]

To mitigate the ADCS performance gap resulting from the SWaP-C constraints, a key focus in current research is the "payload-as-a-sensor" concept. This approach involves utilizing the already-onboard imaging payload as a supplementary ADCS sensor. The goal is two-fold: to either completely replace the expensive star tracker by using the camera for high-accuracy attitude determination, or to significantly improve the overall pose estimation accuracy by fusing the visual data with the coarse Magnetometer and CSS readings, thereby leveraging an existing component for enhanced mission capability.

2.3 RELATIVE POSE ESTIMATION

When discussing pose estimation in the context of space applications, the term is largely borrowed from the field of autonomous robotics [33]. On Earth, a ground vehicle might use localisation algorithms, such as Simultaneous Localisation and Mapping (SLAM), to estimate its position and orientation relative to a map it has created of its environment [33, 34]. In space-based applications, this concept is adapted to relative pose estimation, which is crucial for close-proximity operations like rendezvous, docking, Active Debris Removal (ADR), and on-orbit servicing [3, 18, 34]. The typical scenario involves a “chaser” satellite that must determine the pose, that is, the relative position and attitude of a “target” satellite [3, 18, 34–36]. This concept is illustrated in Figure 2.3 where a chief (chaser) satellite is executing relative pose estimation on a deputy (target) satellite.

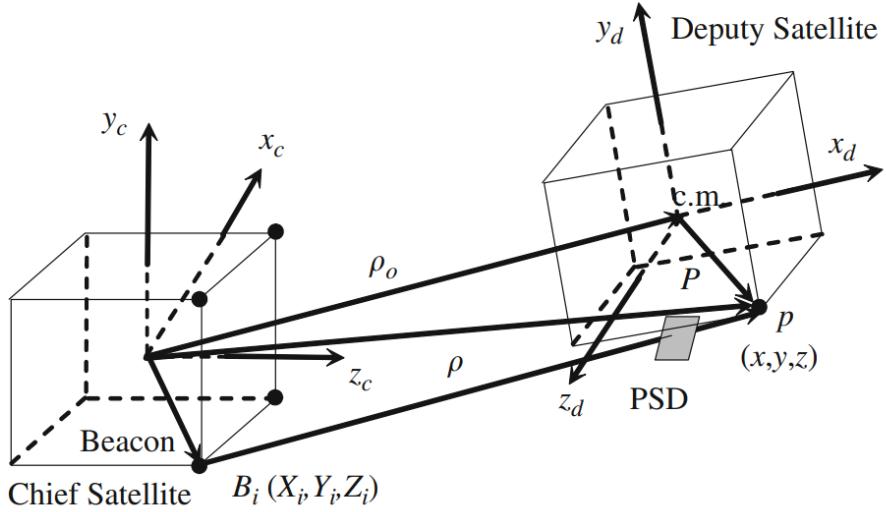


Figure 2.3: Conceptual diagram of a chaser satellite performing relative pose estimation of a target satellite for an on-orbit servicing mission. [3]

Several techniques are employed for relative pose estimation. Visual odometry, using either a single monocular camera or a stereo camera pair, is a popular method due to the low power consumption, cost, and mass of cameras compared to active sensors [34, 36]. Monocular cameras are often favoured for their simplicity, particularly on smaller spacecraft like CubeSats where a sufficient baseline for effective stereo vision is not feasible [18]. While a single camera cannot inherently measure depth, techniques have been developed to overcome this limitation [33]. Stereo vision systems, however, can acquire sparse 3D point clouds through triangulation, making them a viable standalone sensor for tracking unknown targets [33, 34]. Figure 2.4 illustrates pose estimation using a monocular camera and triangulation using a stereo camera [33]. Light Detection and Ranging (LiDAR) is another prominent sensor used to generate 3D point clouds of a target, offering high accuracy over larger distances but at a greater cost and power requirement [18, 33, 34].

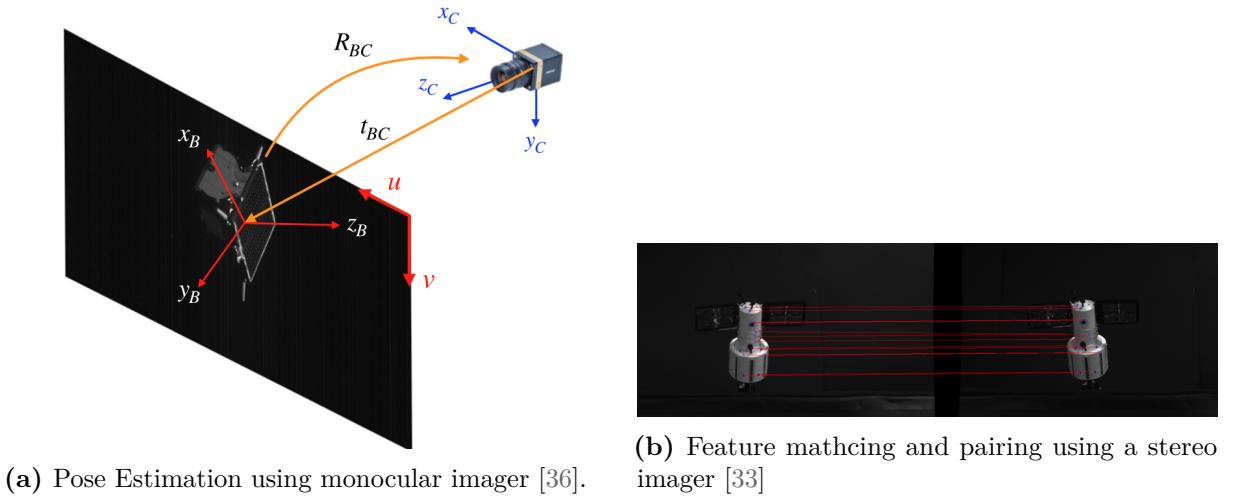


Figure 2.4: Comparison of sensor modalities: (a) Monocular image and (b) Stereo camera pair triangulation.

To extract meaningful data from sensor inputs, various feature-based methods are utilized. These can be categorised into three main types [35]:

- * **Points/Corners:** These features are widely used and form the basis of the **Perspective-n-Point (PnP)** problem, where the pose is determined from correspondences between 3D points on the object and their 2D projections in the image [18, 35, 36]. Algorithms like the Scale-Invariant Feature Transform (SIFT) are commonly used to detect and match these features robustly [36]. However, corner-based methods can be the least robust under varying lighting conditions and can be difficult to track reliably [35].
- * **Lines:** Often found on man-made objects like satellites, line features are generally more robust than points [35]. The **Perspective-n-Lines (PnL)** problem involves estimating pose from line correspondences [35].
- * **Circles/Ellipses:** Features like docking rings or thruster nozzles appear as circles on a satellite, which project as ellipses in an image [34, 35]. Ellipse-based pose estimation is considered the most robust method, especially in noisy conditions and poor lighting, though it can suffer from dual-solution ambiguity where two possible poses can be calculated from a single view [35, 36]. Combining a circle feature with a line feature can help resolve this ambiguity and recover the full 6-DOF pose [35].

Once features are extracted, estimation algorithms process this information to determine the final pose. Classical techniques like the Extended Kalman Filter (EKF) are widely used to fuse sensor measurements with a dynamic or kinematic model of the target's motion

[18, 33, 34, 36]. The EKF recursively estimates the system states, including position, attitude, and their corresponding velocities, making it suitable for tracking moving objects over time [34, 36]. More recently, Convolutional Neural Networks (CNNs) and other deep learning models have emerged as powerful alternatives, outperforming traditional methods in many computer vision tasks [18, 33]. These approaches can directly regress the 6-DOF pose from an image (an “end-to-end” solution) or predict an intermediate representation like the 2D locations of keypoints, which are then used with a PnP solver to calculate the final pose [33, 34]. A significant challenge for these data-driven methods is the scarcity of real, labelled spaceborne imagery for training, leading to a heavy reliance on photorealistic synthetic data [33, 36].

2.4 ABSOLUTE POSE ESTIMATION

In contrast to relative pose estimation absolute pose estimation for a satellite involves finding its state within a fixed, non-moving reference frame. A satellite’s position is typically defined within an absolute inertial reference frame, such as the J2000 International Celestial Reference Frame (ICRF). Its attitude is then described as the orientation of the satellite’s body-fixed frame relative to this inertial frame or, for convenience in many control applications, relative to a local, moving frame like the orbital reference frame.

2.4.1 TRADITIONAL POSE ESTIMATION

Traditionally, the complete pose of a satellite is determined and controlled by two separate subsystems: the Attitude Determination and Control System (ADCS) for orientation, and the Orbit Control System (OCS) for position. As such, these two components of the pose are typically estimated independently.

Position Estimation:

The absolute position of a satellite in orbit is most commonly determined using signals from a Global Navigation Satellite System (GNSS), such as the Global Positioning System (GPS). Onboard GNSS receivers provide the necessary data to compute the satellite’s orbital state vector, its position and velocity, relative to Earth. For applications demanding higher precision, advancements in GNSS technology make use of the International GPS Service (IGS) network, which provides precise orbital and clock corrections. By compensating for satellite clock errors and other biases, these refined techniques can achieve absolute positioning accuracies within approximately 40 cm [37].

While GNSS-based methods are ideal for satellites equipped with receivers, not all spacecraft have such capabilities. In these cases, alternative approaches are employed to estimate orbital position. One of the most widely used methods is based on data published by

the North American Aerospace Defense Command (NORAD), which provides Two-Line Elements (TLEs), which is a standardized format containing the six classical Keplerian orbital elements (e.g., mean motion, eccentricity, and inclination). These elements are propagated using the Simplified General Perturbations model (SGP4), an analytical model that accounts for perturbations such as Earth's gravity harmonics and atmospheric drag. Although less precise than GNSS-based solutions, TLE-based orbit determination remains an essential and accessible tool for tracking satellites, typically achieving positional accuracies within a few kilometers for objects in Low Earth Orbit [4, 38, 39].

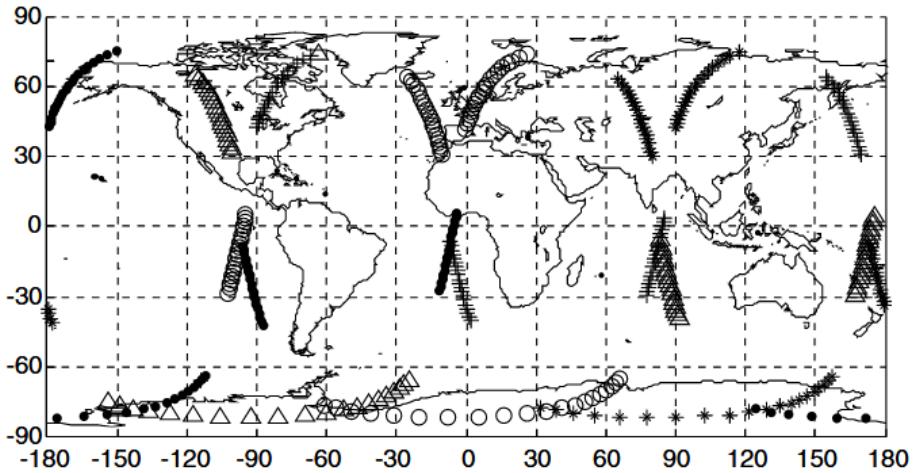


Figure 2.5: NORAD Ground Tracks and Estimation for 4 x 10 min TLE data. [4]

Attitude Estimation:

Attitude determination is an ongoing and critical field of research in spacecraft engineering. Traditionally, a satellite is equipped with a suite of sensors to measure its orientation relative to external references. This sensor suite often includes magnetometers (as seen in Figure 2.7), coarse and fine sun sensors, and star trackers (as seen in Figure 2.6) [40–42]. These are often complemented by internal sensors like gyroscopes which measure the rate of change of the attitude. The data from these various sensors are fused using estimation algorithms to provide a robust and accurate attitude solution.

A brief summary of the different attitude sensors, their capabilities, limitations, and typical accuracies is provided in Table 2.1 and Table 2.2 [43].



Figure 2.6: A typical star tracker used for high-precision attitude determination.[5]



Figure 2.7: A magnetometer for measuring the direction of the Earth's magnetic field.[6]

Sensor Type	Principle of Operation	Typical Accuracy
Star Tracker	Matches observed star patterns to an onboard catalog	High (0.0003° to 0.01°)
Sun Sensor	Measures the light intensity	Coarse (0.005° to 3°)
Magnetometer	Measures local magnetic field vector and compares to a model	Coarse (0.5° to 3°)
Gyroscope (IMU)	Senses angular velocity via inertial effects (e.g., MEMS, FOG)	High rate, low noise (short-term)
Horizon Sensor	Detects infrared contrast at Earth's limb	Moderate (0.1° to 0.25°)

Table 2.1: Sensor type, principle of operation, and typical accuracy.

Sensor Type	Advantages	Limitations
Star Tracker	Very high accuracy, provides 3-axis absolute attitude	Computationally intensive, susceptible to blinding, expensive
Sun Sensor	Simple, reliable, low power	Only provides 2-axis information, inoperable in eclipse
Magnetometer	Low power, works in eclipse, inexpensive	Requires a magnetic field (LEO), needs orbit knowledge, susceptible to interference
Gyroscope (IMU)	High update rate, independent of external references	Suffers from bias and drift, requires initialization and correction
Horizon Sensor	Simple, reliable Earth-pointing reference	Provides only 2-axis (pitch/roll) info, accuracy varies with altitude/season

Table 2.2: Sensor advantages and limitations.

2.4.2 VISUAL BASED NAVIGATION

Visual-based navigation is a method of determining the pose of a vehicle by analyzing sequences of images captured by an onboard camera. This approach has become increasingly important for autonomous systems operating in environments where traditional navigation aids, such as GPS, are unreliable or unavailable [44]. The underlying principle involves detecting and tracking distinctive visual features or patterns across consecutive image frames to infer motion. Broadly, visual navigation methods can be divided into two categories: absolute geo-localization, which estimates a global pose by matching images to a geo-referenced database, and relative pose estimation, which computes motion incrementally between frames [7, 44]. These approaches are often complementary, relative methods ensure continuous short-term tracking, while absolute methods periodically correct accumulated drift by anchoring the system to fixed global references.

Absolute geo-localization relies on matching real-time images captured by a vehicle to a pre-existing geo-referenced database such as satellite imagery or topographic maps. This strategy is widely applied in the Unmanned Aerial Vehicle (UAV) domain, which shares many operational similarities with satellite missions, particularly in vision-based navigation and localization.

A representative example of this approach is the framework developed by Chen et al. for UAVs, which implements a two-stage process for real-time visual geo-localization [7]. The first stage involves offline database preparation, where synthetic color images and depth maps are rendered from satellite and topographic data for a wide range of possible poses. From these, both global and local feature descriptors are extracted and stored in a database. The second stage, online inference, is performed during flight. Here, each incoming “query” image captured by the UAV is compared to the database using global descriptors to retrieve the top candidate matches. The vehicle’s six-degree-of-freedom (6-DOF) pose is then refined by matching local descriptors and solving a Perspective- n -Point (PnP) problem between 2D query image features and corresponding 3D database points derived from the depth maps [7].

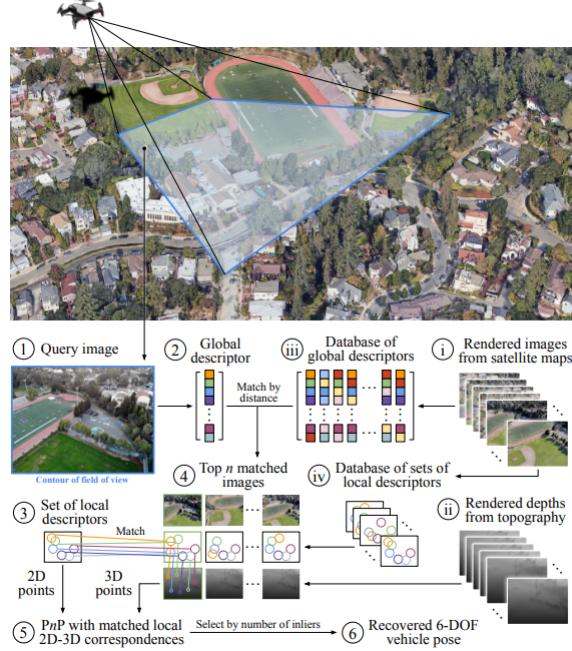


Figure 2.8: A typical visual geo-localization pipeline, illustrating the use of geolocated features and a pre-rendered database. [7].

The principles of visual-based navigation developed for UAVs have also been adapted for satellite applications. In the context of orbital operations, Wu et al. [45] introduced a system that estimates a satellite's absolute position and attitude using an infrared Earth sensor, traditionally an attitude instrument, repurposed for autonomous navigation. The main challenge addressed in this work is cross-modal image matching between the infrared sensor data and a visible-light reference map. To overcome this, the authors employ a hierarchical registration approach composed of two stages:

- * **Coarse Registration:** A Convolutional Neural Network (CNN) extracts deep, abstract features that are robust to differences in resolution, illumination, and distortion [45].
- * **Fine Registration:** Edge-based extended phase correlation is then applied to achieve precise sub-pixel alignment between the sensed and reference images [45].

By combining the outputs of both stages, the system achieves simultaneous estimation of the satellite's 6-DOF pose. A notable contribution of this work is the development of a new Earth background thermal infrared radiation model, required to create a suitable reference dataset for accurate matching [45].

Complementary to this, Klancar et al. [8] proposed a visual-based attitude control system for remote sensing satellites that relies solely on the onboard observation camera. Their approach integrates imaging and control by using the same optical payload for both

Earth observation and attitude determination. The core of this method is an image-based visual servoing algorithm, which uses feature tracking to drive the reaction wheels and maintain a target within the image center. Specifically, Scale-Invariant Feature Transform (SIFT) features are matched between the current image and a reference target image, with the resulting feature displacement used as feedback in a proportional-derivative (PD) control law. The algorithm effectively estimates orientation errors and commands corrective torques, achieving accurate tracking without explicit camera calibration. Using the SGP3 orbital model and Google Earth imagery, simulation results demonstrated robust stabilization and pointing performance [8].

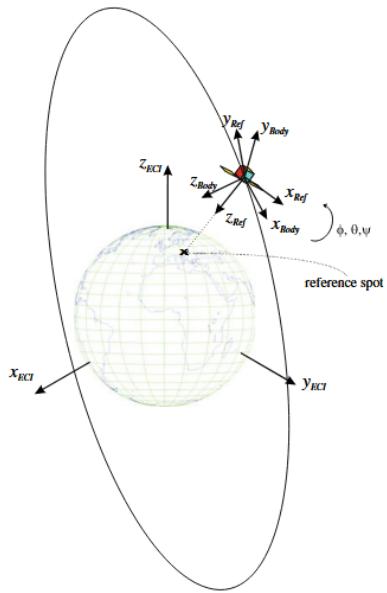


Figure 2.9: Attitude estimation and control reference frames of the satellite [8].

Further refinement of vision-based attitude estimation was presented by Carozza et al. [9], who conducted a detailed analysis of the error sources associated with determining a satellite's three-axis attitude from consecutive overlapping Earth images. Their method estimates attitude changes by registering feature points between successive images, compensating for parallax caused by satellite translation, and modeling the remaining motion as a homography, which is then used to derive quaternion-based attitude changes. The study emphasizes error quantification within the image registration pipeline, which incorporates Shi-Tomasi corner detection, phase correlation, and the Lucas-Kanade Tracker (LKT) with sub-pixel precision, all robustly filtered using RANSAC. Their findings reveal that finite sub-pixel tracking accuracy (on the order of hundredths of a pixel) introduces periodic oscillations and drift in attitude estimation. Moreover, the system exhibits greater sensitivity to roll and pitch compared to yaw, defining an intrinsic accuracy limit governed by the precision of sub-pixel motion tracking [9].

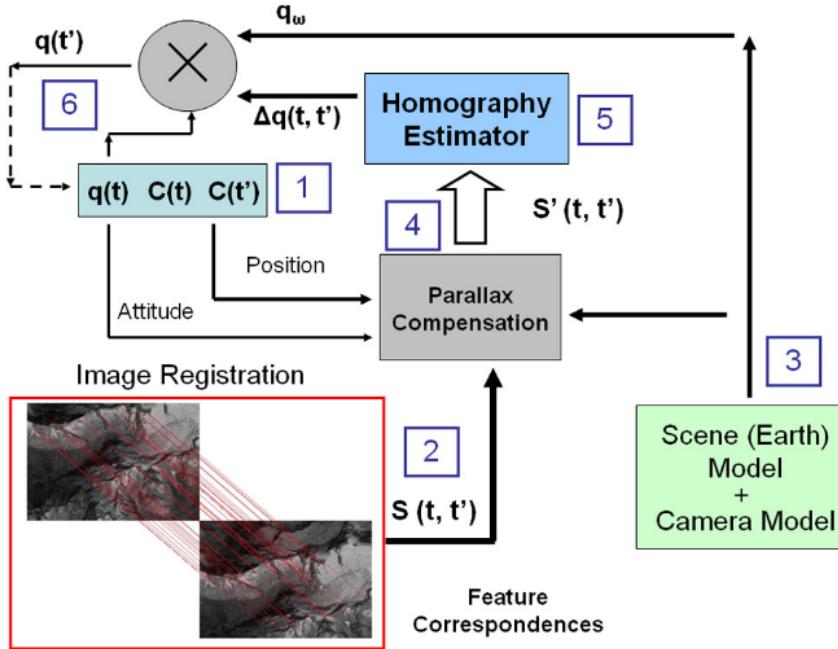


Figure 2.10: Carozza et al. vision-based attitude estimation framework, illustrating the use of feature matching and homography [9].

The proposed work, detailed in this thesis, adopts and extends the vision-based navigation principles established by the methods discussed here, but with a unique focus and architecture. The approach explicitly follows the payload-as-a-sensor methodology, repurposing the main Earth imager for navigation. Unlike the attitude-only systems by Klancar et al. and Carozza et al. [8, 9], this thesis aims for full 6-DOF pose estimation, which is a more comprehensive state space. While related to Chen et al.'s absolute geo-localization and Wu et al.'s use of a reference map [7, 45], the method here frames the problem as a localization component of SLAM, where the "map" is assumed to be known a priori. Furthermore, the core technical contribution is the state estimation framework that optimally leverages the geometric data from these established feature correspondences using an Kalman Filter for multi-sensor fusion. This explicitly addresses the sensor fusion challenge, integrating visual data with traditional auxiliary sensors, a point not centrally focused on by the comparative image registration or visual servoing methods presented.

2.4.2.1 IMAGE PROCESSING

Feature detection and matching form the foundation of image-based navigation and attitude estimation, with various algorithms developed to balance robustness, computational efficiency, and invariance to environmental factors. Kouyama et al. [10] utilized Speeded-Up Robust Features (SURF) in conjunction with a Random Sample Consensus (RANSAC) framework to determine satellite attitude through feature correspondence between observa-

tion images and a registered base map. Their approach emphasizes geometric consistency and outlier rejection, as shown in Figure 2.11, enabling accurate estimation of the attitude matrix even under partial cloud coverage and variable lighting. In contrast, Klancar et al. [11] adopted the Scale-Invariant Feature Transform (SIFT), originally proposed by Lowe [46], which provides strong invariance to scale, rotation, and illumination. While both SURF and SIFT rely on local image gradients to form distinctive descriptors, SIFT’s higher computational load is compensated by its superior distinctiveness, making it particularly effective for visual servoing applications in satellite attitude control as illustrated in Figure 2.12. In Klancar’s framework, matched SIFT features between a reference and a live image generate an image-plane error vector, directly corresponding to angular misalignment, which can then be corrected through feedback control of the reaction wheels.

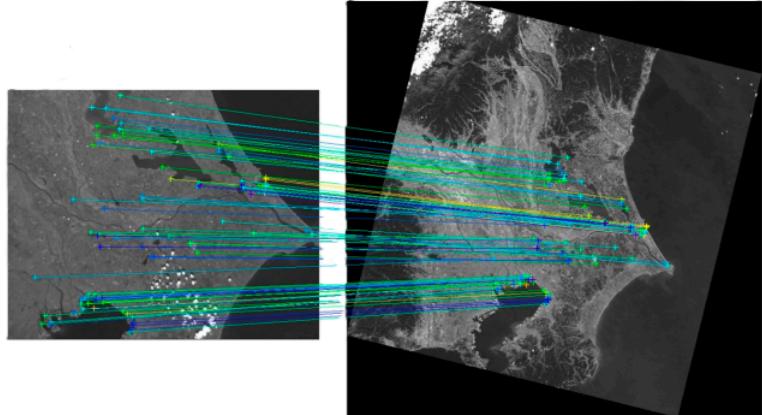


Figure 2.11: Feature Matching using SURF and RANSAC [10]

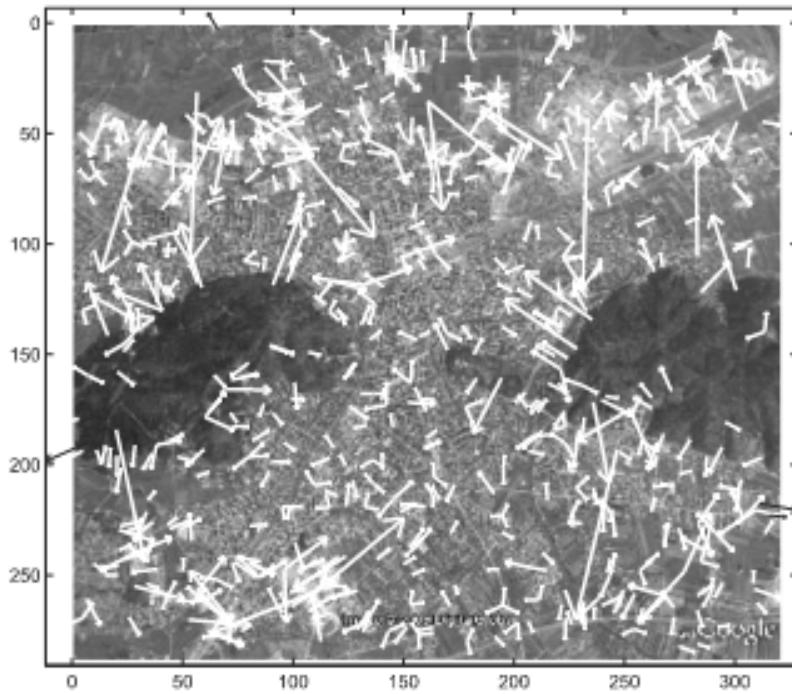


Figure 2.12: Feature matching using the SIFT algorithm and calculating error vectors [11]

For aerial applications, Mateos et al. [12] demonstrated a trade-off between accuracy and computational cost by employing the Oriented FAST and Rotated BRIEF (ORB) descriptor for Visual Odometry (VO) in GPS-denied environments as illustrated in Figure 2.13. ORB forgoes scale-space extrema detection in favor of binary feature extraction, allowing for real-time performance on resource-limited embedded systems. While ORB lacks the precision of SIFT or SURF in handling significant viewpoint and illumination changes, its use of Hamming distance for feature comparison and rapid keypoint generation makes it well-suited for short-baseline motion estimation where inter-frame transformations are small. Across these techniques, a common pipeline emerges: keypoint extraction, descriptor generation, feature matching, and outlier removal. The difference lies primarily in computational intensity and robustness to environmental variation. SIFT and SURF excel in high-fidelity mapping and long-term stability, whereas ORB provides lightweight, near-real-time capability for onboard processing. These algorithms collectively represent the evolution of image-based navigation, from robust but computationally heavy descriptors like SIFT, through SURF’s balance of speed and reliability, to ORB’s optimization for embedded autonomy, underscoring that the optimal choice depends on platform constraints, mission requirements, and environmental dynamics.

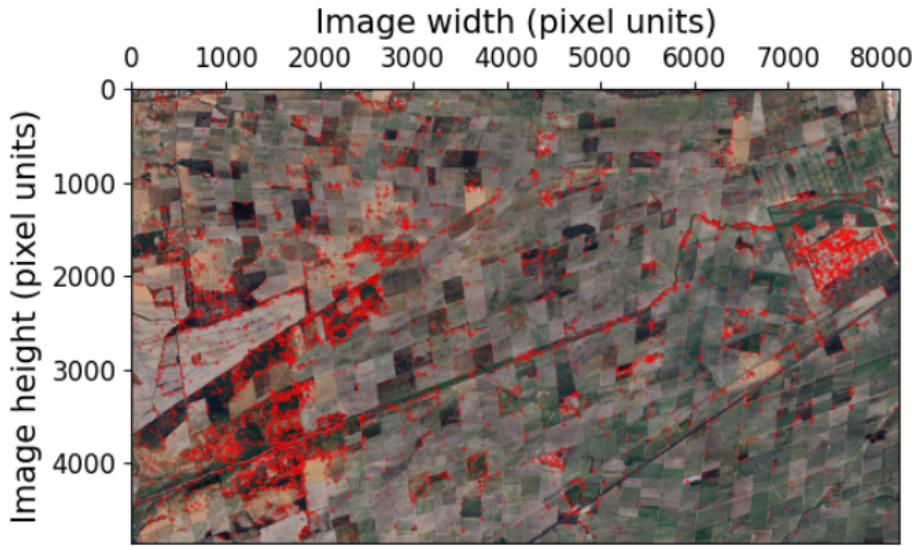


Figure 2.13: Feature detection using ORB [12].

2.4.2.2 STATE ESTIMATION

State estimation techniques for satellite attitude determination have evolved to balance accuracy, robustness, and computational efficiency. Bamber et al. [47] introduces a high-performance geometric inversion method that relates sub-pixel displacements in Earth observation imagery to the satellite's attitude, constructing the Direction Cosine Matrix for full three-axis determination. This deterministic approach delivers highly accurate short-term estimates without recursive filtering or process noise modeling, particularly suitable for high-resolution optical payloads.

In contrast, Caballero et al. [48] presents a two-stage vision-based framework combining Homography-Based Visual Odometry with SLAM. Local motion is estimated from consecutive frames under a planar scene assumption, using robust statistical methods and Singular Value Decomposition to extract rotation and translation. These local estimates are then integrated in a SLAM framework to mitigate cumulative drift, demonstrating the strength of relative motion estimation even without absolute sensors. Building on homography concepts, Caballero et al. [48] uses an Extended Kalman Filter (EKF) to optimize inter-image alignment for mosaic building. By re-estimating homographies during loop closures, the EKF reduces accumulated drift and incorporates measurement uncertainty, illustrating how stochastic filtering can enhance long-term consistency in visual systems.

Robust filtering approaches include the Robust Unscented Kalman Filter (RUKF) [49], which adaptively tunes the measurement noise covariance to detect and compensate for faulty measurements in real time, offering stability under degraded sensor conditions.

Similarly, Carmi et al. [50] proposes an Adaptive Genetic Algorithm-Embedded Quaternion Particle Filter (GA-QPF), representing attitude with weighted particles and estimating gyro biases via a Maximum Likelihood process. This probabilistic, non-parametric approach avoids Gaussian assumptions, maintains quaternion normalization, and is highly robust to nonlinear spacecraft dynamics.

Overall, these studies highlight a spectrum of estimation philosophies: deterministic geometric inversion ([47]) provides high instantaneous accuracy, visual odometry and SLAM [48] enable relative motion estimation with drift correction, EKF-based refinement [48] ensures long-term consistency, and adaptive stochastic methods ([49, 50]) prioritize robustness under uncertainty. Together, they illustrate the trade-offs between short-term precision, temporal continuity, and resilience in satellite attitude and state estimation.

2.5 CONCLUSION

From the literature studied, it is clear that pose estimation is an ongoing topic of interest. A brief summary of the concepts is covered as illustrated in Figure 2.14

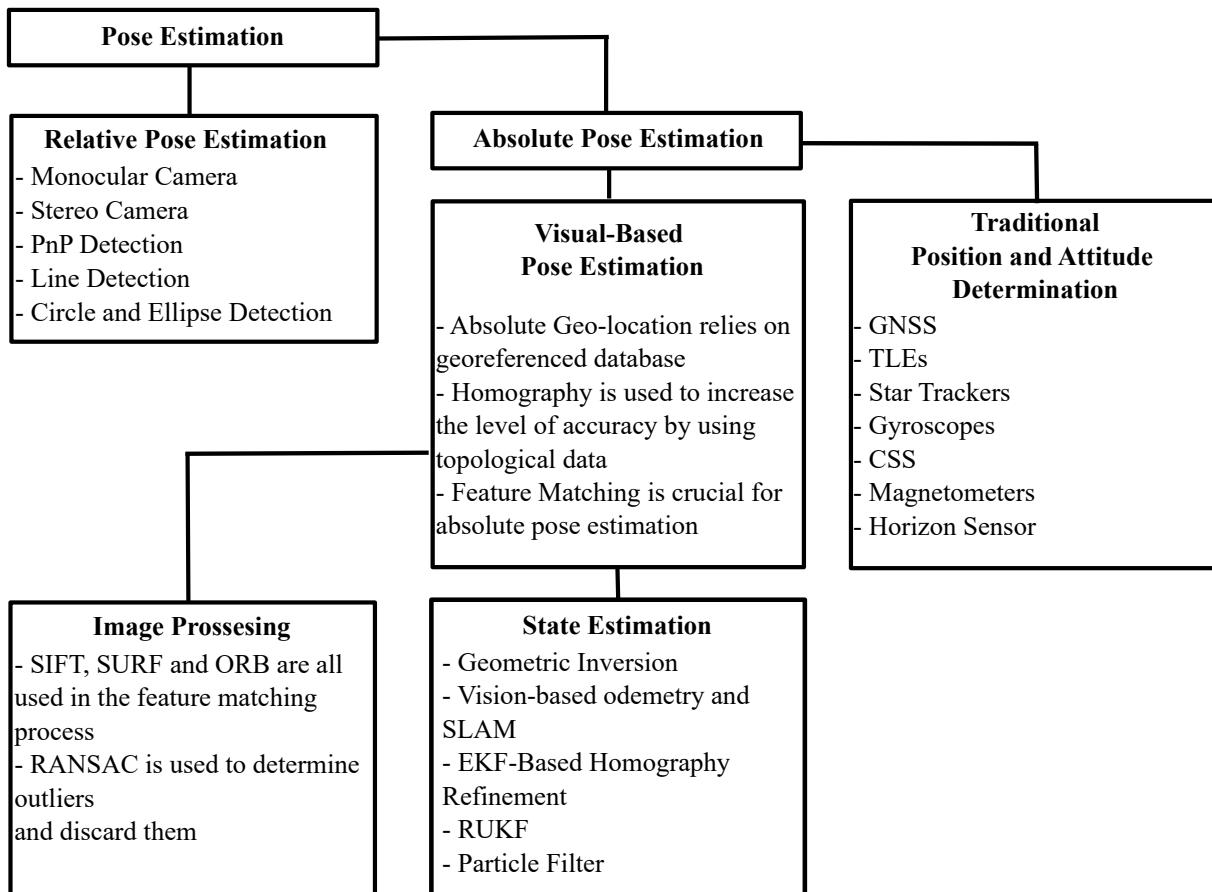


Figure 2.14: Literature study framework and concepts learned out of each section.

The literature demonstrates a steady evolution in satellite attitude and position estimation, moving from traditional sensor-based approaches toward autonomous, vision-driven systems. This shift has been accelerated by the miniaturization of satellite technology, which demands lightweight, low-power alternatives to conventional navigation sensors.

In relative pose estimation, both monocular and stereo camera systems have been widely used. Stereo systems directly recover depth through image disparity, while monocular systems infer motion by tracking features across image sequences. These approaches are particularly valuable for close-proximity operations such as formation flying and rendezvous, where real-time visual feedback can substitute for heavier range or radar sensors.

Absolute pose estimation traditionally depends on star trackers, sun sensors, and magnetometers to provide high-precision attitude data. However, these instruments can be costly and resource-intensive. Recent vision-based methods instead use georeferenced maps, Earth observation images, and online databases to match observed terrain features with known ground references, allowing satellites to infer global orientation and position visually.

In vision-based navigation, effective image processing and feature extraction are critical. Algorithms such as SIFT, SURF, and ORB are frequently employed to detect, describe, and match features between onboard imagery and reference datasets. To ensure robustness against mismatches and outliers, the RANSAC algorithm is often applied during the feature matching stage to refine correspondences and estimate the underlying geometric transformation. The combination of these methods provides reliable feature associations that are invariant to illumination, scale, and rotation, making them highly suitable for spaceborne imaging conditions.

For state estimation and sensor fusion, the literature highlights the extensive use of probabilistic filters such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). These frameworks fuse measurements from multiple sensors, including cameras, gyroscopes, accelerometers, and magnetometers, to produce continuous, noise-tolerant estimates of attitude and angular velocity.

Overall, the reviewed studies reveal a convergence between computer vision, sensor fusion, and nanosatellite technology. This convergence is enabling compact satellites to achieve accurate, autonomous navigation without heavy reliance on traditional sensors. The insights gained from these works form the foundation for the proposed methodology developed in the next chapter.

CHAPTER 3

MODELLING

3.1 INTRODUCTION

To accurately describe the full pose of a satellite using an onboard imaging system, a comprehensive mathematical model of the spacecraft must be established. This chapter presents the fundamental modeling framework required for vision-based satellite pose estimation. The chapter begins by defining the kinematic and dynamic equations governing satellite motion are derived, providing the mathematical foundation for state propagation. The various reference frames utilized throughout this work are systematically defined, including the transformations necessary to relate measurements and states across different coordinate systems. Additionally, this chapter presents the mathematical models for the accompanying sensors integrated within the pose estimation system. These sensor models are essential for a multi-sensor fusion approach.

3.2 RIGID BODY MECHANICS

3.2.1 KINEMATICS

The pose of a rigid body within a reference frame encompasses both its spatial position and angular orientation. The attitude describes the rotational relationship between the body-fixed coordinate system and a known reference coordinate system. This rotational relationship is typically expressed through a rotation matrix, commonly known as a direction cosine matrix (DCM)[33, 51–53]. Elementary rotations around individual coordinate axes are termed coordinate rotations. The fundamental coordinate rotations about the \bar{x} , \bar{y} , and \bar{z} -axes, characterized by rotation angles ϕ , θ , and ψ respectively, can be mathematically expressed as

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad (3.1)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \text{ and} \quad (3.2)$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

Any rotation in 3D space can be described by three coordinate rotations. The DCM describing the rotation from the orbital reference frame \mathcal{O} to the body reference frame \mathcal{B} , $\mathbf{A}_{\mathcal{O}}^{\mathcal{B}}$, can be represented by three Euler angles. Each of the angles corresponds to one coordinate rotation. The order of the Euler 3-2-1 or a Yaw, Pitch, Roll rotation, shown in Figure 3.1, is expressed as

$$\begin{aligned} \mathbf{A}_{\mathcal{O}}^{\mathcal{B}} &= R_z(\psi)R_y(\theta)R_x(\phi) \\ &= \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \\ &= \begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ S\phi S\theta C\psi - C\phi S\psi & S\phi S\theta S\psi + C\phi C\psi & S\phi C\theta \\ C\phi S\theta C\psi + S\phi S\psi & C\phi S\theta S\psi - S\phi C\psi & C\phi C\theta \end{bmatrix} \end{aligned} \quad (3.4)$$

Where S is the sine function and C is the cosine function. The Euler angles are calculated as follows:

$$\phi = \arctan 2 \left(\frac{a_{2,3}}{a_{3,3}} \right), \quad (3.5)$$

$$\theta = \arctan 2 \left(\frac{-a_{1,3}}{\sqrt{a_{1,1}^2 + a_{1,2}^2}} \right), \text{ and} \quad (3.6)$$

$$\psi = \arctan 2 \left(\frac{a_{1,2}}{a_{1,1}} \right). \quad (3.7)$$

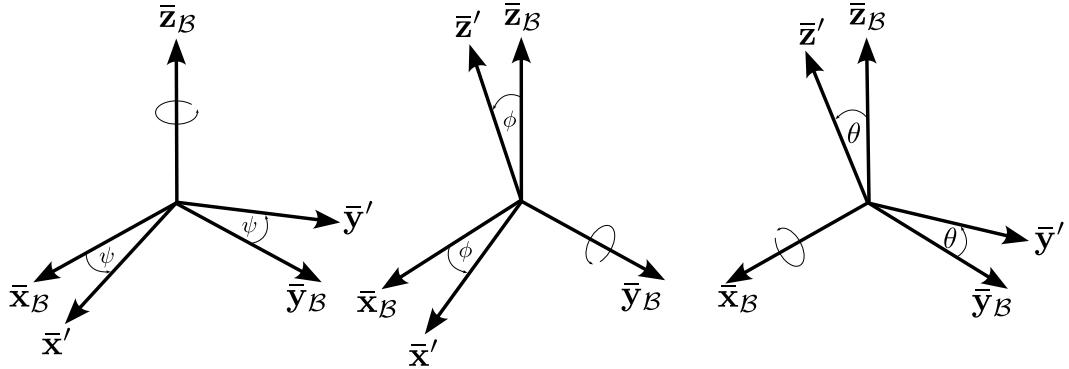


Figure 3.1: The Euler 3-2-1 rotation sequence (yaw-pitch-roll) describes a 3D orientation.

Mathematical singularities occur when using Euler angles to represent large rotations. When both $a_{1,1}$ and $a_{1,2}$ in Equations 3.6 and 3.7 are zero, the expressions for ψ and θ are undefined. This is known as *gimbal lock*, where the changes in the first and third Euler angles are indistinguishable when the second angle nears a critical value.

Alternatively, the DCM can be described using quaternions, which do not have these singularities. The quaternion rotation in Figure 3.2 is expressed by a Euler axis $\bar{\mathbf{e}} = [e_x, e_y, e_z]^T$ and an angle θ

$$\mathbf{q} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ e_x \sin(\theta/2) \\ e_y \sin(\theta/2) \\ e_z \sin(\theta/2) \end{bmatrix} \quad (3.8)$$

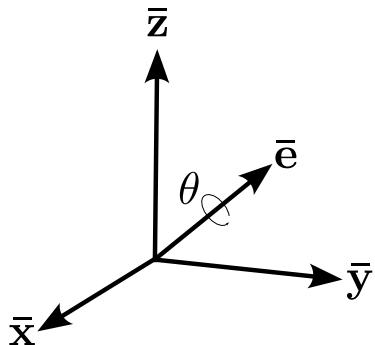


Figure 3.2: Quaterion rotation represented by the rotation axis $\bar{\mathbf{e}}$ and rotation angle θ .

The DCM as a function of quaternion set is expressed as,

$$\mathbf{A}_{\mathcal{O}}^{\mathcal{B}} = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_s q_z) & 2(q_x q_z + q_s q_y) \\ 2(q_x q_y + q_s q_z) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_s q_x) \\ 2(q_x q_z - q_s q_y) & 2(q_y q_z + q_s q_x) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \quad (3.9)$$

Using the normalisation constraint, $q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1$, the DCM simplifies to,

$$\mathbf{A}_{\mathcal{O}}^{\mathcal{B}} = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_s q_z) & 2(q_x q_z + q_s q_y) \\ 2(q_x q_y + q_s q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_s q_x) \\ 2(q_x q_z - q_s q_y) & 2(q_y q_z + q_s q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix}. \quad (3.10)$$

The body-fixed angular rates of the satellite relative to \mathcal{O} , $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O}}$, can also be expressed as a function of quaternions by,

$$\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O}} = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = 2 \begin{bmatrix} -q_x & q_s & -q_z & q_y \\ -q_y & q_z & q_s & -q_x \\ -q_z & -q_y & q_x & q_s \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix}. \quad (3.11)$$

Inversely the quaternion rates as a function of the body rates are,

$$\begin{bmatrix} \dot{q}_s \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{bx} & -\omega_{by} & -\omega_{bz} \\ \omega_{bx} & 0 & \omega_{bz} & -\omega_{by} \\ \omega_{by} & -\omega_{bz} & 0 & \omega_{bx} \\ \omega_{bz} & \omega_{by} & -\omega_{bx} & 0 \end{bmatrix} \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix}. \quad (3.12)$$

Throughout this work, quaternions serve as the primary attitude representation method. Quaternions eliminate rotational sequence ambiguities and define rotations about a clearly specified axis. The trigonometric components of the rotation matrix are inherently embedded within the quaternion-based DCM formulation. Consequently, attitude transformations require only a single matrix operation using quaternions, whereas Euler angle representations necessitate three separate operations.

3.2.2 DYNAMICS

This section delves into the dynamics governing the motion of a satellite, which are fundamentally separated into translational dynamics and rotational dynamics. The satellite's linear motion in orbit is described by Newton's second law, with the discrete-time propagation equations relating position, velocity, and the resultant external forces. For LEO modeling over short periods, the primary forces considered are the Earth's gravitational force (\mathbf{F}_G) and the J_2 perturbation (\mathbf{F}_{J2}), which accounts for the Earth's oblateness. The rotational motion of the satellite, representing its attitude, is governed by the Newton-Euler equations for a rigid body. In the absence of external torques, this

motion is simplified to Euler's equations, which relate the change in angular velocity to the satellite's moments of inertia.

3.2.2.1 TRANSLATIONAL DYNAMICS

Newton's second law governs the linear motion of the satellite with mass m . The discrete-time position and velocity propagation equations are:

$$\mathbf{r}(t) = \mathbf{r}(t-1) + \mathbf{v}(t)\Delta t + \frac{1}{2m}\mathbf{F}(t)\Delta t^2, \quad (3.13)$$

$$\mathbf{v}(t) = \mathbf{v}(t-1) + \frac{1}{m}\mathbf{F}(t)\Delta t, \quad (3.14)$$

where \mathbf{F} denotes the resultant external force applied to the satellite. Given the potential unavailability of precise mass characteristics, the translational dynamics may be adequately represented through kinematic approximations wherein the instantaneous velocity is predominantly governed by the preceding velocity state.

In orbital mechanics, the gravitational force \mathbf{F} acting on a spacecraft is proportional to its position vector \mathbf{r} relative to the center of mass. This force is described by

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{F}\mathbf{u}_r \\ &= \frac{-\mu}{\|\mathbf{r}\|^3}\mathbf{u}_r \end{aligned} \quad (3.15)$$

Where μ denotes Earth's gravitational parameter.

The acceleration of a spacecraft in orbit is influenced by several factors, including the primary gravitational force \mathbf{F}_G , the J_2 perturbation \mathbf{F}_{J2} , atmospheric drag \mathbf{F}_{drag} , solar radiation pressure \mathbf{F}_{sol} , and other miscellaneous perturbations \mathbf{F}_{misc} .

The total force acting on the satellite can therefore be expressed as:

$$\mathbf{F}_{total} = \mathbf{F}_G + \mathbf{F}_{J2} + \mathbf{F}_{drag} + \mathbf{F}_{sol} + \mathbf{F}_{misc} \quad (3.16)$$

However, since we are only modeling Low Earth Orbits (LEO) over a short time period, only the primary gravitational force \mathbf{F}_G and the J_2 perturbation \mathbf{F}_{J2} are considered.

The J_2 perturbation, which accounts for the Earth's oblateness, is modeled in the ECI frame using

$$\mathbf{a}_{J_2} = \frac{3}{2} J_2 \frac{\mu R_E}{\|\mathbf{r}\|^5} \begin{bmatrix} r_x(1 - 5\frac{z^2}{r^2}) \\ r_y(1 - 5\frac{z^2}{r^2}) \\ r_z(3 - 5\frac{z^2}{r^2}) \end{bmatrix} \quad (3.17)$$

In Equation 3.17, μ represents Earth's gravitational parameter, R_E is the mean radius of the Earth, J_2 is the second zonal harmonic coefficient accounting for the Earth's oblateness, and \mathbf{r} denotes the satellite's position vector expressed in the ECI frame.

Then the full propagation turns into:

$$\ddot{\mathbf{r}} = \mathbf{a}_G + \mathbf{a}_{J_2}, \quad (3.18)$$

which is illustrated in Figure 3.3 as the position and velocity changes after one full orbit at an altitude of 500km at an inclination of 48.8° . The Figure 3.4 succeding it, illustrates the error that the J_2 perturbations introduces to the position and velocity of the satellite.

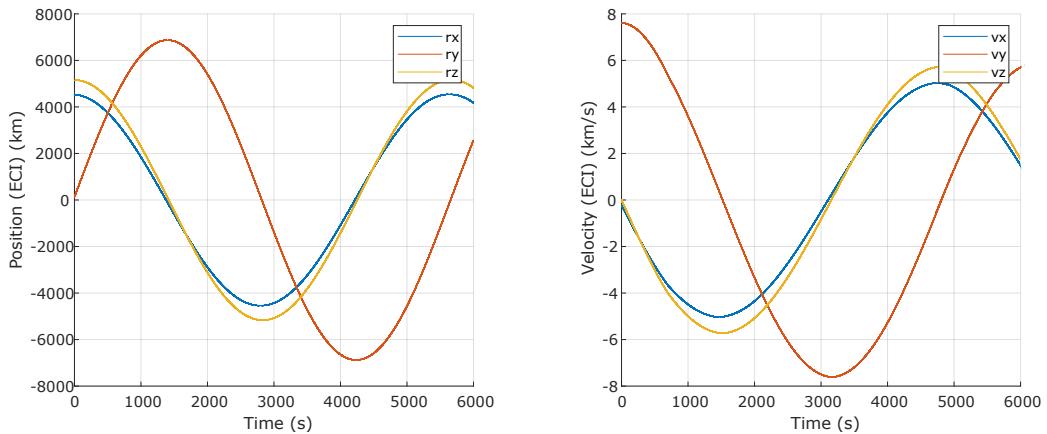


Figure 3.3: The real position and real velocity of the satellite at an altitude of 500km and an inclination of 48.8° , illustrating the change in position and velocity overtime governed by the gravitational kinematics.

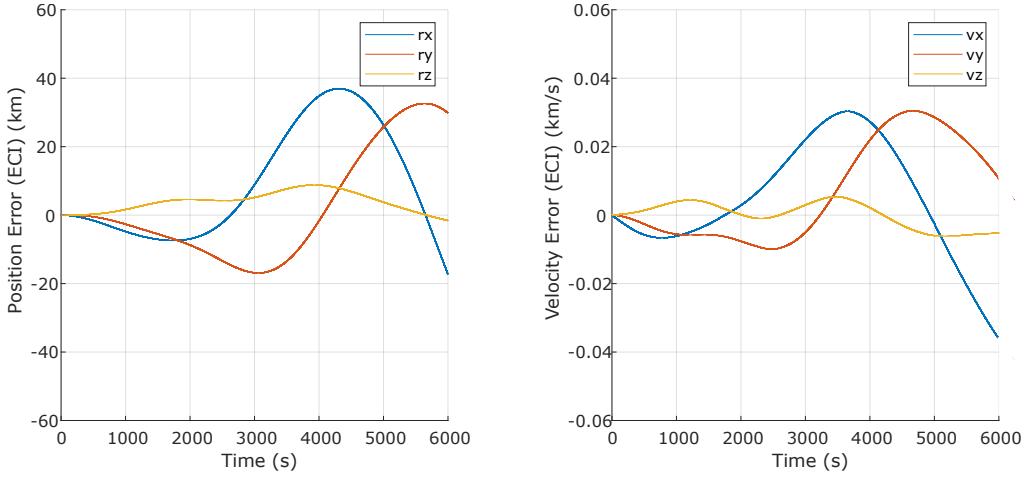


Figure 3.4: The J2 error introduced into system, illustrating its effects on the position and velocity.

3.2.2.2 ROTATIONAL DYNAMICS

The rotational dynamics of a rigid satellite are governed by the Newton-Euler equations, which apply to all rigid inertial bodies. The angular momentum \mathbf{H} of the satellite is expressed as:

$$\dot{\mathbf{H}} = \frac{d\mathbf{H}}{dt} = \mathbf{I}\dot{\boldsymbol{\omega}}, \quad (3.19)$$

where \mathbf{H} is the angular momentum vector, $\boldsymbol{\omega}$ is the angular velocity vector expressed in the body frame, and \mathbf{I} is the diagonalized moment of inertia tensor about the satellite's principal axes. This is a assumption as \mathbf{I} can have crossproduct terms.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.20)$$

In the absence of external torques, the rotational motion about the satellite's center of mass can be described by [54]:

$$I_{xx}\dot{\omega}_x = \omega_y\omega_z(I_{yy} - I_{zz}), \quad (3.21)$$

$$I_{yy}\dot{\omega}_y = \omega_x\omega_z(I_{zz} - I_{xx}), \quad (3.22)$$

$$I_{zz}\dot{\omega}_z = \omega_x\omega_y(I_{xx} - I_{yy}) \quad (3.23)$$

where I_{xx} , I_{yy} , and I_{zz} are the principal moments of inertia, which are constant and determined by the satellite's mass distribution and geometry.

The stability of the satellite's rotational motion is influenced by its moment of inertia. Rotation about the major or minor principal axis is stable, while rotation about the intermediate axis is inherently unstable. Under constant energy conditions, any initial rotation around the intermediate axis will tend to redistribute energy toward the major and minor axes due to nutation effects which is referred to as the Dzhanibekov effect as illustrated in Figure 3.5.

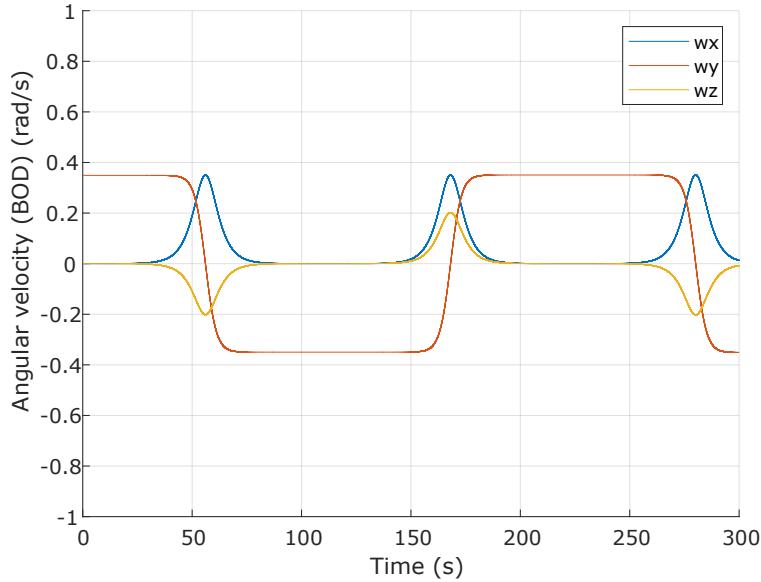


Figure 3.5: The Dzhanibekov effect to verify the dynamics.

To propagate the satellite's attitude over time, the quaternion derivative must be computed. The quaternion $\mathbf{q}_{B/O}$, which represents the rotation of the body frame \mathcal{B} relative to the orbital frame \mathcal{O} , evolves according to:

$$\dot{\mathbf{q}}_{B/O} = \frac{1}{2}(\mathbf{q}_{B/O} \otimes \boldsymbol{\omega}) \quad (3.24)$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity vector in the body frame, and \otimes denotes quaternion multiplication. Expanding this yields:

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{bx} & -\omega_{by} & -\omega_{bz} \\ \omega_{bx} & 0 & \omega_{bz} & -\omega_{by} \\ \omega_{by} & -\omega_{bz} & 0 & \omega_{bx} \\ \omega_{bz} & \omega_{by} & -\omega_{bx} & 0 \end{bmatrix} \mathbf{q} \quad (3.25)$$

where we can see this is the same as the equation we derived in Equation 3.12.

Quaternion propagation is performed using a simple Euler integration scheme. First, the quaternion is advanced in time as:

$$\bar{\mathbf{q}}_{B/I}(t + \Delta t) = \mathbf{q}_{B/I}(t) + \dot{\mathbf{q}}_{B/I}\Delta t \quad (3.26)$$

where $\bar{\mathbf{q}}_{B/I}$ is the unnormalized quaternion. To maintain a valid attitude representation, the quaternion must be renormalized:

$$\mathbf{q}_{B/I}(t + \Delta t) = \frac{\bar{\mathbf{q}}_{B/I}(t + \Delta t)}{||\bar{\mathbf{q}}_{B/I}(t + \Delta t)||} \quad (3.27)$$

This normalization step ensures that the quaternion maintains unit magnitude throughout integration.

In the Figure 3.6 it illustrates that the quaternions change with a constant rotation and is normalised to ensure a valid quaternion representation, it also shows that variations of the inertia tensor \mathbf{I} introduces a coupling effect into the system.

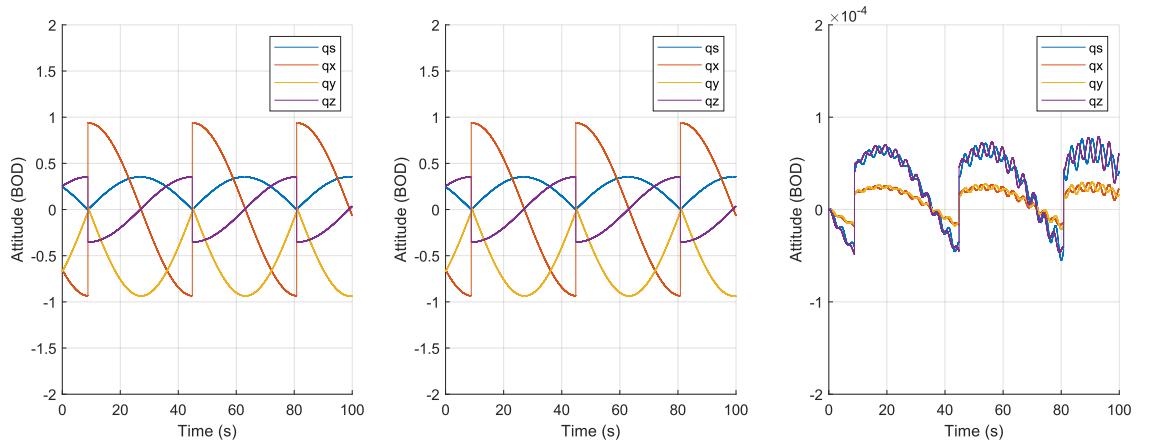


Figure 3.6: Illustration of the attitude coupling effect arising from differences in the spacecraft's principal moments of inertia. Variations in these inertial properties can cause rotations about one axis to induce secondary motions about the other axes, demonstrating the intrinsic coupling between rotational dynamics and the spacecraft's mass distribution.

3.3 REFERENCE FRAME TRANSFORMATIONS

In this work, several different reference frames will be used. To accurately construct the measurement model, it is essential to understand each of these reference frames and the transformations between them.

3.3.1 TRANSFORMATION MATRIX

Transformations between different reference frames are a fundamental part of spacecraft modeling and sensor simulation. These transformations are typically expressed using homogeneous transformation matrices, which combine both rotation and translation components into a single 4×4 matrix.

3.3.1.1 ROTATION MATRIX

In many practical scenarios, such as pure attitude transformations or orbit-to-body frame conversions, only rotational alignment is needed. In such cases, only the DCM, \mathbf{A} , is used like discussed in Section 3.2.1, and the transformation is defined as:

$$\mathbf{v}_B = \mathbf{A}_O^B \cdot \mathbf{v}_O \quad (3.28)$$

Note that:

$$\mathbf{A}_O^B = (\mathbf{A}_B^O)^T \quad (3.29)$$

and that the reverse rotation can be applied.

$$\mathbf{v}_B = (\mathbf{A}_B^O)^T \times \mathbf{v}_O \quad (3.30)$$

This relationship arises because DCM's are orthogonal.

3.3.1.2 TRANSLATION

In many reference frame transformations, it is necessary to account for the translation of the origin when converting from one frame to another. This relationship can be expressed as:

$$\mathbf{v}_B = \mathbf{v}_{B/I} - \mathbf{r}_I \quad (3.31)$$

In this formulation, $\mathbf{v}_{B/I}$ represents the position vector of a point expressed in the body frame B relative to the inertial frame I . The term \mathbf{v}_B denotes the same point expressed purely in the body frame B , while \mathbf{r}_I corresponds to the position of the origin of the body frame with respect to the inertial frame I . Together, these terms define how the translation of the origin between frames is accounted for when transforming position vectors from one reference frame to another.

Alternatively, this translation can be expressed in matrix form, as will be illustrated in the following section.

3.3.1.3 HOMOGENEOUS TRANSFORMATION MATRIX

A homogeneous transformation matrix from frame \mathcal{I} to frame \mathcal{O} is defined as:

$$\mathbf{T}_{\mathcal{I}}^{\mathcal{O}} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}}^{\mathcal{O}} & -\mathbf{A}_{\mathcal{I}}^{\mathcal{O}} \cdot \mathbf{r}_{\mathcal{I}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.32)$$

Here $\mathbf{A}_{\mathcal{I}}^{\mathcal{O}}$ is a 3×3 direction cosine matrix (DCM) describing the orientation of frame \mathcal{O} relative to frame \mathcal{I} and $\mathbf{r}_{\mathcal{I}}$ is the translation vector from the origin of \mathcal{I} to the origin of \mathcal{O} , expressed in frame \mathcal{I} .

To apply this transformation, the position vector must be expressed in homogeneous coordinates by augmenting it with an additional element:

$$\mathbf{r}_{\mathcal{I}}^+ = \begin{bmatrix} \mathbf{r}_{\mathcal{I}} \\ 1 \end{bmatrix} \quad (3.33)$$

The superscript “+” indicates the homogeneous representation of the vector. This formulation allows both rotation and translation to be applied through a single matrix operation.

$$\mathbf{r}_{\mathcal{O}}^+ = \mathbf{T}_{\mathcal{I}}^{\mathcal{O}} \times \mathbf{r}_{\mathcal{I}}^+ \quad (3.34)$$

As shown in Equation 3.34, a single homogeneous transformation matrix can simultaneously perform both rotation and translation. It is important to note that the inverse of a homogeneous transformation matrix is **not** simply its transpose as we say in the case of the DCMs:

$$\mathbf{T}_{\mathcal{I}}^{\mathcal{O}} \neq (\mathbf{T}_{\mathcal{O}}^{\mathcal{I}})^T \quad (3.35)$$

3.3.2 LATITUDE, LONGITUDE, AND ALTITUDE

The position of a feature or a satellite on Earth is commonly described using latitude, longitude, and altitude, collectively denoted as \mathcal{L} :

$$\mathbf{r}_{\mathcal{L}} = \begin{bmatrix} \lambda \\ \phi \\ h \end{bmatrix} \quad (3.36)$$

Latitude (ϕ) specifies the angular distance of a point north or south of the equator, ranging from -90° at the South Pole to 90° at the North Pole. Longitude (λ) defines the angular distance east or west of the Prime Meridian, which passes through Greenwich, and varies

between -180° and 180° . Altitude (h) indicates the height of the point above a chosen reference surface, such as an ellipsoid, geoid, or digital elevation model.

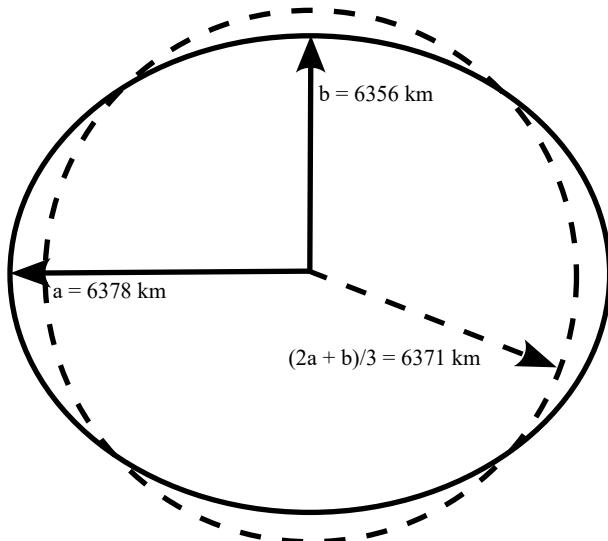


Figure 3.7: The WGS84 reference ellipsoid, a mathematically defined model of the Earth's shape used as the basis for the World Geodetic System 1984 (WGS84). It approximates the Earth as an oblate spheroid with a defined semi-major and semi-minor axis, providing a standard reference for geodetic measurements, satellite positioning, and global navigation systems.

The simplest reference for altitude is the WGS84 ellipsoid as illustrated in Figure 3.7, which approximates the Earth's shape as an oblate spheroid. This mathematical model is widely used in navigation and satellite applications due to its simplicity and global consistency. However, it does not account for the Earth's non-uniform mass distribution or variations in gravity.

To represent the Earth's shape more accurately, the geoid is introduced. The geoid is an equipotential surface of the Earth's gravity field that corresponds to the mean sea level extended beneath the continents. Unlike the ellipsoid, the geoid reflects gravitational anomalies caused by mountains, ocean trenches, and mantle density variations. As such, it provides a more physically meaningful reference for elevation measurements, although it still ignores local terrain variations as illustrated in Figure 3.8.

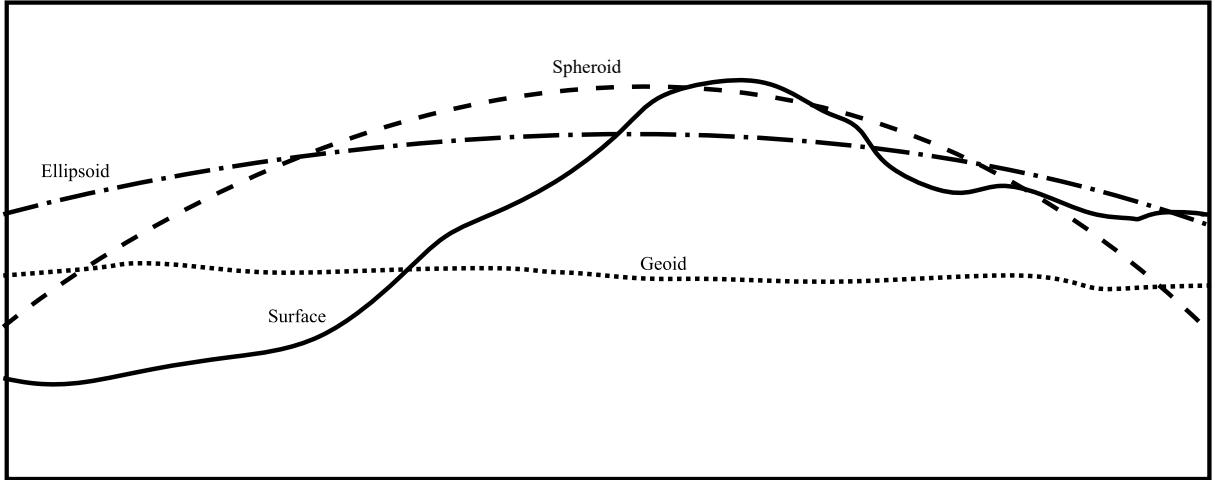


Figure 3.8: Comparison of different Earth models: the perfect sphere, the reference ellipsoid, and the geoid. The spherical model is a simple approximation with constant radius, the ellipsoid accounts for the Earth’s flattening at the poles and bulging at the equator, and the geoid represents the irregular shape of the Earth’s gravity field, closely following mean sea level. This comparison highlights the increasing accuracy and complexity of models used in geodesy, navigation, and Earth observation.

For applications requiring high precision, such as satellite image simulation or georeferencing, a Digital Elevation Model (DEM) is employed. A DEM incorporates detailed topographic information, including mountains, valleys, and other local features, at high spatial resolution as seen in Figure 3.9. While the geoid accounts for gravity-based variations, the DEM further enhances accuracy by modeling the actual terrain surface relative to the reference ellipsoid or geoid. This level of detail is essential for tasks such as line-of-sight calculations, remote sensing, and terrain-aware navigation.

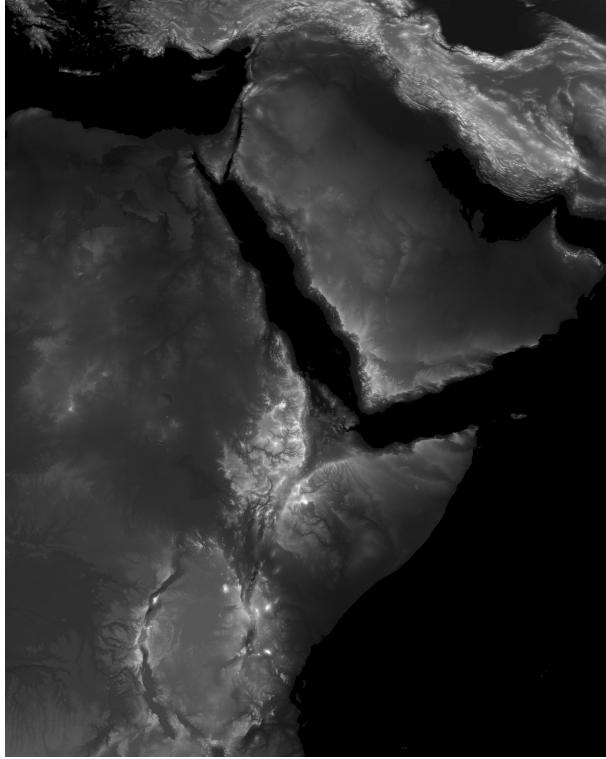


Figure 3.9: A Digital Elevation Model (DEM) of the African continent, generated using USGS software. This DEM provides a detailed representation of the terrain’s elevation, capturing variations in topography such as mountains, valleys, and plateaus, and is widely used in geospatial analysis, environmental modeling, and Earth observation studies.

Unfortunately, Digital Elevation Models (DEMs) are extremely large datasets, often reaching several gigabytes depending on their spatial resolution and coverage area. This makes them impractical for onboard storage and real-time processing on small satellites, which typically have strict limitations on memory, processing power, and data handling capabilities. Incorporating DEMs into onboard algorithms would significantly increase system complexity and resource demands. Therefore, to simplify the implementation for this project, the WGS84 reference ellipsoid will be used as the primary Earth model. Despite being a simplified geometric approximation, WGS84 is widely adopted in global navigation satellite systems (GNSS) and aerospace applications due to its balance of accuracy, computational efficiency, and universal acceptance. This choice enables consistent and efficient onboard calculations while maintaining an acceptable level of positional accuracy for the intended mission.

3.3.3 EARTH CENTERED EARTH FIXED

The Earth-Centered Earth-Fixed (ECEF) frame, also referred to as the Earth-Centered Rotating (ECR) frame, is denoted by \mathcal{R} . This frame is similar to the geodetic frame \mathcal{L} but is expressed in a Cartesian coordinate system. The \bar{z} -axis is aligned with the

Earth's rotation axis, passing through the North Pole, while the \bar{x} -axis intersects the Prime Meridian at the Equator. The \bar{y} -axis completes the right-hand rule, resulting in a fixed frame that rotates with the Earth.

Conversion from \mathcal{L} (latitude, longitude, altitude) to \mathcal{R} (ECEF) is typically performed using the WGS84 transformation. It models the Earth as an oblate spheroid with a semi-major axis of 6,378 km and a flattening factor of 1/298.257223563.

$$\mathbf{r}_{\mathcal{R}} = f(\text{WGS84}, \mathbf{r}_{\mathcal{L}}) \quad (3.37)$$

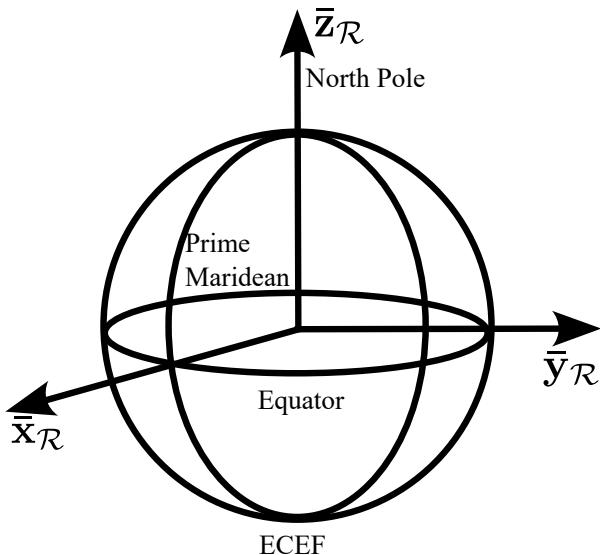


Figure 3.10: The Earth-Centered, Earth-Fixed (ECEF) reference frame, a Cartesian coordinate system fixed relative to the Earth, with its origin at the Earth's center of mass. The \bar{x} -axis points toward the intersection of the equator and the prime meridian, the \bar{y} -axis points 90° east of the X-axis along the equator, and the \bar{z} -axis aligns with the Earth's rotational axis toward the North Pole. This frame is commonly used for satellite navigation, geodesy, and Earth observation applications.

3.3.4 EARTH CENTERED INERTIAL

The Earth-Centered Inertial (ECI) reference frame, denoted by \mathcal{I} , shares its origin with the Earth-Centered Earth-Fixed (ECEF) frame but differs in orientation due to the Earth's rotation. While the \bar{z} -axis remains aligned with the Earth's rotation axis (pointing toward the North Pole), the \bar{x} -axis of the ECI frame points toward the Vernal Equinox—the intersection of the Earth's equatorial plane and the ecliptic plane—while the \bar{y} -axis completes the right-hand rule. Unlike the ECEF frame, which rotates with the Earth, the ECI frame is non-rotating with respect to the distant stars, making it suitable for orbital dynamics and space navigation. The transformation from ECEF to ECI is governed

by Earth's rotation about its \bar{z} -axis at an angular velocity of $\omega_e = 7.2921 \times 10^{-5}$ rad/s, resulting in a rotation of $\theta = \omega_e t$ over time t .

Closely related to the ECI frame is the International Celestial Reference Frame (ICRF), particularly the J2000 realization, which serves as a quasi-inertial reference frame for astronomical and space applications. The ICRF is defined by the positions of distant extragalactic radio sources, such as quasars, which are so far away that they appear fixed relative to Earth's motion, providing an extremely stable reference. The J2000 frame is aligned with the Earth's mean equator and equinox at the epoch J2000.0 (January 1, 2000, 12:00 TT) and is the standard frame for expressing the positions and motions of celestial bodies, satellite orbits, and spacecraft attitudes, especially in deep-space navigation.

To transform from the ECEF to the ECI frame the following DCM is constructed by

$$\mathbf{A}_{\mathcal{R}}^{\mathcal{I}} = R(\omega_e t) = \begin{bmatrix} \cos(-\omega_e t) & -\sin(-\omega_e t) & 0 \\ \sin(-\omega_e t) & \cos(-\omega_e t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and} \quad (3.38)$$

$$\mathbf{T}_{\mathcal{R}}^{\mathcal{I}} = \begin{bmatrix} \mathbf{A}_{\mathcal{R}}^{\mathcal{I}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.39)$$

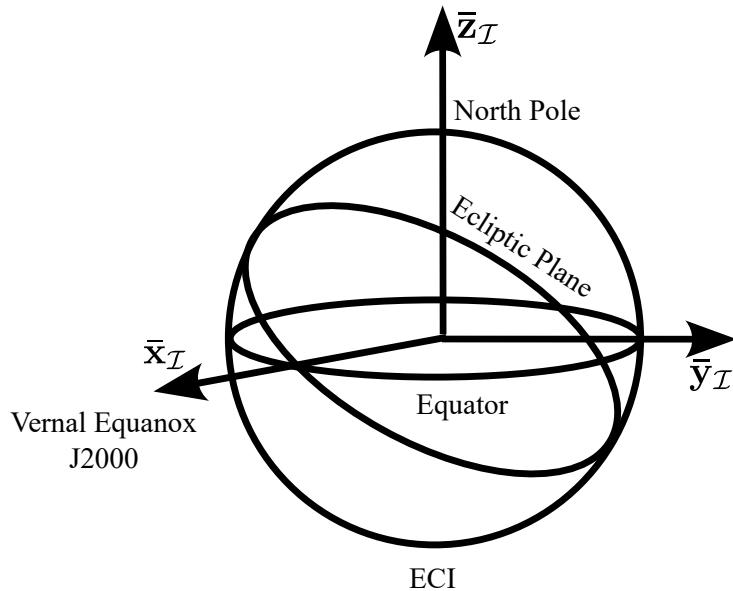


Figure 3.11: The Earth-Centered Inertial (ECI) reference frame, an inertial Cartesian coordinate system with its origin at the Earth's center of mass. Unlike Earth-fixed frames, the ECI frame does not rotate with the Earth, making it suitable for describing the motion of satellites, spacecraft, and other celestial objects relative to the stars. It provides a consistent reference for orbital mechanics and space navigation.

3.3.5 ORBITAL REFERENCE FRAME

The orbital reference frame used is the Local Vertical Local Horizon (LVLH) denoted by \mathcal{O} . The LVLH frame is a rotating, orbit-attached coordinate system commonly used in spacecraft dynamics. It moves with the satellite and is defined relative to its orbit around Earth. The \bar{z} -axis is the local vertical and is also called the Nadir direction, it points to the barycenter of the system, in this case the center of the Earth. The \bar{y} -axis is called the cross track that points out of the orbital plane, typically the anti-angular momentum vector direction (anti-normal to the orbit plane). The \bar{x} -axis is the "Local Horizon" also called "along track" pointing forward it is tangent to the orbit and completes the right hand rule.

If $\mathbf{r}_{\mathcal{I}}$ is the position vector of the satellite and $\mathbf{v}_{\mathcal{I}}$ is the velocity vector of the satellite. The reference frame unit vectors are determined by:

$$\bar{\mathbf{z}}_{\mathcal{O}} = -\frac{\mathbf{r}_{\mathcal{I}}}{\|\mathbf{r}_{\mathcal{I}}\|}, \quad (3.40)$$

$$\bar{\mathbf{y}}_{\mathcal{O}} = \frac{\mathbf{r}_{\mathcal{I}} \times \mathbf{v}_{\mathcal{I}}}{\|\mathbf{r}_{\mathcal{I}} \times \mathbf{v}_{\mathcal{I}}\|}, \text{ and} \quad (3.41)$$

$$\bar{\mathbf{x}}_{\mathcal{O}} = \bar{\mathbf{y}}_{\mathcal{O}} \times \bar{\mathbf{z}}_{\mathcal{O}} \quad (3.42)$$

This reference frame also requires a translation and is performed by

$$\mathbf{A}_{\mathcal{I}}^{\mathcal{O}} = \begin{bmatrix} \bar{\mathbf{x}}_{\mathcal{O}} \\ \bar{\mathbf{y}}_{\mathcal{O}} \\ \bar{\mathbf{z}}_{\mathcal{O}} \end{bmatrix}, \text{ and} \quad (3.43)$$

$$\mathbf{T}_{\mathcal{I}}^{\mathcal{O}} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}}^{\mathcal{O}} & -\mathbf{A}_{\mathcal{I}}^{\mathcal{O}} \mathbf{r}_{\mathcal{I}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.44)$$

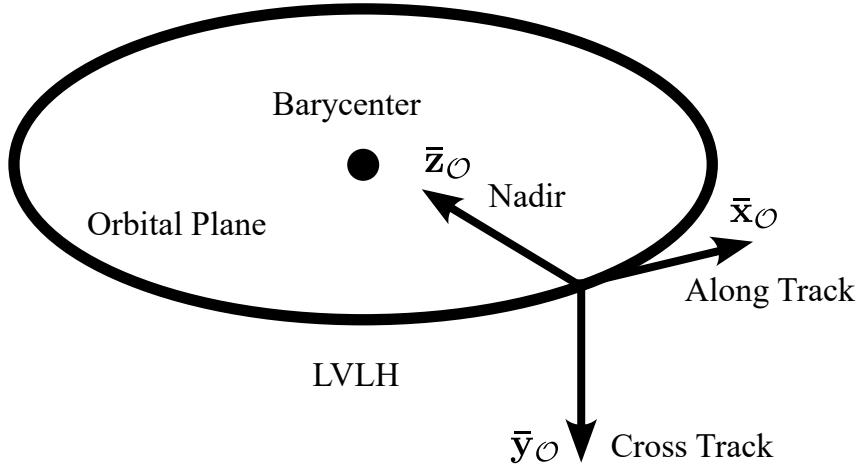


Figure 3.12: The Orbital Reference Frame, also known as the Local Vertical Local Horizontal (LVLH) frame, is a spacecraft-centered coordinate system used to describe positions and orientations relative to an orbiting body. Its origin is at the spacecraft's center of mass, the \bar{z} -axis points toward the center of the Earth (nadir), the \bar{y} -axis is aligned opposite to the orbital angular momentum vector (orbit normal), and the \bar{x} -axis completes the right-handed system, pointing along the velocity vector in the orbital plane. This frame is widely used in attitude control, guidance, and navigation for satellites.

3.3.6 BODY REFERENCE FRAME

The body reference frame denoted by \mathcal{B} is the reference frame of the satellite body itself, with the center point referenced as the center of mass of the satellite body with the \bar{z} -axis defined as the yaw, \bar{x} -axis defined as the roll and the \bar{y} -axis defined as the pitch of the satellite as illustrated in Figure 3.13. Converting from one reference frame to another a DCM needs to be constructed from the true attitude $\mathbf{q}_{\mathcal{B}/\mathcal{O}}$ like in Equation 3.10

$$\mathbf{A}_{\mathcal{O}}^{\mathcal{B}} = f(\mathbf{q}_{\mathcal{B}/\mathcal{O}}), \text{ and} \quad (3.45)$$

$$\mathbf{T}_{\mathcal{O}}^{\mathcal{B}} = \begin{bmatrix} \mathbf{A}_{\mathcal{O}}^{\mathcal{B}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (3.46)$$

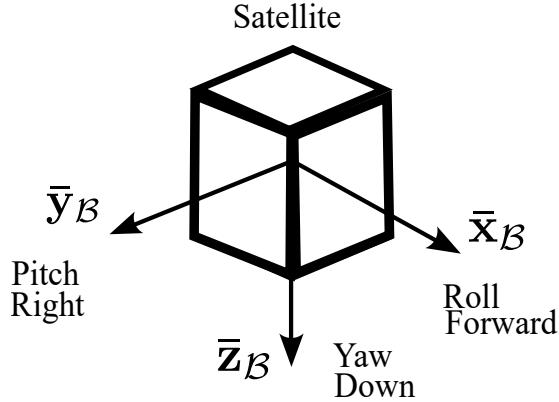


Figure 3.13: The Body Reference Frame, denoted by \mathcal{B} , is fixed to the satellite itself, with its origin at the satellite's center of mass. The axes are defined relative to the satellite's orientation: the \bar{x} -axis corresponds to roll, the \bar{y} -axis to pitch, and the \bar{z} -axis to yaw.

3.3.7 EAST - NORTH - DOWN/ EAST - NORTH - UP

The North-East-Down (NED) and East-North-Up (ENU) reference frames are not prominent reference frames in this work, but it is worth a mention as it is used in sensor modeling and system initialization. Both frames are defined at a specific location by the geodetic latitude ϕ , longitude λ , and altitude h . The ENU frame has axes pointing East, North, and Up, whereas the NED frame has axes pointing North, East, and Down as we can see in Figure 3.14. The transformation from the Earth-Centered Earth-Fixed (ECEF) frame to the ENU frame is:

$$\mathbf{A}_{\mathcal{R}}^{\mathcal{E}} = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix}. \quad (3.47)$$

Similarly, the transformation from the ECEF frame to the NED frame is:

$$\mathbf{A}_{\mathcal{R}}^{\mathcal{N}} = \begin{bmatrix} -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \lambda & \sin \lambda & 0 \\ -\cos \phi \cos \lambda & -\cos \phi \sin \lambda & -\sin \phi \end{bmatrix}. \quad (3.48)$$

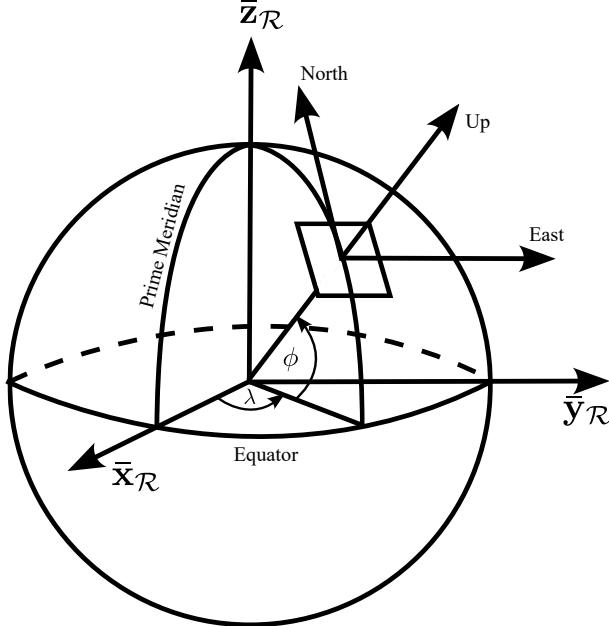


Figure 3.14: The East-North-Up reference frame

3.4 SENSOR MODELLING

To properly model the satellite sensors, it is necessary to define the **truth state vector** \mathbf{x}_{true} , which represents the complete, ideal state of the satellite. All sensor measurements are simulated relative to this true state. The truth state includes the satellite's position and velocity in the Earth-Centered Inertial (ECI) frame, its orientation as a quaternion describing the rotation from the orbital frame to the body frame, and its angular velocity relative to the orbital frame:

$$\begin{aligned}\mathbf{r}_{\mathcal{I},\text{true}} &= [x_{\text{position}}, y_{\text{position}}, z_{\text{position}}]^T, \\ \mathbf{v}_{\mathcal{I},\text{true}} &= [x_{\text{velocity}}, y_{\text{velocity}}, z_{\text{velocity}}]^T, \\ \mathbf{q}_{\mathcal{B}/\mathcal{O},\text{true}} &= [s_{\text{attitude}}, x_{\text{attitude}}, y_{\text{attitude}}, z_{\text{attitude}}]^T, \\ \boldsymbol{\omega}_{\mathcal{B}/\mathcal{O},\text{true}} &= [x_{\text{angular velocity}}, y_{\text{angular velocity}}, z_{\text{angular velocity}}]^T.\end{aligned}\tag{3.49}$$

The full truth state vector is then composed as

$$\mathbf{x}_{\text{true}} = [\mathbf{r}_{\mathcal{I},\text{true}}, \mathbf{v}_{\mathcal{I},\text{true}}, \mathbf{q}_{\mathcal{B}/\mathcal{O},\text{true}}, \boldsymbol{\omega}_{\mathcal{B}/\mathcal{O},\text{true}}]^T.\tag{3.50}$$

Here, $\mathbf{r}_{\mathcal{I},\text{true}}$ and $\mathbf{v}_{\mathcal{I},\text{true}}$ describe the satellite's exact position and velocity in the ECI frame, $\mathbf{q}_{\mathcal{B}/\mathcal{O},\text{true}}$ specifies its true orientation as a quaternion, and $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O},\text{true}}$ represents the angular velocity of the satellite body relative to the orbital reference frame.

3.4.1 GPS SENSOR MODEL

In simulation, GPS measurements are generated using the same underlying orbital dynamics as the true satellite model, based on the two-body problem. This ensures consistency between the satellite's true motion and its simulated sensor data. To emulate realistic sensor characteristics, Gaussian noise and drift are introduced into the measurements.

The GPS measurement model is given by:

$$\mathbf{z}_{GPS,\mathcal{R}}(t) = \mathbf{r}_{\mathcal{R},\text{true}}(t) + \boldsymbol{\eta}_{GPS}(t) + \mathbf{d}_{GPS}(t), \quad (3.51)$$

where $\mathbf{z}_{GPS}(t)$ is the measured position at time t . It is expressed as the sum of the true satellite position $\mathbf{r}_{\mathcal{R},\text{true}}(t)$, propagated through the two-body dynamics, additive zero-mean Gaussian noise $\boldsymbol{\eta}_{GPS}(t)$, and a drift term $\mathbf{d}_{GPS}(t)$ accounting for long-term bias.

$$\mathbf{r}_{\mathcal{R},\text{true}}^+ = \mathbf{T}_{\mathcal{I}}^{\mathcal{R}} \times \mathbf{r}_{\mathcal{I},\text{true}}^+ \quad (3.52)$$

The drift term models the slow, cumulative error typical of low-cost GPS receivers. It follows a random walk process:

$$\mathbf{d}_{GPS}(t) = \mathbf{d}_{GPS}(t - \Delta t) + \mathbf{q}_{GPS}(t), \quad (3.53)$$

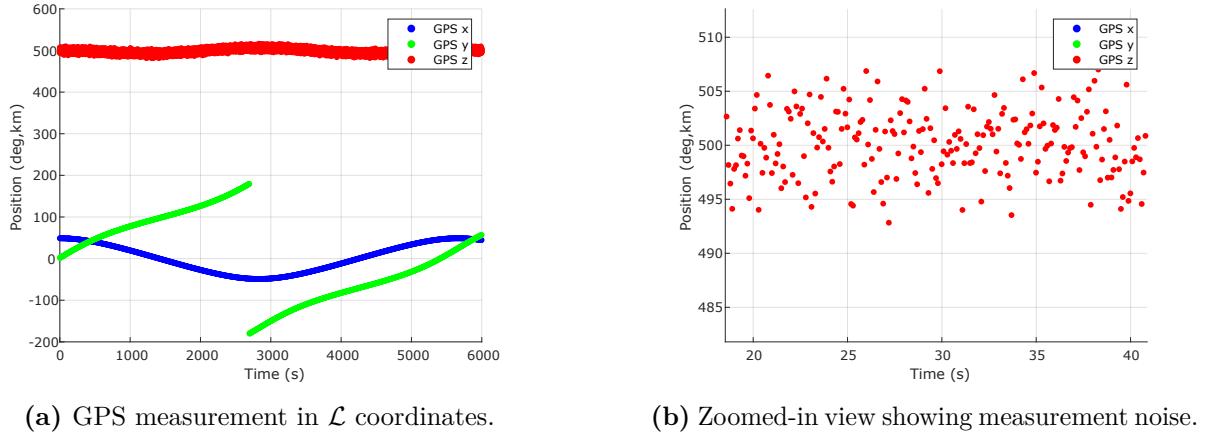
where $\mathbf{d}_{GPS}(t - \Delta t)$ is the previous drift value, and $\mathbf{q}_{GPS}(t)$ is a mean-zero random increment representing the drift rate, typically sampled from a Gaussian distribution:

$$\mathbf{q}_{GPS}(t) \sim \mathcal{N}(0, \sigma_d^2 \mathbf{I}). \quad (3.54)$$

This formulation captures both short-term variability through $\boldsymbol{\eta}_{GPS}(t)$ and long-term bias accumulation via $\mathbf{d}_{GPS}(t)$, resulting in a more realistic sensor simulation for estimation performance testing.

GPS outputs are typically provided in standard NMEA message formats such as the \$GPGGA sentence, which includes time, latitude, longitude, and altitude. To produce equivalent measurements in simulation, the ECEF-frame output $\mathbf{z}_{GPS,\mathcal{R}}(t)$ is transformed to geodetic coordinates \mathcal{L} (latitude, longitude, altitude) using the WGS84 reference model, as shown in Figure 3.15. The resulting groundtrack visualization is given in Figure 3.16.

$$\mathbf{z}_{GPS,\mathcal{L}}(t) = f(\text{WGS84}, \mathbf{z}_{GPS,\mathcal{R}}(t)) \quad (3.55)$$


 (a) GPS measurement in \mathcal{L} coordinates.

(b) Zoomed-in view showing measurement noise.

Figure 3.15: GPS measurement and noise model derived from the \mathbf{x}_{true} state in the \mathcal{L} reference frame.

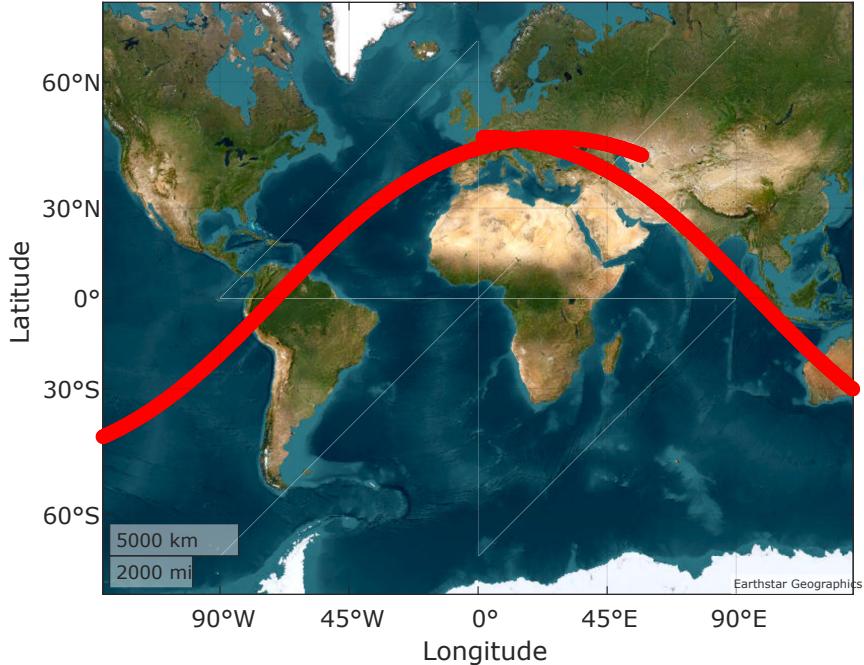


Figure 3.16: Groundtrack representation of the simulated GPS measurements, illustrating the apparent Earth rotation as the satellite completes an orbit.

3.4.2 GYROSCOPE SENSOR MODEL

The gyroscope measures the angular velocity of the satellite body frame relative to the inertial frame, expressed in the body frame, denoted as $\boldsymbol{\omega}_{B/I}^B$. This measurement is essential for attitude determination and control.

To reflect realistic performance, the simulated gyroscope output includes both random noise and time-varying bias (drift), modeled as:

$$\mathbf{z}_{\text{Gyro}}(t) = \boldsymbol{\omega}_{B/I}^B(t) + \boldsymbol{\eta}_{\text{Gyro}}(t) + \mathbf{d}_{\text{Gyro}}(t), \quad (3.56)$$

where $\mathbf{z}_{\text{Gyro}}(t)$ is the measured angular velocity, $\boldsymbol{\eta}_{\text{Gyro}}(t)$ represents zero-mean Gaussian noise, and $\mathbf{d}_{\text{Gyro}}(t)$ models the slow drift bias typical of gyroscopes.

The true angular velocity $\boldsymbol{\omega}_{B/I}$ can be obtained from the simulated state \mathbf{x}_{true} , which contains $\boldsymbol{\omega}_{B/O}$. To relate these quantities, the orbital angular velocity $\boldsymbol{\omega}_{O/I}$ must first be determined:

$$\boldsymbol{\omega}_{O/I}^O = -\omega_o \bar{\mathbf{y}}_O, \quad \omega_o = \sqrt{\frac{\mu}{r^3}}, \quad (3.57)$$

where ω_o is the orbital rate, dependent only on orbital radius for a circular orbit. Expressing this in the body frame gives:

$$\begin{aligned} \boldsymbol{\omega}_{O/I}^B &= \mathbf{A}_O^B \times \boldsymbol{\omega}_{O/I}^O \\ &= \begin{bmatrix} -a_{1,2}\omega_o \\ -a_{2,2}\omega_o \\ -a_{3,2}\omega_o \end{bmatrix}. \end{aligned} \quad (3.58)$$

The total body angular velocity is then obtained as:

$$\boldsymbol{\omega}_{B/I} = \boldsymbol{\omega}_{B/O} + \boldsymbol{\omega}_{O/I}. \quad (3.59)$$

The gyroscope bias evolves according to a random walk model:

$$\mathbf{d}_{\text{Gyro}}(t) = \mathbf{d}_{\text{Gyro}}(t - \Delta t) + \mathbf{q}_{\text{Gyro}}(t), \quad (3.60)$$

where $\mathbf{q}_{\text{Gyro}}(t)$ is a random increment drawn from:

$$\mathbf{q}_{\text{Gyro}}(t) \sim \mathcal{N}(0, \sigma_g^2 \mathbf{I}), \quad (3.61)$$

with σ_g^2 denoting the variance of the gyroscope drift rate.

This model effectively represents both high-frequency noise and long-term bias drift, enabling realistic simulation of Inertial Measurement Unit (IMU) performance. Incorporating these dynamics into the estimation process supports robust attitude and rate reconstruction in the presence of sensor imperfections.

Figure 3.17 illustrates the simulated gyroscope output, showing rotation about the body z -axis at approximately $10^\circ/s$, with the combined effects of measurement noise and sampling rate visible in the resulting signal variance.

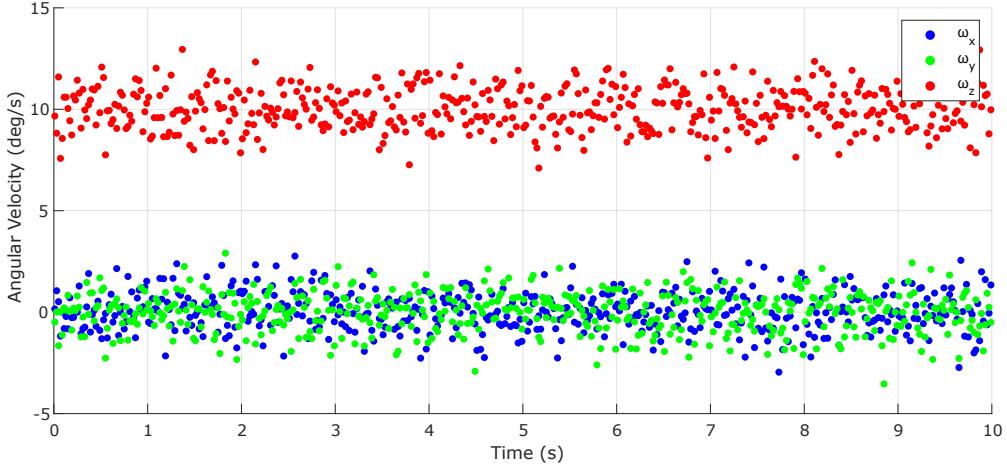


Figure 3.17: Simulated gyroscope measurement showing high-frequency noise and drift.

3.4.3 COARSE SUN SENSOR MODEL

The Coarse Sun Sensor (CSS) is a fundamental attitude sensing instrument in nanosatellite systems, providing an estimate of the Sun direction relative to the satellite's body frame.

The Sun vector is modeled in the Earth-Centered Inertial (ECI) frame as a fixed unit vector pointing in the **X+** direction:

$$\mathbf{S}_{\mathcal{I}} = [1 \ 0 \ 0]^\top \quad (3.62)$$

This simplified model assumes that the Sun direction does not vary during the simulation.

To simulate the Sun vector in the satellite's body frame, the inertial vector is rotated using the satellite's true attitude quaternion. The corresponding direction cosine matrix (DCM) is derived as:

$$\mathbf{S}_{\mathcal{B}}^+ = \mathbf{T}_{\mathcal{O}}^{\mathcal{B}} \times \mathbf{T}_{\mathcal{I}}^{\mathcal{O}} \times \mathbf{S}_{\mathcal{I}}^+ \quad (3.63)$$

where $\mathbf{T}_{\mathcal{I}}^{\mathcal{O}}$ is the transformation matrix from the inertial to orbit frame and $\mathbf{T}_{\mathcal{O}}^{\mathcal{B}}$ is the transformation from the orbital frame to the body frame.

The CubeSat is equipped with six coarse sun sensors, one on each face, aligned along the $\pm X$, $\pm Y$, and $\pm Z$ body axes (see Figure 3.18). Each sensor has a cosine response:

$$z_i = \max(0, \hat{\mathbf{n}}_i^\top \mathbf{S}_B), \quad i = 1, \dots, 6 \quad (3.64)$$

where $\hat{\mathbf{n}}_i$ is the normal vector of the i -th sensor surface.

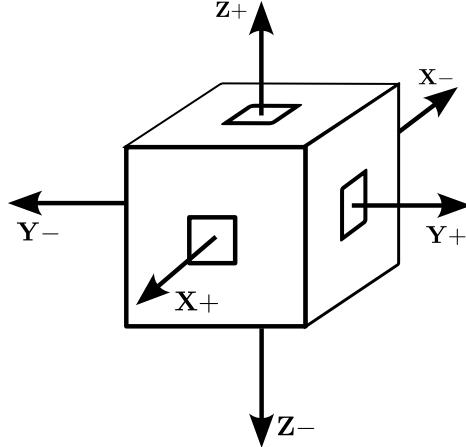


Figure 3.18: Orientation of the six coarse sun sensors on the CubeSat

Each CSS reading is perturbed, where $\boldsymbol{\eta}_{\text{CSS}}$ is a zero-mean Gaussian noise vector with covariance \mathbf{R}_{CSS} :

$$\mathbf{z}_{\text{CSS}} = \max(0, \mathbf{z}_{\text{CSS}} + \boldsymbol{\eta}_{\text{CSS}}) \quad (3.65)$$

$$\boldsymbol{\eta}_{\text{CSS}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\text{CSS}}). \quad (3.66)$$

Negative readings are clamped to zero since physical sensors cannot detect negative intensity.

The estimated Sun vector in the body frame is reconstructed using a weighted sum of the face normals:

$$\hat{\mathbf{S}}_B = \sum_{i=1}^6 z_i \cdot \hat{\mathbf{n}}_i \quad (3.67)$$

The result is normalized to produce a unit vector:

$$\hat{\mathbf{S}}_B = \frac{\hat{\mathbf{S}}_B}{|\hat{\mathbf{S}}_B|} \quad (3.68)$$

In practice, this sensor is limited by its inability to measure the Sun vector during eclipse conditions, as it requires direct visibility of the Sun. However, for the purposes of this project, it is assumed that the Sun remains visible throughout the entire orbit.

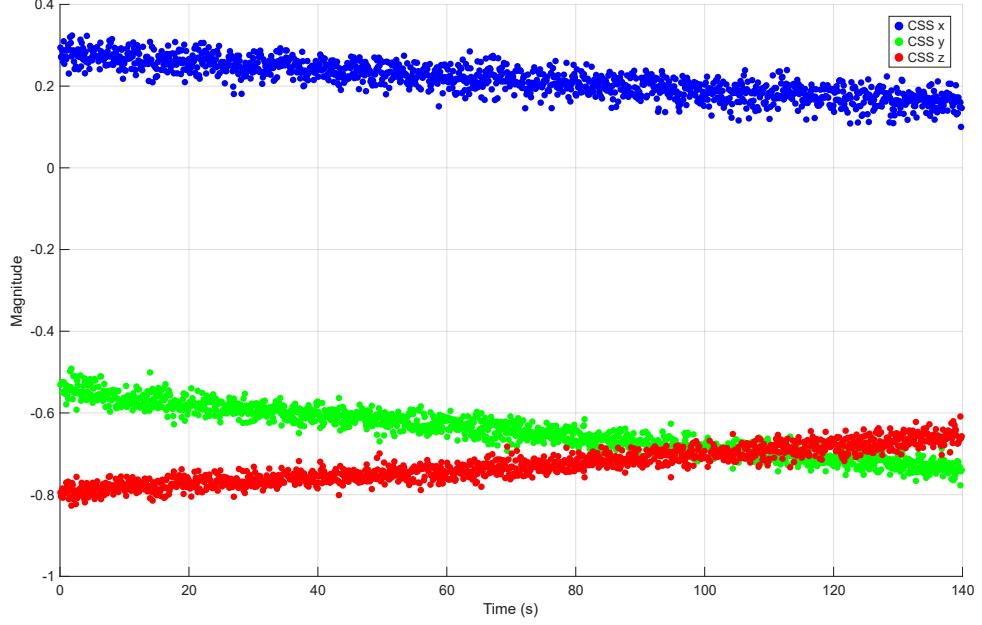


Figure 3.19: Simulated CSS measurement illustrating the magnitude of the direction of the sun and the noise.

3.4.4 STAR TRACKER SENSOR MODEL

The Star Tracker (ST) is a high-accuracy attitude sensor that provides an absolute measurement of the spacecraft's orientation. The way a star tracker works is by capturing an image of a star field. This star field produces a pointing vector to a star in the body frame.

If we define a star catalogue in the ECI frame as

$$\boldsymbol{\Gamma} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\}, \quad (3.69)$$

where each $\mathbf{s}_i \in \mathbb{R}^3$ is a unit vector pointing towards a known star.

The measurement in the body reference frame is given by

$$\mathbf{s}_{i,\mathcal{B}}^+ = \mathbf{T}_{\mathcal{O}}^{\mathcal{B}} \times \mathbf{T}_{\mathcal{I}}^{\mathcal{O}} \times \mathbf{s}_{i,\mathcal{I}}^+, \quad (3.70)$$

where $\mathbf{T}_{\mathcal{O}}^{\mathcal{B}}$ and $\mathbf{T}_{\mathcal{I}}^{\mathcal{O}}$ are transformation matrices dependent on the true spacecraft state \mathbf{x}_{true} . Figure 3.20 illustrated how each star measurement is seen out of the BRF of the satellite

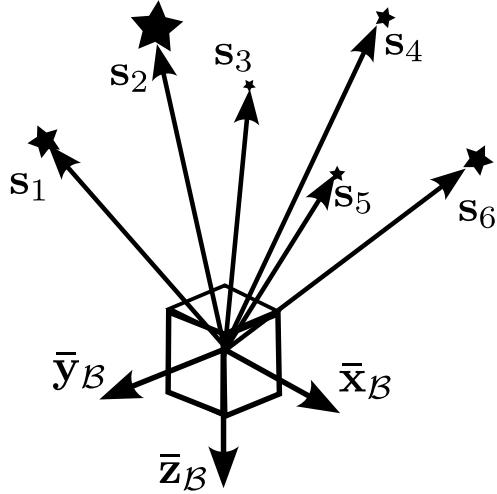


Figure 3.20: Representation of star vectors calculated out of the center of mass of the satellite, each star vector is a unit vector defined in the BRF.

The observed star vectors are corrupted by zero-mean Gaussian noise, affecting each component of the vector measurement independently:

$$\mathbf{z}_{\text{ST},i}(t) = \mathbf{s}_{i,B}(t) + \boldsymbol{\eta}_{\text{ST},i}(t), \quad (3.71)$$

where $\mathbf{z}_{\text{ST},i}(t)$ is the measured star vector at time t for the i^{th} star, and $\boldsymbol{\eta}_{\text{ST},i}(t)$ is a zero-mean Gaussian noise vector with covariance \mathbf{R}_{ST} :

$$\boldsymbol{\eta}_{\text{ST},i}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\text{ST}}). \quad (3.72)$$

This noise accounts for short-term measurement errors due to sensor resolution, optical distortions, and image processing uncertainties. Unlike gyroscope measurements, the star tracker noise does not include a bias or drift term, as star trackers provide absolute attitude references. Figure 3.21 shows the simulated measurements of the first four stars in Γ and illustrates how the direction changes as the satellite orbits.

Normally, a star tracker is constrained to the eclipse section of the orbit, as it can only detect stars most precisely in darkness, but in this project it is assumed that the Star Tracker has a 360° FOV and can see throughout the entire orbit.

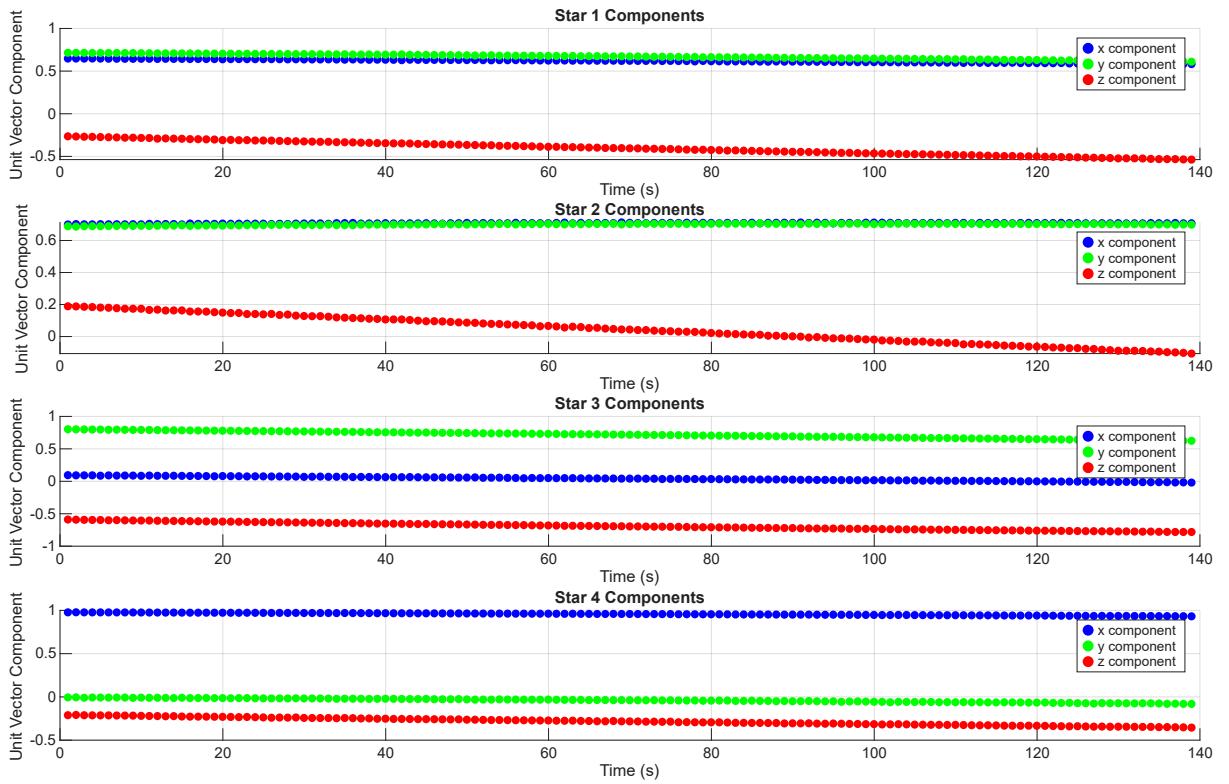


Figure 3.21: Simulated Star Tracker Measurements from the first 4 Stars, the figure shows how the unit vectors (direction) changes as the satellite orbits.

3.4.5 MAGNETOMETER SENSOR MODEL

The spacecraft's magnetometer measures the local magnetic field vector at its current position in space. This measurement provides both the direction and magnitude of the geomagnetic field, which is essential for attitude determination and control. For this project, the magnetic field is modeled using the International Geomagnetic Reference Field (IGRF) defined in the J2000 inertial reference frame. The IGRF is a standard mathematical representation of Earth's main magnetic field and its secular variation, commonly used in space applications to ensure consistency and accuracy in magnetic field estimation.

To get the magnetic field vector in ECI, the use of the matlab function *wrldmagm* in the Aerospace toolbox is used. It gives a value in the NED reference frame given a specific latitude longitude and altitude.

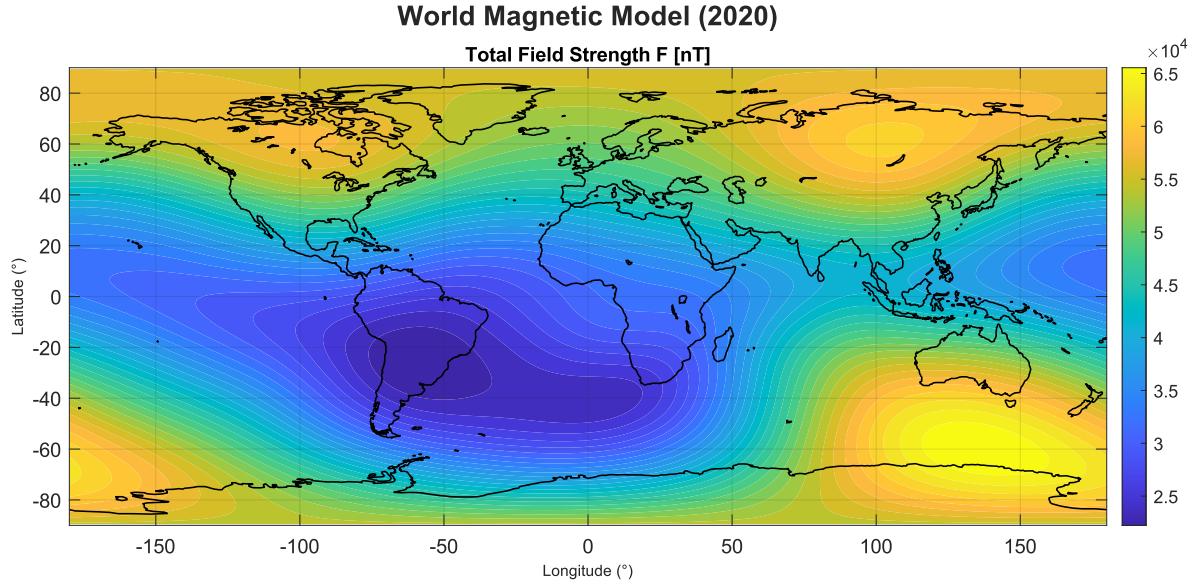


Figure 3.22: The total magnetic field intensity given by the *wrldmgn* function using IGRF.

so first we need to convert the satellites position to LLA using a DCM.

$$\begin{aligned} \mathbf{r}_{\mathcal{R}} &= \mathbf{A}_{\mathcal{I}}^{\mathcal{R}} \times \mathbf{r}_{\mathcal{I}} \\ \mathbf{r}_{\mathcal{L}} &= f(\text{WGS84}, \mathbf{r}_{\mathcal{R}}) \end{aligned} \quad (3.73)$$

The position is then provided as an input to the function.

$$\mathbf{z}_{\text{MAG},\mathcal{N}} = \text{wrldmagn}[\mathbf{r}_{\mathcal{L}}, \text{decimalYear}] \quad (3.74)$$

To get the magnetic field in the body frame we need to transform it with the transformation matrices.

$$\mathbf{z}_{\text{MAG},\mathcal{B}}^+ = \mathbf{T}_{\mathcal{O}}^{\mathcal{B}} \times \mathbf{T}_{\mathcal{I}}^{\mathcal{O}} \times \mathbf{T}_{\mathcal{R}}^{\mathcal{I}} \times \mathbf{T}_{\mathcal{N}}^{\mathcal{R}} \times \mathbf{z}_{\text{MAG},\mathcal{N}}^+ \quad (3.75)$$

The magnetometer measurement is affected by zero-mean Gaussian noise representing sensor inaccuracies such as electronic noise and environmental disturbances. The noisy measurement in the body frame can be expressed as

$$\mathbf{z}_{\text{MAG},\mathcal{B}}(t) = \mathbf{z}_{\text{MAG},\mathcal{B}}^+(t) + \boldsymbol{\eta}_{\text{MAG}}(t), \quad (3.76)$$

where the $\boldsymbol{\eta}_{\text{MAG}}(t)$ is a zero-mean Gaussian noise vector with covariance \mathbf{R}_{MAG} :

$$\boldsymbol{\eta}_{\text{MAG}}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\text{MAG}}). \quad (3.77)$$

This noise models the short-term fluctuations and measurement errors inherent to magnetometer sensors and does not include any long-term bias or drift, assuming that calibration and compensation are applied as illustrated in Figure 3.23.

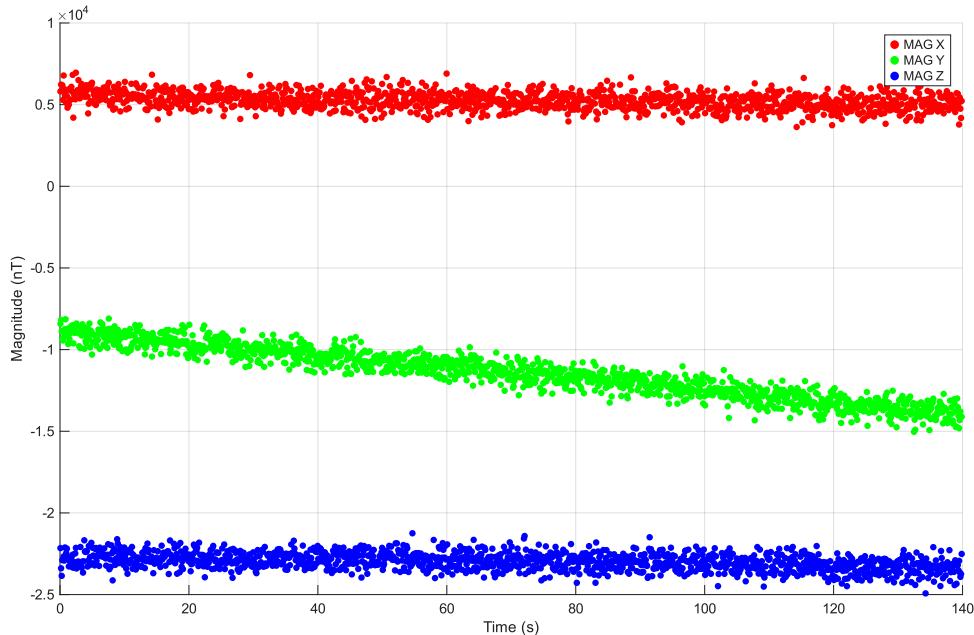


Figure 3.23: Simulated Magnetoemeter measurements indicating the magnetic field strength in nT and which direction it is measured in according to the IGRF.

3.5 CONCLUSION

This chapter presented the complete modelling framework of the satellite system, encompassing the orbital dynamics, attitude kinematics, and sensor representations. The two-body dynamics model provided the foundation for simulating the satellite's motion, while the attitude model described the rotational behaviour through quaternion-based propagation. Furthermore, the inclusion of sensor models such as the star tracker, magnetometer, accelerometer, and GPS enabled realistic measurement simulations that reflect the practical constraints of on-orbit systems. Together, these components established a coherent and physically consistent platform upon which the remainder of the study is built.

Beyond defining the mathematical relationships, the modelling process also clarified how system assumptions and simplifications influence downstream performance. The accuracy of these models directly affects the realism of the simulated imagery and the reliability of the estimated states. In this context, the modelling stage functions not merely as a theoretical formulation, but as a crucial factor influencing the accuracy and interpretability of the subsequent results.

3.5. CONCLUSION

The next chapter, Image Processing (Chapter 4), builds upon this framework to extract spatial and temporal information from the simulated imagery, while State Estimation (Chapter 5) integrates these processed measurements into a unified estimation architecture.

CHAPTER 4

IMAGE PROCESSING

4.1 INTRODUCTION

After modelling the satellite and the auxiliary sensors in Chapter 3, Chapter 4 delves deeper into the theory and methodology used to generate the Earth Tracker measurements. Firstly, the basic theoretical background of satellite imaging is discussed, including image geometry, different scanning techniques, and typical satellite imagery noise. This is followed by a discussion of the simplified camera and lens models used in this work, such as the intrinsic and extrinsic camera parameters and common lens distortions. Finally, the simulator implementation of the Earth Tracker is presented, along with the methodology of how it produces measurements for the pose estimator.

4.2 SATELLITE IMAGE CHARACTERISTICS

4.2.1 IMAGING GEOMETRY

All satellite images share a common set of geometric attributes, illustrated in Figure 4.1. The pixel pitch, denoted as p , refers to the physical size of a single pixel element on the imaging sensor (for example, one CMOS detector element). The sensor width, s , represents the total width of the sensor array and is calculated as:

$$s = I \times p$$

where I is the number of pixels along one dimension of the image (either horizontally or vertically) typically known as resolution. The swath width, S_w , is the total ground area captured by the image and is expressed as:

$$S_w = I \times GSD$$

where GSD (Ground Sampling Distance) defines the ground distance represented by each image pixel. Here, f denotes the focal length of the camera, and h the altitude of the satellite. The relationship between these parameters can be mathematically expressed as:

$$\frac{GSD}{p} = \frac{S_w}{s} = \frac{h}{f} \quad (4.1)$$

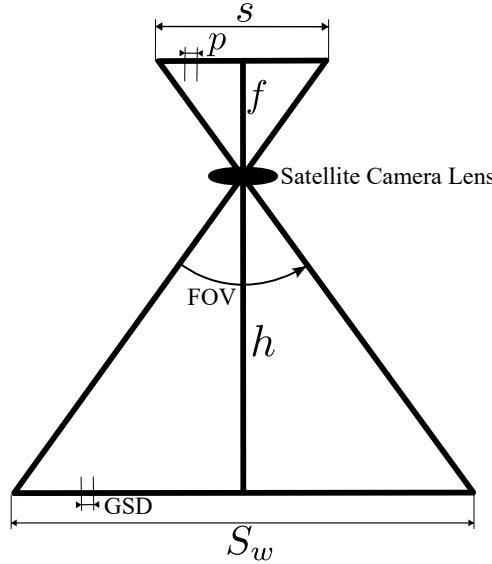


Figure 4.1: The relationship between sensor pitch, focal length, satellite altitude, and Ground Sampling Distance (GSD). The pixel pitch (p) determines the physical size of each sensor element, while the focal length (f) and satellite altitude (h) together define how this sensor geometry maps to the ground. A smaller pixel pitch or a longer focal length results in a finer GSD (higher image resolution), whereas increasing altitude enlarges the GSD, reducing spatial resolution.

The Field of View (FOV) of the satellite is the angle between one image boundary and the opposite image boundary. The FOV can be defined in terms of the focal length f and the sensor width s as:

$$FOV = 2 \tan^{-1} \left(\frac{s}{2f} \right) \quad (4.2)$$

The FOV can then be used to calculate the swath of the satellite at a specific orbital altitude.

GSD is the physical distance between the centers of two adjacent pixels measured on the ground in an image captured by the satellite camera.

As the GSD depends on the FOV, it should be noted that both the S_w and the GSD are affected by the angle at which the satellite camera captures the image. This means that the image becomes distorted when taken off-nadir. As shown in Figure 4.2 in an image by James et al. [13]

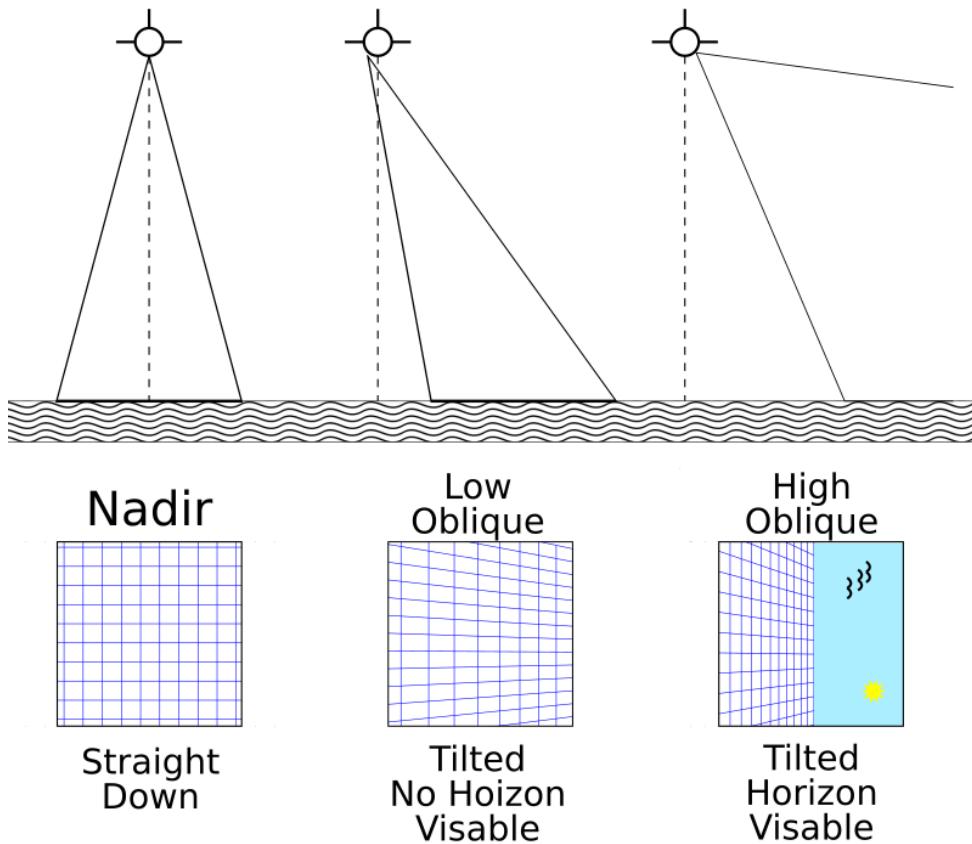
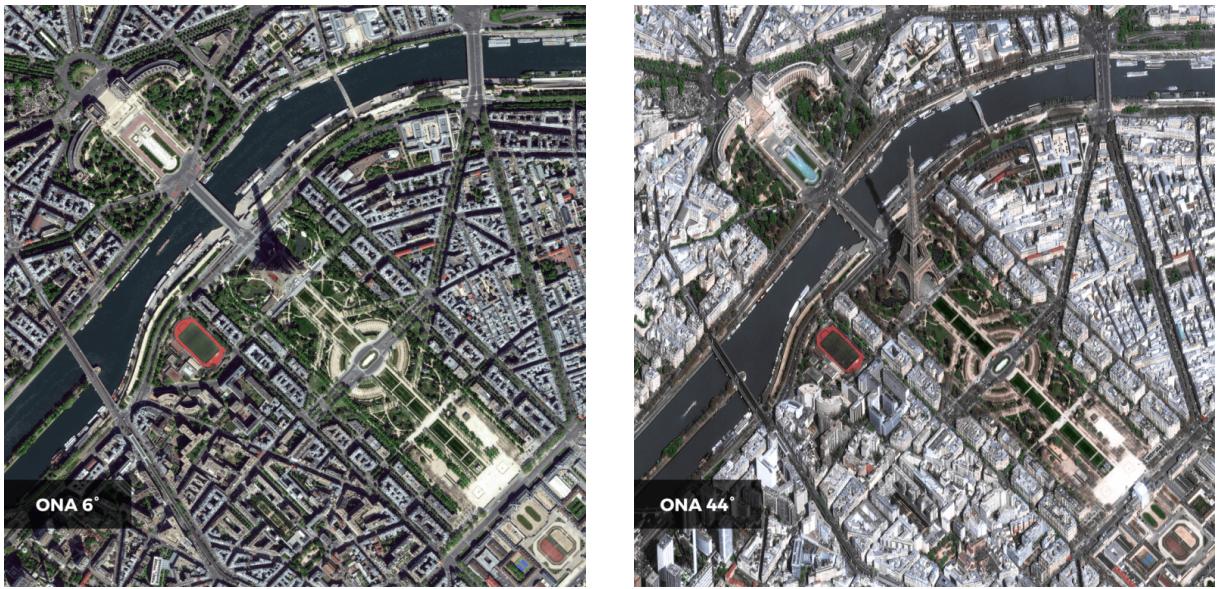


Figure 4.2: An illustration of the distortion of the *GSD* and *FOV* of a satellite image when taken Off-Nadir (at an oblique angle) [13]

Figure 4.3 [55] demonstrates the significant impact an off-nadir angle (ONA) can have on satellite imagery. While ONA is often discussed in relation to its effect on ground sample distance (GSD), the implications extend well beyond image resolution. Viewing the Earth from an angle alters the scene's geometry, introducing distortions that can shift feature positions and complicate image interpretation. These effects become particularly challenging when integrating imagery with Digital Elevation Maps (DEMs), which are typically referenced to a nadir viewing geometry. Applying such DEMs to high-ONA imagery can result in inaccurate terrain corrections, feature misalignment, and increased geolocation errors. In regions with substantial elevation variation, these issues are amplified, reducing the accuracy of orthorectification and making reliable feature extraction more difficult.



(a) The Eiffel tower taken with a 6 Degree off-nadir angle.

(b) The Eiffel tower taken with a 44 degree off-nadir angle.

Figure 4.3: The effect off-nadir angle on satellite imagery.

4.2.2 IMAGE RESOLUTION

Satellite imagery is a fundamental tool for observing and monitoring the Earth's surface. The quality and usefulness of these images are determined by several key characteristics, including spatial, temporal, radiometric, and spectral resolution. Each of these parameters influences the level of detail, frequency of observation, and type of information that can be extracted from the imagery. In this section, we provide an overview of these properties and illustrate their effects with representative examples.

4.2.2.1 SPATIAL RESOLUTION

As defined in 4.2 the GSD defines the spatial resolution of a satellite image, indicating the level of detail that can be resolved on the ground. Smaller GSD values correspond to higher resolution, enabling finer features to be distinguished. This is illustrated in Figure 4.4, where a GSD of 40 m results in lower resolution compared to a GSD of 10 m.



Figure 4.4: Illustration of the relationship between spatial resolution and Ground Sampling Distance (GSD). As GSD increases, each pixel represents a larger area on the ground, resulting in lower image resolution and less detail. Conversely, smaller GSD values correspond to higher resolution, capturing finer features in the scene.

4.2.2.2 TEMPORAL RESOLUTION

Just as satellites have spatial resolution, they also have temporal resolution. Temporal resolution is defined by the frequency with which a satellite revisits a site or target area. For example, Sentinel-2A and Sentinel-2B have a revisit rate of 10 days individually [56], resulting in a temporal resolution of 5 days. This means that by increasing the number of satellites in a constellation, a higher temporal resolution can be achieved. An example of a tree growing is illustrated in Figure 4.5 [14], showing that the higher temporal resolution implies that an area could be observed to be growing steadily and conversely the lower the temporal resolution the more rapid the growth of an area might be seen, indicating less temporal detail. Many satellites with high temporal resolution are placed in sun-synchronous orbits, which are near-polar and allow the satellite to pass over the same location at approximately the same local solar time each day.

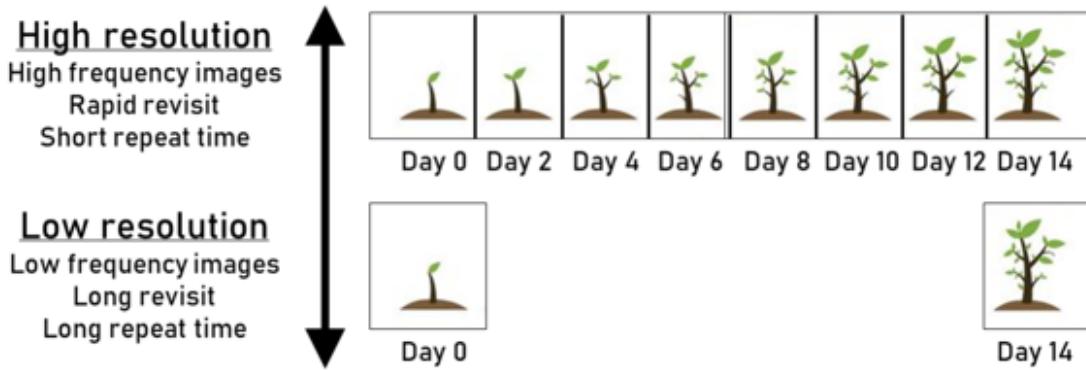


Figure 4.5: Illustration of temporal resolution in satellite imagery represented by a tree growing [14]. Temporal resolution refers to how frequently a satellite revisits the same location. Higher temporal resolution, achieved by shorter revisit intervals or larger satellite constellations, allows changes on the ground, such as vegetation growth or urban development, to be monitored more closely over time.

4.2.2.3 RADIOMETRIC RESOLUTION

A satellite imager also has two other important characteristics: radiometric resolution and spectral resolution. Radiometric resolution describes the sensor's ability to detect small differences in energy and is usually expressed in bits. For instance, an 8-bit sensor can record 256 levels of intensity, while a 12-bit sensor can capture 4096 levels. Spectral resolution, by contrast, refers to the sensor's ability to distinguish between different wavelengths of light. Since sunlight contains a broad range of wavelengths, multispectral imagers (MSI) are designed to respond to specific bands or groups of wavelengths. These may include visible bands such as red, green, and blue, as well as near-infrared (NIR) and shortwave infrared (SWIR). Some sensors, such as panchromatic imagers, cover a wide wavelength range and produce high-resolution black-and-white images. Multispectral and hyperspectral systems are widely used for vegetation monitoring (such as the use of normalized difference vegetation index), water quality studies, mineral exploration, and land-use mapping. Because each material reflects and absorbs light differently across the electromagnetic spectrum, these sensors make it possible to identify and track environmental changes over time. An example is shown in Figure 4.6, which illustrates the spectral bands typically used by a multispectral imager.

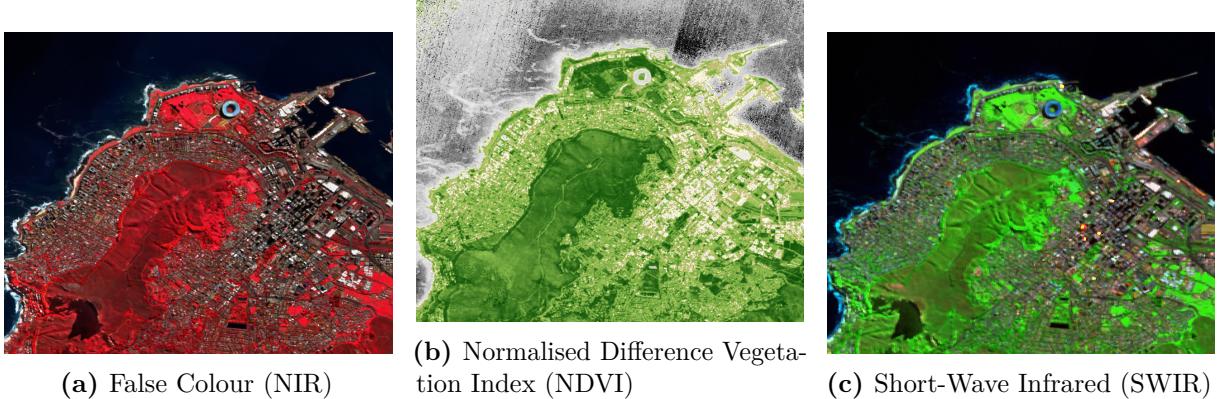


Figure 4.6: Images are captured using different wave lengths and reconstructed to convey different information.

4.2.3 IMAGE DETECTORS

Most Earth observation satellites can be categorized into three types of sensors: pushbroom, whiskbroom, and snapshot [57, 58]. Without going into too much detail, the basic concepts of each sensor mechanism are illustrated in Figure 4.7. A background image is used to convey the concept, although in reality, each sensor element captures just one pixel at a time. The pushbroom sensor, shown in Figure 4.7a, is a type of imager that captures one row of pixels at a time. It relies on the satellite's forward motion to build up the second dimension of the image over time. The whiskbroom sensor, illustrated in Figure 4.7b, captures one pixel at a time by scanning from one side of the swath to the other using a rotating mirror. Like the pushbroom sensor, it also depends on the satellite's movement to form a complete image. Finally, the snapshot sensor, shown in Figure 4.7c, captures a 2D array of the scene in a single time step. Snapshot sensors are considered to have a global shutter, as the entire image is acquired simultaneously.

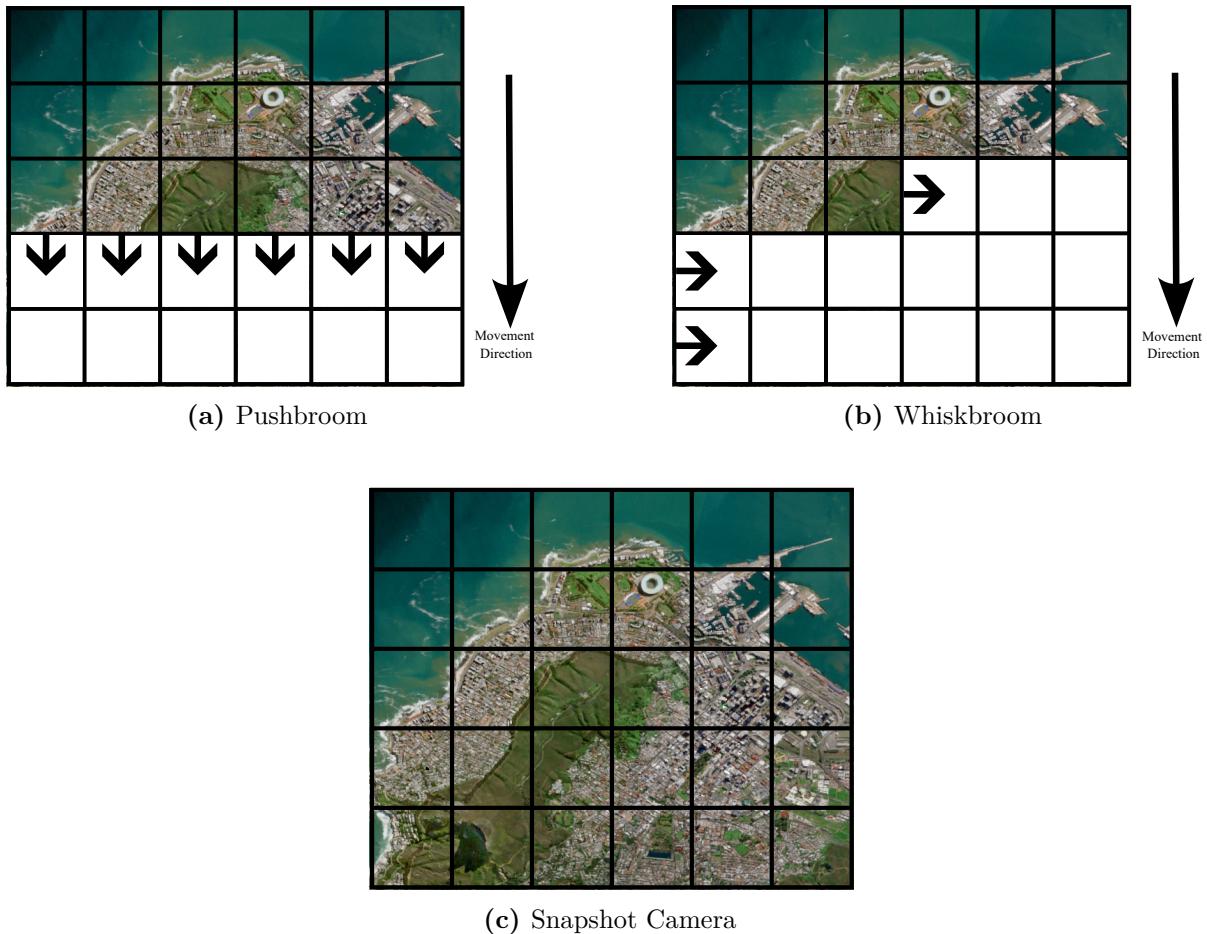


Figure 4.7: Images are captured using different imager detectors.

4.2.4 ENVIRONMENTAL NOISE

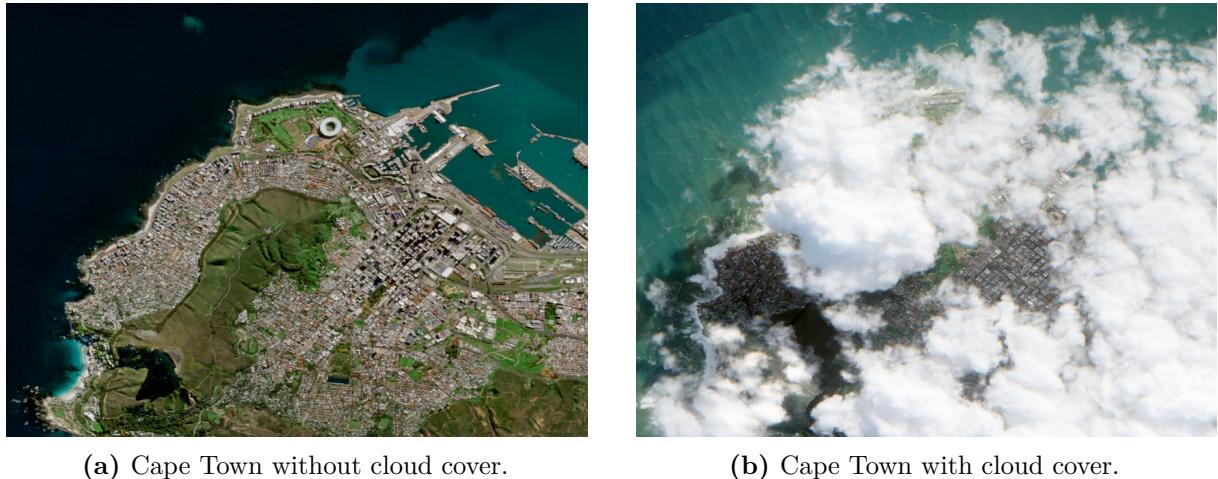
Several types of noise and environmental factors can impact the quality and usability of satellite imagery. Among the most significant are atmospheric interference, cloud cover, and variations due to the time of day or season.

Atmospheric effects play a major role in distorting satellite images. As light from the Earth's surface travels through the atmosphere to reach the satellite sensor, it is subject to scattering and absorption. Particles such as dust, water vapor, and aerosols can scatter incoming light, particularly in shorter wavelengths (e.g., blue), which leads to hazing and reduced image contrast. Atmospheric correction algorithms are often applied during post-processing to compensate for these effects, as we can see in Figure 4.8 [15, 16]



Figure 4.8: Difference in satellite imagery due to atmospheric absorption and scattering [15, 16]

Cloud cover is another major challenge for optical imaging systems. Thick clouds can block the land surface entirely, making it impossible to capture useful data in those areas. Even thin clouds or their shadows can change the spectral signature of a scene, which makes interpretation and classification less reliable. To deal with this, cloud detection and masking methods are often applied, and satellites with higher temporal resolution are used so that cloud-free images can be captured at a later time.



(a) Cape Town without cloud cover.

(b) Cape Town with cloud cover.

Figure 4.9: The effect of cloud cover changing the spectral signature of and image.

Time of day and time of year also have a substantial influence on image appearance and quality. Images captured at different times of the day may exhibit differences in shadow length and direction, which can affect the visual interpretation or performance of automated analysis algorithms. Seasonal changes can also drastically alter the landscape, with vegetation, snow cover, and water bodies all exhibiting different spectral signatures depending on the time of year. For example, a forest may appear lush and green in summer but sparse and brown in winter, even though the physical structure of the landscape remains unchanged.



(a) Cape Town in the wet season.

(b) Cape Town in the dry season.

Figure 4.10: The effect of seasonal and time differences.

Therefore, when analyzing or comparing satellite images, it is essential to consider these factors to ensure accurate interpretation and to avoid drawing incorrect conclusions from noisy or inconsistent data.

4.3 CAMERA MODEL

Accurate modeling of a satellite camera is essential for understanding how three-dimensional scenes are projected onto a two-dimensional image plane. The camera model defines the geometric and photometric relationships between points in the observed scene and their corresponding image measurements. This section introduces the fundamental concepts of camera modeling, beginning with the idealized pinhole model, followed by the definition of intrinsic parameters that map the camera's projection onto a pixel-based image, and finally the extrinsic parameters that relate the camera frame to the satellite body frame. Together, these models form the basis for projecting and interpreting satellite imagery in subsequent image processing and state estimation tasks.

4.3.1 PINHOLE MODEL

The ideal pinhole camera model can be represented as a geometric configuration consisting of an image plane and an optical center, also referred to as the pinhole. Light rays originating from a point in the observed scene pass through the optical center and intersect the image plane, forming an inverted image of the object. The distance between the optical center and the image plane is defined as the focal length, denoted by f , as illustrated in Figure 4.11.

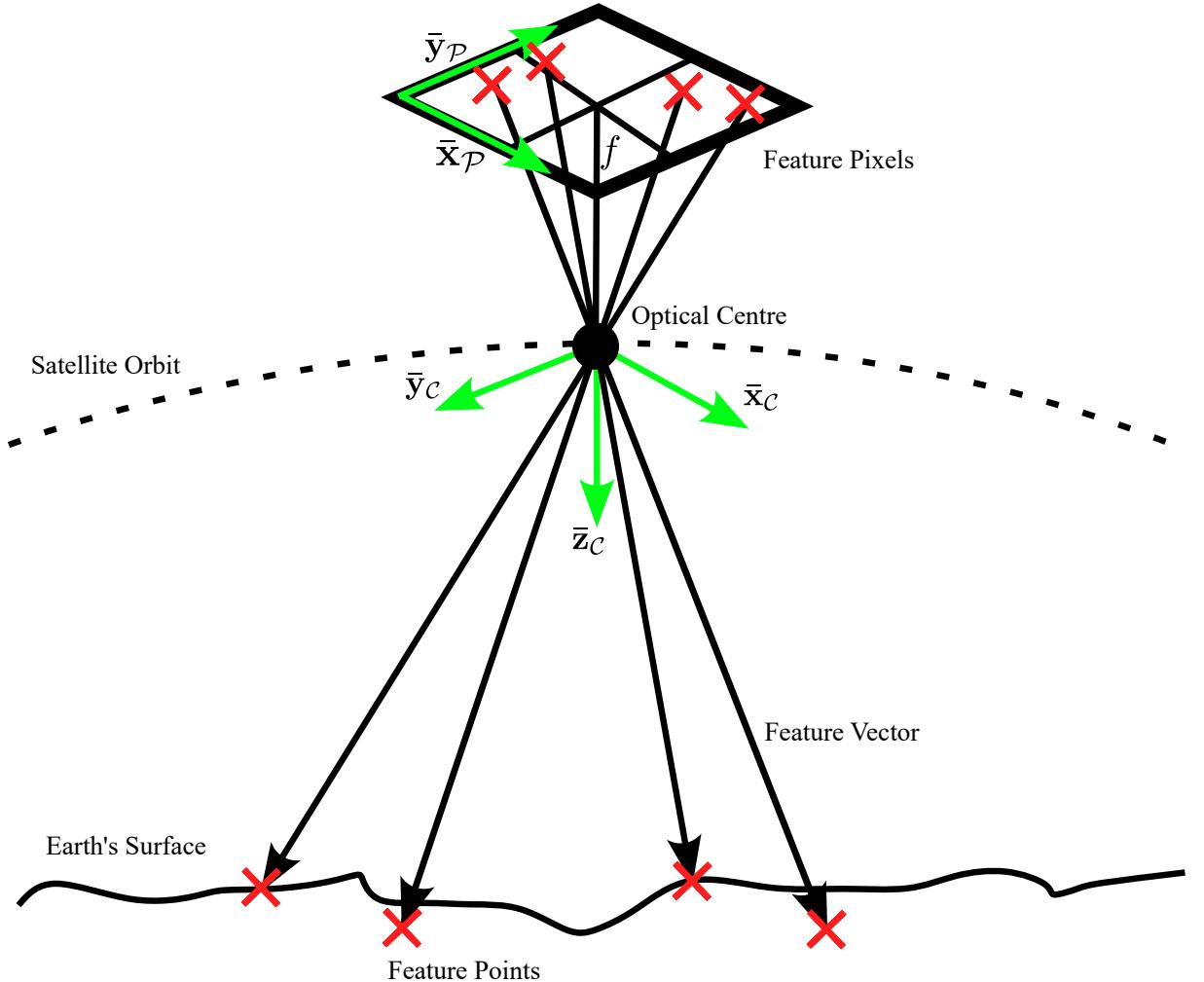


Figure 4.11: A satellite imager illustrated as an ideal pinhole model camera.

The projection from the three-dimensional camera coordinate system \mathcal{C} to the two-dimensional image plane \mathcal{M} is governed by the pinhole camera equation:

$$\begin{bmatrix} x_{\mathcal{M}} \\ y_{\mathcal{M}} \\ -f \end{bmatrix} = \frac{-f}{z_c} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (4.3)$$

As illustrated in Figure 4.11, this projection inherently produces an inverted image, where the x and y axes are flipped with respect to the original scene.

To fit into our transformation scheme this work has established in Chapter 3. This equation can be rewritten as a transformation matrix

$$\mathbf{T}_{\mathcal{C}}^{\mathcal{M}} = \begin{bmatrix} \frac{-f}{z_c} & 0 & 0 & 0 \\ 0 & \frac{-f}{z_c} & 0 & 0 \\ 0 & 0 & \frac{-f}{z_c} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

4.3.2 INTRINSIC CAMERA MATRIX

In digital image representation, the $\bar{x}_{\mathcal{P}}$ -axis typically increases from left to right, and the $\bar{y}_{\mathcal{P}}$ -axis increases from top to bottom, which differs from the conventional mathematical representation of the image plane as can be seen in Figure 4.12. Consequently, an additional transformation is required to map the projected coordinates to the pixel coordinate system. Thus the projection plane coordinates of the projected feature point $\mathbf{f}_{\mathcal{M}}$ can be converted into pixel based measurements by creating a projection matrix also known as the intrinsic camera matrix.

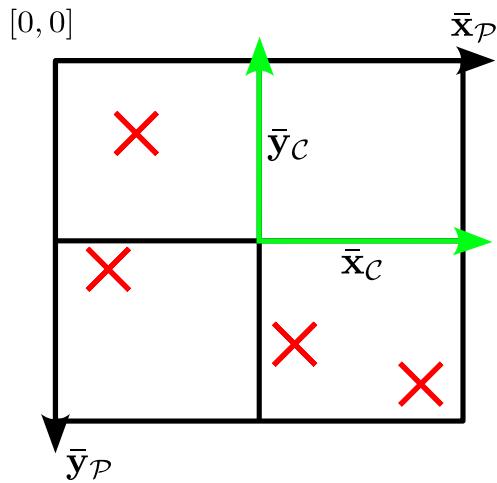


Figure 4.12: The realtionship between the camera refernce frame and the pixel plane reference frame.

Firstly, a horizontal and verticle scale factor s_x and s_y need to be defined as

$$s_x = \frac{I_x}{SensorWidth}, \text{ and} \quad (4.5)$$

$$s_x = \frac{I_y}{SensorHeight}. \quad (4.6)$$

In the above equations, image resolution I_x and I_y refers to the size, in pixels, of the resulting image captures by the modelled camera. *SensorWidth* and *SensorHeight* refers to the physical size of the sensor array, this means the scaling factors s_x and s_y can be seen as having a unit of pixels per distance. This can also be interpreted as the reciprocal of pitch p which is defined as distance per pixel.

Pixel based coordinates of the projected point in this case a feature point $[x_{\mathcal{M}}, y_{\mathcal{M}}, 1]$ can be calculated with,

$$\begin{bmatrix} x_{\mathcal{P}} \\ y_{\mathcal{P}} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & -s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{M}} \\ y_{\mathcal{M}} \\ 1 \end{bmatrix} \quad (4.7)$$

Note the negative sign in front of s_y as the y-axis are flipped as seen in Figure 4.12. The intrinsic matrix in Equation 4.7 can lastly be expanded with the offsets c_x and c_y that ensure that the pixel-based projection plane coordinates are in the lower-right quadrant, as in the convention with digital image. These offsets are defined as:

$$c_x = \frac{I_x}{2}, \text{ and} \quad (4.8)$$

$$c_y = \frac{I_y}{2}. \quad (4.9)$$

Pixel-based coordinates off the projected point $[x_{\mathcal{M}}, y_{\mathcal{M}}, 1]^T$ can be calculated in the typical convention with

$$\begin{aligned} \begin{bmatrix} x_{\mathcal{P}} \\ y_{\mathcal{P}} \\ 1 \end{bmatrix} &= \begin{bmatrix} s_x & \alpha & c_x \\ 0 & -s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{M}} \\ y_{\mathcal{M}} \\ 1 \end{bmatrix} \\ &= \mathbf{K} \begin{bmatrix} x_{\mathcal{M}} \\ y_{\mathcal{M}} \\ 1 \end{bmatrix} \end{aligned} \quad (4.10)$$

where \mathbf{K} is known as the intrinsic matrix. A skewing factor of α is added to adjust for skewing effects of the camera.

The traditional \mathbf{K} matrix can be manipulated to look like our previously defined transformation matrixes, where the z-axis is added to keep the vector format.

$$\mathbf{T}_{\mathcal{C}}^{\mathcal{P}} = \begin{bmatrix} s_x & \alpha & 0 & c_x \\ 0 & -s_y & 0 & c_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

4.3.3 EXTRINSIC CAMERA MATRIX

In the projection equations, it is typically assumed that the point being projected onto the image plane is already expressed in the camera reference frame. However, this is not always the case. In certain situations, the points to be projected must first be transformed

into the camera reference frame.

We need to relate the extrinsic camera frame to the body reference frame and the corresponding DCM is defined as:

$$\mathbf{A}_{\mathcal{B}}^{\mathcal{C}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

It is assumed that the origins of the body frame \mathcal{B} and the camera reference frame \mathcal{C} coincide, so no translation is required. Thus the transformation matrix can be expressed as:

$$\mathbf{T}_{\mathcal{B}}^{\mathcal{C}} = \begin{bmatrix} \mathbf{A}_{\mathcal{B}}^{\mathcal{C}} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (4.13)$$

Thus, the complete transformation chain for converting a 3D feature vector in the body frame to a feature pixel in the pixel plane is:

$$\mathbf{f}_{\mathcal{P}}^+ = \mathbf{T}_{\mathcal{M}}^{\mathcal{P}} \times \mathbf{T}_{\mathcal{C}}^{\mathcal{M}} \times \mathbf{T}_{\mathcal{B}}^{\mathcal{C}} \times \mathbf{f}_{\mathcal{B}}^+ \quad (4.14)$$

4.4 LENS DISTORTION

There are many types of lens distortions that can affect satellite imagery. In this work, we focus on the two primary geometric distortions: radial distortion and tangential distortion [56]. Additionally, chromatic aberration will be addressed, although other distortions exist beyond the scope of this study. As background for modeling these distortions, the following equation defines the squared distance of a pixel from the lens center:

$$r^2 = \mathbf{x}_{\mathcal{P},n}^2 + \mathbf{y}_{\mathcal{P},n}^2 \quad (4.15)$$

Here, $\mathbf{x}_{\mathcal{P},n}$ and $\mathbf{y}_{\mathcal{P},n}$ represent the undistorted pixel coordinates, and r is the radial distance from the optical center of the lens.

4.4.1 RADIAL DISTORTION

Radial distortion, also referred to as barrel or pincushion distortion, is a type of lens aberration in which straight lines appear increasingly curved as their distance from the optical centre grows as seen in Figure 4.13. This effect is caused by imperfections in the lens geometry, leading to magnification that varies with radial distance from the image centre. Barrel distortion causes lines to bow outward, while pincushion distortion causes

them to bend inward, each altering the perceived shape of objects in the image. This distortion is governed by

$$\begin{aligned}\mathbf{x}_{\mathcal{P},r} &= \mathbf{x}_{\mathcal{P},n}(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ \mathbf{y}_{\mathcal{P},r} &= \mathbf{y}_{\mathcal{P},n}(k_1 r^2 + k_2 r^4 + k_3 r^6)\end{aligned}\quad (4.16)$$

Here, $\mathbf{x}_{\mathcal{P},r}$ and $\mathbf{y}_{\mathcal{P},r}$ are the radially distorted pixel coordinates, while $\mathbf{x}_{\mathcal{P},n}$ and $\mathbf{y}_{\mathcal{P},n}$ are the original, undistorted pixels. The variable r is the radial distance from the optical center, defined in Equation 4.15. The coefficients k_1 , k_2 , and k_3 control the magnitude and shape of the distortion, with higher-order terms affecting distortion further from the center.



Figure 4.13: Radial lens distortion applied to a satellite image of Cape Town stadium.

4.4.2 TANGENTIAL DISTORTION

Tangential distortion is a type of lens distortion in which the image appears as if it has been tilted or skewed, making it look misaligned with the sensor array as seen in Figure 4.14. This effect occurs when the lens and image sensor are not perfectly parallel, causing light rays to intersect the sensor at an angle. As a result, straight lines may appear slanted, and the overall image geometry can be subtly warped.

$$\begin{aligned}\mathbf{x}_{\mathcal{P},t} &= p_1(r^2 + 2\mathbf{x}_{\mathcal{P},n}^2) + 2p_2\mathbf{x}_{\mathcal{P},n}\mathbf{y}_{\mathcal{P},n} \\ \mathbf{y}_{\mathcal{P},t} &= 2p_1\mathbf{x}_{\mathcal{P},n}\mathbf{y}_{\mathcal{P},n} + p_2(r^2 + 2\mathbf{y}_{\mathcal{P},n}^2)\end{aligned}\quad (4.17)$$

Here, $\mathbf{x}_{\mathcal{P},t}$ and $\mathbf{y}_{\mathcal{P},t}$ are the tangentially distorted pixel coordinates, while $\mathbf{x}_{\mathcal{P},n}$ and $\mathbf{y}_{\mathcal{P},n}$ are the original, undistorted pixels. The variable r is the radial distance from the optical center,

as defined in Equation 4.15. The coefficients p_1 and p_2 represent the tangential distortion parameters, controlling the degree of image skew caused by misalignment between the lens and sensor. Higher values of these coefficients result in greater apparent tilting or warping of the image.

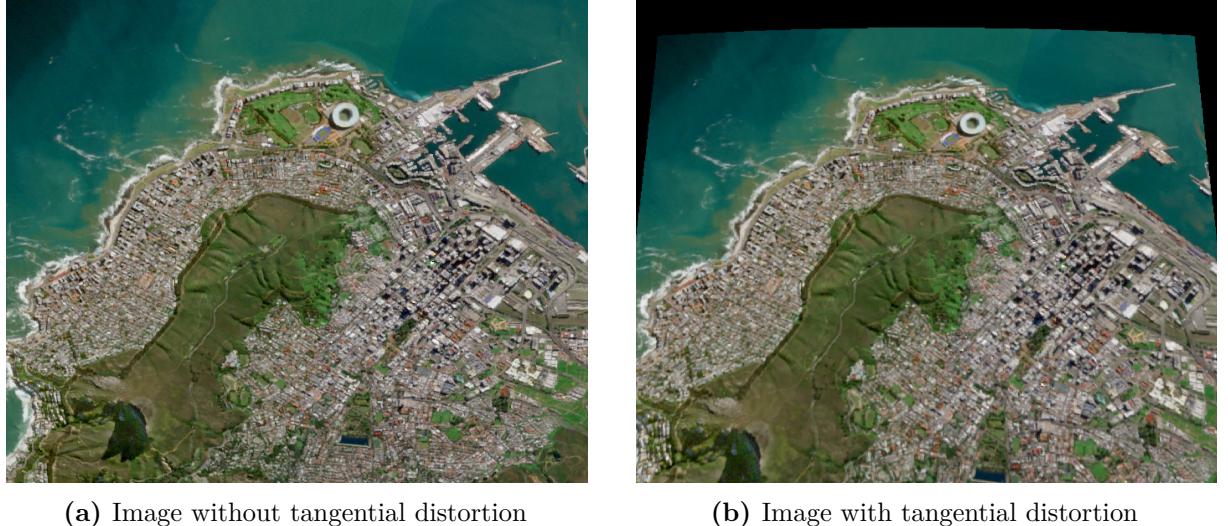


Figure 4.14: Tangential lens distortion applied to a satellite image of Cape Town stadium.

The total geometric distortion can be expressed as the sum of the radial distortion and the tangential distortion components as shown in Equation 4.18. Together, these distortions describe how the lens alters the true geometry of the scene, affecting both the curvature and alignment of image features. Accurate modelling and correction of both components are essential in satellite camera calibration, where precise spatial measurements are required.

$$\begin{aligned}\mathbf{x}_{\mathcal{P},d} &= \mathbf{x}_{\mathcal{P},r} + \mathbf{x}_{\mathcal{P},t} \\ \mathbf{y}_{\mathcal{P},d} &= \mathbf{y}_{\mathcal{P},r} + \mathbf{y}_{\mathcal{P},t}\end{aligned}\tag{4.18}$$

4.4.3 CHROMATIC ABBERATION

Chromatic aberration arises from the nature of light itself. White light is a combination of many wavelengths. Since lenses use refraction to focus light rays, different wavelengths bend by different amounts. As a result, red light will focus on a different plane than green, which in turn focuses on a different plane than blue. Blue light is refracted the most, causing an enlarging effect, as its focal point is closest to the lens as illustrated in Figure 4.15. The Figure 4.16 shows the effect of chromatic aberration in the lens, with the effect most prominent on the edges of the image.

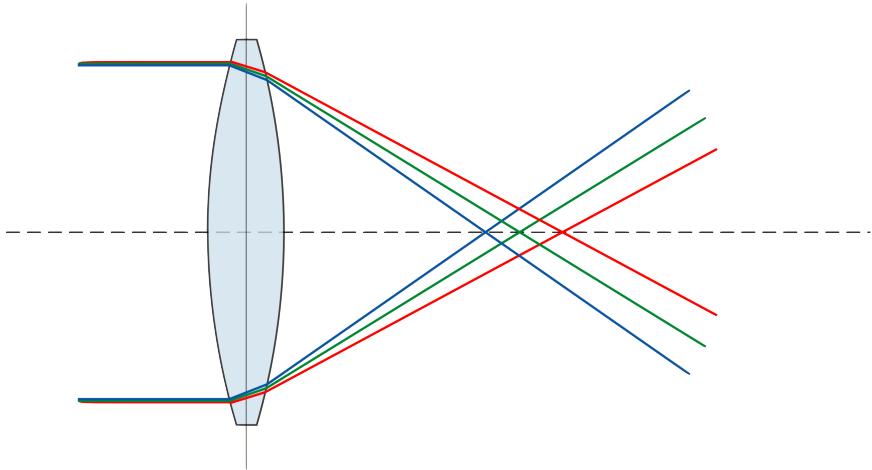


Figure 4.15: The effect of the lens on different wave lengths of light which will cause some wavelengths to be out of focus and enlarged. [17]

This is done by separating the image into its red, green, and blue channels, enlarging each channel by a different scale factor, and then recombining them into a single RGB image. As a result, the blue channel appears the largest, followed by the green channel, with the red channel appearing smallest.



(a) Original Image



(b) Image with chromatic abberation

Figure 4.16: Chromatic Abberation on a satellite image of Cape Town stadium

4.5 EARTH TRACKER

4.5.1 IMAGE GENERATION

To generate the image that the satellite imager will observe, a simulation environment must be established. In this environment, the satellite can be positioned in any orbit around the barycenter of the system, and a model of the Earth can be incorporated into the scene.

To visually represent the Earth, an ellipsoid is placed at the barycenter of the system, with a semi-major axis of 6 378 km and a semi-minor axis of 6 356 km, as shown in Figure 3.7. A low-resolution texture of the Earth is then wrapped around the ellipsoid to serve as a visual guide for aligning the higher-resolution imagery, as illustrated in Figure 4.17.



Figure 4.17: Earth rendered in the simulation environment

Covering the entire Earth in high-resolution imagery would be computationally expensive; therefore, localized high-resolution image patches are used instead. To create a high-resolution image strip with a specified GSD, imagery is downloaded from the Copernicus Browser and stitched together in QGIS, a process known as mosaicing. The Copernicus Browser provides GeoTIFF files, each covering approximately 100 km^2 at a GSD of 15 meters, which are already georeferenced. Figure 4.18 shows an example of mosaicing different satellite images.

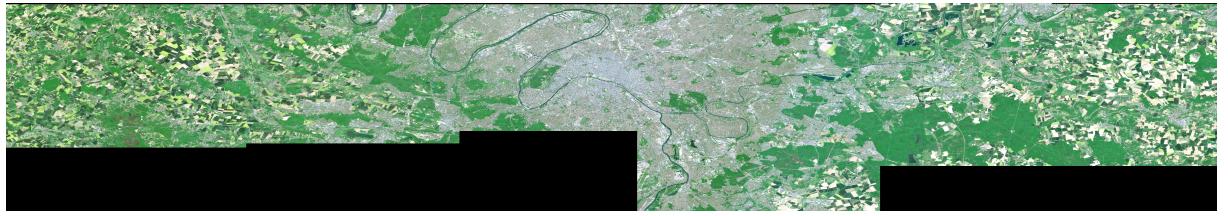


Figure 4.18: High resolution mosaic strip of Paris, France

After mosaicing, the latitude, longitude, and altitude coordinates of each pixel are converted to ECEF coordinates and referenced to the WGS84 ellipsoid. Afterwards, the high resolution strip is placed on top of the simulated Earth as seen in Figure 4.19.



Figure 4.19: This is the high resolution strip placed on the surface of the simulated Earth.

To simulate the Earth's rotation, both the Earth and the high-resolution imagery patch are re-rendered at the rotational speed of the Earth, ω_e

To capture an image, the virtual camera is positioned at the satellite's location within the simulator and oriented in a pure nadir direction. The resulting image is shown in Figure 4.20.



Figure 4.20: An generated satellite imager as seen from the satellite imager

4.5.2 FEATURE DETECTION

The images are first converted to grayscale and then processed using feature detection algorithms. MATLAB's feature detection functions allow the user to specify the number of feature points to detect. In Figure 4.21, 20 of the strongest feature points were selected.

SIFT is slower and more computationally intensive than SURF and ORB, but offers high accuracy, making it suitable for target recognition. SURF provides a faster alternative to SIFT, though with slightly reduced precision. ORB, using binary descriptors and Hamming distance for matching, is extremely fast and ideal for real-time applications, but is more sensitive to heavy noise and large scale variations. While ORB is widely used in real-time applications such as SLAM, it may struggle with rapidly changing satellite imagery, where a more robust approach such as SIFT or SURF would be more ideal.

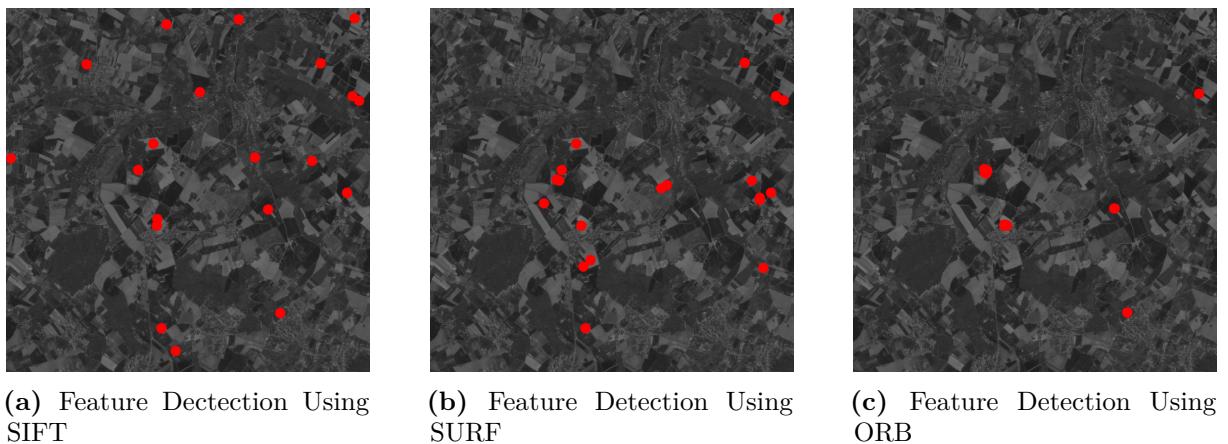


Figure 4.21: The french country side is used to compare the different feature points detected by SIFT, SURF and ORB

4.5.3 EARTH TRACKER ALGORITHM

When a feature point is extracted from an image, it must be converted back into a feature vector, a process known as back-projection [33]. This algorithm essentially performs the inverse of the camera model equation described in Section 4.3,

$$\mathbf{f}_B^+ = \mathbf{T}_C^B \times \mathbf{T}_M^C \times \mathbf{T}_P^M \times \mathbf{f}_P^+. \quad (4.19)$$

Where the transformation matrixes are defined as:

$$\mathbf{T}_P^M = \begin{bmatrix} \frac{1}{s_x} & \alpha & 0 & \frac{-c_x}{s_x} \\ 0 & \frac{1}{-s_y} & 0 & \frac{-c_y}{-s_y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.20)$$

$$\mathbf{T}_M^C = \begin{bmatrix} \frac{-s}{f} I_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}, \text{ and} \quad (4.21)$$

$$\mathbf{T}_C^B = \begin{bmatrix} (\mathbf{A}_B^C)^T & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (4.22)$$

All of these matrices are derived from the camera matrices defined in the camera model. This algorithm performs the 2D-to-3D projection, effectively reversing the 3D-to-2D mapping of the camera. First, it converts a pixel point into coordinates on the camera's image plane in meters by translating the origin and applying the appropriate scaling factors, s_x and s_y . The resulting values are then scaled to represent the feature as projected outside the satellite body frame.

The scaling factor s is particularly important, as information is inevitably lost during the 3D-to-2D projection. To resolve this ambiguity, an assumption about the satellite's operating altitude is required. Here, we assume an altitude of 500 km, giving a scaling factor of $s = 500$ km.

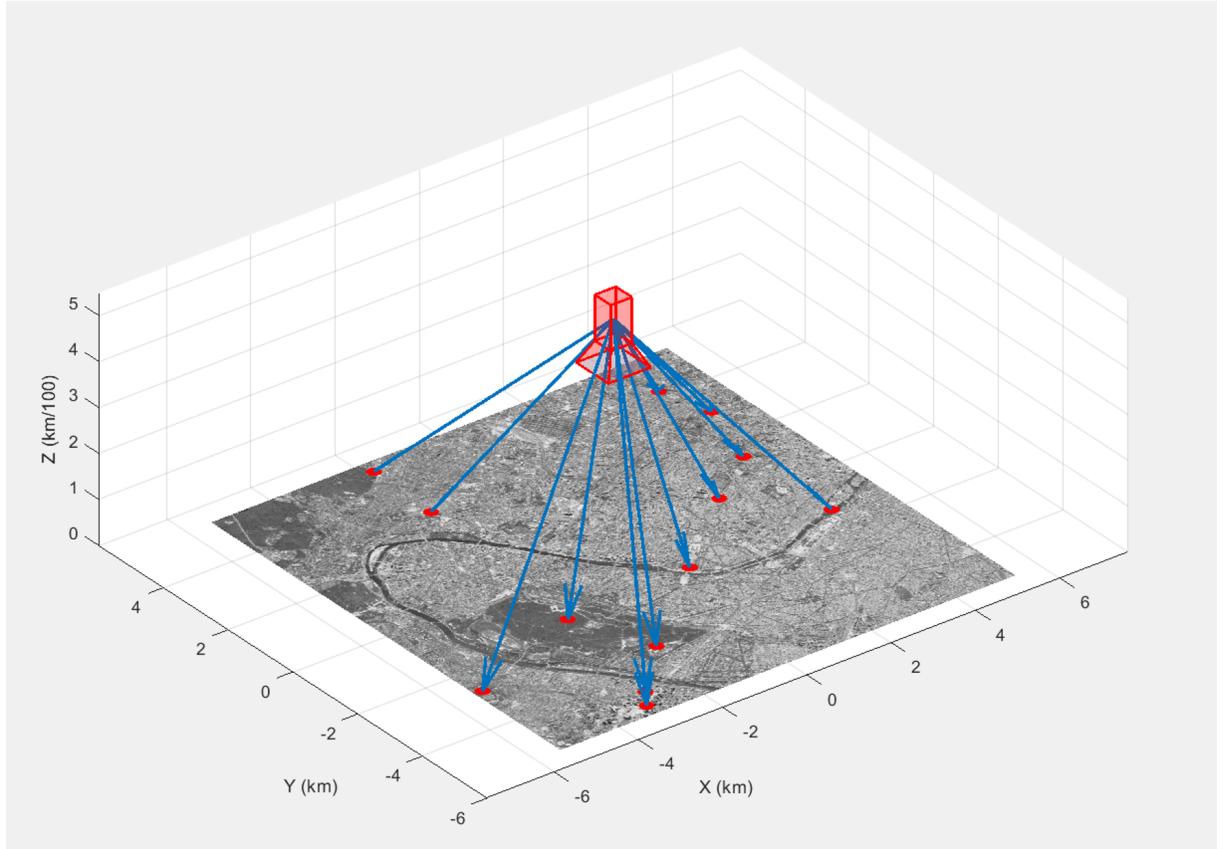


Figure 4.22: An illustration of feature points being projected back to feature vectors.

In Figure 4.22 the z-axis is scaled to show the feature vectors created by the Earth Tracker algorithm.

4.5.4 CATALOGUE GENERATION

In this work, creating a catalogue requires proper geolocation. Although feature matching is beyond the scope of this work and the features are randomly generated, the true position of the satellite is assumed, and raycasting is implemented to bypass feature matching.

First the feature pixels are converted back to rays, exactly the same as the Earth Tracker, but is converted all the way to the ECI reference frame.

$$\mathbf{f}_{\mathcal{I}}^+ = \mathbf{T}_{\mathcal{O}}^{\mathcal{I}} \times \mathbf{T}_{\mathcal{B}}^{\mathcal{O}} \times \mathbf{T}_{\mathcal{C}}^{\mathcal{B}} \times \mathbf{T}_{\mathcal{M}}^{\mathcal{C}} \times \mathbf{T}_{\mathcal{P}}^{\mathcal{M}} \times \mathbf{f}_{\mathcal{P}}^+. \quad (4.23)$$

Then the vector is normalized

$$\bar{\mathbf{f}}_{\mathcal{I}} = \frac{\mathbf{f}_{\mathcal{I}}}{\|\mathbf{f}_{\mathcal{I}}\|}. \quad (4.24)$$

The intersection of a ray with the Earth ellipsoid is computed as follows. Let the ray in ECI coordinates be $\mathbf{p}(t) = \mathbf{r}_{\mathcal{I}} + t \bar{\mathbf{f}}_{\mathcal{I}}$, where $\mathbf{r}_{\mathcal{I}} = [x_0, y_0, z_0]^T$ is the satellite position and

$\bar{\mathbf{f}}_{\mathcal{I}} = [dx, dy, dz]^T$ is the ray direction. Assuming the Earth is an ellipsoid with semi-major axis a and semi-minor axis b , the quadratic equation for the intersection is

$$At^2 + Bt + C = 0, \quad (4.25)$$

with coefficients

$$\begin{aligned} A &= \frac{dx^2 + dy^2}{a^2} + \frac{dz^2}{b^2}, \\ B &= 2 \left(\frac{x_0 dx + y_0 dy}{a^2} + \frac{z_0 dz}{b^2} \right), \text{ and} \\ C &= \frac{x_0^2 + y_0^2}{a^2} + \frac{z_0^2}{b^2} - 1. \end{aligned}$$

The discriminant $\Delta = B^2 - 4AC$ determines whether the ray intersects the ellipsoid. If $\Delta < 0$, no intersection occurs. Otherwise, the solutions are

$$t_1 = \frac{-B - \sqrt{\Delta}}{2A}, \quad t_2 = \frac{-B + \sqrt{\Delta}}{2A}. \quad (4.26)$$

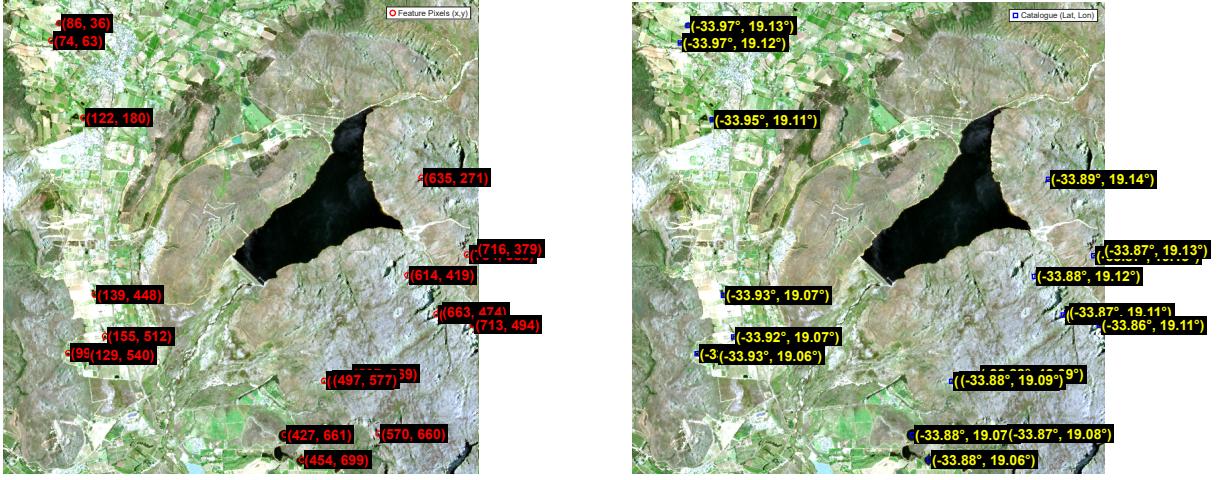
The smallest positive root is chosen as $t_{\text{intersect}}$, corresponding to the closest intersection point in the direction of the ray. The intersection point in ECI coordinates is then

$$\mathbf{p}_{\mathcal{I}} = \mathbf{r}_{\mathcal{I}} + t_{\text{intersect}} \bar{\mathbf{f}}_{\mathcal{I}}. \quad (4.27)$$

After that the feature ray is converted to the ECEF reference frame and then to the LLA point \mathbf{p}_{LLA} .

$$\mathbf{p}_{LLA} = f(\mathbf{T}_{\mathcal{I}}^{\mathcal{R}} \times \mathbf{p}_{\mathcal{I}}, \text{WGS84}) \quad (4.28)$$

The Figure 4.23 illustrates the end results of this algorithm, which used to calculate the geolocations of the feature pixels to create the catalogue.



(a) Satellite image with pixel locations of features detected.

(b) Satellite image of feature pixel locations geolocated using raycasting.

Figure 4.23: Catalogue generation using raycasting.

4.6 CONCLUSION

In this chapter, the process of converting visual odometry from a satellite image to Earth Tracker measurements is described. Firstly, the satellite image geometry is discussed, including parameters such as the GSD, pitch, swath, and FOV. The different types of resolution obtainable from satellite imagery are then explained, namely spatial, temporal, and radiometric resolution. The various methods of capturing these images are described, ranging from classical techniques such as pushbroom and whiskbroom sensors to more modern snapshot sensors. Each of these imagers must also contend with environmental noise sources such as clouds, atmospheric effects, and variations due to time of day or year.

After the geometry of the image is explained, the camera itself is modeled using the pinhole model. From this, an unconventional intrinsic and extrinsic camera model, \mathbf{T}_C^P and \mathbf{T}_B^C respectively, is derived to mirror the format established in Chapter 3. Following the camera modeling, the lens is also modeled. While many types of lens distortions exist, this work focuses on two geometric distortions (radial and tangential) and one radiometric distortion (chromatic aberration).

Finally, the Earth Tracker itself is addressed. The image generation process is first described by adding a high-resolution patch to the simulator for the Earth Tracker to use. The Earth Tracker algorithm is then derived from the intrinsic and extrinsic camera parameters and uses back projection to convert from a 2D feature point to a 3D feature vector. For proper pose estimation the feature geolocation is also needed, but to bypass the feature matching process raycasting is described and used.

CHAPTER 5

STATE ESTIMATION

5.1 INTRODUCTION

After modeling the satellite and sensors in Chapter 3, and designing the Earth Tracker in Chapter 4, Chapter 5 now focuses on the estimator. The chapter begins with a basic background on recursive estimators, followed by a detailed discussion of the Kalman Filter. Next, the system modeling within the pose estimator is presented, starting with the motion model and then the measurement models. The Earth Tracker measurement model is addressed first, followed by the supporting sensor measurement models. Finally, the chapter concludes with a description of the system initialization and design.

5.2 RECURSIVE ESTIMATION

This section introduces the concept of recursive estimators and highlights several commonly used approaches in localization and tracking. A recursive estimator updates the current state distribution by combining the prior distribution of the system states with the latest sensor measurements. The basic operation of a discrete recursive filter is illustrated in Figure 5.1 [59].

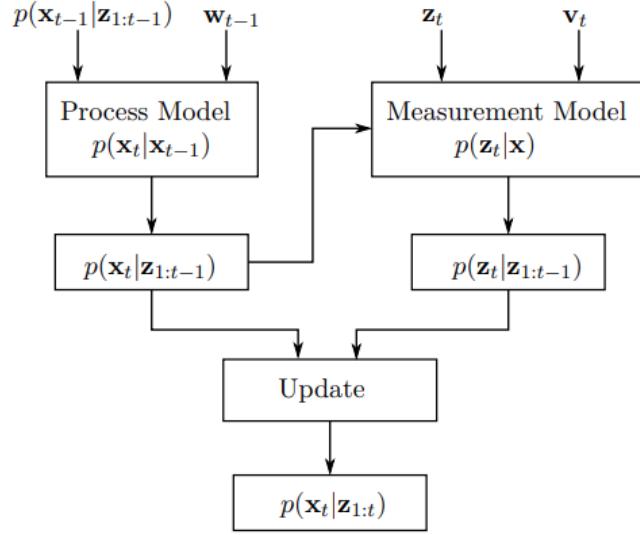


Figure 5.1: Recursive estimator algorithm flowchart [18]

The system state at time t is represented by \mathbf{x}_t in the discrete time domain. The process, or state transition function, is expressed as

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}_{t-1}) \quad (5.1)$$

where \mathbf{f} is either a linear or non-linear transition function, \mathbf{u}_{t-1} represents the control input, and \mathbf{w}_{t-1} represents the process noise. New observations \mathbf{z}_t are available at discrete timesteps and are related to the system state by the measurement function

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t) \quad (5.2)$$

where \mathbf{h} is the observation model and \mathbf{v}_t is the measurement noise. The goal is to obtain the posterior distribution, $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, over the state vector \mathbf{x}_t . This is done by recursively performing process and measurement updates.

At a time t , the posterior distribution over \mathbf{x}_{t-1} is assumed to be known. The prior distribution at t is then calculated as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \cdot d\mathbf{x}_{t-1}. \quad (5.3)$$

The measurement update is used to calculate the new posterior at time t , given the prior state distribution, according to Bayes' rule,

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \quad (5.4)$$

The Kalman Filter is a popular estimator used in pose estimation problems. It is a special case of the Bayes filter, where Gaussian noise distributions are assumed. The initial system

distribution is also assumed Gaussian. A control and measurement update is executed at each sampling instant to update the distribution over the states. If the previous state distribution is Gaussian, then the updated current distribution will also be Gaussian, and therefore the best estimate, $\hat{\mathbf{x}}_t$, is chosen as the mean of the distribution. Different variants of the Kalman Filter exist, of which the Extended and Unscented Kalman Filters are the most popular.

The Extended Kalman Filter (EKF) overcomes the restrictions of the linear filter by approximating non-linear functions to be linear using a first-order Taylor expansion. The mean of the state distribution is used as the linearisation point around which the tangent of the non-linear function is calculated, allowing the use of standard Kalman Filter equations. It is typically more efficient than other non-linear filters, sometimes at the cost of reduced accuracy.

The Unscented Kalman Filter (UKF) uses stochastic linearisation to deal with non-linear systems. Given a distribution with a known mean and covariance, a set of weighted points, known as sigma points, are chosen and transformed using the non-linear function. A new distribution is determined from the transformed sigma points. The process and observation functions do not need to be differentiable and the output is based on values in a larger region, rather than a local approximation.

In this context, the EKF is preferred due to its balance between computational efficiency and estimation accuracy. While the UKF provides improved handling of strong nonlinearities, the EKF remains well suited for systems where the nonlinear behaviour is moderate and the dynamics can be reliably linearised around the current estimate, which typically applies in attitude or pose estimation problems. Furthermore, the EKF's reliance on analytical Jacobians offers greater transparency and control over the estimation process, which is beneficial for interpreting results and diagnosing model behaviour. Its relatively low computational cost makes it attractive for real-time implementations or onboard processing, where resources are limited but estimation accuracy must still be maintained.

5.3 THE EXTENDED KALMAN FILTER

Kalman filters are well suited for localisation problems since the nature of these systems are normally non-linear. Both the linear and non-linear variants of the Kalman Filter are concerned with estimating states using a motion model and is able to perform measurement fusion. The state estimation problem in this work makes use of non-linear system models, and a high-dimensional state space. The EKF is well suited for this problem, since it accommodates non-linear process and observation models and is capable of dealing with

high-dimensional state spaces. The EKF is often used for the Simultaneous Localisation and Mapping (SLAM) problem and is well known in the field of robotics and localisation. The EKF is governed by the following set of equations, as presented in Section 5.3.1 - 5.3.3 and further described in [60].

5.3.1 THE PREDICTION STEP

The prediction step, also known as the time update, projects the current state and error covariance estimates forward to obtain the estimates for the next time step. This is defined as follows:

$$\hat{\mathbf{x}}_{t+1,t} = \mathbf{f}(\hat{\mathbf{x}}_{t,t}) + \mathbf{G}\mathbf{u}_t, \text{ and} \quad (5.5)$$

$$\mathbf{P}_{t+1,t} = \mathbf{F}_t \mathbf{P}_{t,t} \mathbf{F}_t^T + \mathbf{Q} \quad (5.6)$$

where $\hat{\mathbf{x}}$ represents the estimated state, \mathbf{f} is the process model of the system, \mathbf{G} is the control matrix, and \mathbf{u} is the control input. The term \mathbf{P} is the state covariance matrix, which represents the uncertainty of each state. \mathbf{F} is the Jacobian of the process model, and \mathbf{Q} is the process noise covariance matrix. This final matrix, shown in Eqaution 5.6, represents the uncertainty in the process model itself, accounting for discrepancies between the mathematical model and the true physical system.

Since the motion model, \mathbf{f} , is a non-linear vector function, it must be linearized at each time step to apply the standard Kalman Filter equations for covariance propagation. The function is linearized around the most recent state estimate, $\hat{\mathbf{x}}_{t,t}$, using a first-order Taylor series expansion:

$$\mathbf{f}(\hat{\mathbf{x}}) \approx \mathbf{f}(\hat{\mathbf{x}}_{t,t}) + \mathbf{F}_t (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{t,t}) \quad (5.7)$$

where \mathbf{F}_t is the Jacobian matrix of the process model, evaluated at the current state estimate $\hat{\mathbf{x}}_{t,t}$:

$$\begin{aligned} \mathbf{F}_t = \mathbf{F}(\hat{\mathbf{x}}_{t,t}) &= \frac{\partial \mathbf{f}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{t,t}} \\ &= \begin{bmatrix} \frac{\partial f_1}{\partial \hat{x}_1} & \cdots & \frac{\partial f_1}{\partial \hat{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \hat{x}_1} & \cdots & \frac{\partial f_m}{\partial \hat{x}_n} \end{bmatrix} \end{aligned} \quad (5.8)$$

The required partial derivatives for the Jacobian are approximated using midpoint numerical differentiation, expressed as:

$$f'(x) \approx \frac{f(x+e) - f(x-e)}{2e} \quad (5.9)$$

where e denotes a small perturbation applied in both directions to evaluate the gradient. Although a symbolic solution for the Jacobian is theoretically possible, preliminary tests showed that it is computationally expensive and thus impractical for real-time implementation.

5.3.2 THE UPDATE STEP

The update step, also known as the measurement update, corrects the state estimate from the prediction step using a new measurement from a sensor. This process refines the estimate and reduces its uncertainty. The step begins by computing the difference between the actual measurement and the predicted measurement.

First, the innovation or measurement residual, \mathbf{y}_t , is computed. This represents the new information available to the filter, which is the discrepancy between the actual sensor measurement and the predicted measurement:

$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t,t-1}) \quad (5.10)$$

where \mathbf{z}_t is the actual measurement vector and $\mathbf{h}(\cdot)$ is the non-linear measurement model that maps the state space into the measurement space.

Next, the innovation covariance, \mathbf{S}_t , is calculated. It represents the total uncertainty of the innovation, combining the uncertainty of the predicted state and the measurement noise:

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t,t-1} \mathbf{H}_t^T + \mathbf{R}_t \quad (5.11)$$

where \mathbf{H}_t is the Jacobian of the measurement model \mathbf{h} evaluated at the state estimate $\hat{\mathbf{x}}_{t,t-1}$, and \mathbf{R}_t is the measurement noise covariance matrix.

Using the innovation covariance, the optimal Kalman Gain, \mathbf{K}_t , is computed. The gain determines the weight given to the innovation when updating the state estimate:

$$\mathbf{K}_t = \mathbf{P}_{t,t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1} \quad (5.12)$$

Finally, the state estimate and its covariance are corrected using the Kalman Gain and the innovation to produce the final, more accurate estimates for the current time step.

$$\hat{\mathbf{x}}_{t,t} = \hat{\mathbf{x}}_{t,t-1} + \mathbf{K}_t \mathbf{y}_t \quad (5.13)$$

The state covariance is updated to reflect the reduction in uncertainty resulting from the measurement. The Joseph form of the covariance update equation, chosen for its numerical stability, is used:

$$\mathbf{P}_{t,t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t,t-1} (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)^T + \mathbf{K}_t \mathbf{R}_t \mathbf{K}_t^T \quad (5.14)$$

where \mathbf{I} is the identity matrix.

5.3.3 MULTI-SENSOR FUSION TECHNIQUES

EKFs can combine information from multiple sensors in several different ways. Three common fusion strategies are outlined below:

Batch EKF

In a batch EKF, all sensor models are combined into a single filter that directly fuses raw measurements. This “fully integrated” approach can be more optimal because it explicitly accounts for cross-correlations between the different sensor measurements. However, it is computationally more demanding and may become impractical for systems with many heterogeneous sensors.

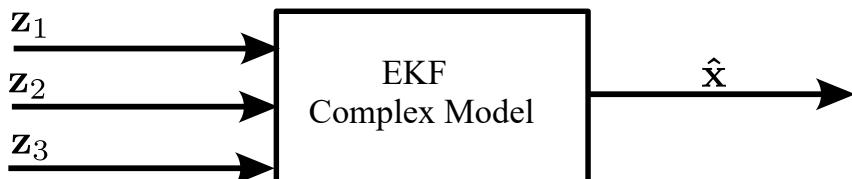


Figure 5.2: The fully integrated Batch EKF architecture, which combines all raw sensor data into a single, complex measurement model $\mathbf{h}(\mathbf{x})$ for concurrent processing.

Multi-Filter EKF

In a multi-filter EKF, each sensor (or group of sensors) maintains its own local EKF. A higher-level master filter then fuses these local estimates into a global state. This approach is modular and makes fault isolation easier, but it requires careful handling of inter-filter correlations and can introduce additional latency.

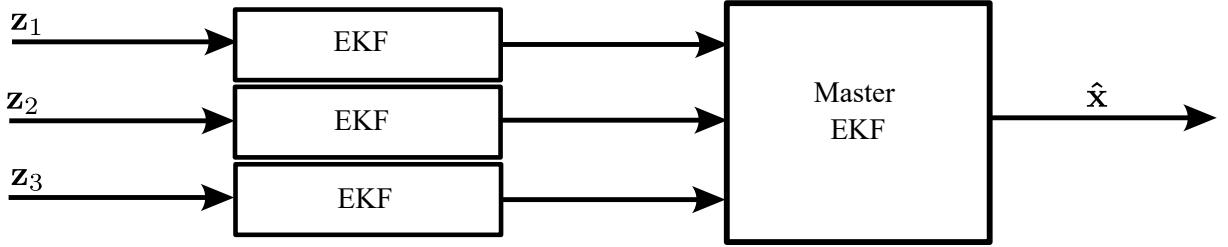


Figure 5.3: The Multi-Filter EKF architecture, featuring independent local filters for each sensor and a master filter to fuse the local estimates into a global state.

Sequential Update EKF

In the sequential update approach, a single EKF processes sensor measurements one at a time rather than combining them into a single batch. This is essentially the standard EKF formulation: a prediction step is followed by multiple updates, each corresponding to one sensor measurement. This method is particularly useful for systems with asynchronous sensors operating at different sampling rates.

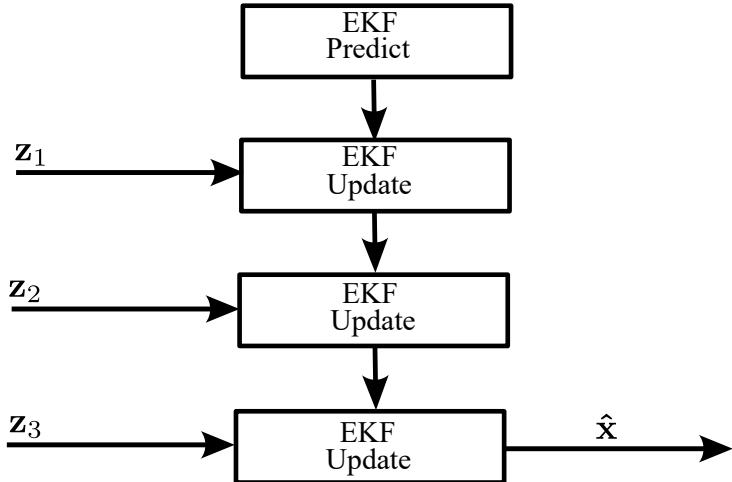


Figure 5.4: The Sequential Update EKF architecture, illustrating a single prediction step followed by multiple, sequential update steps for each individual sensor measurement.

In this work, the sequential update EKF is adopted, as it aligns well with the characteristics of satellite sensor suites. For instance, GPS measurements are relatively slow but absolute, while gyroscopes provide high-frequency angular velocity data that drift over time. Star trackers deliver highly accurate orientation information, though at lower update rates. Coarse sun sensors and magnetometers supply additional measurements that are less accurate but robust and continuously available. Processing these measurements sequentially allows the filter to integrate information at the rate each sensor provides it.

It is important to note that the order of updates in the sequential approach affects the filter's performance. A common strategy is to process measurements from lower-accuracy

or noisier sensors first (e.g., coarse sun sensor, magnetometer), followed by higher-accuracy sensors such as the star tracker. This ensures that the state estimate is progressively refined rather than being skewed or “pulled” away from the true value by noisier updates.

5.4 SYSTEM MODELLING

The system states are defined as,

$$\hat{\mathbf{x}}_t = [\hat{\mathbf{r}}_{\mathcal{I}} \quad \hat{\mathbf{v}}_{\mathcal{I}} \quad \hat{\mathbf{q}}_{\mathcal{B}/\mathcal{O}} \quad \hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{O}}^{\mathcal{B}}]^T \quad (5.15)$$

Figure 5.5 illustrates the state variables defined within the Earth-Centered Inertial (ECI) reference frame. The terms $\mathbf{r}_{\mathcal{I}}$ and $\mathbf{v}_{\mathcal{I}}$ represent the satellite’s position and velocity in the ECI frame, respectively. The attitude of the satellite’s Body frame relative to the Orbital reference frame is denoted by $\mathbf{q}_{\mathcal{B}/\mathcal{O}}$, while the angular velocity of the Body frame relative to the Orbital frame, expressed in Body coordinates, is denoted by $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O}}^{\mathcal{B}}$.

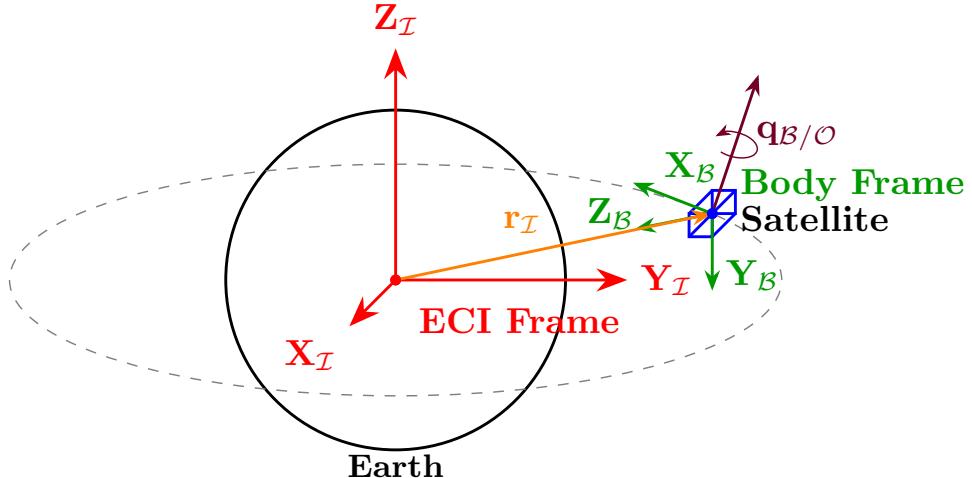


Figure 5.5: Satellite pose estimation states in the ECI reference frame

5.4.1 MOTION MODEL

The rotation of the satellite body is non-linear and the satellite motion from one timestep to the next when using Newton-Euler coupling can be described by the motion model \mathbf{f} ,

$$\hat{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) \quad (5.16)$$

which is expanded to,

$$\hat{\mathbf{x}}_t = \begin{bmatrix} r_{x,t} \\ r_{y,t} \\ r_{z,t} \\ v_{x,t} \\ v_{y,t} \\ v_{z,t} \\ q_{s,t} \\ q_{x,t} \\ q_{y,t} \\ q_{z,t} \\ \omega_{x,t} \\ \omega_{y,t} \\ \omega_{z,t} \end{bmatrix} = \hat{\mathbf{x}}_{t-1} + \begin{bmatrix} v_{x,t-1} \\ v_{y,t-1} \\ v_{z,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{x,t-1} + a_{J2,x,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{y,t-1} + a_{J2,y,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{z,t-1} + a_{J2,z,t-1} \\ \frac{1}{2}(-\omega_{x,t-1}q_{x,t-1} - \omega_{y,t-1}q_{y,t-1} - \omega_{z,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{x,t-1}q_{s,t-1} + \omega_{z,t-1}q_{y,t-1} - \omega_{y,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{y,t-1}q_{s,t-1} - \omega_{z,t-1}q_{x,t-1} + \omega_{x,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{z,t-1}q_{s,t-1} + \omega_{y,t-1}q_{x,t-1} - \omega_{x,t-1}q_{y,t-1}) \\ \frac{1}{I_{xx}}(I_{yy} - I_{zz})\omega_{y,t-1}\omega_{z,t-1} \\ \frac{1}{I_{yy}}(I_{zz} - I_{xx})\omega_{z,t-1}\omega_{x,t-1} \\ \frac{1}{I_{zz}}(I_{xx} - I_{yy})\omega_{x,t-1}\omega_{y,t-1} \end{bmatrix} \Delta t + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ T_x/I_x \\ T_y/I_y \\ T_z/I_z \end{bmatrix} \quad (5.17)$$

The body-fixed axes of the target are chosen to coincide with its principle axes of inertia. The principle moment of inertia are given as,

$$\mathbf{I}_B = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (5.18)$$

External torques, T_x , T_y and T_z , within the model are assumed to be zero and there is thus no control input, \mathbf{u}_t .

Thus

$$\hat{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_{t-1}) \quad (5.19)$$

where

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \begin{bmatrix} r_{x,t} \\ r_{y,t} \\ r_{z,t} \\ v_{x,t} \\ v_{y,t} \\ v_{z,t} \\ q_{s,t} \\ q_{x,t} \\ q_{y,t} \\ q_{z,t} \\ \omega_{x,t} \\ \omega_{y,t} \\ \omega_{z,t} \end{bmatrix} = \hat{\mathbf{x}}_{t-1} + \begin{bmatrix} v_{x,t-1} \\ v_{y,t-1} \\ v_{z,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{x,t-1} + a_{J2,x,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{y,t-1} + a_{J2,y,t-1} \\ -\frac{\mu}{|\mathbf{r}|^3} \cdot r_{z,t-1} + a_{J2,z,t-1} \\ \frac{1}{2}(-\omega_{x,t-1}q_{x,t-1} - \omega_{y,t-1}q_{y,t-1} - \omega_{z,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{x,t-1}q_{s,t-1} + \omega_{z,t-1}q_{y,t-1} - \omega_{y,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{y,t-1}q_{s,t-1} - \omega_{z,t-1}q_{x,t-1} + \omega_{x,t-1}q_{z,t-1}) \\ \frac{1}{2}(\omega_{z,t-1}q_{s,t-1} + \omega_{y,t-1}q_{x,t-1} - \omega_{x,t-1}q_{y,t-1}) \\ \frac{1}{I_{xx}}(I_{yy} - I_{zz})\omega_{y,t-1}\omega_{z,t-1} \\ \frac{1}{I_{yy}}(I_{zz} - I_{xx})\omega_{z,t-1}\omega_{x,t-1} \\ \frac{1}{I_{zz}}(I_{xx} - I_{yy})\omega_{x,t-1}\omega_{y,t-1} \end{bmatrix} \Delta t \quad (5.20)$$

Equation 5.20 illustrates that the position \mathbf{r}_I of the satellite only depends on the velocity \mathbf{v}_I , whereas the velocity depends on the acceleration, defined by the Newton-Euler equation. The acceleration of the model is composed of the gravitational acceleration \mathbf{a}_G and the \mathbf{a}_{J2} which both were defined in Section 3.2.2. The attitude of the satellite is determined by the quaternion dynamic equation defined in Equation 3.12 and the angular velocity is defined with principle axis coupling.

5.4.2 SENSOR MEASUREMENT MODEL

This section parallels Section 3.4. While the earlier section constructed the measurement equations from the true satellite states, here the focus shifts to the *estimated* states. These provide the basis for predicting sensor measurements within the Extended Kalman Filter (EKF). The predicted outputs $\hat{\mathbf{z}}$ are then compared to the actual noisy measurements \mathbf{z} to form the innovation $\hat{\mathbf{y}}$.

In general, each sensor measurement model can be written as

$$\hat{\mathbf{z}} = \mathbf{h}(\hat{\mathbf{x}}_t), \quad (5.21)$$

where $\hat{\mathbf{x}}_t$ is the current state estimate. Since $\mathbf{h}(\cdot)$ is nonlinear, it is linearised about $\hat{\mathbf{x}}_t$ to obtain the measurement Jacobian

$$\mathbf{H}_t = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_t}. \quad (5.22)$$

The resulting Jacobian is used in the EKF update step. The following subsections

summarise the sensor-specific forms of $\mathbf{h}(\hat{\mathbf{x}}_t)$.

5.4.2.1 EARTH TRACKER MEASUREMENT MODEL

The Earth Tracker sends two inputs to the EKF, the measurement of the Earth Tracker itself

$$\mathbf{z}_{ET} = \mathbf{f}_{\mathcal{B}} = \begin{bmatrix} x_{\mathcal{B}} \\ y_{\mathcal{B}} \\ z_{\mathcal{B}} \end{bmatrix} \quad (5.23)$$

and the geolocated feature vector through the catalogue generation described in Section 4.5.4 also referred to as the catalogue vector \mathbf{p}_{LLA} .

To transform this vector into a vector to be compared to the measurement, the following transformations are used

$$\mathbf{f}_{\mathcal{R}} = f(\mathbf{p}_{LLA}, WGS84) \quad (5.24)$$

and

$$\hat{\mathbf{f}}_{\mathcal{B}}^+ = \hat{\mathbf{T}}_{\mathcal{O},t}^{\mathcal{B}} \times \hat{\mathbf{T}}_{\mathcal{I},t}^{\mathcal{O}} \times \hat{\mathbf{T}}_{\mathcal{R},t}^{\mathcal{I}} \times \mathbf{f}_{\mathcal{R}}^+ \quad (5.25)$$

where $\hat{\mathbf{T}}_{\mathcal{O},t}^{\mathcal{B}}$ is calculated using the estimated state $\hat{\mathbf{q}}_{\mathcal{B}/\mathcal{O}}$, $\hat{\mathbf{T}}_{\mathcal{I},t}^{\mathcal{O}}$ is calculated using the estimated state $\hat{\mathbf{r}}_{\mathcal{I}}$ and $\hat{\mathbf{v}}_{\mathcal{I}}$, and $\hat{\mathbf{T}}_{\mathcal{R},t}^{\mathcal{I}}$ is calculated by the rotation speed of the Earth ω_e . This is done for each for each feature vector measurement, where the estimated Earth Tracker measurement is defined as:

$$\hat{\mathbf{z}}_{ET} = \hat{\mathbf{f}}_{\mathcal{B}} = \mathbf{h}(\hat{\mathbf{x}}_t, \mathbf{p}_{LLA}, WGS84). \quad (5.26)$$

5.4.2.2 GPS MEASUREMENT MODEL

The GPS receiver outputs measurements in geodetic coordinates (ϕ, λ, h) (latitude, longitude, and altitude). These must first be transformed into the Earth-Centered Earth-Fixed (ECEF) frame \mathcal{R} from the ECI frame \mathcal{I} using the estimated state information. The predicted GPS measurement is therefore expressed as

$$\hat{\mathbf{r}}_{\mathcal{R}}^+ = \hat{\mathbf{T}}_{\mathcal{I},t}^{\mathcal{R}} \times \hat{\mathbf{r}}_{\mathcal{I}}^+ \quad (5.27)$$

$$\hat{\mathbf{z}}_{GPS} = f(\hat{\mathbf{r}}_{\mathcal{R}}, WGS84) \quad (5.28)$$

$$\hat{\mathbf{z}}_{GPS} = \mathbf{h}_{GPS}(\hat{\mathbf{x}}_t, WGS84), \quad (5.29)$$

where $\hat{\mathbf{r}}_{\mathcal{I}}$ is the estimated position vector in the inertial frame and $\hat{\mathbf{T}}_{\mathcal{I},t}^{\mathcal{R}}$ is the transformation from ECI to ECEF at time t . The WGS84 reference model is used for the ECEF-to-geodetic conversion.

5.4.2.3 GYROSCOPE MEASUREMENT MODEL

The gyroscope measures the angular velocity of the body frame relative to the inertial frame, expressed in the body frame, denoted as $\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}}$. Using the estimated states, the predicted measurement is

$$\hat{\mathbf{z}}_{\text{Gyro}} = \hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{I}} = \hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{O}} + \hat{\boldsymbol{\omega}}_{\mathcal{O}/\mathcal{I}}, \quad (5.30)$$

where $\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{O}}$ is taken directly from the estimated state vector, while $\hat{\boldsymbol{\omega}}_{\mathcal{O}/\mathcal{I}}$ is computed from the estimated orbital dynamics as

$$\hat{\boldsymbol{\omega}}_{\mathcal{O}/\mathcal{I}}^{\mathcal{O}} = -\hat{\omega}_o \hat{\mathbf{y}}_{\mathcal{O}}, \quad (5.31)$$

with

$$\hat{\omega}_o = \sqrt{\frac{\mu}{||\hat{\mathbf{r}}_{\mathcal{I}}||^3}}, \quad (5.32)$$

where $\hat{\mathbf{r}}_{\mathcal{I}}$ is the estimated position in the inertial frame and $\hat{\omega}_o$ is the magnitude of the angular velocity given a circular orbit. This vector is then expressed in the body frame using the estimated attitude DCM:

$$\hat{\boldsymbol{\omega}}_{\mathcal{O}/\mathcal{I}}^{\mathcal{B}} = \hat{\mathbf{A}}_{\mathcal{O}}^{\mathcal{B}} \times \hat{\boldsymbol{\omega}}_{\mathcal{O}/\mathcal{I}}^{\mathcal{O}}. \quad (5.33)$$

The measurement can thus be expressed as:

$$\hat{\mathbf{z}}_{\text{GYR}} = \mathbf{h}_{\text{GYR}}(\hat{\mathbf{x}}_t, \mu), \quad (5.34)$$

5.4.2.4 COARSE SUN SENSOR MEASUREMENT MODEL

The Sun direction in the inertial frame is modeled as the fixed unit vector

$$\hat{\mathbf{S}}_{\mathcal{I}} = [1 \ 0 \ 0]^T, \quad (5.35)$$

and is transformed into the body frame using the estimated attitude:

$$\hat{\mathbf{S}}_{\mathcal{B}}^+ = \hat{\mathbf{T}}_{\mathcal{O}}^{\mathcal{B}} \times \hat{\mathbf{T}}_{\mathcal{I}}^{\mathcal{O}} \times \hat{\mathbf{S}}_{\mathcal{I}}^+, \quad (5.36)$$

A satellite normally carries six CSS units aligned with the $\pm X$, $\pm Y$, and $\pm Z$ body axes. The predicted response of the i -th sensor is given by

$$\hat{z}_i = \max \left(0, \hat{\mathbf{n}}_i^\top \hat{\mathbf{S}}_{\mathcal{B}} \right), \quad i = 1, \dots, 6, \quad (5.37)$$

where $\hat{\mathbf{n}}_i$ is the known unit normal vector of the i -th face. Negative values are clamped to zero to reflect the physical sensor limitation.

Collecting all sensor outputs, the nonlinear measurement model is written as

$$\hat{\mathbf{z}}_{\text{CSS}} = \mathbf{h}_{\text{CSS}}(\hat{\mathbf{x}}_t, \hat{\mathbf{S}}_{\mathcal{I}}). \quad (5.38)$$

5.4.2.5 MAGNETOMETER MEASUREMENT MODEL

The magnetometer provides a measurement of the local geomagnetic field vector in the spacecraft body frame. In the estimator, the predicted magnetic field is generated using the estimated spacecraft position and attitude.

The local magnetic field is obtained from the International Geomagnetic Reference Field (IGRF). The estimated spacecraft position in the inertial frame $\hat{\mathbf{r}}_{\mathcal{I}}$ is first converted into latitude, longitude, and altitude (LLA) using the WGS84 Earth model:

$$\begin{aligned} \hat{\mathbf{r}}_{\mathcal{R}} &= \hat{\mathbf{A}}_{\mathcal{I}}^{\mathcal{R}} \times \hat{\mathbf{r}}_{\mathcal{I}}, \\ \hat{\mathbf{r}}_{\mathcal{L}} &= f(\text{WGS84}, \hat{\mathbf{r}}_{\mathcal{R}}). \end{aligned} \quad (5.39)$$

The LLA coordinates are then passed to the IGRF model to compute the geomagnetic field vector in the NED \mathcal{N} frame:

$$\hat{\mathbf{z}}_{\text{MAG}, \mathcal{N}} = \text{wrldmagn}[\hat{\mathbf{r}}_{\mathcal{L}}, \text{decimalYear}]. \quad (5.40)$$

This is subsequently rotated into the spacecraft body frame using the estimated attitude:

$$\hat{\mathbf{z}}_{\text{MAG}, \mathcal{B}}^+ = \hat{\mathbf{T}}_{\mathcal{O}}^{\mathcal{B}} \times \hat{\mathbf{T}}_{\mathcal{I}}^{\mathcal{O}} \times \hat{\mathbf{T}}_{\mathcal{R}}^{\mathcal{I}} \times \hat{\mathbf{T}}_{\mathcal{N}}^{\mathcal{R}} \times \hat{\mathbf{z}}_{\text{MAG}, \mathcal{N}}^+. \quad (5.41)$$

The nonlinear measurement model is therefore expressed as

$$\hat{\mathbf{z}}_{\text{MAG}} = \mathbf{h}_{\text{MAG}}(\hat{\mathbf{x}}_t, \text{IGRF}). \quad (5.42)$$

5.4.2.6 STAR TRACKER MEASUREMENT MODEL

Using the estimated attitude, the predicted star vector in the body frame is given by

$$\hat{\mathbf{s}}_{i, \mathcal{B}}^+ = \hat{\mathbf{T}}_{\mathcal{O}}^{\mathcal{B}} \times \hat{\mathbf{T}}_{\mathcal{I}}^{\mathcal{O}} \times \mathbf{s}_{i, \mathcal{I}}^+, \quad (5.43)$$

where $\hat{\mathbf{T}}_{\mathcal{O}}^{\mathcal{B}}$ and $\hat{\mathbf{T}}_{\mathcal{I}}^{\mathcal{O}}$ are transformation matrices derived from the estimated attitude quaternion $\hat{\mathbf{q}}_t$ and position $\hat{\mathbf{r}}_t$. The same star catalogue $\boldsymbol{\Gamma}$, defined in the inertial frame, is used as the reference.

The corresponding predicted measurement is then

$$\hat{\mathbf{z}}_{\text{ST},i} = \hat{\mathbf{s}}_{i,\mathcal{B}}^+. \quad (5.44)$$

This predicted measurement is compared to the actual noisy measurement from the sensor, defined in (3.4.4), as part of the EKF update step.

In the EKF, the measurement function for the star tracker can be expressed as

$$\hat{\mathbf{z}}_{\text{ST}} = \mathbf{h}_{\text{ST}}(\hat{\mathbf{x}}_t, \boldsymbol{\Gamma}), \quad (5.45)$$

5.5 SYSTEM INITIALIZATION

The orbital system is initialized using the following parameters:

- * **Latitude**: Initial geodetic latitude of the satellite.
- * **Longitude**: Initial geodetic longitude of the satellite.
- * **Altitude**: Initial altitude above the WGS84 ellipsoid.
- * **Roll (X-axis)**: Initial roll angle of the satellite camera with respect to the orbital reference frame.
- * **Pitch (Y-axis)**: Initial pitch angle of the satellite camera with respect to the orbital reference frame.
- * **Yaw (Z-axis)**: Initial yaw angle of the satellite camera with respect to the orbital reference frame.

5.5.1 POSITION INITIALISATION

The geodetic latitude, longitude, and altitude are first converted to an inertial position vector using the WGS84 Earth model. This involves a transformation from the local geodetic frame to the Earth-Centered Earth-Fixed (ECEF) frame, followed by a rotation into the inertial frame:

$$\mathbf{r}_{\mathcal{R}} = f(\mathbf{r}_{\mathcal{L}}, \omega_e, \text{WGS84}) \quad (5.46)$$

$$\mathbf{r}_{\mathcal{I}} = \mathbf{A}_{\mathcal{R}}^{\mathcal{I}} \times \mathbf{r}_{\mathcal{R}} \quad (5.47)$$

5.5.2 VELOCITY INITIALISATION

To compute the initial velocity vector, it is assumed that the satellite is in a near-circular orbit, and thus its velocity vector is orthogonal to its position vector. While multiple solutions satisfy this constraint, the velocity direction is resolved using the local east unit vector, which is always tangential to the satellite's position on the Earth's surface. The eastward direction is given by:

$$\mathbf{u}_{\text{east}} = \frac{1}{\|\cdot\|} \begin{bmatrix} -\sin(\lambda) \\ \cos(\lambda) \\ 0 \end{bmatrix} \quad (5.48)$$

where λ is the geodetic longitude. The vector is normalized to obtain a unit direction. This unit direction can then be rotated to increase the inclination of the orbit, where the inclination of the orbit is constrained by $\min(\lambda)$ and $\max(\lambda + 90^\circ)$.

The magnitude of the orbital velocity is computed using the equation:

$$\|\mathbf{v}\| = \sqrt{\frac{\mu}{\|\mathbf{r}\|}} \quad (5.49)$$

where μ is the standard gravitational parameter, and $\|\mathbf{r}\|$ is the norm of the inertial position vector.

The inertial velocity vector is then calculated as:

$$\mathbf{v}_{\mathcal{I}} = \|\mathbf{v}\| \cdot \mathbf{u}_{\text{east}} \quad (5.50)$$

5.5.3 ATTITUDE INITIALISATION

The initial attitude of the satellite is determined by multiplying two DCMs and then converting the DCM to a quaternion $\mathbf{q}_{\mathcal{B}/\mathcal{O}}$.

$$\mathbf{A}_{\mathcal{O}}^{\mathcal{B}} = \mathbf{A}_{\mathcal{C}}^{\mathcal{B}} \times \mathbf{A}_{\mathcal{O}}^{\mathcal{C}} \quad (5.51)$$

where $\mathbf{A}_{\mathcal{C}}^{\mathcal{B}}$ is the DCM defining the orientation of the Body frame relative to the Camera frame, and $\mathbf{A}_{\mathcal{O}}^{\mathcal{C}}$ is the DCM for the Camera frame relative to the Orbital frame. The matrix $\mathbf{A}_{\mathcal{O}}^{\mathcal{C}}$ is computed from the initial roll, pitch, and yaw parameters using the 3-2-1 Euler angle rotation sequence discussed in Section 3.2.1.

Finally, the resulting direction cosine matrix is converted into its equivalent unit quaternion representation, $\mathbf{q}_{\mathcal{B}/\mathcal{O}}$, which is used to initialize the attitude state in the filter.

$$\mathbf{q}_{\mathcal{B}/\mathcal{O}} = \text{rotm2quat}(\mathbf{A}_{\mathcal{O}}^{\mathcal{B}})$$

5.5.4 ANGULAR VELOCITY INITIALISATION

The initial angular velocity is defined as a rotation of the Camera frame, \mathcal{C} , relative to the Orbital frame, \mathcal{O} . To be used in the state vector, this velocity must be transformed into the Body frame, \mathcal{B} . The total angular velocity of the Body frame relative to the Orbital frame is determined using the angular velocity addition formula:

$$\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O}}^{\mathcal{B}} = \boldsymbol{\omega}_{\mathcal{B}/\mathcal{C}}^{\mathcal{B}} + \boldsymbol{\omega}_{\mathcal{C}/\mathcal{O}}^{\mathcal{B}} \quad (5.52)$$

Each term is considered separately. First, assuming the camera is rigidly mounted to the satellite's body, their relative angular velocity is zero:

$$\boldsymbol{\omega}_{\mathcal{B}/\mathcal{C}}^{\mathcal{B}} = \mathbf{0} \quad (5.53)$$

Second, the angular velocity of the Camera frame relative to the Orbital frame, which is initially given in Camera coordinates ($\boldsymbol{\omega}_{\mathcal{C}/\mathcal{O}}^{\mathcal{C}}$), must be transformed into Body frame coordinates. This is achieved by pre-multiplying the vector by the appropriate DCM:

$$\boldsymbol{\omega}_{\mathcal{C}/\mathcal{O}}^{\mathcal{B}} = \mathbf{A}_{\mathcal{C}}^{\mathcal{B}} \times \boldsymbol{\omega}_{\mathcal{C}/\mathcal{O}}^{\mathcal{C}} \quad (5.54)$$

Combining these terms yields the final expression for the initial angular velocity of the satellite.

5.6 SYSTEM INTEGRATION

5.6.1 SATELLITE SYSTEM

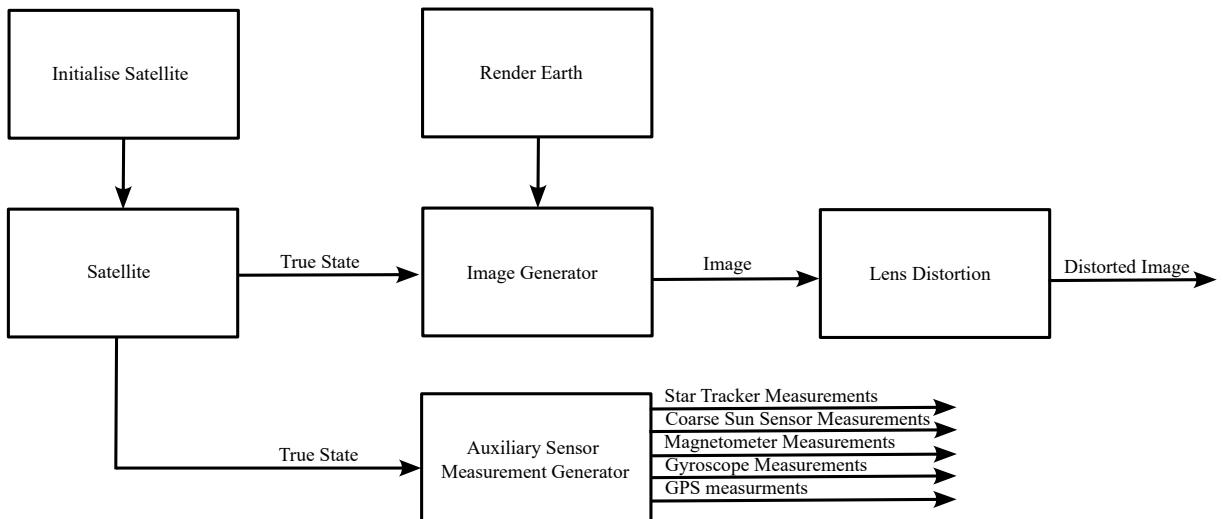


Figure 5.6: Block diagram of the satellite simulation architecture. The 'true' motion model provides the ground truth state to two parallel pipelines: one generating distorted camera images and the other generating auxiliary sensor data.

As illustrated in the Figure 5.6, the simulation architecture is centered around a "true" motion model that generates the satellite's ground truth state after initialization. This true state is then propagated down two parallel paths.

In the first path, the state is sent to the Image Generator, which uses a rendered model of the Earth to create a clean image as seen from the satellite's perspective. This image is subsequently processed by a lens distortion function that applies realistic tangential, radial, and chromatic distortions. In the second path, the true state is concurrently fed into an Auxiliary Sensor Measurement Generator. This component produces simulated satellite sensor suite data, which is used to validate and test the sensor fusion capabilities of the Earth Tracker system.

5.6.2 EARTH TRACKER SYSTEM

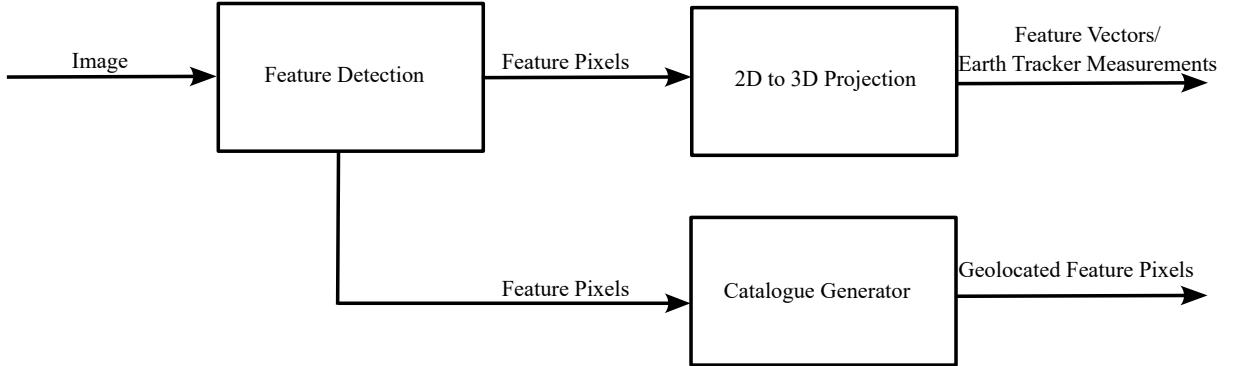


Figure 5.7: Block diagram of the Earth Tracker System’s data processing pipeline. An input image undergoes feature detection, and the resulting feature pixels are processed through two concurrent streams: 2D-to-3D projection to generate the primary measurement vector, and catalogue generation to create geolocated features for pose estimation.

Within the Earth Tracker system, an incoming image, either distorted or undistorted, is first processed for feature detection. Utilizing one of several algorithms (such as SIFT, SURF, or ORB, as detailed in Section 4.5.2), the system identifies key feature pixels. These detected features are then channeled into two concurrent processes.

The first process is a 2D-to-3D projection, which reconstructs spatial feature vectors from the feature pixels to produce the primary Earth Tracker measurements. Simultaneously, the second process sends the feature pixels to a Catalogue Generator. This module creates a catalogue of geolocated features required for the pose estimation algorithm.

It is important to note that this methodology is employed because features are detected dynamically in each image. As implementing true geolocation via feature matching is beyond the scope of this work, a ray-casting technique is used.

5.6.3 POSE ESTIMATION SYSTEM

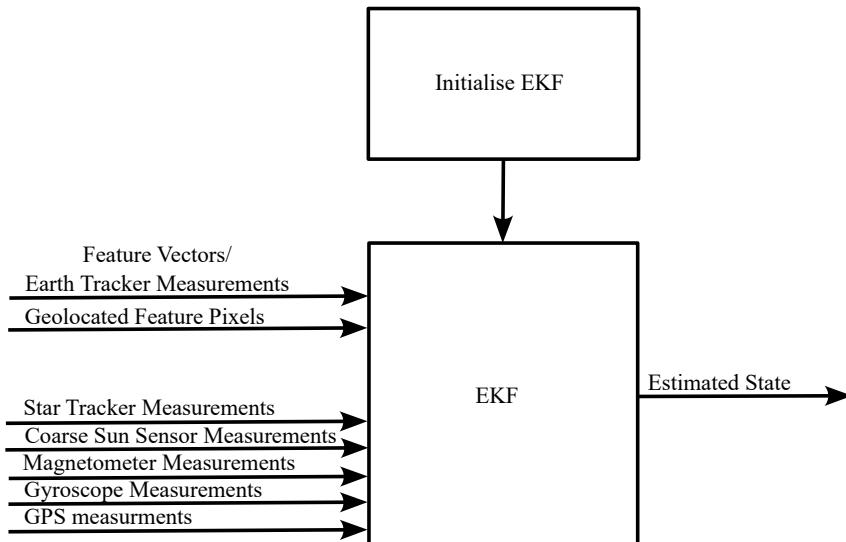


Figure 5.8: Block diagram of the Pose Estimation System. A sequential update EKF fuses the Earth Tracker measurement, the geolocated catalogue vector, and auxiliary sensor data to produce an optimal estimate of the satellite’s state.

The pose estimation system, after initialised, takes in all the auxiliary sensor measurements, Earth Tracker measurements and Catalogue vectors and uses the sequential update EKF to estimate the current state of the system.

5.7 CONCLUSION

This chapter has detailed the theoretical formulation and practical implementation of the state estimation system for the satellite. Beginning with the fundamentals of recursive Bayesian estimation, the Extended Kalman Filter (EKF) was selected as the most suitable framework, given its proven ability to handle the non-linear dynamics and high-dimensional state space inherent to the pose estimation problem.

A comprehensive 13-element state vector, comprising the satellite’s inertial position and velocity, attitude quaternion, and body-frame angular velocity, was defined. The corresponding non-linear motion model was developed by integrating orbital mechanics, quaternion kinematics, and Euler’s equations of rigid body motion to accurately propagate the state over time. A key contribution of this work was the derivation of the measurement models required by the filter. This included a novel model for the vision-based Earth Tracker and a suite of models for the auxiliary sensors, including GPS, gyroscopes, sun sensors, magnetometers, and a star tracker, which map the estimated state into the respective sensor measurement spaces.

To effectively integrate data from this heterogeneous and asynchronous sensor suite, a sequential update EKF architecture was adopted. This strategy allows the filter to process measurements as they become available, ensuring timely and robust state corrections. Furthermore, a detailed initialization procedure for the state vector, derived from a set of initial orbital and attitude parameters, was established to ensure a valid and stable starting point for the recursive estimation process.

With the estimator's architecture, state vector, dynamic models, and measurement models now fully defined, the system is complete. The subsequent chapter will present the simulation results and evaluate the performance of this state estimation framework under various operational scenarios.

CHAPTER 6

EXPERIMENTS

6.1 INTRODUCTION

This aim of this chapter is to verify the theory that what provided in the previous chapters using the simulation environment that is developed. This chapter will begin with describing the test setup in the simulation environment, afterwhich three tests will be run. The first test will determine the accuracy of the sensor itself and the robustness of the state estimator in sensor fusion. Then a test is run where the systems ability to accurately determine the pose of the satellite depending of the amount of features present in the image. Lastly, a test against the robustness of the system against temporal distortions.

6.2 TEST CONFIGURATION

The simulation enviroment is developed in MATLAB. Each test wil be run in a high inclined orbit over Cape Town.

The camera parameters will be standardised to the following values. The camera is based on the TriScape100x satellite camera from Simera Sense, which both has the same focal length, but the resolution as decreased so that the simulation can porcess the images faster. And the Pixel size has been increased as the smallest GSD gathers from the copernicus browser has a GSD of 15m.



Figure 6.1: The simera sense TriScape Camera

Characteristic	Simulated Camera	Simera Sense Camera	Units
Horizontal Resolution	720	4096	pixel
Vertical Resolution	720	3072	pixel
Focal Length	580	580	mm
Pitch	17.4	5.5	μm
GSD @ 500 km	15	4.75	m/pixel
Swath	10.8	19.4	km

And the initial conditions of the satellite are indicated in the values below, these are considered as the default values. If a value is changed, it will explicitly be mentioned in the subsequent test.

Constants		
Attribute	Simulator	Filter
Sample Rate	30 Hz	30 Hz
GM	$3.986 \times 10^5 \text{ km}^3/\text{s}^2$	$3.986 \times 10^5 \text{ km}^3/\text{s}^2$
R_{earth}	$6.378 \times 10^3 \text{ km}$	$6.378 \times 10^3 \text{ km}$
J_2	1.082×10^{-3}	0
ω_e	$7.292 \times 10^{-5} \text{ rad/s}$	$7.292 \times 10^{-5} \text{ rad/s}$
<i>InertiaTensor</i>	$\text{diag}(1, 1, 1) \text{ kg} \cdot \text{m}^2$	$\text{diag}(1.05, 0.95, 1.02) \text{ kg} \cdot \text{m}^2$
Initial States		
Attribute	Simulator	Filter
Latitude	-33.90°	-33.91°
Longitude	18.41°	18.42°
Altitude	500 km	500 km
Roll	0°	0°
Pitch	0.5°	0°
Yaw	0°	0°
Roll Rate	$2^\circ/\text{s}$	$0^\circ/\text{s}$
Pitch Rate	$0^\circ/\text{s}$	$0^\circ/\text{s}$
Yaw Rate	$0^\circ/\text{s}$	$0^\circ/\text{s}$

The EKF Covariance Matrix P and Process Noise Matrix Q is also initialised using the following parameters

6.3. SENSOR TEST

Attribute	P	Q
r_x	1000 m	100 m
r_y	1000 m	100 m
r_z	1000 m	100 m
v_x	100 m/s	50 m/s
v_y	100 m/s	50 m/s
v_z	100 m/s	50 m/s
Att	10°	5°
w_x	1°/s	0.5°/s
w_y	1°/s	0.5°/s
w_z	1°/s	0.5°/s

6.3 SENSOR TEST

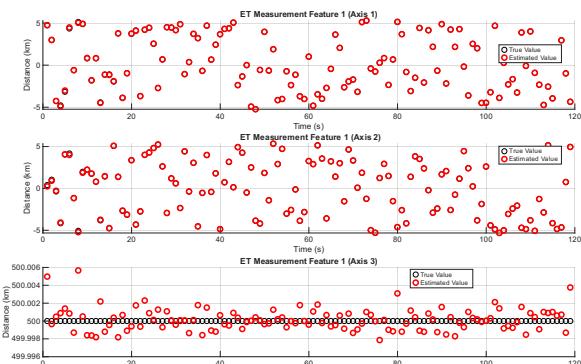
In the sensor test we are going to evaluate the performance of the Earth Tracker.

6.3.1 EARTH TRACKER ALONE

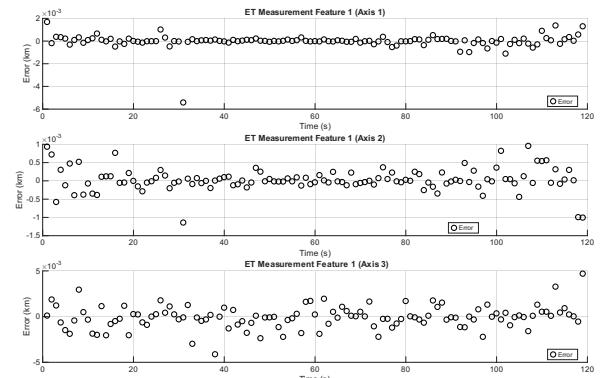
In this section we are going to test the Earth Tracker alone to see it's capabilities. The Earth Tracker is calibrated as follows.

Sensor	Sample Rate	Noise
ET	1 Hz	10 m
ET	1 Hz	1500 m

Table 6.1: Sensor characteristics including sample rate, noise, and drift.



(a) ET Feature 1 true and estimated measurement



(b) The error in feature 1 true and estimated measurements

Figure 6.2: Earth Tracker Measruement and measruemetn error of Feature 1

6.3. SENSOR TEST

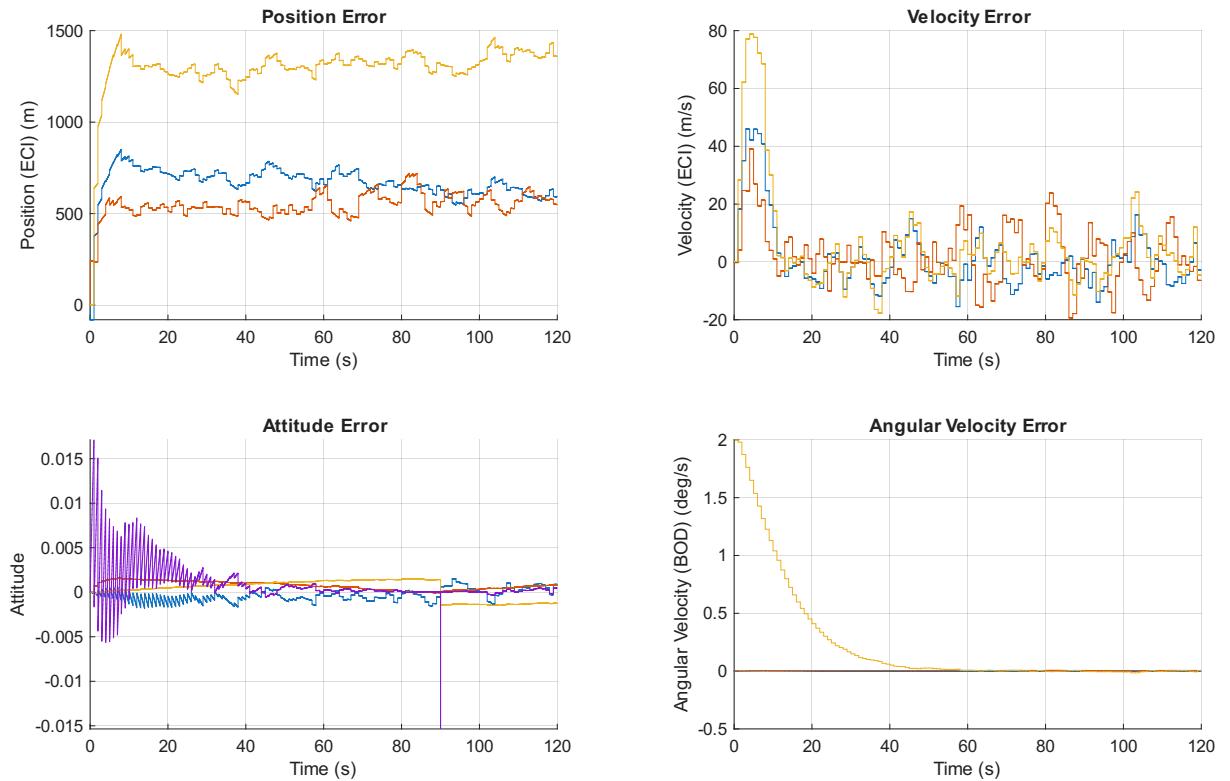


Figure 6.3: The error graph between the true and estimated state generated by the Eart Tracker

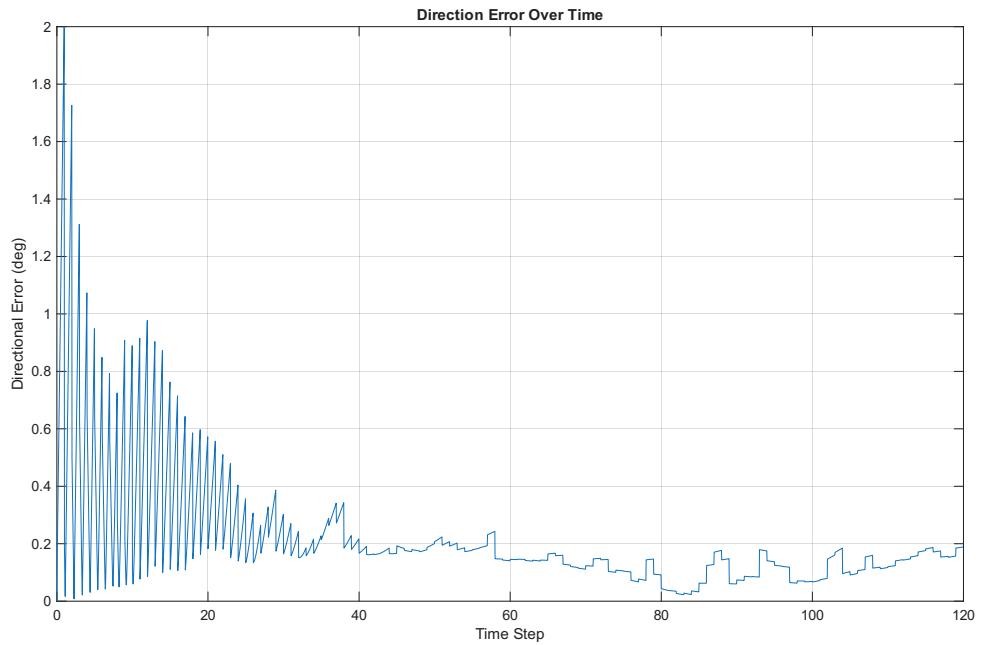


Figure 6.4: The Attitude Error of the Earth Tracker

In Figure ?? we can see that the Earth Tracker can estimated the position an attitude quite well, there is a large position error that is about 1,4km in the z- direction this is

expected as the Earth tracker only uses the lens characteristics to estimate the position of the satellite, but for attitude estimation we can see that there is maximum error of 2° and an average error of 0.208° , this means that the ET tracker and definitely be used as an attitude sensor alone if needed.

6.3.2 THE ADCS SUITE

In this test we want to know if it would be viable to replace the star tracker entirely with an Earth Tracker which is a sensor already on the satellite being under utilised. So the test conducted below is of the combination of the GPS, GYRO, CSS, MAG.

It is important to note that the order of which the sensors are updated are crucial to the results. In this simulation the order is ;

Rank	Sensor	Reason
1st	GPS	Adds Position estimation which will aid the Attitude Estimation Sensors
2nd	Gyro	Adds direct Angular Velocity Measurements that will aid the Attitude Estimation Sensors
3rd	CSS	Is the most coarse of the Attitude Sensors
4th	MAG	Is less coarse than the CSS but less fine than the Star Tracker
5th	ET	The Earth Tracker provides estimation of the full state of the system, expected to
6th	ST	The most accurate Sensor

Table 6.2: Sensor ranking by filter order and justification.

The sensors are calibrated as follows.

Sensor	Sample Rate	Noise	Drift
GPS	10 Hz	5 m	0.1 m/ \sqrt{hr}
Gyro	30 Hz	0.1 $^\circ$	0.01 $^\circ/\sqrt{hr}$
CSS	10 Hz	5 $^\circ$	
MAG	10 Hz	500 nT	

Table 6.3: Sensor characteristics including sample rate, noise, and drift.

6.3. SENSOR TEST

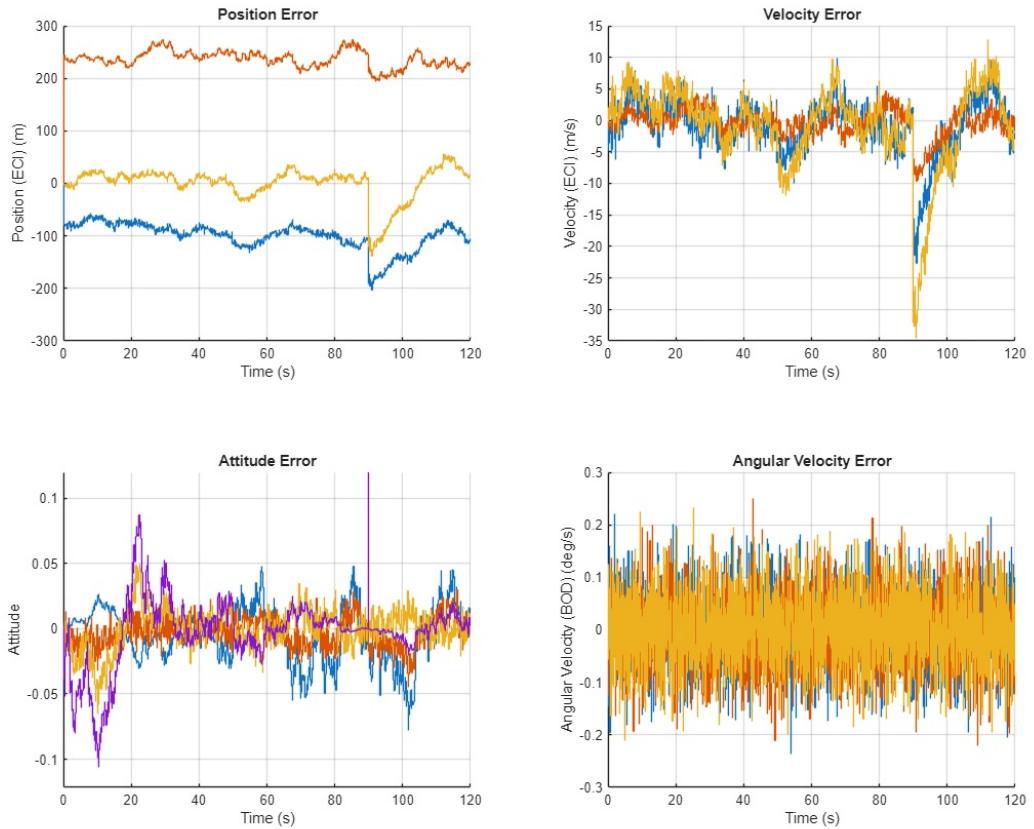


Figure 6.5: The error graph between the true and estimated state generated by the ADCS suite

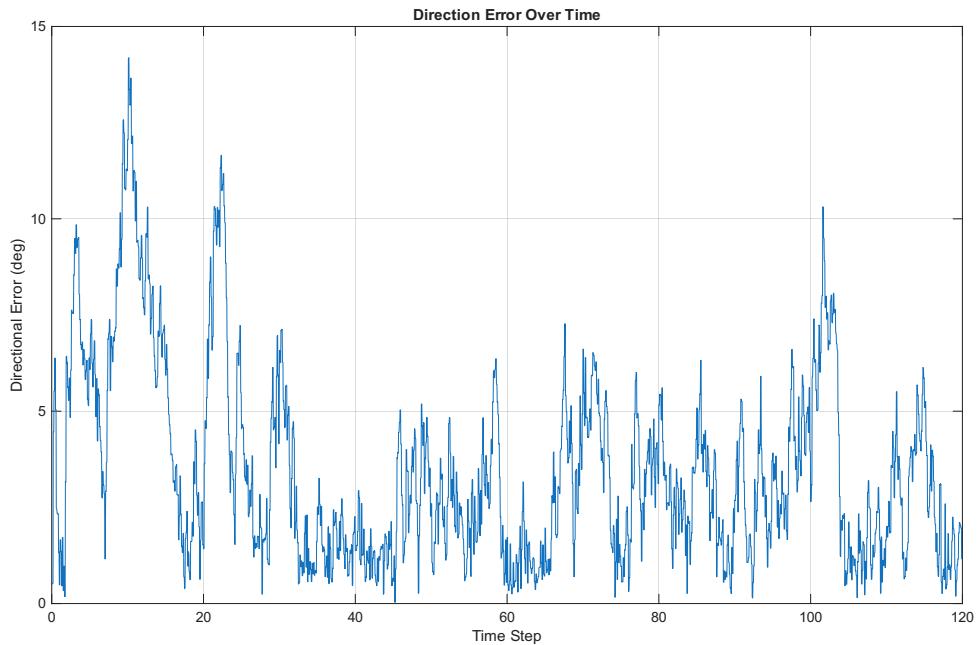


Figure 6.6: The Attitude Error of the ADCS Suite

In Figure ?? we can see that there is a constant position error in the x and y position coordinates, x begin around -100 m and y being around -250 m this can be because the main position sensor is the GPS and it measures x and Y coordinates in latitude and longitude and these are hard to convert in the measurement model. We can also see an attitude direction error of a maximum of 14° and a average of 2.84° .

6.3.3 EARTH TRACKER VS STAR TRACKER

In this section we are going to compare the Earth Tracker and the Star Tracker Head On.

The star tracker is initialised with this characteristics

Sensor	Sample Rate	Noise
ST	1 Hz	30 arcseconds

Table 6.4: Sensor characteristics including sample rate, noise, and drift.

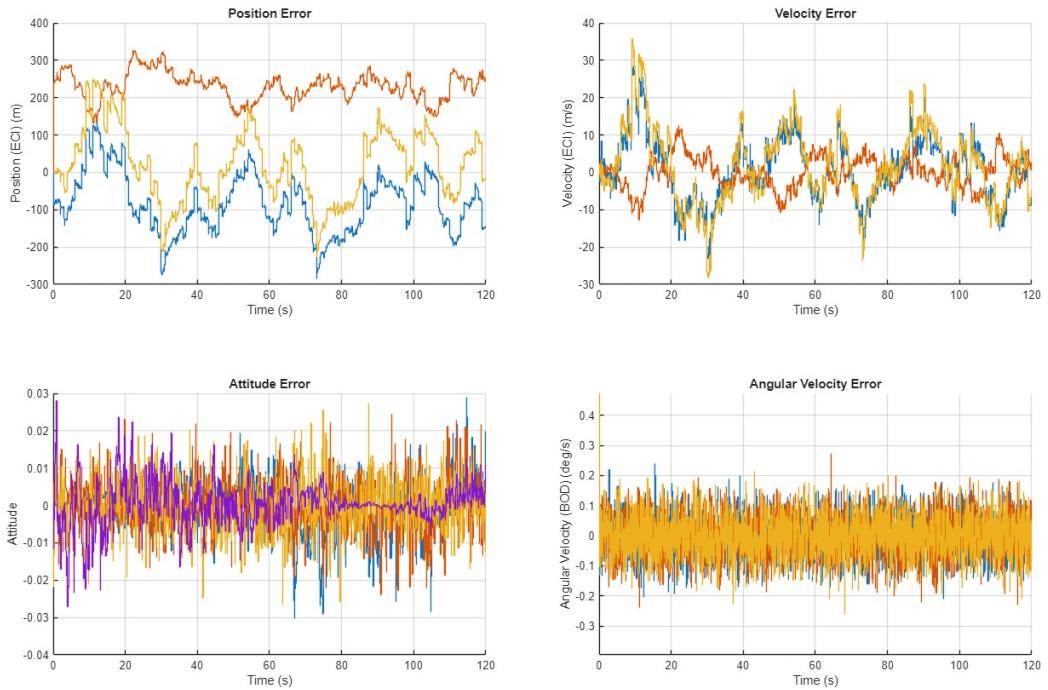


Figure 6.7: The error graph between the true and estimated state generated by the ADCS suite

6.3. SENSOR TEST

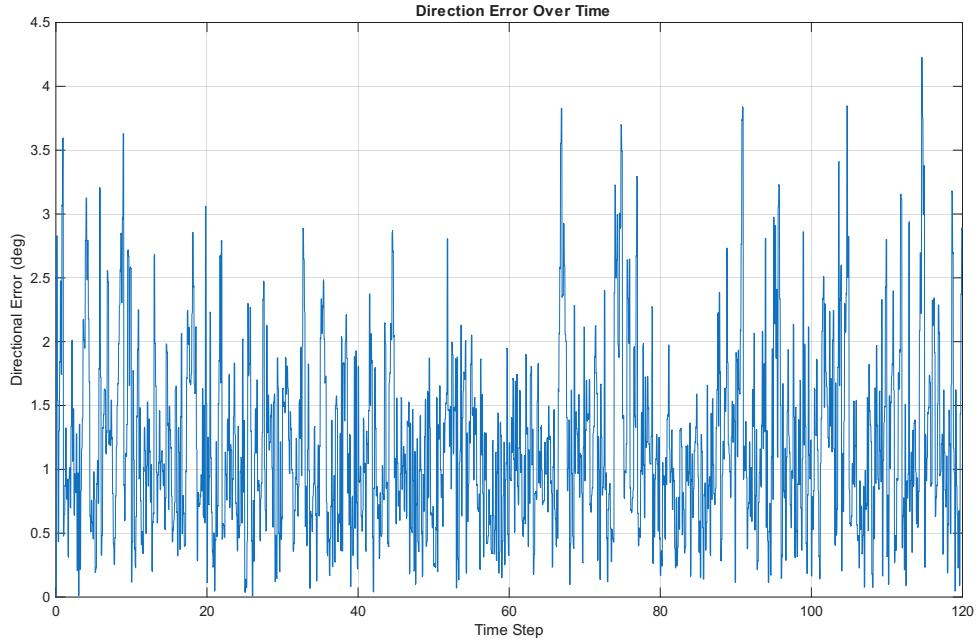


Figure 6.8: The Attitude Error of the ADCS Suite

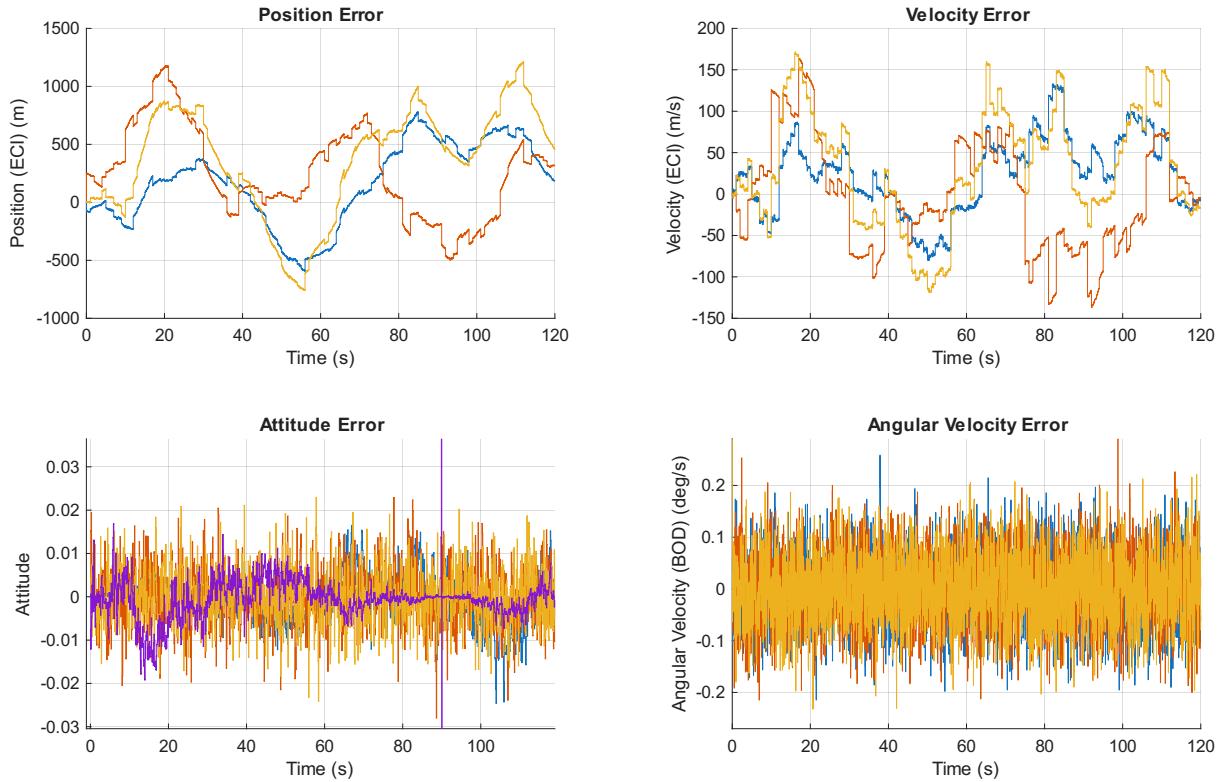


Figure 6.9: The error graph between the true and estimated state generated by the ADCS suite

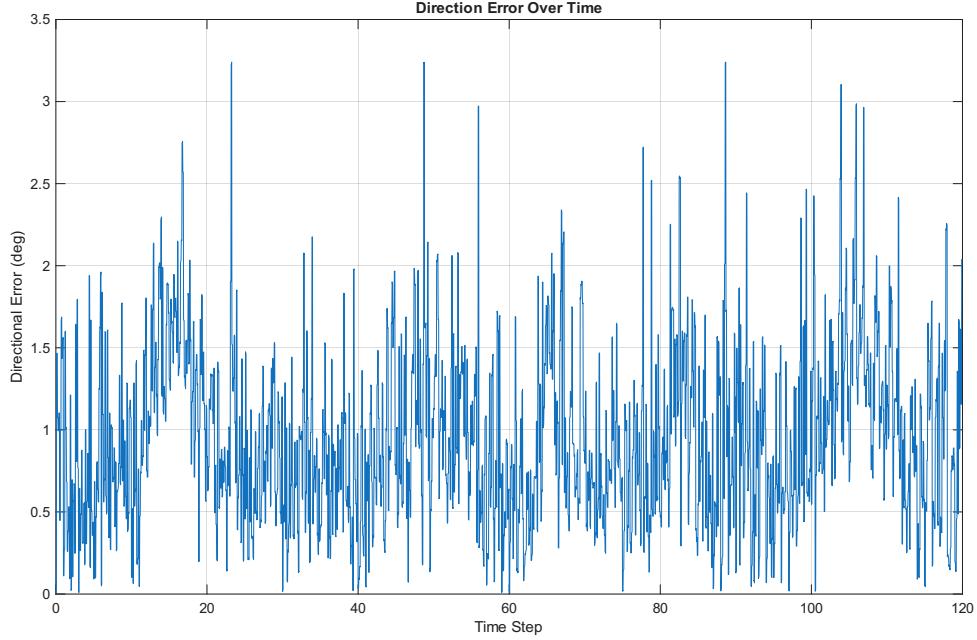


Figure 6.10: The Attitude Error of the ADCS Suite

Summary of results.

Experiment	Position Error	Attitude Error
ET	1526 m	0.208 °
ET	453 m	0.851 °
ADCS Suite	264 m	2.957 °
ET + ADCS Suite	273 m	1.187 °
ST + ADCS Suite	722 m	0.954 °

Table 6.5: Comparison of position and attitude errors for different experiments.

6.4 FEATURE TEST

In this section we are going to inspect how the amount of features influences the accuracy of the Earth Tracker. This idea can be extended to lens distortion. As a lens is more and more distorted it will become increasingly harder for the feature matcher to acquire correct features and thus less features will be used.

The Earth Tracker is set to:

Sensor	Sample Rate	Noise
ET	1 Hz	1500 m

Table 6.6: Sensor characteristics including sample rate, noise, and drift.

Detector	1		5		10	
	Pos Error	Att Error	Pos Error	Att Error	Pos Error	Att Error
SIFT	304 m	32.953°	601 m	0.607°	834 m	0.424°
SURF	247 m	33.429°	594 m	0.664°	808 m	0.458°
ORB	321 m	32.546°	543 m	0.760°	743 m	0.536°

Table 6.7: Performance of feature detectors for different numbers of features, showing position and attitude errors.

For the sake of completion lens distortion is also tested.

Table 6.8: Lens Distortion Parameters

Type	Parameter	Value
Radial	K_1	0.001
	K_2	0.001
	K_3	0.001
Tangential	P_1	0.0001
	P_2	0.0001
Chromatic	R	0.98
	G	1
	B	1.02

Multiple test did verify that the system is heavily dependent on accurate feature matching and proper lens correction.

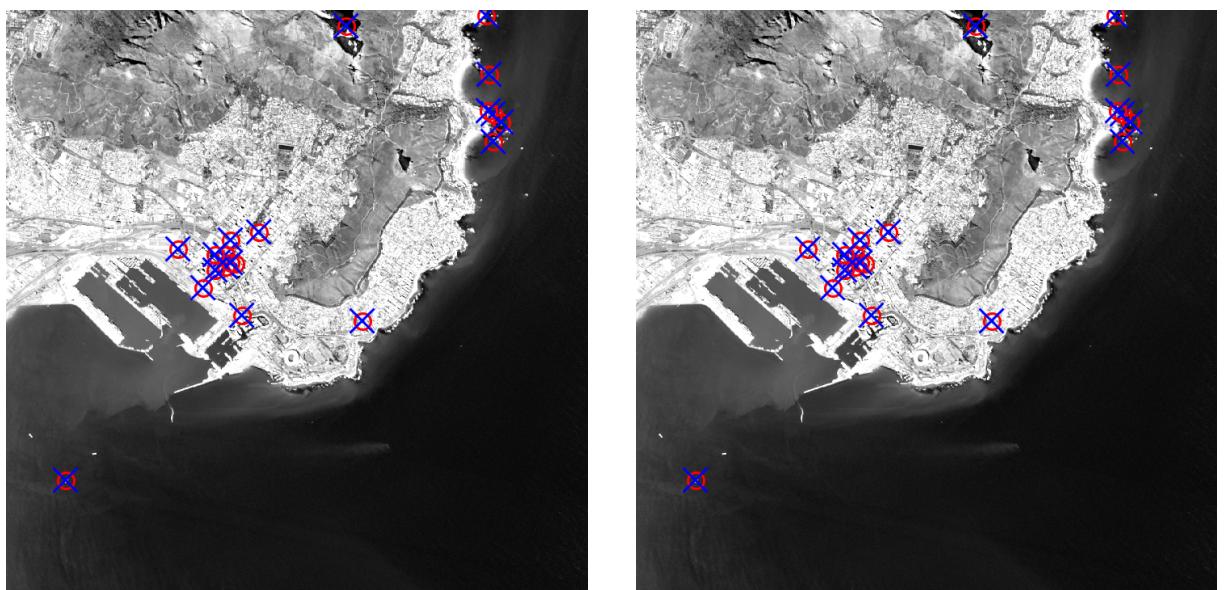


Figure 6.11: Earth Tracker Measurement with lens distortion

Even with normal lens distortion parameters the measurements from the Earth Tracker was useless.

6.5 TEMPORAL TEST

One of the very practical aspects of feature images is that it is properly time stamped, this test is to check how robust the system is against timestamp errors. For this the suite is used again, but now at different sampling periods

Experiment	With Time Delay		Without Time Delay	
	Position Error	Attitude Error	Position Error	Attitude Error
ET 1 Hz	279 m	1.210°	265 m	1.105°
ET 15 Hz	347 m	0.713°	m	°
ET 30 HZ	385 m	0.665°		

Table 6.9: Comparison of position and attitude errors for different experiments with and without time delay.

6.6 CONCLUSION

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

7.2 FUTURE WORK

For future work more development needs to be done for cloud matching and masking as my system really - Proper Feature matching In this work there is a huge gap left open in the feature matching scope. In this work feature matching was considered "perfect" using the raycasting method. But in reality there would be a feature matching component and an on-board database generation component also seen in the literature

- Dynamic features - Cloud masking - Robust Feature Detection - The ET system only works on circular orbits - System can be simplified to use phase estimation instead of full Cartesian estimation - Different Types of Biomes - More accurate models - Field Testing - Lens Correction - Computational Analysis

REFERENCES

- [1] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, “Towards the thousandth cubesat: A statistical overview,” 2019.
- [2] A. S. University, 2019, <https://www.nanosats.eu/sat/phoenix-cubesat>.
- [3] Y. Xing, X. Cao, S. Zhang, H. Guo, and F. Wang, “Relative position and attitude estimation for satellite formation with coupled translational and rotational dynamics,” *Acta Astronautica*, vol. 67, pp. 455–467, 8 2010.
- [4] E. Kahr, O. Montenbruck, and K. P. G. O’Keefe, “Estimation and Analysis of Two-Line Elements for Small Satellites,” *Journal of Spacecraft and Rockets*, vol. 50, no. 2, pp. 433–439, Mar. 2013, <https://arc.aiaa.org/doi/10.2514/1.A32352>.
- [5] C. Leibbrandt, C. Leibbrandt, and A. Scholtz, “Cubestar manual.”
- [6] C. Leibbrandt, C. Leibbrandt, C. Leibbrandt, and J. Miller, “Cubemag manual.”
- [7] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath, “Real-time geo-localization using satellite imagery and topography for unmanned aerial vehicles,” in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 2275–2281.
- [8] G. Klančar, S. Blažič, D. Matko, and G. Mušič, “Image-based attitude control of a remote sensing satellite,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 66, pp. 343–357, 2012.
- [9] L. Carozza and A. Bevilacqua, “Error analysis of satellite attitude determination using a vision-based approach,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 19–29, Sep. 2013, <https://linkinghub.elsevier.com/retrieve/pii/S0924271613001354>.
- [10] T. Kouyama, A. Kanemura, S. Kato, N. Imamoglu, T. Fukuhara, and R. Nakamura, “Satellite Attitude Determination and Map Projection Based on Robust Image Matching,” *Remote Sensing*, vol. 9, no. 1, p. 90, Jan. 2017, <https://www.mdpi.com/2072-4292/9/1/90>.

- [11] G. Klančar, D. Matko, G. Mušič, T. Rodič, and S. Blažič, “Remote attitude control using delayed visual information - nemo hd case study,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19. IFAC Secretariat, 2013, pp. 571–576.
- [12] P. Mateos-Ramirez, J. Gomez-Avila, C. Villaseñor, and N. Arana-Daniel, “Visual Odometry in GPS-Denied Zones for Fixed-Wing Unmanned Aerial Vehicle with Reduced Accumulative Error Based on Satellite Imagery,” *Applied Sciences*, vol. 14, no. 16, p. 7420, Aug. 2024, <https://www.mdpi.com/2076-3417/14/16/7420>.
- [13] J. Dietrich, 2016, <https://adv-geo-research.blogspot.com/2016/08/off-nadir-res-calc.html>.
- [14] W. G. S. Centre, 2025, <https://www.usgs.gov/media/images/aquatic-remote-sensing-examples-high-and-low-temporal-resolution>.
- [15] J. A. K. Steven A Ackerman, “Meatoerology: understanding the atmospher,” 2007.
- [16] A. Collison, “A clear look at the earth’s surface: Planet’s surface reflectance beta,” 2017, <https://www.planet.com/pulse/a-clear-look-at-the-earths-surface-planets-surface-reflectance-beta/>.
- [17] B. Mellish, “Chromatic aberration,” 2025, https://en.wikipedia.org/wiki/Chromatic_aberration.
- [18] W. C. D. Jongh, H. W. Jordaan, and C. E. V. Daalen, “Space debris: Pose estimation using stereo vision,” 2019, <https://scholar.sun.ac.za>.
- [19] J. Bouwmeester and J. Guo, “Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology,” *Acta Astronautica*, vol. 67, no. 7-8, pp. 854–862, Oct. 2010.
- [20] C. Toth and G. Józków, “Remote sensing platforms and sensors: A survey,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, May 2016.
- [21] J. D. Lesikar, “Attitude Accuracy Study for the Earth Observing System (EOS) AM-1 Spacecraft.”
- [22] Y. Zhang, L. Yang, Y. Cheng, and K. Ying, “A Performance Evaluation Approach for Satellite Attitude Control System in Tracking Mode,” *Applied Sciences*, vol. 14, no. 7, p. 2867, Mar. 2024, <https://www.mdpi.com/2076-3417/14/7/2867>.
- [23] B. Ibrahim, “Study and analyze the attitude determination and control subsystem (adcs) of a small remote sensing satellite,” *figshare*, 2020.
- [24] H. Jones, “The future impact of much lower launch cost,” 2018.
- [25] N. Drenthe, “Cost estimating of commercial smallsat launch vehicles,” 2018.

- [26] N. Adilov, “An analysis of launch cost reductions for lowe earth orbit satellites,” 2022.
- [27] G. Nagel, “Nanosatellite applied to optical earth observation a review,” 2020.
- [28] A. A. Mahmoud, T. T. Elazhary, and A. Zaki, “Remote sensing cubesat,” in *Sensors, Systems, and Next-Generation Satellites XIV*, vol. 7826. SPIE, 10 2010, p. 78262I.
- [29] B. Bomani, “Cubesat technology past and present: Current state of the art survey,” 2021.
- [30] K. Bouzoukis, “An overview of cubesat mission and application,” 2025.
- [31] D. Selva and D. Krejci, “A survey and assessment of the capabilities of cubesats for earth observation,” pp. 50–68, 5 2012.
- [32] M. H. B. Azami, N. C. Orger, V. H. Schulz, T. Oshiro, and M. Cho, “Earth observation mission of a 6u cubesat with a 5-meter resolution for wildfire image classification using convolution neural network approach,” *Remote Sensing*, vol. 14, 4 2022.
- [33] G. J. J. Korf, “Pose estimation of space objects,” 2024.
- [34] M. Sabatini, R. Volpe, and G. B. Palmerini, “Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing,” *Acta Astronautica*, vol. 171, pp. 323–334, 6 2020.
- [35] C. Meng, Z. Li, H. Sun, D. Yuan, X. Bai, and F. Zhou, “Satellite pose estimation via single perspective circle and line,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, pp. 3084–3095, 12 2018.
- [36] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märkens, and S. D’Amico, “Satellite pose estimation challenge: Dataset, competition design, and results,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, pp. 4083–4098, 10 2020.
- [37] S.-C. Han, J. H. Kwon, and C. Jekeli, “Accurate absolute gps positioning through satellite clock error estimation.”
- [38] F. R. Hoots, P. W. Schumacher, and R. A. Glover, “History of Analytical Orbit Modeling in the U. S. Space Surveillance System,” *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 174–185, Mar. 2004, <https://arc.aiaa.org/doi/10.2514/1.9161>.
- [39] S. T. Goh and K.-S. Low, “Real-time estimation of satellite’s two-line elements via positioning data,” in *2018 IEEE Aerospace Conference*. Big Sky, MT: IEEE, Mar. 2018, pp. 1–7, <https://ieeexplore.ieee.org/document/8396414/>.

- [40] J. S. Bennett, B. E. Vyhalek, H. Greenall, E. M. Bridge, F. Gotardo, S. Forstner, G. I. Harris, F. A. Miranda, and W. P. Bowen, “Precision magnetometers for aerospace applications: A review,” 8 2021.
- [41] M. Zahran and M. Aly, “A solar cell based coarse sun sensor for a small leo satellite . . . a solar cell based coarse sun sensor for a small leo satellite attitude determination.”
- [42] C. C. Liebe, “Star trackers for attitude determination,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 10, pp. 10–16, 6 1995.
- [43] J. C. Springmann, “Satellite Attitude Determination with Low-Cost Sensors.”
- [44] I. Parra, M. A. Sotelo, D. F. Llorca, C. Fernández, A. Llamazares, N. Hernández, and I. García, “Visual odometry and map fusion for gps navigation assistance.”
- [45] W. Wu, Z. Jin, and C. Wang, “Satellite position and attitude estimation using an infrared earth sensor,” *Infrared Physics and Technology*, vol. 141, 9 2024.
- [46] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, <https://link.springer.com/10.1023/B:VISI.0000029664.99615.94>.
- [47] D. Bamber, D. P. Palmer, and D. S. Mackin, “High Performance Attitude Determination through Analysis of Geometric Distortions within Earth Observational Satellite Imagery.”
- [48] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, “Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs,” *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 137–161, Mar. 2009, <http://link.springer.com/10.1007/s10846-008-9257-y>.
- [49] H. E. Soken, C. Hajiiev, and S.-i. Sakai, “Robust Kalman filtering for small satellite attitude estimation in the presence of measurement faults,” *European Journal of Control*, vol. 20, no. 2, pp. 64–72, Mar. 2014, <https://linkinghub.elsevier.com/retrieve/pii/S0947358013002094>.
- [50] A. Carmi and Y. Oshman, “Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations,” *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 232–241, Jan. 2009, <https://arc.aiaa.org/doi/10.2514/1.35878>.
- [51] W. C. de Jongh, “Space debris: Pose estimation using stereo vision,” 2019.
- [52] H. W. Jordaan, “Spinning solar sail: The deployment and control of a spinning solar sail satellite,” 2016.

- [53] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," 2006.
- [54] H. K. Pavel M. Trivailo, "Utilisation of the "dzhanibekov's effect" for possible future space missions," 2017.
- [55] Z. Haikove, "Understanding ona in satellite imagery: What is off nadir angle and what is it used for?" 2024, <https://www.euspaceimaging.com/blog/2024/01/31/what-is-ona-off-nadir-angle-in-satellite-imagery/>.
- [56] P. A. Q. Wang, "Spatio-temporal fusion for daily sentinel-2 images," 2018.
- [57] J. E. Fowler, "Comprehenssive pushbroom and whsikbroom sensing for hyperspectral remote-sensing imaging," 2014.
- [58] S. Nag, "Multispectral snapshot imagers onboard small satellite formations for multi-angular remote sensing," 2017.
- [59] W. Jiang, A. Ferrero, S. Salicone, and Q. Zhang, "A possible way to perform recursive bayesian estimate in the possibility domain," pp. 3218–3227, 2017.
- [60] A. Becker, "Kalman filter from the ground up," 2023.

APPENDIX A

APPENDIX TITLE GOES HERE
