
# 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 trough this journey.
* Pinkmatter for your financial support for taking the risk of investing in me and my future. I will be enternally 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 collogues 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.

Date: 1 September 2025

# 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 corrospondance 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 discripters 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 possiblity 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

# List of Figures

# List of Tables

# Nomenclature

## Variables and Functions

### Constants

| | |
|---|---|
| $\omega_e$ | Rotation speed of the Earth |
| $c$ | A constant. |

### Functions

| | |
|---|---|
| $f$ | A function. |

### Variables

| | |
|---|---|
| $\mathbf{x}$ | A variable. |

# Acronyms and Abbreviations

| | |
|---|---|
| **ADCS** | Attitude Determinination and Control System |
| **BRF** | Body Refrence Frame |
| **CRF** | Camera Reference Frame |
| **DCM** | Direction Cosine Matrix |
| **ECEF** | Earth Centred Earth Fixed |
| **ECI** | Earth Centred Inertial |
| **EKF** | Extended Kalman Filter |
| **GPS** | Global Positioning System |
| **LLA** | Lattitude Longitude and Altitude Reference Frame |
| **LVLH** | Local Vertical Local Horizon |
| **SLAM** | Simultanous Localisation and Mapping |

# Definitions

## A

**Attitude** The orientation of a satellite in space.

## P

**Pose** The combination of a satellite's position and attitude.

## S

**State Estimation** The ability to determine a state of a system using mathematical models.

**Student** is an entity needing a thesis to transcend the state of being a student.

# Variables and Functions

## Constants

| | |
|---|---|
| $\omega_e$ | Rotation speed of the Earth |
| $c$ | A constant. |

## Functions

| | |
|---|---|
| $f$ | A function. |

## Variables

| | |
|---|---|
| $\mathbf{x}$ | A variable. |

# Chapter 1

# Introduction

## 1.1 Problem Background

- Satellites are getting smaller - Because this leads to stallites having reduced costs and timelines - This is enables by the miniraturisation of electronics

- One of the big industires in satellites is remote sensing - Remote Sening is the application where satellites are used to monitor the Earth - One of the applications is to take images of the Earth

- This leads to the problem that high accuracy is needed to take images of the targets on the Earth's surface - COTS components which is mainly used on small satellites lack the accuracy needed - Magnetometers is to low of an accuracy - Star Trackers have the right accuracy, but is expensive

## 1.2 Proposed Solution

- Proposed solution is to develop an estimation algorithm that can estimate the full state of the satellite - The Full State of a Satellite is its postition in Space and its attitude or its orientation in space. - The satellite uses the imager itself to determine position and attitude. - This can lead to reduce costs as the satellite is using an instrument which is already onboard the satellite. - Utilising the components when it is idle - Observing the target directly

## 1.3 Document Outline

- Chapter 2: Wil investigate previous sensors that is being used to determine Propose - Previous techniques estimating the pose - Some light touching on feature detection as this is crucial to the pose estimation system

- Chapter 3: Wil introduce the modelling of the system - Rigid Body Kinematics - Position Kinematics - Attitude Kinematics - Kalman Filters - Extended Kalman Filters

- Chapter 4: Measurement Generation - Feature detection - PinHole Camera Model. - The Plant - The Plant Model - The Measurment Model

- Chapter 5: State estimation - The Extended Kalman Filter - Update Step - Prediction Step - Simulator

- Chapter 6 is results

- Chapter 7 is Conclusion - Future Work

# Chapter 2

# Literature

## 2.1 Introduction

# Chapter 3

# Modelling

## 3.1 Introduction

This project focuses on the pose estimation of a satellite using satellite images. This is essentially a localistion problem and requires a realistic description of the system. The aim of this chapter is to sufficiently define the problemand the proposed solution. Estimation algorithms is discussed and an estimator is chosen to solve the localisation problem. Further, attitude representations of a rigid body is introduced along with the dynamic and kinematic models used to describe a satellite in inertial space. Attention is given to quaternion attitude representations along with their propagation using angular rates.

## 3.2 Problem Defintion

A satellite orbiting Earth in the Earth-Centered Inertial (ECI) reference frame performs Earth observation missions, continuously capturing high-resolution imagery of the planet's surface for scientific, commercial, or operational purposes. To fulfill mission objectives effectively, the satellite must provide not only high-quality imagery but also precise geographic information about observed areas. This requires accurate knowledge of the satellite's six-degree-of-freedom pose (three-dimensional position and three-dimensional attitude) relative to the ECI frame at the moment each image is captured.

Traditional satellite pose determination relies on external systems such as Global Navigation Satellite Systems (GNSS) and ground-based tracking networks. However, this thesis investigates an autonomous approach where the satellite performs "visual navigation" by identifying known ground features in its imagery and using these observations to determine its orbital state. The satellite essentially performs "reverse GPS" - instead of receiving position signals from space, it observes recognizable landmarks on Earth's surface and computes its pose from these visual references.

The core technical challenge lies in the transformation from raw imagery to precise pose estimates. This involves several interdependent problems: **(1) Feature Detection** - identifying which pixels in the imagery correspond to cataloged landmarks among millions of pixel observations; **(2) Geometric Inversion** - solving the complex inverse problem of determining six-dimensional pose from two-dimensional image projections of three-dimensional landmarks with known geographic coordinates; and **(3) Uncertainty Management** - handling measurement noise, feature detection errors, and dynamic orbital motion in real-time. This thesis assumes the availability of a pre-established catalog of ground features with precisely known geographic coordinates in the ECI frame. The feature matching problem - associating detected image features with specific catalog entries - is considered solved through prior knowledge of the observed terrain and existing geographic databases.

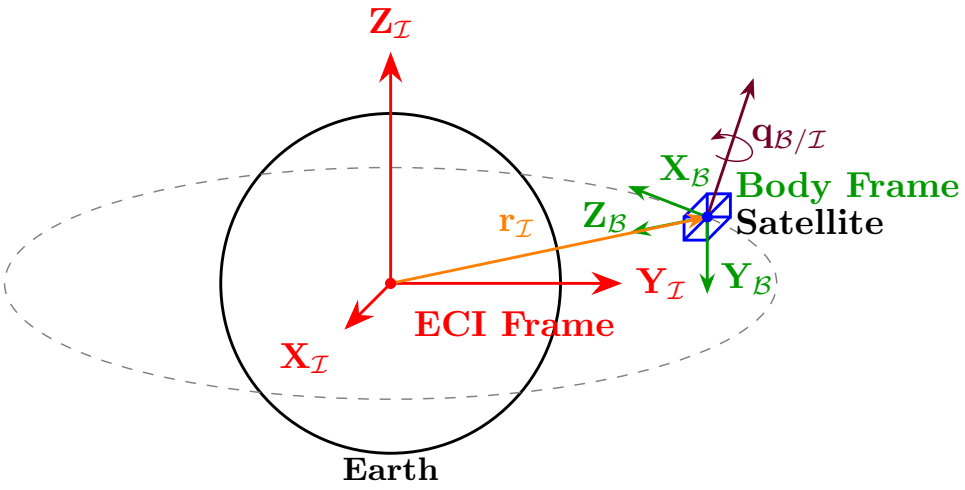


**Figure 3.1:** Satellite pose estimation concept showing orbital geometry, reference frames

This problem structure aligns with a simplified version of the Simultaneous Localization and Mapping (SLAM) framework. In this context, **localization** corresponds to determining the satellite's pose relative to the ECI frame using observations of cataloged features, while the **mapping** component is reduced to feature catalog utilization rather than creation. Since the geographic locations of observable features are assumed known a priori, the primary focus becomes the pose estimation problem given established feature correspondences.

The satellite's pose estimation system must account for the dynamic nature of orbital motion, the geometric relationship between the camera frame and satellite body frame, and the projection characteristics of the imaging system, while maintaining computational efficiency suitable for real-time onboard processing.

## 3.3 Refrence Frame Transsformations

In this masters we are going to encounter a few different refrence frames. To accurateley to create the measurement model we should have an understanding of all the different reference models and how to transform from one to another

### 3.3.1 Reference frame Conversions

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 $4 \times 4$ matrix.

**Homogeneous Transformation Matrix** A homogeneous transformation matrix from frame $\mathcal{I}$ to frame $\mathcal{O}$ is defined as:

$$\mathbf{A}_{\mathcal{I}}^{\mathcal{O}} = \begin{bmatrix} \mathbf{R}_{\mathcal{I}}^{\mathcal{O}} & -\mathbf{R}_{\mathcal{I}}^{\mathcal{O}} \cdot \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.1}$$

Here, $\mathbf{R}_{\mathcal{I}}^{\mathcal{O}}$ is a $3 \times 3$ direction cosine matrix (DCM) that describes the orientation of frame $\mathcal{O}$ relative to frame $\mathcal{I}$, and $\mathbf{t}$ is a translation vector from the origin of $\mathcal{I}$ to the origin of $\mathcal{O}$, expressed in frame $\mathcal{I}$.

This formulation ensures that both the direction and magnitude of vectors are preserved during the transformation. However, it is important to note that the inverse transformation is **not** obtained by simply switching the frame labels:

$$\mathbf{A}_{\mathcal{I}}^{\mathcal{O}} \neq \mathbf{A}_{\mathcal{O}}^{\mathcal{I}} \tag{3.2}$$

**Rotation-Only Transformations** In many practical scenarios, such as pure attitude transformations or body-to-inertial frame conversions, only rotational alignment is needed. In such cases, only the DCM $\mathbf{R}$ is used, and the transformation is defined as:

$$\mathbf{v}_{\mathcal{O}} = \mathbf{R}_{\mathcal{I}}^{\mathcal{O}} \cdot \mathbf{v}_{\mathcal{I}} \tag{3.3}$$

Note that:

$$\mathbf{R}_{\mathcal{I}}^{\mathcal{O}} = \left(\mathbf{R}_{\mathcal{O}}^{\mathcal{I}}\right)^{\top} \tag{3.4}$$

This relationship arises because direction cosine matrices are orthogonal.

**Special Case: Intrinsic Camera Matrix** In computer vision and photogrammetry applications, a special case of reference frame transformation is encountered during camera

projection. The intrinsic camera matrix $\mathbf{K}$ maps 3D points in the camera frame to 2D points on the image plane. While $\mathbf{K}$ is not a full homogeneous transformation matrix, it serves a similar purpose in transforming between reference representations, and can be interpreted as a scaled, non-rigid projection matrix with a fixed internal structure:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Here, $f_x$ and $f_y$ are the focal lengths in pixels, and $(c_x, c_y)$ represents the principal point on the image sensor. Which will be explained in further sections.

### 3.3.2 Lattitude, longitude and altitude

The lattitude, longitude of a feature or the position of the satellite is donated with the $\mathcal{L}$. The lattitude of a feature is the position of how high or low it above the equator, having a range of $-90°$ to $90°$. The longitude is based of the greenwich maridian, a longitude line that pases through the north- and south pole, it has a range of $-180°$ to $180°$. The altitude is measured form the the "WGS84" elliptical globe.

$$\mathbf{r}_{\mathcal{L}} = \begin{bmatrix} \lambda \\ \phi \\ h \end{bmatrix} \tag{3.6}$$

### 3.3.3 Earth Ceneterd Earth Fixed

The Earth Ceneterd Earth Fixed refrence frame is represented by the $\mathcal{F}$ and is very simular to the $\mathcal{L}$ reference frame with the z-axis alligned with the northpole and the x-axis points at the crossing of the Prime Maridian an the Equator, where the y-axis completes the right hand rule. The x,y and z-axis is defined in kilometers. To covert from $\mathcal{L}$ to $\mathcal{F}$ is to use a "WGS84" transfomr. Where WGS84 stands for World Geodetic System 1984, which is the standard coordinate system used for Global Positioning System (GPS). The WGS84 transformation uses a reference ellipsoid that uses a semi-major axis of 6,378 km and a flatting of 1/298.2 .

$$\mathbf{A}_{\mathcal{L}}^{\mathcal{F}} = f(\text{WGS84}) \tag{3.7}$$

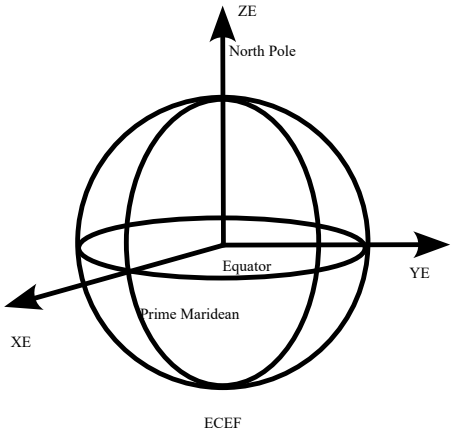


**Figure 3.2:** This figure represents the Earth Ceterned Earth Fixed reference frame with the X-axis defined in the direction of the equator and primemaridean with z-axis defined as the north pole and the y-axis completing the right had rule.

### 3.3.4 Earth Ceneterd Inertial

The Earth Centered Inertial refrence fream (ECI) refrenced by $\mathcal{I}$ shares a refrence frame axis with the ECEF, but is rotated about the z-axis. This rotation is governed by the rotation speed of the earth $\omega_e$ which is $7.2921 \times 10^{-5}$ rad/s and time $t$. To transform from the ECEF reference frame to the ECI reference frame one should rotate the Earth clockwise e.i.

$$\mathbf{A}_{\mathcal{F}}^{\mathcal{I}} = R(\omega_e t) = \begin{bmatrix} \cos(-\omega_e t) & -\sin(-\omega_e t) & 0 \\ \sin(-\omega_e t) & \cos(-\omega_e t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

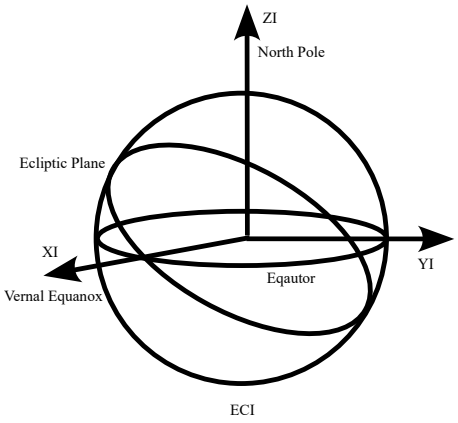


**Figure 3.3:** This figure represents the Earth Centered Inertial Reference frame with the x-axis defined in the direction of the vernel equanox, which is the defined as the crossing of the ecliptic plane and the eqautor, the z-axis is defined as the north pole and the y-axis completes the rght hand rule.

### 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 corrdinate system commonly used in spacecraft dynamics. It moves with the satellite and is defined relative to its orbit around Earth. The x-axis is the "Local Horizon" also called "along track" pointing forward it is tangent to the orbit and points in the direction of motion. The 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 y-axis is called the cross track it completes the right handed system. It points out of the orbital plane, typically the angular momentum vector direction (normal to the orbit plane).

if $\mathbf{r}$ is the position vector of the satellite and $\mathbf{v}$ is the velocity vector of the satellite. The equation for the reference frame is:

$$\bar{z}_{\mathcal{O}} = -\frac{\mathbf{r}}{||\mathbf{r}||} \tag{3.9}$$

$$\bar{y}_{\mathcal{O}} = \frac{\mathbf{r} \times \mathbf{v}}{||\mathbf{r} \times \mathbf{v}||} \tag{3.10}$$

$$\bar{x}_{\mathcal{O}} = \bar{y}_{\mathcal{O}} \times \bar{z}_{\mathcal{O}} \tag{3.11}$$

For this reference frame there should also be a refrence frame translation introduced. Which is done by substracing $\mathbf{r}$ from the vector

$$\mathbf{f}_{\mathcal{O}} = \mathbf{A}_{\mathcal{I}}^{\mathcal{O}} \times (\mathbf{f}_{\mathcal{I}} - \mathbf{r}_{\mathcal{I}}) \tag{3.12}$$

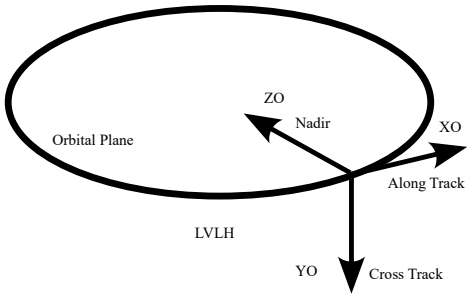


**Figure 3.4:** This figure represent the Local Vertical Local Horizon Referenece frame, more commonly refered to as the orbitial refrence frame. In the orbitial refrence frame the z-axis is defined as Nadir, meaning that it always points to the center of mass perfectly parallel to the oribital plane, the x-axis is defined as the vector tangentail to the plane, also known as the along track vector, and y is the completion of the right hand rule.

### 3.3.6 Body Referenece 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 z-axis defined as the yaw, x-axis defined as the roll and the y-axis defined as the pitch of the satellite. With the body frame z-axis and the orbital reference frame as alligned at initialisiation

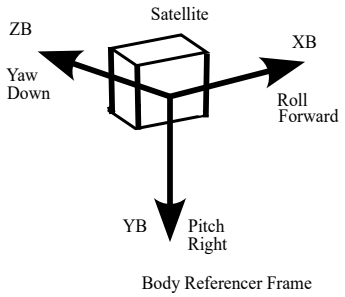


**Figure 3.5**

### 3.3.7 Camera Reference Frame

The camera refrence frame denoted by $\mathcal{C}$. In many Earth Observation missions the camera has its own reference frame defined relative to the Earth. Where the x-axis is pointing directly to the Earth's surface indicatting the cameras roll axis, with the y-axis representing the pitch axis represents an angle ahead or behind of the orbit and the z-axis representing the yaw axis.

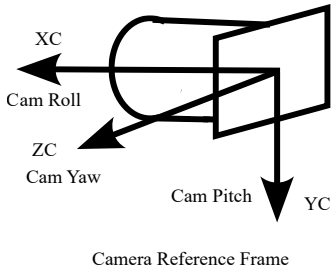


**Figure 3.6**

$$\mathbf{f}_{\mathcal{C}} = \mathbf{A}_{\mathcal{O}}^{\mathcal{C}} \times \mathbf{f}_{\mathcal{O}} \tag{3.13}$$

### 3.3.8 Image Reference Frame

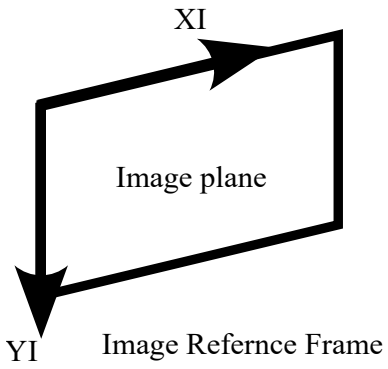


**Figure 3.7**

### 3.3.9 East - North - Up

## 3.4 Rigid Body Mechanics

### 3.4.1 Kinematics

The pose of a rigid body in a refrence frame consists of the position and attitude of the body. The attitude, or orientation of a body-fixed reference frame to a known reference frame. This is usualy represented by a rotation matrix, often referred to as a direction cosine Matrix (DCM). A rotation about a single coordinate axis is referred to as a coordinate rotation. A coordinate rotation about the x-,y- and z-axes with angles $\phi$, $\theta$ and $\psi$, of the body can be respectivley describes as, [Willem de Jong p.23]

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3.14}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \tag{3.15}$$

$$R_z(\psi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Any rotation in 3D space can be described by three coordinate rotations. The DCM describing the attitude of the target in the camera reference frame (CRF), $\mathbf{A}_{\mathcal{C}}^{\mathcal{B}}$, can be represented by three Eular angles. Each of the angles corresponds to one coordinate rotation. The order of the Eular 1-2-3 rotation, shown in Figure 3.5, is expressed as

$$\boldsymbol{A}_{\mathcal{C}}^{\mathcal{B}} = R_x(\phi)R_y(\theta)R_z(\psi) \tag{3.17}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \tag{3.18}$$

$$\begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ S\phi S\theta C\psi - C\phi S\psi & S\phi S\theta S\psi + C\phi C\psi & S\phi C\theta \\ C\phi S\theta C\psi + S\phi S\psi & C\phi S\theta S\psi - S\phi C\psi & C\phi C\theta \end{bmatrix} \tag{3.19}$$

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

$$\phi = \arctan 2 \left( \frac{a_{2,3}}{a_{3,3}} \right) \tag{3.20}$$

$$\theta = \arctan 2 \left( \frac{-a_{1,3}}{\sqrt{a_{1,1}^2} + \sqrt{a_{1,2}^2}} \right) \tag{3.21}$$

$$\psi = \arctan 2 \left( \frac{a_{1,2}}{a_{1,1}} \right) \tag{3.22}$$

mathematicl singularities occur when using Eular angles to represent large rotations. When both $a_{1,1}$ and $a_{1,2}$ in Equation 3.17 are zero, the expressions for $\psi$ and $\theta$ ar undefined. This is known as *gimbal lock*, where the changes in the first and third Eular angles are indistinguishable when the second angle nears a criticual value. Alternatively, the DCM can be described using quaternions, which do not have these singularities. The quaternion rotation is Figure ?? is expressed by the Eular axis $\bar{\mathbf{e}} = [e_x, e_y, e_z]^T$ and the angle $\theta$

$$\mathbf{q} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ e_x \sin(\theta/2) \\ e_y \sin(\theta/2) \\ e_z \sin(\theta/2) \end{bmatrix} \tag{3.23}$$

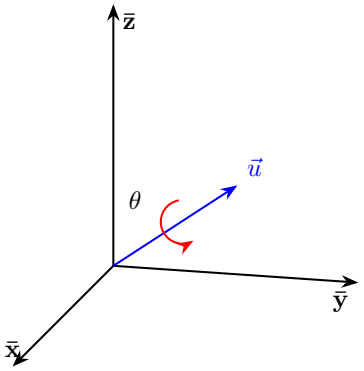


**Figure 3.8:** Quaternion Rotation

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

$$\mathbf{A}_{\mathcal{C}}^{\mathcal{B}} = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_s q_z) & 2(q_x q_z + q_s q_y) \\ 2(q_x q_y + q_s q_z) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_s q_x) \\ 2(q_x q_z - q_s q_y) & 2(q_y q_z + q_s q_x) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.24}$$

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

$$\mathbf{A}_{\mathcal{C}}^{\mathcal{B}} = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y-q_sq_z) & 2(q_xq_z+q_sq_y) \\ 2(q_xq_y+q_sq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z-q_sq_x) \\ 2(q_xq_z-q_sq_y) & 2(q_yq_z+q_sq_x) & 1-2(q_x^2+q_y^2) \end{bmatrix} \tag{3.25}$$

The body-fixed angular rates of the satellite in CRF, $\omega_{\mathcal{C}}^{\mathcal{B}}$, is expressed as a function of qauternions by,

$$\omega_{\mathcal{C}}^{\mathcal{B}} = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = 2 \begin{bmatrix} -q_x & q_s & -q_z & q_y \\ -q_3 & q_4 & q_1 & -q_2 \\ -q_4 & -q_3 & q_2 & q_s \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix} \tag{3.26}$$

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

$$\begin{bmatrix} \dot{q}_s \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{bx} & -\omega_{by} & -\omega_{bz} \\ \omega_{bx} & 0 & \omega_{bz} & -\omega_{by} \\ \omega_{by} & -\omega_{bz} & 0 & \omega_{bx} \\ \omega_{bz} & \omega_{by} & -\omega_{bx} & 0 \end{bmatrix} \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{3.27}$$

Quaternions will be used throughout this thesis for attitude representations. Quaternions do not have ambiguity regarding the order of rotations and the rotation is around a well-defined axis. The sin and cosine elements of the rotation matrix are already encodedin the quaternion form of the DCM. Therfore, only one matrix operation is required for attitude transforms, where Eular angles reguire three.

### 3.4.2 Dynamics

#### 3.4.2.1 Translational Dynamics

For the 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:

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{v}_t \Delta t + \frac{1}{2m}\mathbf{F}(t)\Delta t^2 \tag{3.28}$$

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \frac{1}{m}\mathbf{F}(t)\Delta t \tag{3.29}$$

where $\mathbf{F}(t)$ represents the net external force acting on the satellite. For the orbital environment considered in this work, where external perturbations are negligible compared to gravitational forces, and given that precise mass properties may not be available, the translational motion can be approximated using kinematic models where the current velocity depends primarily on the previous velocity state.

In orbital mechanics the $\mathbf{F}$ is proportional to the distance $\mathbf{r}$ of the center of mass where the gravitional force is governed by the following equation.

$$\ddot{\mathbf{r}} = \mathbf{F}\mathbf{u}_r \tag{3.30}$$

$$\ddot{\mathbf{r}} = \frac{-\mu}{||\mathbf{r}||^3}\mathbf{u}_r \tag{3.31}$$

The acceleration of the space craft can be influnced by different factors like $\mathbf{F}_G$ gravity, $\mathbf{F}_{J2}$ J2 perturbations, $\mathbf{F}_{drag}$ atmospheric drag, $\mathbf{F}_{sol}$ solar radiation presure and other perturbations $\mathbf{F}_{misc}$.

This wil mean the total force on the satellite would becomes

$$\mathbf{F}_{total} = \mathbf{F}_G + \mathbf{F}_{J2} + \mathbf{F}_{drag} + \mathbf{F}_{sol} + \mathbf{F}_{misc} \tag{3.32}$$

But as we are only going to model LEO (Low Earth Orbits) for a short period of time onlt the $\mathbf{F}_G$ and $\mathbf{F}_{J2}$ is considered.

Modelling the J2 pertubation is moddled in ECI using the following equation

$$\mathbf{a}_{J2} = \frac{3}{2} * J_2 * \frac{\mu R_E}{||\mathbf{r}||^5} \begin{bmatrix} r_x(1 - 5\frac{z^2}{r^2}) \\ r_y(1 - 5\frac{z^2}{r^2}) \\ r_z(3 - 5\frac{z^2}{r^2}) \end{bmatrix} \tag{3.33}$$

Where

* $\mu$ is Earth's gravitational parameter.
* $R_E$ is the radius of the Earth.
* $J_2$ is the J2 factor
* $\mathbf{r}$ is the satellite position in ECI

#### 3.4.2.2 Rotational Dynamics

The rotational dynamics of a rigid body satellite can be described using the Newton-Euler equations, which are applicable to all rigid inertial bodies . The angular momentum of the satellite is expressed as:

$$\dot{\mathbf{H}} = \frac{d\mathbf{H}}{dt} = \mathbf{I}\dot{\boldsymbol{\omega}} \tag{3.34}$$

where $\mathbf{H}$ represents the angular momentum vector and $\mathbf{I}$ is the diagonalized moment of inertia tensor about the satellite's principal axes. In the absence of external torques, the

rotational kinematics of a rigid satellite about its center of mass can be described by Euler's rotational equations:

$$I_{xx}\dot{\omega}_x = \omega_y\omega_z(I_{yy} - I_{zz}) \tag{3.35}$$

$$I_{yy}\dot{\omega}_y = \omega_x\omega_z(I_{zz} - I_{xx}) \tag{3.36}$$

$$I_{zz}\dot{\omega}_z = \omega_x\omega_y(I_{xx} - I_{yy}) \tag{3.37}$$

where $I_{xx}$, $I_{yy}$, and $I_{zz}$ are the principal moments of inertia, which remain constant and depend on the satellite's mass distribution and geometric configuration.

The stability characteristics of the satellite's rotational motion are governed by its mass distribution. According to Marsden and Ratiu , rotation about the major and minor principal axes is inherently stable, while rotation about the intermediate axis exhibits unstable behavior. Under constant energy conditions, any initial rotation about the intermediate axis will gradually redistribute energy to the major and minor axes through nutation effects.

To propagate the quaternion representing the satellite's attitude over time, the quaternion derivative must first be computed. The time derivative of the quaternion $\mathbf{q}_{B/I}$, which describes the rotation from the inertial frame to the body frame, is calculated using quaternion multiplication with the angular velocity vector:

$$\dot{\mathbf{q}}_{B/I} = \frac{1}{2}(\mathbf{q}_{B/I} \otimes \boldsymbol{\omega}) \tag{3.38}$$

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

$$\dot{\mathbf{q}}_{B/I} = \frac{1}{2}\begin{bmatrix} q_{B/I,0}\omega_x - q_{B/I,3}\omega_y + q_{B/I,2}\omega_z \\ q_{B/I,3}\omega_x + q_{B/I,0}\omega_y - q_{B/I,1}\omega_z \\ -q_{B/I,2}\omega_x + q_{B/I,1}\omega_y + q_{B/I,0}\omega_z \\ -q_{B/I,1}\omega_x - q_{B/I,2}\omega_y - q_{B/I,3}\omega_z \end{bmatrix} \tag{3.39}$$

where $q_{B/I,0}$, $q_{B/I,1}$, $q_{B/I,2}$, and $q_{B/I,3}$ are the scalar and vector components of the quaternion, respectively.

The quaternion integration is performed using a simple Euler integration scheme. First, the quaternion is propagated forward in time using:

$$\bar{\mathbf{q}}_{B/I}(t + \Delta t) = \mathbf{q}_{B/I}(t) + \dot{\mathbf{q}}_{B/I}\Delta t \tag{3.40}$$

where $\bar{\mathbf{q}}_{B/I}(t + \Delta t)$ represents the unnormalized quaternion after integration. Since quaternion integration may introduce numerical errors that violate the unit quaternion constraint, the result must be renormalized:

$$\mathbf{q}_{B/I}(t+\Delta t) = \frac{\bar{\mathbf{q}}_{B/I}(t+\Delta t)}{||\bar{\mathbf{q}}_{B/I}(t+\Delta t)||} \tag{3.41}$$

This normalization step ensures that the quaternion maintains its unit magnitude, preserving the validity of the attitude representation.

## 3.5 Sensor Modelling

### 3.5.1 GPS Measurement Model

In the simulation, GPS measurements are generated using the same underlying dynamics as the truth model, which is based on the two-body problem. This ensures consistency between the true satellite motion and the measurement framework. However, to emulate realistic sensor behavior, the GPS measurements are corrupted by both noise and drift.

The GPS measurement model is expressed as:

$$\mathbf{z}_{GPS}(t) = \mathbf{x}_{\text{true}}(t) + \boldsymbol{\eta}_{GPS}(t) + \mathbf{d}_{GPS}(t), \tag{3.42}$$

where:

* $\mathbf{z}_{GPS}(t)$ is the observed GPS measurement at time $t$,
* $\mathbf{x}_{\text{true}}(t)$ is the true state of the system as propagated by the two-body equations of motion,
* $\boldsymbol{\eta}_{GPS}(t)$ represents zero-mean Gaussian measurement noise, and
* $\mathbf{d}_{GPS}(t)$ is the GPS drift.

The drift component models the slow, unbounded accumulation of error that is characteristic of certain classes of low-cost GPS receivers. It is implemented as a random walk process:

$$\mathbf{d}_{GPS}(t) = \mathbf{d}_{GPS}(t-\Delta t) + \mathbf{q}_{GPS}(t), \tag{3.43}$$

where:

* $\mathbf{d}_{GPS}(t-\Delta t)$ is the drift at the previous timestep,
* $\mathbf{q}_{GPS}(t)$ is a zero-mean stochastic increment modeling the drift rate, typically drawn from a Gaussian distribution:

$$\mathbf{q}_{GPS}(t) \sim \mathcal{N}(0, \sigma_d^2 \mathbf{I}). \tag{3.44}$$

This formulation captures both short-term measurement variability through $\boldsymbol{\eta}_{GPS}(t)$ and long-term bias trends via $\mathbf{d}_{GPS}(t)$, allowing for more realistic testing and evaluation of estimation algorithms under degraded or imperfect sensing conditions.

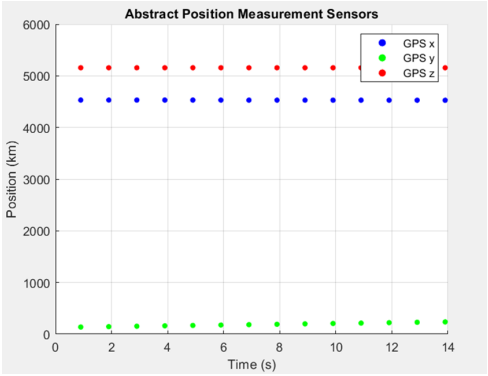


**Figure 3.9**

Insert Image of Modelled GPS Results

### 3.5.2 Gyroscope Measurement Model

The gyroscope provides a measurement of the angular velocity of the body frame relative to the inertial frame, expressed in the body frame. This quantity is denoted as $\boldsymbol{\omega}^B_{B/I}$, and forms a critical part of attitude determination and estimation systems.

To simulate a realistic sensor, the gyroscope measurement is corrupted by both random noise and a time-varying bias, or drift. The measurement model is given by:

$$\mathbf{z}_{\text{Gyro}}(t) = \boldsymbol{\omega}^B_{B/I}(t) + \boldsymbol{\eta}_{\text{Gyro}}(t) + \mathbf{d}_{\text{Gyro}}(t), \tag{3.45}$$

where:

- $\mathbf{z}_{\text{Gyro}}(t)$ is the observed gyroscope measurement at time $t$,
- $\boldsymbol{\omega}^B_{B/I}(t)$ is the true angular velocity of the body frame relative to the inertial frame,
- $\boldsymbol{\eta}_{\text{Gyro}}(t)$ is zero-mean Gaussian noise representing short-term measurement error, and
- $\mathbf{d}_{\text{Gyro}}(t)$ is the gyroscope drift, modeled as a time-varying bias.

The drift is modeled as a random walk process, capturing slow variations in the sensor bias over time:

$$\mathbf{d}_{\text{Gyro}}(t) = \mathbf{d}_{\text{Gyro}}(t - \Delta t) + \mathbf{q}_{\text{Gyro}}(t), \tag{3.46}$$

with the stochastic increment defined as:

$$\mathbf{q}_{\text{Gyro}}(t) \sim \mathcal{N}(0, \sigma_g^2 \mathbf{I}), \tag{3.47}$$

where $\sigma_g^2$ represents the drift rate variance of the gyroscope.

This model allows for the representation of both high-frequency noise and long-term integration drift, which are commonly observed in practical inertial measurement units (IMUs). Incorporating this model into the estimation framework enables more robust and accurate state reconstruction in the presence of sensor imperfections.

### 3.5.3 Star Tracker

The Star Tracker (ST) is a high-accuracy attitude sensor that provides an absolute measurement of the spacecraft's orientation, typically in quaternion form. Unlike relative sensors such as gyroscopes, the Star Tracker outputs an independent estimate of the spacecraft's attitude by capturing star field images and comparing them to a catalog.

In this simulation, the Star Tracker measurement is generated by perturbing the true attitude quaternion with a small random rotation. This models the sensor's measurement noise, which is assumed to follow a zero-mean Gaussian distribution.

**Noise Quaternion Generation** To simulate this noise, a small random rotation axis is sampled from a standard normal distribution and normalized. A noise quaternion $\mathbf{q}_{\text{noise}}$ is then constructed from this axis and a given angular noise magnitude $\theta$ (specified in degrees):

$$\mathbf{q}_{\text{noise}} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \hat{\mathbf{u}} \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{3.48}$$

where $\hat{\mathbf{u}}$ is a unit vector sampled from $\mathcal{N}(0,1)^3$ and $\theta = \deg 2rad(ST_{\text{noise}})$.

**Measurement Model** The measured Star Tracker quaternion is computed by applying the noise quaternion to the true attitude quaternion using the Hamilton product:

$$\mathbf{z}_{ST} = \mathbf{q}_{\text{noise}} \otimes \mathbf{q}_{\text{true}} \tag{3.49}$$

This represents a small perturbation of the true attitude, emulating realistic sensor output.

This model captures the key behavior of a real Star Tracker by applying small rotational noise to the true spacecraft attitude. It is suitable for use in truth-model simulations and Kalman filter evaluations, providing high-fidelity yet controllable measurement uncertainty.

Add Some Results

### 3.5.4 Coarse Sun Sensor

The Coarse Sun Sensor (CSS) is a fundamental attitude sensing instrument in nanosatellite systems, providing an estimate of the Sun direction relative to the satellites body frame. This subsection outlines the modeling approach for simulating CSS measurements, including the transformation of reference frames, sensor configuration, noise characteristics, and estimation logic.

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

$$\mathbf{S}_{\mathcal{I}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\top} \tag{3.50}$$

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

**Transformation to Body Frame** 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:

$$\mathbf{S}_{\mathcal{B}} = R_I^B \cdot \mathbf{S}_{\mathcal{I}} \tag{3.51}$$

where $R_I^B$ is the DCM from the inertial to body frame.

**Sensor Layout and Response** 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 3.12). Each sensor has a cosine response:

$$z_i = \max\left(0, \hat{\mathbf{n}}_i^{\top} \mathbf{S}_{\mathcal{B}}\right), \quad i = 1, \ldots, 6 \tag{3.52}$$

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

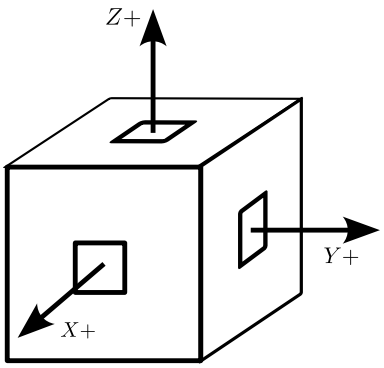


**Figure 3.10:** Orientation of the six coarse sun sensors on the CubeSat

**Measurement Noise** Each CSS reading is perturbed with zero-mean Gaussian noise. The noise standard deviation $\sigma$ is specified in degrees and converted to radians:

$$\mathbf{z}_{\text{CSS}} = \max\left(0, \mathbf{z}_{\text{CSS}} + \mathcal{N}(0, \sigma^2)\right), \quad \sigma = \deg 2rad(\text{CSS}_{\text{noise}}) \tag{3.53}$$

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

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

$$\hat{\mathbf{S}}_{\mathcal{B}} = \sum_{i=1}^{6} z_i \cdot \hat{\mathbf{n}}_i \tag{3.54}$$

The result is normalized to produce a unit vector:

$$\hat{\mathbf{S}}_{\mathcal{B}} = \frac{\hat{\mathbf{S}}_{\mathcal{B}}}{|\hat{\mathbf{S}}_{\mathcal{B}}|} \tag{3.55}$$

Add some results

### 3.5.5 Magnetometer

The magnetometer measurement is modeled as a unit vector pointing along the direction of the Earth's magnetic field as observed from the satellite body frame. To approximate this field, a simplified model is used in which the magnetic field points toward the geographic North Pole and is tangential to the Earth's surface at the satellit's location, with an optional dip angle to simulate inclination.

**Step 1: Earth-Fixed Transformation**

To determine the orientation of the magnetic field relative to the Earth-fixed frame, the satellite position vector $\mathbf{r}_{\mathcal{I}}$ in the inertial (ECI) frame is first transformed to the Earth-fixed (ECEF) frame using a rotation matrix that accounts for Earth's rotation angle at time $t$:

$$\mathbf{r}_{\mathcal{R}} = R_{\mathcal{I}}^{\mathcal{R}}(t) \cdot \mathbf{r}_{\mathcal{I}} \tag{3.56}$$

where $R_{\mathcal{I}}^{\mathcal{R}}(t)$ is a time-dependent rotation matrix based on the Earth rotation rate $\omega_e$.

**Step 2: Direction to Magnetic North**

The geographic North Pole is approximated by a fixed point on the Z-axis of the Earth-fixed frame:

$$\mathbf{p}_{NP,\mathcal{R}} = \begin{bmatrix} 0 \\ 0 \\ R_E \end{bmatrix} \tag{3.57}$$

The direction vector from the satellite to the North Pole is then computed as:

$$\mathbf{d}_{NP} = \mathbf{p}_{NP,\mathcal{R}} - \mathbf{r}_{\mathcal{R}} \tag{3.58}$$

**Step 3: Tangential Magnetic Field Model**

The local radial unit vector from the Earth's center is:

$$\mathbf{u}_{r,\mathcal{R}} = \frac{\mathbf{r}_{\mathcal{R}}}{|\mathbf{r}_{\mathcal{R}}|} \tag{3.59}$$

To simulate a magnetic field that is tangential to Earth's surface, the component of $\mathbf{d}_{NP}$ in the radial direction is removed:

$$\mathbf{z}_{\text{Mag},\mathcal{R}} = \mathbf{d}_{NP} - (\mathbf{d}_{NP} \cdot \mathbf{u}_{r,\mathcal{R}})\mathbf{u}_{r,\mathcal{R}} \tag{3.60}$$

This tangential field vector is then normalized:

$$\mathbf{z}_{\text{Mag},\mathcal{R}} = \frac{\mathbf{z}_{\text{Mag},\mathcal{R}}}{|\mathbf{z}_{\text{Mag},\mathcal{R}}|} \tag{3.61}$$

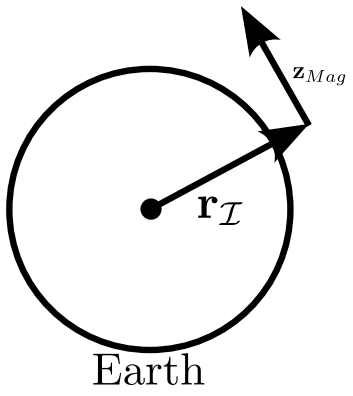


**Figure 3.11**

**Step 4: Dip Angle Adjustment**

To simulate magnetic inclination, a rotation about the local East direction is applied to tilt the magnetic field by a dip angle $\delta$, which is measured downward from the local horizontal plane:

$$\mathbf{z}^{\text{incl}}_{\text{Mag},\mathcal{R}} = \cos(\delta)\,\mathbf{z}_{\text{Mag},\mathcal{R}} + \sin(\delta)\,(\mathbf{e}_{\text{East}} \times \mathbf{z}_{\text{Mag},\mathcal{R}}) \tag{3.62}$$

Here, $\mathbf{e}_{\text{East}}$ is the local east direction, obtained via:

$$\mathbf{e}_{\text{East}} = \frac{\mathbf{u}_{r,\mathcal{R}} \times \mathbf{z}_{\text{Mag},\mathcal{R}}}{|\mathbf{u}_{r,\mathcal{R}} \times \mathbf{z}_{\text{Mag},\mathcal{R}}|} \tag{3.63}$$

**Step 5: Reference Frame Transformations**

The inclined magnetic field vector is transformed back to the inertial frame using:

$$\mathbf{z}_{\text{Mag},\mathcal{I}} = R^{\mathcal{I}}_{\mathcal{R}}(t) \cdot \mathbf{z}^{\text{incl}}_{\text{Mag},\mathcal{R}} \tag{3.64}$$

The final transformation to the spacecraft body frame is performed using the spacecraft attitude quaternion $\mathbf{q}^{\mathcal{I}}_{\mathcal{B}}$, resulting in:

$$\mathbf{z}_{\text{Mag},\mathcal{B}} = R(\mathbf{q}^{\mathcal{I}}_{\mathcal{B}}) \cdot \mathbf{z}_{\text{Mag},\mathcal{I}} \tag{3.65}$$

where $R(\mathbf{q})$ is the direction cosine matrix corresponding to the quaternion $\mathbf{q}$.

**Step 6: Measurement Noise**

To simulate realistic sensor output, a small-angle random rotation is applied to the vector $\mathbf{z}_{\text{Mag},\mathcal{B}}$ to represent magnetometer noise. This is modeled by sampling a noise rotation axis and applying a perturbation angle drawn from a Gaussian distribution with a standard deviation $\sigma_{\text{noise}}$ (in radians). The rotation is implemented via a noise quaternion and applied to the magnetic field vector.

**Step 7: Output**

The final estimated magnetometer vector $\hat{\mathbf{z}}_{\text{Mag},\mathcal{B}}$ is normalized and output for use in state estimation:

$$\hat{\mathbf{z}}_{\text{Mag},\mathcal{B}} = \frac{\mathbf{z}_{\text{Mag},\mathcal{B}}}{|\mathbf{z}_{\text{Mag},\mathcal{B}}|} \tag{3.66}$$

This provides a realistic simulation of a three-axis magnetometer, with both geographic dependence and sensor-level noise, for use in spacecraft attitude determination or estimation filters.

### 3.5.6 TRIAD Attitude Estimation

The TRIAD (Tri-Axial Attitude Determination) algorithm is a deterministic method used to estimate a spacecraft's attitude from two independent, non-collinear direction measurements expressed in both the body and inertial frames. In this work, the TRIAD method is applied using sun vector measurements from a coarse sun sensor and magnetic field measurements from a three-axis magnetometer.

**Step 1: Sensor Measurements in Body Frame**

Let $\mathbf{s}_{\mathcal{B}}$ denote the unit vector pointing toward the Sun in the body frame, and $\mathbf{m}_{\mathcal{B}}$ the unit magnetic field vector in the body frame. These vectors are obtained from sensor measurements and normalized:

$$\mathbf{s}_{\mathcal{B}} = \frac{\mathbf{s}_{\mathcal{B}}}{|\mathbf{s}_{\mathcal{B}}|}, \quad \mathbf{m}_{\mathcal{B}} = \frac{\mathbf{m}_{\mathcal{B}}}{|\mathbf{m}_{\mathcal{B}}|} \tag{3.67}$$

**Step 2: Reference Vectors in Inertial Frame**

The inertial-frame counterparts to the measured vectors are defined as follows:

- The Sun vector in the inertial (ECI) frame is approximated by the unit vector:

$$\mathbf{s}_{\mathcal{I}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.68}$$

- The magnetic field vector in the inertial frame, $\mathbf{m}_{\mathcal{I}}$, is computed based on the spacecraft position $\mathbf{r}_{\mathcal{I}}$, a user-defined magnetic dip angle, and Earth's rotation:

$$\mathbf{m}_{\mathcal{I}} = \text{MagnetometerModel}(\mathbf{r}_{\mathcal{I}}, t, \omega_e, \delta) \tag{3.69}$$

Both reference vectors are normalized:

$$\mathbf{s}_{\mathcal{I}} = \frac{\mathbf{s}_{\mathcal{I}}}{|\mathbf{s}_{\mathcal{I}}|}, \quad \mathbf{m}_{\mathcal{I}} = \frac{\mathbf{m}_{\mathcal{I}}}{|\mathbf{m}_{\mathcal{I}}|} \tag{3.70}$$

**Step 3: Constructing Orthogonal Triads**

Two orthonormal vector triads are constructed from the body and reference measurements.

- In the inertial frame:

$$\mathbf{v}_1^{\mathcal{I}} = \mathbf{s}_{\mathcal{I}} \tag{3.71}$$

$$\mathbf{v}_2^{\mathcal{I}} = \frac{\mathbf{s}_{\mathcal{I}} \times \mathbf{m}_{\mathcal{I}}}{|\mathbf{s}_{\mathcal{I}} \times \mathbf{m}_{\mathcal{I}}|} \tag{3.72}$$

$$\mathbf{v}_3^{\mathcal{I}} = \mathbf{v}_1^{\mathcal{I}} \times \mathbf{v}_2^{\mathcal{I}} \tag{3.73}$$

- In the body frame:

$$\mathbf{v}_1^{\mathcal{B}} = \mathbf{s}_{\mathcal{B}} \tag{3.74}$$

$$\mathbf{v}_2^{\mathcal{B}} = \frac{\mathbf{s}_{\mathcal{B}} \times \mathbf{m}_{\mathcal{B}}}{|\mathbf{s}_{\mathcal{B}} \times \mathbf{m}_{\mathcal{B}}|} \tag{3.75}$$

$$\mathbf{v}_3^{\mathcal{B}} = \mathbf{v}_1^{\mathcal{B}} \times \mathbf{v}_2^{\mathcal{B}} \tag{3.76}$$

These vectors form right-handed orthonormal bases (triads) in their respective frames.

**Step 4: Attitude Rotation Matrix**

The attitude rotation matrix $R_{\mathcal{I}}^{\mathcal{B}}$, which rotates vectors from the inertial frame to the body frame, is computed by aligning the inertial and body triads:

$$T_{\mathcal{I}} = \begin{bmatrix} \mathbf{v}_1^{\mathcal{I}} & \mathbf{v}_2^{\mathcal{I}} & \mathbf{v}_3^{\mathcal{I}} \end{bmatrix}, \quad T_{\mathcal{B}} = \begin{bmatrix} \mathbf{v}_1^{\mathcal{B}} & \mathbf{v}_2^{\mathcal{B}} & \mathbf{v}_3^{\mathcal{B}} \end{bmatrix} \tag{3.77}$$

$$R_{\mathcal{I}}^{\mathcal{B}} = T_{\mathcal{B}} \cdot T_{\mathcal{I}}^{\top} \tag{3.78}$$

**Step 5: Quaternion Conversion**

The attitude quaternion $\mathbf{q}_{\mathcal{B}}^{\mathcal{I}}$ corresponding to the rotation matrix $R_{\mathcal{I}}^{\mathcal{B}}$ is computed using a standard matrix-to-quaternion conversion:

$$\mathbf{q}_{\mathcal{B}}^{\mathcal{I}} = \text{rotm2quat}(R_{\mathcal{I}}^{\mathcal{B}}) \tag{3.79}$$

This quaternion is expressed in scalar-first format:

$$\mathbf{q}_{\mathcal{B}}^{\mathcal{I}} = \begin{bmatrix} q_s & q_x & q_y & q_z \end{bmatrix}^{\top} \tag{3.80}$$

The TRIAD method thus yields a closed-form attitude estimate without optimization or iteration. While it is highly efficient, its accuracy depends on the orthogonality and noise properties of the sensor measurements.

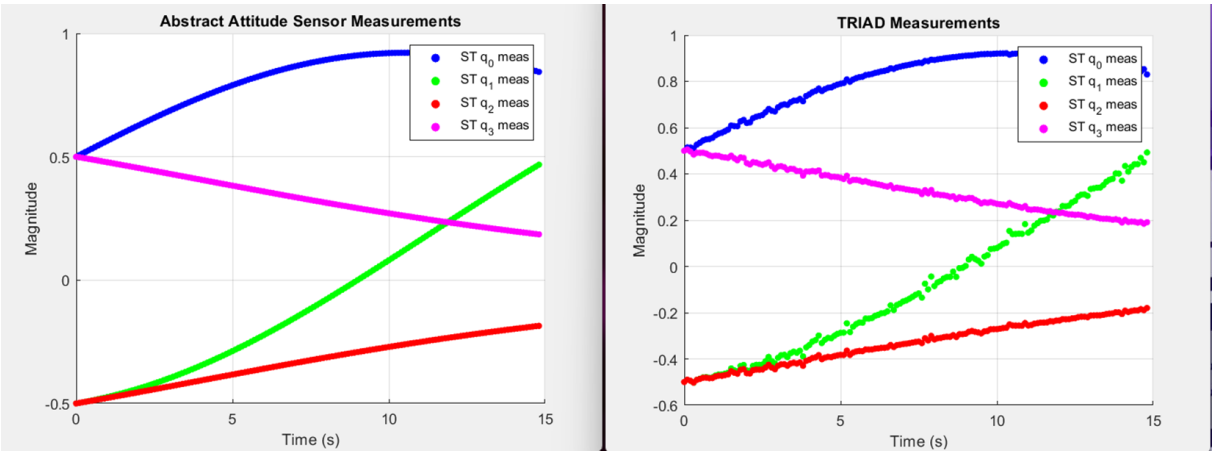


**Figure 3.12**

## 3.6 Conclusion

# Chapter 4

# Image Processing

## 4.1 Introduction

## 4.2 Pinhole Camera Model

The ideal pinhole camera can be described as a plane and an optical center (a.k.a) the pinhole. Light will travel from an object throught the optical center. And hit the plane at the opposite end of the optical center. The distance between the opitical center and the plane is called the focal length $f$.

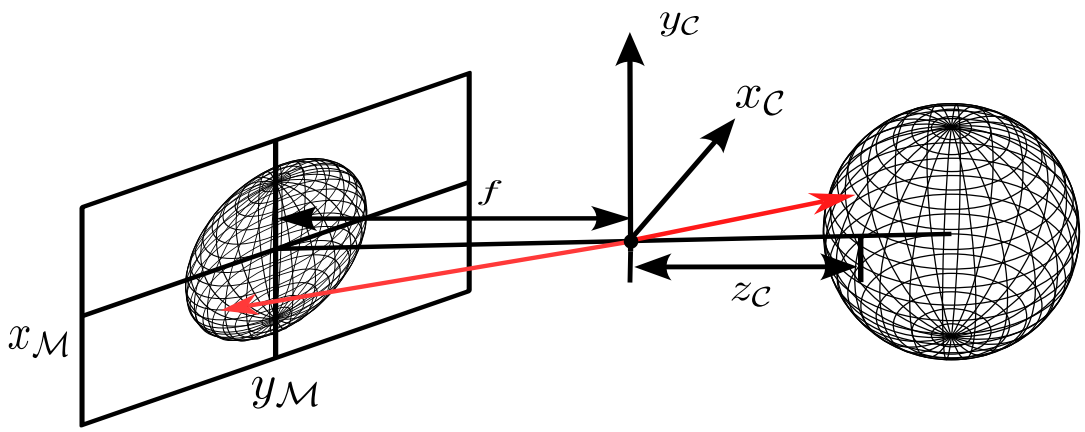


**Figure 4.1:** PinHole Model

The equation for the pinhole camera model is the following.

$$\begin{bmatrix} x_{\mathcal{M}} \\ y_{\mathcal{M}} \\ 1 \end{bmatrix} = \frac{-f}{z_{\mathcal{C}}} \begin{bmatrix} x_{\mathcal{C}} \\ y_{\mathcal{C}} \\ z_{\mathcal{C}} \end{bmatrix} \tag{4.1}$$

As we can see in the figure this also causes the image tp flip.

Also images are measured with the x-axis going from right to left and the y-axis going from top to bottom so one more transformation needs to be done.

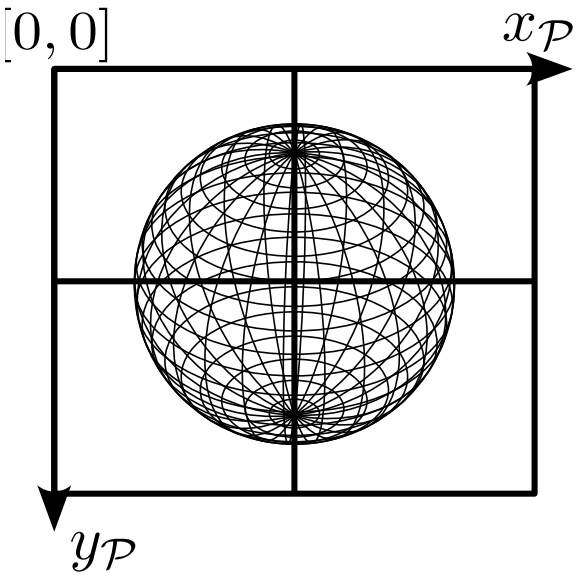


**Figure 4.2:** Image Plane

$$\begin{bmatrix} x_{\mathcal{P}} \\ y_{\mathcal{P}} \end{bmatrix} = \begin{bmatrix} -x_{\mathcal{M}} + \frac{\text{ImgWidth}}{2} \\ y_{\mathcal{M}} + \frac{\text{ImgHeight}}{2} \end{bmatrix} \tag{4.2}$$

### 4.2.1 Intrinsic Camera Parameters

The projection plane coordinates of the projected point P can be converted into pixel baed measurements by expanding the projection matrix.

A horizontal and verticle scale factor $s_u$ and $s_v$ are defined as

$$s_u = \frac{\text{horizontal image resolution}}{\text{horizontal sensor size}} \tag{4.3}$$

$$s_v = \frac{\text{vertical image resolution}}{\text{vertical sensor size}} \tag{4.4}$$

In the above equations, image resolution refers to the size, in pixels, of the resulting image captures by the modelled camera. Sensor size refers to the physical size of the sensor, this means the scaling factor can be seen as having a unit of pixels per distance.

These scaling factors can be incorporated with the focal length of the camera to create factors $f_u$ and $f_v$ so that

$$f_u = s_u f \tag{4.5}$$

$$f_v = s_v f \tag{4.6}$$

Pixel based coordinates of the projected point $\mathbf{p}$ can be calculated with

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & 0 \\ 0 & f_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{4.7}$$

Where the s is the scaling factor. The Projection matrix in Equation ... can lastly be expanded with the offsets $o_u$ amd $o_v$ 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

$$o_u = \frac{\text{horizontal image resolution}}{2} \tag{4.8}$$

$$o_v = \frac{\text{vertical image resolution}}{2} \tag{4.9}$$

Pixel-based coordinated off the projected point $\mathbf{p}$ can be calculated in the typical convention with

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & \alpha & o_u 0 & f_v & o_v 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{4.10}$$

$$= \mathbf{K} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{4.11}$$

where $\mathbf{K}$ is known as the intrinsic matrix a skewing factor of $\alpha$ is added to adjust for skewing affects of the camera.

### 4.2.2 Extrensic Camera Parameters

In the equations it is assumed that the point being projected onto the image plane reference is defined in the camera reference frame. This is not always the case and in certain circumstances the points that have to projected will have to be corrected the camera reference frame first.

An extrinsic camera matrix performs this conversion taks, it is made of a DCM $\mathbf{D}$ and translation vector $\mathbf{t}$ so that

$$[\mathbf{D}|\mathbf{t}] = \begin{bmatrix} d_{11} & d_{12} & d_{13} & t_1 \\ d_{21} & d_{22} & d_{23} & t_2 \\ d_{31} & d_{32} & d_{33} & t_3 \end{bmatrix} \tag{4.12}$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{D}|\mathbf{t}] \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \tag{4.13}$$

### 4.2.3 Back Projection

The intrinsic camera matrix discussed is typicall used to project 3D points in the camera referecne frame down to the 2D image plane reference frame. It can also be used to project 2D coordinates on the image plane reference frame back into 3D space.

Assume the scaling factor $s$ is known or assumed.

The 3D vector can be reconstructed.

$$\mathbf{d} = s * \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.14}$$

## 4.3 Satellite Image Characteristics

### 4.3.1 Ground Sample Distance

Ground Sampling Distance (GSD) is the real-world distance between the centers of two adjacent pixels measured on the ground in an image captured by a remote sensing system or satellite. It represents the spatial resolution of the imaging sensor and determines the level of detail visible in the image — smaller GSD values correspond to higher resolution, allowing finer features on the ground to be distinguished.

Where the mathematical relationship is

$$GSD = \frac{\text{altitude} * \text{pixelsize} * \text{resolution}}{\text{focal length}} \tag{4.15}$$

### 4.3.2 Imaging Geometry

The field of view (FOV) of the satellite imager is calculated using the relationship between the camera's focal length, sensor dimensions, and pixel size. The vertical field of view is

determined by the equation $\text{FOV}_v = 2 \times \arctan\left(\frac{I_y \times p_s}{2f}\right)$, where $I_y$ is the image height in pixels, $p_s$ is the pixel size, and $f$ is the focal length. For the horizontal field of view, the image width $I_x$ is substituted for $I_y$ in the calculation. This angular field of view defines the observable ground area from the satellite's orbital altitude, with the ground sample distance (GSD) providing the metric resolution per pixel according to $\text{GSD} = \frac{p_s \times h}{f}$, where $h$ is the altitude above the target surface. These geometric relationships are fundamental to the measurement model, as they establish the transformation between pixel coordinates in the image plane and the corresponding angular directions in the camera reference frame, enabling precise feature vector calculations for pose estimation.

### 4.3.3 Lens Distortions

## 4.4 Feature Detection and Description

### 4.4.1 Classical Feature Detectors

#### 4.4.1.1 SIFT

Scale-Invariant Feature Transform (SIFT) is a computer vision algorithm designed to detect and describe local features in images that remain stable under various transformations including scaling, rotation, and illumination changes. The algorithm operates in four main stages: scale-space extrema detection using Difference of Gaussians (DoG), keypoint localization through sub-pixel refinement, orientation assignment based on local gradient histograms, and descriptor generation using a 128-dimensional feature vector.

The scale-space representation is constructed by convolving the input image $I(x, y)$ with Gaussian kernels of increasing standard deviation:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.16}$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ is the Gaussian kernel. The DoG function approximates the Laplacian of Gaussian for efficient keypoint detection:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{4.17}$$

where $k$ is a constant multiplicative factor between adjacent scales.

For each detected keypoint, the dominant orientation is determined by computing the

gradient magnitude and direction:

$$m(x,y) = \sqrt{[L(x+1,y) - L(x-1,y)]^2 + [L(x,y+1) - L(x,y-1)]^2} \tag{4.18}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{4.19}$$

The descriptor is constructed by sampling gradients in a 16×16 pixel neighborhood around the keypoint, subdivided into 4×4 blocks with 8-bin orientation histograms, resulting in a 128-dimensional feature vector ($4 \times 4 \times 8 = 128$). Each descriptor element is weighted by the gradient magnitude and a Gaussian window centered at the keypoint. The resulting 128-dimensional descriptor provides robust matching capabilities across different viewing conditions, making SIFT particularly suitable for satellite imagery where features must be reliably detected despite changes in lighting, seasonal variations, and slight geometric distortions. However, the computational complexity of SIFT can be limiting for real-time applications, requiring careful consideration of the trade-off between feature quality and processing speed in resource-constrained satellite systems.

#### 4.4.1.2 SURF

Speeded-Up Robust Features (SURF) is a computer vision algorithm developed as a faster alternative to SIFT while maintaining comparable performance in feature detection and description. SURF achieves computational efficiency through the use of integral images and approximations of the Laplacian of Gaussian operator, making it particularly suitable for real-time applications in resource-constrained satellite systems.

The algorithm utilizes integral images $I_\Sigma(x,y)$ to enable rapid computation of rectangular area sums:

$$I_\Sigma(x,y) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i,j) \tag{4.20}$$

where $I(i,j)$ represents the intensity at pixel $(i,j)$. This allows any rectangular sum to be computed in constant time using only four array references.

SURF approximates the Laplacian of Gaussian using box filters that can be evaluated efficiently with integral images. The determinant of the Hessian matrix is used for keypoint detection:

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{4.21}$$

where $D_{xx}$, $D_{yy}$, and $D_{xy}$ are the convolution responses of the image with the second-order Gaussian derivatives, approximated using box filters. The factor 0.9 is an empirical weight to balance the expression.

For orientation assignment, SURF computes Haar wavelet responses in the $x$ and $y$

directions within a circular neighborhood:

$$d_x = \text{Haar}_x * I(x, y) \tag{4.22}$$

$$d_y = \text{Haar}_y * I(x, y) \tag{4.23}$$

The dominant orientation is determined by summing all responses within a sliding window of $\frac{\pi}{3}$ radians.

The SURF descriptor is typically 64-dimensional (compared to SIFT's 128), constructed by dividing a 20×20 pixel region around the keypoint into 4×4 sub-regions. For each sub-region, the wavelet responses are summed to create a 4-dimensional descriptor vector $\mathbf{v} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|]$. This results in a $4 \times 4 \times 4 = 64$-dimensional descriptor that provides robust matching while requiring significantly less computational resources than SIFT, making it advantageous for satellite pose estimation applications where processing efficiency is critical.

#### 4.4.1.3 ORB

Oriented FAST and Rotated BRIEF (ORB) is a binary feature descriptor that combines the FAST keypoint detector with the BRIEF descriptor, enhanced with orientation compensation to achieve rotation invariance. ORB is designed for real-time applications and provides significant computational advantages over SIFT and SURF, making it particularly suitable for resource-constrained satellite systems where processing efficiency is paramount.

The FAST (Features from Accelerated Segment Test) detector identifies keypoints by examining the intensity values of 16 pixels arranged in a circle around a candidate point $p$. A pixel $p$ is classified as a corner if there exists a set of $n$ contiguous pixels in the circle that are all brighter than $I_p + t$ or all darker than $I_p - t$, where $I_p$ is the intensity of pixel $p$ and $t$ is a threshold:

$$\text{FAST}(p) = \begin{cases} 1 & \text{if } \exists \text{ arc of length } n \text{ such that } \forall x \in \text{arc} : |I_x - I_p| > t \\ 0 & \text{otherwise} \end{cases} \tag{4.24}$$

To achieve rotation invariance, ORB computes the intensity centroid to determine keypoint orientation. The moments of a patch are calculated as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{4.25}$$

$$\text{Centroid: } \mathbf{C} = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{4.26}$$

The orientation angle is then determined by:

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \tag{4.27}$$

The BRIEF descriptor is modified to create rBRIEF (rotated BRIEF) by applying a rotation matrix to the sampling pattern. For a set of $n$ binary tests, each test $\tau$ compares the intensities of two pixels:

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases} \tag{4.28}$$

The rotated sampling pattern is computed as:

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S} \tag{4.29}$$

where $\mathbf{R}_\theta$ is the rotation matrix and $\mathbf{S}$ is the original sampling pattern.

The final ORB descriptor is a 256-bit binary string computed by applying the rotated BRIEF tests. This binary representation enables extremely fast matching using the Hamming distance, computed as:

$$d_H(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} a_i \oplus b_i \tag{4.30}$$

where $\oplus$ denotes the XOR operation. The computational efficiency and low memory requirements of ORB make it ideal for satellite pose estimation applications where real-time performance and limited computational resources are critical constraints.

#### 4.4.1.4 Comparison

## 4.5 Measurement Extraction

### 4.5.1 Image Generation

#### 4.5.1.1 Rendering the Earth

To create an image

Fisrt a High resolution image is sticth in QGIS, a type fo Geographic information system, used to open an edit geolocated images. Images a downloaded from the copernicus with a GSD of 15 m.

Insert Image of Paris Inert Image of Pyramids Insert Image of Hawaii Volcano

After an high resolution image is sitched. It is rasterised. This rasterised image with all its goelocation data is then projected onto a WGS84 ellipsoid.

**Figure 4.3:** High Resolution Image projected on Ellipsoid

The Low resolution Earth in the Backgorund is used as a place holder to test results

### 4.5.2 Earth Tracker Algorithm

The Earth Tracker works on the same principle as Back projection.

**Step 1: Translate pixels to optical center**

The second step involes translating the pixel cooridnates to the optical center as the origin. This transformation accounts for the camera's principle point offset, where $I_x$ and $I_y$ represent width and height, repsectivly. The resulting coordinate vector $\mathbf{f}_{\mathcal{M}/\mathcal{S}}$ represents position Relative to the camera boresight.

$$\mathbf{f_{M/S}} = \begin{bmatrix} f_{\mathcal{M}} - \frac{I_x}{2} \\ f_{\mathcal{M}} - \frac{I_y}{2} \end{bmatrix} \text{(Pixels)} \tag{4.31}$$

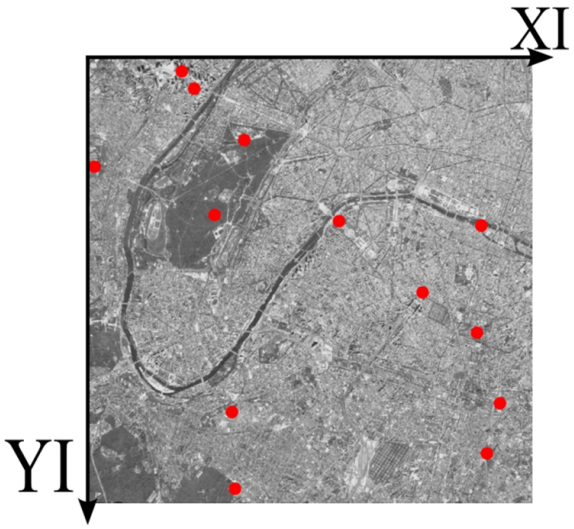


**Figure 4.4:** Origin Correction

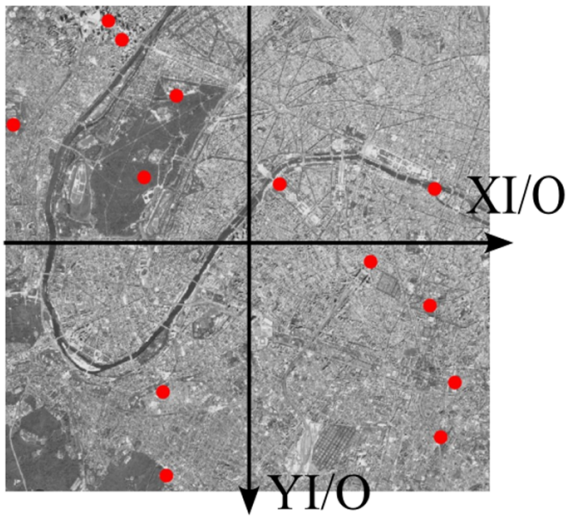


**Figure 4.5:** Origin Correction 2

**Step 2: Three-Dimentional Ray Vector Construction**

This step transform the 2 dimensional pixel coordinates into the three-dimensional direction vector in the camera frame. The vector $\mathbf{f}_{\mathcal{M}/\mathcal{F}}$ represents the feqture direction relative to the camera focal point, where the z-component is dtermined by the effective focal lenght in pixels

$$\mathbf{f}_{\mathcal{M}/\mathcal{F}} = \begin{bmatrix} f_{\mathcal{M}/\mathcal{S}_{\S}} \\ f_{\mathcal{M}/\mathcal{S}_{\dagger}} \\ \frac{fl}{ps} \end{bmatrix} \tag{4.32}$$

**Step 3: Scale dorection vector**

The 3rd step performs a cordinate transformation to orient the feature vector in the appropriate referecne direction. This operation inverts and scales the vector so that it is correct in the camera reference frame

The sacling used is the assumed GSD

$$\mathbf{f} = GSD \times \mathbf{f} \tag{4.33}$$

### 4.5.3 Geolocation Process

For the geolocation.

**Step 1: Chip Image**

**Step 2: Chip features**

**Step 3: Use SIFT for feauture matching**

**Step 4: Add geolocation**

# Chapter 5

# State Estimation

## 5.1 Introduction

## 5.2 Recursive Estimation

This section describes the element of recrusive estimators along with a few commonly-used estimators in the fields of localisation and tracking. A recursive estimoter uses the previous distrubution over a set of systm states and curent sensor data to estimte the current state distribution. Figure 3.1 illustrates the basic workings of a discrete recursive filter.

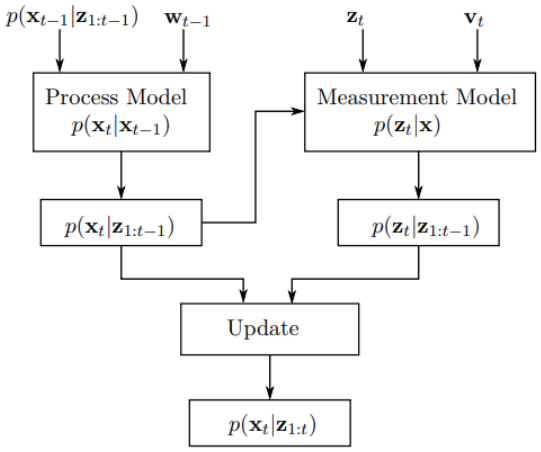


**Figure 5.1:** Recursive estimator algorithm flowchart

The state vector is represented by $\mathbf{x}_t$ for the estimation problem in the discrete time domain. The process, or the state transition function, is expressed As

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

where $\mathbf{f}$ is either a linear or non-linear transtion fucntion and $\mathbf{w}_t$ represents the process noise. New observation data, $\mathbf{z}_t$, is available at discrete timesteps and can be related to $\mathbf{x}_t$ by the measurement function,

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

The measurement uncertainty is represented by $\mathbf{v}_t$ and $\mathbf{h}$ is the observation model, which can also either be linear or non-linear. The goal is to obtain the posterior distribution, $p(\mathbf{x}|\mathbf{z}_{1:t})$, over the state vector $\mathbf{x}_t$. This is done by recursively performing the process and measurement updates.

At a time $t$, the posterior distribution over $\mathbf{x}_{t-1}$ at time $t-1$ is known and the prior distribution at t is calcualted As

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

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

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

The Kalman filter is a popular estimator used in pose estimation porblems. It is a special case of the Bayes filter, where the Guassian noise distributions are assumed. The assumption is also made that the initial distibution of the system can be represented by a Gaussian distribution. A control and measurement update is executed at each sampling instant to update the distibutionover the states. If the previous state distribution is Gaussian, then the updated current distribution will also be Gaussian and therefore the best estimate is chosen as the mean of the distribution.

Differen variants of the Kalman Filter exists, of which the extended and unscented Kalman Filter are the most popular, eah with their own unique characteristics.

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 position of the state vector is used as the linearisation point around which the tangent of the non-linear function is calcualted, allowing the use of standard Kalman Filter equations. It is typically more efficient than other non-linear filters which sometimes comes at a cost or reduced accuracy.

The unscented Kalman Filter (UKF) uses stochiatic 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 vlaues in a larger

region, rather than a local approximation.

## 5.3 Kalman Filter

Kalman filters are well suited for localistation problems since the nature of these systems are normally non-linear. Both the linear and non-linear variants of the Kalman Filter are concenerd with estimating states using motion model to perform this data fusion. The following equations are used to model the system.

$$\mathbf{x}_t = A_t\mathbf{x}_{t-1} + B\mathbf{u}_{t-1} + \epsilon\mathbf{z}_t = C_t\mathbf{x}_t + \zeta \tag{5.5}$$

where

$$\epsilon = \mathcal{N}(\mathbf{0}; R)\zeta = \mathcal{N}(\mathbf{0}; Q) \tag{5.6}$$

The matrices R and Q are the known covarianve matrices of the process and observation noise, repsectivly and the matrices A, B and C form part of the linear functions.

These equations can be used to calculate the posterior distrobution

$$p(\mathbf{y}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}) = \mathcal{N}(\mu_t; \Sigma_t) \tag{5.7}$$

The state estimation problem in this project makes use of non-linear system models, and a high-dimensinal state space. The EKF is well suited for this problem, since it accomodates non-linear process and observation models and is capable of dealing with high-dimensional state spaces. The EKF os often use for the SLAM problem and is well known in the field of robotics and localisation.

The rigid body motion models and measurements models of systems are, however, non-linear. Therefore, a general non-linear description is used for the motion and measurement models

$$\mathbf{y}_t = \mathbf{g}(\mathbf{y}_{t-1}, \mathbf{u}_t) + \boldsymbol{\epsilon} \tag{5.8}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{y}_t) + \boldsymbol{\zeta} \tag{5.9}$$

The motion and measurement functions, $\mathbf{g}$ and $\mathbf{h}$, are non-linear vector functions. These are linearised to enable the use of the Kalman filter equations. The non-linear vector functions, $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})]^T$, is linearised around its mean value, $\mu$, using a Taylor series expansion.

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\boldsymbol{\mu}) + \mathbf{f}'(\boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu}) \tag{5.10}$$

where

## 5.4 System Modelling

### 5.4.1 Motion Model

### 5.4.2 Measurement Model

## 5.5 Simulation

## 5.6 Practical Consideration

### 5.6.1 Number of Features

### 5.6.2 Outliers

## 5.7 Conclusion

# Chapter 6

# System Integration

## 6.1 System Diagram

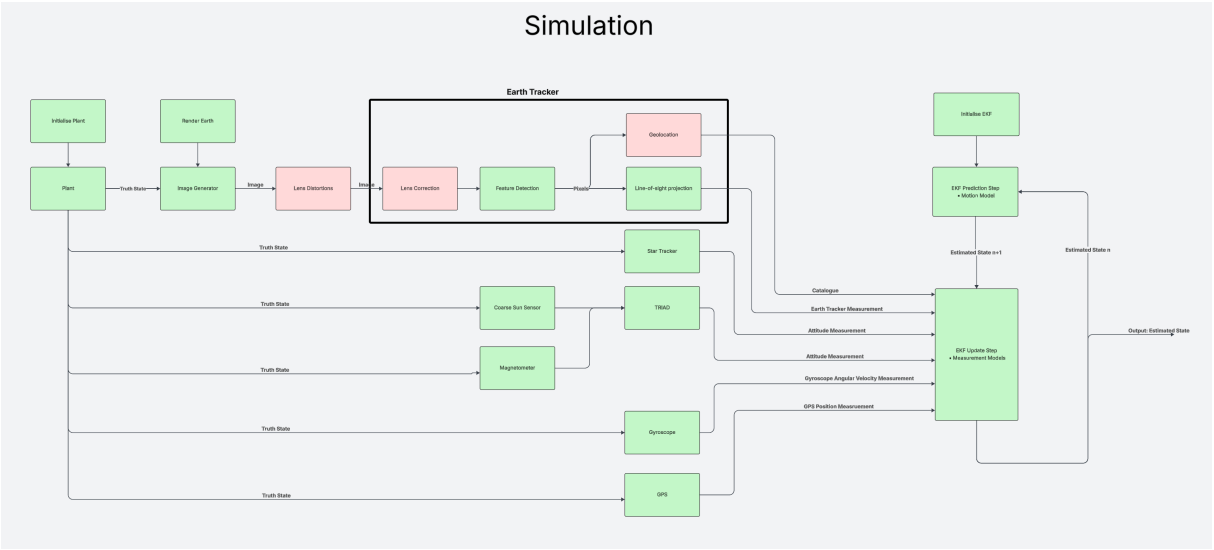


**Figure 6.1:** Image Plane

## 6.2 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 body with respect to the orbital reference frame.
* **Pitch (Y-axis)**: Initial pitch angle of the satellite body with respect to the orbital reference frame.
* **Yaw (Z-axis)**: Initial yaw angle of the satellite body with respect to the orbital reference frame.

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:

$$\mathbf{r}_{\mathcal{I}} = T_{\mathcal{R}}^{\mathcal{I}} \left( T_{\mathcal{L}}^{\mathcal{R}}(\mathbf{r}_{\mathcal{L}}, \omega_e, t) \right) \tag{6.1}$$

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:

$$\mathbf{u}_{\text{east}} = \frac{1}{||\cdot||} \begin{bmatrix} -\sin(\lambda) \\ \cos(\lambda) \\ 0 \end{bmatrix} \tag{6.2}$$

where $\lambda$ is the geodetic longitude. The vector is normalized to obtain a unit direction.

The magnitude of the orbital velocity is computed using the vis-viva equation:

$$||\mathbf{v}|| = \sqrt{\frac{\mu}{||\mathbf{r}||}} \tag{6.3}$$

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

The inertial velocity vector is then calculated as:

$$\mathbf{v}_{\mathcal{I}} = ||\mathbf{v}|| \cdot \mathbf{u}_{\text{east}} \tag{6.4}$$

Add some pictures for visualisation

The satellite's initial attitude is defined by two sequential quaternion rotations:

* $q_{\mathcal{O}/\mathcal{I}}$: the quaternion representing the rotation from the inertial (ECI) frame $\mathcal{I}$ to the orbital reference frame $\mathcal{O}$,

* $q_{\mathcal{B}/\mathcal{O}}$: the quaternion representing the rotation from the orbital frame $\mathcal{O}$ to the satellite body frame $\mathcal{B}$,

These quaternions are constructed using the initial orbital position (from latitude, longitude, and altitude) and the initial roll, pitch, and yaw of the satellite body with respect to the orbital frame.

The total attitude of the satellite body with respect to the inertial frame is given by the quaternion composition:

$$q_{\mathcal{B}/\mathcal{I}} = q_{\mathcal{B}/\mathcal{O}} \otimes q_{\mathcal{O}/\mathcal{I}} \tag{6.5}$$

where $\otimes$ denotes quaternion multiplication, performed right-to-left (i.e., the rotation $q_{\mathcal{O}/\mathcal{I}}$ is applied first, followed by $q_{\mathcal{B}/\mathcal{O}}$).

**Reference Frame Alignment at Initialization** At the simulation start time $t = 0$, it is assumed that the Earth-Centered, Earth-Fixed (ECEF) frame $\mathcal{R}$ is aligned with the inertial frame $\mathcal{I}$. This is a common simplification used in orbital mechanics when time is initialized at a known Greenwich Mean Sidereal Time (GMST), typically zero. This alignment implies:

$$R^{\mathcal{I}}_{\mathcal{R}}(t = 0) = \mathbf{I}_{3\times 3} \tag{6.6}$$

where $R^{\mathcal{I}}_{\mathcal{R}}$ is the rotation matrix from ECEF to ECI. This simplifies the initial attitude calculations and ensures consistent initialization across reference frames.

**Angular Velocity Initialization** The initial angular velocity of the satellite body is specified relative to the orbital frame:

$$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{O}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6.7}$$

Since the ECI and ECEF frames are aligned at $t = 0$, and the orbital frame is defined in the ECI frame, this implies that the angular velocity of the body with respect to the inertial frame is initially equivalent to the angular velocity with respect to the orbital frame:

$$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{I}}(t = 0) = \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{O}}(t = 0) \tag{6.8}$$

This relationship holds only at the initialization instant. As time progresses, the orbital frame rotates relative to the inertial frame due to the satellite's motion, and the

distinction between $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{I}}$ and $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{O}}$ becomes significant and must be handled accordingly in the attitude propagation.

# Chapter 7

# Experiments

## 7.1 Introduction

## 7.2 Conclusion

# Chapter 8

# Conclusion and Future Work

## 8.1 Conclusion

## 8.2 Future Work

# References

# Appendix A

# Appendix title goes here