



CHALMERS
UNIVERSITY OF TECHNOLOGY



Automated Satellite Tracking for Continuous Internet Access

Design and Implementation of a Dynamic Antenna Control System for Vehicular Internet Connectivity

Master's thesis in Systems, control and mechatronics

Armin Khorsandi, Ayman Alzein

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

MASTER'S THESIS 2025

Automated Satellite Tracking for Continuous Internet Access

Design and Implementation of a Dynamic Antenna Control System
for Vehicular Internet Connectivity

Armin Khorsandi
Ayman Alzein



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025

Automated Satellite Tracking for Continuous Internet Access
Design and Implementation of a Dynamic Antenna Control System for Vehicular
Internet Connectivity
Armin Khorsandi
Ayman Alzein

© Armin Khorsandi, 2025, arminkh@chalmers.se
© Ayman Alzein, 2025, aymana@chalmers.se

Industrial Supervisor: Badr Aldeen Alhaffar, Mechatronics SATCUBE
Academic Supervisor: Knut Åkesson, Chalmers University of Technology
Examiner: Knut Åkesson, Chalmers University of Technology

Master's Thesis 2025
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Control of two-axis gimbal system on moving car

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Automated Satellite Tracking for Continuous Internet Access
Design and Implementation of a Dynamic Antenna Control System for Vehicular
Internet Connectivity

Armin Khorsandi

Ayman Alzein

Department of Electrical Engineering

Chalmers University of Technology

Abstract

This thesis presents the design of a 2-DoF gimbal control system for vehicle-mounted antennas to maintain stable satellite alignment for uninterrupted Internet connectivity. Addressing challenges posed by vehicle movement, the work encompasses developing a simulation environment incorporating mathematical model of an antenna system, sensor fusion algorithms, and advanced control strategies, including complementary filter, Kalman filter, PID controller, and LQR. The results revealed that while the complementary filter responded more quickly to changes in system dynamics, it suffered from drift, whereas the Kalman filter provided more stable estimations at the cost of a delayed response due to its reliance on model dynamics. The study adopted a hybrid control strategy, employing an LQI controller on the outer gimbal angle and a PID controller on the inner gimbal angle. Future work may integrate signal strength sensors, and adopt quaternion-based orientation to overcome gimbal lock. Conducting a robustness analysis of the controllers can further enhance system reliability. This research provides valuable insights into filtering and control techniques, offering a robust framework for applications requiring precise orientation estimation and efficient control in dynamic systems.

Keywords: Satellite Tracking, Dynamical model, Complementary Filter, Kalman Filter, PID Controller, Linear Quadratic Regulator.

Acknowledgements

The authors would like to express their heartfelt gratitude to everyone who provided guidance and support throughout this thesis. We extend our sincere thanks to SATCUBE for giving us the unique opportunity to be the first students to conduct research on their innovative product. Special appreciation goes to our industrial supervisor, Badr Aldeen Alhaffar, for his invaluable guidance and support during our time at SATCUBE, and to our academic supervisor and examiner, Knut Åkesson, for his encouragement and expertise throughout the process. We are also deeply thankful to the academic and industrial experts at Chalmers University of Technology and SATCUBE, whose insights and shared experiences enriched this project with both practical and theoretical perspectives.

Armin Khorsandi, Ayman Alzein, Gothenburg, January 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

Acc	Accelerometer
ARE	Algebraic Riccati Equation
CE	Characteristic Equation
DoF	Degree of Freedom
ESC	Electronic speed controller
GNSS	Global Navigation Satellite System
Gyro	Gyroscope
LHP	Left Half plane
LQI	Linear Quadratic Integral
LQR	Linear Quadratic Regulator
\mathcal{L}	Laplace transformation
\mathcal{N}	Normal distribution
RHP	Right Half plane
rot	Rotation matrix
TJV	Target Joints' Values
WGN	White Gaussian Noise

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

ap	Action point
α	Alpha axis
β	Beta axis
e	Extended model
eff	Effective
est	Estimated value
F	Friction
$inner$	Inner gimbal
$outer$	Outer gimbal
ref	Reference
r	Reaction
s	Sample / Satellite
V	Vehicle
x, y, z	Indices for specifying the direction

Parameters

a, b	Friction coefficient
Δt	Time discretization step (time interval)
g	Gravitational acceleration
γ	Tuning coefficient for complementary filter
J	Inertia

K	Controller gain
K_p, K_d, K_i	PID tuning parameters
Q, R	Weighting matrices

Variables

A, B, C	State space matrices
α	Outer gimbal angle
az	Global azimuth angle
β	Inner gimbal angle
d	Offset vector between center of gravity and a rotation axis
e	Control error
el	Global elevation angle
G, F	Transfer functions
H	Angular momentum / Open loop transfer function / Measurement model matrix
I	Integral state / Indicator equation / Identity matrix
L	Closed loop transfer function
λ	Local polarization angle
m	Mass
O	Rotation matrix of the orientation angles
ω	Angular velocity
p	Poles
pol	Global polarization angle
r	Reference
s	Laplace Variable
S	ARE solution matrix
T	Time constant / Torque
t	Time
θ	Pitch angle
φ	Roll angle
ψ	Yaw angle
u	Input vector
x	Coordinates / State vector

y	Coordinates / Output vector
z	Coordinate



Contents

List of Acronyms	ix
Nomenclature	xi
List of Figures	xix
1 Introduction	1
1.1 Background	1
1.2 Problem Description	1
1.3 Purpose and Goals	2
1.4 Project Scope	2
1.5 Research Questions	2
1.6 Limitations	3
2 Theory	5
2.1 Dynamical modeling	6
2.1.1 Equations of Motion	7
2.1.1.1 Inner gimbal rotation	8
2.1.1.2 Outer gimbal rotation	9
2.1.1.3 Summary	10
2.1.2 Friction modeling	10
2.1.3 ESC Modeling	12
2.1.4 Sensor modeling	14
2.1.4.1 Encoders	14
2.1.4.2 Differential GNSS	14
2.1.4.3 Accelerometer	15
2.1.4.4 Gyroscope	17
2.2 Kinematics	19
2.2.1 Global Target	19
2.2.2 Target joint values	21
2.3 Processing step for sensors	22
2.3.1 Encoder Data Processing	22
2.3.2 Gyroscope Data Processing	22
2.3.3 Accelerometer Data Processing	23
2.4 Sensor Fusion	24
2.4.1 Complementary filter	24
2.4.1.1 Application in This Research	26

2.4.2	Kalman filter	26
2.4.2.1	Prediction Step	27
2.4.2.2	Update Step	27
2.4.2.3	Application in This Research	28
2.4.2.3.1	Kalman based on IMU and differential GNSS	29
2.4.2.3.2	Kalman Filter Based on IMU-Only	29
2.4.2.4	Tuning the Complementary and Kalman Filters	30
2.5	Control Loop	31
2.5.1	PID Controllers	34
2.5.1.1	P Controller	35
2.5.1.2	D Controller	35
2.5.1.3	PD Controller	36
2.5.1.4	PID Controller	37
2.5.2	LQR Controllers	39
2.5.2.1	LQR model	40
2.5.2.2	LQI model	40
2.5.2.3	Tuning LQR	41
3	Methodology	43
3.1	Literature Study	43
3.2	Simulation Environment	44
3.3	System Modeling Using Simscape	45
3.4	Trajectory Design for Validation	46
3.5	3D Visualization	47
4	Results	49
4.1	Modeling Results	49
4.1.1	The Linear Equations of Motion	49
4.1.2	The Simplified Gimbal Transfer Function	50
4.2	Sensor Model	52
4.3	Sensor Processing and State Estimation	53
4.3.1	Complementary Filter	54
4.3.2	Kalman Filter Based on IMU and Differential GNSS	55
4.3.3	Kalman Filter Based Only on IMU	56
4.4	Controller Design	58
5	Discussion	63
5.1	Sensors	63
5.1.1	Magnetometer	63
5.1.2	GNSS and translation error	64
5.1.2.1	Error in Global Target Elevation Angle	64
5.1.2.2	Error in Global Target Azimuth Angle	65
5.2	Answering Research Questions	65
6	Conclusion	67
7	Future Recommendations	69

7.0.1	Signal Strength	69
7.0.2	Euler Angles to Quaternions	70
7.0.3	Robustness Analysis	70
Bibliography		71

List of Figures

2.1	General block diagram of the system, showcasing the interactions between the satellite's fixed position inputs, vehicle orientation and position, kinematics (Global target and TJV), sensor modeling, filtering, control, and dynamics (G_{Gimbal}). The 3D block provides a real-time visualization of the system through Simscape Multibody. . .	6
2.2	Terminal's components and corresponding reference coordinate systems. Antenna's line of sight is along y_β	7
2.3	Different friction models: a) Coulomb's friction; b) coulomb's and viscous friction; c) Simplified model incorporating coulomb's, viscous, and static friction; d) A realistic representation of the friction force [1].	11
2.4	Comparison of the step response of the actual ESC and the first-order model. The ESC is tasked with delivering 90% of the maximum torque specified by the motor manufacturer.	13
2.5	Encoder model in simulation. The model takes the true angle (in degrees) and converts it to encoder counts. Here, <i>Max Counts</i> denotes the total number of pulses generated by the encoder per revolution, as specified in the device datasheet.	14
2.6	Differential GNSS module model in simulation. It takes the position and the heading action points at each time step and converts them to signals from the GNSS sensor. The position measurements are assumed to be ideal, whereas for the heading estimation, the <i>WGN</i> is characterized based on analysis of logged sensor data.	15
2.7	A top view of the terminal shown in Figure 2.2. When the car changes its heading rate, the antenna moves in the opposite direction to maintain line-of-sight with the satellite. The difference between these angular accelerations results in a net angular acceleration acting on the system, introducing an induced acceleration Acc_i in the x direction of the accelerometer's body frame.	16
2.8	The accelerometer model in simulation. As mentioned, external linear accelerations introduced by the car are not considered.	17
2.9	The gyroscope model in simulation. The term "Random walk" is a standard term for <i>WGN</i> in gyroscope contexts. Both the bias instability and random walk are modeled based on the logged data. .	18

2.10	The function Global Target calculates the target elevation, azimuth, and polarization angles in a global reference frame, based on the positions of the terminal and the satellite. Meanwhile, the function Target Joint Values computes the local joint angles β , λ and α required to adjust for the terminal's orientation, ensuring the antenna is correctly aligned with the satellite.	19
2.11	An illustration depicting how azimuth and elevation angles are calculated.	20
2.12	Encoder processing step to estimate the angle α as a block diagram.	22
2.13	Gyroscope processing step to estimate roll, pitch, and yaw angles as a block diagram. The estimated α is differentiated and subtracted from the gyroscope's z-axis measurements. The data is then rotated back along the z-axis by α to isolate the raw measurements, which are subsequently integrated to estimate the orientation angles.	23
2.14	Accelerometer processing step to estimate roll and pitch angles as a block diagram. The accelerometer data is processed by removing the effects of angles α and ψ_{est} and rotating the data back along the z-axis. The roll and pitch angles are then calculated using the equations above.	24
2.15	The simulation model representing the dynamics from requested torques to angles. The model includes two uncoupled systems, one for each axis.	31
2.16	A simplified version of the block diagram shown in Figure 2.15, focusing on the relationship between a single requested torque and its resulting angle. For simplicity, the loop transfer function depicted in blue is denoted as $G_{sub}(s)$, while the transfer function for the saturation is denoted as $F(s)$	32
2.17	Closed-loop control system for G_{Gimbal} using a P-controller with gain k_p	35
2.18	Closed-loop control system for G_{Gimbal} using a D-controller with gain k_d	36
2.19	Closed-loop control system for G_{Gimbal} using a PD-controller.	37
3.1	Overview of the simulation system implemented in Simulink. This diagram mirrors the conceptual block diagram shown in figure 2.1.	44
3.2	Trajectory created for validation of the feedback control system, featuring varying road conditions, including pitch, roll, speed bumps, roadside inclines, and gravel roads, with 90-degree turns of different radius.	47
3.3	The 3D simulation environment created using Simscape Multibody. The vehicle, equipped with the antenna system mounted on its roof is shown tracking the trajectory. The antenna remains aligned with the satellite and the line of site represented by the red line.	48

4.1	The graph on the right is a zoomed-in view of the left graph. A simulation was performed for both the nonlinear and linear models in the simulation environment. The models were subjected to torque inputs at $t = 1$ second, with $T_\alpha = 0.3$ N and $T_\beta = 0.05$ N. For the inner gimbal, the β angles showed complete overlap between the two models. However, the α angles exhibited a slight deviation, reaching approximately 10 degrees after 10 seconds of simulation, as highlighted in the zoomed-in graph.	50
4.2	Heading angle estimations provided by the differential GNSS module recorded over an extended period with the module stationary.	52
4.3	This figure presents sub-figures A to E, illustrating the sensor signals while driving along the validation trajectory and locking the outer gimbal to ensure that the accelerometer and gyroscope data represent the vehicle frame. Sub-figure A corresponds to the trajectory segment shown in Figure 3.2. Sub-figure B depicts the action points (true values) of the vehicle orientation in the simulation. Sub-figure C shows the estimated heading angle using the differential GNSS sensor model. Sub-figure D illustrates the acceleration signal generated by the accelerometer model. Sub-figure E represents the angular velocity signal generated by the gyroscope model.	53
4.4	Orientation estimation using a complementary filter with a gain of 0.9. The figure shows the estimation After 10 laps around the validation trajectory. At the end of the trajectory, the orientation action points are $[roll, pitch, yaw] = [0, 0, 360]$ deg, while the estimation is $[1.40, 1.44, 362.01]$ deg indicating drift.	54
4.5	Orientation estimation using a Kalman filter based on IMU and differential GNSS.	55
4.6	A comparison of the estimation performance between the complementary filter and the Kalman filter. The figure shows the estimation results after running 10 laps of the validation trajectory at segment 6, which simulates a roll change on a strait road. When the roll action point's rate changes, the complementary filter responds immediately, while the Kalman filter exhibits a slight delay before adjusting to the action point. However, the complementary filter shows a noticeable drift (offset) from the action point.	56
4.7	Orientation estimation using a Kalman filter based only on IMU.	56
4.8	A PD controller is implemented using the parameter provided in equation 4.3, and the control error and the control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.	59

4.9	A PID controller is implemented using the parameters in equation 4.4, and the control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.	60
4.10	An LQR controller is implemented using the weighting matrices in equation 4.5 and the state-space model in equation 2.74. The control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.	61
4.11	An Extended LQR controller is implemented using the weighting matrices in equation 4.8 and the state-space model in equation 2.76. The control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.	62
5.1	The elevation error introduced by the vehicle's height above sea level, the height of the vehicle where the terminal is mounted and the offset between the inner and outer gimbals.	64
5.2	Top view: GNSS estimates the vehicle to be at P_{est} , while it could be anywhere inside the green circle, representing the uncertainty of the circle.	65

1

Introduction

This chapter introduces the topic of satellite Internet, with a specific focus on satellite-based Internet on the move, and explores its significance as a vital modern solution for various organizations. It provides a concise overview of the primary problem addressed in this research. Additionally, the goals and limitations of the project are outlined to provide context for the study.

1.1 Background

Internet connectivity has become a fundamental part of modern life, integrated into nearly every aspect of daily activities and professional tasks. Its availability is crucial in various locations and circumstances. For example, a wealthy individual on a yacht in the middle of the ocean relies on a stable Internet connection for communication and entertainment. A journalist reporting from a conflict zone depends on internet access to provide timely updates. Health organizations conducting missions in remote or disaster-stricken areas need reliable connectivity to coordinate with headquarters. Rescue teams require Internet access to carry out lifesaving operations effectively. In all of these scenarios, satellite-based Internet emerges as a viable solution, offering uninterrupted connectivity even in the most challenging environments.

1.2 Problem Description

Many satellite operators position their satellites in geostationary orbit, including those considered in this thesis. These satellites are located around the equator and remain fixed relative to the Earth. To establish a stable connection with such satellites, ground terminals equipped with directional antennas are required.

This arrangement poses a significant challenge, particularly in maintaining a clear line of sight between the satellite and the ground terminal's antenna. For stationary ground terminals, proper installation and calibration ensure reliable connectivity. However, when these terminals are mounted on moving vehicles, maintaining a stable connection becomes significantly more complex. The antennas must automatically adjust in real-time to account for vehicle movement, continuously aligning to maintain an uninterrupted link with the satellite.

This research is conducted in collaboration with SATCUBE, a leading company specializing in satellite-based Internet solutions for seamless connectivity in remote and demanding locations to address these challenges by focusing on the design and implementation of a simulation environment and a control system for vehicle-mounted satellite terminals. Specifically, the satellite terminal, equipped with an antenna with two degrees of freedom, is mounted on a moving vehicle. The antenna will be aligned with a known geostationary satellite to maintain continuous Internet connectivity.

1.3 Purpose and Goals

The purpose of this thesis is to design and develop an automated satellite tracking system capable of controlling the antenna's movement to maintain precise, active, and stable alignment with a satellite.

1.4 Project Scope

Defining a clear and focused project scope is essential to achieving the purpose of this research. The key aspects of the project scope are outlined below:

- Developing a mathematical model of the antenna system to accurately represent its dynamics and behavior.
- Implementing advanced sensor fusion and filtering techniques to process sensor data effectively within the simulation environment, ensuring accurate state estimation.
- Designing and implementing a control system for the antenna to ensure alignment with the known satellite.
- Conducting rigorous validation of the control system through a series of experiments in the simulation environment, replicating real-world scenarios.

This defined scope serves as the foundation for the project, guiding the development process and ensuring that all efforts remain focused on achieving the primary objective.

1.5 Research Questions

This research aims to address the following key questions:

- Which filtering technique offers the highest precision and reliability for processing data in this specific context?
- What methods can be used to evaluate and compare different control strategies, and which control algorithm demonstrates superior effectiveness when implemented in MATLAB/Simulink?

1.6 Limitations

This research is conducted within a constrained timeframe, which leads to several exclusions and limitations:

- **Hardware Implementation:** Due to time constraints, the system will not be implemented on a physical prototype and is only going to be developed and evaluated in a simulation environment.
- **Limited Algorithms and Methods:** Only a limited range of sensors, filtering techniques, and controllers will be explored and implemented, reflecting the time limitations of the study.

Furthermore, the sensitive nature of proprietary information related to the company's product restricts access to certain data and resources, which further shapes the scope of this research.

2

Theory

This chapter establishes the theoretical foundation for the research, encompassing a comprehensive analysis of system dynamics, sensor modeling, data processing, and control mechanisms. It begins with the fundamental equations of motion that govern the physical interactions within the system and explores the complexities of friction and torque dynamics. The kinematics section then provides an understanding of the spatial relationships and orientation adjustments necessary for system alignment.

Subsequent sections delve into the detailed modeling and processing of sensor data, including encoders, gyroscopes, accelerometers, and Global Navigation Satellite System (GNSS). The chapter also introduces sensor fusion techniques, comparing the effectiveness of complementary and Kalman filters in integrating multi-sensor data for optimal state estimation. Finally, the control loop design is presented, with an emphasis on feedback controllers such as PID and LQR, highlighting their roles in achieving system stability and responsiveness. This theoretical groundwork serves as the basis for the simulations and experimental validations discussed in later chapters.

Figure 2.1 shows the overall system architecture modeled in Simulink. It highlights the interactions between key subsystems, including the satellite's fixed position inputs (X , Y , Z) and the vehicle's time-varying position and orientation inputs (X , Y , Z) along with roll, pitch, and yaw. The "Global target" and "TJV" blocks are discussed in Section 2.2. The "Sensors" and "Estimator" blocks are covered in Sections 2.1.4 and 2.3. The "Controller" block is presented in Section 2.5, and the " G_{Gimbal} " block, which incorporates friction, the ESC, and the antenna's dynamics, is detailed in Section 2.1.

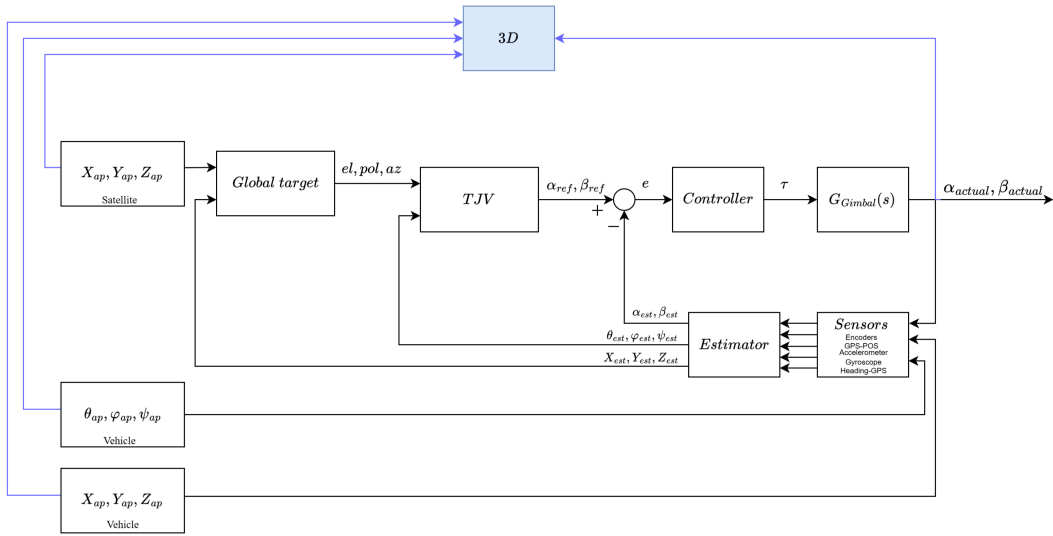


Figure 2.1: General block diagram of the system, showcasing the interactions between the satellite’s fixed position inputs, vehicle orientation and position, kinematics (Global target and TJV), sensor modeling, filtering, control, and dynamics (G_{Gimbal}). The 3D block provides a real-time visualization of the system through Simscape Multibody.

2.1 Dynamical modeling

The mechanical system comprises two main components: the base and the antenna, which consists of an outer and an inner gimbal. The global coordinate frame $[x_g, y_g, z_g]$ serves as the reference for all angular measurements. Since the terminal is mounted on a vehicle, the orientation and position of the base are aligned with the vehicle. The outer gimbal has its own local coordinate frame $[x_\alpha, y_\alpha, z_\alpha]$, while the inner gimbal’s frame $[x_\beta, y_\beta, z_\beta]$ is shifted from the outer gimbal along the z_α axis.

To align the antenna with the satellite, precise calculations are needed to determine the angles α and β . A detailed explanation of these calculations is provided in section 2.2.

The terminal is equipped with two motors: the beta motor, which controls the rotation of the inner gimbal around its x_β axis, corresponding to angle β , and the alpha motor, which rotates the outer gimbal around its z_α axis, corresponding to angle α . A simplified representation of the terminal is illustrated in Figure 2.2.

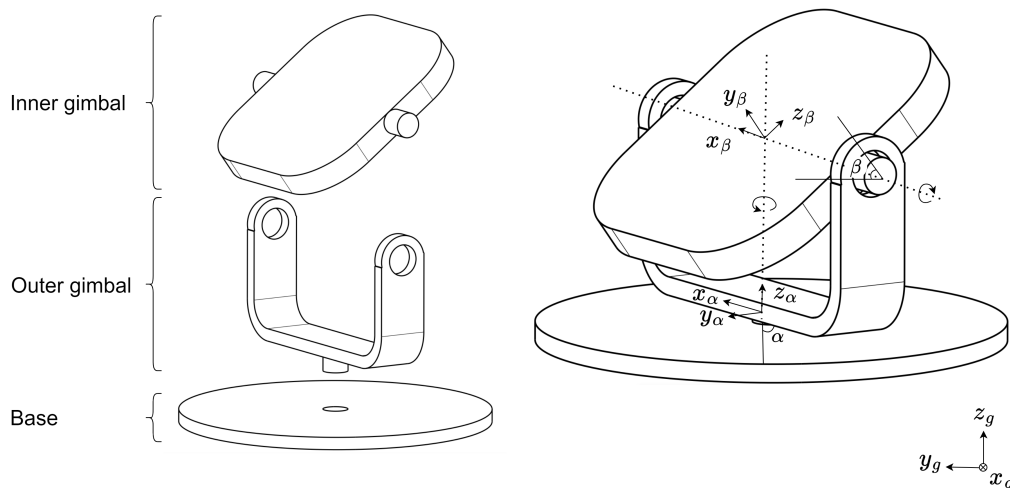


Figure 2.2: Terminal's components and corresponding reference coordinate systems. Antenna's line of sight is along y_β

2.1.1 Equations of Motion

When torque is applied to a rigid body, it causes the body to rotate about the axis aligned with the torque. The angular acceleration produced is inversely proportional to the body's moments and products of inertia. The moments of inertia correspond to the diagonal elements of the inertia tensor, while the products of inertia are represented by the off-diagonal elements.

If the products of inertia are non-zero, the applied torque can also induce rotational motion about the other two axes, a phenomenon known as the gyroscopic effect [3]. For example, when torque is applied along the x-axis, the products of inertia J_{xy} and J_{xz} cause rotation around the y and z axes, respectively. The same principle applies for torque applied along the y and z axes. According to Meriam et al. [3], the equation of motion for a rigid body can be written as follows:

$$\dot{H} = \sum T - \omega \times H \quad (2.1)$$

Where $H = J\omega$ is the angular momentum. If the body is constrained to rotate around only one axis (the axis where the torque is applied), the constraints are considered as a locking mechanism which applies a reaction torque, $T_{reaction}$ that matches the magnitude but opposes the direction of any torque that would otherwise cause rotation about the other two axes due to the products of inertia. As a result, this reaction torque cancels out any rotational motion about the other axes, ensuring the body rotates solely around the axis of applied torque.

$$J\dot{\omega} = T_{applied} - \omega \times J\omega - T_{reaction} \quad (2.2)$$

$$\begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} - \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} - \begin{bmatrix} T_{rx} \\ T_{ry} \\ T_{rz} \end{bmatrix} \quad (2.3)$$

$$\begin{cases} J_{xx}\dot{\omega}_x + J_{xy}\dot{\omega}_y + J_{xz}\dot{\omega}_z = T_x - \omega_y(J_{xz}\omega_x + J_{yz}\omega_y + J_{zz}\omega_z) + \omega_z(J_{xy}\omega_x + J_{yy}\omega_y + J_{yz}\omega_z) - T_{rx} \\ J_{xy}\dot{\omega}_x + J_{yy}\dot{\omega}_y + J_{yz}\dot{\omega}_z = T_y - \omega_z(J_{xx}\omega_x + J_{xy}\omega_y + J_{xz}\omega_z) + \omega_x(J_{xz}\omega_x + J_{yz}\omega_y + J_{zz}\omega_z) - T_{ry} \\ J_{xz}\dot{\omega}_x + J_{yz}\dot{\omega}_y + J_{zz}\dot{\omega}_z = T_z - \omega_x(J_{xy}\omega_x + J_{yy}\omega_y + J_{yz}\omega_z) + \omega_y(J_{xx}\omega_x + J_{xy}\omega_y + J_{xz}\omega_z) - T_{rz} \end{cases} \quad (2.4)$$

In equation 2.4, T_x, T_y and T_z represent the torques applied along each axis. Meanwhile, T_{rx}, T_{ry} and T_{rz} denote the reaction torques that constrain the body's rotation in two of the axes. This type of motion constraint, often referred to as rotational locking, can be achieved using ball bearings. However, ball bearings introduce friction torques due to their mechanical resistance. As a result, the applied torque is reduced by the friction torque, and the relationship can be expressed as

$T_{applied} = T_{motor} - T_{friction}$. Thus, the actual torque acting on the system is the motor torque minus the torque lost to friction.

A similar interpretation applies to the rotation of each gimbal. The shapes of both the inner and outer gimbals are asymmetrical, so their inertia tensors are non-diagonal. Each gimbal has one degree of freedom (DoF) and rotates using ball bearings.

2.1.1.1 Inner gimbal rotation

The entire terminal is already 3D modeled in detail, with the masses correctly positioned. As a result, the inertia tensor, J , can be easily extracted by specifying the correct coordinate system in the 3D modeling software. The inertia that the beta motor needs to overcome is that of the inner gimbal, J_β , represented as a constant 3×3 matrix. Since the applied torque is only provided around the x_β axis, the applied torque vector will be $T_{applied} = [T_x; 0; 0]$.

The reaction torques must lock the rotation around the y_β and z_β axes. The angular accelerations around y and z are induced by the applied torque $T_{applied}$ on x_β , as it is the only external torque acting on the inner gimbal. This coupling between the angular accelerations in y and z and the applied torque on x_β arises from the off-diagonal elements in the inertia tensor, specifically J_{xy} and J_{xz} .

To prevent angular acceleration in the y and z directions, the reaction torque $T_{reaction}$ must cancel out all terms in equation 2.4, including the coupling terms involving J_{xy} and J_{xz} . This ensures that the coupling between the applied torque, T_x , and the angular accelerations $\dot{\omega}_y$ and $\dot{\omega}_z$ is eliminated.

$$T_{reaction} = \begin{bmatrix} T_{rx} \\ T_{ry} \\ T_{rz} \end{bmatrix} = \begin{bmatrix} -J_{xy}\dot{\omega}_y - J_{xz}\dot{\omega}_z - J_{xz}\omega_x\omega_y + J_{xy}\omega_x\omega_z \\ -J_{xy}\dot{\omega}_x - J_{xy}\omega_y\omega_z - J_{xz}\omega_x^2 + J_{xz}\omega_x^2 \\ -J_{xz}\dot{\omega}_x - J_{xy}\omega_x^2 + J_{xy}\omega_y^2 + J_{xz}\omega_y\omega_z \end{bmatrix} \quad (2.5)$$

Consequently, the rotational equations describing the relationship between the applied torque and the resulting rotation around all axes of the inner gimbal are as follows:

$$\begin{bmatrix} J_{xx}\dot{\omega}_x = T_x - \omega_y(J_{yz}\omega_y + J_{zz}\omega_z) + \omega_z(J_{yy}\omega_y + J_{yz}\omega_z) \\ J_{yy}\dot{\omega}_y + J_{yz}\dot{\omega}_z = -\omega_z(J_{xx}\omega_x) + \omega_x(J_{yz}\omega_y + J_{zz}\omega_z) \\ J_{yz}\dot{\omega}_y + J_{zz}\dot{\omega}_z = -\omega_x(J_{yy}\omega_y + J_{yz}\omega_z) + \omega_y(J_{xx}\omega_x) \end{bmatrix} \quad (2.6)$$

2.1.1.2 Outer gimbal rotation

In the same way as the inner gimbal, the motion in the outer gimbal can also be expressed as shown in equation 2.4. In this case, the applied torque is $T_{applied} = [0; 0; T_z]$, as the alpha motor exerts torque along the z_α -axis. The reaction torques in the outer gimbal counteract the resulting rotations around the x_α and y_α axes. These undesired rotations arise from the off-diagonal inertia terms, J_{xz} and J_{yz} , which couple the applied torque along z_α to rotations around the other axes. Therefore, the reaction torques are as follows:

$$T_{reaction} = \begin{bmatrix} T_{rx} \\ T_{ry} \\ T_{rz} \end{bmatrix} = \begin{bmatrix} -J_{xz}\dot{\omega}_z - J_{xz}\omega_x\omega_y - J_{yz}\omega_y^2 + J_{yz}\omega_z^2 \\ -J_{yz}\dot{\omega}_z - J_{xz}\omega_z^2 + J_{xz}\omega_x^2 + J_{yz}\omega_y\omega_x \\ -J_{xz}\dot{\omega}_x - J_{yz}\dot{\omega}_y - J_{yz}\omega_x\omega_z + J_{xz}\omega_y\omega_z \end{bmatrix} \quad (2.7)$$

Consequently, the rotational equations describing the relationship between the applied torque and the resulting rotation around all axes of the outer gimbal are as follows:

$$\begin{bmatrix} J_{xx}\dot{\omega}_x + J_{xy}\dot{\omega}_y = -\omega_y(J_{zz}\omega_z) + \omega_z(J_{xy}\omega_x + J_{yy}\omega_y) \\ J_{xy}\dot{\omega}_x + J_{yy}\dot{\omega}_y = -\omega_z(J_{xx}\omega_x + J_{xy}\omega_y) + \omega_x(J_{zz}\omega_z) \\ J_{zz}\dot{\omega}_z = T_z - \omega_x(J_{xy}\omega_x + J_{yy}\omega_y) + \omega_y(J_{xx}\omega_x + J_{xy}\omega_y) \end{bmatrix} \quad (2.8)$$

The alpha motor must counteract the combined inertia of both the outer and inner gimbals, as the inner gimbal is attached to the outer gimbal. Since the inner gimbal's moment of inertia influences the outer gimbal variably, depending on the beta angle, the J matrix in equation 2.8 should incorporate the outer gimbal's inertia, J_α , and the transformed inertia of the inner gimbal, J_β , which varies based on the beta angle β .

$$J_{total} = J_\alpha + J_{\beta\alpha} \quad (2.9)$$

J_α and J_β are obtained from the 3D modeling software, while $J_{\beta\alpha}$ is calculated by transforming J_β into the outer gimbal's coordinates and then adding the inertial offset.

$$J_{\beta\alpha} = rot_x(\beta) J_\beta rot_x^{-1}(\beta) + m_\beta \cdot (|d|^2 I_3 - dd') \quad (2.10)$$

In equation 2.10, $\text{rot}_x(\beta)$ is the standard rotation matrix around the x axis, where the rotation angle is β . The term $\text{rot}_x(\beta) J_\beta \text{rot}_x^{-1}(\beta)$ is derived from the similarity transformation [5].

The term $m_\beta \cdot (|d|^2 I_3 - dd')$ is derived from the generalized parallel axis theorem [4] and represents the effect of the inner gimbal's center of gravity being offset from the outer gimbal's coordinate system. Where m_β is the mass of the inner gimbal, I_3 is an identity matrix of size 3×3 , and d is a 3×1 vector that includes the offsets along x_α , y_α , and z_α .

2.1.1.3 Summary

Incorporating the disturbances arising from the presence of products of inertia, and then including T_{reaction} to counteract the rotational motion around the other two axes, restricts the rotation to the desired axis. This results in the angular velocities and accelerations of the other axes being zero. By applying this to equations 2.6 and 2.8, the outcome can be expressed as follows:

$$\begin{cases} J_{\beta_{xx}} \dot{\omega}_{\beta_x} = T_x & \text{In the inner gimbal} \\ J_{\text{total}_{zz}} \dot{\omega}_{\alpha_z} = T_z & \text{In the outer gimbal} \end{cases} \quad (2.11)$$

By assuming that the mass center of the inner gimbal is located on the rotational axis of the outer gimbal, and that $J_{\beta_{yy}} = J_{\beta_{zz}}$, it follows that $J_{\text{total}_{zz}} = J_{\alpha_{zz}} + J_{\beta_{yy}}$. This assumption introduces a slight modification to Equation 2.11, leading to linearized equations describing the motion of the system as follows:

$$\begin{cases} J_{\beta_{xx}} \dot{\omega}_{\beta_x} = T_x & \text{For the inner gimbal,} \\ (J_{\alpha_{zz}} + J_{\beta_{yy}}) \dot{\omega}_{\alpha_z} = T_z & \text{For the outer gimbal.} \end{cases} \quad (2.12)$$

2.1.2 Friction modeling

The outer gimbal is mounted to the base using a ball bearing, and the cables run from the base to the outer gimbal through the bearing using a slip ring. The combination of the slip ring and the bearing introduces friction that must be considered. The inner gimbal is mounted to the outer gimbal using another ball bearing, and there are some cables connecting the inner and outer gimbals. These cables are flexible, and their resistance is assumed to be negligible.

The friction force can be described using different models, as shown in figure 2.3. According to Olsson et al, [1], the friction force in linear motion is represented by various models, each addressing certain aspects while neglecting others. The model illustrated in figure 2.3a describes the friction force as a sign function, where the friction remains constant regardless of the velocity; this model only considers coulomb's friction. The model in figure 2.3b presents a linear proportional relationship between the friction force and velocity, accounting for both coulomb's friction and viscous

friction. The model depicted in figure 2.3c incorporates the same relationships as model b but also addresses static friction when the velocity is zero. The most accurate representation is shown in figure 2.3d, where the model takes into account static friction and the dependencies between friction force and velocity, indicating that the friction force decreases from the static friction value at low velocities before it starts to increase again. Similarly, in rotational motion, the same models can be applied where the friction force in linear motion will be represented as friction torque in rotational motion.

The model illustrated in figure 2.3b is the most effective to use in this research since it captures the relationship between velocity and friction torques, and is relatively easier to model by conducting experiments on the physical terminal.

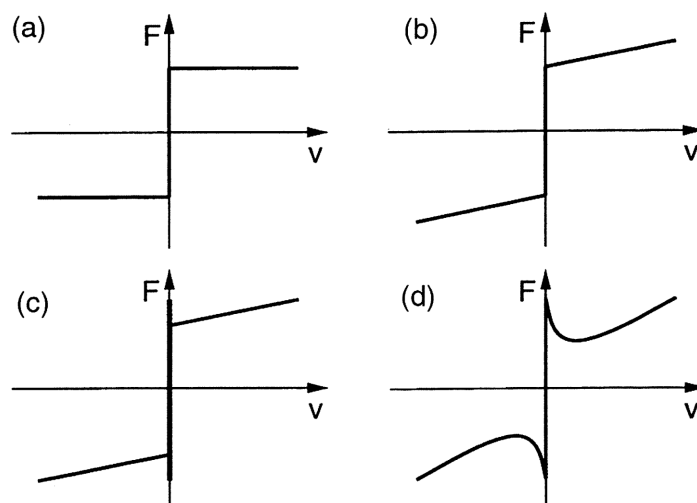


Figure 2.3: Different friction models: a) Coulomb's friction; b) coulomb's and viscous friction; c) Simplified model incorporating coulomb's, viscous, and static friction; d) A realistic representation of the friction force [1].

Assuming the friction torque is linear, it can be described by the equation

$$T_{\text{friction}} = a\omega + b \quad (2.13)$$

To obtain the parameters a and b , an experiment can be conducted where the output torque of each motor is recorded while maintaining a constant angular speed. By conducting the experiment twice for each motor at two different constant angular speeds, it is possible to find the equation of the fitting line that passes through the recorded points, resulting in the values of a and b for each gimbal.

2.1.3 ESC Modeling

The Electronic Speed Controller (ESC) is responsible for generating the desired torque on the motor shaft as demanded by the controller. However, there is a delay between the torque request and the actual torque reaching its desired magnitude. This delay arises due to the internal dynamics of the ESC. For modeling purposes, the ESC is treated as a black box. A step torque input is applied to the physical hardware, and the resulting torque output is recorded. The system dynamics are then identified based on this recorded torque data.

The dynamics of the ESC are approximated using a first-order transfer function:

$$\frac{1}{Ts + 1} \tag{2.14}$$

In this expression, T is the time constant, which represents the time required for the motor's response to reach 63.2% of its final value in reaction to a step input.

The value of 63.2% is derived from the step response equation in the time domain. Let $y(t)$ denote the system's step response. The input to the system is a unit step function with a magnitude of 1, which has a Laplace transform of $\frac{1}{s}$. The step response in the Laplace domain, denoted as $Y(s)$, is obtained by multiplying the unit step function by the first-order transfer function. To find the time-domain step response $y(t)$, the inverse Laplace transform of $Y(s)$ is computed:

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \mathcal{L}^{-1}\left\{\frac{1}{s} \cdot \frac{1}{Ts + 1}\right\} = 1 - e^{-\frac{t}{T}}$$

When $t = T$, the system reaches one time constant, and the output attains 63.2% of its final value. At this point, $y(t) = 1 - e^{-1} \approx 0.632$.

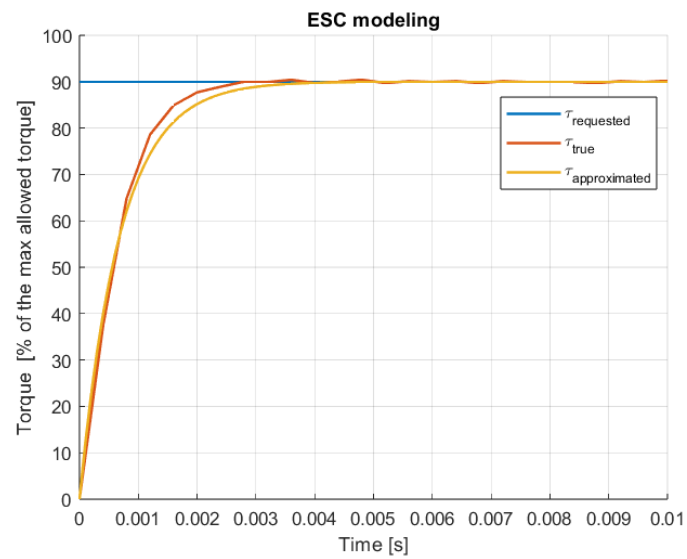


Figure 2.4: Comparison of the step response of the actual ESC and the first-order model. The ESC is tasked with delivering 90% of the maximum torque specified by the motor manufacturer.

2.1.4 Sensor modeling

The whole system is designed inside a simulation environment where a simulation scenario is inputted to the system and a sensor block will take the vehicle's position and orientation at each time step and convert them to sensors' signals. By having the sensors' signals inside the simulation, it is possible to validate the sensor-fusion algorithms inside the simulation environment. Sensors are characterized on the basis of both gathered data and the data sheets provided by manufacturers.

2.1.4.1 Encoders

An incremental encoder is a device mounted on the motor shaft that generates pulses corresponding to the shaft's rotation. The resolution of the encoder is determined by the number of pulses produced per revolution. To accurately monitor the shaft's position, the encoder count must be reset to a known reference point through a process known as *homing*. From this reference position, the shaft angle can be tracked by accumulating the encoder pulses.

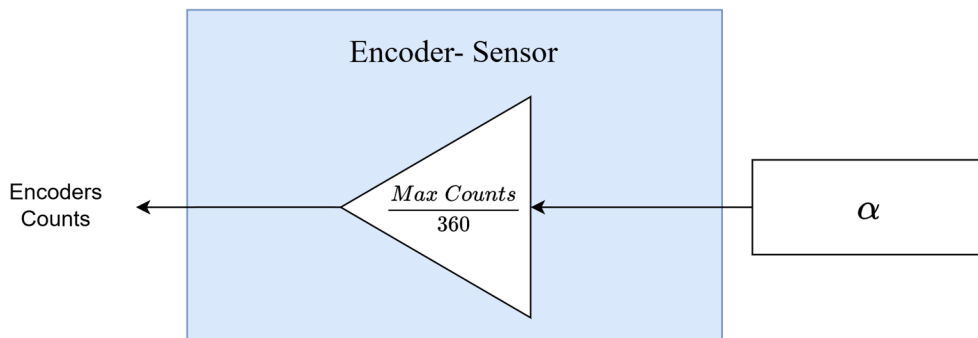


Figure 2.5: Encoder model in simulation. The model takes the true angle (in degrees) and converts it to encoder counts. Here, *Max Counts* denotes the total number of pulses generated by the encoder per revolution, as specified in the device datasheet.

2.1.4.2 Differential GNSS

A Differential GNSS sensor is a dual GNSS module that provides information about the position on Earth and the absolute angle relative to the north. This is achieved by comparing the positions of two GNSS antennas mounted on opposite sides of the terminal.

In simulation, position accuracy is assumed to be ideal, as translational errors have minimal impact on the system due to the significant distance between the antenna and the satellites.

However, the heading angle estimation provided by the module is noisy and requires characterization and modeling within the simulation.

By logging the heading estimation from the module over several hours, it was observed that the only significant error source was White Gaussian Noise (*WGN*) with

a constant standard deviation over time. The standard deviation was determined from the logged data.

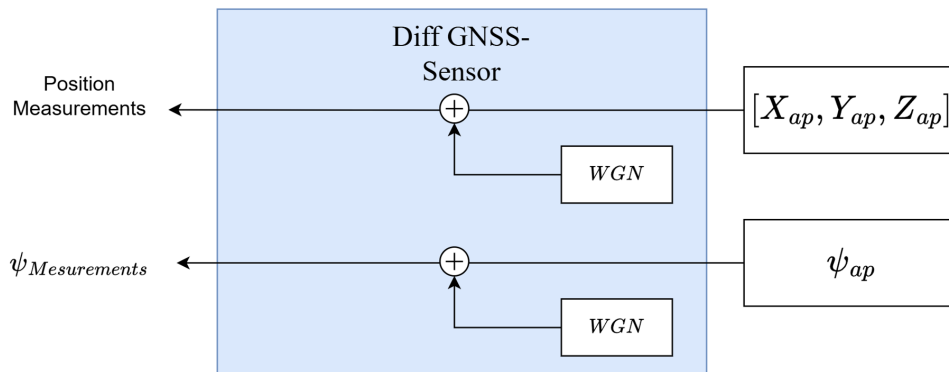


Figure 2.6: Differential GNSS module model in simulation. It takes the position and the heading action points at each time step and converts them to signals from the GNSS sensor. The position measurements are assumed to be ideal, whereas for the heading estimation, the *WGN* is characterized based on analysis of logged sensor data.

2.1.4.3 Accelerometer

A three-axis accelerometer is a device that measures linear acceleration along all three cartesian axes, x , y , and z , in its own body frame. When the accelerometer is at rest, it only measures the gravitational acceleration g . The effect of gravitational acceleration on the accelerometer depends on the orientation of the accelerometer, which is mounted on the outer gimbal. By knowing the relationship between the IMU body frame and the global frame (i.e., the angle α), the impact of the gravitational acceleration is calculated as follows:

$$Acc_g = \begin{bmatrix} Acc_{x_g} \\ Acc_{y_g} \\ Acc_{z_g} \end{bmatrix} = rot_z(\alpha) \cdot \begin{bmatrix} -g \cdot \cos(\theta) \cdot \sin(\varphi) \\ g \cdot \sin(\theta) \\ g \cdot \cos(\varphi) \cdot \cos(\theta) \end{bmatrix} \quad (2.15)$$

In addition to the gravitational acceleration, the linear acceleration caused by the motion of the vehicle is assumed to be negligible, as the system is primarily used when the vehicle is stationary or moving at a relatively constant velocity.

Since the accelerometer is mounted with an offset from the rotational axis of the outer gimbal, the angular acceleration of the gimbal, $\ddot{\alpha}$, introduces a linear acceleration in the body frame of the accelerometer, as illustrated in Figure 2.7. However, as the system actively maintains alignment with the satellite, the angular acceleration ($\ddot{\alpha}$) counteracts the rotation of the vehicle ($\ddot{\psi}$), resulting in a net angular acceleration in the world frame. This introduces an induced acceleration Acc_i in the x direction of the accelerometer's body frame.

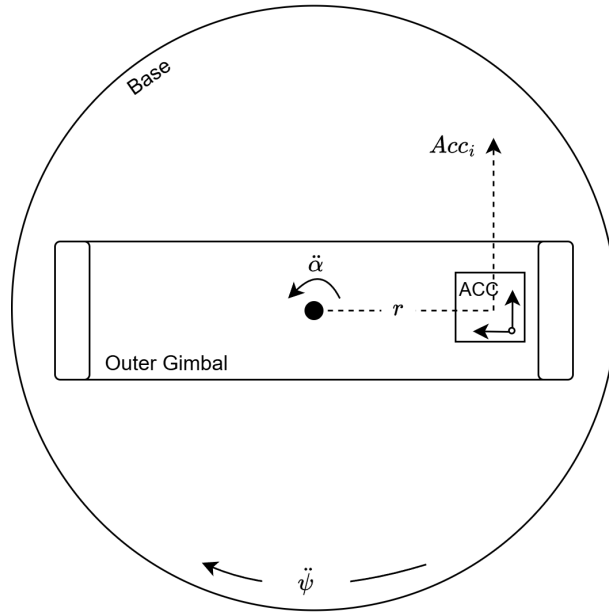


Figure 2.7: A top view of the terminal shown in Figure 2.2. When the car changes its heading rate, the antenna moves in the opposite direction to maintain line-of-sight with the satellite. The difference between these angular accelerations results in a net angular acceleration acting on the system, introducing an induced acceleration Acc_i in the x direction of the accelerometer's body frame.

The induced acceleration is also proportional to the distance, r , between the accelerometer position and the rotation axis.

$$Acc_i = \begin{bmatrix} (\ddot{\alpha} - \ddot{\psi}) \cdot r \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

Finally, a white gaussian noise (WGN), characterized by logging and analyzing accelerometer measurements, is added to the model. The total acceleration is then expressed as:

$$Acc_{total} = Acc_g + Acc_i + WGN \quad (2.17)$$

Expanding equation 2.17 yields the following result:

$$Acc_{total} = rot_z(\alpha) \cdot \begin{bmatrix} -g \cdot \cos(\theta) \cdot \sin(\varphi) \\ g \cdot \sin(\theta) \\ g \cdot \cos(\varphi) \cdot \cos(\theta) \end{bmatrix} + \begin{bmatrix} (\ddot{\alpha} - \ddot{\psi}) \cdot r \\ 0 \\ 0 \end{bmatrix} + WGN$$

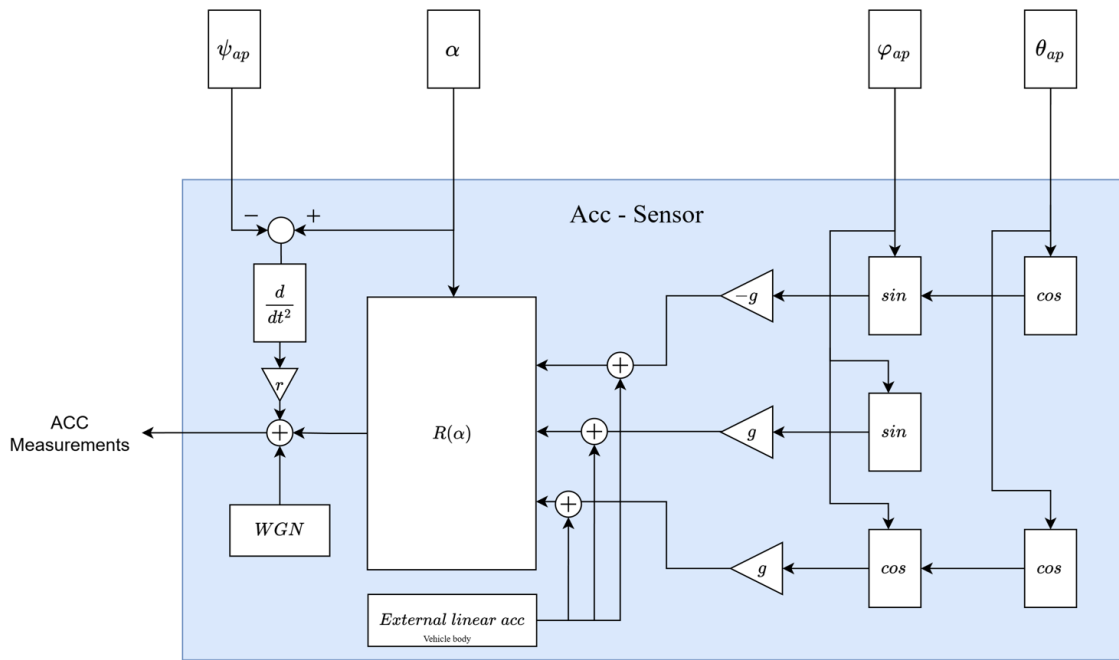


Figure 2.8: The accelerometer model in simulation. As mentioned, external linear accelerations introduced by the car are not considered.

2.1.4.4 Gyroscope

A three-axis gyroscope is a device that measures angular velocity along all three cartesian axes, x , y , and z , in its own body frame. Since the gyroscope is mounted on the outer gimbal, its readings in the body frame must be compensated for the gimbal angle α . Consequently, the gyro readings represent the derivative of the car's orientation over time.

$$Gyro = rot_z(\alpha) \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.18)$$

This situation is similar to the one described in Figure 2.7. A change in the car's heading ($\dot{\psi}$) is counteracted by the antenna's motion in the opposite direction ($\dot{\alpha}$). This results in a net angular velocity, which is the difference between these two components.

$$Gyro = rot_z(\alpha) \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\dot{\alpha} \end{bmatrix} \quad (2.19)$$

According to the gyroscope's datasheet, calibration is necessary to reduce drift, which occurs when gyro data is integrated to calculate angles. Calibration is performed by locking the motors to ensure that the gimbal remains stationary, parking

the vehicle, and calculating the mean value of the readings, referred to as the *Bias*. This bias is then subtracted from the gyro readings.

The bias is unstable and varies over time. A high-quality gyroscope exhibits good bias stability, meaning the bias can be effectively compensated. Bias instability can be modeled as a constant value added to the gyro readings. To measure bias instability, the gyroscope was operated in a stationary state while its data was recorded. From this recorded data, the bias instability was quantified, revealing the drift in degrees per second.

Finally, a white gaussian noise (*WGN*) is added to the model, which is also characterized based on the recorded data.

$$Gyro = rot_z(\alpha) \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\dot{\alpha} \end{bmatrix} + Bias\ instability + WGN \quad (2.20)$$

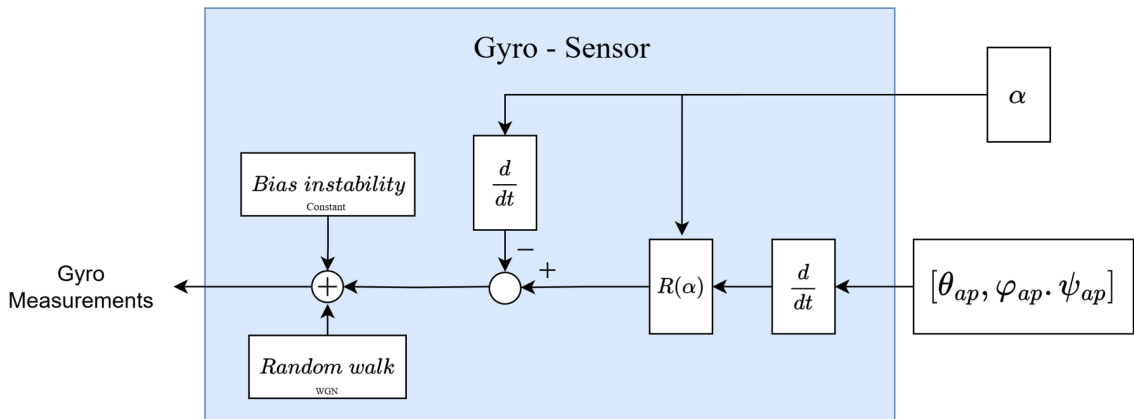


Figure 2.9: The gyroscope model in simulation. The term "R andom walk" is a standard term for *WGN* in gyroscope contexts. Both the bias instability and random walk are modeled based on the logged data.

2.2 Kinematics

The antenna will be aligned with the satellite by first rotating around the vertical axis, starting from geographic north. This angle of rotation is called the azimuth. Then, the antenna will be tilted upward from the horizon to point towards the satellite, and this angle is known as the elevation [2]. Once the antenna is aligned with the satellite, it must rotate around the line of sight between the antenna and the satellite to match the polarization of the electromagnetic signal transmitted by the satellite. Instead of physically rotating the antenna to align with the satellite's polarization, the signal processing hardware can compensate for the polarization mismatch through software adjustments. By knowing both the terminal's position and the satellite's position, the global azimuth and elevation target angles can be calculated, while the polarization target angle is provided by the satellite. Additionally, by knowing the vehicle's orientation (roll, pitch, yaw), reference values for the α and β angles can be determined.

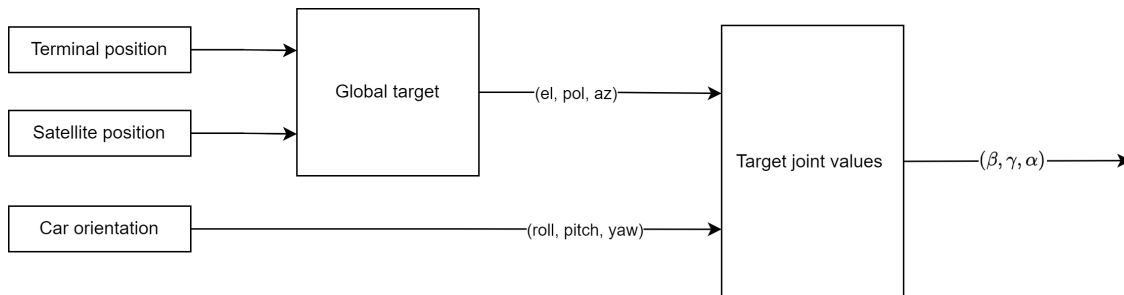


Figure 2.10: The function Global Target calculates the target elevation, azimuth, and polarization angles in a global reference frame, based on the positions of the terminal and the satellite. Meanwhile, the function Target Joint Values computes the local joint angles β , λ and α required to adjust for the terminal's orientation, ensuring the antenna is correctly aligned with the satellite.

2.2.1 Global Target

The satellite, with which the antenna needs to align, is located at a fixed position above the equator. In simulation, both the vehicle's and the satellite's positions are expressed in cartesian coordinates, (x, y, z) , whereas in the real world, they are expressed in spherical coordinates $(r, \text{longitude}, \text{latitude})$.

- In the real world: Given the vehicle's position at $(r_0, \text{lon}_v, \text{lat}_v)$ and the satellite's position at $(r_0 + h, \text{lon}_s, \text{lat}_s)$, where:
 - lon_v is the vehicle's longitude, while lat_v denotes the vehicle's latitude. The same is for the satellite, expressed as lon_s and lat_s respectively.
 - r_0 is the Earth's radius.
 - h is the satellite's altitude (height above sea level).

2. Theory

- The latitude of the satellite, lat_s , is 0 degrees since all geostationary satellites are positioned above the equator in the geostationary orbit.

The global target elevation and azimuth angles are given by Maral et al, [2], as following :

$$\text{Elevation} = \tan^{-1} \left(\frac{\cos(\text{lat}_v) \cdot \cos(\text{lon}_v - \text{lon}_s) - \frac{r_0}{r_0+h}}{\sqrt{1 - (\cos(\text{lat}_v) \cdot (\text{lon}_v - \text{lon}_s))^2}} \right) \quad (2.21)$$

$$a = \tan^{-1} \left(\frac{\tan(|\text{lon}_v - \text{lon}_s|)}{\sin(\text{lat}_v)} \right)$$

	Satellite EAST of antenna	Satellite WEST of antenna
North Hemisphere	<i>Azimuth</i> = $180 - a$	<i>Azimuth</i> = $180 + a$
South Hemisphere	<i>Azimuth</i> = a	<i>Azimuth</i> = $360 - a$

- In simulation: Calculating the satellite's orientation relative to the vehicle (or terminal's base) can be done using the following equations, where the variables correspond to the figure 2.11.

$$\text{Azimuth} = \text{atan2}(\Delta y, \Delta x) \quad (2.22)$$

The azimuth angle represents the satellite's position in the xy -plane relative to the terminal base.

$$\text{Elevation} = \text{atan2}(\Delta z, L) \quad (2.23)$$

The elevation angle represents the vertical angle between the terminal base and the satellite, indicating how high the satellite appears in the sky relative to the horizon from the terminal's perspective. It is calculated based on the difference in altitude (Δz) and the horizontal distance (L) between the terminal and the satellite.

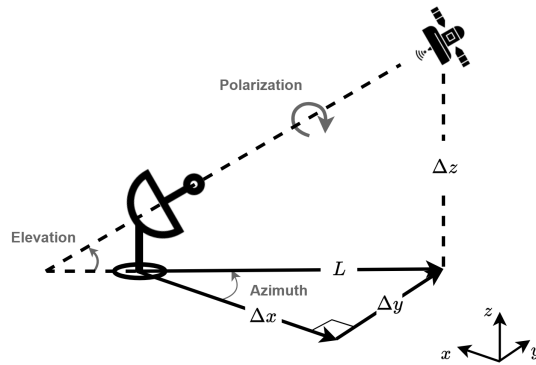


Figure 2.11: An illustration depicting how azimuth and elevation angles are calculated.

2.2.2 Target joint values

The orientation of an object can be described using three rotational matrices representing rotations about the x , y , and z axes. Given the known orientation of the satellite in the global frame and the estimated orientation of the terminal base, it is possible to calculate the reference angles β , λ , and α using the following formulas:

$$O_{Target} = O_{Base} \cdot O_{Antenna} \quad (2.24)$$

$$O_{Antenna} = O_{Base}^{-1} \cdot O_{Target} \quad (2.25)$$

$$rot_z(\alpha) \cdot rot_x(\beta) \cdot rot_y(\lambda) = O_{Base}^{-1} \cdot O_{Target}$$

$$\begin{bmatrix} c\alpha c\lambda - s\alpha s\beta s\lambda & -s\alpha c\beta & c\alpha s\lambda + s\alpha s\beta c\lambda \\ s\alpha c\lambda + c\alpha s\beta s\lambda & c\alpha c\beta & s\alpha s\lambda - c\alpha s\beta c\lambda \\ -c\beta s\lambda & s\beta & c\beta c\lambda \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (2.26)$$

From equation 2.26, it is possible to solve:

- $\beta = \sin^{-1}(R_{32})$
- $\lambda = \text{atan2}\left(\frac{-R_{31}}{R_{33}}\right)$
- $\alpha = \text{atan2}\left(\frac{-R_{12}}{R_{22}}\right)$

2.3 Processing step for sensors

The data collected from encoders, gyroscope, accelerometer, and differential GNSS must be processed before being fused to accurately estimate the vehicle's orientation at each time step. The processing of sensors data is essential to extract roll, pitch, and yaw angles, which represent the vehicle's orientation. The encoders, gyroscope, and accelerometer data require specific processing steps, while the heading measurements from the differential GNSS sensor do not need any processing since they directly provide the estimated yaw angle of the vehicle. The following sections explain the processing steps for each sensor in detail.

2.3.1 Encoder Data Processing

The encoder measurements are recorded in counts, representing the rotational position of a shaft. To calculate the angle α (in degrees) from these counts, a straightforward conversion is applied. Since one revolution equals 360 degrees, the angle is determined by dividing 360 by the maximum number of counts per revolution (as specified in the encoder's datasheet) and then multiplying the result by the encoder's measured counts. This process is illustrated in Figure 2.12.

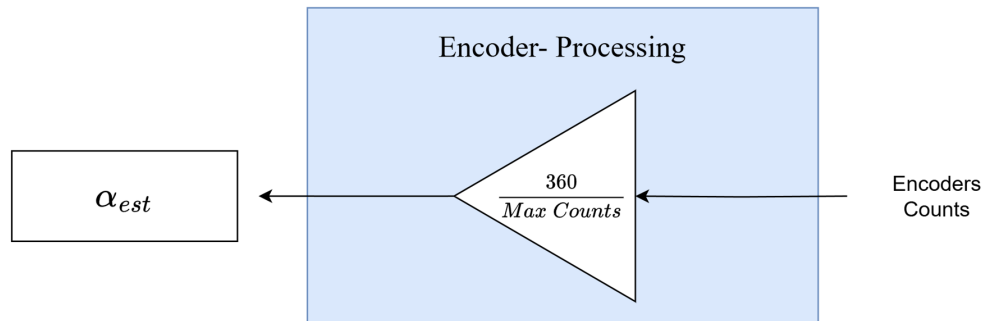


Figure 2.12: Encoder processing step to estimate the angle α as a block diagram.

2.3.2 Gyroscope Data Processing

The gyroscope provides angular velocity measurements and is affected by rotations due to the angle α as explained in section 2.1.4.4. To accurately estimate roll, pitch, and yaw angles, this influence of α must first be removed. The corrected data is then rotated back along the z-axis to eliminate any effects from α . Afterward, the angular velocity data are integrated to compute the orientation angles.

This process is visualized in Figure 2.13.

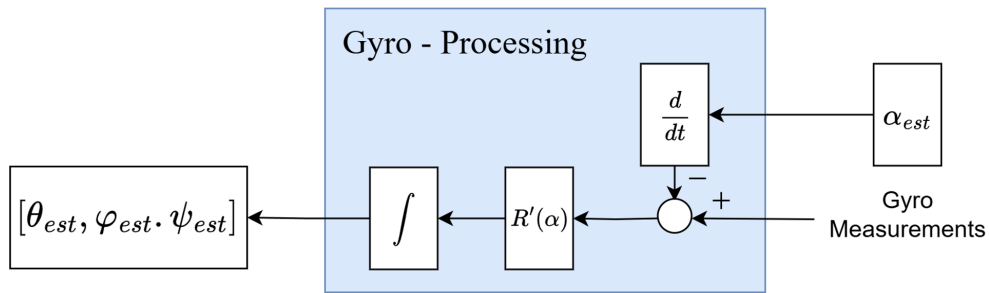


Figure 2.13: Gyroscope processing step to estimate roll, pitch, and yaw angles as a block diagram. The estimated α is differentiated and subtracted from the gyroscope's z-axis measurements. The data is then rotated back along the z-axis by α to isolate the raw measurements, which are subsequently integrated to estimate the orientation angles.

2.3.3 Accelerometer Data Processing

The accelerometer measures linear accelerations along the x, y and z axes. However, these measurements are influenced by additional effects from the angles α and ψ_{ap} as explained in section 2.1.4.3. These effects must be removed, and the data must be rotated back along the z-axis to provide usable measurements.

After removing all the additional effects, the accelerometer data looks as in the equation 2.27:

$$Acc_g = \begin{bmatrix} Acc_{x_g} \\ Acc_{y_g} \\ Acc_{z_g} \end{bmatrix} = \begin{bmatrix} -g \cdot \cos(\theta) \cdot \sin(\varphi) \\ g \cdot \sin(\theta) \\ g \cdot \cos(\varphi) \cdot \cos(\theta) \end{bmatrix} \quad (2.27)$$

From this processed data, it is possible to directly estimate the roll (φ) and pitch (θ) angles. The roll angle (φ) is estimated as in equation 2.28. Using the Pythagorean trigonometric identity, it is possible to estimate the pitch angle (θ) as well, as shown in equation 2.29.

$$\varphi_{est} = -\arctan\left(\frac{acc_x}{acc_z}\right) \quad (2.28)$$

$$\theta_{est} = \arctan\left(\frac{acc_y}{\sqrt{acc_x^2 + acc_z^2}}\right) \quad (2.29)$$

The processing workflow is depicted in Figure 2.14.

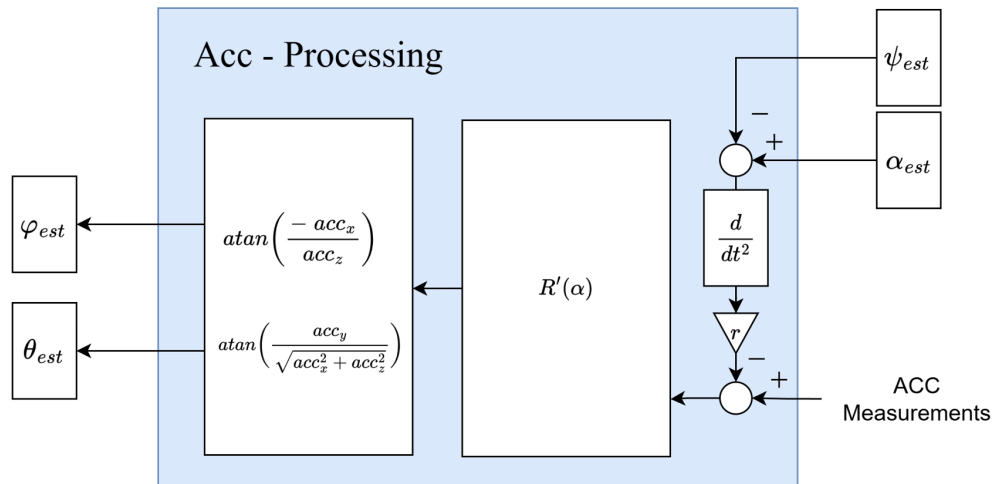


Figure 2.14: Accelerometer processing step to estimate roll and pitch angles as a block diagram. The accelerometer data is processed by removing the effects of angles α and ψ_{est} and rotating the data back along the z-axis. The roll and pitch angles are then calculated using the equations above.

2.4 Sensor Fusion

After processing the sensor data, the next step is to fuse the measurements from all sensors. Sensor fusion aims to combine the strengths of individual sensors while compensating for their limitations. For this research, two prominent fusion algorithms were considered: the complementary filter and the Kalman filter. These algorithms were tested and compared to evaluate their performance in estimating the vehicle's orientation.

2.4.1 Complementary filter

The complementary filter is a simple yet effective sensor fusion technique that combines the outputs of two sensors using a low-pass and a high-pass filter [6]. The low-pass filter is applied to the accelerometer data because accelerometers are prone to high-frequency noise. By suppressing this noise, the low-pass filter helps retain the reliable low-frequency components of the accelerometer readings. On the other hand, a high-pass filter is applied to the gyroscope data because gyroscopes provide accurate measurements of angular velocity, particularly for detecting sudden motions. The high-pass filter ensures that these rapid changes are captured while mitigating the effects of low-frequency drift, which is a common limitation of gyroscopes. The complementary filter equation in the continuous time domain is derived as follows [6]:

$$\theta_{\text{est}} = G_{\text{lpf}}(s) \cdot \theta_a + G_{\text{hpf}}(s) \cdot \theta_g \quad (2.30)$$

$$\begin{aligned} G_{\text{lpf}}(s) &= \frac{1}{1 + T_C s}, & G_{\text{hpf}}(s) &= \frac{T_C s}{1 + T_C s} \\ \theta_{\text{est}} &= \frac{1}{1 + T_C s} \cdot \theta_a + \frac{T_C s}{1 + T_C s} \cdot \frac{1}{s} \cdot \omega_g \\ (1 + T_C s) \cdot \theta_{\text{est}} &= \theta_a + T_C \cdot \omega_g \end{aligned} \quad (2.31)$$

Here, the variables and filters are defined as follows:

- $G_{\text{lpf}}(s)$: Low-pass filter for accelerometer data.
- $G_{\text{hpf}}(s)$: High-pass filter for gyroscope data.
- T_C : Time constant of the filter.
- θ_a : Angle estimate from accelerometer.
- θ_g : Integrated angle from gyroscope.
- ω_g : Angular velocity from gyroscope.
- θ_{est} : Resultant angle estimation from the complementary filter.

The next step involves converting Equation 2.31 from Laplace domain to the time domain, resulting in:

$$\theta_{\text{est}}(t) + T_C \cdot \frac{d\theta_{\text{est}}(t)}{dt} = \theta_a(t) + T_C \cdot \omega_g(t) \quad (2.32)$$

To implement the filter in discrete time, Equation 2.32 is discretized, assuming a sampling time of T_s :

$$\begin{aligned} \frac{d\theta(t)}{dt} &\rightarrow \frac{\theta(n) - \theta(n-1)}{T_s} \\ \theta_{\text{est}}(n) + T_C \cdot \frac{\theta_{\text{est}}(n) - \theta_{\text{est}}(n-1)}{T_s} &= \theta_a(n) + T_C \cdot \omega_g(n) \\ \theta_{\text{est}}(n) &= \frac{T_C}{T_C + T_s} \cdot \theta_{\text{est}}(n-1) + \frac{T_s}{T_C + T_s} \cdot \theta_a(n) + \frac{T_C \cdot T_s}{T_C + T_s} \cdot \omega_g(n) \end{aligned} \quad (2.33)$$

To simplify the notation, the complementary filter constants are defined as:

$$\gamma = \frac{T_C}{T_C + T_s}, \quad 1 - \gamma = \frac{T_s}{T_C + T_s} \quad 0 \leq \gamma \leq 1$$

Substituting these constants, the final complementary filter equation is given as:

$$\theta_{\text{est}}(n) = \gamma \cdot [\theta_{\text{est}}(n-1) + T_s \cdot \omega_g(n)] + (1 - \gamma) \cdot \theta_a(n) \quad (2.34)$$

The constant γ determines the weighting of gyroscope and accelerometer data in the complementary filter. A value of $\gamma = 1$ implies complete reliance on gyroscope measurements, while $\gamma = 0$ means only accelerometer data is used. In practice, γ is chosen between these extremes to balance the noise characteristics and drift of the sensors, resulting in a robust angle estimation.

2.4.1.1 Application in This Research

One of the main limitations of the complementary filter, as applied in this research, is its inability to estimate the yaw angle effectively using accelerometer data. Unlike pitch and roll, the yaw angle cannot be derived from accelerometer measurements, as it depends primarily on the horizontal plane, which accelerometers cannot directly measure. Consequently, the yaw angle must be estimated solely through the integration of gyroscope measurements. However, this method leads to drift over time, resulting in an inaccurate yaw estimation.

Additionally, the complementary filter is inherently limited to combining data from only two sensors, as shown in Equation 2.34. Since the constants γ and $1 - \gamma$ are directly applied to the estimations from these two sensors, any scenario where one of the sensors becomes unavailable or provides intermittent data leads to erroneous outputs. This limitation is particularly problematic in this research for yaw angle estimation. In such cases, the filter relies solely on the integration of gyroscope data, which worsens the drift problem over time due to the lack of corrective input from a secondary sensor.

2.4.2 Kalman filter

The Kalman filter is a special case of Bayesian filtering where the process and measurement models are linear and Gaussian [7]. It has long been regarded as the optimal solution for many tracking and data prediction tasks, due to its ability to minimize the mean squared error.

For the state vector x_k (which represents the variables we want to estimate at time k) and the measurement y_k (which represents sensors observations), the Kalman filter operates under the following process and measurement models [7]:

$$x_k = A_{k-1}x_{k-1} + q_{k-1}, \quad q_{k-1} \sim \mathcal{N}(0, Q_{k-1}) \quad (2.35)$$

$$y_k = H_k x_k + r_k, \quad r_k \sim \mathcal{N}(0, R_k) \quad (2.36)$$

Here:

- x_k : The state vector at time k , containing the quantities being estimated.
- A_{k-1} : The state transition matrix, which describes how the states evolve from $k - 1$ to k based on the system dynamics.
- q_{k-1} : The process noise, modeled as Gaussian noise with zero mean and covariance Q_{k-1} , representing uncertainty in the process model.

- y_k : The measurement vector at time k , containing sensors observations.
- H_k : The measurement model matrix, which maps the state x_k to the expected measurement y_k .
- r_k : The measurement noise, Gaussian noise with zero mean and covariance R_k , representing uncertainty in the measurements.

The Kalman filter consists of two main steps: **Prediction** and **Update**.

2.4.2.1 Prediction Step

The prediction step estimates the state and covariance at time k using information from time $k - 1$ [7]:

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} \quad (\text{Predicted state estimate}) \quad (2.37)$$

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^\top + Q_{k-1} \quad (\text{Predicted state covariance}) \quad (2.38)$$

Here:

- $\hat{x}_{k|k-1}$: The predicted state at time k given the state estimate from time $k - 1$.
- $P_{k|k-1}$: The predicted covariance matrix, representing the uncertainty of the state prediction.

2.4.2.2 Update Step

The update step corrects the predicted state and covariance using the new measurement y_k [7]:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k V_k \quad (\text{Updated state estimate: Prediction + Correction}) \quad (2.39)$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^\top \quad (\text{Updated covariance matrix}) \quad (2.40)$$

Here:

- $\hat{x}_{k|k}$: The updated state estimate at time k .
- $P_{k|k}$: The updated covariance matrix, representing the uncertainty after incorporating the measurement.
- K_k : The Kalman gain, which determines the weight given to the new measurement compared to the prediction.
- V_k : The innovation or residual, which is the difference between the actual measurement y_k and the predicted measurement.
- S_k : The innovation covariance, which measures the uncertainty in the innovation.

The intermediate variables are defined as:

$$K_k = P_{k|k-1} H_k^\top S_k^{-1} \quad (\text{Kalman Gain}) \quad (2.41)$$

$$V_k = y_k - H_k \hat{x}_{k|k-1} \quad (\text{Innovation: Measurement residual}) \quad (2.42)$$

$$S_k = H_k P_{k|k-1} H_k^\top + R_k \quad (\text{Innovation Covariance: Total uncertainty in the residual}) \quad (2.43)$$

The Kalman gain K_k determines how much the measurement y_k influences the updated state. A higher Kalman gain gives more weight to the measurement, while a lower gain prioritizes the prediction or the process.

2.4.2.3 Application in This Research

In this research, the Kalman filter was designed using a constant turn rate model, which includes the orientation angles (roll, pitch, and yaw) and their angular rates as states. A constant turn rate model was chosen because orientation changes in a vehicle typically occur gradually, making it reasonable to assume that angular accelerations are negligible. This assumption implies that the angular rates of the orientation angles remain approximately constant over time, with only minor variations due to process noise.

With this model as the process model, the Kalman filter state vector comprises six states: the orientation angles (roll, pitch, and yaw) and their respective angular rates. The angular rates are modeled as constant values perturbed by process noise, while the orientation angles are updated by integrating the angular rates over each time step, also with the addition of process noise. This results in the following state-space representation:

$$\begin{bmatrix} \varphi_k \\ \dot{\varphi}_k \\ \theta_k \\ \dot{\theta}_k \\ \psi_k \\ \dot{\psi}_k \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \varphi_{k-1} \\ \dot{\varphi}_{k-1} \\ \theta_{k-1} \\ \dot{\theta}_{k-1} \\ \psi_{k-1} \\ \dot{\psi}_{k-1} \end{bmatrix} + q_{k-1}, \quad q_{k-1} \sim \mathcal{N}(0, Q_{k-1}) \quad (2.44)$$

where φ , θ , and ψ represent the roll, pitch, and yaw angles, respectively; $\dot{\varphi}$, $\dot{\theta}$, and $\dot{\psi}$ are their angular rates; T_s is the sampling time; and q_{k-1} is the process noise, which is modeled as Gaussian with zero mean and covariance Q_{k-1} .

The tuning of the process noise covariance matrix Q_{k-1} , both in general and in this research, is discussed in Section 2.4.2.4.

Using this process model, two Kalman filter variants were developed to estimate the vehicle's orientation. These variants differ in their measurement models and are detailed in the following sections.

2.4.2.3.1 Kalman based on IMU and differential GNSS

This filter integrates measurements from a 6-axis IMU (gyroscope and accelerometer) and a differential GNSS sensor. The differential GNSS sensor provides heading measurements (yaw angle), which is used to refine the orientation estimation and mitigate IMU drift over time. The available measurements in this variant include roll and pitch angles derived from the accelerometer (discussed in Section 2.3.3), the yaw angle from the differential GNSS sensor, and the angular rates from the gyroscope. These measurements are combined to produce the following measurement model:

$$\begin{bmatrix} \varphi_{m_k} \\ \dot{\varphi}_{m_k} \\ \theta_{m_k} \\ \dot{\theta}_{m_k} \\ \psi_{m_k} \\ \dot{\psi}_{m_k} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \varphi_k \\ \dot{\varphi}_k \\ \theta_k \\ \dot{\theta}_k \\ \psi_k \\ \dot{\psi}_k \end{bmatrix} + r_k, \quad r_k \sim \mathcal{N}(0, R_k) \quad (2.45)$$

Here:

- φ_{m_k} , θ_{m_k} , and ψ_{m_k} represent the measured roll, pitch, and yaw angles, respectively.
- $\dot{\varphi}_{m_k}$, $\dot{\theta}_{m_k}$, and $\dot{\psi}_{m_k}$ represent the measured angular rates.
- φ_k , θ_k , and ψ_k are the true roll, pitch, and yaw angles, while $\dot{\varphi}_k$, $\dot{\theta}_k$, and $\dot{\psi}_k$ are the true angular rates.
- r_k is the measurement noise, assumed to follow a Gaussian distribution with zero mean and covariance R_k .

2.4.2.3.2 Kalman Filter Based on IMU-Only

This variant of the Kalman filter relies exclusively on measurements from the 6-axis IMU, which includes a gyroscope and an accelerometer. The gyroscope provides angular velocity data, enabling the estimation of changes in orientation over time, while the accelerometer is used to estimate the roll and pitch angles.

In the absence of heading (yaw angle) measurements from a GNSS sensor, this filter estimates the yaw angle using only gyroscope data. However, gyroscope measurements are susceptible to drift over time due to cumulative integration errors, making the yaw angle estimation prone to long-term inaccuracies. In addition, this filter requires an initial sky-scanning procedure to establish a reference or starting position relative to a known heading (e.g., relative to a satellite's position).

The measurement model used for this filter is expressed as follows:

$$\begin{bmatrix} \varphi_{m_k} \\ \dot{\varphi}_{m_k} \\ \theta_{m_k} \\ \dot{\theta}_{m_k} \\ \dot{\psi}_{m_k} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \varphi_k \\ \dot{\varphi}_k \\ \theta_k \\ \dot{\theta}_k \\ \psi_k \\ \dot{\psi}_k \end{bmatrix} + r_k, \quad r_k \sim \mathcal{N}(0, R_k) \quad (2.46)$$

Here, the measurement model matrix \mathbf{H} directly relates the measured roll (φ_{m_k}), pitch (θ_{m_k}), and their angular velocities to the state variables. Since yaw (ψ_k) is not directly observable without GNSS data and only the yaw rate ($\dot{\psi}_{m_k}$) is directly observable by the gyroscope, the measurement model matrix \mathbf{H} is a 5x6 matrix and differs from the previous filter's measurement model matrix.

2.4.2.4 Tuning the Complementary and Kalman Filters

The complementary filter has only one tuning parameter, γ , with its complement $1 - \gamma$. Tuning this filter primarily involves understanding the characteristics of the two sensors being fused. If the gyroscope estimation is highly accurate but suffers from drift over time, γ is chosen to be close to 1, placing greater trust in the gyroscope's short-term estimates while using the accelerometer primarily to correct for drift.

In contrast, tuning the Kalman filter is more complex due to its reliance on covariance matrices to model process and measurement uncertainties. Specifically, the process covariance matrix \mathbf{Q} is the primary tuning parameter. Each state in the Kalman filter requires careful adjustment of its corresponding variance in \mathbf{Q} .

For instance, when the GNSS sensor is identified as a noisy measurement source, and the gyroscope measurements are observed to be more reliable, the following adjustments are made:

- The process model for states influenced by the GNSS sensor, such as the yaw angle, is given higher trust compared to the sensor's measurements. This ensures that the filter relies more on the model's predictions than on the noisy GNSS data.
- Conversely, the states measured by the gyroscope, such as roll angular rate, are weighted more heavily in the filter compared to their corresponding process models. This allows the filter to leverage the gyroscope's reliability for these states.

2.5 Control Loop

Before discussing the control design, it is important to summarize the modeling of G_{Gimbal} , which represents the complete mechatronic system to be controlled. The model G_{Gimbal} includes the following components:

- **ESC and Motor:** The ESC saturates the maximum requested torque T_{req} to a value derived from the motor's data sheet to prevent damage to the motor. The delay in the ESC is modeled as a first-order transfer function with a time constant T , as shown in Equation 2.14.
- **Friction:** Friction is modeled as a function of angular velocity with parameters a and b , as shown in Equation 2.13. Here, a describes the Viscous friction, which relates angular velocity to friction torque, while b describes Coulomb friction.
- **Equations of Motion:** The relationship between the effective torque T_{eff} and the angle is described by the equations of motion. The time-domain model is shown in Equation 2.12. By applying the Laplace transform, the model becomes:

$$\begin{cases} \alpha(s) = \frac{1}{J_{total} s^2} T_{eff}(s) \\ \beta(s) = \frac{1}{J_{\beta} s^2} T_{eff}(s) \end{cases} \quad (2.47)$$

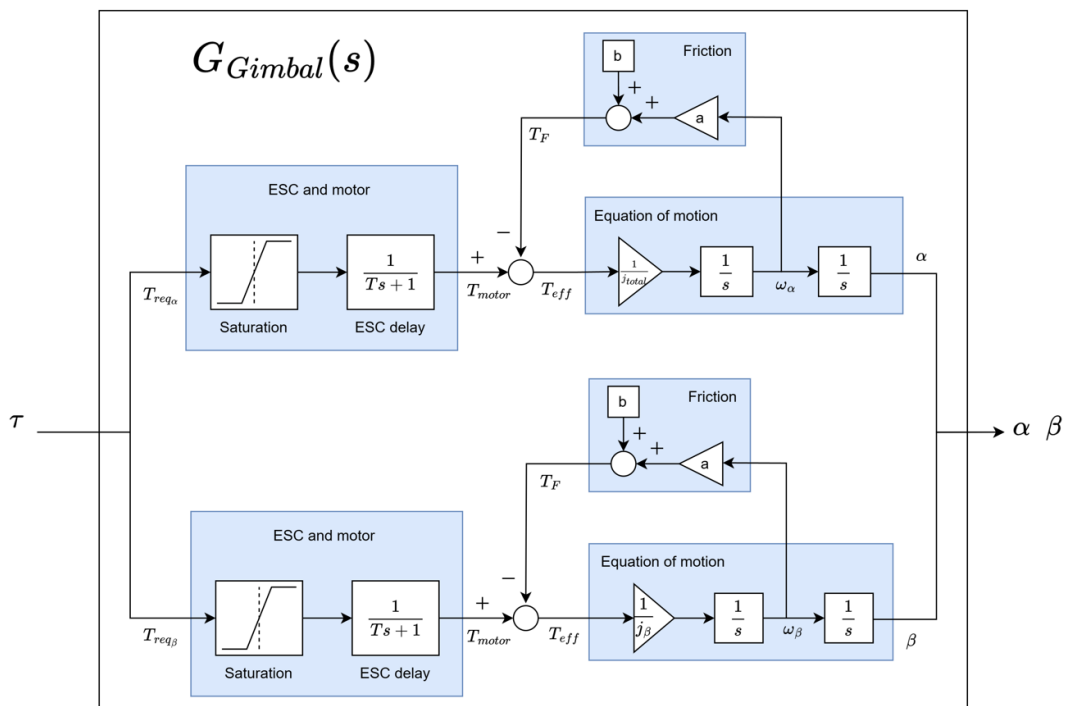


Figure 2.15: The simulation model representing the dynamics from requested torques to angles. The model includes two uncoupled systems, one for each axis.

The transfer function $G_{Gimbal}(s)$, which describes the relationship between the re-

requested torque and the resulting angles, can be derived as follows:

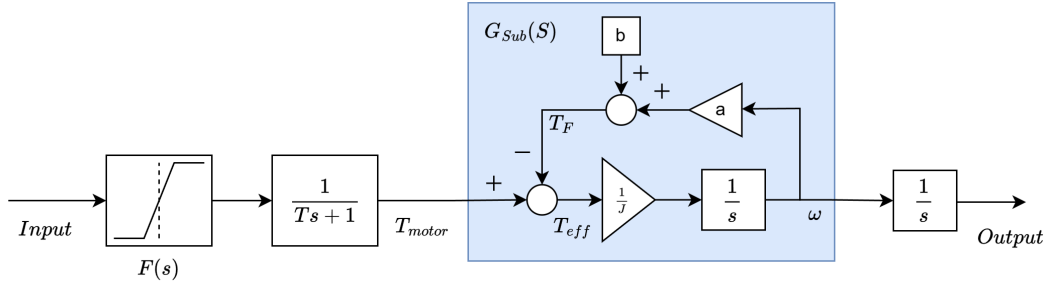


Figure 2.16: A simplified version of the block diagram shown in Figure 2.15, focusing on the relationship between a single requested torque and its resulting angle. For simplicity, the loop transfer function depicted in blue is denoted as $G_{sub}(s)$, while the transfer function for the saturation is denoted as $F(s)$.

The output angle can be expressed as:

$$\text{Output} = \frac{1}{s} \cdot G_{sub}(s) \cdot T_{motor}$$

where:

$$G_{sub}(s) \cdot T_{motor} = \frac{-b}{a + Js} + \frac{1}{a + Js} \cdot T_{motor}$$

Substituting this expression into the output equation gives:

$$\text{Output} = \frac{1}{s} \cdot \left(\frac{-b}{a + Js} + \frac{1}{a + Js} \cdot T_{motor} \right)$$

The motor torque T_{motor} is given by:

$$T_{motor} = \frac{1}{Ts + 1} \cdot F(s) \cdot \text{Input}$$

Substituting this into the output equation results in:

$$\text{Output} = \frac{1}{s} \cdot \left\{ \frac{-b}{a + Js} + \frac{1}{a + Js} \cdot \left(\frac{1}{Ts + 1} \cdot F(s) \cdot \text{Input} \right) \right\}$$

Simplifying the equation further yields:

$$\text{Output} = \frac{-1}{Js^2 + as} \cdot b + \frac{1}{TJs^3 + (Ta + J)s^2 + as} F(s) \cdot \text{Input} \quad (2.48)$$

Experimental results indicate that the friction parameters a and b are very small, and the time constant T of the ESC delay is considered to be negligible, as shown in Figure 2.4. Therefore, for control design purposes, it is possible to make the following assumptions:

- The system does not reach saturation; therefore, $F(s) = 1$.
- The ESC response is very fast; hence, $T \rightarrow 0$.
- The impact of friction on the system is negligible; thus, $a, b \rightarrow 0$.

With these assumptions, Equation 2.48 is simplified to:

$$\frac{\text{Output}}{\text{Input}} = \frac{1}{Js^2}. \quad (2.49)$$

Thus, the simplified transfer function $G_{Gimbal}(s)$, used for control design, becomes:

$$G_{Gimbal}(s) = \begin{cases} \frac{1}{J_{total} s^2}, & \text{Outer gimbal,} \\ \frac{1}{J_{\beta} s^2}, & \text{Inner gimbal.} \end{cases} \quad (2.50)$$

The block diagrams in Figures 2.15 and 2.16 present a simplified representation of how friction is modeled. However, the actual behavior of the friction torque is nonlinear.

The sign of the friction torque, depends on the sign of ω because it always opposes the direction of rotation. Additionally, the friction torque acts as a reaction torque. In other words, if the motor torque is less than the friction torque, no effective torque will be applied to the system, $T_{motor} - T_f = 0$.

$$T_f(\omega) = \begin{cases} (a|\omega| + b), & \text{when } \omega > 0, \\ -(a|\omega| + b), & \text{when } \omega < 0, \\ T_{motor}, & \text{when } |T_{motor}| < a|\omega| + b. \end{cases} \quad (2.51)$$

The controller's primary objective is to ensure system stability and following a reference point. According to control theory, a feedback system is stable if its poles are located in the left half-plane (LHP) of the complex plane [12]. Moving the poles further to the left increases the system's responsiveness and speed, but at the cost of higher control effort. This can lead to saturation due to torque requests exceeding the motor's maximum limits.

By tuning the controller, the poles of the closed loop system will be adjusted. Placing the poles at the origin of the complex plane results in marginal stability. In this case, the system may oscillate indefinitely without damping, which is undesirable for most practical applications. Therefore, careful pole placement is crucial to balance stability, responsiveness, and control signal limitations.

2.5.1 PID Controllers

A PID controller is a widely used type of closed-loop controller [8]. It consists of three components:

- **Proportional (P):** This component applies a control action proportional to the error signal. It helps reducing the error by providing a direct response to the difference between the desired and actual values.
- **Derivative (D):** This component corresponds to the derivative of the error signal. It anticipates changes in the error and improves the system's responsiveness, helping to make its reaction quicker and more stable.
- **Integral (I):** This component is proportional to the accumulation of the error over time. It addresses steady-state errors by ensuring that any persistent deviation from the desired value is eliminated.

The combination of these three terms allows the PID controller to provide a balance between responsiveness, stability, and steady-state accuracy, making it effective for a wide range of control applications.

2.5.1.1 P Controller

When the system is controlled by a proportional (P) controller, the open-loop transfer function $H(s)$ becomes:

$$H(s) = G_{\text{Controller}} \cdot G_{\text{Gimbal}} = k_p \cdot \frac{1}{Js^2} \quad (2.52)$$

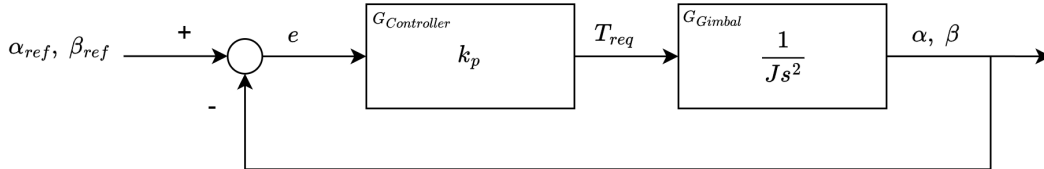


Figure 2.17: Closed-loop control system for G_{Gimbal} using a P-controller with gain k_p .

This leads to a closed-loop transfer function $L(s)$ as follows:

$$L(s) = \frac{H(s)}{1 + H(s)} = \frac{k_p \cdot \frac{1}{Js^2}}{1 + k_p \cdot \frac{1}{Js^2}} = \frac{k_p}{Js^2 + k_p} \quad (2.53)$$

The stability of the system with a proportional gain k_p can be determined by analyzing the poles of the closed-loop transfer function. The poles are the roots for the characteristic equation, (CE), i.e., the denominator of $L(s)$:

$$\text{Poles} = \pm \sqrt{\frac{-k_p}{J}} \quad (2.54)$$

From this equation, the stability depends on the value of k_p :

- If $k_p > 0$, the poles are purely imaginary and lie on the imaginary axis of the complex plane. This results in a marginally stable system.
- If $k_p < 0$, one pole is in the Left Half-Plane (LHP), and the other is in the Right Half-Plane (RHP). This results in an unstable system.

2.5.1.2 D Controller

When the system is controlled by a derivative (D) controller, the open-loop transfer function $H(s)$ is:

$$H(s) = G_{\text{Controller}} \cdot G_{\text{Gimbal}} = s k_d \cdot \frac{1}{Js^2} \quad (2.55)$$

Usually, when using a D-controller, a low-pass filter is added to the error signal. This prevents the derivative action from amplifying high-frequency noise, which could cause instability or erratic behavior. However, for stability analysis, the filter can be excluded.

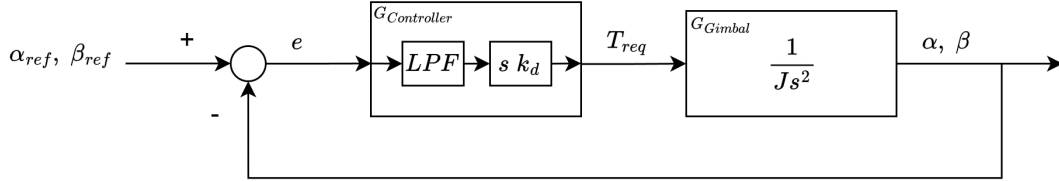


Figure 2.18: Closed-loop control system for G_{Gimbal} using a D-controller with gain k_d .

The resulting closed-loop transfer function $L(s)$ is given by:

$$L(s) = \frac{H(s)}{1 + H(s)} = \frac{s k_d \cdot \frac{1}{Js^2}}{1 + s k_d \cdot \frac{1}{Js^2}} = \frac{k_d}{Js + k_d} \quad (2.56)$$

To determine the system's stability with the derivative gain k_d , the poles of the closed-loop transfer function are analyzed. The poles are the roots for the CE:

$$\text{Poles} = \frac{-k_d}{J} \quad (2.57)$$

- If $k_d > 0$, the pole lies in the Left Half-Plane (LHP), resulting in a stable system.
- If $k_d < 0$, the pole lies in the Right Half-Plane (RHP), resulting in an unstable system.

2.5.1.3 PD Controller

By combining P and D components in a PD controller, the open-loop transfer function $H(s)$ becomes:

$$H(s) = G_{\text{Controller}} \cdot G_{\text{Gimbal}} = (k_p + s k_d) \cdot \frac{1}{Js^2} \quad (2.58)$$

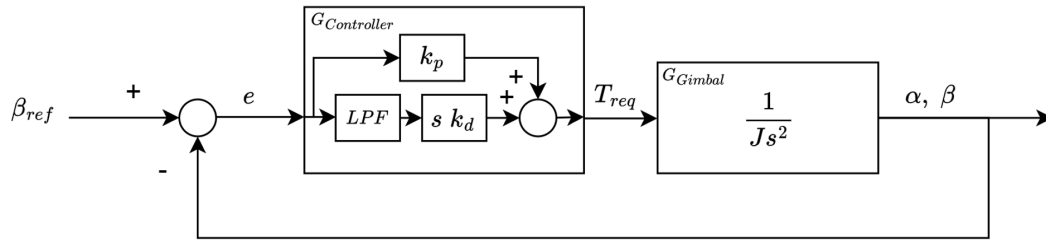


Figure 2.19: Closed-loop control system for G_{Gimbal} using a PD-controller.

The resulting closed-loop transfer function $L(s)$ is:

$$L(s) = \frac{k_d s + k_p}{J s^2 + k_d s + k_p} \quad (2.59)$$

The poles of the closed-loop system are determined by the roots of the CE:

$$\text{Poles} = \begin{cases} p_1 = -\frac{k_d}{2J} + \sqrt{\frac{k_d^2}{4J^2} - \frac{k_p}{J}} \\ p_2 = -\frac{k_d}{2J} - \sqrt{\frac{k_d^2}{4J^2} - \frac{k_p}{J}} \end{cases} \quad (2.60)$$

To simplify the tuning process and ensuring stability, a constraint can be imposed to place the poles on top of each other. This means:

$$p_1 = p_2 = -\frac{k_d}{2J} \Leftrightarrow k_p = \frac{k_d^2}{4J} \quad (2.61)$$

By selecting k_p and k_d such that they satisfy Equation 2.61, the system will be critically damped with both poles located in LHP, ensuring stability.

2.5.1.4 PID Controller

Introducing only an integral (I) controller to the system results in the closed-loop transfer function:

$$L(s) = \frac{\frac{k_i}{s} \cdot \frac{1}{J s^2}}{1 + \frac{k_i}{s} \cdot \frac{1}{J s^2}} = \frac{k_i}{J s^3 + k_i} \quad (2.62)$$

The poles of this system are given by the following:

$$\text{Poles} = \sqrt[3]{\frac{-k_i}{J}} \quad (2.63)$$

This results in three poles:

2. Theory

- If $k_i < 0$, there is one real pole in the right half-plane (RHP) and two complex poles in the left half-plane (LHP).
- If $k_i > 0$, there is one real pole in the LHP and two complex poles in the RHP.

Thus, the system is unstable for any value of k_i .

Adding both proportional and integral components (PI controller) leads to the following closed-loop transfer function:

$$L(s) = \frac{k_p s + k_i}{J s^3 + k_p s + k_i}. \quad (2.64)$$

Determining the poles directly is not straightforward; instead, the Routh-Hurwitz criterion is applied to identify the conditions for stability. The characteristic equation (CE) is as follows:

$$CE = J s^3 + 0 s^2 + k_p s + k_i. \quad (2.65)$$

According to the Routh-Hurwitz criterion, a necessary condition for stability is that all coefficients of the characteristic equation must be strictly positive [12]. However, the coefficient of s^2 is zero, violating this condition. Therefore, the system is not stable for any combination of k_p and k_i .

Including all components in a PID controller results in the following closed-loop transfer function:

$$L(s) = \frac{k_d s^2 + k_p s + k_i}{J s^3 + k_d s^2 + k_p s + k_i}. \quad (2.66)$$

The characteristic equation (CE) is:

$$CE = J s^3 + k_d s^2 + k_p s + k_i. \quad (2.67)$$

Constructing the Routh-Hurwitz table for this characteristic equation:

$$\begin{array}{c|ccc} s^3 & J & k_p & 0 \\ s^2 & k_d & k_i & 0 \\ s^1 & \frac{k_d k_p - J k_i}{k_d} & 0 & \\ s^0 & k_i & & \end{array}$$

For stability, the first column of the Routh-Hurwitz table must be strictly positive. This results in the following conditions:

- $k_d > 0$ and $k_i > 0$

- $k_p > J \frac{k_i}{k_d}$

These conditions ensure that all the roots of the characteristic equation have negative real parts, making the system stable.

2.5.2 LQR Controllers

Linear Quadratic Regulator (LQR) is an optimal full-state feedback linear controller. An LQR controller is considered optimal because it minimizes a trade-off between the control error and the control effort by using weight matrices [9], Q_1 for the states and Q_2 for the inputs. These weight matrices define the relative importance of the states and the control effort in the system. The optimization problem is mathematically expressed as [9]:

$$\min \left(\|e\|_{Q_1}^2 + \|u\|_{Q_2}^2 \right), \quad (2.68)$$

where u represents the control action. In this report, the control action corresponds to the requested torque, T_{req} .

In a proportional controller (e.g., Figure 2.17), the control signal is expressed as:

$$T_{\text{req}} = k_p \cdot e,$$

where e is the difference between the reference and estimated angles.

In contrast, a full-state feedback controller uses a vector of state errors. For the gimbal system, the states include angles and their derivatives, resulting in a richer feedback mechanism.

To design the LQR controller, the system dynamics is usually expressed in state-space form:

$$\begin{cases} \dot{x}(t) = A x(t) + B u(t), \\ y(t) = C x(t), \end{cases} \quad (2.69)$$

where $x(t)$ is the state vector, $u(t)$ is the input vector, and $y(t)$ is the output vector.

The controller gain matrix, k_{LQR} , that satisfies the optimization criteria in Equation 2.68, is given by [9]:

$$k_{\text{LQR}} = Q_2^{-1} B^T S, \quad (2.70)$$

where S is a symmetric semipositive matrix obtained by solving the Algebraic Riccati Equation (ARE) [9]:

$$A^T S + S A + C^T Q_1 C - S B Q_2^{-1} B^T S = 0. \quad (2.71)$$

Using Equation 2.70, the resulting k_{LQR} guarantees stability with all poles in the stability region [9]. These poles can be computed as follows:

$$\text{Poles} = \text{eig}(A - B k_{\text{LQR}}).$$

2.5.2.1 LQR model

The system to be controlled is described in the Laplace domain (see Equation 2.50). By applying the inverse Laplace transform, the original variables can be expressed as:

$$\begin{cases} T_{\text{req}\alpha} = J_{\text{total}} \cdot \dot{\omega}_\alpha, \\ \omega_\alpha = \dot{\alpha}, \\ T_{\text{req}\beta} = J_{\text{total}} \cdot \dot{\omega}_\beta, \\ \omega_\beta = \dot{\beta}. \end{cases} \quad (2.72)$$

Let x represent the state vector, which includes $[\alpha, \omega_\alpha, \beta, \omega_\beta]'$. The input vector, u , includes $[T_{\text{req}\alpha}, T_{\text{req}\beta}]'$. The state-space model can then be written as:

$$\dot{x} = Ax + Bu, \quad (2.73)$$

where:

$$\underbrace{\begin{bmatrix} \dot{\alpha} \\ \dot{\omega}_\alpha \\ \dot{\beta} \\ \dot{\omega}_\beta \end{bmatrix}}_x = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \alpha \\ \omega_\alpha \\ \beta \\ \omega_\beta \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{1}{J_{\text{total}}} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_\beta} \end{bmatrix}}_B \underbrace{\begin{bmatrix} T_{\text{req}\alpha} \\ T_{\text{req}\beta} \end{bmatrix}}_u. \quad (2.74)$$

2.5.2.2 LQI model

An LQR controller with integral action is referred to as LQI. The LQI controller can eliminate the steady-state error over time. By adding additional integrating states, the state-space model is extended to include the new integral states.

The integral state is a variable that accumulates the error over time:

$$\begin{cases} I_\alpha = \int (r_\alpha - \alpha) dt \\ I_\beta = \int (r_\beta - \beta) dt \end{cases} \Rightarrow \begin{cases} \dot{I}_\alpha = (r_\alpha - \alpha) \\ \dot{I}_\beta = (r_\beta - \beta) \end{cases} \quad (2.75)$$

the extended state-space model will be:

$$\underbrace{\begin{bmatrix} \dot{\alpha} \\ \dot{\omega}_\alpha \\ \dot{I}_\alpha \\ \dot{\beta} \\ \dot{\omega}_\beta \\ \dot{I}_\beta \end{bmatrix}}_{x_e} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}}_{A_e} \underbrace{\begin{bmatrix} \alpha \\ \omega_\alpha \\ I_\alpha \\ \beta \\ \omega_\beta \\ I_\beta \end{bmatrix}}_{x_e} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{1}{J_{\text{total}}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_\beta} \\ 0 & 0 \end{bmatrix}}_{B_e} \underbrace{\begin{bmatrix} T_{\text{req}_\alpha} \\ T_{\text{req}_\beta} \end{bmatrix}}_u + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_\alpha \\ r_\beta \end{bmatrix} \quad (2.76)$$

2.5.2.3 Tuning LQR

The tuning of an LQR controller is achieved by adjusting the elements of the weighting matrices, Q_1 and Q_2 . These matrices serve as design parameters that balance the trade-off between control accuracy and control effort [9].

- Q_1 is a square matrix with a size equal to the number of states in the system.
- Q_2 is a square matrix with a size equal to the number of inputs.

A common and practical approach to tuning is to assume that cross-coupling penalties are zero [9], as long as the physical system does not impose explicit constraints (for example, limited power availability for simultaneously driving multiple actuators). In most cases, there is no significant cross-coupling, so both matrices can be assumed diagonal.

Each diagonal element in Q_1 corresponds to a penalty for a specific state, while each diagonal element in Q_2 corresponds to a penalty for a specific input. Increasing a particular diagonal element in Q_1 prioritizes minimizing the error in the corresponding state. This results in a controller that emphasizes accuracy in that state but may require higher control effort or compromise the performance of other states. Increasing a particular diagonal element in Q_2 penalizes the corresponding control input, effectively making the input more "expensive". This leads to a slower system response, as the controller reduces its reliance on that input to avoid high control effort.

By systematically adjusting these diagonal elements, the LQR controller can be tuned to achieve a desired balance between performance and control effort for the specific application.

3

Methodology

This chapter presents the methodology employed for developing and validating the two-axis gimbal system. It begins with a comprehensive literature review that establishes the theoretical and technical foundation for the research. The discussion then transitions to a detailed overview of the simulation environment, focusing on the tools and techniques used to model, simulate, and validate the system effectively.

The chapter further delves into the modeling of the physical system using Simscape, leveraging an acausal approach to accurately capture the nonlinear dynamics of the gimbal system. Following this, the design of a rigorous validation trajectory is described, simulating diverse real-world road conditions and challenges to thoroughly evaluate the control system's performance.

Lastly, the chapter highlights the development of a 3D visualization environment, which provides an intuitive and immersive representation of the system's behavior. This visualization enhances the validation process by offering a clear and dynamic perspective of how the system operates in real-world scenarios.

3.1 Literature Study

This research relied exclusively on published articles from reputable journals, conference proceedings, and authoritative books as reference materials. Relevant literature was identified through a systematic search for key topics directly related to the study's objectives. Particular emphasis was placed on selecting works that provided theoretical and technical foundations for the modeling, simulation, and validation processes.

The selected references played a critical role in guiding the research methodology, offering insights into best practices, established theories, and recent advancements in the field. This comprehensive literature review ensured that the study was grounded in a strong academic and technical framework, aligning it with state-of-the-art research.

3.2 Simulation Environment

To model the physical system of the two-axis gimbal and to design and implement the entire control loop, a robust simulation environment was essential. MATLAB, in conjunction with Simulink, was selected as the software suite to achieve these goals. The system parameters and required inputs, such as the fixed satellite position and test scenarios defining the vehicle's orientation and position over time, were created in MATLAB. Since both MATLAB and Simulink are products of MathWorks, the seamless integration between the two proved highly advantageous, enabling efficient modeling and simulation of the system.

Simulink provided an ideal platform for designing, simulating, and validating physical systems, including control loops, due to its extensive library of built-in functions and its ability to integrate MATLAB scripts. The built-in visualization tools, such as Simulink Scopes and MATLAB plotting functions, were invaluable for monitoring system behavior throughout the research. These tools facilitated iterative refinement and optimization of the models and controllers, ensuring that the simulation environment evolved alongside the research.

Figure 3.1 illustrates the complete simulation model implemented in Simulink, showcasing the integration of system components for modeling, control design, and validation. The mechatronic system, represented by $G_{Gimbal}(s)$, incorporates the equations of motion for the rotational components, friction modeling, and the dynamics of the motor driver (ESC). In addition, the simulation includes sensor models, enabling validation of the entire system in a virtual environment. This holistic approach allowed the simulation to accurately mimic real-world conditions and provided a reliable framework for testing and analysis.

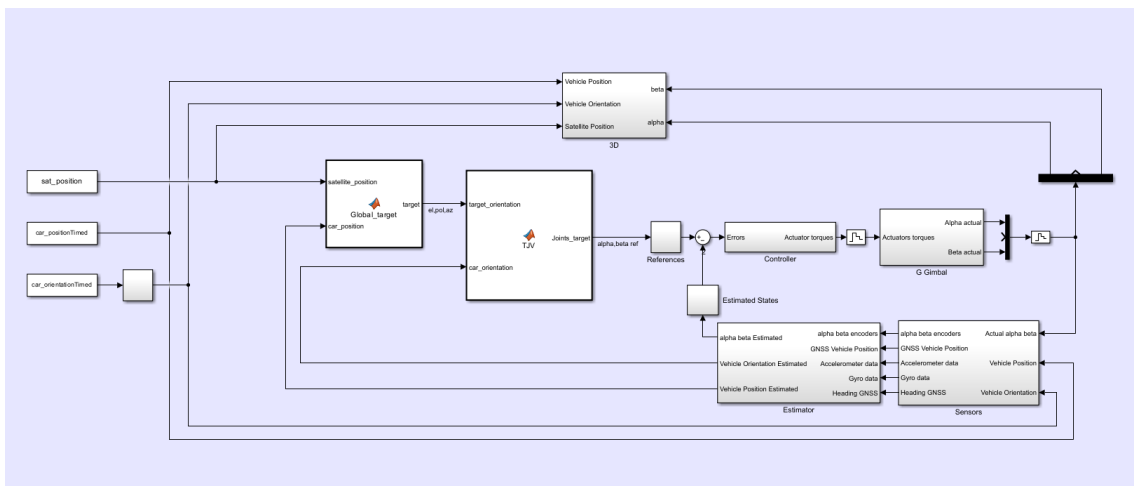


Figure 3.1: Overview of the simulation system implemented in Simulink. This diagram mirrors the conceptual block diagram shown in figure 2.1.

The simulation environment takes a predefined test scenario as input, expressed in terms of action points. A test scenario specifies the satellite's fixed position, the vehicle's position, and its orientation over time. Using these inputs, the global target parameters, elevation (el), polarization (pol), and azimuth (az), are computed via a function called "Global Target".

With the global target angles and the vehicle's orientation as inputs, the joint angles α and β , which correspond to the actuator states, are calculated. These angles define the positioning of the system's actuators. Meanwhile, the vehicle states, including its orientation, are estimated using sensor signals generated within the simulation environment through a dedicated block called the "Estimator".

Based on the estimated states derived from these sensor signals, a feedback control configuration computes and regulates the torques of each motor to achieve precise system operation.

Additionally, the simulation environment includes a comprehensive 3D visualization, making it easier to observe the system's behavior and validate its performance. The block labeled "3D" processes signals from the simulation environment to generate this visualization, leveraging Simscape Multibody for real-time rendering. This block is further detailed in Section 3.5, where its role in the visualization of the system is thoroughly explained.

As the research progressed, the system grew increasingly complex. To maintain clarity and facilitate understanding, Simulink's tools were utilized to partition the system into manageable subsystems. This modular approach not only enhanced the organization but also streamlined the development process, making the simulation environment more efficient and user-friendly.

3.3 System Modeling Using Simscape

Simscape, an acausal and equation-based modeling language, was utilized to model the nonlinear dynamics of the two-axis gimbal system. Unlike traditional signal-flow modeling approaches, Simscape allows users to define system dynamics through equations, enabling the direct modeling of physical relationships. This feature makes it particularly effective for solving complex nonlinear systems, such as the two-axis gimbal.

The use of Simscape was crucial in the early stages of the research, as it facilitated an accurate representation of the nonlinear dynamics required to understand the system's behavior. Modeling the gimbal system by directly writing its equations of motion and solving them within the Simscape environment proved to be a more effective approach compared to signal-flow-based methods. This approach ensured that the physical system was represented with greater fidelity, providing valuable insights into the gimbal's dynamic response. After confirming that Equations 2.6 and 2.8 could be expressed as Equation 2.11, the two-axis gimbal dynamics were implemented directly in Simulink. This approach was chosen because the dynamics,

having been simplified, no longer required the use of Simscape.

3.4 Trajectory Design for Validation

Validating the performance of the feedback control system was a critical aspect of this research. To achieve this, a trajectory was designed and generated using MATLAB, simulating a variety of real-world road conditions. This trajectory was used to thoroughly test the controller and filtering algorithms under diverse and challenging scenarios.

The trajectory incorporated road segments with varying pitch, roll, and yaw angles, as well as challenging features such as speed bumps, roadside inclines, and gravel roads. The design of these road types was a collaborative effort, involving brainstorming sessions within the research team and consultations with SATCUBE supervisors. This ensured the trajectory met operational requirements while representing realistic driving conditions.

The trajectory was shaped like a rectangle, measuring 480 meters in length and 180 meters in width. It featured four 90-degree turns with varying radius, ranging from 30 meters to 50 meters, to evaluate the controller's ability to handle both sharp and smooth turns. Additionally, pitch and roll angles were designed to reflect realistic worst-case scenarios. The maximum pitch angle used in the trajectory was 20 degrees, based on industry benchmarks [10], while the maximum roll angle was set to 45 degrees. Although no specific source was available for the roll angle, it was chosen to be sufficiently high to represent a challenging scenario. Special features, including speed bumps, gravel roads, and roadside inclines, were designed to simulate worst-case conditions that a vehicle might encounter during operation. These features were introduced to assess how such challenges impact the performance of the control system. Additionally, the trajectory's key parameters were made adjustable, allowing the trajectory to be easily modified if needed. The resulting trajectory is illustrated in Figure 3.2.

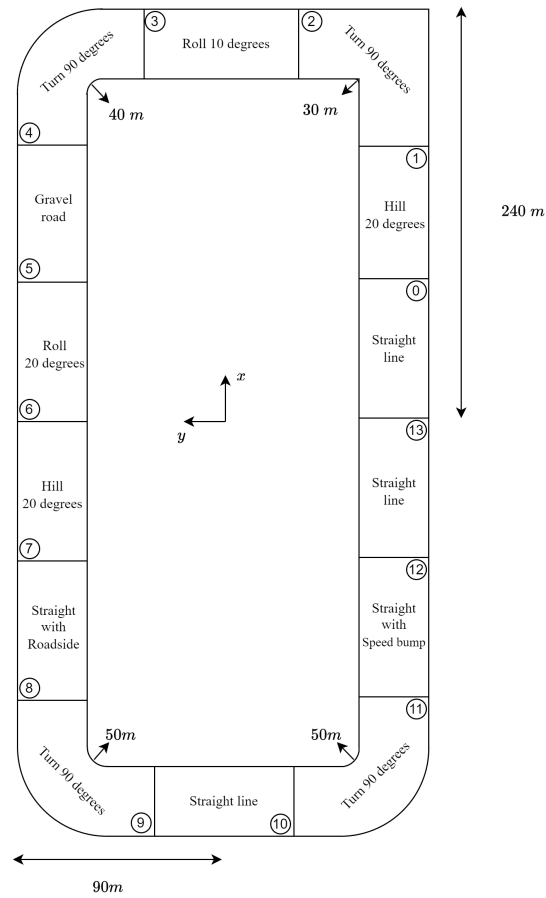


Figure 3.2: Trajectory created for validation of the feedback control system, featuring varying road conditions, including pitch, roll, speed bumps, roadside inclines, and gravel roads, with 90-degree turns of different radius.

A critical consideration during trajectory development was the vehicle's velocity, as it significantly influences the controller's response. To address this, the trajectory was parameterized in MATLAB, enabling dynamic adjustment of the vehicle's speed for each simulation run. The total duration of the run was subsequently calculated based on the selected velocity.

Additionally, to assess the performance of the filtering algorithm and evaluate the potential impact of gyroscope drift over extended periods, a parameter was introduced to define the number of rounds the vehicle should complete before finishing the trajectory. This feature provided a valuable means of validating the filtering algorithm's robustness and its ability to mitigate drift over time.

3.5 3D Visualization

To enhance the validation of the control system, more than numerical or graphical signal observations were required. A visual representation of the system's behavior provided an additional, intuitive layer of verification. To this end, a 3D simulation

environment was developed using CAD models imported into Simscape Multibody. This visualization features a vehicle with an antenna terminal mounted on its roof, the track the vehicle follows, and a satellite. A virtual line indicates the direction the antenna is pointing, providing a clear visual representation of the system's alignment.

The 3D visualization proved invaluable in verifying the controller's performance, particularly in relation to the kinematic behavior outlined in Section 2.2. It enabled the team to observe the vehicle navigating the trajectory while the gimbal system actively maintained alignment with the satellite, providing critical insights into the system's real-world operation.

To safeguard SATCUBE's intellectual property, the antenna system's 3D model was intentionally simplified. Its actual dynamics and detailed geometry were excluded while still retaining the essential features needed for functional validation. Despite this simplification, the visualization served as a highly effective tool for demonstrating system functionality and validating the control system's performance in a clear and comprehensible manner, as illustrated in Figure 3.3.

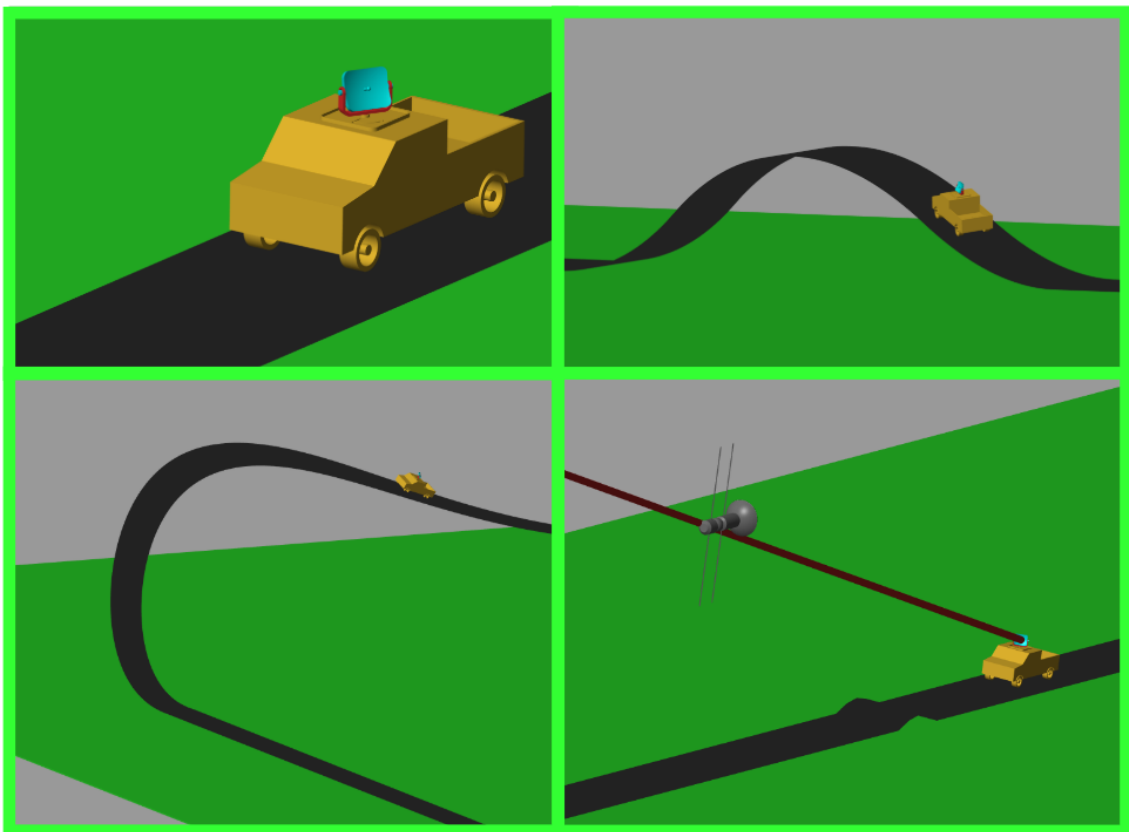


Figure 3.3: The 3D simulation environment created using Simscape Multibody. The vehicle, equipped with the antenna system mounted on its roof is shown tracking the trajectory. The antenna remains aligned with the satellite and the line of site represented by the red line.

4

Results

This chapter presents the outcomes of the study, encompassing the validation of the system modeling, sensor data processing, state estimation, and controller design evaluation. Through comprehensive simulations, the performance of the proposed approaches is analyzed and illustrated using relevant graphs and figures. These results validate the accuracy of the sensor models and demonstrate the effectiveness of the processing and estimation methods, including complementary and Kalman filtering techniques. Additionally, the controller design strategies, from PD and PID controllers to LQR and extended LQR approaches, are evaluated.

Furthermore, this chapter provides the tuning parameters and some specific values for critical components that correspond to the physical prototype and hardware utilized by SATCUBE. These details bridge the gap between simulation and real-world implementation, ensuring the presented results align closely with the capabilities and constraints of the actual system.

4.1 Modeling Results

This section evaluates the impact of several assumptions made in the theoretical analysis to simplify the model and enable controller design. Specifically, the evaluation focuses on the linear equation of motion and the transfer function that represents the complete mechatronic system, including the ESC, friction, and the equation of motion.

4.1.1 The Linear Equations of Motion

Using the equation shown in 2.11, the motion of the outer gimbal is nonlinear due to the variable $J_{total_{zz}}$, which includes $J_{\beta\alpha}$, a nonlinear function of the state β as shown in 2.10. Therefore, an assumption was made to simplify the system and derive the linear equation in 2.12. The results of simulating both the nonlinear model and the linear model are presented in the figure below.

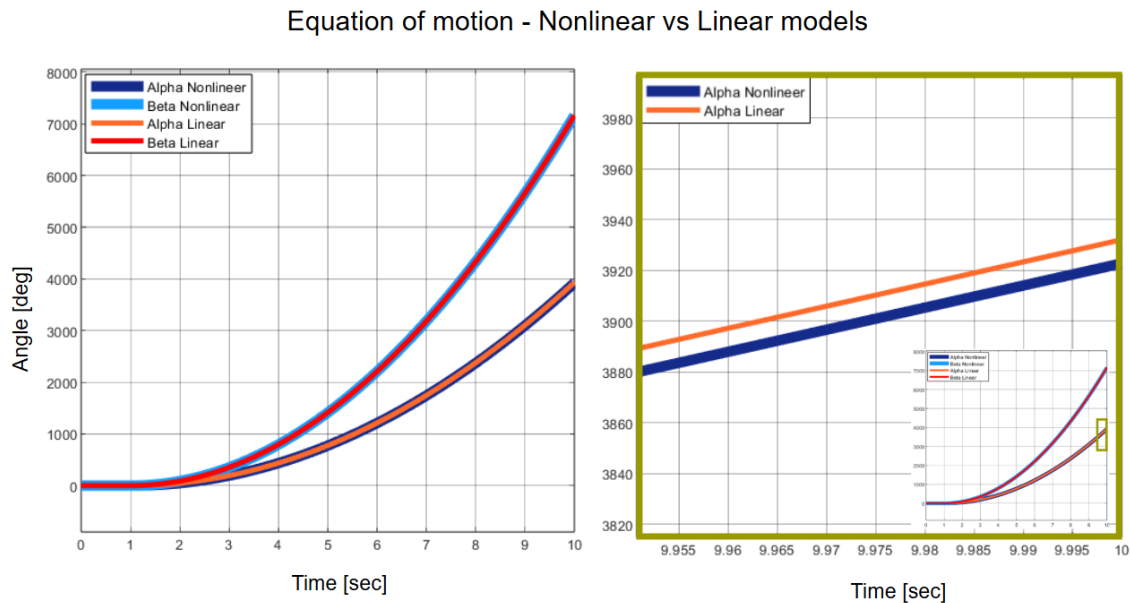


Figure 4.1: The graph on the right is a zoomed-in view of the left graph. A simulation was performed for both the nonlinear and linear models in the simulation environment. The models were subjected to torque inputs at $t = 1$ second, with $T_\alpha = 0.3$ N and $T_\beta = 0.05$ N. For the inner gimbal, the β angles showed complete overlap between the two models. However, the α angles exhibited a slight deviation, reaching approximately 10 degrees after 10 seconds of simulation, as highlighted in the zoomed-in graph.

The β angles are identical in both models, as expected, because their expressions are the same in both cases. However, replacing $J_{total_{zz}}$ in the nonlinear model with $J_{\alpha_{zz}} + J_{\beta_{yy}}$ introduces a small, negligible difference, as shown in Figure 4.1.

4.1.2 The Simplified Gimbal Transfer Function

The gimbal transfer function is based on two uncoupled transfer functions: one for the inner gimbal and another for the outer gimbal. Each includes the *ESC and motor* transfer function, *Friction*, and *Equation of Motion*, as shown in Figure 2.15. By making the following assumptions, the complete transfer function 2.48 was simplified to the form shown in 2.50. The assumptions were:

- The torque does not reach the motors' saturation level.
- The time delay T in the ESC is negligible.
- The impact of friction on the system is negligible.

The motors were selected to meet the angular acceleration specifications of the product. Therefore, motor saturation is avoided under normal conditions. Furthermore, assuming that the controllers are carefully tuned, they will not request torque values exceeding the motors' saturation limits during operation.

The ESC was modeled as a first-order transfer function with a time constant T . From Figure 2.4, the time constant was estimated to be $T = 0.000684$ s, which is sufficiently small to be considered negligible.

The friction torque T_f was modeled as a linear function of angular velocity, expressed as $T_f = a\omega + b$. To determine the parameters a and b , experiments were conducted where the output torque of each motor was recorded while maintaining a constant angular velocity. For each motor, the experiment was repeated at two different constant angular velocities. The resulting linear equations from the recorded data provided the values of a and b for each gimbal:

$$\begin{cases} T_{f_\alpha}(\omega) = 0.0269 \omega + 0.2713 \\ T_{f_\beta}(\omega) = 0.0055 \omega + 0.0274 \end{cases} \quad (4.1)$$

As shown in Equation 4.1, the viscous friction coefficient (a) is very small, and its impact is negligible because ω is typically small in the intended use case of the system. In contrast, Coulomb friction, represented by the constant b , plays a more significant role, especially if the system were to be controlled using an open-loop approach. However, with an active feedback controller in place, this friction is effectively managed.

4.2 Sensor Model

The sensors were modeled in the simulation environment as explained in Section 2.1.4. This section provides the results of the sensor modeling and explains how the noise parameters were obtained.

The differential GNSS module was used to provide the position and the absolute heading angle. In the simulation, position measurements were also assumed to be ideal, as the error in position estimation was negligible. On the other hand, heading estimation is crucial, and its noise must be characterized. To achieve this, the differential GNSS module was mounted stationary, with a distance of 58 cm between its two antennas, outdoors for more than 3 hours, and its output data were recorded. The recorded heading angle estimations exhibited white Gaussian noise (WGN) with a mean value of 0 and a standard deviation of 1.0704 *deg*, calculated from the recorded data.

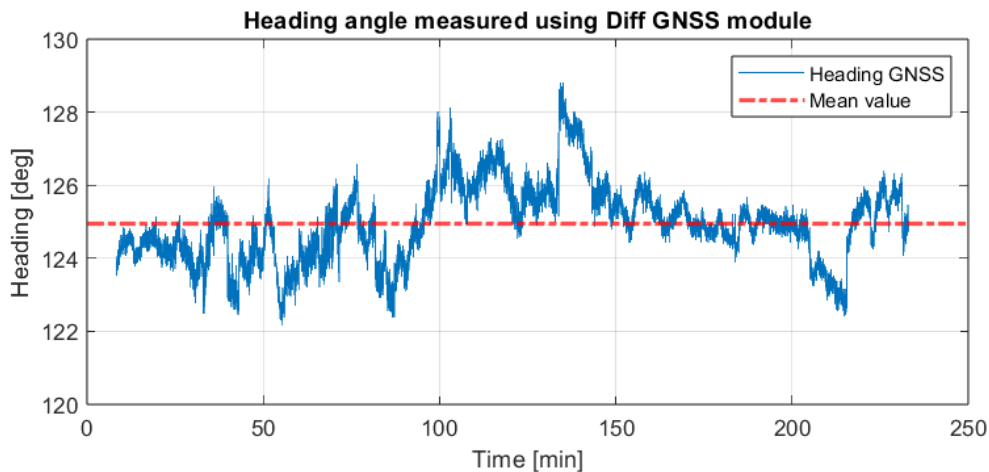


Figure 4.2: Heading angle estimations provided by the differential GNSS module recorded over an extended period with the module stationary.

A similar approach was used to characterize the WGN in the accelerometer and the random walk in the gyroscope. The IMU was mounted stationary on a test bench for 40 minutes, and its data were recorded. Since both the accelerometer and gyroscope have three axes, the standard deviation for each axis was calculated separately. The mean value of the three axes was then considered to represent the overall standard deviation for the accelerometer and gyroscope.

The standard deviation of the WGN in the accelerometer was calculated to be 0.0081 m/s^2 , while the standard deviation of the random walk in the gyroscope was 0.0272 deg/s .

Bias instability was also determined by recording the gyroscope data over approximately one hour. The first 30 seconds of the recorded data were used to calibrate the gyroscope, i.e., compensating for the initial bias error. The remaining data were integrated to convert the gyroscope's angular velocity data into angular displacement.

Since the sensor was stationary during recording, the angle should have remained at zero; however, it drifted by 9 deg/h. This drift value was considered as bias instability.

4.3 Sensor Processing and State Estimation

This section presents the results of estimating roll and pitch using only the accelerometer, roll, pitch, and yaw using the gyroscope, and heading estimation using the differential GNSS module. Additionally, results from more advanced filtering methods, such as complementary filters and Kalman filters, are provided.

By driving through the validation trajectory shown in Figure 3.2, the sensor model generated the corresponding sensor signals.

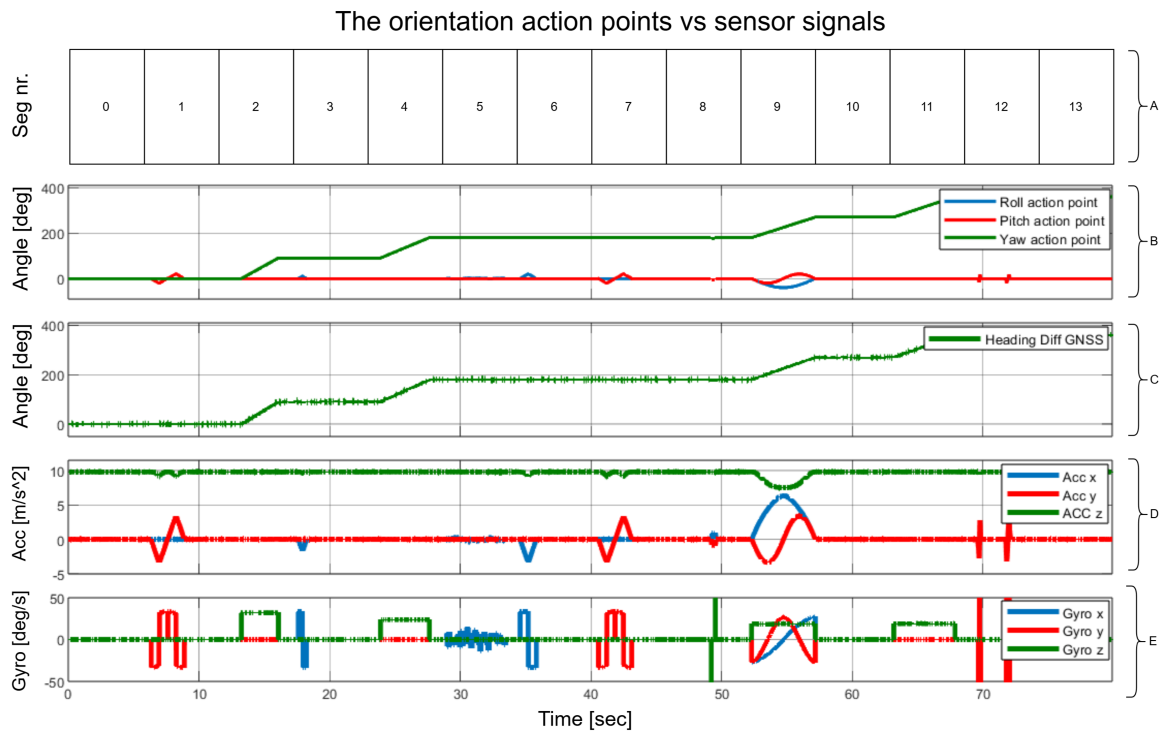


Figure 4.3: This figure presents sub-figures A to E, illustrating the sensor signals while driving along the validation trajectory and locking the outer gimbal to ensure that the accelerometer and gyroscope data represent the vehicle frame. Sub-figure A corresponds to the trajectory segment shown in Figure 3.2. Sub-figure B depicts the action points (true values) of the vehicle orientation in the simulation. Sub-figure C shows the estimated heading angle using the differential GNSS sensor model. Sub-figure D illustrates the acceleration signal generated by the accelerometer model. Sub-figure E represents the angular velocity signal generated by the gyroscope model.

4.3.1 Complementary Filter

The signals from the accelerometer were processed using Equations 2.28 and 2.29 to estimate the roll and pitch angles. However, the estimated signals included high-frequency noise, and the yaw angle could not be estimated using the accelerometer. Integrating the gyroscope signals provided smooth estimations but introduced drift over time due to bias instability.

The complementary filter combines the roll and pitch angles estimated from the accelerometer with those estimated from the gyroscope. The yaw angle estimation relies solely on the gyroscope.

In this research, the gyroscope angle estimation was found to be accurate due to integration of noise, with its primary limitation being drift. To mitigate this, the complementary filter parameter γ was set to a value close to 1, specifically $\gamma = 0.9$. This choice placed significant weight on the gyroscope measurements while leveraging the accelerometer to correct for drift. The value $\gamma = 0.9$ was selected to achieve a balance between noise and drift compensation, reducing drift while maintaining a smooth and reliable estimation.

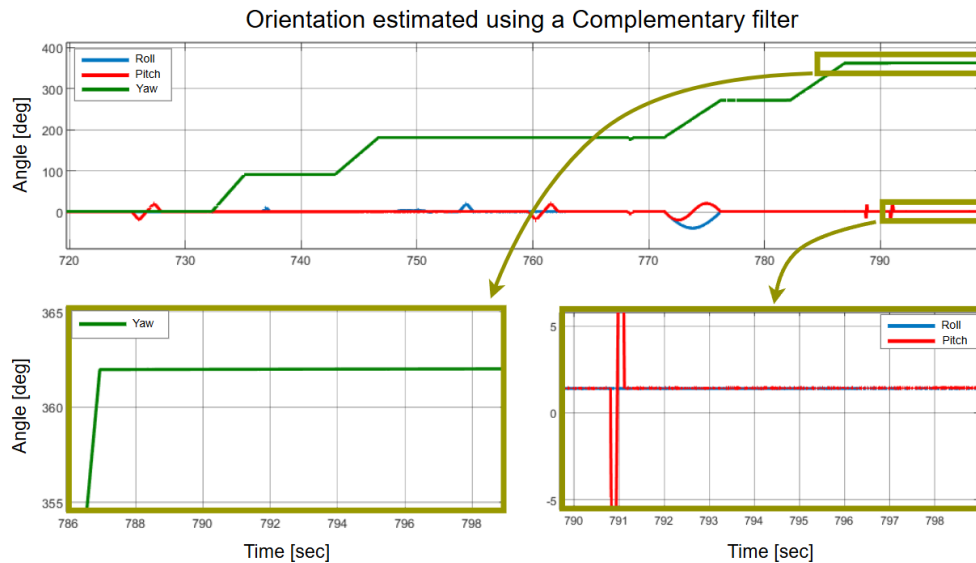


Figure 4.4: Orientation estimation using a complementary filter with a gain of 0.9. The figure shows the estimation After 10 laps around the validation trajectory. At the end of the trajectory, the orientation action points are $[roll, pitch, yaw] = [0, 0, 360]$ deg, while the estimation is $[1.40, 1.44, 362.01]$ deg indicating drift.

Figure 4.4 illustrates the drift in roll and pitch estimations, which reached approximately 1.42 degrees after 10 laps around the validation trajectory. The yaw angle, estimated via gyro integration, exhibited a drift of 2.01 degrees. It is evident that roll and pitch estimations benefit from the complementary filter's correction using the accelerometer, exhibiting less drift compared to the yaw estimation based solely on gyro integration.

4.3.2 Kalman Filter Based on IMU and Differential GNSS

A Kalman filter with a constant turning rate model was implemented, utilizing roll and pitch estimations from the accelerometer, heading estimations from the differential GNSS, and angular velocities for all angles from the gyroscope.

To tune the Kalman filter, the diagonal values of the process noise covariance matrix Q corresponding to each state were determined through trial and error. This iterative process ensured an optimal balance between trusting the process model and the sensor measurements, while adhering to the principle of appropriately weighting noisy sensors.

The tuned Q matrix is as follows:

$$Q = \text{diag}[10^{-8}, 10^{-3}, 10^{-8}, 10^{-3}, 10^{-8}, 10^{-3}]$$

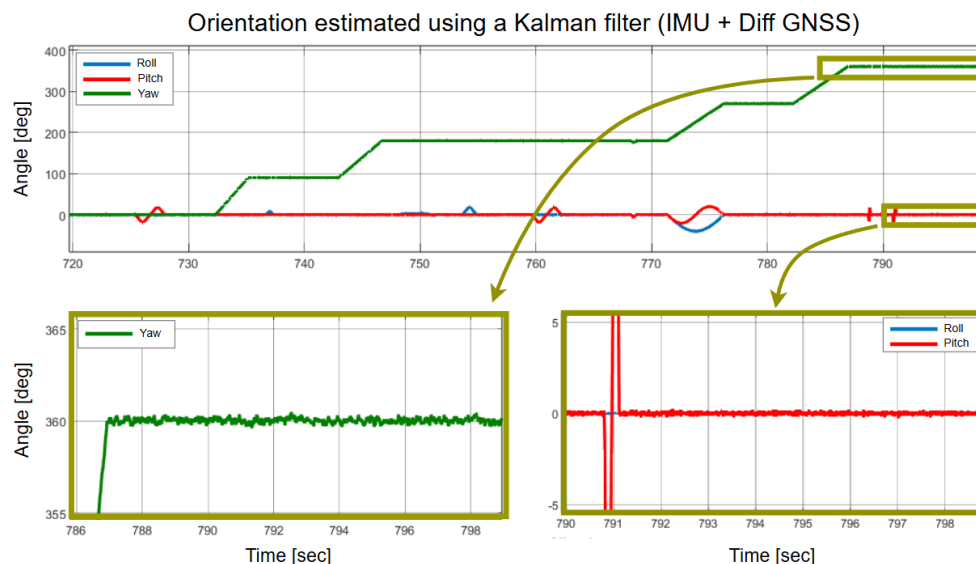


Figure 4.5: Orientation estimation using a Kalman filter based on IMU and differential GNSS.

The Kalman filter successfully compensated for the drift inherent in the gyroscope measurements. After completing 10 laps around the validation trajectory, the drift was reduced to 0.0 deg across all axes, as shown in Figure 4.5. Including the differential GNSS heading estimation introduced some noise to the yaw estimation. However, it effectively eliminated drift in the yaw axis.

Additionally, the Kalman filter is less responsive than the complementary filter due to dynamics in its model. Figure 4.6 illustrates the difference in behavior between the complementary filter and the Kalman filter.

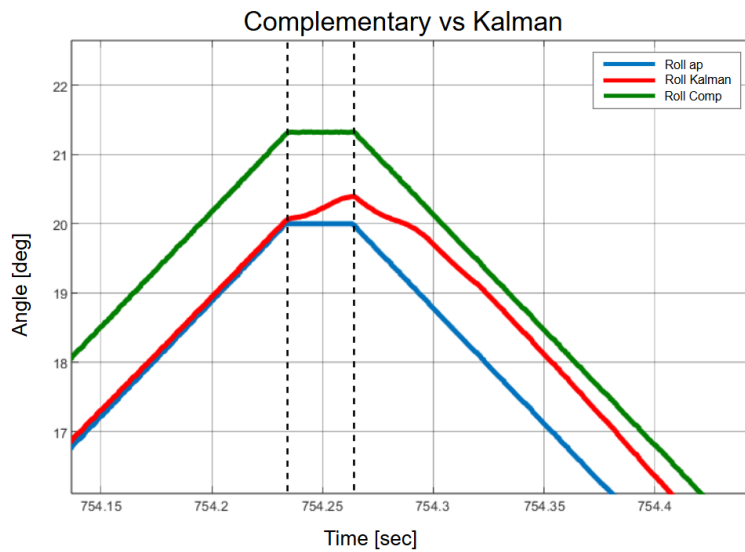


Figure 4.6: A comparison of the estimation performance between the complementary filter and the Kalman filter. The figure shows the estimation results after running 10 laps of the validation trajectory at segment 6, which simulates a roll change on a strait road. When the roll action point's rate changes, the complementary filter responds immediately, while the Kalman filter exhibits a slight delay before adjusting to the action point. However, the complementary filter shows a noticeable drift (offset) from the action point.

4.3.3 Kalman Filter Based Only on IMU

In the absence of heading information from the differential GNSS, a modified version of Kalman filter continued to perform similarly for roll and pitch estimations. However, the main difference lied in the yaw estimation, which in this case relied solely on the gyroscope. As a result, the yaw estimation experienced drift over time.

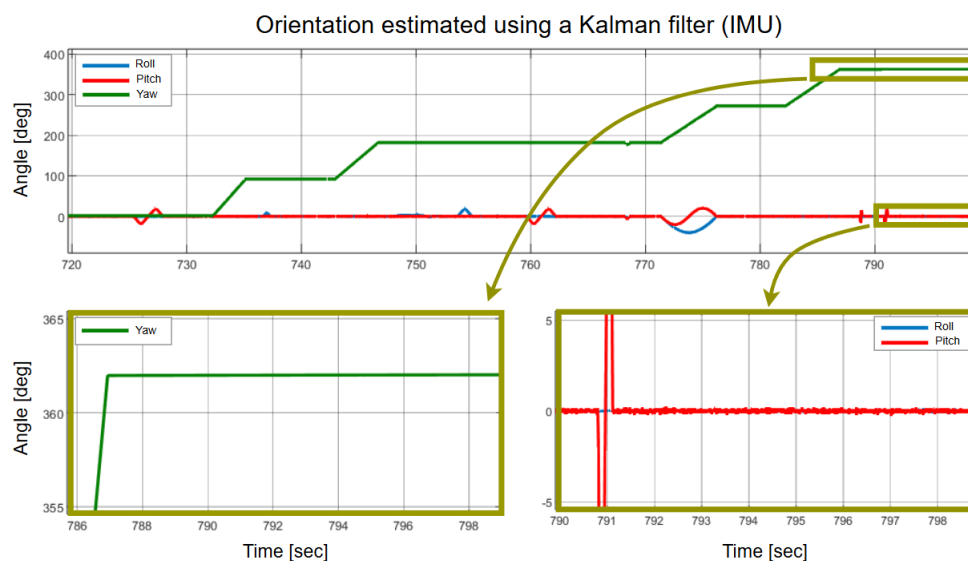


Figure 4.7: Orientation estimation using a Kalman filter based only on IMU.

The yaw angle drifted by approximately 2 degrees after running the simulation for 10 laps around the validation trajectory, whereas the roll and pitch estimations remained unaffected by the absence of heading information from the differential GNSS.

4.4 Controller Design

To evaluate the performance of different controllers in tracking a reference trajectory, an indicator I is defined. This indicator represents the accumulated control error over one lap of the validation trajectory. It is computed as follows:

$$I = \int_0^{\text{Lap time}} |e| dt \quad (4.2)$$

Controlling the system G_{gimbal} using a closed-loop approach with a D-controller results in a stable system, as shown in the poles equation 2.57. However, even though the system is stable, the controller cannot track the reference angles; it only responds to changes in the reference. In contrast, using a PD-controller ensures a stable system, as demonstrated in equation 2.61, and enables tracking of the reference angles.

As suggested in section 2.5.1.3, one tuning approach is to constrain the poles by placing them on top of each other. By choosing the poles at -7 for the outer gimbal and -6 for the inner gimbal, the controller parameters are calculated as:

$$\begin{cases} Poles_{Outer\ gimbal} = -7 \Leftrightarrow [K_p = 3.268, K_d = 0.934] \\ Poles_{Inner\ gimbal} = -6 \Leftrightarrow [K_p = 0.108, K_d = 0.036] \end{cases} \quad (4.3)$$

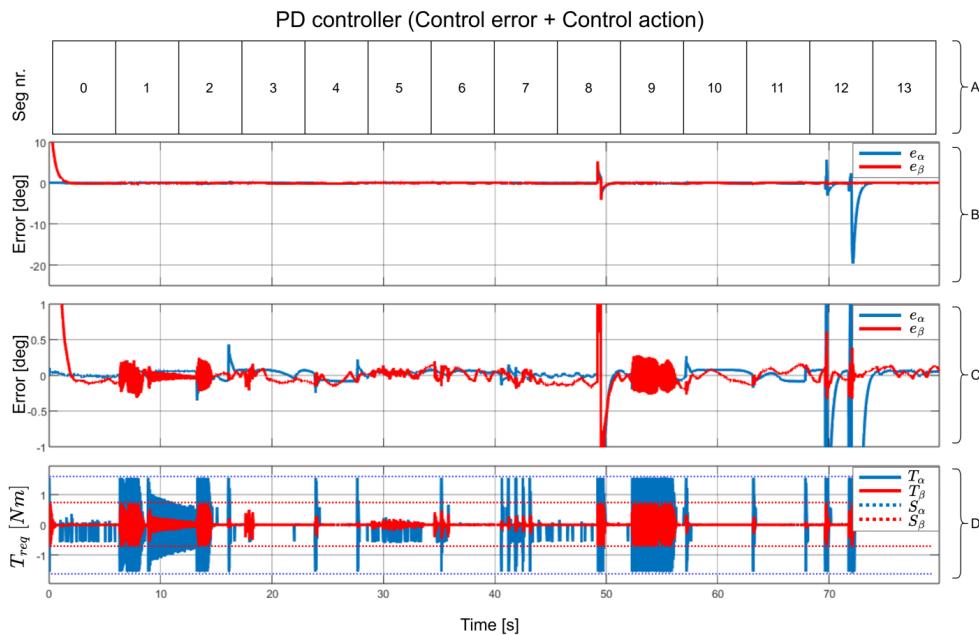


Figure 4.8: A PD controller is implemented using the parameter provided in equation 4.3, and the control error and the control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.

The maximum control error was -20 deg in the α angle while driving on trajectory segment 12, which corresponds to the speed bumps. Furthermore, the α motor reached saturation while driving on segments 1, 2, 4, 6, 7, 8, 9, 11, and 12, while the β motor reached saturation on segments 1, 2, 8, 9, and 12.

By evaluating Equation 4.2 in the simulation, the indicator values were found to be $I_\alpha = 15.13$ and $I_\beta = 13.87$.

A PID controller was also implemented, and the control parameters were selected through an iterative trial-and-error approach, ensuring compliance with the stability conditions outlined in section 2.5.1.4.

The chosen parameters are:

$$\begin{cases} [K_p = 0.8, K_i = 0.05, K_d = 0.05] & P = [-300.5598, -0.0399] \\ [K_p = 0.4, K_i = 0.02, K_d = 0.01] & P = [-749.7671, -0.0080] \end{cases} \quad (4.4)$$

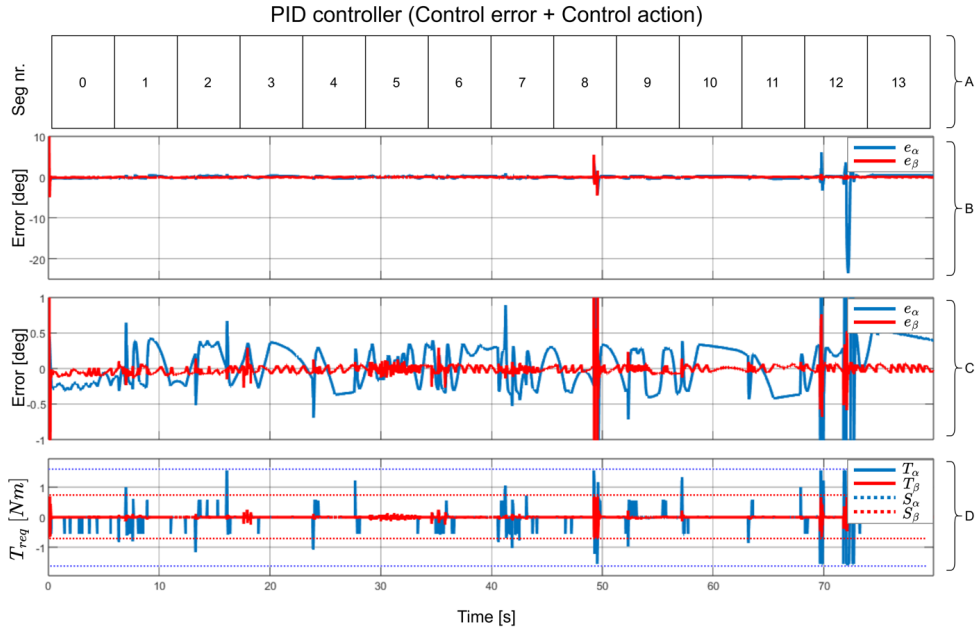


Figure 4.9: A PID controller is implemented using the parameters in equation 4.4, and the control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.

The motors rarely reached saturation; however, the control error in alpha was significantly larger compared to the PD controller case while the error in beta was smaller. The indicator values were $I_\alpha = 27.72$ and $I_\beta = 5.10$.

A more intuitive approach to designing and tuning the controller is to use an LQR controller, which employs weighting parameters for the states and the input. The state weighting matrix Q_1 and the input weighting matrix Q_2 were chosen through iterative trial and error and the chosen values are as follows:

$$\begin{cases} Q_1 = \text{diag} [2, 1, 2, 1] \\ Q_2 = \text{diag} [1, 300] \end{cases} \quad (4.5)$$

This results in the feedback gain matrix and the system poles as follows:

$$K_{LQR} = \begin{bmatrix} 1.41 & 1.09 & 0 & 0 \\ 0 & 0 & 0.08 & 0.06 \end{bmatrix} \quad (4.6)$$

$$\text{Poles} = \begin{bmatrix} -1.42 \\ -1.42 \\ -14.93 \\ -19.19 \end{bmatrix} \quad (4.7)$$

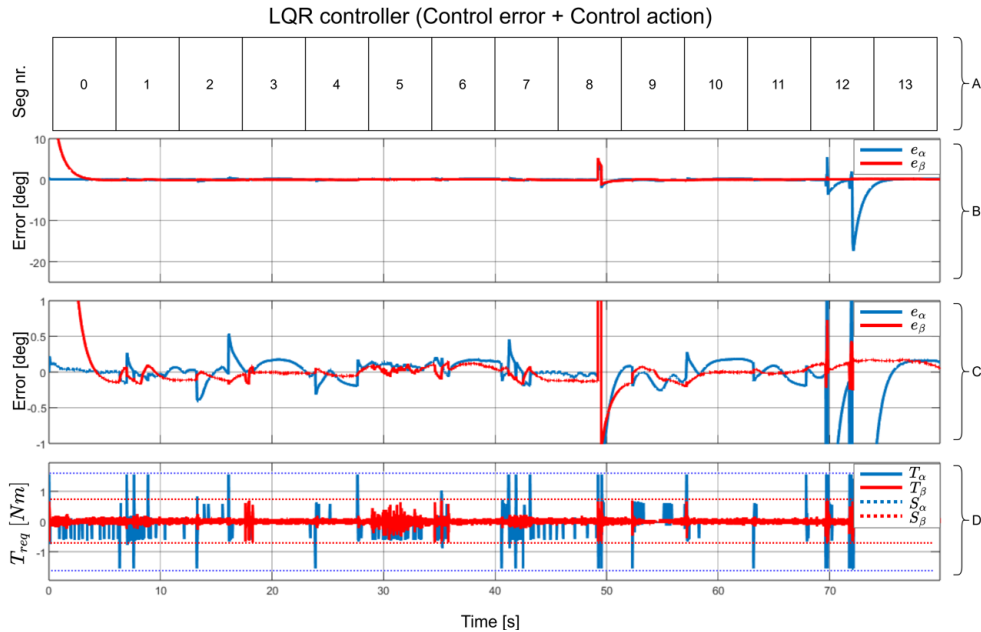


Figure 4.10: An LQR controller is implemented using the weighting matrices in equation 4.5 and the state-space model in equation 2.74. The control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different magnitude scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.

The control errors in LQR were slightly smaller than those observed in the PID controller case, although the motors experienced marginally higher levels of saturation. $I_\alpha = 22.21$ and $I_\beta = 29.64$.

By adding an integral state, the state-space model will be extended to be as shows in equation 2.76. This state penalties the accumulated error in the same way as the I component in a PID controller. The state weighting matrix Q_1 and the input weighting matrix Q_2 were chosen to be as follows:

$$\begin{cases} Q_1 = \text{diag} [2 & 1 & 10 & 2 & 1 & 5] \\ Q_2 = \text{diag} [1, & 300] \end{cases} \quad (4.8)$$

This results in the feedback gain matrix and the system poles as follows:

$$K_{LQI} = \begin{bmatrix} 3.09 & 1.19 & -3.16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.15 & 0.07 & -0.13 \end{bmatrix} \quad (4.9)$$

$$\text{Poles} = \begin{bmatrix} -1.27 + 0.79i \\ -1.27 - 0.795i \\ -1.44 + 1.05i \\ -1.44 - 1.05i \\ -14.93 + 0.00i \\ -19.19 + 0.00i \end{bmatrix} \quad (4.10)$$

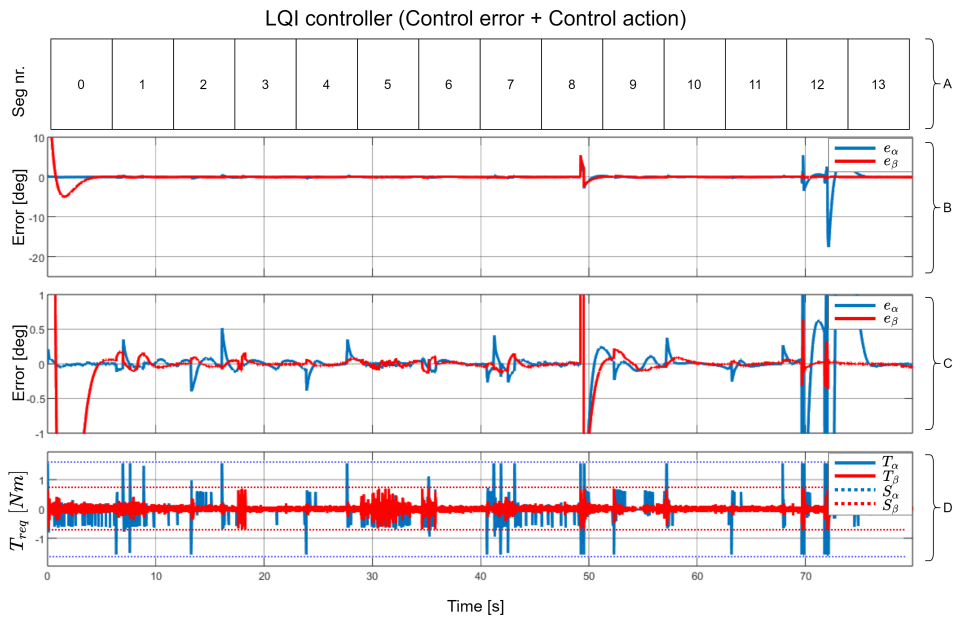


Figure 4.11: An Extended LQR controller is implemented using the weighting matrices in equation 4.8 and the state-space model in equation 2.76. The control error e and control action T_{req} are recorded. The errors are shown in subfigures B and C with different scaling, while subfigure C illustrates the control action, where T_α and T_β are the requested torques from the system, and S_α and S_β are the saturation levels for each motor. The figure is obtained by running the full simulation model with the Kalman filter based on IMU and differential GNSS activated.

As shown in Figure 4.11, using an LQI controller helped to minimize the error in the angle in Comparison with the LQR while maintaining nearly the same motor utilization as in the standard LQR controller case. The Indicator values were $I_\alpha = 19.17$ and $I_\beta = 22.08$.

5

Discussion

The Discussion chapter provides a detailed analysis of the findings from this research, addressing the research questions and evaluating the methodologies and results. It explores the performance of sensors, filtering techniques, and control strategies, emphasizing the trade-offs and challenges encountered during the development process. Key decisions, such as the exclusion of the magnetometer due to calibration and interference issues, are explained alongside the rationale for selecting alternative approaches. This chapter aims to critically link the results to the research objectives, offering a cohesive understanding of the study's outcomes.

5.1 Sensors

The Sensors section examines the role and impact of different sensors considered in this research, including the magnetometer and GNSS. While the magnetometer was initially explored for yaw estimation, its susceptibility to interference and the impracticality of achieving reliable calibration in the prototype led to its exclusion from the study. Instead, the focus shifted to the GNSS and IMU sensors, whose performance was critically evaluated to ensure accurate orientation and positioning. This section analyzes the sources of errors, such as translation inaccuracies, and their effects on target elevation and azimuth angle estimations, laying the foundation for understanding how sensor selection and performance influence the overall system reliability.

5.1.1 Magnetometer

Magnetometers are widely used in orientation estimation tasks, particularly for determining the yaw or heading angle of a vehicle. In this research, their primary application would have been to estimate the yaw angle. However, magnetometers typically require careful calibration along all three axes to provide accurate magnetic field measurements necessary for reliable yaw estimation [11].

In the physical prototype developed by the company, the placement of the magnetometer posed significant challenges. It was installed in close proximity to one of the actuators, which made it difficult to move the sensor along the spatial axes for proper calibration. This compromised the ability to achieve the required calibration accuracy prior to usage. Furthermore, magnetometers are inherently sensitive to

external magnetic fields. When subjected to interference from external sources or nearby components, such as the motors on the platform, they can produce incorrect magnetic field readings [11].

Given these limitations, and considering that the algorithms and methods developed in this research were intended for implementation on the physical prototypes, the magnetometer was excluded from the scope of this study. The decision was based on the impracticality of achieving reliable calibration in the prototype setup and the susceptibility of magnetometers to interference, which would have rendered them unsuitable for this specific application.

5.1.2 GNSS and translation error

In this section, the impact of factors such as assumptions and sensor uncertainties used to calculate the target elevation and azimuth angles, as discussed in section 2.2, will be analyzed. Additionally, it will be shown that the resulting errors in both elevation and azimuth angle estimations are minimal.

5.1.2.1 Error in Global Target Elevation Angle

In real world, the vehicle position is given at (r_0, lon_v, lat_v) where r_0 considered to be the earth radius. However, the true elevation angle must account for the the height of the vehicle above sea level, Δh_3 , the height of the vehicle where the terminal is mounted, Δh_2 , as well as the offset between the inner gimbal and the outer gimbal, Δh_1 . Ignoring these offsets introduces an error in the global elevation target angle.

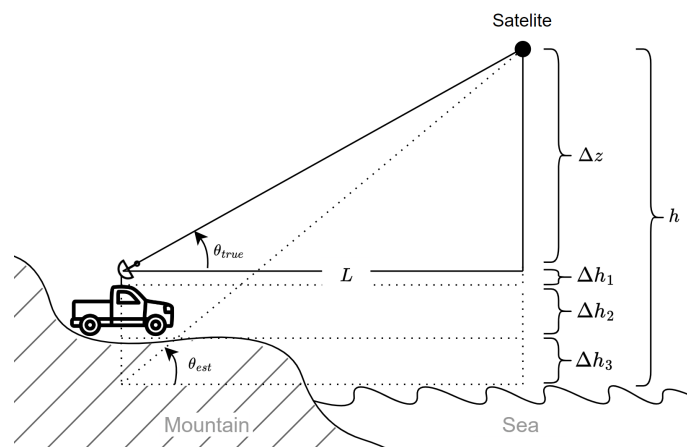


Figure 5.1: The elevation error introduced by the vehicle's height above sea level, the height of the vehicle where the terminal is mounted and the offset between the inner and outer gimbals.

The error in the elevation angle, e_{el} , can be calculated as the difference between θ_{est} and θ_{true} , as shown in equation 5.1. Since $\Delta z \gg (\Delta h_1 + \Delta h_2 + \Delta h_3)$, the elevation angle error, e_{el} , is minimal.

$$e_{el} = \text{atan}\left(\frac{h}{L}\right) - \text{atan}\left(\frac{\Delta z}{L}\right) \quad (5.1)$$

5.1.2.2 Error in Global Target Azimuth Angle

Using GNSS to determine the longitude and latitude of a vehicle in real world, introduces an error, e_{az} , in calculating the global target azimuth angle due to GNSS uncertainty.

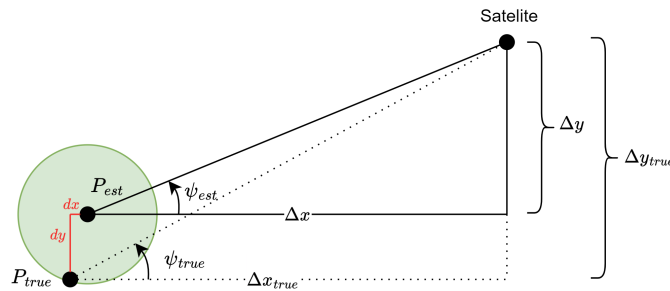


Figure 5.2: Top view: GNSS estimates the vehicle to be at P_{est} , while it could be anywhere inside the green circle, representing the uncertainty of the circle.

The error in the azimuth angle is the difference between the true and estimated azimuth angles, and is assumed to be negligible because $\Delta x \gg dx$ and $\Delta y \gg dy$. The error e_{az} is calculated as follows:

$$e_{az} = \text{atan}\left(\frac{\Delta y_{true}}{\Delta x_{true}}\right) - \text{atan}\left(\frac{\Delta y}{\Delta x}\right) \quad (5.2)$$

5.2 Answering Research Questions

Q1: *Which filtering technique offers the highest precision and reliability for processing data in this specific context?*

This research identifies the Kalman filter based on IMU and differential GNSS as the most precise and reliable filtering technique among those analyzed and validated. As detailed in Section 4.3, the complementary filter demonstrated greater responsiveness compared to the Kalman filter. However, accuracy in vehicle orientation estimation, a critical factor affecting the estimation of α and β angles, takes precedence over responsiveness. Although the Kalman filter exhibits a slight delay due to the dynamics of its process model, this is an acceptable trade-off when compared to the drift issues observed with the complementary filter.

Furthermore, between the two Kalman filter variants studied (with and without differential GNSS), the version incorporating differential GNSS outperformed the

alternative. Without differential GNSS, the yaw angle, a state lacking direct measurement, was susceptible to drift, compromising overall reliability.

In conclusion, the Kalman filter based on IMU and differential GNSS proved to be the most effective solution for accurately and reliably estimating the vehicle's orientation, making it the optimal choice among the filtering techniques explored in this research.

Q2: *What methods can be used to evaluate and compare different control strategies, and which control algorithm demonstrates superior effectiveness when implemented in MATLAB/Simulink?*

In this research, the evaluation of control strategies was based on two key criteria: error minimization and motor utilization. Stability analysis of different combinations of PID controllers identified two stable configurations, PD and PID controllers, which were selected alongside LQR and LQI controllers for further evaluation, as detailed in Section 2.5.

As discussed in Section 4.4, the PD controller exhibited the lowest error for the α angle and the second-lowest error for the β angle. However, it caused significant motor saturation, making it less practical for real-world scenarios where efficient motor utilization is critical. In contrast, the PID controller effectively minimized motor saturation and achieved the lowest total error for the β angle, though it exhibited the highest total error for the α angle.

The LQR and LQI controllers demonstrated a balanced performance, achieving low errors while maintaining acceptable motor utilization. The LQI controller was particularly effective, as it successfully eliminated steady-state errors, a limitation observed in the standard LQR controller. As detailed in Section 4.4, the LQI controller achieved a better error minimization for the α angle compared to the PID controller, while maintaining motor saturation levels comparable to those of the PID controller.

It is important to note that these results were influenced by the specific tuning of the controllers performed during this research. Different tuning parameters could potentially lead to varying levels of performance for the control algorithms. However, based on the current tuning, the LQI controller emerged as the most effective strategy for controlling the α angle, meeting the control objectives of minimizing steady-state errors and avoiding motor saturation. Similarly, the PID controller was identified as the optimal choice for controlling the β angle, achieving minimal steady-state error and efficient motor utilization.

Consequently, the final control implementation utilized the LQI controller for the α angle and the PID controller for the β angle.

6

Conclusion

The purpose of this thesis was to develop an automated satellite tracking system for vehicle-mounted antennas, ensuring continuous alignment with geostationary satellites. This was achieved through a literature study, mathematical modeling, sensor fusion implementation, control algorithm design, and validation within a simulation environment.

The key findings from this research showed that the Kalman filter, based on IMU and differential GNSS data, was identified as the most effective sensor fusion technique due to its accuracy and reliability in orientation estimation. Additionally, in the evaluation of control strategies, the LQI controller proved to be the best for α angle control, minimizing steady-state error and maintaining motor efficiency, while the PID controller was most effective for β angle control due to its balance of low error and minimal motor saturation. These findings were based on the specific tuning applied in this research, which played a significant role in determining the performance of the controllers.

Overall, the developed system demonstrated the ability to maintain reliable satellite alignment, ensuring continuous connectivity. The simulation-based approach provided valuable insights into the dynamics and control of vehicle-mounted satellite terminals, establishing a strong foundation for future research and potential hardware implementation.

7

Future Recommendations

This chapter explores potential avenues for further development and improvement of the system. Key topics include the integration of signal strength sensors to enhance state estimation and control, the adoption of quaternions to address limitations in orientation representation, and the analysis of controller robustness. Robustness considerations, such as the system's ability to withstand disturbances, parameter variations, and noise, are critical for ensuring reliable performance under real-world conditions. These possibilities highlight opportunities for refining the system's accuracy, resilience, and overall robustness in future iterations or dedicated research projects.

7.0.1 Signal Strength

Signal strength sensors can play a crucial role in optimizing the performance of the antenna system by providing feedback on the strength of the signal received from the satellite. These sensors can help evaluate how accurately the antenna is aligned with the satellite, enabling more precise control of the gimbals. For instance, a beacon receiver is a sensor specifically designed for this purpose. By measuring the signal strength, it could be integrated into the system to enhance state estimation and improve gimbal control.

However, such sensors come with challenges. Signal strength readings can be adversely affected by environmental factors such as tall buildings, obstacles, or other sources of interference. In these scenarios, the sensor may indicate a weak signal, falsely suggesting that the antenna is misaligned, even when it is accurately pointing at the satellite.

Despite these limitations, signal strength sensors could still be valuable for specific applications, such as providing temporary corrections or serving as a secondary validation mechanism. Further research is needed to explore the feasibility of integrating these sensors into the system and to develop robust algorithms capable of mitigating the impact of environmental disturbances. Investigating the potential of signal strength sensors could significantly enhance the accuracy and reliability of the system in future iterations of this project.

7.0.2 Euler Angles to Quaternions

Gimbal lock is a phenomenon that occurs when two of the three rotational axes in a system become aligned, causing a loss of one degree of freedom in rotation representation. This problem arises when using Euler angles for orientation, as they rely on three sequential rotations around fixed axes, which can lead to singularities in certain configurations.

To address this issue, orientation can instead be represented using quaternions. Quaternions are a mathematical representation of rotation in three-dimensional space that avoids the singularity problem inherent to Euler angles. By expressing rotations as four-dimensional vectors, quaternions ensure smooth and continuous orientation tracking, regardless of the system's orientation. This approach is computationally efficient and eliminates the risk of gimbal lock.

Implementing quaternions for orientation representation would not affect the feedback control loop or the state estimation process, as the core system dynamics remain unchanged. This improvement is worth considering for the future development of this project, particularly to enhance robustness and reliability in handling complex rotations.

7.0.3 Robustness Analysis

Robustness is a critical factor in evaluating the performance of controllers for any system, particularly when ensuring reliability under varying conditions. Several aspects must be considered when analyzing the robustness of controllers, including stability margins, sensitivity to parameter variations, resilience to external disturbances, and the ability to handle modeling inaccuracies. These factors play a significant role in determining the optimal controller for a system.

A system's response can be influenced by various external and internal factors, such as jitter, disturbances, noise, actuator limitations, and latency in control execution. Robustness analysis involves systematically evaluating how well a controller performs under these challenging conditions. It enables the identification of weaknesses in the control design and offers insights into potential improvements.

This type of analysis could form the basis for future research or even a dedicated thesis on this project. Investigating and optimizing the robustness of the system's controllers would not only enhance their reliability but also provide a deeper understanding of the system's behavior under real-world operating conditions.

Bibliography

- [1] H. Olsson, K.J. Åstöm, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky (1998). Friction Models and Friction Compensation. *European Journal of Control*, 179.
- [2] Gerard Maral and Michel Bousquet (2009). Pointing angles of an earth station antenna. In *Satellite Communications Systems* (4th ed., pp. 390-392). Wiley. ISBN: 978-0-470-71458-4.
- [3] J. L. Meriam and L.G. Kraige (2013). Momentum and energy equations of motions. In *Meriam's Engineering Mechanics: Dynamics* (7th ed.). Wiley.
- [4] A. R. Abdulghany (2017). Generalization of parallel axis theorem for rotational inertia. *American Journal of Physics*.
- [5] S. Yoon and J. B. Lundberg (2001). Equations of Motion for a Two-Axes Gimbal System. *IEEE*.
- [6] Narkhede, P., Poddar, S., Walambe, R., Ghinea, G., Kotecha, K. (2021). Cascaded Complementary Filter Architecture for Sensor Fusion in Attitude Estimation. *Sensors* (Basel, Switzerland), 21(6), 1937.
- [7] Simo Sarkkå and Lennart Svensson (2023). Bayesian Filtering and Smoothing. Second Edition. *Cambridge University Press*.
- [8] Alur, R. (2015). *Principles of Cyber-Physical Systems*. MIT Press. ISBN: 9780262029117.
- [9] Glad, T. , Ljung, L. (2000). *Control Theory: Multivariable and Nonlinear Methods*. Taylor , Francis. ISBN: 0-7484-0878-9 (pbk), 0-7484-0877-0 (hbk).
- [10] Guinness World Records (2020). *Baldwin Street in New Zealand reinstated as the world's steepest street*.
- [11] The MathWorks Inc. (2024). *Magnetometer Calibration Documentation, Natick, Massachusetts: The MathWorks Inc.* <https://se.mathworks.com/help/fusion/ug/magnetometer-calibration.html>
- [12] Bengt Lennartson (2000, 2002). *Reglerteknikens grunder*. Studentlitteratur ISBN: 9789144024165..

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY