Technical University of Munich

Institute of Flight System Dynamics

Actuator Failure Detection Strategies for a Quadrotor UAV

–

Strategien zur Erkennung von Aktuatorfehlern für ein Quadrotor-UAV

Semester Thesis

Author: Aqib Habib

Matriculation Number: 03797032

Supervisors: Zoe Mbikayi

Examiner: Prof. Dr.-Ing. Florian Holzapfel

March 28, 2026

Statutory Declaration

I, Aqib Habib, declare on oath towards the Institute of Flight System Dynamics of Technical University of Munich, that I have prepared the present Semester Thesis independently and with the aid of nothing but the resources listed in the bibliography.

This thesis has neither as-is nor similarly been submitted to any other university.

Garching, March 28, 2026

Aqib Habib

Abstract

Unmanned Aerial Vehicles (UAVs) have become an essential platform for research and commercial applications, where reliability and safety are critical for mission success. Among various UAV configurations, quadrotors are widely adopted due to their mechanical simplicity and high manoeuvrability. However, their performance and stability can deteriorate significantly in the presence of actuator faults, which may lead to a complete loss of control if not detected and handled promptly. This thesis investigates model-based Actuator Failure Detection Strategies *focusing on observers for a quadrotor UAV operating under nominal and faulty conditions.*

A complete six degrees-of-freedom (6-DOF) nonlinear dynamic model of the quadrotor is developed, including translational, rotational, and kinematic relations between the body, Earth-Centred Inertial, Earth-Centred Earth-Fixed, and Navigation frame. The forces and moments produced by the rotors are modelled based on aerodynamic coefficients from existing literature, and the control architecture is implemented as a cascaded PID structure comprising position, attitude, and altitude control loops. The nominal control and model parameters are derived from the Parrot Mambo Mini Drone configuration.

To detect actuator faults, several model-based fault detection and diagnosis (FDD) methods are implemented and evaluated. These model-based observers exploit analytical redundancy between measured and estimated system states to generate residual signals sensitive to actuator faults. The fault models considered include partial and complete loss of effectiveness. Linearization around the hover equilibrium is carried out to facilitate the observer design.

Simulation studies are performed in MATLAB/Simulink to evaluate the performance of the proposed detection strategies under different fault magnitudes and flight conditions. The results demonstrate that residual-based detection schemes can successfully identify actuator faults with high sensitivity. The comparative analysis highlights the trade-off between modelling and computational complexity, fault detection time, and robustness against parameter uncertainty for each observer type.

Overall, the thesis contributes to enhancing the reliability of quadrotor UAVs by providing a systematic framework for actuator fault modelling, detection, and analysis.

Table of Contents

List of Figures

List of Tables

Table of Symbols

Sets

Symbol	Description
$\boldsymbol{h}$	nonlinear component of state dynamics
$\boldsymbol{h}_m$	measurement model
$\boldsymbol{x}$	state vector
$\hat{\boldsymbol{x}}$	estimated state vector
$\boldsymbol{f}$	dynamics
$\boldsymbol{u}$	input vector
$\boldsymbol{\nu}$	virtual controls vector
$\boldsymbol{y}$	output vector
$\hat{\boldsymbol{y}}$	estimated output vector

Greek Letters and Variables

Symbol	Description
A	dynamic matrix of the reference model
B	input matrix of the reference model
C	output matrix of the reference model
D	feed-forward matrix of the reference model
c_T	thrust coefficient
c_M	moment coefficient
e_x	x-position error
e_y	y-position error
e_{v_x}	x-velocity error
e_{v_y}	y-velocity error
e_Φ	roll angle error
e_Θ	pitch angle error
e_Ψ	yaw angle error
$\vec{F}$	force vector
g	constant gravitational acceleration
$(I)^G_{BB}$	total moment of inertia
I_{xx}	moment of inertia along x-axis
I_{yy}	moment of inertia along y-axis
I_{zz}	moment of inertia along z-axis
$J(t)$	decision function
K	observer-gain matrix
k_{px}	P gain of x-position controller
k_{dx}	D gain of x-position controller
k_{py}	P gain of y-position controller
k_{dy}	D gain of y-position controller
k_{pz}	P gain of z-position controller

k_{dz}	D gain of z-position controller
k_{iz}	I gain of z-position controller
$k_{p\Phi}$	P gain of roll attitude controller
$k_{d\Phi}$	D gain of roll attitude controller
$k_{i\Phi}$	I gain of roll attitude controller
$k_{p\Theta}$	P gain of pitch attitude controller
$k_{d\Theta}$	D gain of pitch attitude controller
$k_{i\Theta}$	I gain of pitch attitude controller
$k_{p\Psi}$	P gain of yaw attitude controller
$k_{d\Psi}$	D gain of yaw attitude controller
$k_{i\Psi}$	I gain of yaw attitude controller
k_{aw}	anti-windup gain
l	arm length
$\vec{M}$	moment vector
m	mass
M_{NB}	rotation matrix, rotates a vector from the B into the N-frame
p	roll rate
q	pitch rate
R	Radius of Propeller
R_ψ	yaw rotation matrix
r	yaw rate
$\mathbf{r}$	residual vector
$\vec{\boldsymbol{r}}$	position vector
t_f	fault time
T	total thrust
$\tilde{T}$	faulty thrust
$\tilde{\tau}$	faulty torque
T_d	desired thrust
$\vec{\boldsymbol{v}}$	velocity vector
x	x-position in inertial frame
x_d	desired x-position
y	y-position in inertial frame
y_d	desired y-position
z	z-position in inertial frame
z_d	desired z-position
Γ_i	angle between the B-frames's x-axis X_B and rotor arm
ξ_i	fault parameter
$\vec{\omega}_{NB}$	rotational rate of the B-frame w.r.t. the N-frame
$\vec{\omega}_{IB}$	rotational rate of the I-frame w.r.t. the B-frame
Ω_i	rotational speed of rotors
Φ	roll angle (Euler angles)
Φ_d	desired roll angle

Θ	pitch angle (Euler angles)
Θ_d	desired pitch angle
Ψ	yaw angle (Euler angles)
Ψ_d	desired yaw angle
ρ	air density
τ_x	roll moment
τ_y	pitch moment
τ_z	yaw moment

Indices

Symbol	**Description**
B	Body-fixed frame
E	Earth centered Earth-fixed frame
I	Earth centered inertial frame
N	local navigation frame
O	NED Frame

1. Motivation

Unmanned Aerial Vehicles (UAVs), particularly quadrotors, have gained remarkable attention in both research and industry due to their high agility, vertical takeoff and landing (VTOL) capability, and simple mechanical structure. They have found extensive applications in areas such as aerial photography, search and rescue, environmental monitoring, logistics, and autonomous inspection [1] . However, as UAVs are increasingly deployed in safety-critical and autonomous operations, ensuring their reliability, safety, and fault resilience has become a fundamental requirement [2].

In a quadrotor, all actuators contribute simultaneously to both lift and attitude control, any malfunction in the propulsion system, sensors, or control hardware can lead to serious instability or even complete mission failure [3]. In such scenarios, Fault Detection and Diagnosis (FDD) becomes essential for identifying abnormal behaviour early.

Model-based methods offer strong physical interpretability but rely on accurate system modelling, whereas data-driven techniques are flexible and adaptive, but require large datasets [4].

This thesis investigates Fault Detection Strategies for a Quadrotor UAV, their modelling foundations, mathematical formulations, and implementation frameworks. The aim is to compare different model-based fault detection approaches, focusing on observers, and evaluate their performance through simulations.

1.1. Introduction

Fault Detection and Diagnosis (FDD) schemes for multirotor UAVs are essential for ensuring safe and reliable operation, especially when dealing with actuator faults, which can lead to severe performance degradation or complete system failure. These schemes can be broadly classified into model-based and data-based methods, each with its own strengths and applications [5].

1.2. Model Based

1.2.1. State Estimation

State estimation techniques utilise observers and filters to estimate the system states and generate residuals for fault detection. The Thau observer is a nonlinear observer applied to Lipschitz systems and has been successfully used for actuator fault diagnosis in quadrotors and hexarotors [6, 7]. This observer requires the system to be observable and the nonlinear function to be Lipschitz continuous.

The Luenberger observer, being the linear counterpart of the Thau observer, has also been widely applied in multirotor FDD systems. It has been used for sensor fault detection and estimation [8] as well as for actuator fault diagnosis [9]. While simpler to implement than nonlinear observers, the Luenberger observer's performance is limited to linear or linearized systems and can be sensitive to measurement noise.

Other observers include:

- Sliding-mode observers for robust fault reconstruction under uncertainties, applied to quadrotor actuator fault diagnosis [10, 11].
- H-infinity observers for handling disturbances and model uncertainties in quadrotor systems [12, 13].
- Unknown input observers to decouple disturbances from residuals in actuator fault detection [14, 15].
- Kalman filters, including Extended and Unscented variants, for state estimation under noise in multirotor systems [16, 17].

1.2.2. Parameter Estimation

Parameter estimation methods detect faults by identifying changes in system parameters. For example, the recursive least squares method has been used to estimate thrust and torque coefficients to detect rotor faults such as blade deformation or damage [18]. These methods require accurate system models and are sensitive to external disturbances.

1.2.3. Simultaneous State or Parameter Estimation

This approach combines state and parameter estimation to provide comprehensive fault diagnosis. Techniques such as the Two-Stage Kalman Filter (TSKF) and Three-Stage Kalman Filter have been applied to estimate both system states and actuator loss of effectiveness [16, 17, 19]. While effective, these methods are computationally intensive and sensitive to initialization.

1.2.4. Parity Space

Parity space methods generate residuals based on analytical redundancy relations. These methods have been applied to both sensor and actuator fault detection in multirotors. For instance, nonlinear parity relations have been used to isolate actuator faults in hexarotors [20]. Differential flatness-based analytical redundancy has also been employed for sensors and actuators diagnosis [21]. These approaches are fast but can be sensitive to measurement noise and disturbances.

1.3. Data-Based Methods

1.3.1. Statistical

Statistical methods use data-driven techniques to detect faults by analyzing the statistical properties of system measurements. Principal Component Analysis (PCA) has been adopted for online fault detection of motor and rotor faults [22]. Statistical classification techniques including discriminant analysis, canonical variate analysis, and partial least squares regression have been compared for isolating quadrotor actuator faults [23]. Fault detection and isolation methods based on recursive least squares and weighted linear least squares have been detailed for diagnosing propulsion system faults of octorotor UAVs [24, 25].

1.3.2. Qualitative Methods

Qualitative methods include signal analysis techniques such as Discrete Wavelet Transform (DWT) and Fourier analysis, which are used to detect faults like unbalanced propellers by analyzing audio signals [26] or vibration spectra measured by accelerometers [27]. Motor current signature analysis has been employed to detect shorted turns in motors [28]. Fuzzy logic techniques, such as adaptive interval type-2 fuzzy logic, have been proposed as actuator fault observers for modified quadrotor UAVs [29].

1.3.3. Neural Networks

Neural networks have shown significant promise in actuator fault diagnosis. Fuzzy adaptive resonance neural networks have been employed to detect whether motors are operating in normal or faulty condition [30]. Parallel banks of recurrent neural networks have been designed to precisely estimate the severity of actuator faults [31]. Extreme learning machines have been proposed for actuator fault detection with computational efficiency suitable for small-scale microprocessors [32]. Hybrid CNN-LSTM models have been developed for actuator fault diagnosis in hexarotor UAVs [33].

1.4. Structure

The rest of this thesis is structured as follows:

- **Chapter 2 – Quadrotor Dynamics:** Presents the mathematical modelling of the quadrotor, including translational, rotational, and kinematic equations. The derivations are based on Newton–Euler formalism, and the aerodynamic forces and moments generated by each rotor are expressed as functions of thrust and moment coefficients.
- **Chapter 3 – Quadrotor Control:** Explains the cascaded control structure used for stabilising and manoeuvring the quadrotor. The chapter details the position, attitude, and al-

titude control loops, followed by the motor-mixing algorithm that maps control commands to individual rotor thrusts.

- **Chapter 4 – Actuator Faults, Sensors and Observers:** Defines actuator fault types such as loss of effectiveness, bias, and stuck faults. It provides their mathematical representation and discusses their physical implications on the generated forces and moments of the quadrotor. Introduces the sensor model used for the residual evaluation module, and finally, the derivation of observers.
- **Chapter 5 – Actuator Fault Detection and Diagnosis:** Introduces the theory and implementation of observer-based fault detection and diagnosis (FDD) methods, their comparison, and how residuals are generated and evaluated. It also evaluates the false alarms caused by parametric uncertainty.
- **Chapter 6 – Conclusion and Results:** Summarizes the key findings, highlights the effectiveness of the proposed detection schemes.

2. Quadrotor Dynamics

In this section, the dynamic model of a quadrotor system is derived. The quadrotor considered in this study is the *Parrot Mambo* minidrone [34], which is shown in Figure 2–1.

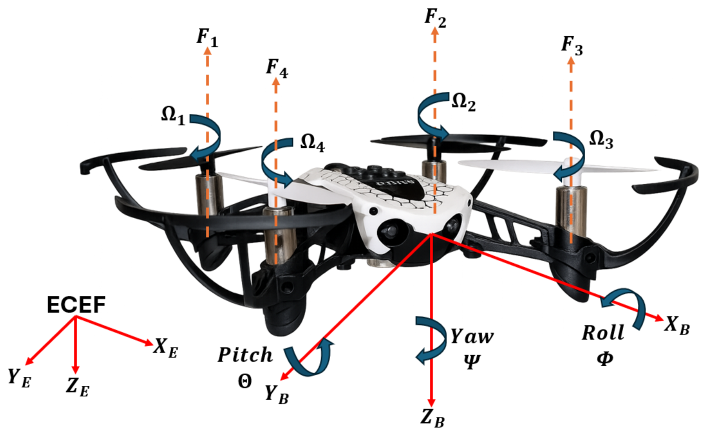


Figure 2–1.: Parrot Mambo Drone

2.1. Rigid Body Equations of Motion

The quadrotor dynamics are described using three main coordinate frames: the Earth-Centered Inertial (ECI) frame (I), the Earth-Centered Earth-Fixed (ECEF) frame (E), and the body-fixed frame (B). The ECEF frame rotates with respect to the ECI frame at the Earth rotation rate $\vec{\omega}^{IE}$ and share the same origin but differ by a time-varying rotation. In addition to the Earth rotation rate $\vec{\omega}^{IE}$, a transport rate $\vec{\omega}^{EO}$ is introduced when a local navigation frame is defined on the Earth's surface. The transport rate represents the angular velocity of the local frame with respect to the ECEF frame due to translational motion over the curved Earth. A further description of the frames used in this thesis is provided in Appendix A.

The following assumptions are introduced for the quadrotor, which are derived from [35].

- **Assumption 1: Flat and non-rotating Earth** The usual range and achievable velocities of a multirotor system are small so that the Earth can be modelled as flat and non-rotating. This means that the transport rate $\vec{\omega}^{EO}$ and the Earth's rotation rate $\vec{\omega}^{IE}$ are negligible because rotations from moving over the curved Earth and from Earth spinning are small compared to the quadrotor's own motions.
- **Assumption 2: Rigid Body** Any two points of the multirotor have a constant distance between each other. Furthermore, the multirotor has a constant mass and a constant moment of inertia.
- **Assumption 3: Centre of Gravity** The system considered here have a symmetric mass distribution, and it is assumed that the centre of gravity coincides with the origin of the

body-fixed axes to avoid any additional dynamics generated by the offset.

- **Assumption 4: Aerodynamic Forces and Moments** Aerodynamic forces and moments regarding the body of the quadrotor are neglected due to near hover and low-speed conditions explored in this work.

2.2. Translational Dynamics

The system's dynamics are derived from Newton's second law, where linear and angular momentum are conserved. The motion equations relative to the centre of gravity G can be expressed in the body-fixed coordinate frame as described in [35]:

$$m\Big((\vec{v}^G)_B^{IB} + (\vec{\omega}^{IB})_B \times (\vec{v}^G)_B^I\Big) = (\vec{F}_T^G)_B \tag{2–1}$$

where

m : Mass of the body
$(\vec{v}^G)_B^{IB}$: Velocity of centre of gravity with respect to ECI frame, derived with respect to body frame, and expressed in the body frame
$(\vec{\omega}^{IB})_B$: Angular rates of the rigid body with respect to the ECI frame, and expressed in the body frame
$(\vec{v}^G)_B^I$: The velocity of the centre of gravity with respect to the ECI frame, and expressed in the body frame
$(\vec{F}_T^G)_B$: Total force vector expressed in the body frame

Translational dynamics can be further simplified by describing the acceleration with respect to the local navigation frame, as elaborated in [35], and is explained in Appendix A. The acceleration relative to the ECI system is:

$$(\vec{v}^G)^{II} = (\vec{v}^G)^{IB} + \vec{\omega}^{IB} \times (\vec{v}^G)^I \tag{2–2}$$

From Assumption 1 (flat and non-rotating Earth) and the definition of the N-frame, it follows that:

$$\vec{\omega}^{IN} = \vec{\omega}^{IE} + \vec{\omega}^{EN} = 0 \tag{2–3}$$

Hence, using the transport theorem, the acceleration expressed relative to the inertial frame is equal to that expressed relative to the navigation frame:

$$(\dot{\vec{v}}^G)^{II} = (\dot{\vec{v}}^G)^{NN} \tag{2–4}$$

Substituting this result into Equation 2–1 yields the simplified translational dynamics:

$$m(\dot{\vec{v}}^G)_N^{NN} = M_{NB}(\vec{F}_T^G)_B \tag{2–5}$$

The total force $(\vec{F}_T^G)_B$: contains the sum of propulsion (P), aerodynamics (A) and gravitational (G) forces:

$$(\vec{F}_T^G)_B = (\vec{F}_P^G)_B + (\vec{F}_A^G)_B + M_{BN}(\vec{F}_G)_N \tag{2–6}$$

where the gravitational force is modelled as a constant force field:

$$(\vec{F}_G)_N = m(\vec{g})_N = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2–7}$$

where g is the constant gravitational acceleration and $\vec{g}$ is the gravitational vector.

From Assumption 4 in section 2.1, the aerodynamic forces and moments on the quadrotor are negligible, i.e. $(\vec{F}_A^G)_B = 0, (\vec{M}_A^G)_B = 0$.

The total propulsion forces for quadrotors derived from [36] are:

$$(\vec{F}_P^G)_B = \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ -c_T \rho A_r (\Omega_i^2 R^2) \end{bmatrix} \tag{2–8}$$

where

c_T : Thrust coefficient
ρ : Density of air
R : Radius of propeller
Ω_i : Rotational speed of i-th rotor
A_r : Disk area = πR^2

Substituting all definitions of forces into Equation 2–5 yields the translational acceleration in the local navigation frame as:

$$(\dot{\vec{v}}^G)_N^{NN} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + M_{NB} \frac{1}{m} \sum\nolimits_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ -c_T \rho A_r (\Omega_i^2 R^2) \end{bmatrix} \tag{2–9}$$

where M_{NB} describes the rotation matrix from the body frame to the local navigation frame as:

$$M_{NB} = \begin{bmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & \sin\Theta \\ \sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi & \sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi & \sin\Phi\cos\Theta \\ \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi & \cos\Phi\cos\Theta \end{bmatrix} \tag{2–10}$$

2.3. Rotational Dynamics

Using Newton's 2nd law again, the total moments of a body with respect to the centre of gravity can be expressed in the body fixed frame as described in [35]:

$$(I^G)_{BB}(\dot{\vec{\omega}}^{IB})_B^B + (\vec{\omega}^{IB})_B \times (I^G)_{BB}(\vec{\omega}^{IB})_B = (\vec{M}_T^G)_B \tag{2–11}$$

where

$(\dot{\vec{\omega}}^{IB})_B^B$: Angular accelerations of the rigid body with respect to the ECI frame, and expressed in the body frame
$(\vec{M}_T^G)_B$: Total moment vector expressed in the body frame
$(I^G)_{BB}$: The moment of inertia with respect to the centre of gravity expressed in the body frame

We can use the simplifications again made for Equation 2–1 to get:

$$(I^G)_{BB}(\dot{\vec{\omega}}^{NB})_B^B + (\vec{\omega}^{NB})_B \times (I^G)_{BB}(\vec{\omega}^{NB})_B = (\vec{M}_T^G)_B \tag{2–12}$$

The total moment $(\vec{M}_T^G)_B$, contains the sum of propulsion (P), aerodynamics (A) and gravitational (G) moments. Due to the assumptions outlined in section 2.1 (Assumption 4), the aerodynamic and gravitational moments are negligible.

The propulsion moments consist of two parts: the yaw moments generated by the propeller's reaction and the roll and pitch moments produced by the offset of each rotor position from the centre of gravity (CoG), i.e. due to lever-arm. The total moment can thus be expressed as:

$$(\vec{M}_T^G)_B = \sum_{i=1}^{4} \Big((\vec{M}_{P_i}^G)_B + (\vec{r}^{GP_i})_B \times (\vec{F}_{P_i}^G)_B \Big) \tag{2–13}$$

The first term in Equation 2–13 corresponds to the propeller's reaction moment. In contrast, the second term represents the moments generated due to the propulsion forces acting at a distance from the vehicle's CoG as described in [35, 37]. By expanding further, the total moment can be rewritten as:

$$(\vec{M}_T^G)_B = \sum_{i=1}^{4} \left(\begin{bmatrix} 0 \\ 0 \\ -c_M \rho A_r (\Omega_i |\Omega_i| R^3) \end{bmatrix} + \begin{bmatrix} l\cos(\Gamma_i) \\ l\sin(\Gamma_i) \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ -c_T \rho A_r (\Omega_i^2 R^2) \end{bmatrix} \right) \tag{2–14}$$

where

c_M : Moment coefficient
$(\vec{r}^{GP_i})_B$: Position vector of the i-th propeller, with respect to the centre of gravity, and expressed in the body frame
l : Length of propeller arms
$\Omega_i|\Omega_i|$: Signed squared rotational speed
Γ_i : Angle between the body fixed axis $\vec{X}_B$ and rotor arm

For the x-quadrotor configuration as shown in Figure 2–1, it holds that:

$$\Gamma_i \in \{45^\circ,\, 135^\circ,\, 225^\circ,\, 315^\circ\}$$

2.4. Kinematic Relations

In this thesis, the Euler angle representation (Φ, Θ, Ψ) is adopted, as it provides an intuitive interpretation of roll, pitch, and yaw motions as described in [38].

The body angular rates (p, q, r) as a function of the Euler angle rates $(\dot{\Phi}, \dot{\Theta}, \dot{\Psi})$ can be defined as [38]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \cos\Theta\sin\Phi \\ 0 & -\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{2–15}$$

In practice, it is more convenient to use the inverse relationship, expressing Euler angle rates in terms of body angular velocities. This transformation is given by:

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \frac{\sin\Phi}{cos\Theta} & \frac{\cos\Phi}{\cos\Theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2–16}$$

3. Quadrotor Control

This section presents the overview of the quadrotor parameters, state space model, control allocation, motor mixing and cascaded controller. The cascaded control structure consists of position, attitude, and altitude control loops.

3.1. Parameters of Quadrotor

The quadrotor parameters used in this work, as summarised in Table 3–1, include the physical properties of the vehicle as well as its motor and geometric characteristics. The model parameters are referenced from [34, 39–41].

Table 3–1.: Quadrotor Parameters

Parameter	Value	Description	Unit
m	0.0630	Mass	kg
c_M	0.00078264	Moment coefficient	-
c_T	0.0107	Thrust coefficient	-
I_{xx}	6.86×10^{-5}	Moment of inertia along x-axis	kgm^2
I_{yy}	9.2×10^{-5}	Moment of inertia along y-axis	kgm^2
I_{zz}	1.366×10^{-4}	Moment of inertia along z-axis	kgm^2
g	9.80665	Constant gravitational acceleration	m/s^2
R	0.0325	Radius of propeller	m
ρ	1.225	Air density	kg/m^3
l	0.0624	Arm length	m
Ω_{max}	500	Maximum motor command	-
K_{Ω^2}	13840.8	Motor command to Ω^2 gain	-

3.2. State Space Model

The complete quadrotor dynamics can be represented in state-space form by combining the translational dynamics, rotational dynamics, and kinematic relations derived in previous sections in Equations 2–1, 2–12 and 2–16 from Section 2.2, 2.3, and 2.4. This representation facilitates controller design and analysis.

3.2.1. State Vector Definition

The state vector of a quadrotor can be defined as:

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \Phi & \Theta & \Psi & p & q & r \end{bmatrix}^T \in \mathbb{R}^{12}$$

where

x, y, z**:** Position states
$\dot{x}, \dot{y}, \dot{z}$**:** Velocity states
Φ, Θ, Ψ**:** Euler angles
p, q, r**:** Angular rates

3.2.2. Input Vector Definition

The control input vector consists of the virtual controls:

$$\boldsymbol{\nu} = \begin{bmatrix} T & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^4 \tag{3–1}$$

where

T**:** Total thrust
τ_x, τ_y, τ_z**:** Roll, pitch and yaw moments

3.2.3. Nonlinear State Equations

In this section, the subscripts and superscripts describing the propulsion forces in Equation 2–8 are omitted to simplify the state-space equations. The propulsion forces are represented as T in this section, and the moments from Equation 2–14 are referred to as τ_x, τ_y, τ_z.

Acceleration Dynamics

Simplifying Equation 2–9 and denoting $\vec{v}$ as following:

$$\begin{aligned} \ddot{x} &= \frac{T}{m}(\cos\Phi\cos\Psi\sin\Theta + \sin\Phi\sin\Psi) \\ \ddot{y} &= \frac{T}{m}(\sin\Theta\cos\Phi\sin\Psi - \sin\Phi\cos\Psi) \\ \ddot{z} &= g - \frac{T}{m}(\cos\Theta\cos\Phi) \end{aligned} \tag{3–2}$$

Angular Rate Dynamics

Simplifying Equation 2–12 and denoting $\vec{\omega}$ as following:

$$\begin{aligned} \dot{p} &= \frac{1}{I_{xx}}\left(\tau_x + (I_{yy} - I_{zz})qr\right) \\ \dot{q} &= \frac{1}{I_{yy}}\left(\tau_y + (I_{zz} - I_{xx})pr\right) \\ \dot{r} &= \frac{1}{I_{zz}}\left(\tau_z + (I_{xx} - I_{yy})pq\right) \end{aligned} \tag{3–3}$$

where $(I^G)_{BB} =$ diag(I_{xx}, I_{yy}, I_{zz}) is the inertia matrix.

3.2.4. Control Allocation

The relationship between virtual controls $\boldsymbol{\nu}$ and individual rotor thrusts is given by the control allocation matrix:

$$\boldsymbol{\nu} = \boldsymbol{B}_a \boldsymbol{u} \tag{3–4}$$

where $\boldsymbol{u} = [T_1, T_2, T_3, T_4]^T$ are the individual rotor thrusts.

for an x-configuration quadrotor [42]:

$$\mathbf{B}_a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{-1}{\sqrt{2}}l & \frac{1}{\sqrt{2}}l & \frac{1}{\sqrt{2}}l & \frac{-1}{\sqrt{2}}l \\ \frac{-1}{\sqrt{2}}l & \frac{-1}{\sqrt{2}}l & \frac{1}{\sqrt{2}}l & \frac{1}{\sqrt{2}}l \\ \frac{k_M}{k_T} & -\frac{k_M}{k_T} & \frac{k_M}{k_T} & -\frac{k_M}{k_T} \end{bmatrix} \tag{3–5}$$

where

$$k_T = c_T \rho A_r R^2$$
$$k_M = c_M \rho A_r R^3$$

The inverse relationship for control allocation is:

$$\mathbf{u} = \mathbf{B}_a^+ \boldsymbol{\nu} \tag{3–6}$$

where $\mathbf{B}_a^+$ is the inversion of $\mathbf{B}_a$.

3.3. Jacobian Matrices

This section presents the Jacobian matrices for all the states of a quadrotor, which are later used in the design of observers in Chapter 4.

A Jacobian matrix is defined as:

$$\mathbf{F}(\boldsymbol{x}, \boldsymbol{\nu}) = \begin{bmatrix} f_1(x_1, \ldots, x_n, \nu_1, \ldots, \nu_n) \\ \vdots \\ f_n(x_1, \ldots, x_n, \nu_1, \ldots, \nu_n) \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{\nu} = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_n \end{bmatrix} \tag{3–7}$$

$$\mathbf{J}(\boldsymbol{x}) = \frac{\partial \mathbf{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{3–8}$$

Using Equations 2–16, 3–2, 3–3 and the definition of the Jacobian matrix, we get:

$$\mathbf{J}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{F}_{v\eta} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{F}_{\eta\eta} & \mathbf{F}_{\eta\omega} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{F}_{\omega\omega} \end{bmatrix} \tag{3–9}$$

where

$$\mathbf{F}_{v\eta} = \frac{T}{m}\begin{bmatrix} -\cos\Psi\sin\Theta\sin\Phi + \sin\Psi\cos\Phi & \cos\Psi\cos\Theta\cos\Phi & -\sin\Psi\sin\Theta\cos\Phi + \cos\Psi\sin\Phi \\ -\sin\Psi\sin\Theta\sin\Phi - \cos\Psi\cos\Phi & \sin\Psi\cos\Theta\cos\Phi & \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi \\ -\cos\Theta\sin\Phi & -\sin\Theta\cos\Phi & 0 \end{bmatrix}$$

$$\mathbf{F}_{\eta\eta} = \begin{bmatrix} \tan\Theta(q\cos\Phi - r\sin\Phi) & \sec^2\Theta(q\sin\Phi + r\cos\Phi) & 0 \\ -(q\sin\Phi + r\cos\Phi) & 0 & 0 \\ \frac{\cos\Phi}{\cos\Theta}q - \frac{\sin\Phi}{\cos\Theta}r & \frac{\tan\Theta}{\cos\Theta}(q\sin\Phi + r\cos\Phi) & 0 \end{bmatrix} \qquad \mathbf{F}_{\eta\omega} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \dfrac{\sin\Phi}{\cos\Theta} & \dfrac{\cos\Phi}{\cos\Theta} \end{bmatrix}$$

$$\mathbf{F}_{\omega\omega} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{I_{zz}-I_{xx}}{I_{yy}}r & 0 & \frac{I_{zz}-I_{xx}}{I_{yy}}p \\ \frac{I_{xx}-I_{yy}}{I_{zz}}q & \frac{I_{xx}-I_{yy}}{I_{zz}}p & 0 \end{bmatrix}$$

The Jacobian matrix for the virtual inputs is defined as:

$$\boldsymbol{G}(\boldsymbol{\nu}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\nu}} = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{G}_{vT} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{I}^{-1} \end{bmatrix} \tag{3–10}$$

where

$$\mathbf{G}_{vT} = \frac{1}{m}\begin{bmatrix} \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi \\ \sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi \\ \cos\Theta\cos\Phi \end{bmatrix} \qquad \mathbf{I}^{-1} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

3.4. Linearization of Quadrotor Dynamics Around Hover

A steady hover is defined by zero translational velocity and angular rate, constant position, and level attitude:

$$\boldsymbol{x} = \begin{bmatrix} x_o & y_o & z_o & 0 & 0 & 0 & 0 & 0 & \Psi_o & 0 & 0 & 0 \end{bmatrix}^T$$

Evaluating Equation 3–9 and 3–10 at hover, we get:

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{F}_{v\eta,0} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{G}_0 = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{e}_3/m & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{I}^{-1} \end{bmatrix} \tag{3–11}$$

where

$$\mathbf{F}_{v\eta,0} = g \begin{bmatrix} \sin\Psi_0 & \cos\Psi_0 & 0 \\ -\cos\Psi_0 & \sin\Psi_0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

3.5. Cascaded PID Control Architecture

The cascaded Proportional-Integral-Derivative (PID) control architecture represents a well-established methodology for quadrotor stabilisation and trajectory tracking, as demonstrated in comparative studies including [34]. The controller architecture is shown in Figure 3–1.

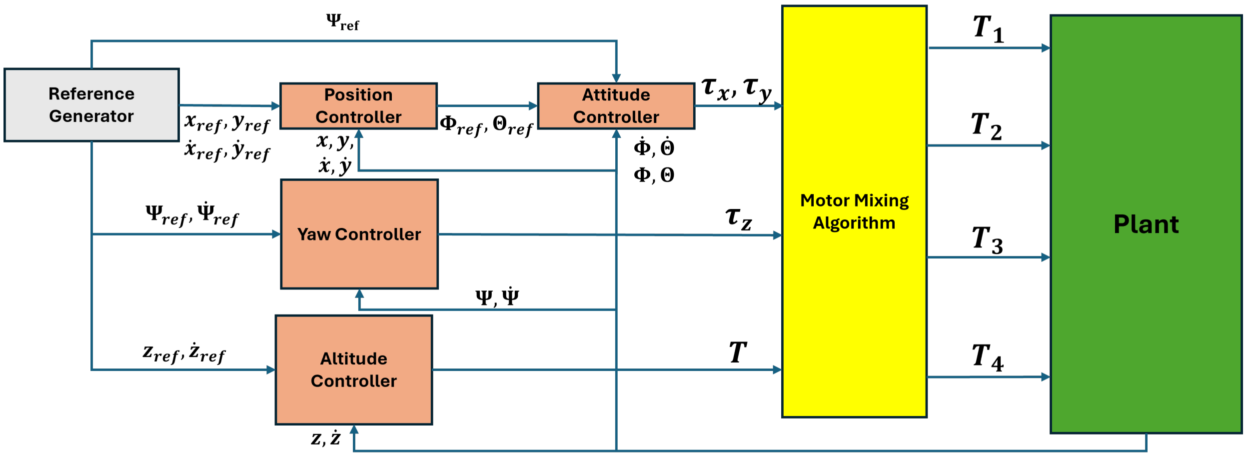


Figure 3–1.: Control Architecture of Cascaded Control

3.6. Position and Altitude Control

The outer-loop controller generates desired attitude and thrust commands based on position and velocity errors:

$$\Phi_d = k_{py}(y_d - y) + k_{dy}(\dot{y}_d - \dot{y}) \tag{3–12}$$

$$\Theta_d = k_{px}(x_d - x) + k_{dx}(\dot{x}_d - \dot{x}) \tag{3–13}$$

$$T_d = mg + k_{pz}(z_d - z) + k_{dz}(\dot{z}_d - \dot{z}) \tag{3–14}$$

where

k_{px}, k_{dx} : PD gains of X position controller
k_{py}, k_{dy} : PD gains of Y position controller
k_{pz}, k_{dz} : PD gains of altitude controller
x_d, y_d, z_d : Desired or reference position signals
$\dot{x}_d, \dot{y}_d, \dot{z}_d$: Desired or reference velocity signals
Φ_d, Θ_d, T_d : Desired or reference roll, pitch and thrust signals

3.7. Attitude Control

The attitude controller stabilises roll, pitch, and yaw angles using angular rate feedback:

$$\tau_x = k_{p\Phi}(\Phi_d - \Phi) + k_{d\Phi}(p_d - p) + k_{i\Phi}\int e_\Phi \, dt \tag{3–15}$$

$$\tau_y = k_{p\Theta}(\Theta_d - \Theta) + k_{d\Theta}(q_d - q) + k_{i\Theta}\int e_\Theta \, dt \tag{3–16}$$

$$\tau_z = k_{p\Psi}(\Psi_d - \Psi) + k_{d\Psi}(r_d - r) + k_{i\Psi}\int e_\Psi \, dt \tag{3–17}$$

where

$k_{p\Phi}, k_{d\Phi}, k_{i\Phi}$: PID gains of roll attitude controller
$k_{p\Theta}, k_{d\Theta}, k_{i\Theta}$: PID gains of pitch attitude controller
$k_{p\Psi}, k_{d\Psi}, k_{i\Psi}$: PID gains of yaw attitude controller
Φ_d, Θ_d, Ψ_d : Desired roll, pitch and yaw Euler angles
$\dot{\Phi}_d, \dot{\Theta}_d, \dot{\Psi}_d$: Desired roll, pitch and yaw Euler rates
e_Φ, e_Θ, e_Ψ : Roll, pitch and yaw Euler angle error

An anti-windup compensation term is implemented to avoid integrator saturation:

$$\dot{I}_\Phi = e_\Phi - k_{aw}\int e_{\Phi_{k-1}}(\tau)d\tau \tag{3–18}$$

$$\dot{I}_\Theta = e_\Theta - k_{aw}\int e_{\Theta_{k-1}}(\tau)d\tau \tag{3–19}$$

where

$\dot{I}_\Phi$: Roll Anti-Windup Term
$\dot{I}_\theta$: Pitch Anti-Windup Term
k_{aw} : Anti-Windup Gain

To rotate the position error to match the drone's heading, a yaw correction matrix R_Ψ is used for desired roll and pitch:

$$\begin{bmatrix}\Theta_d \\ \Phi_d\end{bmatrix} = \begin{bmatrix}k_{p_x} & 0 \\ 0 & k_{py}\end{bmatrix} F_{sat}\left[R_\psi \begin{bmatrix}e_x \\ e_y\end{bmatrix}\right] + \begin{bmatrix}k_{dx} & 0 \\ 0 & k_{dy}\end{bmatrix}\begin{bmatrix}e_{v_x} \\ e_{v_y}\end{bmatrix} \tag{3–20}$$

where

R_ψ : Yaw rotation matrix
e_x : X position error
e_y : Y position error
e_{v_x} : X velocity error
e_{v_y} : Y velocity error

The matrix $R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$ accounts for the yaw when converting position errors into the drone's heading-aligned frame, and F_{sat} is a saturation nonlinearity set at $\pm 3m$ to prevent large error commands [34].

3.8. Numerical Evaluation

In this thesis, the controller uses:

- A PID controller for roll/pitch attitude control
- A PD controller for yaw
- A PD controller for X and Y position
- A PD controller for altitude
- An anti-windup gain for the attitude controller

The different PID gains are summarised in Table 3–2, some of which are referenced from [34].

Table 3–2.: Cascaded PID Controller Gains

Control Loop	Proportional (P)	Integral (I)	Derivative (D)
Position Control			
X-Position	2.15	-	-0.12
Y-Position	1.45	-	0.15
Z-Position	1.5	-	2
Rate/Attitude Control			
Roll	0.013	0.01	0.002
Pitch	0.01	0.01	0.0028
Yaw	0.004	-	0.012
Anti-Windup			
Anti-windup (k_{aw})	0.001		

3.9. Controller Performance Validation

To validate the performance of the controller, a reference rectangular trajectory, which is stretched by a low-frequency sinusoidal wave outlined in [34], is used. The trajectory reference signals are defined as:

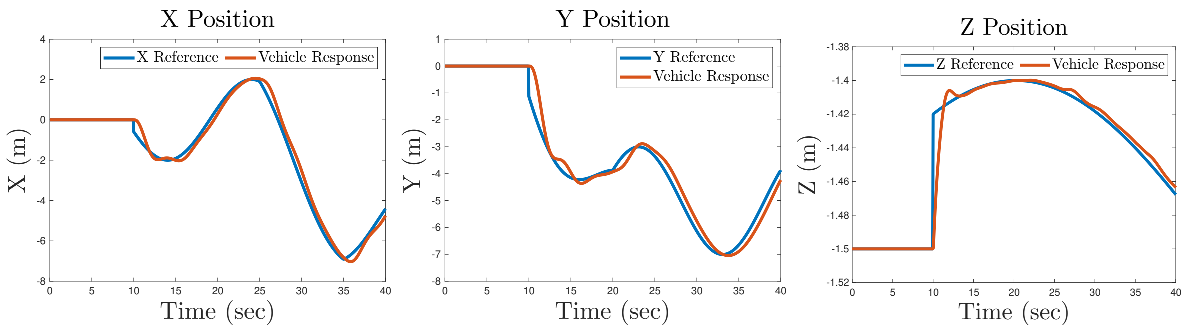
$$X_{ref}(t) = \int_0^t V_{X,0}(\tau)d\tau + 2\sin(0.1\pi t + 0.3) \cdot (t - 10)$$
$$Y_{ref}(t) = \int_0^t V_{Y,0}(\tau)d\tau + 2\sin(0.1\pi t + 0.6) \cdot (t - 10)$$
$$Z_{ref}(t) = -1.5 + 0.1\sin(0.02\pi t + 0.3) \cdot (t - 10)$$

where the velocity components $V_{X,0}(t)$ and $V_{Y,0}(t)$ are defined by the following piecewise functions:

$$V_{X,0}(t) = \begin{cases} -0.5, & t \in [15, 20] \\ 0.5, & t \in [25, 30] \\ 0, & \text{otherwise} \end{cases}$$

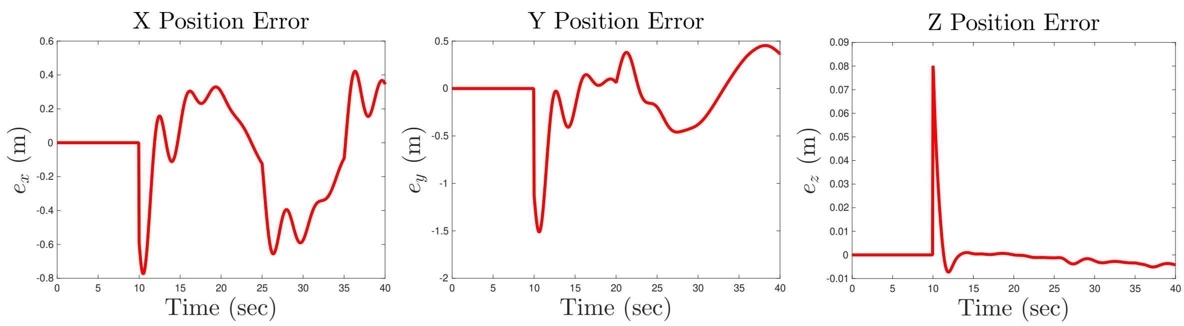
$$V_{Y,0}(t) = \begin{cases} -0.5, & t \in [10, 20] \\ 0.5, & t \in [40, 50] \\ 0, & \text{otherwise} \end{cases}$$

The performance of the implemented PID controller was evaluated through simulation in a trajectory tracking scenario. The results, illustrating the drone's position and attitude over time, are presented in Figures 3–2, and 3–4 and their errors in Figures 3–3, and 3–5.



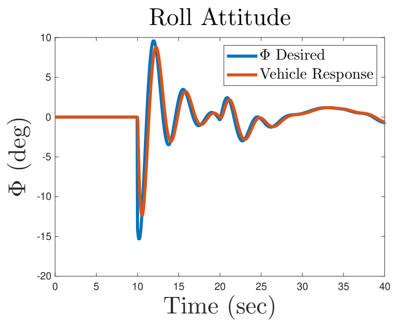
(a) Performance of the controller along x axis **(b) Performance of the controller along y axis** **(c) Performance of the controller along z axis**

Figure 3–2.: Tracking Performances of x y and z positions

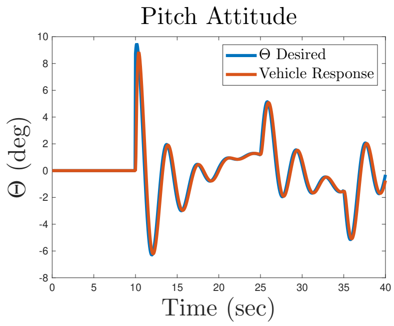


(a) Error of the controller along x axis **(b) Error of the controller along y axis** **(c) Error of the controller along z axis**

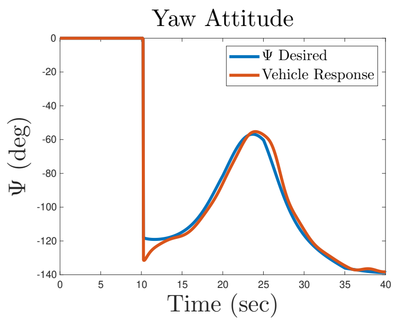
Figure 3–3.: Error of x y and z positions



(a) Performance of the controller along roll

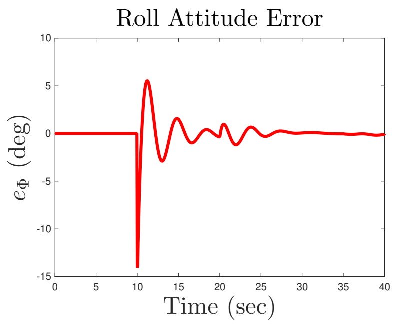


(b) Performance of the controller along pitch

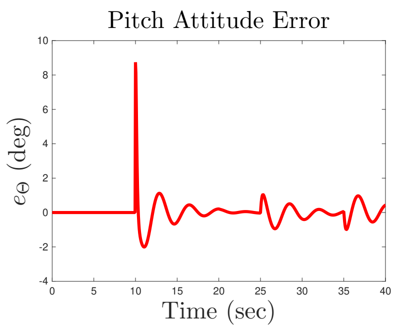


(c) Performance of the controller along yaw

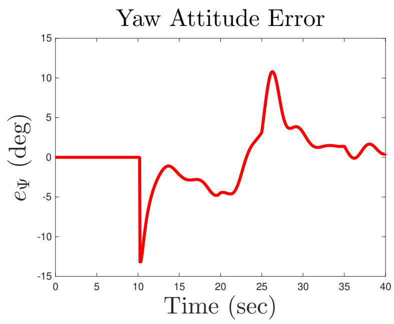
Figure 3–4.: Tracking Performances of roll, pitch and yaw



(a) Error of the controller along roll



(b) Error of the controller along pitch



(c) Error of the controller along yaw

Figure 3–5.: Error of roll, pitch and yaw

The Figure 3–2 shows a noticeable phase lag, where the quadrotor response follows the reference with a small time delay, especially in the x and y position channels. This occurs because the cascaded PID structure does not control lateral motion directly. To move in the x or y direction, the controller must first generate the required roll and pitch angles, and only then does the quadrotor produce the corresponding translational motion. This multi-stage response introduces delay and makes the output trail behind the reference. The phase lag is smaller in the z-axis because altitude is controlled directly through collective thrust. Therefore, the observed phase lag is a normal characteristic of cascaded quadrotor control and reflects the slower outer-loop dynamics compared to the faster inner attitude loop. The attitude responses in Figure 3–4 further clarify the behaviour of the cascaded controller. The roll and pitch angles track their desired values with relatively fast dynamics, although both channels show short transient peaks and damped oscillations after abrupt command changes.

Overall, the results confirm that the cascaded PID controller is effective for basic quadrotor stabilisation and trajectory tracking, and the performance of the controller is according to what is discussed in [34] when compared to the LQR and MPC controllers.

4. Actuator Faults, Sensors and Observers

In this chapter, actuator fault modelling, design of observers and sensor models are introduced.

4.1. Actuator Fault Modelling and its Types

This section defines the types of actuator faults in quadrotors and integrates them into the dynamic model derived in Chapter 2.

Actuator faults in quadrotors can be broadly categorised as described in [5]:

1. Partial Loss of Effectiveness (LOE): A rotor produces less thrust than commanded without complete failure. This can be caused by propeller damage, motor wear, or battery voltage sag.
2. Complete Failure (Lock-in-Place): A rotor stops entirely ($\Omega_i = 0$) and provides zero thrust. This is a severe fault typically caused by motor burnout or electronic speed controller (ESC) failure.
3. Bias or Offset Fault: A constant offset is added to the commanded rotor speed, leading to a persistent thrust error.

This work primarily addresses Partial LOE and Complete Failure, as they are the most critical for vehicle stability [43].

The fault is incorporated into the system as a multiplicative factor on the thrust produced by each rotor. The faulty thrust $\tilde{T}_i$ and faulty torque $\tilde{\tau}_{M_i}$ for the i-th rotor are modeled as [5]:

$$\tilde{T}_i = \xi_i \cdot c_T \rho A_r (\Omega_i^2 R^2) \tag{4–1}$$

$$\tilde{\tau}_{M_i} = \xi_i \cdot c_M \rho A_r (\Omega_i |\Omega_i| R^3) \tag{4–2}$$

where ξ_i is the *fault parameter* for the i-th actuator, defined as:

$$\xi_i = \begin{cases} 1, & \text{for no fault (healthy actuator)} \\ \eta_i, & \text{for Partial LOE, where } 0 < \eta_i < 1 \quad \text{for } t \geq t_f \\ 0, & \text{for Complete Failure} \quad \text{for } t \geq t_f \end{cases} \tag{4–3}$$

Here, t_f is the unknown time of fault occurrence.

4.2. Measured States and Noise

In this thesis, the system states are assumed to be measured and estimated using onboard sensors, including a gyroscope and an inertial measurement unit (IMU). The measured and estimated states are the attitude angles, and angular rates. Parrot Mambo Mini Drone uses MPU6050, manufactured by InvenSense Inc. [41], and common noise magnitudes of MPU6050 are summarised in Table 4–1, which are reported in [44, 45].

$$\begin{aligned} \Phi_m &= \Phi + \nu_\Phi & \Theta_m &= \Theta + \nu_\Theta & \Psi_m &= \Psi + \nu_\Psi \\ \dot{\Phi}_m &= \dot{\Phi} + \nu_{\dot{\Phi}} & \dot{\Theta}_m &= \dot{\Theta} + \nu_{\dot{\Theta}} & \dot{\Psi}_m &= \dot{\Psi} + \nu_{\dot{\Psi}} \end{aligned}$$

where $\nu_{(\cdot)}$ denotes the measurement noise for each state.

The following table summarises the approximate magnitude of noise used in the model.

Table 4–1.: IMU Noise and Attitude Estimation Uncertainty for MPU–6050-Based Systems

Signal	**Description**	**Typical Magnitude**
Gyroscope rates (p, q, r)	Direct sensor measurement (MEMS gyro)	0.009–$0.010\ ^\circ/\mathrm{s}/\sqrt{\mathrm{Hz}}$
Attitude angles (Φ, Θ, Ψ)	Estimated states via IMU sensor fusion	$\pm 1.5^\circ$ (typical accuracy)

These values are used in Section 5.3 for the design of the residual evaluation module.

4.3. Observers and their Design

This section introduces commonly used observers for fault detection [5], including Luenberger, Thau, Extended Kalman Filter and Unscented Kalman Filter.

4.3.1. Luenberger Observer

The Luenberger observer serves as a fundamental model-based approach for state estimation and fault detection in linear systems. For the quadrotor platform, this method provides an effective means of detecting actuator faults by comparing the measured system outputs with those estimated by the observer. The design of the Luenberger observer in this section is derived from [9, 46].

Consider the linear system representation:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \boldsymbol{w}_k \\ \boldsymbol{y}(t) &= \mathbf{C}\boldsymbol{x}(t) + \boldsymbol{v}_k \end{aligned} \tag{4–4}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the commanded input, and $\boldsymbol{y}(t) \in \mathbb{R}^p$ is the measured output, $\boldsymbol{w}_k \sim \mathcal{N}(0, \mathbf{Q})$ and $\boldsymbol{v}_k \sim \mathcal{N}(0, \mathbf{R})$ are process and measurement noise, respectively.

The Luenberger observer is formulated as:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) &= \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{K}[\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)] \\ \hat{\boldsymbol{y}}(t) &= \boldsymbol{C}\hat{\boldsymbol{x}}(t) \end{aligned} \tag{4–5}$$

where $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^n$ is the estimated state vector, $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is the observer gain, and $\hat{\boldsymbol{y}}(t)$ are the estimated measured outputs.

The error dynamics of the Luenberger observer are governed by:

$$\dot{\boldsymbol{x}}(t) - \dot{\hat{\boldsymbol{x}}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) - \mathbf{A}\hat{\boldsymbol{x}}(t) - \mathbf{B}\boldsymbol{u}(t) - \mathbf{K}[\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)] + \boldsymbol{w}_k \tag{4–6}$$

Simplifying Equation 4–6 using Equations 4–4, 4–5, we get:

$$\dot{e}(t) = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})e(t) - \boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k} \tag{4–7}$$

where $e(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ is the state estimation error.

The analysis of the observer error dynamics is essential for understanding the residual components induced by faults and parameter uncertainty, which are presented in Section 5.1 and 5.5.

4.3.2. Thau Observer

The Thau observer represents a significant advancement in model-based fault detection for nonlinear systems, particularly suitable for quadrotor applications where system dynamics exhibit inherent nonlinearities. Unlike linear observers, the Thau observer explicitly accounts for system nonlinearities, making it particularly effective for fault diagnosis in multirotor UAVs [47, 48].

The Thau observer is designed for nonlinear systems that satisfy specific structural conditions. For the quadrotor system described by the nonlinear model in Chapter 2, the observer formulation follows the approach established in [7, 49].

Consider the nonlinear system representation:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}_k \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{v_k} \end{aligned} \tag{4–8}$$

where $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ represents the nonlinear components of the system dynamics.

The Thau observer is formulated as:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) &= \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)) + \boldsymbol{K}(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)) \\ \hat{\boldsymbol{y}}(t) &= \boldsymbol{C}\hat{\boldsymbol{x}}(t) \end{aligned} \tag{4–9}$$

where $\boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))$ represents the estimated nonlinear components of the system dynamics.

The error dynamics of the Thau observer are governed by:

$$\dot{\boldsymbol{x}}(t)-\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))-\boldsymbol{A}\hat{\boldsymbol{x}}(t)-\boldsymbol{B}\boldsymbol{u}(t)-\boldsymbol{h}(\hat{\boldsymbol{x}}(t),\boldsymbol{u}(t))-\boldsymbol{K}(\boldsymbol{y}(t)-\hat{\boldsymbol{y}}(t))+\boldsymbol{w}_k \tag{4–10}$$

Simplifying Equation 4–10 using Equations 4–8, 4–9 we get:

$$\dot{e}(t) = (\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})e(t)+\boldsymbol{h}(\hat{\boldsymbol{x}}(t),\boldsymbol{u}(t))-\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))-\boldsymbol{K}\boldsymbol{v_k}+\boldsymbol{w_k} \tag{4–11}$$

The stability and convergence of the Thau observer depend on two critical conditions [50, 51]:

1. Observability Condition: The pair $(\boldsymbol{C},\boldsymbol{A})$ must be observable
2. Lipschitz Condition: The nonlinear function $\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))$ must be locally Lipschitz with constant γ, satisfying:

$$|\boldsymbol{h}(\boldsymbol{x}_1(t),\boldsymbol{u}(t))-\boldsymbol{h}(\boldsymbol{x}_2(t),\boldsymbol{u}(t))| \le \gamma|\boldsymbol{x}_1(t)-\boldsymbol{x}_2(t)| \tag{4–12}$$

If the observer is stable, then in the fault-free case, the estimation error remains bounded and ideally converges to zero, so the residual stays small. This gives a clear baseline when a fault occurs, the residual departs from zero mainly because of the fault, not because the observer itself is unstable.

The observer gain $\boldsymbol{K}$ is determined through the solution of the Lyapunov equation:

$$\boldsymbol{A}^T\boldsymbol{P}_\zeta+\boldsymbol{P}_\zeta\boldsymbol{A}-\boldsymbol{C}^T\boldsymbol{C}+\zeta\boldsymbol{P}_\zeta=0 \tag{4–13}$$

where $\zeta > 0$ is a design parameter chosen to ensure positive definiteness of $\mathbf{P}_\zeta$, and the gain matrix is computed as:

$$\boldsymbol{K}=\boldsymbol{P}_\zeta^{-1}\boldsymbol{C}^T \tag{4–14}$$

For the quadrotor application, the nonlinear components $\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))$ include the Coriolis forces, gyroscopic effects, and gravitational components, which satisfy the Lipschitz condition within the typical flight envelope [7, 48].

4.3.3. Extended Kalman Filter

The Extended Kalman Filter (EKF) represents a significant extension of the standard Kalman Filter to handle nonlinear systems through linearization techniques.

For quadrotor fault detection, the EKF offers a practical compromise between computational complexity and estimation accuracy. By linearizing the nonlinear dynamics around the current state estimate, the EKF can effectively handle the inherent nonlinearities in quadrotor dynamics while maintaining the recursive structure of the original Kalman Filter.

The EKF is designed for nonlinear discrete-time systems, and the equations are derived from [52, 53]:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k \\ \boldsymbol{y}_k &= h_m(\boldsymbol{x}_k) + \boldsymbol{v}_k \end{aligned} \tag{4–15}$$

where $f(\cdot)$ and $h_m(\cdot)$ are nonlinear functions representing the system dynamics and measurement models, respectively.

The EKF operates similarly to the KF but uses linearised versions of the system matrices computed at each time step.

The prediction step of EKF is:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} &= f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) \\ \boldsymbol{P}_{k|k-1} &= \boldsymbol{J}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{J}_{k-1}^T + \boldsymbol{Q} \end{aligned} \tag{4–16}$$

where $\boldsymbol{P}_{k-1|k-1}$ is covariance matrix of estimated states, and $\boldsymbol{J}_{k-1}$ is the Jacobian matrix of $f(\cdot)$ evaluated at the previous state estimate as:

$$\boldsymbol{J}_{k-1} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}} \tag{4–17}$$

The update step incorporates the measurements through:

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - \boldsymbol{H}_k(\hat{\boldsymbol{x}}_{k|k-1}) \tag{4–18}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R} \tag{4–19}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1} \tag{4–20}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k \tag{4–21}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1} \tag{4–22}$$

where $\tilde{\boldsymbol{y}}_k$ is the innovation term, $\boldsymbol{S}_k$ is the innovation covariance, $\boldsymbol{K}_k$ is the Kalman gain, $\boldsymbol{H}_k$ is the Jacobian matrix of $h_m(\cdot)$ evaluated at the predicted state as:

$$\boldsymbol{H}_k = \left.\frac{\partial h_m}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{4–23}$$

The process noise covariance matrix $\boldsymbol{Q}$ and measurement noise covariance matrix $\boldsymbol{R}$ are critical tuning parameters that govern the Kalman filter's performance [54]. $\boldsymbol{R}$ is typically determined empirically by calculating the variance of sensor measurements under static conditions [55], like hovering in a quadrotor. In contrast, $\boldsymbol{Q}$ is often estimated through analytical modelling of unrepresented disturbances, followed by iterative refinement [56].

The state estimation error of the EKF is governed by:

$$\boldsymbol{e}_{k|k} = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k} \tag{4–24}$$

The predicted error becomes:

$$\boldsymbol{e}_{k|k-1} = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} \tag{4–25}$$

Using Equations 4–15 and 4–16, Equation 4–25 becomes:

$$\boldsymbol{e}_{k|k-1} = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) - f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1} \tag{4–26}$$

By first-order linearization around the current estimate as in Equation 4–17, Equation 4–26 can be approximated as:

$$\boldsymbol{e}_{k|k-1} \approx \boldsymbol{J}_{k-1}\boldsymbol{e}_{k-1|k-1} + \boldsymbol{w}_{k-1} \tag{4–27}$$

Using the EKF measurement update from Equation 4–21, the estimation error after correction is:

$$\boldsymbol{e}_{k|k} = \boldsymbol{e}_{k|k-1} - \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k = \boldsymbol{e}_{k|k-1} - \boldsymbol{K}_k(\boldsymbol{h}_m(\boldsymbol{x}_k) - \boldsymbol{h}_m(\hat{\boldsymbol{x}}_{k|k-1}) + \boldsymbol{v}_k) \tag{4–28}$$

Applying first-order linearization of the measurement model as in Equation 4–23 gives:

$$\boldsymbol{e}_{k|k} \approx \boldsymbol{e}_{k|k-1} - (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{e}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{v}_k \tag{4–29}$$

Hence, the approximate error dynamics as a function of prediction and measurement update equation becomes:

$$\boldsymbol{e}_{k|k} \approx (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{J}_{k-1}\boldsymbol{e}_{k-1|k-1} + (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{w}_{k-1} - \boldsymbol{K}_k\boldsymbol{v}_k \tag{4–30}$$

This shows that the EKF estimation error evolves as a locally linear time-varying stochastic system, whose behaviour depends on the Jacobians and Kalman gain.

4.3.4. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) represents a more recent advancement in nonlinear estimation that addresses some limitations of the EKF. The UKF uses a deterministic sampling approach known as the unscented transform to propagate the mean and covariance through nonlinear transformations. This approach avoids the need for explicit Jacobian calculations and provides more accurate estimation for highly nonlinear systems [57].

The UKF is designed for the same nonlinear discrete-time system as the EKF Equation 4–15. Instead of linearising the nonlinear functions, the UKF uses the unscented transform to approximate the probability distribution of the state as described in [58].

Given an n-dimensional state vector $\boldsymbol{x}$ with mean $\bar{\boldsymbol{x}}$ and covariance $\mathbf{P}$, a set of $2n+1$ sigma points $\mathcal{X}^{(i)}$ are generated as follows:

$$\begin{aligned}
\mathcal{X}^{(0)} &= \bar{\boldsymbol{x}} \\
\mathcal{X}^{(i)} &= \bar{\boldsymbol{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}}\right)_i, \quad i = 1, \dots, n \\
\mathcal{X}^{(i)} &= \bar{\boldsymbol{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}}\right)_{i-n}, \quad i = n+1, \dots, 2n
\end{aligned} \tag{4–31}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter, α controls the spread of the sigma points (typically $10^{-4} \leq \alpha \leq 1$), and κ is a secondary scaling parameter (often set to 0 or $3-n$). The matrix square root $\sqrt{(n+\lambda)\mathbf{P}}$ is computed via Cholesky decomposition.

Prediction Step:

1. **Sigma Point Generation:**

$$\mathcal{X}_{k-1}^{(i)} = \hat{\boldsymbol{x}}_{k-1|k-1} \pm \sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}} \tag{4–32}$$

2. **Propagation through Dynamics:**

$$\mathcal{X}_{k|k-1}^{(i)} = f(\mathcal{X}_{k-1}^{(i)}, \boldsymbol{u}_{k-1}), \quad i = 0, \ldots, 2n \tag{4–33}$$

3. **Predicted State and Covariance:**

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{X}_{k|k-1}^{(i)} \tag{4–34}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}\right)^T + \mathbf{Q} \tag{4–35}$$

Update Step:

1. **Measurement Sigma Points:**

$$\mathcal{Z}_k^{(i)} = h(\mathcal{X}_{k|k-1}^{(i)}), \quad i = 0, \ldots, 2n \tag{4–36}$$

2. **Predicted Measurement and Covariance:**

$$\hat{\boldsymbol{z}}_k = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Z}_k^{(i)} \tag{4–37}$$

$$\mathbf{P}_{zz} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{Z}_k^{(i)} - \hat{\boldsymbol{z}}_k\right)\left(\mathcal{Z}_k^{(i)} - \hat{\boldsymbol{z}}_k\right)^T + \mathbf{R} \tag{4–38}$$

$$\mathbf{P}_{xz} = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\mathcal{Z}_k^{(i)} - \hat{\boldsymbol{z}}_k\right)^T \tag{4–39}$$

3. **Kalman Gain and State Update:**

$$\mathbf{K}_k = \mathbf{P}_{xz}\mathbf{P}_{zz}^{-1} \tag{4–40}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k\left(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k\right) \tag{4–41}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{P}_{zz}\mathbf{K}_k^T \tag{4–42}$$

Weight Calculation:

The weights $W_m^{(i)}$ and $W_c^{(i)}$ are given by:

$$W_m^{(0)} = \frac{\lambda}{n+\lambda} \tag{4–43}$$

$$W_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{4–44}$$

$$W_m^{(i)} = W_c^{(i)} = \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, 2n \tag{4–45}$$

where β incorporates prior knowledge of the distribution (optimally $\beta = 2$ for Gaussian distributions).

Unlike the EKF, the UKF does not explicitly use Jacobian matrices. Instead, the nonlinear error propagation is captured through the unscented transformation [57]. Therefore, the UKF error dynamics are not usually written in an explicit analytical form. The UKF has an error structure similar to the EKF, but the local dynamics are obtained implicitly through sigma points rather than explicit linearization.

5. Actuator Fault Detection and Diagnosis

This chapter presents an evaluation of observer-based fault detection strategies. The performance of the Luenberger, Thau, Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF) is assessed under identical operating conditions, actuator fault and parameter uncertainty scenarios. Furthermore, it introduces the concept of residual generation for different observers and their evaluation for fault detection.

5.1. Residual Generation

This section presents the residual generation principle used for detecting actuator faults. Residuals are constructed as the error between measured outputs and observer-predicted outputs. Under nominal (healthy) operation and with a properly designed observer, the estimation error converges to a small neighbourhood governed by process and measurement noise. In the presence of actuator faults (and, potentially, modelling uncertainty), persistent excitation is injected into the estimation error dynamics, producing residual signals that can be exploited for fault detection.

Faulty Plant Model:

Consider a general nonlinear system with an actuator effectiveness fault:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= f(\boldsymbol{x}(t), \xi \boldsymbol{u}(t)) + \boldsymbol{w_k} \\ \boldsymbol{y}(t) &= h_m(\boldsymbol{x}(t)) + \boldsymbol{v_k} \end{aligned} \tag{5–1}$$

Where the matrix ξ denotes the actuator effectiveness mapping (healthy: $\xi = I$, faulty: $\xi \neq I$).

An observer provides a state estimate $\hat{\boldsymbol{x}}(t)$ and an output estimate $\hat{\boldsymbol{y}}(t)$ of the form:

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) &= f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)) + \boldsymbol{K}(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)) \\ \hat{\boldsymbol{y}}(t) &= h_m(\hat{\boldsymbol{x}}(t)) \end{aligned} \tag{5–2}$$

The residual is defined as:

$$\boldsymbol{r}(t) \triangleq \boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t) \tag{5–3}$$

Subtracting Equation 5–2 from 5–1 yields the general error dynamics of observers as:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = \underbrace{\Big(f(\boldsymbol{x}(t), \xi \boldsymbol{u}(t)) - f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)\Big)}_{\text{fault term}} - \boldsymbol{K}\Big(h_m(\boldsymbol{x}(t)) - h_m(\hat{\boldsymbol{x}}(t))\Big) + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v}_k + \boldsymbol{w}_k\Big)}_{\text{noise term}} \tag{5–4}$$

The residual is directly linked to the estimation error through the measurement map. In the linear output case $h_m(\boldsymbol{x}(t)) = \boldsymbol{C}\boldsymbol{x}(t)$, and Equation 5–3 reduces to:

$$\boldsymbol{r}(t) = \boldsymbol{C}e(t) + \boldsymbol{v_k} \tag{5–5}$$

Equation 5–4 shows that actuator faults (i.e., $\xi \neq I$) introduce an additional excitation term through the error between $\xi\boldsymbol{u}(t)$ and $\boldsymbol{u}(t)$, which prevents $e(t)$ from converging to zero, yielding persistent residuals. Furthermore, it is assumed that $f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))$ and $f(\boldsymbol{x}(t), \boldsymbol{\xi u}(t))$ do not contain any modelling uncertainty for all the observers in this section and is discussed further in Section 5.5.

5.2. Residual Generation for Different Observer Cases

5.2.1. Luenberger Observer (Linear Case)

Using the Equations from Section 4.3.1 and Equation 5–4 for a linear case, the error dynamics of a faulty plant for a Luenberger observer can be formulated as:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = \underbrace{\Big((\boldsymbol{A} - \boldsymbol{KC})\boldsymbol{e}(\boldsymbol{t})\Big)}_{\text{nominal dynamics}} + \underbrace{\Big(\boldsymbol{B}(\xi\boldsymbol{u}(t) - \boldsymbol{u}(t))\Big)}_{\text{fault term}} + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k}\Big)}_{\text{noise term}} \tag{5–6}$$

In a healthy case ($\xi = I$):

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = (\boldsymbol{A} - \boldsymbol{KC})\boldsymbol{e}(\boldsymbol{t}) - \boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k} \tag{5–7}$$

If $\boldsymbol{K}$ is chosen such that $(\boldsymbol{A} - \boldsymbol{KC})$ is stable, then $e(t) \to 0$ in the noise-free case, and the residual is small and bounded. In the faulty case ($\xi \neq I$), the term $\boldsymbol{B}(\xi\boldsymbol{u}(t) - \boldsymbol{u}(t))$ produces a persistent excitation, resulting in non-zero residuals.

5.2.2. Thau Observer (Nonlinear Case)

Using the Equations from Section 4.3.2 and Equation 5–4 for a non-linear case, the error dynamics of a faulty plant for a Thau observer can be formulated as:

$$\dot{\boldsymbol{e}}(t) = \underbrace{\Big((\boldsymbol{A} - \boldsymbol{KC})\boldsymbol{e}(\boldsymbol{t})\Big)}_{\text{nominal dynamics}} + \underbrace{\Big(\boldsymbol{B}(\xi\boldsymbol{u}(t) - \boldsymbol{u}(t)) + \boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t)) - \boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))\Big)}_{\text{fault term}} + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k}\Big)}_{\text{noise term}} \tag{5–8}$$

In the faulty case the additional term $B(\xi\boldsymbol{u}(t) - \boldsymbol{u}(t)) + \boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}) - \boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))$ drives sustained residual growth.

5.2.3. Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF):

Using the Equations from Section 4.3.3 and 4.3.4 and Equation 5–4 for a non-linear case, the error dynamics of a faulty plant for an Extended Kalman Filter can be formulated as:

$$\boldsymbol{e_{k|k-1}} = f(\boldsymbol{x_{k-1}}, \xi\boldsymbol{u_{k-1}}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{k-1|k-1}}, \boldsymbol{u_{k-1}}) + \boldsymbol{w_{k-1}} \tag{5–9}$$

Using first-order linearization around the healthy operating point gives:

$$\boldsymbol{e_{k|k-1}} \approx \boldsymbol{J_{x,k-1}}\boldsymbol{e_{k-1|k-1}} + \boldsymbol{J_{u,k-1}}(\boldsymbol{\xi u_{k-1}} - \boldsymbol{u_{k-1}}) + \boldsymbol{w_{k-1}} \tag{5–10}$$

where

$$\boldsymbol{J}_{x,k-1} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1},\boldsymbol{u}_{k-1}}$$
$$\boldsymbol{J}_{u,k-1} = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1},\boldsymbol{u}_{k-1}}$$

Since the fault is in the actuator and not in the sensor, the measurement update remains the same as in Equation 4–29. Substituting the prediction error from Equation 5–10 into Equation 4–29 yields:

$$\boldsymbol{e}_{k|k} \approx \boldsymbol{e}_{k|k-1} - (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{e}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{v}_k \tag{5–11}$$

Hence, the approximate error dynamics as a function of prediction and measurement update equation becomes:

$$\boldsymbol{e}_{k|k} \approx \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{J}_{x,k-1}\boldsymbol{e}_{k-1|k-1}}_{\text{nominal dyanmics}} + \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{J}_{u,k-1}(\boldsymbol{\xi}\boldsymbol{u}_{k-1} - \boldsymbol{u}_{k-1})}_{\text{fault term}} + \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{w}_{k-1} - \boldsymbol{K}_k\boldsymbol{v}_k}_{\text{noise term}} \tag{5–12}$$

For the UKF, the actuator effectiveness fault is modeled through the faulty input $\boldsymbol{u}_k = \xi\boldsymbol{u}_k$, while the filter prediction is still based on the healthy input $\boldsymbol{u}_k$. As a result, an error arises between the actual plant propagation and the UKF predicted state. Unlike the EKF, the UKF does not express this error through explicit Jacobians and is captured implicitly by the unscented transformation.

5.3. Residual Evaluation

For each observer, the following residuals are evaluated for the respective states:

$$\boldsymbol{r}(t) = \begin{bmatrix} \boldsymbol{r}_{\dot{\Phi}} & \boldsymbol{r}_{\dot{\Theta}} & \boldsymbol{r}_{\dot{\Psi}} & \boldsymbol{r}_{\Phi} & \boldsymbol{r}_{\Theta} & \boldsymbol{r}_{\Psi} & \end{bmatrix}^T \tag{5–13}$$

These states are directly influenced by actuator faults and are therefore highly sensitive to changes in thrust and torque generation [59]. Under nominal conditions, the residuals remain bounded around zero due to noise. The occurrence of an actuator fault causes a significant deviation in the residual magnitude, enabling fault detection through threshold comparison. The residual evaluation module in this section is derived using the approaches of [7, 49].

A fault is declared when the decision function exceeds a predetermined threshold:

$$J(t) > J_{th} \tag{5–14}$$

where J_{th} is selected to balance false alarm rates and detection sensitivity. If the threshold is set too conservatively, nominal residual variations, as shown in Figure 5–1, may exceed it and lead to false alarms. Conversely, an excessively high threshold increases the detection time, since the residual must accumulate longer before crossing the decision limit. In this thesis, a constant decision function is used for the residuals. Its selection depends on the system dynamics, observer gains, and sensor noise characteristics, as reflected in the residual generation equations in Section 5.1. Furthermore, this choice can be justified through the observer error dynamics, since the residual arises directly from the estimation error and its propagation. In particular, the error dynamics determine how disturbances, modelling uncertainties, and faults are transmitted to the residual, thereby influencing its transient response, steady-state magnitude, and sensitivity to different fault scenarios. Therefore, analyzing the observer error dynamics under both nominal and faulty conditions provides a basis for selecting a decision function that achieves an effective trade-off between fault sensitivity and robustness to noise and uncertainties.

The following decision functions are used for the respective residuals based on the residuals near the hover condition. The residuals of the observers under the nominal hover condition are shown in Figure 5–1.

$$J(t) = \begin{cases} -0.15 \leq r_{\Phi}, r_{\Theta}, r_{\Psi} \leq 0.15 \\ -0.008 \leq r_{\dot{\Phi}}, r_{\dot{\Theta}}, r_{\dot{\Psi}} \leq 0.008 \end{cases} \tag{5–15}$$

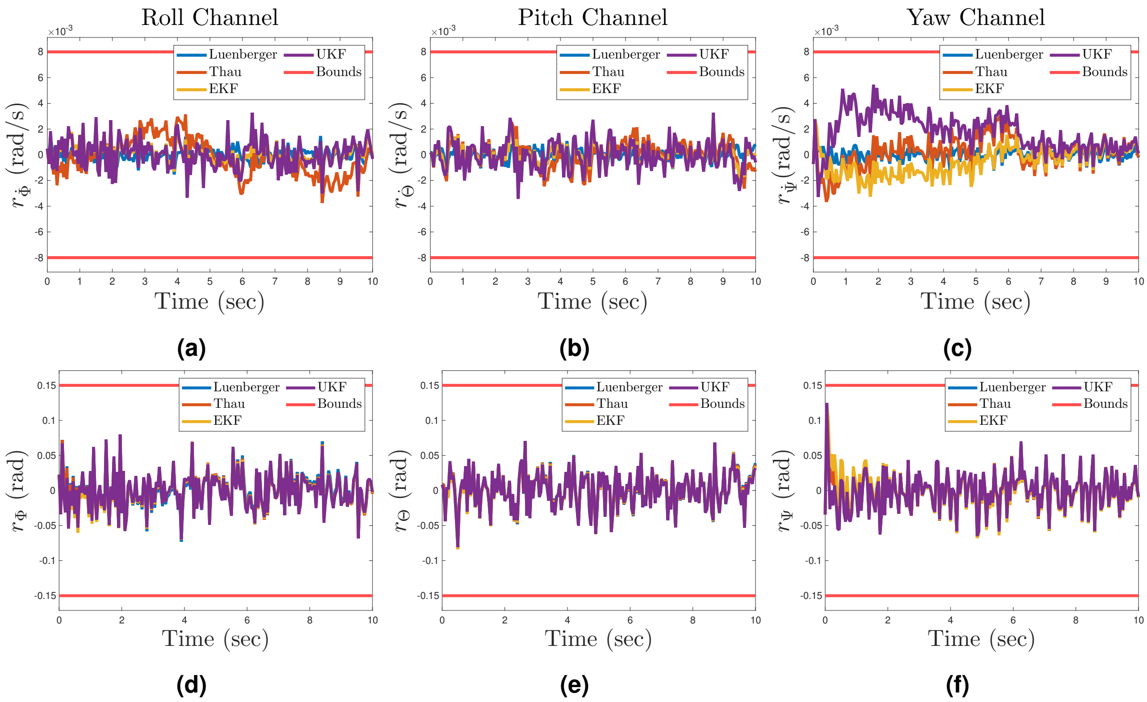


Figure 5–1.: Nominal Residuals of Observers During Hover

5.4. Actuator Fault Scenarios

The fault detection performance of the considered observers was evaluated using four actuator fault scenarios involving different magnitudes of actuator effectiveness loss during hover, and they are summarised in Table 5–1.

Table 5–1.: Actuator Fault Scenarios

Fault ID	Actuator	Fault Type	Magnitude	Injection Time $t_f(s)$
F1	Motor 1	Loss of Effectiveness	50%	12
F2	Motor 1	Loss of Effectiveness	100%	12
F3	Motor 2	Loss of Effectiveness	50%	12
F4	Motor 2	Loss of Effectiveness	100%	12

The fault detection times for each observer are reported in Tables 5–2 to 5–5. In all scenarios, the fault injection time was set to $t_f = 12\,\mathrm{s}$, and detection time was defined as the instant when the residual exceeded its predefined threshold. The residuals for actuator fault scenario F1 and F2 are shown in Figure 5–2 and 5–3.

Table 5–2.: Fault Detection Time (s) for Scenario F1

Method	Φ	$\dot{\Phi}$	Θ	$\dot{\Theta}$	Ψ	$\dot{\Psi}$
Luenberger	12.6	12.05	12.9	12.05	12.65	12.05
Thau	12.1	12.1	12.2	12.1	12.3	12.1
EKF	12.2	12.1	12.2	12.1	12.5	12.1
UKF	12.2	12.1	12.2	12.1	12.2	12.1

Table 5–3.: Fault Detection Time (s) for Scenario F2

Method	Φ	$\dot{\Phi}$	Θ	$\dot{\Theta}$	Ψ	$\dot{\Psi}$
Luenberger	12.1	12.05	12.15	12.05	12.15	12.05
Thau	12.1	12.1	12.1	12.1	12.2	12.1
EKF	12.1	12.1	12.1	12.1	12.2	12.1
UKF	12.1	12.1	12.1	12.1	12.2	12.1

Table 5–4.: Fault Detection Time (s) for Scenario F3

Method	Φ	$\dot{\Phi}$	Θ	$\dot{\Theta}$	Ψ	$\dot{\Psi}$
Luenberger	12.7	12.05	12.9	12.05	12.75	12.05
Thau	12.10	12.1	12.2	12.1	12.3	12.10
EKF	12.2	12.1	12.2	12.1	12.5	12.1
UKF	12.2	12.1	12.2	12.1	12.2	12.1

Table 5–5.: Fault Detection Time (s) for Scenario F4

Method	Φ	$\dot{\Phi}$	Θ	$\dot{\Theta}$	Ψ	$\dot{\Psi}$
Luenberger	12.15	12.05	12.15	12.05	12.15	12.05
Thau	12.1	12.1	12.1	12.1	12.2	12.1
EKF	12.1	12.1	12.1	12.1	12.2	12.1
UKF	12.1	12.1	12.1	12.1	12.2	12.1

The results in Figure 5–2 and 5–3 show that during Scenario F1 and F2, the observer residuals remain centred close to zero and stay within the predefined bounds. Clear change appears at the fault instant around $t_f = 12\text{s}$ (marked by the dashed vertical line). The most fault-sensitive residual channels, particularly $r_{\dot{\Phi}}$,$r_{\dot{\Theta}}$,$r_{\dot{\Psi}}$ exhibited sharp excursions. By contrast, the residuals in r_{Φ},r_{Θ},r_{Ψ} showed slightly slower progression towards the bounds during scenario F1.

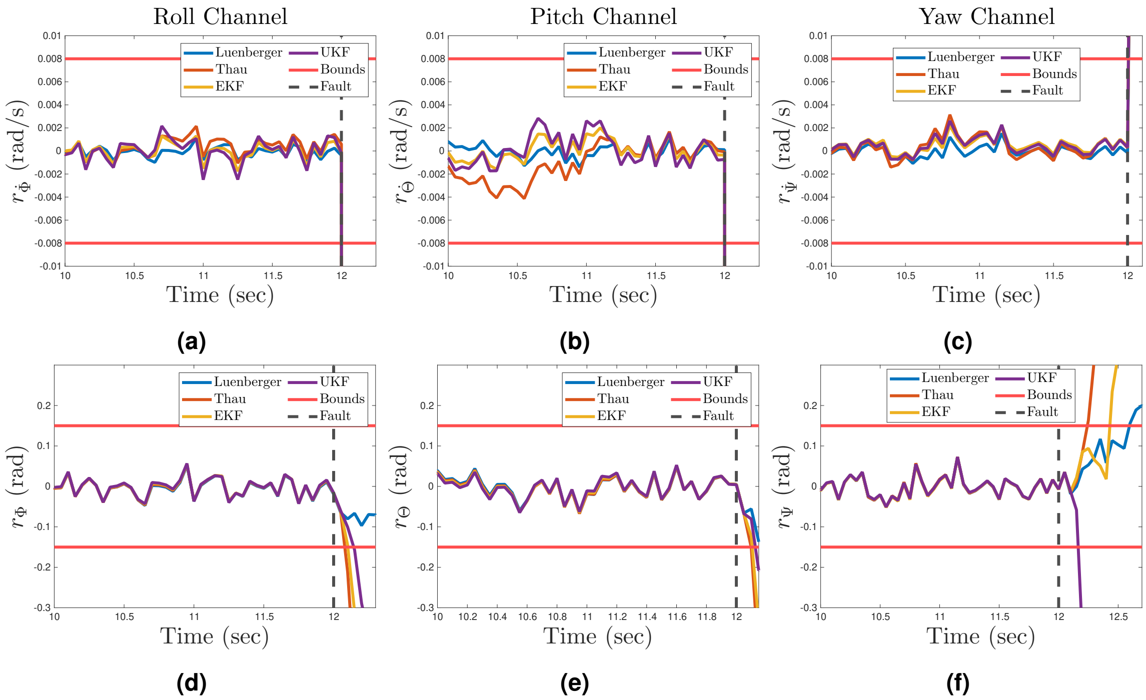


Figure 5–2.: Residuals of Observers During Scenario F1

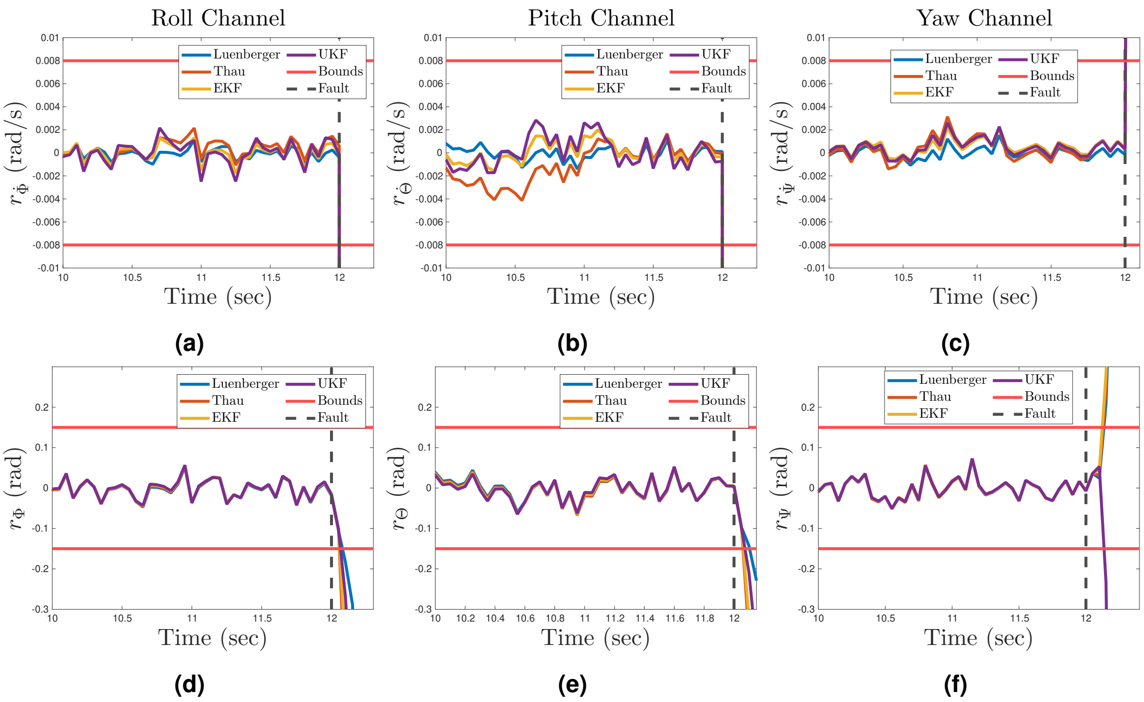


Figure 5–3.: Residuals of Observers During Scenario F2

5.5. Modelling Uncertainty

In practical applications, accurate knowledge of system parameters, such as moments of inertia or mass, are not guaranteed. Variations may arise due to payload changes, battery depletion, manufacturing tolerances, or environmental effects. Consequently, a reliable fault detection scheme must remain robust against modelling uncertainties while avoiding false alarms under nominal operating conditions.

To evaluate robustness, the performance of all observers was tested under parametric uncertainties, while no actuator fault was present. The uncertain parameters were modelled as multiplicative deviations from their nominal values, expressed as:

$$\tilde{I}_{xx} = I_{xx}(1 \pm \sigma_{I_{xx}}), \quad \tilde{I}_{yy} = I_{yy}(1 \pm \sigma_{I_{yy}}), \quad \tilde{I}_{zz} = I_{zz}(1 \pm \sigma_{I_{zz}}) \tag{5–16}$$

Where $\sigma_{I_{xx}}$, $\sigma_{I_{yy}}$ and $\sigma_{I_{zz}}$ represents the percentage uncertainty in inertia.

5.5.1. Unified Error Dynamics: Faults and Parameter Uncertainty

Consider the nonlinear faulty system with parameter uncertainty:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= f(\boldsymbol{x}(t), \xi \boldsymbol{u}(t), \tilde{\theta}) + \boldsymbol{w}_k \\ \boldsymbol{y}(t) &= h_m(\boldsymbol{x}(t)) + \boldsymbol{v}_k \end{aligned} \tag{5–17}$$

where the uncertain parameter set is:

$$\tilde{\theta} \triangleq \{\tilde{I}_{xx}, \tilde{I}_{yy}, \tilde{I}_{zz}\} \tag{5–18}$$

The implemented observers uses the nominal parameters θ and nominal input $\boldsymbol{u}(t)$:

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t) &= f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), \theta) + \boldsymbol{K}(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)) \\ \hat{\boldsymbol{y}}(t) &= h_m(\hat{\boldsymbol{x}}(t))\end{aligned} \tag{5–19}$$

The general error dynamics can be described as:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) - f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), \theta) - \boldsymbol{K}\Big(h_m(\boldsymbol{x}(t)) - h_m(\hat{\boldsymbol{x}}(t))\Big) - \boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w}_k \tag{5–20}$$

To separate nominal terms from faults and uncertainty, add and subtract $f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta)$:

$$\begin{aligned}\dot{\boldsymbol{e}}(\boldsymbol{t}) = &\underbrace{\Big(f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta) - f(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), \theta) - \boldsymbol{K}(h_m(\boldsymbol{x}(t)) - h_m(\hat{\boldsymbol{x}}(t)))\Big)}_{\text{nominal dynamics}} \\ &+ \underbrace{\Big(f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) - f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta)\Big)}_{\text{fault + parameter uncertainty term}} + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k}\Big)}_{\text{noise term}}\end{aligned} \tag{5–21}$$

The fault and parameter uncertainty term decomposes into:

$$\begin{aligned}\Big(f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) - f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta)\Big) = &\underbrace{\Big(f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \theta) - f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta)\Big)}_{\text{actuator fault contribution}} \\ &+ \underbrace{\Big(f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) - f(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \theta)\Big)}_{\text{parameter uncertainty contribution}}\end{aligned} \tag{5–22}$$

Therefore, even in the healthy actuator case ($\xi = I$), parameter uncertainty yields a non-zero excitation of the error dynamics. This expression highlights that uncertainty behaves like an additional structured input to the system, generating persistent residuals.

5.5.2. Luenberger Observer with Faults and Parameter Uncertainty

Consider the linear system with faults and parameter uncertainty:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} + \boldsymbol{\Delta A})\,\boldsymbol{x}(t) + (\boldsymbol{B} + \boldsymbol{\Delta B})\,\xi\boldsymbol{u}(t) + \boldsymbol{w}_k \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{v}_k\end{aligned} \tag{5–23}$$

The resulting error dynamics for a Luenberger Observer are:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = \underbrace{\Big((\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\boldsymbol{e}(\boldsymbol{t})\Big)}_{\text{nominal dynamics}} + \underbrace{\Big(\boldsymbol{B}(\xi\boldsymbol{u}(t) - \boldsymbol{u}(t))\Big)}_{\text{fault term}} + \underbrace{\Big(\boldsymbol{\Delta A}\,\boldsymbol{x}(t) + \boldsymbol{\Delta B}\xi\boldsymbol{u}(t)\Big)}_{\text{parameter uncertainty}} + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k}\Big)}_{\text{noise term}} \tag{5–24}$$

Healthy actuators ($\xi = I$) yield:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\boldsymbol{e}(\boldsymbol{t}) + \boldsymbol{\Delta A}\,\boldsymbol{x}(t) + \boldsymbol{\Delta B}\,\boldsymbol{u}(t) - \boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k} \tag{5–25}$$

This shows that modelling uncertainty generates non-zero residuals even in the absence of faults.

5.5.3. Thau Observer with Faults and Parameter Uncertainty

Consider the nonlinear system with faults and parameter uncertainty:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\xi\boldsymbol{u}(t) + \boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) + \boldsymbol{w}_k \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{v_k}\end{aligned} \tag{5–26}$$

The resulting error dynamics for a Thau Observer are:

$$\dot{\boldsymbol{e}}(\boldsymbol{t}) = (\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})\boldsymbol{e}(\boldsymbol{t}) + \boldsymbol{B}(\xi\boldsymbol{u}(t)-\boldsymbol{u}(t)) + \boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \tilde{\theta}) - \boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), \theta) - \boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k} \tag{5–27}$$

Add and subtract $\boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \theta)$:

$$\begin{aligned}\dot{e}(t) = {} & \underbrace{\Big((\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})e(t)\Big)}_{\text{nominal dynamics}} + \underbrace{\Big(\boldsymbol{B}(\xi\boldsymbol{u}(t)-\boldsymbol{u}(t))\Big)}_{\text{fault term}} + \underbrace{\Big(\boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \theta) - \boldsymbol{h}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), \theta)\Big)}_{\text{bounded via Lipschitz}} \\ & + \underbrace{\Big(\boldsymbol{h}(\boldsymbol{x}(t), \xi u(t), \tilde{\theta}) - \boldsymbol{h}(\boldsymbol{x}(t), \xi\boldsymbol{u}(t), \theta)\Big)}_{\text{parameter uncertainty term}} + \underbrace{\Big(-\boldsymbol{K}\boldsymbol{v_k} + \boldsymbol{w_k}\Big)}_{\text{noise term}}\end{aligned} \tag{5–28}$$

Hence, Thau observer residuals are driven both by actuator effectiveness loss and by parametric error.

5.5.4. EKF and UKF with Faults and Parameter Uncertainty

Consider the discrete-time nonlinear system with faults and parameter uncertainty:

$$\begin{aligned}\boldsymbol{x_{k+1}} &= f(\boldsymbol{x_k}, \xi\boldsymbol{u_k}, \tilde{\theta}_k) + \boldsymbol{w_k}, \\ \boldsymbol{y_k} &= h_m(\boldsymbol{x_k}) + \boldsymbol{v_k}.\end{aligned} \tag{5–29}$$

The EKF prediction uses the nominal model:

$$\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} = f(\hat{\boldsymbol{x}}_{\boldsymbol{k-1|k-1}}, \boldsymbol{u_{k-1}}, \theta_{k-1}) \tag{5–30}$$

The predicted error becomes:

$$\boldsymbol{e_{k|k-1}} = f(\boldsymbol{x_{k-1}}, \xi_{k-1}\boldsymbol{u_{k-1}}, \tilde{\theta}_{k-1}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{k-1|k-1}}, \boldsymbol{u_{k-1}}, \theta_{k-1}) + \boldsymbol{w}_{k-1} \tag{5–31}$$

Using first-order linearization around the nominal operating point gives:

$$\boldsymbol{e_{k|k-1}} \approx \boldsymbol{J_{x,k-1}}\boldsymbol{e_{k-1|k-1}} + \boldsymbol{J_{u,k-1}}(\boldsymbol{\xi u_{k-1}} - \boldsymbol{u_{k-1}}) + \boldsymbol{J_{\theta,k-1}}\tilde{\boldsymbol{\theta}}_{\boldsymbol{k-1}} + \boldsymbol{w_{k-1}} \tag{5–32}$$

where

$$J_{\theta,k-1} = \frac{\partial f}{\partial \theta}\bigg|_{\hat{x}_{k-1|k-1},\, u_{k-1},\, \hat{\theta}_{k-1}}$$

Hence, the approximate error dynamics as a function of prediction and measurement update utilising Equations from Section 5.1and 4.3 becomes:

$$\boldsymbol{e}_{k|k} \approx \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{J}_{x,k-1}\boldsymbol{e}_{k-1|k-1}}_{\text{nominal dynamics}} + \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\Big[\boldsymbol{J}_{u,k-1}(\boldsymbol{\xi u}_{k-1} - \boldsymbol{u}_{k-1}) + \boldsymbol{J}_{\theta,k-1}\tilde{\boldsymbol{\theta}}_{k-1}\Big]}_{\text{fault and parameter uncertainty term}} + \underbrace{(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{w}_{k-1} - \boldsymbol{K}_k \boldsymbol{v}_k}_{\text{noise term}} \tag{5–33}$$

Equation 5–33 is an extension of Equation 5–12, which includes the parametric uncertainty terms in addition to the fault term.

For the UKF, the actuator effectiveness fault and parameter uncertainty are modeled through the faulty input $\boldsymbol{u_k} = \xi \boldsymbol{u_k}$, and uncertain parameter $\tilde{\theta}$ while the filter prediction is still based on the healthy input $\boldsymbol{u_k}$ and nominal parameters θ. As a result, an error arises between the actual plant propagation and the UKF predicted state.

5.5.5. Dynamics of Parameter Uncertainty

This section makes the parameter uncertainty explicit by showing how $\tilde{I}_{xx}$, $\tilde{I}_{yy}$ and $\tilde{I}_{zz}$ affect quadrotor dynamics.

Roll Rate Dynamics:

The true roll-rate dynamics are:

$$\dot{p} = \frac{1}{\tilde{I}_{xx}}\Big(\tau_x + (\tilde{I}_{yy} - \tilde{I}_{zz})qr\Big) \tag{5–34}$$

while the nominal model uses:

$$\dot{p}_{\text{nom}} = \frac{1}{I_{xx}}\Big(\tau_x + (I_{yy} - I_{zz})qr\Big) \tag{5–35}$$

The exact error is:

$$\Delta\dot{p} = \frac{1}{\tilde{I}_{xx}}\Big(\tau_x + (\tilde{I}_{yy} - \tilde{I}_{zz})qr\Big) - \frac{1}{I_{xx}}\Big(\tau_x + (I_{yy} - I_{zz})qr\Big) \tag{5–36}$$

Pitch Rate Dynamics:

The true pitch-rate dynamics are:

$$\dot{q} = \frac{1}{\tilde{I}_{yy}}\Big(\tau_y + (\tilde{I}_{zz} - \tilde{I}_{xx})pr\Big) \tag{5–37}$$

and the nominal model uses:

$$\dot{q}_{\text{nom}} = \frac{1}{I_{yy}}\Big(\tau_y + (I_{zz} - I_{xx})pr\Big) \tag{5–38}$$

The exact error is:

$$\Delta\dot{q} = \frac{1}{\tilde{I}_{yy}}\Big(\tau_y + (\tilde{I}_{zz} - \tilde{I}_{xx})pr\Big) - \frac{1}{I_{yy}}\Big(\tau_y + (I_{zz} - I_{xx})pr\Big) \tag{5–39}$$

Substituting $\tilde{I}_{xx} = I_{xx}(1 \pm \sigma_{I_{xx}})$, $\tilde{I}_{yy} = I_{yy}(1 \pm \sigma_{I_{yy}})$, and $\tilde{I}_{zz} = (1 \pm \sigma_{I_{zz}})$ shows how inertia uncertainty introduces a structured excitation into roll-rate and pitch-rate estimation error and thus into attitude-related residuals.

These terms directly change the angular acceleration response to the same applied torques and rate products, increasing residual sensitivity during manoeuvres or when uncertainties are large.

5.5.6. Practical Implications for Fault Detection

With actuator faults and parameter uncertainty present, residuals can be interpreted as:

$$\boldsymbol{r}(t) = \boldsymbol{r}_{\text{nominal}}(t) + \boldsymbol{r}_{\text{noise}}(t) + \boldsymbol{r}_{\text{fault}}(t, \xi) + \boldsymbol{r}_{\text{param}}(t, \tilde{\theta}) \tag{5–40}$$

where $\boldsymbol{r}_{\text{param}}$ may be non-zero even for $\xi = I$. Consequently, thresholds and residual evaluation tuned only for noise may yield false alarms under modelling uncertainty.

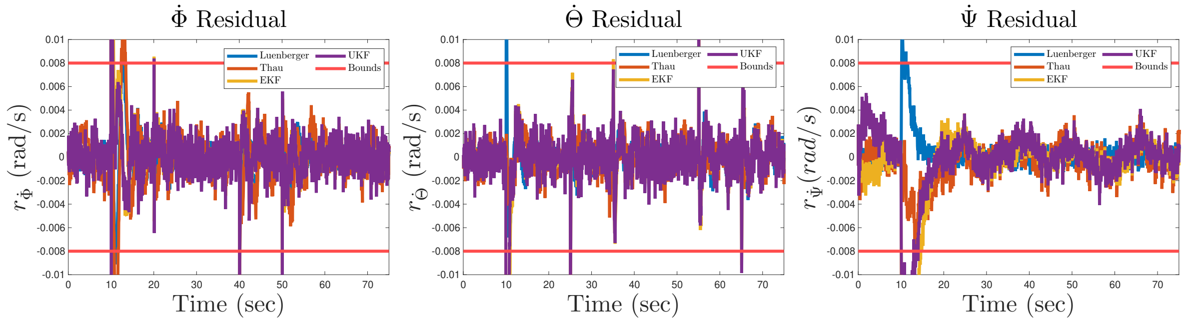
5.5.7. Parameter Uncertainty Scenarios

To simulate the parametric uncertainty without actuator faults, the moments of inertia were varied, and their effect on residuals was evaluated during a trajectory highlighted in Section 3.9. The scenarios are summarised in Table 5–6.

Table 5–6.: Parameter Uncertainty Scenarios

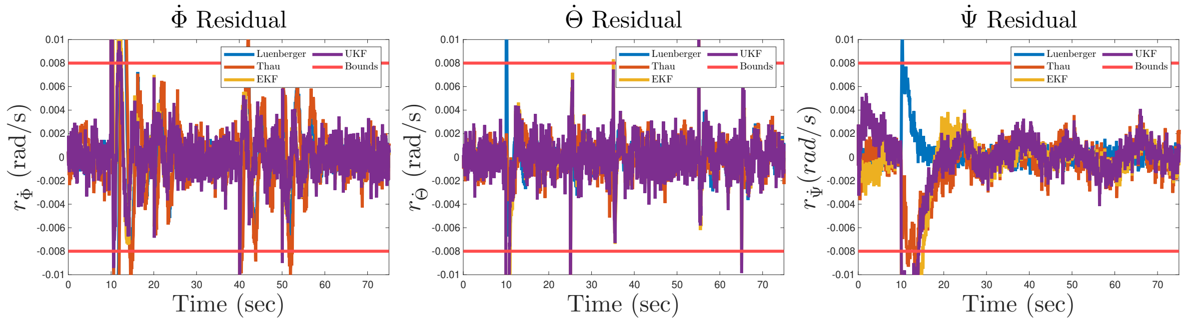
Scenario ID	Parameter	Magnitude
U1	$\sigma_{I_{xx}}$	5%
U2	$\sigma_{I_{xx}}$	10%
U3	$\sigma_{I_{yy}}$	5%
U4	$\sigma_{I_{yy}}$	10%

The residuals for angular rates for scenario U1 and U2 are shown in Figures 5–4 and 5–5.



(a) (b) (c)

Figure 5–4.: Residuals of Observers During Scenario U1



(a) (b) (c)

Figure 5–5.: Residuals of Observers During Scenario U2

Figures 5–4, and 5–5 show that residuals are very sensitive to parametric uncertainty, causing spikes and false alarms during manoeuvres, because the thresholds are constant, and the residual evaluation module is only designed for noise.

The false alarm rate for the parametric uncertainty scenarios was calculated as follows:

$$FAR\,\% = \frac{N_{false\ alarms}}{N_{healthy\ samples}} * 100 \tag{5–41}$$

A total of 2000 samples were evaluated with a sampling time of $T_s = 0.05s$. The Tables 5–7 and 5–8 summarise the FAR for each of the observers in Scenario U1 and U2.

Table 5–7.: FAR During Scenario U1

Observer	$FAR_{\dot{\Phi}}$	$FAR_{\dot{\Theta}}$	$FAR_{\dot{\Psi}}$
Luenberger	0.25%	0.05%	0.35%
Thau	0.35%	0%	0.05%
EKF	0.2%	0.25%	0.2%
UKF	0.3%	0.2%	0.2%

Table 5–8.: FAR During Scenario U2

Observer	$FAR_{\dot{\Phi}}$	$FAR_{\dot{\Theta}}$	$FAR_{\dot{\Psi}}$
Luenberger	0.35%	0.05%	0.1%
Thau	0.65%	0%	0.35%
EKF	0.2%	0.25%	0.05%
UKF	0.35%	0.25%	0.1%

The false alarm rates in both scenarios remain low for all observers, with all values below 1%, indicating that the residual thresholds are well selected and that healthy operation rarely triggers incorrect fault declarations. Since 2000, samples were evaluated, even the largest FAR of 0.65% corresponds to only 13 false-alarm samples. Overall, the results show that all observers maintain acceptable false alarm performance, which confirms that the selected thresholds are not overly conservative.

5.5.8. Modelling Complexity of Observers

An important criterion for selecting a fault detection observer is the modelling complexity required for successful implementation. Table 5–9 summarises the relative modelling complexity of the evaluated observers.

Table 5–9.: Modelling Complexity Comparison of Observers

Observer	Linear Model	Nonlinear Model	Noise Model	Jacobians
Luenberger	Required	Not required	Not required	Not required
Thau	Required for gain	Required	Not required	Not required
EKF	Not required	Required	Required	Required
UKF	Not required	Required	Required	Not required

In this thesis, the term modelling complexity refers to the effort required to derive, tune, and implement an observer model for fault detection. The comparison is made qualitatively using the requirements for (i) linear or nonlinear process models, (ii) noise covariance modelling, and (iii) Jacobian derivation. Under these criteria, the Luenberger observer has the lowest modelling burden, since it only requires a linearised model and observer gain selection. The Thau observer has moderate complexity because it requires a nonlinear model but does not require explicit noise covariance modelling or Jacobian computation. The EKF and UKF impose higher complexity due to the need for nonlinear state propagation and noise covariance tuning. In addition, the EKF requires Jacobian computation, while the UKF avoids Jacobians at the expense of increased online computation by sigma point propagation.

6. Conclusion and Results

This thesis investigated model-based actuator fault detection strategies for a quadrotor UAV in hover using observer-generated residuals and threshold-based decision logic. A nonlinear quadrotor model was implemented and linearised around hover to support both linear and nonlinear observer designs. A baseline cascaded PID controller ensured stable nominal flight, providing a consistent reference for injecting actuator loss-of-effectiveness faults and evaluating detection performance. Four observer-based schemes were designed and compared: a Luenberger observer, a Thau observer, an Extended Kalman Filter (EKF), and an Unscented Kalman Filter (UKF). Faults were modelled through an actuator effectiveness mapping that reduces the contribution of the rotor to the overall force and torque generation. Residuals were computed as the error between measured outputs and observer-predicted outputs, and a fault was declared when the residual decision logic exceeded predefined thresholds. Finally, the effect of parametric uncertainty was explored on residuals and false alarm rate.

6.1. Key Findings

Simulation results showed that observer-based residual generation is effective for detecting actuator faults under nominal conditions. Detection typically occurred shortly after the fault injection time $t_f = 12\,\mathrm{s}$ when the fault produced sufficient excitation in the measured states. The angular rates were the fastest to detect faults, as they are most sensitive to actuator faults. The parametric uncertainty in the model caused persistent residuals even in the healthy case scenarios. These uncertainties caused false alarms during manoeuvres, which involved sudden changes in roll rate and pitch rate.

6.2. Limitations and Future Work

The presented framework relies on fixed thresholds for the residual evaluation module. While effective in the tested theoretical scenarios, modelling uncertainty, particularly parameter variations such as moments of inertia, can generate transient residuals during manoeuvres and will lead to false alarms in fault-free cases. Future work should therefore consider adaptive or statistically normalised decision logic, such as innovation-based normalisation in EKF/UKF , windowed detectors (e.g., moving RMS or CUSUM) [60] to improve reliability across a broader flight envelope and finally, experimental validation would be an important next step to confirm performance under real sensor characteristics, unmodelled dynamics, and environmental disturbances. In addition, this thesis focused primarily on fault detection rather than full fault isolation and severity estimation. Extending the framework toward fault identification could include structured residual design to enable control reallocation.

References

[1] M. Idrissi, M. Salami, and F. Annaz. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms. DOI: 10.1007/s10846-021-01527-7.

[2] G. K. Fourlas and G. C. Karras. A Survey on Fault Diagnosis and Fault-Tolerant Control Methods for Unmanned Aerial Vehicles. DOI: 10.3390/machines9090197.

[3] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl. Evaluating the risk of unmanned aircraft ground impacts. DOI: 10.1109/MED.2008.4602249.

[4] Md. N. Mowla, D. Asadi, and F. Sohel. Real-time fault detection in multirotor UAVs using lightweight deep learning and high-fidelity simulation data with single and double fault magnitudes. DOI: 10.1007/s40747-025-02195-y.

[5] M. Saied, H. Shraim, and C. Francis. A Review on Recent Development of Multirotor UAV Fault-Tolerant Control Systems. DOI: 10.1109/MAES.2023.3327697.

[6] A. Freddi, S. Longhi, and A. Monteriù. A Diagnostic Thau Observer for a Class of Unmanned Vehicles. DOI: 10.1007/s10846-012-9650-4.

[7] N. P. Nguyen, N. X. Mung, and S. K. Hong. Actuator Fault Detection and Fault-Tolerant Control for Hexacopter. DOI: 10.3390/s19214721.

[8] L. Qin, X. He, R. Yan, and D. Zhou. Active fault-tolerant control for a quadrotor with sensor faults. DOI: 10.1007/s10846-017-0474-0.

[9] A. A. Farahani, M. E. Dashiri, and F. Akhoundi. Fault Detection of a Quadrotor Actuator Based on the Luenberger Estimation Method. DOI: 10.22034/IJRRS.2025.8.2.4.

[10] C. Liu, B. Jiang, and K. Zhang. Incipient fault detection using an associated adaptive and sliding-mode observer for quadrotor helicopter attitude control systems. DOI: 10.1007/s00034-015-0229-8.

[11] M. Saied, H. Shraim, C. Francis, I. Fantoni, and B. Lussier. Actuator fault diagnosis in an octorotor UAV using sliding modes technique: Theory and experimentation. DOI: 10.1109/ECC.2015.7330772.

[12] X. Ren. Observer design for actuator failure of a quadrotor. DOI: 10.1109/ACCESS.2020.3017522.

[13] E. Abouselima, D. Ichalal, and S. Mammar. Quadrotor control and actuator fault detection: LQG versus robust H-/H observer. DOI: 10.1109/SYSTOL.2019.8864775.

[14] F. Xu, J. Tan, X. Wang, V. Puig, B. Liang, and B. Yuan. Mixed active/passive robust fault detection and isolation using set-theoretic unknown input observers. DOI: 10.1109/TASE.2017.2776998.

[15] L. Negash, S. Kim, and H. Choi. An eigenstructure assignment embedded unknown input observer approach for actuator fault detection in quadrotor dynamics. DOI: 10.1016/j.ifacol.2016.09.073.

[16] Y. Zhong, W. Zhang, Y. Zhang, J. Zuo, and H. Zhan. Sensor fault detection and diagnosis for an unmanned quadrotor helicopter. DOI: 10.1007/s10846-019-01002-4.

[17] M. Amoozgar, A. Chamseddine, and Y. Zhang. Experimental test of a two-stage Kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. DOI: 10.1007/s10846-012-9757-7.

[18] X. He, J. Chaudemar, J. Huang, and F. Defay. Fault tolerant control of a quadrotor based on parameter estimation techniques and use of a reconfigurable PID controller. DOI: 10.1109/MED.2016.7536052.

[19] A. Chamseddine, D. Theilliol, Y. Zhang, C. Join, and C. Rabbath. Active fault-tolerant control system design with trajectory re-planning against actuator faults and saturation: Application to a quadrotor unmanned aerial vehicle. DOI: 10.1002/acs.2451.

[20] J. Su, J. He, P. Cheng, and J. Chen. Actuator fault diagnosis of a hexacopter: A nonlinear analytical redundancy approach. DOI: 10.1109/MED.2017.7984152.

[21] M. Saied, T. Mahairy, C. Francis, H. Shraim, H. Mazeh, and M. ElRafei. Differential flatness-based approach for sensors and actuators fault diagnosis of a multirotor UAV. DOI: 10.1016/j.ifacol.2019.12.066.

[22] Y. Wang, B. Jiang, N. Lu, and J. Pan. Hybrid modeling based double-granularity fault detection and diagnosis for quadrotor helicopter. DOI: 10.1016/j.nahs.2015.12.005.

[23] J. Park, C. Jun, J. Jeong, and D. Chang. Real-time quadrotor actuator fault detection and isolation using multivariate statistical analysis techniques with sensor measurements. DOI: 10.23919/ICCAS50221.2020.9268391.

[24] M. Kuric, B. Lacevic, N. Osmic, and A. Tahirovic. RLS-based fault-tolerant tracking control of multirotor aerial vehicles. DOI: 10.1109/AIM.2017.8014173.

[25] M. Frangenberg, J. Stephan, and W. Fichter. Fast actuator fault detection and reconfiguration for multicopters. DOI: 10.2514/6.2015-1766.

[26] J. Rangel-Magdaleno, J. Urena-Urena, A. Hernandez, and C. Perez-Rubio. Detection of unbalanced blade on UAV by means of audio signal. DOI: 10.1109/ROPEC.2018.8661459.

[27] B. Ghalamchi and M. Mueller. Vibration-based propeller fault diagnosis for multicopters. DOI: 10.1109/ICUAS.2018.8453400.

[28] D. Ray, T. Roy, and S. Chattopadhyay. Skewness scanning for diagnosis of a small inter-turn fault in quadcopter's motor based on motor current signature analysis. DOI: 10.1109/JSEN.2020.3038786.

[29] S. Zeghlache, A. Djerioui, L. Benyetrou, T. Benslimane, H. Mekki, and A. Bouguerra. Fault tolerant control for modified quadrotor via adaptive type-2 fuzzy backstepping subject to actuator faults. DOI: 10.1016/j.isatra.2019.04.034.

[30] F. Pourpanah, B. Zhang, R. MA, and Q. Hao. Anomaly detection and condition monitoring of UAV motors and propellers. DOI: 10.1109/ICSENS.2018.8589572.

[31] B. Wang, Y. Shen, and Y. Zhang. Active fault-tolerant control for a quadrotor helicopter against actuator faults and model uncertainties. DOI: 10.1016/j.ast.2020.105745.

[32] C. Li, Y. Zhang, and P. Li. Extreme learning machine based actuator fault detection of a quadrotor helicopter. DOI: 10.1177/1687814017705068.

[33] J. Fu, C. Sun, Z. Yu, and L. Liu. A hybrid CNN-LSTM model based actuator fault diagnosis for six-rotor UAVs. DOI: 10.1109/CCDC.2019.8832706.

[34] M. Okasha, J. Kralev, and M. Islam. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. DOI: 10.3390/aerospace9060298.

[35] G. P. Falconí Salazar. Adaptive Fault Tolerant Control for VTOL Aircraft with Actuator Redundancy [PhD thesis]. DOI: 10.14459/2021md1595568.

[36] R. Mahony, V. Kumar, and P. Corke. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. DOI: 10.1109/MRA.2012.2206474.

[37] R. W. Prouty. *Helicopter Performance, Stability, and Control*.

[38] D. Mellinger and V. Kumar. Minimum Snap Trajectory Generation and Control for Quadrotors. DOI: 10.1109/ICRA.2011.5980409.

[39] S. Karaman and F. Riether. *Getting Started with the MIT Rolling Spider MATLAB Toolbox*. Tech. rep.

[40] F. Riether. Agile Quadrotor Maneuvering Using Tensor-Decomposition [Master's thesis MIT].

[41] A. Taame, I. Lachkar, A. Abouloifa, I. Mouchrif, and A. El Aroudi. Nonlinear Output Feedback Control for Parrot Mambo UAV: Robust Complex Structure Design and Experimental Validation. DOI: 10.3390/asi8040095.

[42] A. Lone, Q. Zhu, H. Nemati, and P. Mercorelli. Dynamic Inversion-Enhanced U-Control of Quadrotor Trajectory Tracking. DOI: 10.3390/drones8100599.

[43] M. W. Mueller and R. D'Andrea. Stability and Control of a Quadrocopter Despite the Complete Loss of One, Two, or Three Propellers. DOI: 10.1109/ICRA.2014.6906588.

[44] J. Mohd Sultana, N. H. Zani, M. Azuani, S. Z. Ibrahim, and A. M. Yusop. Analysis of Inertial Measurement Accuracy using Complementary Filter for MPU6050 Sensor. DOI: 10.17576/jkukm-2022-34(5)-24.

[45] A. Rudyk, A. O. Semenov, N. Kryvinska, and A. Safonyk. Strapdown Inertial Navigation Systems for Positioning Mobile Robots—MEMS Gyroscopes Random Errors Analysis Using Allan Variance Method. DOI: 10.3390/s20174841.

[46] D. G. Luenberger. Observing the State of a Linear System. DOI: 10.1109/TME.1964.4323124.

[47] A. Freddi, S. Longhi, and A. Monteriù. A hierchial Fault Diagnosis System for a Mini-Quadrotor. DOI: 10.3182/20090630-4-ES-2003.00012.

[48] Z. Cen, H. Noura, T. B. Susilo, and Y. A. Younes. Robust Fault Diagnosis for Quadrotor UAVs Using Adaptive Thau Observer. DOI: 10.1007/s10846-013-9921-8.

[49] A. Freddi, S. Longhi, and A. Monteriù. Actuator fault detection system for a mini-quadrotor. DOI: 10.1109/ISIE.2010.5637750.

[50] A. Freddi, S. Longhi, A. Monteriù, and M. Prist. Actuator Fault Detection and Isolation System for an Hexacopter. DOI: 10.1109/MESA.2014.6935563.

[51] A. Hasan and T. A. Johansen. hierchial Actuator Fault Diagnosis in Multirotor UAVs. DOI: 10.1109/ICUAS.2018.8453420.

[52] V. C. Nguyen and H. M. La. A Class of Hierarchical Sliding Mode Control based on Extended Kalman filter for Quadrotor UAVs. DOI: 10.48550/arXiv.2504.02851.

[53] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. DOI: 10.1002/9780470377819.

[54] H. A. G. Pointon, B. J. McLoughlin, C. Matthews, and F. A. Bezombes. Towards a Model Based Sensor Measurement Variance Input for Extended Kalman Filter State Estimation. DOI: 10.3390/drones3010019.

[55] S. Park, M.-S. Gil, H. Im, and Y.-S. Moon. Measurement Noise Recommendation for Efficient Kalman Filtering over a Large Amount of Sensor Data. DOI: 10.3390/s19051168.

[56] A. J. A. Tavares Jr. and N. M. F. Oliveira. A Novel Approach for Kalman Filter Tuning for Direct and Indirect Inertial Navigation System/Global Navigation Satellite System Integration. DOI: 10.3390/s24227331.

[57] E. A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. DOI: 10.1109/ASSPCC.2000.882463.

[58] K. Xiong, C. W. Chan, and H. Y. Zhang. UNSCENTED KALMAN FILTER FOR FAULT DETECTION. DOI: 10.3182/20050703-6-CZ-1902.01821.

[59] B. Strack van Schijndel, S. Sun, and C. de Visser. Fast Fault Detection on a Quadrotor using Onboard Sensors and a Kalman Filter Approach. DOI: 10.48550/arXiv.2102.06439.

[60] M. Kaminski. A Rapid Robust Fault Detection Algorithm for Flight Control Reconfiguration [PhD thesis Flight System Dynamics TUM].

[61] F. Holzapfel. *Flight System Dynamics 1 & 2 - Lecture Notes*.

A. Coordinate Systems

This appendix defines the coordinate systems used throughout this thesis. The following coordinate systems are employed to describe the motion of the quadrotor defined in the literature of the Institute of Flight Dynamics [61].

A.1. Earth Centered Inertial (ECI) Frame

The Earth Centered Inertial (ECI) frame is an inertial reference frame with its origin at the Earth's center. It does not rotate with the Earth and is used as the primary inertial frame for Newton's laws of motion.

- **Index:** I
- **Purpose:** Euclidean Frame (Inertial Axis System – Newton's Law may be used)
- **Origin:** Centre of Earth
- **Translation:** Around the Sun and with solar system
- **Rotation:** None
- x_I**-axis:** In equatorial plane, in the vernal equinox direction
- y_I**-axis:** In equatorial plane to form a right-hand system with the x_I-axis and the y_I-axis
- z_I**-axis:** Rotation axis of the Earth

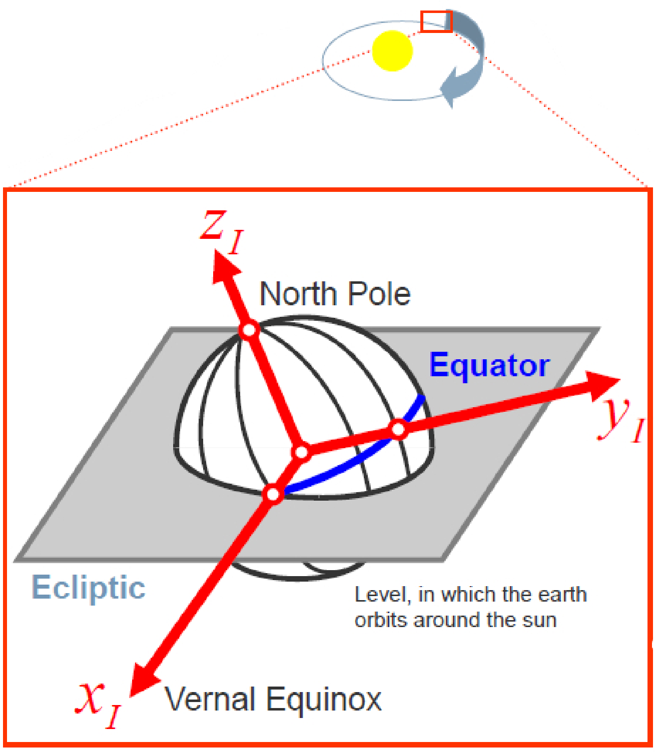


Figure A–1.: Earth Centered Inertial Frame (I)

A.2. Earth-Centered Earth-Fixed (ECEF) Frame

The Earth-Centered Earth-Fixed (ECEF) frame has its origin at the Earth's center but rotates with the Earth.

- **Index:** E
- **Purpose:** Navigation or Position Frame used to specify position
- **Origin:** Centre of Earth
- **Translation:** Moves with ECI-System
- **Rotation:** Rotates with Earth's rotation rate $\vec{\omega}^{IE}$
- x_E**-axis:** In equatorial plane, points through the Greenwich Meridian
- y_E**-axis:** In equatorial plane to form a right-hand system with the x_E-axis and z_E-axis
- z_E**-axis:** Rotation axis of the Earth

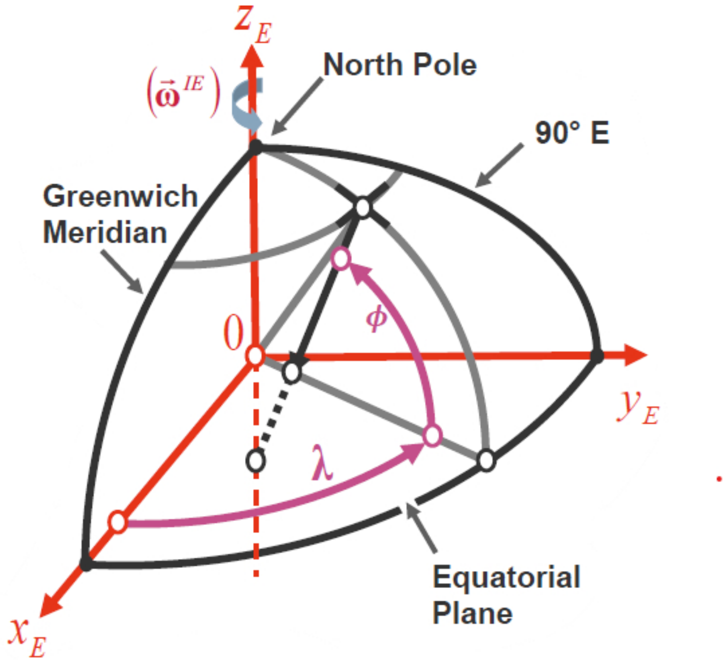


Figure A–2.: Earth-Centered Earth-Fixed Frame (E)

A.3. North-East-Down (NED) Frame

The North-East-Down (NED) frame is a local tangent plane used as a reference for aircraft attitude.

- **Index:** O
- **Purpose:** Attitude or Orientation Frame used to specify the attitud of the plane
- **Origin:** Aircraft reference point (R)
- **Translation:** Moves with aircraft reference point
- **Rotation:** Rotates with transport rate to keep NED alignment $\vec{\omega}^{EO}$
- x_O**-axis:** Parallel to the local geoid surface, pointing to the geographic north pole
- y_O**-axis:** Parallel to the local geoid surface, pointing east to form a right-hand system with x_O-axis and z_O-axis
- z_O**-axis:** Pointing downwards, perpendicular to the local geoid surface

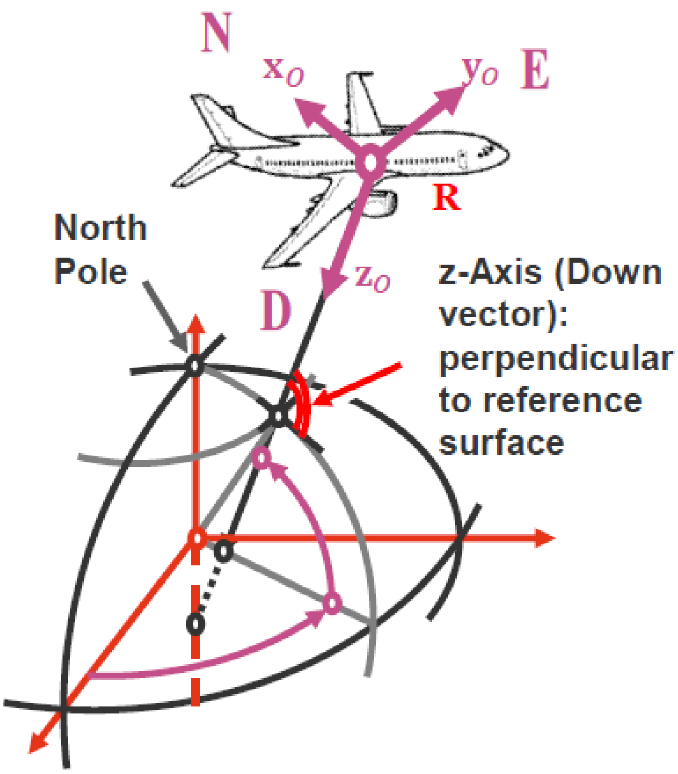


Figure A–3.: North-East-Down Frame (O)

A.4. Body-Fixed Frame (B)

The body-fixed frame is attached to the aircraft and moves with it.

- **Index:** B
- **Purpose:** Notation Frame – used for forces and moments
- **Origin:** Aircraft centre of gravity or reference point
- **Translation:** Moves with aircraft reference point R
- **Rotation:** Rotates with the rigid body aircraft
- x_B**-axis:** Points forward (longitudinal axis)
- y_B**-axis:** Points to the right (lateral axis)
- z_B**-axis:** Points downward (completes right-handed system)

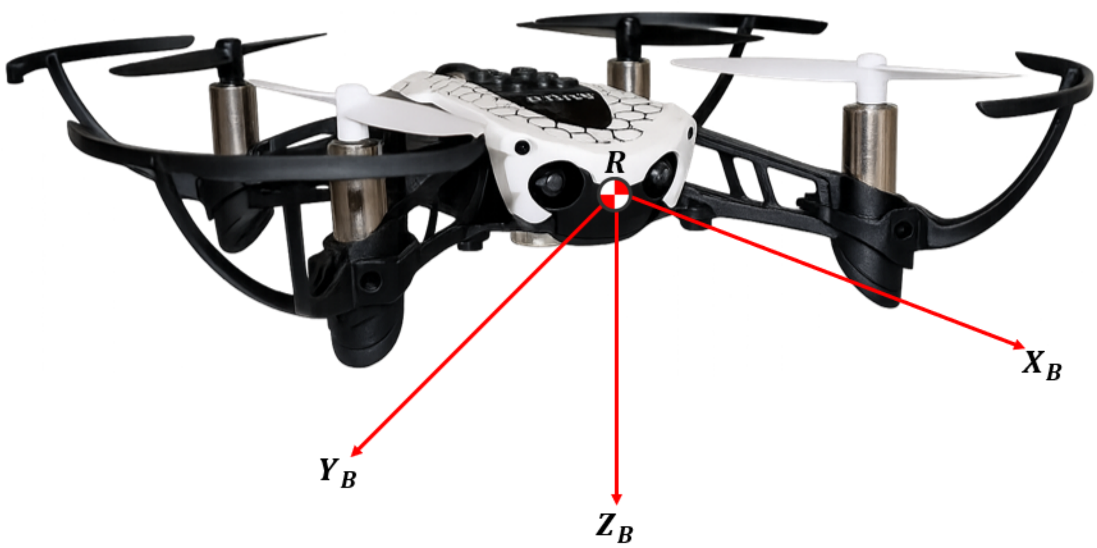


Figure A–4.: Body-Fixed Frame (B)

A.5. Local Navigation System (N)

The local navigation system is an Euclidean space used for aircraft navigation.

- **Index:** N
- **Purpose:** Navigation Frame, derived from NED-Frame to specify the position
- **Origin:** Fixed point on Earth's surface
- **Axes:** Coincide with NED axes when reference point R is at origin
- **Translation:** None
- **Rotation:** Only the angular rate of the Earth, because fixed to the Earth. $\vec{\omega}^{EN} = 0$ with respect to ECEF frame
- x_N**-axis:** Parallel to the local geoid surface, pointing to a direction that deviates from the alignment angle χ_N from the north direction
- y_N**-axis:** Parallel to the local geoid surface, to form a right-hand system with x_N-axis and z_N-axis
- z_N**-axis:** Pointing downwards, perpendicular to the local geoid surface

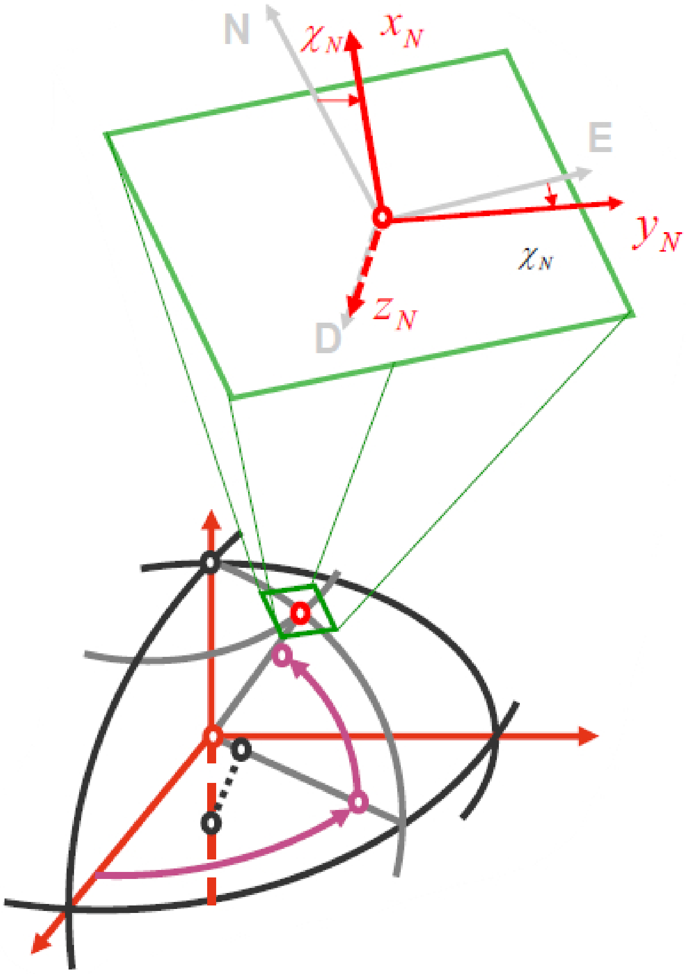


Figure A–5.: Local Navigation System (N)

A.6. Coordinate Transformations

The transformation between coordinate systems is achieved through rotation matrices. Key transformations include:

- $\mathbf{M}_{NB}$: Rotation from body frame to local navigation frame
- $\mathbf{M}_{BN}$: Rotation from local navigation frame frame to body frame ($\mathbf{M}_{BN} = \mathbf{M}_{NB}^T$)
- $\mathbf{M}_{EI}$: Rotation from ECI to ECEF frame
- $\mathbf{M}_{OE}$: Rotation from ECEF to NED frame

The time derivative of rotation matrices follows the strapdown equation:

$$\dot{\mathbf{M}}_{NB}(t) = \mathbf{M}_{NB}(t) \cdot \Omega((\vec{\omega}^{NB})_B(t)) \tag{A–1}$$

where $\Omega(\cdot)$ is the skew-symmetric matrix operator.