A Practical Exploration of a Dual Quaternion based Unscented Kalman Filter for Pose Estimation

Master of Science Thesis

For the Degree of Master of Science in Systems and Control at
Delft University of Technology

Thomas Nagy Zambo

2024-12-09

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft Center for
Systems and Control

Abstract

This thesis develops an approach for rigid body pose estimation using dual quaternion mathematics. Dual quaternions combine quaternion mathematics with concepts from dual numbers, offering a compact framework for describing the six-degrees-of-freedom of rigid body motion. Using quadrotor dynamics as a test case, this work formulates a dual quaternion model of dynamics and implements an Unscented Kalman Filter (UKF) for state estimation. Using concepts from Lie theory, the traditional UKF is extended to the unit dual quaternion manifold, while appropriately handling state covariance.

In addition to simulations of the dual quaternion Lie Algebraic UKF (LAUKF), results are presented of the LAUKF running on a purpose built flight computer. The flight computer software is developed using embedded Rust within the `RTIC2` framework, and discussion is included on the benefits and challenges encounter when developing the flight computer.

Contents

Appendix

List of Figures

List of Tables

1 Introduction

The field of control systems has seen significant advancements in recent years, with quadrotors emerging as one of the most versatile and widely studied platforms for both research and real-world applications [1], [2]. Quadrotors, due to their low cost construction, present a complex and rich testbed for control and estimation methodologies. Their maneuverability and ease of operation make them valuable in various fields, including aerial surveying, environmental monitoring, and autonomous delivery systems [3].

In parallel, quaternion-based representations have become the preferred method for describing orientation due to their minimal representation and singularity-free properties [4]. When describing rigid body motion in all 6 degrees of freedom, it is essentially required to develop two separate sets of equations: a set of equations on the unit quaternion manifold to describe rotation, and a set of equations in Euclidean space to describe translation. If it is desired to control the complete pose of a vehicle (combined position and orientation), the separate mathematical domains usually results in creation of separate, coupled, position and attitude controllers [5]. In tension with this fact, there are a number of applications where it is required to control the pose of a vehicle [6], [7], [8]. In this case it is desirable to formulate the control problem as a *pose* tracking problem [9].

This has led to the exploration of dual quaternion based control systems. Dual quaternions are a mathematical tool that extends quaternion algebra to represent both translation and rotation in a single, unified framework. While quaternions are well-established in the control and estimation literature, dual quaternions remain relatively under explored, particularly in under-actuated systems like quadrotors [10]. Additionally, despite the interest in dual quaternion-based representations for pose control, state estimation within this framework remains under examined. State estimation is a critical component of control systems as it provides real-time information about the vehicle's state, including its position, velocity, and orientation, based on sensor measurements [11]. In particular, the development of filters and observers that operate directly on the unit dual quaternion manifold is an open challenge. Such tools would need to address issues such as preserving the unit norm constraint inherent to dual quaternions and effectively fusing noisy measurements from diverse sensors. Most importantly, their successful application on real platforms is required for dual-quaternion based filters to be adopted.

This thesis aims to progress in this area by developing a dual quaternion model for quadrotor dynamics, implementing a compatible control scheme, and developing an Unscented Kalman Filter (UKF) for state estimation that leverages the dual quaternion framework. The quadrotor was chosen as the vehicle to study for its before mentioned low cost and wide applicability.

This thesis begins by establishing the mathematical foundations of quaternions and dual quaternions, covering their algebraic properties, associated Lie groups and algebras, and their applications in rigid body motion. The dual quaternion approach is particularly advantageous in the context of kinematic and dynamic modeling, as it enables a compact representation of motion in all six degrees of freedom. Dynamic modeling of the quadrotor introduces the relevant forces and applies dual quaternion-based dynamics required to capture the relationship between control inputs and the vehicle's resultant pose. In this model, dual quaternions enable direct numerical integration of both translational and rotational accelerations, providing a unified framework that simplifies the modeling of motion.

To provide pose estimation in the presence of noisy sensor data, this thesis develops a dual quaternion-based UKF for quadrotor state estimation. The UKF is particularly well-suited to nonlinear systems like quadrotors, as it captures the higher-order statistics of nonlinear transformations more accurately than linear approximations. However, representing covariance within the dual quaternion space introduces unique challenges. Traditional interpretations of the Kalman filter often do not accurately consider the properties of quaternion random variables [12]. By applying Lie algebra concepts, this thesis modifies the UKF to handle the covariance of the dual quaternion state appropriately.

A quaternion based controller is implemented in order to test the UKF on the quadrotor model. The controller is not developed using dual quaternions as the field of dual quaternion control on under actuated systems is still and active area of research [10]. Development of such a controller was deemed out of scope for this thesis. Instead, this thesis employs a Proportional-Derivative control scheme with feed-forward (PD+) for trajectory and attitude control. The chosen PD+ control strategy enables the realistic excitation of the quadrotor model for evaluation of the UKF.

Finally, the results of the UKF are presented, running on a purpose-built flight computer. The flight computer was developed as an exploratory exercise in creating a “greenfield” design, leveraging modern components and using embedded Rust as the programming language of choice.

As of 2024, Rust can no longer be considered a *new* programming language, having gained significant traction in systems-level programming due to its memory safety guarantees and robust modern tooling. However, in the embedded systems domain, Rust remains quite novel, having only achieved certifications for safety-critical use in 2023. Consequently, the tooling ecosystem for Rust in embedded systems is still evolving, with industry standards only beginning to take shape. This thesis then contributes to this feild by examining one potential avenue that an engineer might take when implementing state estimation algorithms on embedded hardware in Rust.

To conclude, this thesis explores the potential of dual quaternions in pose estimation, specifically in the context quadrotor applications. The implementation of the UKF on an embedded system takes step towards bringing the theory out of simulations and into the real world.

2 Background

This chapter introduces the fundamentals of quaternions and dual quaternions which are essential building blocks for subsequent chapters. These concepts will be applied in the following chapters to develop a dual quaternion model of quadrotor dynamics, along with the control strategy and an Unscented Kalman filter in following chapters. While much of the material presented here has been thoroughly established by other authors [13], [14], [15], rigid body motion in a dual quaternion frameworks remains somewhat of a niche topic. This thesis attempts to tie together gaps in understanding that one might encounter when approaching this material for the first time.

2.1 Reference Frames

As a preliminary, the reference frames used to define the vehicle's orientation, position, and motion are defined. These frames provide a basis for describing the movement of the quadrotor in space. The relevant references frames are: the local East-North-Up and North-East-Down frames, the body frame, and the desired frame.

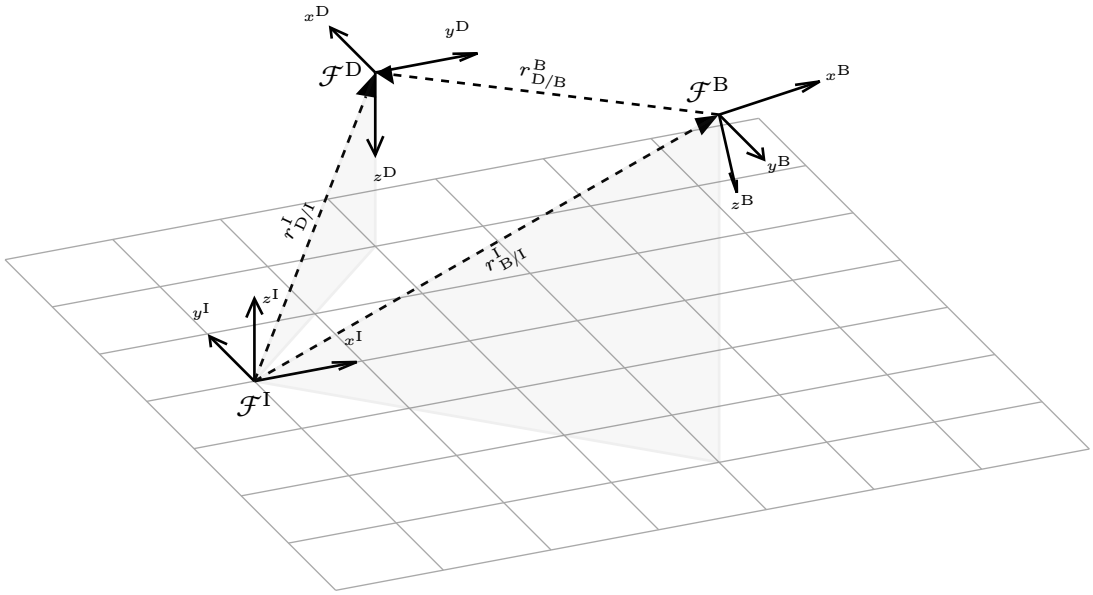

Figure 1: Reference Frames

East-North-Up Frame

The ENU frame, denoted as $\mathcal{F}^{\mathrm{I}}$, is a common coordinate reference frame used in aerial navigation, and serves as the inertial frame of reference. The origin of the ENU frame is generally set at the initial position of the quadrotor, often its take-off point or another meaningful location in inertial space such as a target destination. The ENU frame shall be used to develop the kinematics and dynamics of the quadrotor. In this coordinate system:

- The x-axis points eastward.
- The y-axis points northward.
- The z-axis points upward, completing a right-handed coordinate system.

North-East-Down Frame

The NED frame, denoted as $\mathcal{F}^{\mathrm{I}'}$ (not shown in Figure 1), is an alternative way to express the inertial frame. Is is useful in that it allows the development of attitude control systems following aerospace conventions regarding the expression of roll, pitch, and yaw. In this coordinate system:

- The x-axis points northward.
- The y-axis points eastward.
- The z-axis points downward, completing a right-handed coordinate system.

Body Frame

The body frame, denoted as $\mathcal{F}^{\mathrm{B}}$, is a coordinate system attached to the quadrotor itself, specifically at its center of mass. This frame provides a convenient reference for defining the quadrotor's orientation and movements relative to its own structure. As the quadrotor maneuvers, the body frame moves and rotates with it. In the body frame:

- The x-axis extends forward, through the nose of the vehicle.
- The y-axis extends starboard, through the side of the vehicle.
- The z-axis points downward, completing a right-handed coordinate system.

Desired Frame

The desired frame, denoted as $\mathcal{F}^{\mathrm{D}}$, represents the ideal or target orientation of the quadrotor as specified by the guidance system. The desired frame serves as a goal state for the control system.

Frame Notations

The notation used in this thesis for representing frame transformations will be introduced prior to the discussion on quaternions to clarify the intentions of each transformation:

- A rotation q describing the rotation that aligns frame $\mathcal{F}^{\mathrm{Y}}$ with $\mathcal{F}^{\mathrm{X}}$ is denoted as $\mathrm{q}_{\mathrm{X/Y}}$.
- The the vector r describing the location of frame $\mathcal{F}^{\mathrm{Y}}$'s origin, expressed in the frame $\mathcal{F}^{\mathrm{X}}$ is denoted as $r^{\mathrm{X}}_{\mathrm{X/Y}}$. The same vector expressed in $\mathcal{F}^{\mathrm{Y}}$ coordinates is denoted as $r^{\mathrm{Y}}_{\mathrm{X/Y}}$.
- Forces and torques acting on the quadrotor, expressed in the frame $\mathcal{F}^{\mathrm{X}}$ are denoted as f^{X} and τ^{X} respectively.

2.2 Quaternions

Quaternions present the most compact representation of rotations that avoids gimbal lock or other mathematical singularities [16]. A quaternion is a complex, four dimensional number composed of one real and three imaginary components. The set of all quaternions is denoted by $\mathbb{H}$.

$$\mathrm{q} = q_1 i + q_2 j + q_3 k + q_0 \omega \in \mathbb{H}, \quad q_0, q_1, q_2, q_3 \in \mathbb{R} \tag{1}$$

Where the basic quaternion parameters follow the relationship:

$$\begin{gathered} i^2 = j^2 = k^2 = ijk = -1 \\ ij = k, jk = i, ki = j \end{gathered} \tag{2}$$

A quaternion can be considered in vector form as:

$$\mathrm{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_0 \end{bmatrix}^\top \in \mathbb{H}, \quad q_0, q_1, q_2, q_3 \in \mathbb{R} \tag{3}$$

Or parameterized by an imaginary vector and real scalar component:

$$\mathrm{q} = (\vec{q}, q_0) \in \mathbb{H}, \quad \vec{q} \in \mathbb{R}^3, \ q_0 \in \mathbb{R} \tag{4}$$

Operation	Equation
Addition	$\mathrm{a} + \mathrm{b} = \left(\vec{a} + \vec{b}, a_0 + b_0\right) \in \mathbb{H}$
Scalar Multiplication	$\lambda \mathrm{a} = (\lambda \vec{a}, \lambda a_0) \in \mathbb{H}$
Product	$\mathrm{a} \otimes \mathrm{b} = \left(a_0 \vec{b} + b_0 \vec{a} + \vec{a} \times \vec{b}, a_0 b_0 - \vec{a} \cdot \vec{b}\right) \in \mathbb{H}$
Conjugation	$\mathrm{a}^* = (-\vec{a}, a_0) \in \mathbb{H}$
Dot Product	$\mathrm{a} \cdot \mathrm{b} = \frac{1}{2}(\mathrm{a} \otimes \mathrm{b}^* + \mathrm{b} \otimes \mathrm{a}^*) = \left(0_{3\times 1}, a_0 b_0 + \vec{a} \cdot \vec{b}\right) \in \mathbb{R}$
Cross Product	$\mathrm{a} \times \mathrm{b} = \frac{1}{2}(\mathrm{a} \otimes \mathrm{b} - \mathrm{b}^* \otimes \mathrm{a}^*) = \left(a_0 \vec{b} + b_0 \vec{a} + \vec{a} \times \vec{b}, 0\right) \in \mathbb{H}^{\mathrm{p}}$
Norm	$\lVert \mathrm{a} \rVert = \sqrt{\mathrm{a} \otimes \mathrm{a}^*} = \sqrt{a_1^2 + a_2^2 + a_3^2 + a_0^2} \in \mathbb{R}$
Inverse	$\mathrm{a}^{-1} = \frac{\mathrm{a}^*}{\lVert \mathrm{a} \rVert} \in \mathbb{H}$

Table 1: Basic Quaternion Algebra

Pure Quaternions

A pure quaternion is one whose parameterized by a zero real component and a non-zero imaginary component. Any vector $r \in \mathbb{R}^3$ can be represented by a pure quaternion whose imaginary component is equal to the vector. With this parameterization, pure quaternions serve as a useful construction for embedding $\mathbb{R}^3$ within $\mathbb{H}$. The set of all pure quaternions is denoted by $\mathbb{H}^{\mathrm{p}}$.

$$\mathrm{r} = (r, 0) \in \mathbb{H}^{\mathrm{p}}, \quad r \in \mathbb{R}^3 \tag{5}$$

This act of embedding three dimensional vectors in quaternion space is common in quaternion algebra, so for the sake of simplicity, this thesis omits additional notation when performing straightforward vector embedding. In such cases, it will be understood from the context that the mathematical objects involved represent pure quaternions parameterized by vectors.

Unit Quaternions

An important subset of quaternions is the unit quaternion. A unit quaternion is defined as a quaternion whose norm equals one, or equivalently, $\mathrm{q}\mathrm{q}^* = 1$. The set of all unit quaternions is denoted as $\mathbb{H}^{\mathrm{u}}$. Unit quaternions are particularly useful for describing rotations in three dimensional space. For example, the rotation θ about a unit vector n describing the transformation between frames $\mathcal{F}^{\mathrm{I}}$ and $\mathcal{F}^{\mathrm{B}}$ can be expressed by a unit quaternion as:

$$\mathrm{q}_{\mathrm{B/I}} = \left(n \sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right)\right) \in \mathbb{H}^{\mathrm{u}} \tag{6}$$

In this thesis, all representations of rotations or attitudes will be expressed as unit quaternions. The rotation of vectors in three dimensional space becomes straightforward with unit quaternions. The vector r expressed in the reference frame $\mathcal{F}^{\mathrm{I}}$ can be rotated into the reference frame $\mathcal{F}^{\mathrm{B}}$ by:

$$r^{\mathrm{B}} = \mathrm{q}_{\mathrm{B/I}}^* \otimes r^{\mathrm{I}} \otimes \mathrm{q}_{\mathrm{B/I}} \in \mathbb{H}^{\mathrm{p}} \tag{7}$$

Quaternion Lie Theory

Lie theory provides a framework for studying continuous transformations using mathematical objects called Lie groups and Lie algebras [17]. This theory is especially useful when dealing with objects that rotate, move, or transform smoothly, as in rigid body motion. A Lie group is a set of transformations that forms a continuous group. Rotations are a common example: an object can be continuously rotated through a sequence of small angles. A Lie algebra is the associated *linearized* version of the Lie group, capturing the infinitesimal transformations. For rotation groups, the Lie

algebra captures the angular velocities. Two useful quaternion transformations come from Lie theory. The exponential map relates elements from a Lie algebra to its corresponding Lie group:

$$\exp(\mathrm{q}) = \cos(\|\mathrm{q}\|) + \frac{\mathrm{q}}{\|\mathrm{q}\|}\sin(\|\mathrm{q}\|) \in \mathbb{H}^{\mathrm{u}}, \quad \mathrm{q} \in \mathbb{H}^{\mathrm{p}} \tag{8}$$

While the natural logarithm mapping projects a quaternion into its associated Lie algebra:

$$\ln(\mathrm{q}) = \begin{cases} \frac{\vec{q}}{\|\vec{q}\|}\arccos(q_0) & \|\vec{q}\| \neq 0 \\ 0 & \|\vec{q}\| = 0 \end{cases} \in \mathbb{R}^3, \quad \mathrm{q} \in \mathbb{H}^{\mathrm{u}} \tag{9}$$

A geometrical interpretation of quaternion Lie theory imagines the Lie group of unit quaternions $\mathbb{H}^{\mathrm{u}}$ as a four dimensional hyper-sphere with unit radius. The associated quaternion Lie algebra of a particular quaternion is a Euclidean space $\mathbb{R}^3$ tangent to the manifold $\mathbb{H}^{\mathrm{u}}$ at that quaternion.

Quaternion Rotational Kinematics

The quaternion differential equation describing the rotation of a rigid body is given by (10). In this equation, $\mathrm{q}_{\mathrm{B/I}}$ is a unit quaternion describing the orientation of the body frame $\mathcal{F}^{\mathrm{B}}$ relative to the inertial frame $\mathcal{F}^{\mathrm{I}}$. The angular velocity of the rigid body $\omega^{\mathrm{B}}_{\mathrm{B/I}}$ is a pure quaternion parameterized by the vector of angular rates in the body frame.

$$\dot{\mathrm{q}}_{\mathrm{B/I}} = \frac{1}{2}\mathrm{q}_{\mathrm{B/I}} \otimes \omega^{\mathrm{B}}_{\mathrm{B/I}} \in \mathbb{H} \tag{10}$$

Examining (10) it is clear that since the angular velocity is unbounded, the resulting quaternion product would violate the unit norm constraint. To integrate the quaternion differential equation without violating the unit norm, the exponential map is applied to the angular velocity in (10), mapping from angular velocities to quaternion rotations while preserving the unit norm constraint.

$$\mathrm{q}_{\mathrm{B/I}_{k+1}} = \mathrm{q}_{\mathrm{B/I}_k} \otimes \exp\left(\frac{\Delta t}{2}\omega^{\mathrm{B}}_{\mathrm{B/I}_k}\right) \in \mathbb{H}^{\mathrm{u}} \tag{11}$$

2.3 Dual Quaternions

Dual quaternions combine the concept of dual numbers with quaternion algebra. Dual numbers introduce the dual unit ϵ with the properties $\epsilon^2 = 0$ and $\epsilon \neq 0$. Where standard dual numbers have a real and dual part, dual quaternions have a real and dual quaternion:

$$\mathbf{q} = \mathrm{q}_{\mathrm{r}} + \epsilon\mathrm{q}_{\mathrm{d}} \in \mathbb{H}_{\mathrm{d}}, \quad \mathrm{q}_{\mathrm{r}}, \mathrm{q}_{\mathrm{d}} \in \mathbb{H} \tag{12}$$

Operation	**Equation**
Scalar Multiplication	$\lambda\mathbf{a} = \lambda\mathrm{a}_{\mathrm{r}} + \epsilon\lambda\mathrm{a}_{\mathrm{d}} \in \mathbb{H}_{\mathrm{d}}$
Product	$\mathbf{a} \otimes \mathbf{b} = \mathrm{a}_{\mathrm{r}} \otimes \mathrm{b}_{\mathrm{r}} + \epsilon(\mathrm{a}_{\mathrm{r}} \otimes \mathrm{b}_{\mathrm{d}} + \mathrm{a}_{\mathrm{d}} \otimes \mathrm{b}_{\mathrm{r}}) \in \mathbb{H}_{\mathrm{d}}$
Conjugation	$\mathbf{a}^* = \mathrm{a}^*_{\mathrm{r}} + \epsilon\mathrm{a}^*_{\mathrm{d}} \in \mathbb{H}_{\mathrm{d}}$
Cross Product	$\mathbf{a} \times \mathbf{b} = \mathrm{a}_{\mathrm{r}} \times \mathrm{b}_{\mathrm{r}} + \epsilon(\mathrm{a}_{\mathrm{d}} \times \mathrm{b}_{\mathrm{r}} + \mathrm{a}_{\mathrm{r}} \times \mathrm{b}_{\mathrm{d}}) \in \mathbb{H}_{\mathrm{d}}$
Norm	$\|\mathbf{a}\| = \mathrm{a}_{\mathrm{r}} \cdot \mathrm{a}_{\mathrm{r}} + \epsilon(2\mathrm{a}_{\mathrm{r}} \cdot \mathrm{a}_{\mathrm{d}}) \in \mathbb{R}_{\mathrm{d}}$

Table 2: Basic Dual Quaternion Algebra

Unit Dual Quaternions

As with standard quaternions, a unit dual quaternion can also be defined as a dual quaternion whose norm is equal to one, or equivalently, $\mathbf{q}\mathbf{q}^* = 1$. Unit quaternions are useful tool to represent the *pose* of an object. Consider the body frame $\mathcal{F}^{\mathrm{B}}$ with orientation $\mathrm{q}_{\mathrm{B/I}}$ and position $r^{\mathrm{B}}_{\mathrm{B/I}}$ with respect to the inertial frame $\mathcal{F}^{\mathrm{I}}$. The complete pose of the body frame can be represented by the unit dual quaternion constructed in (13).

$$\mathbf{q_{B/I}} = \mathrm{q}_{\mathrm{B/I}} + \epsilon\left(\frac{1}{2}\mathrm{q}_{\mathrm{B/I}} r^{\mathrm{B}}_{\mathrm{B/I}}\right) \in \mathbb{H}^{\mathrm{u}}_{\mathrm{d}} \tag{13}$$

Dual Quaternion Lie Theory

The set of unit dual quaternions $\mathbb{H}^{\mathrm{u}}_{\mathrm{d}}$ forms a Lie group corresponding to rigid body motions. This Lie group can be mapped to its associated Lie algebra, which describes the infinitesimal transformations (rotational and translational velocities) of the rigid body. The exponential map of dual quaternions is:

$$\exp(\mathbf{q}) = \cos(\|\mathbf{q}\|) + \frac{\mathbf{q}}{\|\mathbf{q}\|}\sin(\|\mathbf{q}\|) \in \mathbb{H}^{\mathrm{u}}_{\mathrm{d}}, \quad \mathbf{q} \in \mathbb{H}^{\mathrm{p}}_{\mathrm{d}} \tag{14}$$

While (14) is mathematically valid, it is cumbersome to work with in practice. The exponential map can be more conveniently expressed as (15) [18]:

$$\exp(\mathbf{q}) = \exp(\mathrm{q}_{\mathrm{r}})(1 + \epsilon \mathrm{q}_{\mathrm{d}}) \in \mathbb{H}^{\mathrm{u}}_{\mathrm{d}}, \quad \mathbf{q} \in \mathbb{H}^{\mathrm{p}}_{\mathrm{d}} \tag{15}$$

The natural logarithm that maps from the dual quaternion Lie group to its Lie algebra is:

$$\ln(\mathbf{q}) = \begin{cases} \|\ln(\mathrm{q}_{\mathrm{r}})\| + \epsilon\frac{\mathrm{q}_{\mathrm{d}}}{\|\mathrm{q}_{\mathrm{r}}\|} & \|\mathrm{q}_{\mathrm{r}}\| \neq 0 \\ 0 & \|\mathrm{q}_{\mathrm{r}}\| = 0 \end{cases} \in \mathbb{R}^6, \quad \mathbf{q} \in \mathbb{H}^{\mathrm{u}}_{\mathrm{d}} \tag{16}$$

While much further from physical intuition, the Lie group of unit dual quaternions $\mathbb{H}^{\mathrm{u}}_{\mathrm{p}}$ can be visualized as a eight-sphere with unit radius representing combined rotational and translational transformations. The associated Lie algebra forms a six-dimensional space $\mathbb{R}^6$ tangent to a dual quaternion on the manifold, capturing the instantaneous linear and angular velocities.

Dual Quaternion Kinematics

The dual quaternion differential equation that governs the change in the pose of a rigid body is given by (17). Leveraging concepts from screw theory [18], this equation introduces the dual twist term $\boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}}$. In screw theory, any motion can be described as a combination of rotation about, and translation along, a screw axis. The dual twist provides a compact expression of the evolution of a body's pose (17).

$$\dot{\mathbf{q}}_{\mathbf{B/I}} = \frac{1}{2}\mathbf{q}_{\mathbf{B/I}} \otimes \boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}} \in \mathbb{H}_{\mathrm{d}} \tag{17}$$

The twist $\boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}}$ represents the combination of angular and linear velocity. Due to the rotation of the body frame, the kinematic transport equation must be applied when expressing the translational component of the dual twist [19]. This results in the additional $\omega \times r$ term appearing in the dual component.

$$\boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}} = \omega^{\mathrm{B}}_{\mathrm{B/I}} + \epsilon\left(\dot{r}^{\mathrm{B}}_{\mathrm{B/I}} + \omega^{\mathrm{B}}_{\mathrm{B/I}} \times r^{\mathrm{B}}_{\mathrm{B/I}}\right) \in \mathbb{H}^{\mathrm{p}}_{\mathrm{d}} \tag{18}$$

As with the standard quaternion kinematic equation, naive integration of (17) will result in the violation of the unit norm constraint. To address this, the exponential map of the dual twist can be used to preserve the unit norm:

$$\mathbf{q}_{\mathbf{B/I}_{k+1}} = \mathbf{q}_{\mathbf{B/I}} \otimes \exp\left(\frac{\Delta t}{2}\boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}}\right) \in \mathbb{H}^{\mathrm{u}}_{\mathrm{d}} \tag{19}$$

Dual Quaternion Dynamics

Dual quaternion dynamics introduces the dual wrench term $\boldsymbol{f}^{\mathbf{B}}$, which represents the combined forces and moments applied to the body.

$$\boldsymbol{f}^{\mathbf{B}} = \tau^{\mathrm{B}} + \epsilon f^{\mathrm{B}} \in \mathbb{H}^{\mathrm{p}}_{\mathrm{d}} \tag{20}$$

Similar to how the dual twist encapsulates motion in kinematics, the dual wrench provides a compact representation of forces in screw theory, where forces and torques can be resolved along a screw axis. This allows dual quaternions to represent dynamic equations that combine rotational and translational effects in a unified framework, enabling the direct integration of forces and moments to compute the evolution of the dual twist and subsequently, the body's pose. Taking the derivative of (18) with respect to the inertial frame yields:

$$\dot{\boldsymbol{\xi}}^{\mathbf{B}}_{\mathbf{B/I}} = \dot{\omega}^{\mathrm{B}}_{\mathrm{B/I}} + \epsilon\left(\dot{\omega}^{\mathrm{B}}_{\mathrm{B/I}} \times r^{\mathrm{B}}_{\mathrm{B/I}} + \omega^{\mathrm{B}}_{\mathrm{B/I}} \times \dot{r}^{\mathrm{B}}_{\mathrm{B/I}} + \ddot{r}^{\mathrm{B}}_{\mathrm{B/I}}\right) \in \mathbb{H}^{\mathrm{p}}_{\mathrm{d}} \tag{21}$$

Substituting with the familiar equations of linear and rotational dynamics $\ddot{r} = m^{-1} f$ and $\dot{\omega} = J^{-1}(\tau - \omega \times J\omega)$ results in the dual quaternion dynamic differential equation (22) where the separation of internal forces arising from the translation and rotation of the rigid body $\boldsymbol{F}^{\mathbf{B}}_{\mathbf{B/I}}$, and external forces acting on the rigid body $\boldsymbol{u}^{\mathbf{B}}_{\mathbf{B/I}}(\boldsymbol{f}^{\mathbf{B}})$ is made explicit:

$$\dot{\boldsymbol{\xi}}^{\mathbf{B}}_{\mathbf{B/I}} = \boldsymbol{F}^{\mathbf{B}}_{\mathbf{B/I}} + \boldsymbol{u}^{\mathbf{B}}_{\mathbf{B/I}}(\boldsymbol{f}^{\mathbf{B}}), \quad \begin{cases} \boldsymbol{F}^{\mathbf{B}}_{\mathbf{B/I}} = a^{\mathrm{B}}_{\mathrm{B/I}} + \epsilon\left(a^{\mathrm{B}}_{\mathrm{B/I}} \times r^{\mathrm{B}}_{\mathrm{B/I}} + \omega^{\mathrm{B}}_{\mathrm{B/I}} \times \dot{r}^{\mathrm{B}}_{\mathrm{B/I}}\right) \\ \boldsymbol{u}^{\mathbf{B}}_{\mathbf{B/I}}(\boldsymbol{f}^{\mathbf{B}}) = J^{-1}\tau^{\mathrm{B}} + \epsilon\left(J^{-1}\tau^{\mathrm{B}} \times r^{\mathrm{B}}_{\mathrm{B/I}} + m^{-1} f^{\mathrm{B}}\right) \\ a^{\mathrm{B}}_{\mathrm{B/I}} = -J^{-1}\left(\omega^{\mathrm{B}}_{\mathrm{B/I}} \times J\omega^{\mathrm{B}}_{\mathrm{B/I}}\right) \end{cases} \tag{22}$$

Unlike with the kinematics, the evolution of the twist does require that the unit norm constraint is maintained, thus the integration of (21) is expressed as:

$$\boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}_{\boldsymbol{k+1}}} = \boldsymbol{\xi}^{\mathbf{B}}_{\mathbf{B/I}_{\boldsymbol{k}}} + \Delta t \dot{\boldsymbol{\xi}}^{\mathbf{B}}_{\mathbf{B/I}_{\boldsymbol{k}}} \tag{23}$$

Fictitious Forces

While the dynamics above are adequate for implementing stabilizing control [14], they do not account for the well-known fictitious forces that act within a rotating reference frame [19], [20]. Including these fictitious forces is essential for accurately propagating vehicle dynamics when developing a simulation. The forces in question are:

$$\begin{aligned} f^{\mathrm{B}}_{\mathrm{Coriolis}} &= -2m\omega^{\mathrm{B}}_{\mathrm{B/I}} \times \dot{r}^{\mathrm{B}}_{\mathrm{B/I}} \\ f^{\mathrm{B}}_{\mathrm{Centrifugal}} &= -m\omega^{\mathrm{B}}_{\mathrm{B/I}} \times \left(\omega^{\mathrm{B}}_{\mathrm{B/I}} \times r^{\mathrm{B}}_{\mathrm{B/I}}\right) \\ f^{\mathrm{B}}_{\mathrm{Euler}} &= -m\dot{\omega}^{\mathrm{B}}_{B/I} \times r^{\mathrm{B}}_{\mathrm{B/I}} \end{aligned} \tag{24}$$

2.4 Quadrotor Force Model

With the framework of dual quaternion dynamics in place, the specific forces that govern a quadrotor's behavior can be examined. Quadrotors are under-actuated vehicles, operating with six degrees of freedom while only being capable of generating forces in four of those degrees. Each of the quadrotor's four rotors can provide thrust in the vertical direction, and through varying their speeds, they can indirectly produce control moments for rotational movements (θ - pitch, φ - roll, and ψ - yaw).

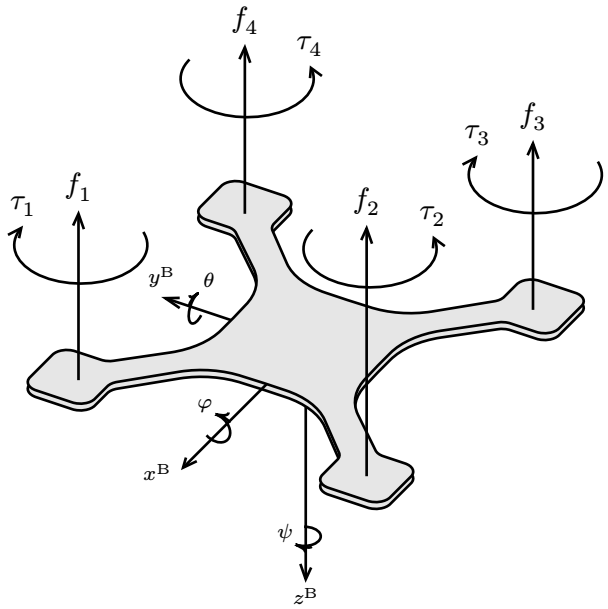

Figure 2: Quadrotor Model

Control Forces

To represent the control forces acting of the quadrotor, the control allocation matrix A can be introduced [20], which maps control inputs to the generated forces and moments. In the case of a quadrotor in a X-configuration:

$$\begin{bmatrix} f_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ d_1' c_T & -d_2' c_T & -d_3' c_T & d_4' c_T \\ d_1' c_T & d_2' c_T & -d_3' c_T & -d_4' c_T \\ c_M & -c_M & c_M & -c_M \end{bmatrix}}_{A} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \tag{25}$$

- w_n - Rotor speed.
- c_T - Rotor coefficient of thrust.
- c_M - Rotor coefficient of moment.
- d_n' - $\frac{\sqrt{2}}{2} d_n$, where d_n is the length of the rotor arm.

Rotor Dynamics

The rotors of a quadrotor possess their own intrinsic dynamics that impact the overall response of the system. Unlike the idealized scenario, where desired rotor speeds are achieved instantly, real-world rotor dynamics involve a delay due to physical and electronic limitations [20]. This delay arises from the time required for the motor to accelerate or decelerate the rotor blades, which in turn affects the quadrotor's thrust and torque generation.

In this thesis it will be assumed that the quadrotor will use digital motor control. This eliminates the need to explicitly model the transfer function of electronic speed controller (ESC) which typically relates input voltage to rotor speed. This simplification is possible because the ESC characteristics are managed by an external digital controller, removing the need for a separate voltage-to-speed model. However, the underlying motor dynamics still influence the response time and need to be accounted for. Each rotor's brushless DC motor can be approximated by a first-order low-pass filter:

$$\dot{w}_n = -\frac{1}{T_m} w_n + \frac{1}{T_m} w_{n,u} \tag{26}$$

- T_m - Motor times constant.
- $w_{n,u}$ - Rotor speed command.

This first-order response, characterized by a time constant, effectively applies a form of damping to the rotor's response. As a result, any requested changes in thrust or torque experience a smoothing effect, gradually approaching the desired values rather than achieving them instantaneously.

External Forces

As with any aerospace vehicle, a quadrotor experiences various aerodynamic forces. However, under the operating conditions assumed in this thesis, aerodynamic forces such as drag are negligible and can be treated as unmodeled disturbances in the system. The only external force explicitly considered is gravity, which acts in the negative z^{I} direction of the inertial frame. To express this gravitational force in the quadrotor's body frame, the following transformation is applied:

$$f_g^{\mathrm{B}} = \mathrm{q}_{\mathrm{B/I}} \otimes f_g^{\mathrm{I}} \otimes \mathrm{q}_{\mathrm{B/I}}^{*} \tag{27}$$

3 Pose Determination

Accurately determining the pose of a quadrotor is a prerequisite for control and navigation. Pose estimation in quadrotors is particularly challenging due to the widespread use of low-cost Inertial Measurement Units (IMU) using Micro Electro-Mechanical Systems (MEMS), which are often suffer from significant measurement noise. This section introduces a method for pose determination using an Unscented Kalman Filter (UKF) designed with dual quaternions that is cognizant of these effects.

To address the challenges associated with representing the covariance of states in the dual quaternion space $\mathbb{H}_{\mathrm{d}}$, the Kalman filter will be modified using concepts from Lie theory, which allows for a more robust treatment of the state's uncertainty.

3.1 Sensor Measurement Models

MEMS IMU sensors have a couple of well known practical limitations. Gyroscopic measurements are known to be low noise, however suffer from slowly changing, near constant, bias instability. This fact prevents attitude estimation through naive integration of gyroscopic rates to be useful for any meaningful duration and also prevents the determination of angular acceleration by simple differentiation due to noise. The first step in utilizing these sensor measurements is to understand their measurement models.

Gyroscope Measurement Model
The angular rate of the quadrotor $\omega_{\mathrm{B/I}}^{\mathrm{B}}$ can be measured directly using the onboard gyroscope. It is well known that the gyroscopic rate measurements produced by these sensors suffer from noise and bias corruption [21], [22], [23]. Dropping the frame notation for a moment, the measurement model can be described by:

$$\tilde{\omega}(t) = \omega(t) + n_{\omega}(t) + \beta_{\omega}(t) \tag{28}$$

Where $\tilde{\omega}$ is the measured value, ω is the true value, n_{ω} is additive noise, and β_{ω} is a measurement bias. Modeling the noise vector $n_{\omega}(t)$ as Gaussian white noise accurately captures the behavior of the the MEMS IMU [22].

$$\begin{aligned} E[n_{\omega}(t)] &\equiv 0_{3\times 1} \\ E\big[n_{\omega}(t_1) n_{\omega}^{\top}(t_2)\big] &= N_{\omega}\delta(t_1 - t_2) \end{aligned} \tag{29}$$

The bias present in the gyroscope readings can be modeled as a random walk:

$$\begin{aligned} \dot{\beta}_{\omega}(t) &= n_{\beta_{\omega}}(t) \\ E\Big[n_{\beta_{\omega}}(t)\Big] &\equiv 0_{3\times 1} \\ E\Big[n_{\beta_{\omega}}(t_1) n_{\beta_{\omega}}^{\top}(t_2)\Big] &= N_{\beta_{\omega}}\delta(t_1 - t_2) \end{aligned} \tag{30}$$

Where the matrices N_{ω} and $N_{\beta_{\omega}}$ are diagonal matrices that contain the variance of the noise and bias for the individual axes:

$$\begin{aligned} N_{\omega} &= \mathrm{diag}\big(\sigma_{\omega,x}^2,\ \sigma_{\omega,y}^2,\ \sigma_{\omega,z}^2\big) \\ N_{\beta_{\omega}} &= \mathrm{diag}\Big(\sigma_{\beta_{\omega},x}^2,\ \sigma_{\beta_{\omega},y}^2,\ \sigma_{\beta_{\omega},z}^2\Big) \end{aligned} \tag{31}$$

Discretizing the previous equations yields the following discrete time model:

$$\tilde{\omega}_k = \omega_k + n_{\omega,k} + \beta_{\omega,k} \tag{32}$$

$$n_{\omega,k} = N_\omega n_k, \quad n_k \sim \mathcal{N}(0,1)_{3\times 1} \tag{33}$$

$$\beta_{\omega,k} = \beta_{\omega,k-1} + n_{\beta_\omega,k} \quad \begin{cases} n_{\beta_\omega,k} = N_{\beta_\omega} n_k \\ n_k \sim \mathcal{N}(0,1)_{3\times 1} \end{cases} \tag{34}$$

With the appropriate modifications made to the process variances for discrete time:

$$\begin{aligned} N_{\omega_k} &= \frac{N_\omega}{\sqrt{\Delta t}} \\ N_{\beta_\omega,k} &= N_{\beta_\omega}\sqrt{\Delta t} \end{aligned} \tag{35}$$

Accelerometer Measurement Model

The acceleration of the quadrotor $a^{\mathrm{B}}_{\mathrm{B/I}}$ can be measured by the onboard MEMS accelerometer. It shares the same measurement model as the gyroscope [22]. However, the bias instability of MEMS accelerometers is much smaller than that of MEMS gyroscopes. Thus over the short duration considered in this thesis, the discrete time measurement model is:

$$\tilde{a}_k = a_k + n_{a,k} \tag{36}$$

Magnetometer Measurement Model

The MEMS magnetometer presents additional complications compared to the gyroscope and accelerometer. First, it measures the Earth's magnetic field vector b at the quadrotor's current location. This is an inertial quantity that must be rotated into the body frame. Furthermore, the magnetometer is susceptible to hard and soft iron effects that influence sensor measurements [24], [25], [26]. In practice, the bias from hard iron effects is more appropriate to handle via calibration [27]. However, distortion due to soft iron effects varies based on current draw, rotor speed, and other electromagnetic influences on the vehicle. The soft iron effects introduce a time-varying distortion matrix that must be identified on-line. For this thesis, both hard and soft-iron effects are neglected due to the high complexity and the negligible benefit to the main goal of the research. The discrete-time measurement model is:

$$\tilde{b}_k = \left(\mathrm{q}_{\mathrm{B/I}} \otimes b_k\right) + n_{b,k} \tag{37}$$

GPS Measurement Model

GPS measurements are influenced by a range of factors, including inaccuracies in satellite orbit models, signal reflections from buildings and terrain (multipath effects), atmospheric disturbances, and clock errors. These factors introduce complex and non-Gaussian noise into the measurements [28]. However, for the purposes of this thesis, a simple Gaussian noise model was selected to represent the measurement errors. Although this approach does not capture all potential error sources, it provides a practical and computationally efficient approximation that aligns with the thesis goals and the expected accuracy requirements of the quadrotor system. The Gaussian noise model serves as a first-order approximation, balancing simplicity with the level of fidelity needed to meet the design objectives. In this case the measurement model of the vehicles position $r^{\mathrm{I}}_{\mathrm{B/I}}$ becomes:

$$\tilde{r}_k = r_k + n_{r,k} \tag{38}$$

3.2 Kalman Filtering

The Kalman filter is a cornerstone of state estimation for dynamic systems. It offers an elegant and computationally efficient approach to combine noisy measurements with predictive models. It operates optimally under the assumptions of linear dynamics and Gaussian noise, leveraging these properties to estimate system states. However many real-world systems, such as quadrotors, are non-linear in nature. In this regime, the Kalman filter's effectiveness diminishes. To address this, extensions such as the Extended or Unscented Kalman Filter have been developed.

This thesis examines the UKF as the UKF is found to be particularly compatible with the modifications from Lie theory, resulting in a conceptually satisfying final result. As Kalman filters and UKFs have been extensively studied [19], [29], basic knowledge is assumed.

3.3 Unscented Transformation

Unscented Kalman Filtering relies on the the Unscented Transformation, a deterministic sampling of a non-linear function, to estimate how the statistical properties change between the function's inputs and outputs [30]. The Unscented Transformation considers a random variable x with mean $\bar{x}$ and covariance $P^{xx} \in \mathbb{R}^L$ propagated through a non-linear function $y = f(x)$.

First, a set of $2L + 1$ sample points, known as *sigma points*, are created using the columns of the square root of the covariance matrix P^{xx}. The matrix $\sqrt{P^{xx}}$ can be calculated using the Cholesky decomposition of P^{xx}.

$$\mathcal{X} = \left\{\bar{x}, \bar{x} + \sqrt{\mathcal{W}_i^e P_i^{xx}}, \bar{x}\right) - \sqrt{\mathcal{W}_i^e P_i^{xx}}\}, \quad i \in [1, L] \tag{39}$$

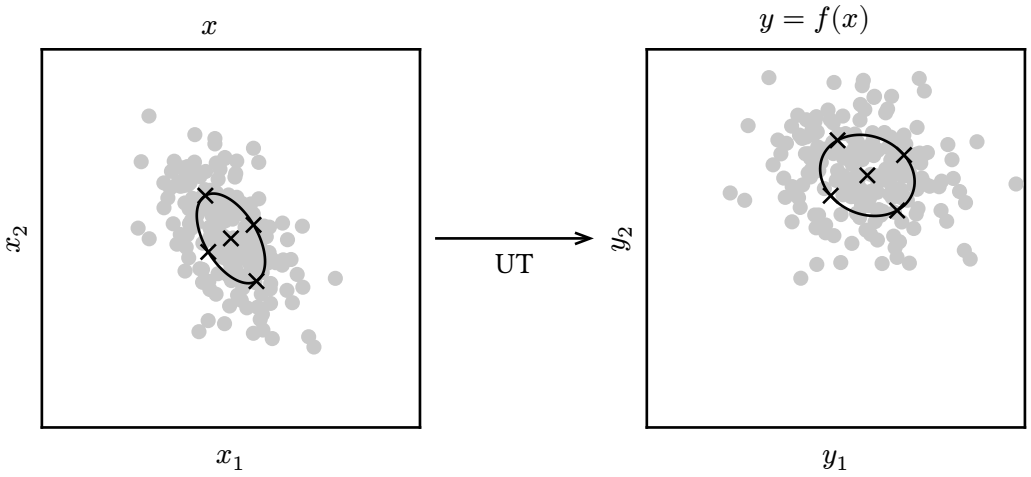

Figure 3: Unscented Transformation

An illustrative example of the unscented transformation. The mean and covariance of a transformation $y = f(x)$ is estimated by propagating *sigma points* (denoted by $\times$) through the transformation.

The sigma points are then propagated through the non-linear function $f(\boldsymbol{x})$ to create a set of points $\mathcal{Y}$.

$$\mathcal{Y} = \{f(\mathcal{X}_i)\}, \quad i \in [1, 2L + 1] \tag{40}$$

The set of points $\mathcal{Y}$ is then assumed to be a collection of samples of the Gaussian random variable y. With this assumption the mean $\bar{y}$ and covariance P^{yy} of y can be calculated along with the cross-covariance between x and y.

$$\bar{y} \simeq \sum_{i=1}^{2L+1} \mathcal{W}_i^m \mathcal{Y}_i \tag{41}$$

$$P^{yy} \simeq \sum_{i=1}^{2L+1} \mathcal{W}_i^c (\mathcal{Y}_i - \bar{y})(\mathcal{Y}_i - \bar{y})^\top \tag{42}$$

$$P^{xy} \simeq \sum_{i=1}^{2L+1} \mathcal{W}_i^c (\mathcal{X}_i - \bar{x})(\mathcal{Y}_i - \bar{y})^\top \tag{43}$$

The weights in the above equations correspond to the sigma point expansion weights $\mathcal{W}^e$, the mean contraction weights $\mathcal{W}^m$, and the covariance contraction weights $\mathcal{W}^c$. The goal of these weights is to create a combination of expansions and contractions that cancel each-other and do not scale the final result [31].

$$\mathcal{W}^e = \{(L+\lambda)_i\}, \quad i \in [1, L] \tag{44}$$

$$\mathcal{W}^m = \left\{ \frac{\lambda}{L+\lambda}, \left(\frac{1}{2(L+\lambda)} \right)_i \right\}, \quad i \in [1, 2L] \tag{45}$$

$$\mathcal{W}^c = \left\{ \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, \left(\frac{1}{2(L+\lambda)} \right)_i \right\}, \quad i \in [1, 2L] \tag{46}$$

The common factor λ is calculated using the two of the tuning parameters associated with the unscented transform: α and κ.

$$\lambda = \alpha^2 (L + \kappa) - L \tag{47}$$

The tuning parameters of the unscented transformation control how the Gaussian distribution is captured and directly affects the error of the mean and covariance estimates [30], [31], [32]. While many heuristics exist for how these parameters should be set, work by Nielsen et al. shows that those methods do not result in a UKF that accurately captures the statistical properties of underlying random variable [33]. In spite of this fact, for the sake of simplicity this thesis follows the heuristics laid out in [31]. The tuning parameters, along with their traditionally understood descriptions, are as follows:

- $\alpha \in (0, 1]$, which controls the spread of sigma points around $\bar{x}$. With $\alpha \to 0$ contracting the sigma points around $\bar{x}$ and $\alpha = 1$ leaving the sigma points unaffected.
- $\beta \in [0, \infty)$, which incorporates knowledge of the distribution of x.
- $\kappa \in (-\infty, \infty)$, which controls the spread of sigma points around $\bar{x}$. Positive values expand the sigma points around $\bar{x}$ while negative values contract them. While this parameter is partially redundant to α it is important to note that negative values of κ can lead to numerical instability.

3.4 Quaternion Covariance and Lie Theory

While quaternions offer minimal, singularity free representations of attitude, it has long been noted that its associated $\mathbb{R}^4$ covariance is unsuitable for use in Kalman filters due to its rank deficiency. This is a consequence of the unit norm constraint [34], [35], [36]. The rank deficiency can result singular or ill-conditioned covariance matrices under certain conditions. Carmi et al. present an illustrative example of this phenomenon in two dimensions [36]. Their example is reproduced here to provide a sense of the issue at hand:

Consider a random vector $x \in \mathbb{R}^2$, constrained to the unit circle:

$$\boldsymbol{x} \triangleq [\cos(\theta), \sin(\theta)]^\top, \quad \theta \sim \mathcal{U}\left(-\frac{a}{2}, \frac{a}{2}\right), \quad a \in \mathbb{R} \tag{48}$$

The eigenvalues of the associated covariance matrix $\mathrm{cov}(\boldsymbol{x})$ are:

$$\lambda_1 = \frac{1}{2} + \frac{\sin(a)}{2a} - \left(\frac{2\sin\left(\frac{a}{2}\right)}{a}\right)^2, \quad \lambda_2 = \frac{1}{2} - \frac{\sin(a)}{2a} \tag{49}$$

Under specific conditions, such as $a \to 0$, the covariance matrix becomes ill-conditioned.

$$\lim_{a \to 0} \frac{\lambda_1}{\lambda_2} = 0 \tag{50}$$

Several proposed solutions to this problem exist, from the attitude error vector state representation [16], to the application of manifold theory [12]. Perhaps the most conceptually satisfying option is the use of Lie theory in the development of the Kalman filter [27], [31], [37]. Lie theory notes that unit quaternions form a Lie group, thus it is possible to project unit quaternions onto a three dimensional space tangent to a location on the unit quaternion manifold. Algebraic operations in this lower dimensional tangent space, such as vector addition and subtraction, form the Lie algebra of the unit quaternion. It is possible to replace Euclidean operations at various points in the Kalman filter with their Lie algebra counterparts to express the covariance associated with the quaternion state in its $\mathbb{R}^3$ tangent space, resolving the aforementioned issues with $\mathbb{R}^4$ covariance matrices [31].

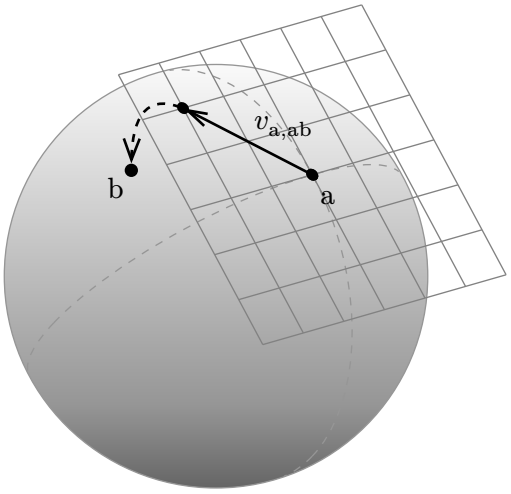

Figure 4: Visualization of Lie Algebra

Representing the four dimensional unit quaternion manifold as a three dimensional sphere, Figure 4 depicts how a vector displacement in the Lie algebra of the quaternion a maps to a new quaternion b on the surface of the manifold.

Several operations in the unscented transformation, such as the additive perturbations in (39), have no immediate equivalent on the quaternion manifold. Here it is useful to turn to Lie theory to develop the operation necessary to perform the unscented transformation on a quaternionic variable.

Quaternion Difference

Since Lie algebra is performed in a space tangent to the unit quaternion manifold, it is required to define the point of tangency when using Lie algebraic operations. For the following equations the Lie algebra is defined to be tangent to quaternion a. The Lie algebraic difference between two unit quaternions a and b is given by (51), where the resulting vector difference $v_{\mathrm{a,ab}}$ is defined in the tangent space established by quaternion a.

$$\mathrm{b} \ominus \mathrm{a} \triangleq \ln(\mathrm{b} \otimes \mathrm{a}^*) = v_{\mathrm{a,ab}} \in \mathbb{H}^{\mathrm{p}} \tag{51}$$

Quaternion Displacement

Similarly, the Lie algebraic displacement of a unit quaternion a by a vector $v_{\mathrm{a,ab}}$ in its tangent space is given by (52), where the resulting quaternion b is projected back onto the unit quaternion manifold from the tangent space of a.

$$\mathrm{a} \oplus v_{\mathrm{a,ab}} \triangleq \exp\left(v_{\mathrm{a,ab}}\right) \otimes \mathrm{a} = \mathrm{b} \in \mathbb{H}^{\mathrm{u}} \tag{52}$$

For any practical implementation of a Kalman filter for attitude determination, other states such as the biases of various sensors will be estimated alongside the quaternion attitude. Since these states are Euclidean, it is not necessary to involve Lie theory. It is useful in this case to consider the state vector in a partitioned manner, consisting of quaternionic and euclidean states. For the quaternionic states, addition and subtraction become the Lie algebra operations defined earlier. The Euclidean states require no special treatment. The operators $\oplus$ and $\ominus$ will be used from here on out to represent this mixed treatment of the state vector.

Quaternion Mean

A final operation must defined in order to calculate the covariance of a unit-quaternion attitude and that is the mean of a set of quaternions. Specifically, for the implementation of the Unscented Kalman Filter in the next section, the *weighted-mean* will be required. Several techniques exist to determine the weighted mean of a quaternion, often iterative methods [31]. The closed form calculation presented by Markley et al. will be used in this thesis [38]. Given a set of n unit-quaternions $\mathcal{Q}$ and scalar weights $\mathcal{W}$ the average quaternion $\bar{\mathrm{q}}$ is calculated as the eigenvector associated with the largest eigenvalue of the intermediate matrix M.

$$M \triangleq \sum_{i=1}^{n} \mathcal{W}_i \mathcal{Q}_i \mathcal{Q}_i^{\top} \tag{53}$$

The largest eigenvalue and vector can be readily found through the SVD decomposition of M. In the case of mixed state vectors, quaternionic states will be treated with this method while euclidean states will be calculated via the normal weighted mean equation.

3.5 Extension to Dual Quaternions

The formalism developed for quaternions naturally extends to dual quaternions, allowing consistent handling of covariance and state propagation in the higher-dimensional manifold. Building on the established Lie algebra framework, the difference, displacement, and weighted mean operation can be redefined to accommodate the structure of the dual quaternion.

Dual Quaternion Difference

Extending the difference operation to dual quaternions is not as straightforward as applying the natural logarithm to their product. The coupling of orientation and position in the dual term of a unit dual quaternion representing a pose (13) requires additional considerations to maintain the validity of the joint term. Inspired by Li et al. [39], the apporiate treatment of the dual quaternion difference requires decoupling of the position from orientation before computing the difference.

$$\mathbf{b} \ominus \mathbf{a} \triangleq \ln(\mathrm{b_r} \otimes \mathrm{a_r}^*) + \epsilon(\mathrm{a_r}^* \otimes (2\mathrm{b_r}^* \otimes \mathrm{b_d} - 2\mathrm{a_r}^* \otimes \mathrm{a_d})) \in \mathbb{H}_{\mathrm{d}}^{\mathrm{p}} \tag{54}$$

Dual Quaternion Displacement

Similarly, the dual quaternion displacement operation requires that the position of the pose representation be decoupled from the orientation prior to calculating the displacement.

$$\mathbf{a} \oplus v_{\mathrm{a,ab}} \triangleq v_{(\mathrm{a,ab})_{\mathrm{r}}} \otimes \mathrm{a_r} + \epsilon\left(\frac{1}{2}\left(v_{(\mathrm{a,ab})_{\mathrm{d}}} + \mathrm{a_r}^* \otimes \mathrm{a_d}\right)\right) \in \mathbb{H}_{\mathrm{d}}^{\mathrm{u}} \tag{55}$$

Dual Quaternion Mean

Finally, like the previous operations, the dual quaternion representing the mean pose of a set of poses is obtained by combining the quaternion mean of their orientations with the Euclidean mean of their positions.

$$M \triangleq \sum_{i=1}^{n} \mathcal{W}_{\mathrm{r},i} \mathcal{Q}_{\mathrm{r},i} \mathcal{Q}_{\mathrm{r},i}^{\top} + \epsilon \frac{1}{2}\left(\sum_{i=1}^{n} \mathcal{W}_{\mathrm{r},i} \mathcal{Q}_{\mathrm{r},i} \mathcal{Q}_{\mathrm{r},i}^{\top} \otimes \sum_{i=1}^{n} \mathcal{W}_{\mathrm{d},i} \mathcal{Q}_{\mathrm{r},i}^{*} \otimes \mathcal{Q}_{\mathrm{d},i}\right) \tag{56}$$

3.6 Measurement Model

The pose of the quadrotor can be reconstructed using measurements of the acceleration vector, the magnetic field vector, and the GPS position. According to [27], [40], assuming that the accelerometer primarily measures the vector along which gravity acts, it can be used to construct a quaternion that rotates the body frame to an intermediate reference frame $\mathcal{F}^{\mathrm{Z}}$, whose z-axis aligns with the NED z-axis. Using the normalized acceleration measurement $\tilde{a}$, the quaternion can be constructed as:

$$\tilde{\mathrm{q}}_{\mathrm{Z/B}} = \begin{cases} \left(\left[-\frac{\tilde{a}_y}{2\lambda_1} \;\; \frac{\tilde{a}_x}{2\lambda_1} \;\; 0\right]^{\top}, \lambda_1\right), & \tilde{a}_z \geq 0 \\ \left(\left[\lambda_2 \;\; 0 \;\; \frac{\tilde{a}_x}{2\lambda_2}\right]^{\top}, -\frac{\tilde{a}_y}{2\lambda_2}\right), & \tilde{a}_z < 0 \end{cases} \tag{57}$$

Where :

$$\lambda_1 = \sqrt{\frac{\tilde{a}_z + 1}{2}}, \quad \lambda_2 = \sqrt{\frac{1 - \tilde{a}_z}{2}} \tag{58}$$

Similarly, using the magnetometer reading a quaternion can be constructed that rotates the inertial frame about its z-axis to align to the intermediate $\mathcal{F}^{\mathrm{Z}}$ frame. First the normalized measured magnetic feild vector $\tilde{b}$ is rotated into the $\mathcal{F}^{\mathrm{Z}}$ frame:

$$\tilde{l} = \tilde{\mathrm{q}}_{\mathrm{Z/B}}^{*} \otimes \tilde{b} \otimes \tilde{\mathrm{q}}_{\mathrm{Z/B}} \tag{59}$$

This is then used to construct the a quaternion that rotates the inertial frame to the intermediate frame:

$$\tilde{\mathrm{q}}_{\mathrm{I'/Z}} = \begin{cases} \left(\left[0 \;\; 0 \;\; \frac{\lambda_3}{\sqrt{2\Gamma}}\right]^{\top}, \frac{\lambda_3}{\sqrt{2\Gamma}}\right), & l_x \geq 0 \\ \left(\left[0 \;\; 0 \;\; \frac{\lambda_4}{\sqrt{2\Gamma}}\right]^{\top}, \frac{l_y}{\sqrt{2}\lambda_4}\right), & l_x < 0 \end{cases} \tag{60}$$

Where:

$$\Gamma = l_x^2 + l_y^2, \quad \lambda_3 = \sqrt{\Gamma + l_x\sqrt{\Gamma}}, \quad \lambda_4 = \sqrt{\Gamma - l_x\sqrt{\Gamma}} \tag{61}$$

Finally, the body attitude is reconstructed as:

$$\tilde{\mathrm{q}}_{\mathrm{B/I}} = \left(\tilde{\mathrm{q}}_{\mathrm{Z/B}} \otimes \tilde{\mathrm{q}}_{\mathrm{I'/Z}}\right)^* \otimes \mathrm{q}_{\mathrm{I'/I}} \otimes \left(\tilde{\mathrm{q}}_{\mathrm{Z/B}} \otimes \tilde{\mathrm{q}}_{\mathrm{I'/Z}}\right) \tag{62}$$

The dual quaternion representing pose can be simply constructed as:

$$\tilde{\mathbf{q}}^{\mathbf{B}}_{\mathbf{B/I}} = \tilde{\mathrm{q}}_{\mathrm{B/I}} + \epsilon\left(\frac{1}{2}\tilde{\mathrm{q}}_{\mathrm{B/I}}\tilde{r}^{\mathrm{B}}_{\mathrm{B/I}}\right) \tag{63}$$

Where $\tilde{r}^{\mathrm{B}}_{\mathrm{B/I'}}$ is the GPS position of the quadrotor rotated into the body frame:

$$\tilde{r}^{\mathrm{B}}_{\mathrm{B/I}} = \tilde{\mathrm{q}}^*_{\mathrm{B/I}} \otimes \tilde{r}^{\mathrm{I}}_{\mathrm{B/I}} \otimes \tilde{\mathrm{q}}_{\mathrm{B/I}} \tag{64}$$

In practice, the GPS provides measurements at a much slower rate than the other sensors on board the quadrotor. Rather than limiting the UKF to run at the frequency of the GPS update, the previous estimate of position can be used in the measurement model when no new GPS measurement is available. This will dead reckon the vehicles position based on the accelerometer measurements.

The above equations can be expressed as the non-linear measurement function $g(\tilde{m}_k)$ where the measurement vector $\tilde{m}_k$ is:

$$\tilde{m}_k = \begin{bmatrix}\tilde{a}_k^\top & \tilde{b}_k^\top & \tilde{r}_k^\top\end{bmatrix}^\top \tag{65}$$

As the measurement function is non-linear, the mean measurement y_k and its associated covariance can R_k can be found but propagating the individual sensor covariances through the unscented transformation:

$$\begin{aligned}
\mathcal{M}_k &= \left\{\tilde{m}_k, \tilde{m}_k \oplus \sqrt{\mathcal{W}_i^e R_{\tilde{m},i}}, \tilde{m}_k \oplus -\sqrt{\mathcal{W}_i^e R_{\tilde{m},i}}\right\}, \quad i \in [1, L] \\
\mathcal{M}_{k|k} &= \left\{g\left(\mathcal{M}_{k,i}\right)\right\}, \quad i \in [1, 2L+1] \\
y_k &= \mathrm{mean}\Big(\left\{\mathcal{M}_{k|k}\right\}, \left\{\mathcal{W}^m\right\}\Big) \\
R_k &= \sum_i \mathcal{W}_i^c\left(\mathcal{M}_{k|k,i} \ominus y_k\right)\left(\mathcal{M}_{k|k,i} \ominus y_k\right)^\top
\end{aligned} \tag{66}$$

3.7 Process Model

A prediction of the quadrotors state can be constructed through knowledge of the kinematics of the vehicle. First the measured angular rate is treated by removing the estimated bias:

$$\hat{\omega}_k = \tilde{\omega}_k - \hat{\beta}_{\omega,k} \tag{67}$$

At the same time an estimate of the vehicles velocity can be propagated with the readings taken from the accelerometer:

$$\hat{v}_k = \hat{v}_{k-1} + (\tilde{a}_k + g)\Delta t \tag{68}$$

The angular rate and translational velocity are then fused into an estimate of the vehicles dual quaternion twist at the current time-step:

$$\hat{\boldsymbol{\xi}}_k = \hat{\omega}_k + \epsilon(\hat{v}_k + \hat{\omega}_k \times \hat{r}_k) \tag{69}$$

Finally the previous pose of the vehicle can be propagated to yield an estimate of the pose at the current time-step:

$$\hat{\mathbf{q}}_k = \hat{\mathbf{q}}_{k-1} \otimes \exp\left(\frac{1}{2}\hat{\boldsymbol{\xi}}_k \Delta t\right) \tag{70}$$

These equations comprise the nonlinear process model $f(\hat{x}_{k-1}, \tilde{m}_k, \tilde{\omega}_k)$.

3.8 Observation Model

As all state of the UKF are directly observable except for the rate biases, a simple observation model $h(\hat{x}_k)$ can be used that discards the bias from the state.

3.9 Lie Algebraic Unscented Kalman Filter

Combining the traditional UKF with the concepts from Lie Algebra results in the following LAUKF algorithm [27], [31]. It is worth noting that with the appropriate use of the quaternion and dual quaternion operations, this algorithm is equally valid for a state with a quaternion representation of attitude and a Euclidean representation of position as for a state with a dual quaternion representation of pose.

State estimate:

$$\begin{aligned}
\mathcal{X}_{k-1} &= \left\{\hat{x}_{k-1}, \hat{x}_{k-1} \oplus \sqrt{\mathcal{W}_i^e P_{k-1,i}^{xx}}, \hat{x}_{k-1} \oplus -\sqrt{\mathcal{W}_i^e P_{k-1,i}^{xx}}\right\}, \quad i \in [1, L] \\
\mathcal{X}_{k|k-1} &= \left\{f(\mathcal{X}_{k-1,i}; \tilde{\omega}_k)\right\}, \quad i \in [1, 2L+1] \\
\hat{x}_{k|k-1} &= \operatorname{mean}\Big(\left\{\mathcal{X}_{k|k-1}\right\}, \{\mathcal{W}^m\}\Big) \\
P_{k|k-1}^{xx} &= \sum_i \mathcal{W}_i^c \Big(\mathcal{X}_{k|k-1,i} \ominus \hat{x}_{k|k-1}\Big)\Big(\mathcal{X}_{k|k-1,i} \ominus \hat{x}_{k|k-1}\Big)^\top + Q
\end{aligned} \tag{71}$$

Measurement update:

$$\begin{aligned}
\mathcal{Y}_{k|k-1} &= \left\{h\Big(\mathcal{X}_{k|k-1,i}\Big)\right\}, \quad i \in [1, 2L+1] \\
\hat{y}_{k|k-1} &= \operatorname{mean}\Big(\left\{\mathcal{Y}_{k|k-1}\right\}, \{\mathcal{W}^m\}\Big) \\
P_{k|k-1}^{yy} &= \sum_i \mathcal{W}_i^c \Big(\mathcal{Y}_{k|k-1,i} \ominus \hat{y}_{k|k-1}\Big)\Big(\mathcal{Y}_{k|k-1,i} \ominus \hat{y}_{k|k-1}\Big)^\top + R_k \\
P_{k|k-1}^{xy} &= \sum_i \mathcal{W}_i^c \Big(\mathcal{X}_{k|k-1,i} \ominus \hat{x}_{k|k-1}\Big)\Big(\mathcal{Y}_{k|k-1,i} \ominus \hat{y}_{k|k-1}\Big)^\top
\end{aligned} \tag{72}$$

State update:

$$\begin{aligned}
K_k &= P_{k|k-1}^{xy}\Big(P_{k|k-1}^{yy}\Big)^{-1} \\
\hat{x}_k &= \hat{x}_{k|k-1} \oplus K_k\Big(y_k - \hat{y}_{k|k-1}\Big) \\
P_k^{xx} &= P_{k|k-1}^{xx} - K_k P_{k|k-1}^{yy} K_k^\top
\end{aligned} \tag{73}$$

3.10 Implementation Details

A key step in the computation of the LAUKF algorithm involves generating sigma points via the matrix square root of the covariance matrix. However, this presents a challenge since the square root of a matrix is not always guaranteed to exist. The authors of [31] and [27] recommend using the Cholesky decomposition to compute the matrix square root.

In practice, this operation often fails, particularly when using the dual quaternion representation of pose. One potential remedy is to perturb the diagonal elements of the covariance matrix and retry the decomposition when the initial attempt fails. While this approach worked for the quaternion-based LAUKF, it introduced additional computational overhead and was ultimately ineffective for pose representations using dual quaternions.

To address this, a simple fallback method for approximating the matrix square root was employed whenever the Cholesky decomposition failed.

$$\sqrt{P} \approx \frac{P + I}{2} \tag{74}$$

4 Control

This chapter examines the design of a quaternion-based trajectory and attitude controller. A natural question that arises, given that this thesis has presented a *dual quaternion* framework for quadrotor dynamics and pose estimation, is why not implement the control scheme with dual quaternions? While dual quaternions provide a compact and efficient 3D pose representation, their integration into control algorithms remains an area of active research [10], [14], [41]. Notably, few dual quaternion control schemes have been published for under-actuated systems like quadrotors [10]. The unit dual quaternion's double coverage of Euclidean space, where two distinct dual quaternions can represent the same pose, presents a challenge for control schemes. If not accounted for, the controller could potentially destabilize due to using the incorrect representation, a phenomenon known as *unwinding* [42]. Since the main objective of this thesis is to develop and evaluate the dual quaternion UKF, a simpler control scheme was chosen to facilitate testing without introducing the added complexities associated with dual quaternion-based control.

The controller presented here is a Proportional-Derivative controller with feed-forward (PD+) based off work presented by Andersen et al. [43]. It consists of three subsystems: the guidance controller, attitude controller, and translation controller. The guidance controller defines the desired attitude of the quadrotor based on a target position. The attitude controller then attempts to track this desired attitude and calculates the necessary torque to correct the orientation. Finally, the translation controller manages the quadrotor's translation toward the target position, calculating the desired thrust force.

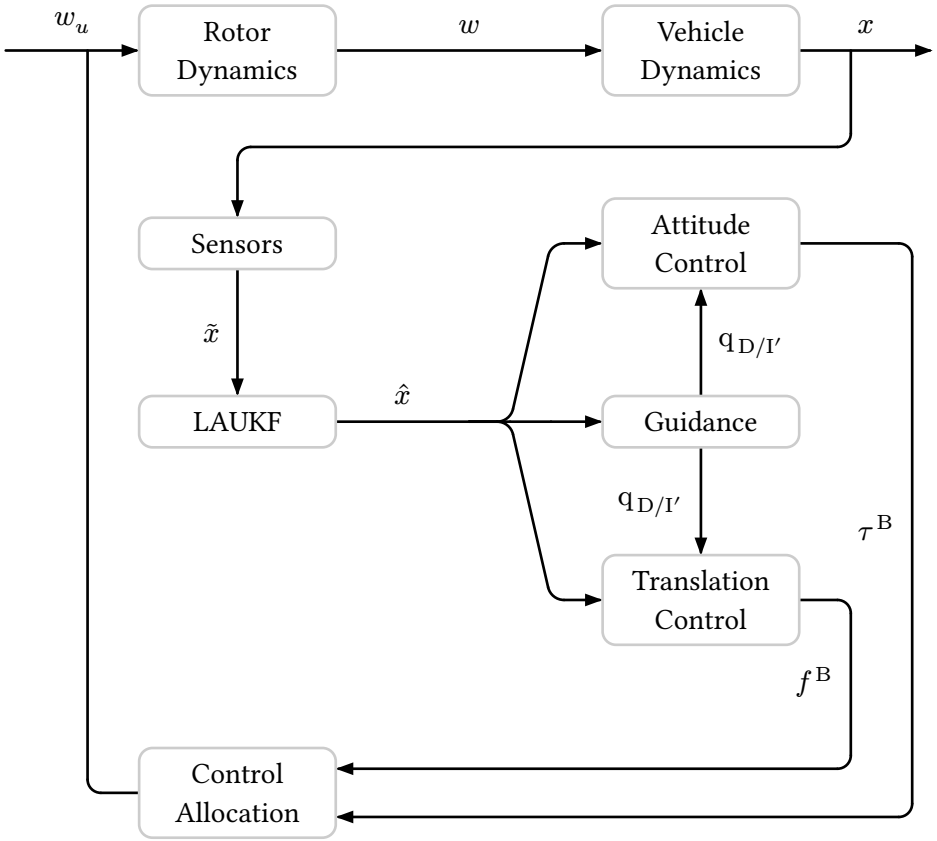

Figure 5: Control System Block Diagram

4.1 Guidance

The guidance system in Andersen et al. operates on the notion that the quadrotor translational control is limited to its thrust axis. Therefore the role of the guidance system is to generate a desired orientation that aligns the thrust axis with the positional error.

Position Error

If the attitude controller is to be developed using traditional aerospace conventions, the NED frame must be used when expressing the guidance error:

$$e^{\mathrm{I}'}_{\mathrm{D/B}} = r^{\mathrm{I}'}_{\mathrm{D/I}'} - r^{\mathrm{I}'}_{\mathrm{B/I}'} \tag{75}$$

Similarly the velocity error is:

$$\dot{e}^{\mathrm{I}'}_{\mathrm{D/B}} = -\dot{r}^{\mathrm{I}'}_{\mathrm{B/I}'} \tag{76}$$

Desired Orientation

The desired orientation is that which would align the entirety of the positional error with the thrust axis of the vehicle. This can be expressed as:

$$e^{\mathrm{D}}_{\mathrm{D/B}} = \begin{bmatrix} 0 & 0 & -\|e^{\mathrm{I}'}_{\mathrm{D/B}}\| \end{bmatrix}^{\top} = \mathrm{q}^{*}_{\mathrm{D/I}'} \otimes e^{\mathrm{I}'}_{\mathrm{D/B}} \otimes \mathrm{q}_{\mathrm{D/I}'} \tag{77}$$

The desired orientation $\mathrm{q}^{*}_{\mathrm{D/I}'}$ can then be derived using the axis-angle parameterization of a quaternion (6):

$$\mathrm{q}_{\mathrm{D/I}'} = \left(n \sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right) \right) \tag{78}$$

The rotation axis can be calculated as:

$$n = \frac{e^{\mathrm{D}}_{\mathrm{D/B}} \times e^{\mathrm{I}'}_{\mathrm{D/B}}}{\|e^{\mathrm{D}}_{\mathrm{D/B}} \times e^{\mathrm{I}'}_{\mathrm{D/B}}\|} \tag{79}$$

Andersen et. al [43] notes that it is desirable to impose limits on the generated guidance orientation. Without saturation, it is possible for the guidance system that would pitch or roll the quadrotor over so far that would not be able to generate enough thrust to maintain altitude. Thus they propose (80) and the introduction of two control parameters k_1 and k_2, where k_1 imposes a maximum pitch or roll, and k_2 controls the aggressiveness of the guidance system.

$$\theta = k_1 \arctan\left(k_2 \, \|e^{\mathrm{I}'}_{\mathrm{D/B}}\|\right), \quad k_1, k_2 \in \mathbb{R} \tag{80}$$

Desired Rate

To develop a PD attitude controller, it is necessary to derive the desired angular rate $\omega^{\mathrm{D}}_{\mathrm{D/I}'}$, which specifies how fast the orientation should change to maintain alignment with the target position. Differentiating (77) with respect to time yields:[1]

$$\dot{e}^{\mathrm{D}}_{\mathrm{D/B}} = \mathrm{q}^{*}_{\mathrm{D/I}} \otimes \dot{e}^{\mathrm{I}}_{\mathrm{D/B}} \otimes \mathrm{q}_{\mathrm{D/I}} + e^{\mathrm{D}}_{\mathrm{D/B}} \times \omega^{\mathrm{D}}_{\mathrm{D/I}} \tag{81}$$

Rearranging (81) the rate $\omega^{\mathrm{D}}_{\mathrm{D/I}'}$ can be isolated:

$$\omega^{\mathrm{D}}_{\mathrm{D/I}'} = \left[e^{\mathrm{D}}_{\mathrm{D/B}}\right]^{\dagger}_{\times} \left(\dot{e}^{\mathrm{D}}_{\mathrm{D/B}} - \mathrm{q}^{*}_{\mathrm{D/I}'} \otimes \dot{e}^{\mathrm{I}'}_{\mathrm{D/B}} \otimes \mathrm{q}_{\mathrm{D/I}'}\right) \tag{82}$$

[1] This thesis derives a slightly different version of (81) compared to Andersen et. al [43].

Where $[\cdot]_\times$ is a skew-symmetric matrix parameterized by a vector and † denotes the Moore-Penrose pseudo-inverse. Noting that $\left[e^{\mathrm{D}}_{\mathrm{D/B}}\right]^{\dagger}_{\times} \dot{e}^{\mathrm{D}}_{\mathrm{D/B}} = 0$ the expression simplifies to:

$$\omega^{\mathrm{D}}_{\mathrm{D/I'}} = \left[e^{\mathrm{D}}_{\mathrm{D/B}}\right]^{\dagger}_{\times} \left(\mathrm{q}^{*}_{\mathrm{D/I'}} \otimes \dot{e}^{\mathrm{I'}}_{\mathrm{D/B}} \otimes \mathrm{q}_{\mathrm{D/I'}}\right) \tag{83}$$

4.2 Attitude Control

Andersen et al. present a quaternion-based PD+ attitude controller that tracks the desired orientation by minimizing both the attitude and rate errors. To enhance performance, a feed-forward term is included to account for the vehicle's rotational inertia. While Andersen et al. define the quaternion error using the small-angle assumption as described in [16], this thesis adopts an alternative approach by defining the attitude error through the Lie algebraic difference (51). This method provides a more precise representation of orientation discrepancies, particularly for larger angles.

$$\mathrm{e}_{\mathrm{B/D}} = \mathrm{q}_{\mathrm{B/I'}} \ominus \mathrm{q}_{\mathrm{D/I'}} \tag{84}$$

The angular rate error is defined as:

$$\omega^{\mathrm{B}}_{\mathrm{D/B}} = \omega^{\mathrm{B}}_{\mathrm{B/I}} - \mathrm{q}^{*}_{\mathrm{B/D}} \otimes \omega^{\mathrm{D}}_{\mathrm{D/I'}} \otimes \mathrm{q}_{\mathrm{B/D}}, \quad \mathrm{q}_{\mathrm{B/D}} \triangleq \mathrm{q}^{*}_{\mathrm{D/I'}} \otimes \mathrm{q}_{\mathrm{B/I'}} \tag{85}$$

The attitude controller is then defined as:[2]

$$\tau^{\mathrm{B}} = \omega^{\mathrm{B}}_{\mathrm{B/I}} \times J\omega^{\mathrm{B}}_{\mathrm{B/I}} - k_{\mathrm{e}}\mathrm{e}_{\mathrm{B/D}} - k_{\omega}\omega^{\mathrm{B}}_{\mathrm{D/B}} \tag{86}$$

4.3 Translation Control

The translation controller is responsible for managing the quadrotor's translational towards the target point. Given the position and velocity errors:

$$\begin{aligned} e^{\mathrm{I}}_{\mathrm{D/B}} &= r^{\mathrm{I}}_{\mathrm{D/I}} - r^{\mathrm{I}}_{\mathrm{B/I}} \\ \dot{e}^{\mathrm{I}}_{\mathrm{D/B}} &= -\dot{r}^{\mathrm{I}}_{\mathrm{B/I}} \end{aligned} \tag{87}$$

The PD+ controller becomes:[3]

$$f^{\mathrm{I}} = f^{\mathrm{I}}_{g} - k_p e^{\mathrm{I}}_{\mathrm{D/B}} - k_d \dot{e}^{\mathrm{I}}_{\mathrm{D/B}} \tag{88}$$

The desired thrust can be expressed in the body frame as:

$$f^{\mathrm{B}} = \begin{bmatrix} 0 & 0 & \|f^{\mathrm{I}}\| \end{bmatrix}^{\top} \tag{89}$$

4.4 Control Allocation

With the desired forces and torques calculated by the controller, the control inputs for the individual rotors can be found using the control allocation equation. Rearranging (25):

$$\begin{bmatrix} w_1^2 & w_2^2 & w_3^2 & w_4^2 \end{bmatrix}^{\top} = A^{-1} \begin{bmatrix} f^{\mathrm{B}} & \tau^{\mathrm{B}}_{x} & \tau^{\mathrm{B}}_{y} & \tau^{\mathrm{B}}_{z} \end{bmatrix}^{\top} \tag{90}$$

[2] This thesis does not consider tracking a moving target, hence the attitude controller is slightly simplified from Andersen et. al [43]

[3] This thesis does not consider tracking a moving target, hence the translation controller is slightly simplified from Andersen et. al [43]

4.5 Limitations

While this controller is conceptually elegant and straightforward to implement, it does present some limitations. The individual attitude and translation controllers are capable, yet the guidance subsystem can produce undesirable behavior under specific conditions. Andersen et al. [43] identifies degenerate cases where $\|e^{\mathrm{D}}_{\mathrm{D/B}} \times e^{\mathrm{I}'}_{\mathrm{D/B}}\| = 0$ or $\|^{\mathrm{I}'}_{\mathrm{D/B}}\| = 0$. However, these represent situations where the guidance system becomes numerically degenerate. In practical tests it was observed that when the quadrotor's thrust axis approaches perfect alignment with the positional error vector ($\|e^{\mathrm{D}}_{\mathrm{D/B}} \times e^{\mathrm{I}'}_{\mathrm{D/B}}\| \to 0$), the vehicle may destabilize. This destabilization was not due to numerical issues in the guidance system, but because of severe chatter in the desired attitude.

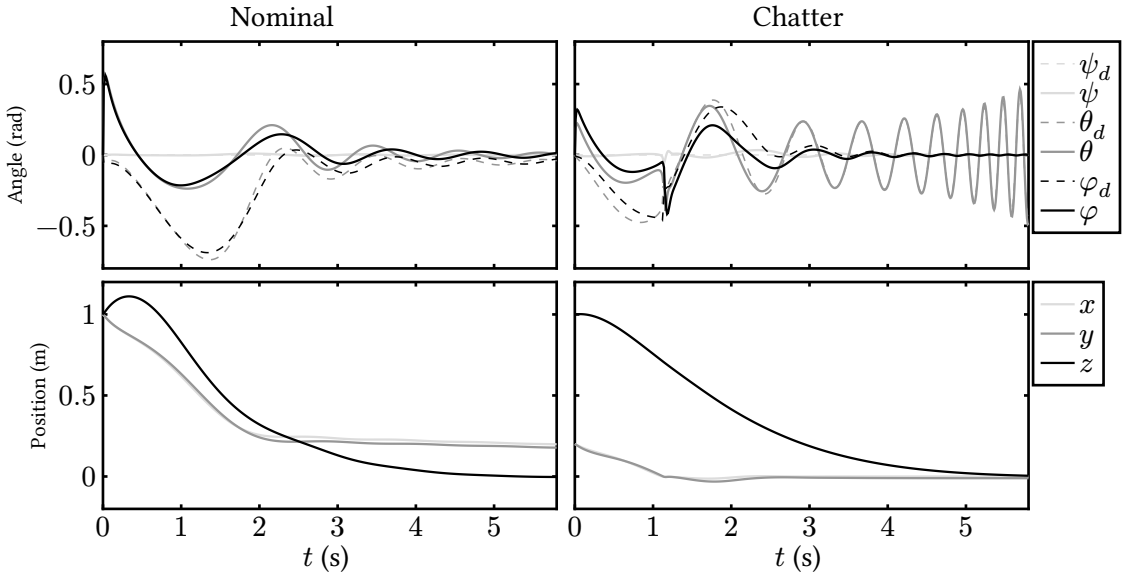

Figure 6: Guidance Nominal Vs. Chatter Behavior

Nominal performance is exhibited when the quadrotor approaches the origin from the initial condition $r_o = [1 \ 1 \ 1]^\top$, avoiding alignment between the thrust axis and error vector. Chatter is exhibited when approaching the origin from the initial condition $r_o = [0.1 \ 0.1 \ 1]^\top$. Alignment of the thrust axis and error vector is achieved at about $t = 1$s, resulting in destabilization of the desired pitch angle.

In cases of time-varying trajectories, as examined in the original paper, this scenario is unlikely. However, when approaching a desired position from directly above or below (such as during takeoff), this chatter behavior can occur. To prevent chatter, it is effective to approach the target position primarily within the xy-plane of the inertial frame. Nonetheless, this approach can introduce a significant steady-state positional error due to the absence of an integral error term in the guidance controller.

This behavior is acceptable for the purposes of this thesis, as the focus is on testing the LAUKF rather than implementing a robust guidance system. It is only necessary that the vehicle remains within a certain spatial bound during testing for practical reasons. In simulations, the initial positional error of the quadrotor can be configured to lie primarily in the xy plane. For the physical vehicle, takeoff can rely solely on attitude and translation control until the quadrotor reaches the same plane as the target position.

5 Flight Computer Design

This chapter provides an overview of the electrical design of the flight computer. It begins by outlining the selected sensors, including their performance characteristics and the criteria used for their selection. The chapter then explains the implementation of the flight computer software, which is written in Rust. It includes a discussion of the benefits and limitations associated with using this programming language in the context of the flight computer.

5.1 Electrical Design

The electrical design prioritized manufacturability above all else. The primary constraint was that the assembly should be achievable by a reasonably skilled individual using minimal specialized equipment. While low-cost turnkey circuit board assembly services are widely available, they come with trade-offs, including longer turnaround times and restrictions on component selection. For this thesis, hand-assembling the flight computer eliminated part-selection limitations and reduced the time from completed design to prototype by approximately four weeks.

Using solder paste with an appropriate stencil, components as small as imperial 0402 were successfully placed by hand with the aid of vacuum tweezers and optical magnification. Designing the board with all components on a single side enabled re-flow soldering with affordable ovens while still achieving acceptable results. Details of the electrical and Printed Circuit Board (PCB) design are included in the Appendix. Manufacturing a complete board took approximately two hours. It was observed that the alignment of the solder paste stencil with the PCB during solder application was the most critical factor affecting the quality of the final product. Therefore, any investments to improve the manufacturing process should focus on this step.

Table 3 highlights the key components used in the flight computer along with performance metrics relevant to the Unscented Kalman Filter.

Function	**Part**	**Performance**	
MCU	Atmel SAMV71	CPU Type	Cortex-M7
		Clock Speed	300MHz
		Ram	384Kb
		Program Memory	2048Kb
Accelerometer + Gyroscope	Bosch BMI088	Resolution	A: 0.9 mg
			G: 0.004 °/s
		Noise Density	A: 175 $\mu g/\sqrt{Hz}$
			G: 0.014 $°/s/\sqrt{Hz}$
		Bandwidth	532Hz
GPS	uBlox ZED-F9P	Horizontal Positional Accuracy	1.0m
		Vertical Positional Accuracy	1.5m
		Max Update Rate	5 Hz

Table 3: Key Components and Their Performance

5.2 Software

The LAUKF was implemented in Rust on the Atmel SAMV71 Micro-controller Unit (MCU). Given the relatively immature state of embedded Rust, particularly regarding support for the chosen MCU, it was decided not to use any preexisting Hardware Abstraction Layers (HAL). Instead, the MCU's registers and peripherals were programmed directly using a Peripheral Access Crate (PAC). The PAC was automatically generated by the open-source tool `svd2rust`, which provided a straightforward workflow with minimal build and tooling challenges, making it easy to get started.

However, implementing MCU peripheral drivers in Rust presented some challenges. Embedded programming inherently involves working with hardware registers, which are global, mutable singletons. This paradigm conflicts with Rust's strong emphasis on memory safety and its strict ownership model. Consequently, expressing these concepts ergonomically within Rust required careful considerations. In the end it was found that using a delayed initialization of Rust `struct`'s provided a reasonable balance between ergonomic code and memory safety.

```
pub struct Mck {
    fclk: NonNull<Fclk>,
    div: Div,
}

impl Mck {
    // WARNING: Dangling pointer.
    pub(crate) const fn uninit() -> Self {
        Self {
            fclk: NonNull::dangling(),
            div: Div::Div1,
        }
    }

    // Only called during Hal::init().
    pub(crate) fn default(&mut self, fclk: &Fclk) {
        self.fclk = NonNull::from(fclk);
    }
}
```

Listing 1: Code Snippet from MCU Main Clock Driver

> Certain peripherals on the MCU require variables with `'static` lifetimes. While this is achievable within the RTIC framework, it is restricted to `const` expressions, and the use of generics in such static variables is also constrained.
>
> In the provided code snippet, the main clock of the MCU needs a reference to the free-running clock in order to determine its own clock speed (see SAMV71 data-sheet section 31.3). To address this requirement, the delayed initialization method demonstrated proved to be an effective solution. This approach only requires the programmer to ensure that the variable is properly initialized after its declaration.

In a more conventional embedded Rust approach, resource management was handled using a token system. This method provided a convenient way to decouple individual resources, such as pins or memory channels, from their parent peripherals.

If a tokenized resource required a reference back to its parent peripheral—for instance, a Pulse Width Modulation (PWM) channel needing knowledge of the PWM peripheral's clock speed—this could be implemented using the delayed initialization technique described earlier.

```
pub struct Token<T> {
    inner: NonNull<T>,
}

impl<T> From<&T> for Token<T> {
    fn from(reference: &T) -> Self {
        Self {
            inner: NonNull::from(reference),
        }
    }
}

impl<T> Token<T> {
    pub fn consume(mut self) -> &'static mut T {
        unsafe { self.inner.as_mut() }
    }
}
```

Listing 2: Code Snippet from Peripheral Token

Certain peripherals contain resources that can be distributed to different tasks. For example the PIO peripheral (see SAMV71 data-sheet section 32) has different input/output pins that can be used by user code. In this case it is convenient for the PIO driver to hold `Option<Token<Pin>>` as a map of the resources it currently holds. User code can then `.take()` the token to prevent other tasks from inadvertently accessing that resource. The user code can `.consume()` the token to yield a mutable reference to the underlying resource.

The UKF code was implemented in the `RTIC2` framework. Similarly to the decision to program directly at the PAC level instead of using a HAL, the choice to use `RTIC2` stemmed from a desire to employ a minimalist framework rather than a full-fledged Real-Time Operating System (RTOS).

`RTIC2` is designed for embedded systems and offers a lightweight, hardware-focused task scheduling approach. It operates purely on hardware interrupts, allowing tasks to be scheduled and preempted based on interrupt priorities configured in the MCU registers. Unlike traditional RTOSs, which often rely on software-based schedulers with threads or processes, `RTIC2` directly maps tasks to interrupts, ensuring highly deterministic behavior with minimal overhead.

One of the key features of `RTIC2` is its ability to manage shared resources between tasks safely. It achieves this by wrapping shared resources in `mutex` abstractions. These mutexes ensure memory safety and prevent data races, leveraging Rust's ownership model and compile-time guarantees.

This combination of interrupt-based scheduling and safe resource sharing makes `RTIC2` particularly well-suited for resource-constrained embedded environments, where efficiency and reliability are paramount.

The Direct Memory Access (DMA) peripheral of the MCU was also used in the implementation of the UKF code. DMA is a hardware feature that enables data transfers between memory and peripherals, or between memory regions, without involving the CPU. This allows for efficient handling of high-throughput or repetitive data transfers, freeing the CPU to focus on more computationally intensive tasks, such as executing the UKF algorithm.

In this implementation, DMA was utilized to offload tasks like reading sensor data from peripheral interfaces or writing computed results to the output peripherals. By setting up DMA channels, the transfers were managed automatically, with minimal CPU intervention. Figure 7 show a simplified interaction between a selection of software tasks and the DMA within the `RTIC2` framework.

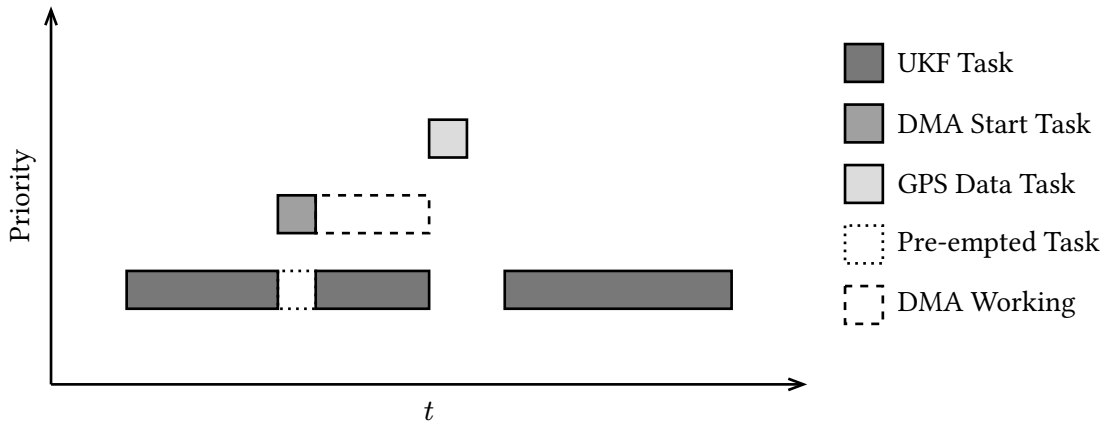

Figure 7: Representative Code Flow

Simplified representation of code flow within RTIC framework. The UKF Task is started by a timer driven interrupt. The DMA Start Task is triggered when the MCU senses a rising edge on the GPS_TX_READY line (see Appendix A). The GPS Data Task is triggered by an interrupt generated by the DMA on completion of data read. In this scenario, a GPS data packet is ready to be received partway through execution of the UKF Task. This pre-empts the UKF Task, starting a DMA transfer before yielding back to the UKF Task. The DMA transfer completes in the background, triggering a data processing step, resulting in a new GPS measurement ready for the next execution of the UKF task.

At the conclusion of this thesis work, the chosen implementation of Rust code on the embedded device was found to be effective and straightforward to work with. The decision to use a thinner abstraction layer, avoiding typical HALs, was viewed positively. This approach allowed the code to more closely reflect the behavior of the peripherals as described in the SAMV71 data-sheet, increasing the transparency of the implementation.

The use of the `RTIC2` framework was also well-regarded. Its minimalistic design avoided reliance on bloated embedded development suites, enabling the use of lightweight, open-source tools and a straightforward build chain. This simplicity streamlined the development process and reduced the overhead of mainlining build systems experienced in past `C++` projects.

However, this approach came with trade-offs. While `RTIC2` provides a relatively simple set of core functionalities, the interactions between these features can lead to a highly complex system. This complexity requires careful design and thorough understanding of task interactions to ensure predictable behavior.

6 Results

This section presents the results of the LAUKF developed in this thesis. To evaluate its performance, a quadrotor simulation was implemented using the Rust programming language, which served as a virtual test bed for the LAUKF before hardware integration was possible. The simulation environment was designed to mimic realistic dynamics and sensor noise, providing a controlled and repeatable platform for analysis.

Two simulated test cases are presented to demonstrate the LAUKF's capabilities under different conditions. In each case the performance of the filter is qualitatively evaluated and potential strengths and weaknesses highlighted, offering insights into its practical applicability for real-world quadrotor control and navigation tasks.

6.1 Simulations

The simulation was implemented with dual quaternion dynamics to propagate the quadrotor's state. This allowed validation of the process model used in the LAUKF independently of the filter implementation. The simulation parameters were selected to replicate realistic quadrotor behavior. The physical properties are based on measurements taken from an actual vehicle. By using these realistic parameters, the simulation environment served as a reliable analog for the real system.

To control the quadrotor in the simulated environment, the PD+ controllers were heuristically tuned to achieve adequate performance. Although not optimal, the controller provided sufficient to excite the system when evaluating the LAUKF.

The measurement properties were based on the manufacturer data-sheets of the sensors integrated into the flight computer discussed in the previous chapter. Incorporating these specifications into the simulation ensured that the sensor data closely mirrored what would be encountered in real-world operations.

The simulations were conducted at fixed time-step of $\Delta t = 0.01\text{s}$ with the assumption that all sensor data was available except for the GPS. The GPS measurement were incorporated at a reduced rate of $T_r = 5\text{Hz}$ to emulate the multi-rate behavior of the real system.

Physical Properties	**Controller Properties**	**Measurement Properties**
$m = 1.5259\text{kg}$	$k_{\text{e}} = 50.0$	$N_{\omega} = 0.01\text{rad/s}$
$J = \text{diag}\left(\begin{bmatrix} 0.002473\text{kg}\cdot\text{m}^2 \\ 0.002685\text{kg}\cdot\text{m}^2 \\ 0.004403\text{kg}\cdot\text{m}^2 \end{bmatrix}\right)$	$k_{\omega} = 2.0$	$N_{\beta_{\omega}} = 0.01\text{rad/s}$
$c_T = 1.5E^{-6}$	$k_p = 1.0$	$N_a = 0.000015\text{m/s}^2$
$c_M = 1.9E^{-8}$	$k_d = 2.0$	$N_b = 0.0041\text{G}$
$d_1 = d_2 = d_3 = d_4 = 0.113\text{m}$	$k_1 = 0.33$	$N_r = 0.5\text{m}$
$T_m = 10.0\text{s}$	$k_2 = 1.0$	

Table 4: Simulation Parameters

Figure 8 illustrates the tracking performance of the LAUKF during a target approach maneuver. The quadrotor's motion in this scenario is smooth, allowing assessment of the filter's capabilities under nominal conditions. Notably, the LAUKF demonstrates good tracking performance in attitude estimation, with minimal deviation from the true attitude.

A degradation in position tracking is observed along the z-axis as the quadrotor nears the target location. This decline is likely due to an over estimation of the vehicles vertical velocity which becomes more pronounced in the final stages of the maneuver. The tension between GPS position updates and the dead reckoning of velocity and acceleration is evident. Given the GPS measurement accuracy of ± 1m, the LAUKF covariance placed on the measurements it large. This prevents the GPS update from correcting the state estimate fully. If the system was excited again, the track performance would resume.

The simulation duration could not be extended beyond five seconds due to the onset of chatter in the guidance controller. As shown in Figure 6, this behavior was tolerable when evaluating the controller independently. However, with the integration of the LAUKF, the severe chatter introduced numerical instability, causing the state covariance matrix to become singular. This highlights a sensitivity of the filter to erratic input conditions.

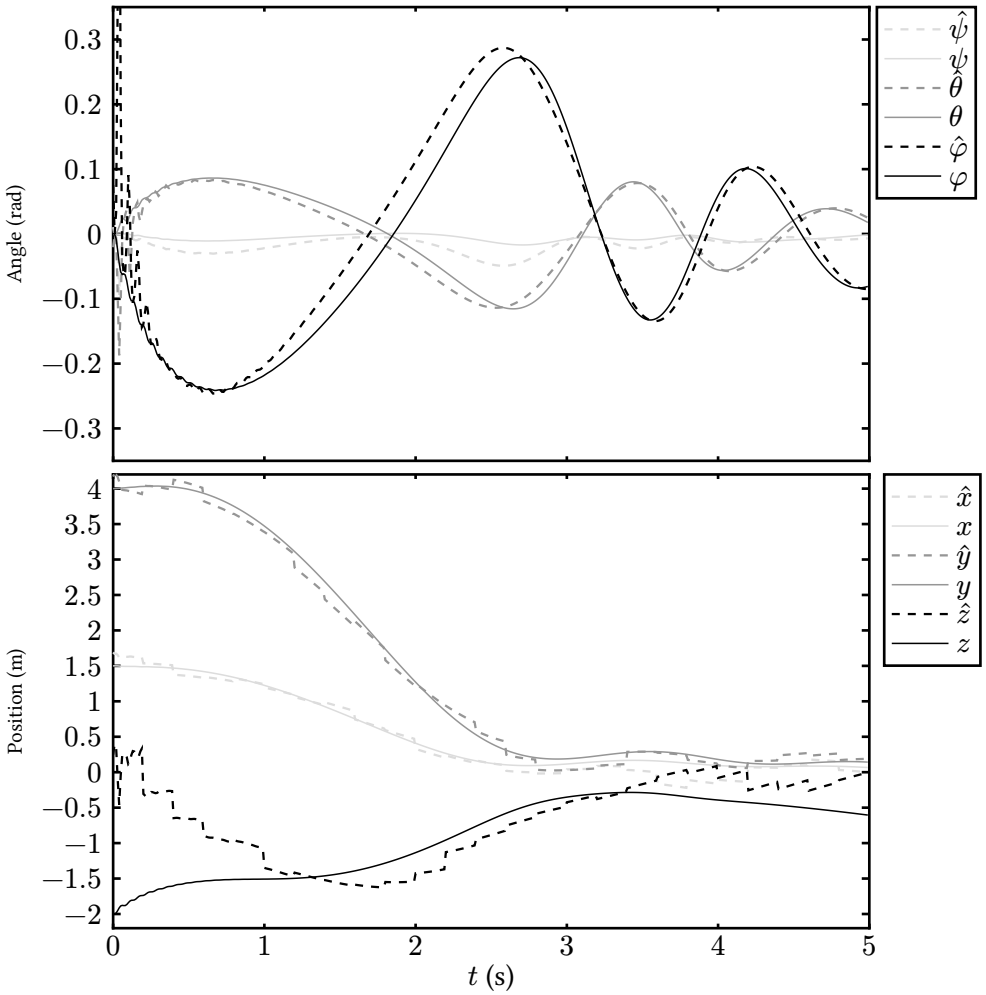

Figure 8: LAUKF Tracking During Simulated Target Approach

Simulation of a quadrotor approaching the target position $r_d = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ from initial conditions $r_o = \begin{bmatrix} 4 & 1.5 & 2 \end{bmatrix}^\top$

Figure 9 illustrates the tracking performance of the LAUKF during a disturbance rejection scenario. Upon the introduction of the disturbance, an immediate degradation in attitude estimation is observed. This occurs because the acceleration vector measured by the accelerometer becomes corrupted by the disturbance, leading to a misinterpretation of the disturbance-induced acceleration as a component of gravity.

As the controller excites the system to compensate for the disturbance, the LAUKF gradually resumes accurate tracking. This demonstrates the filter's capability to recover from transient errors caused by external perturbations, provided the system dynamics remain within a manageable range for the filter's design.Another notable behavior is the overestimation of attitude changes caused by the high angular rates commanded by the controller. This effect is particularly evident in regions where the true attitude of the vehicle exhibits high-frequency variations.

Similar to the observations in Figure 8, a degradation in position tracking is also observed when the quadrotor cancels out most of its translational velocity. However, in the disturbance rejection case, the prior flight was much more erratic. This caused the degradation to be exacerbated as the velocity had been severely over estimated due to the additional acceleration caused by the disturbance.

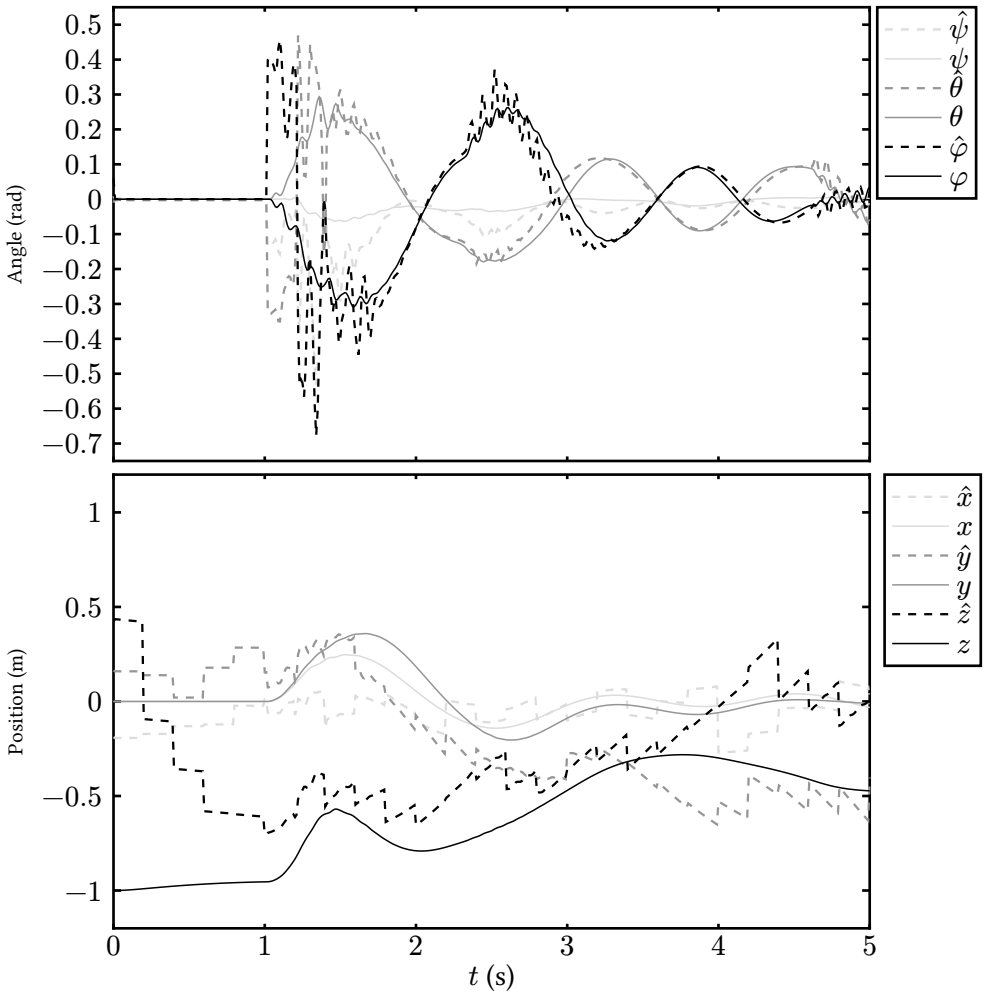

Figure 9: LAUKF Tracking During Simulated Disturbance Rejection

Simulation of a quadrotor attempting to maintain position after the introduction of a disturbance force $f_d = [10 \;\; 10 \;\; 10]^\top$ from $t = 1$s to $t = 1.2$s

6.2 Practical Test

The LAUKF was also tested on the flight computer described in the previous chapter. The flight computer was maneuvered in an OptiTrack setup to serve as a ground truth to compare the LAUKF results against. Due to *severe* time constraints, it was not possible to calibrate the sensors on the flight computer or the OptiTrack system before conducting the test. Consequently, the OptiTrack recordings for roll and yaw angles were rendered unusable, leaving only the pitch measurements as viable for comparison. The pitch measurements obtained from the OptiTrack system matched the predictions made by the LAUKF, providing partial validation of the filter's performance.

Furthermore, the test environment presented additional challenges. The physical location of the OptiTrack system used was such that the flight computer received an extremely poor GPS signal, which significantly limited the reliability of the GPS data for comparison against the OptiTrack position measurements. Despite these limitations, the results for the pitch motion and the y-axis position are presented here to demonstrate the implementation of the LAUKF in hardware, albeit under constrained and sub-optimal conditions. These findings serve as a proof of concept for the LAUKF's deployment on physical hardware.

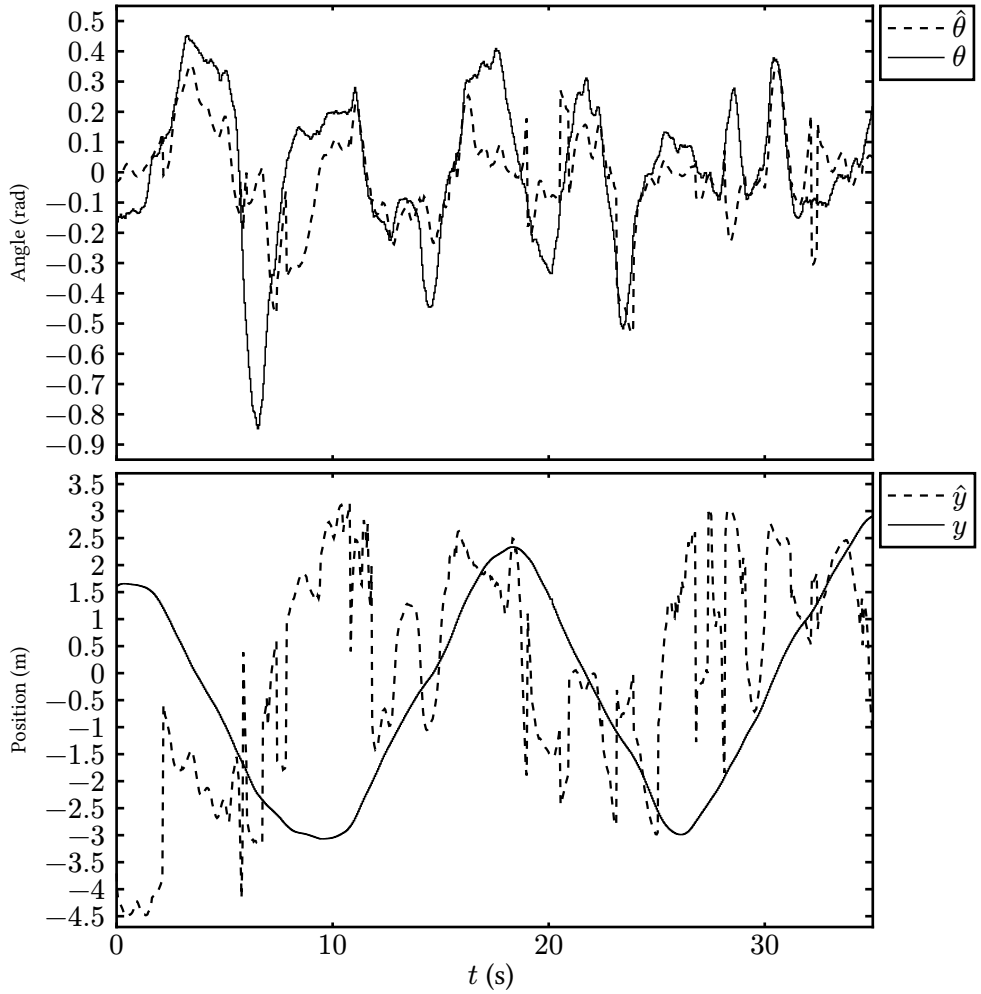

Figure 10: Practical LAUKF Tracking Compared Against OptiTrack Ground Truth

7 Conclusion

This thesis presents an approach for rigid body pose estimation utilizing dual quaternions, offering a uniform mathematical framework for handling the six degrees of freedom in rigid body motion. The thesis primarily focuses on the development of a dual quaternion model of quadrotor dynamics and its integration with an Unscented Kalman Filter for state estimation. By leveraging Lie theory, the traditional UKF framework was extended to handle the unit dual quaternion manifold, ensuring accurate state covariance representation. The proposed Lie Algebraic Unscented Kalman Filter was partially validated through simulations and practical testing on a purpose built flight computer.

The simulation results demonstrate that the LAUKF is capable of estimating the quadrotor's attitude, despite some degradation in position tracking during high dynamic conditions. The simulations also highlight limitations in the guidance controller, which introduced instability due to chatter when tracking was performed under certain conditions, especially the vehicle approached the target position from above or below.

The integration of the LAUKF with a physical flight computer demonstrated the feasibility of applying this approach to real-world systems. While the practical testing was constrained by time and calibration issues, the partial validation of measurements with the OptiTrack system serves as a proof of concept for the deployment of the filter on physical hardware. The challenges faced in the testing environment, further emphasize the need for precise sensor calibration and highlight the potential for further optimization of the LAUKF technique.

Ultimately, this thesis explores the applications of dual quaternions in the feild of pose estimation, particularly within the context of quadrotor systems. The approach developed here offers a promising solution for real-time state estimation in dynamic environments, with potential applications in a variety of robotic and aerial systems. Future work should focus on refining the simulation environment of the LAUKF, as well as addressing the challenges related to implementation on physical hardware, in order to more completely assess the performance of the LAUKF. Future works are also encouraged to perform performance comparisons to other pose estimation frameworks.

Bibliography

[1] M. Idrissi, M. Salami, and F. Annaz, "A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, p. 22, 2022, doi: 10.1007/s10846-021-01527-7.

[2] A. Asignacion and S. Satoshi, "Historical and Current Landscapes of Autonomous Quadrotor Control: An Early-Career Researchers' Guide," *Drones*, vol. 8, no. 3, p. 72, 2024, doi: 10.3390/drones8030072.

[3] D. De Mori, "Flying Qualities of Multi-rotor UAVs: market analysis and parameter definition," 2021.

[4] B. SCHLETZ, "Use of quaternions in Shuttle guidance, navigation, and control," *Guidance and Control Conference*, 1982, doi: 10.2514/6.1982-1557.

[5] O.-E. Fjellstad and T. Fossen, "Position and attitude tracking of AUV's: a quaternion feedback approach," *IEEE Journal of Oceanic Engineering*, no. 4, pp. 512–518, 1994, doi: 10.1109/48.338387.

[6] B. Razgus, E. Mooij, and D. Choukroun, "Relative Navigation in Asteroid Missions Using Dual Quaternion Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 1–16, 2017, doi: 10.2514/1.g002805.

[7] R. Kinzie, R. Bevilacqua, D. Seo, J. W. Conklin, and P. J. Wass, "Dual quaternion-based dynamics and control for gravity recovery missions," *Acta Astronautica*, vol. 213, pp. 764–776, 2023, doi: 10.1016/j.actaastro.2023.10.007.

[8] J. Arrizabalaga and M. Ryll, "Towards Time-Optimal Tunnel-Following for Quadrotors," *2022 International Conference on Robotics and Automation (ICRA)*, vol. 0, pp. 4044–4050, 2022, doi: 10.1109/icra46639.2022.9811764.

[9] J. Arrizabalaga and M. Ryll, "Pose-Following with Dual Quaternions," *2023 62nd IEEE Conference on Decision and Control (CDC)*, vol. 0, pp. 5959–5966, 2023, doi: 10.1109/cdc49753.2023.10383688.

[10] T. A. Johansen, T. S. Andersen, and R. Kristiansen, "PD+ Based Trajectory Tracking of the Underactuated Quadrotor Platform using Dual Quaternions," *2019 American Control Conference (ACC)*, 2019, doi: 10.23919/acc.2019.8814924.

[11] M. F. Coelho, K. Bousson, and K. Ahmed, "Survey of nonlinear state estimation in aerospace systems with Gaussian p riors," *Advances in Aircraft and Spacecraft Science*, vol. 7, no. 6, pp. 495–516, 2020, doi: 10.12989/aas.2020.7.6.495.

[12] P. Bernal-Polo and H. Martínez-Barberá, "Kalman Filtering for Attitude Estimation with Quaternions and Concepts from Manifold Theory," *Sensors*, vol. 19, no. 1, p. 149, 2019, doi: 10.3390/s19010149.

[13] N. Filipe and P. Tsiotras, "Simultaneous position and attitude control without linear and angular velocity feedback using dual quaternions," *2013 American Control Conference*, pp. 4808–4813, 2013, doi: 10.1109/acc.2013.6580582.

[14] H. Abaunza, J. Cariño, P. Castillo, and R. Lozano, "Quadrotor Dual Quaternion Control," *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pp. 195–203, 2015, doi: 10.1109/red-uas.2015.7441007.

[15] X. Wang and C. Yu, "Feedback Linearization Regulator with Coupled Attitude and Translation Dynamics Based on Unit Dual Quaternion," *2010 IEEE International Symposium on Intelligent Control*, pp. 2380–2384, 2010, doi: 10.1109/isic.2010.5612894.

[16] F. L. Markley, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, 2012, doi: 10.2514/2.5048.

[17] B. C. Hall, "Lie Groups, Lie Algebras, and Representations, An Elementary Introduction," *Graduate Texts in Mathematics*, 2015, doi: 10.1007/978-3-319-13467-3.

[18] B. V. Adorno, "Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra - Part I: Fundamentals," *HAL*, 2017.

[19] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Wiley, 2016.

[20] A. P. French and M. G. Ebison, *Introduction to CLASSICAL MECHANICS*. 1986. doi: 10.1007/978-94-009-4119-9_9.

[21] N. K. *et al.*, "Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion," *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II*, p. 99126, 2016, doi: 10.1117/12.2234255.

[22] J. M. Barrett, M. A. Gennert, W. R. Michalson, and J. L. Center, "Analyzing and Modeling an IMU for Use in a Low-Cost Combined Vision and Inertial Navigation System," *2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 19–24, 2012, doi: 10.1109/tepra.2012.6215648.

[23] N. Trawny and S. I. Roumeliotis, *Indirect Kalman Filter for 3 D Attitude Estimation*. 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:14569549

[24] J. Metge, R. Mégret, A. Giremus, Y. Berthoumieu, and T. Décamps, "Calibration of an inertial-magnetic measurement unit without external equipment, in the presence of dynamic magnetic disturbances," *Measurement Science and Technology*, vol. 25, no. 12, p. 125106, 2014, doi: 10.1088/0957-0233/25/12/125106.

[25] H. Lee, C. Lee, H. Jeon, J. J. Son, Y. Son, and S. Han, "Interference-Compensating Magnetometer Calibration With Estimated Measurement Noise Covariance for Application to Small-Sized UAVs," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 10, pp. 8829–8840, 2019, doi: 10.1109/tie.2019.2950841.

[26] M. Silic and K. Mohseni, "Correcting Current-Induced Magnetometer Errors on UAVs: An Online Model-Based Approach," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 1067–1076, 2020, doi: 10.1109/jsen.2019.2945201.

[27] A. C. B. Chiella, B. O. S. Teixeira, and G. A. S. Pereira, "Quaternion-Based Robust Attitude Estimation Using an Adaptive Unscented Kalman Filter," *Sensors*, vol. 19, no. 10, p. 2372, 2019, doi: 10.3390/s19102372.

[28] A. W. R. Soundy, B. J. Panckhurst, P. Brown, A. Martin, T. C. A. Molteno, and D. Schumayer, "Comparison of Enhanced Noise Model Performance Based on Analysis of Civilian GPS Data," *Sensors*, vol. 20, no. 21, p. 6050, 2020, doi: 10.3390/s20216050.

[29] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007. [Online]. Available: https://books.google.nl/books?id=6Ne76uYOlVwC

[30] S. Haykin, "Kalman Filtering and Neural Networks," 2023, doi: 10.1002/0471221546.

[31] B. J. Sipos, "Application of the Manifold-Constrained Unscented Kalman Filter," *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 30–43, 2008, doi: 10.1109/plans.2008.4569967.

[32] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, pp. 1628–1632, 1995, doi: 10.1109/acc.1995.529783.

[33] K. Nielsen, C. Svahn, H. Rodriguez-Deniz, and G. Hendeby, "UKF Parameter Tuning for Local Variation Smoothing," *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, vol. 0, pp. 1–8, 2021, doi: 10.1109/mfi52462.2021.9591188.

[34] E. Leffert, F. Markley, and M. Shuster, "Kalman filtering for spacecraft attitude estimation," *20th Aerospace Sciences Meeting*, 1982, doi: 10.2514/6.1982-70.

[35] M. E. Pittelkau, "An Analysis of the Quaternion Attitude Determination Filter," *The Journal of the Astronautical Sciences*, vol. 51, no. 1, pp. 103–120, 2003, doi: 10.1007/bf03546317.

[36] A. Carmi and Y. Oshman, "On the Covariance Singularity of Quaternion Estimators," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, doi: 10.2514/6.2007-6814.

[37] A. M. Sjøberg and O. Egeland, "Lie Algebraic Unscented Kalman Filter for Pose Estimation," *arXiv*, 2020, doi: 10.48550/arxiv.2005.00385.

[38] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging Quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1193–1197, 2012, doi: 10.2514/1.28949.

[39] K. Li, F. Pfaff, and U. D. Hanebeck, "Unscented Dual Quaternion Particle Filter for SE(3) Estimation," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 647–652, 2021, doi: 10.1109/lcsys.2020.3005066.

[40] R. G. Valenti, I. Dryanovski, and J. Xiao, "A Linear Kalman Filter for MARG Orientation Estimation Using the Algebraic Quaternion Algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 467–481, 2015, doi: 10.1109/tim.2015.2498998.

[41] L. Jin, Y. Lou, L.-A. Chen, and Q. Lu, "The Unified Tracking Controller for a Tilt-Rotor Unmanned Aerial Vehicle Based on the Dual Quaternion," *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 1356–1363, 2022, doi: 10.1109/icus55513.2022.9986880.

[42] H. T. Kussaba, L. F. Figueredo, J. Y. Ishihara, and B. V. Adorno, "Hybrid kinematic control for rigid body pose stabilization using dual quaternions," *Journal of the Franklin Institute*, vol. 354, no. 7, pp. 2769–2787, 2017, doi: 10.1016/j.jfranklin.2017.01.028.

[43] T. S. Andersen and R. Kristiansen, "Quaternion Guidance and Control of Quadrotor," *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1567–1601, 2017, doi: 10.1109/icuas.2017.7991509.

Appendix A - Electrical Design

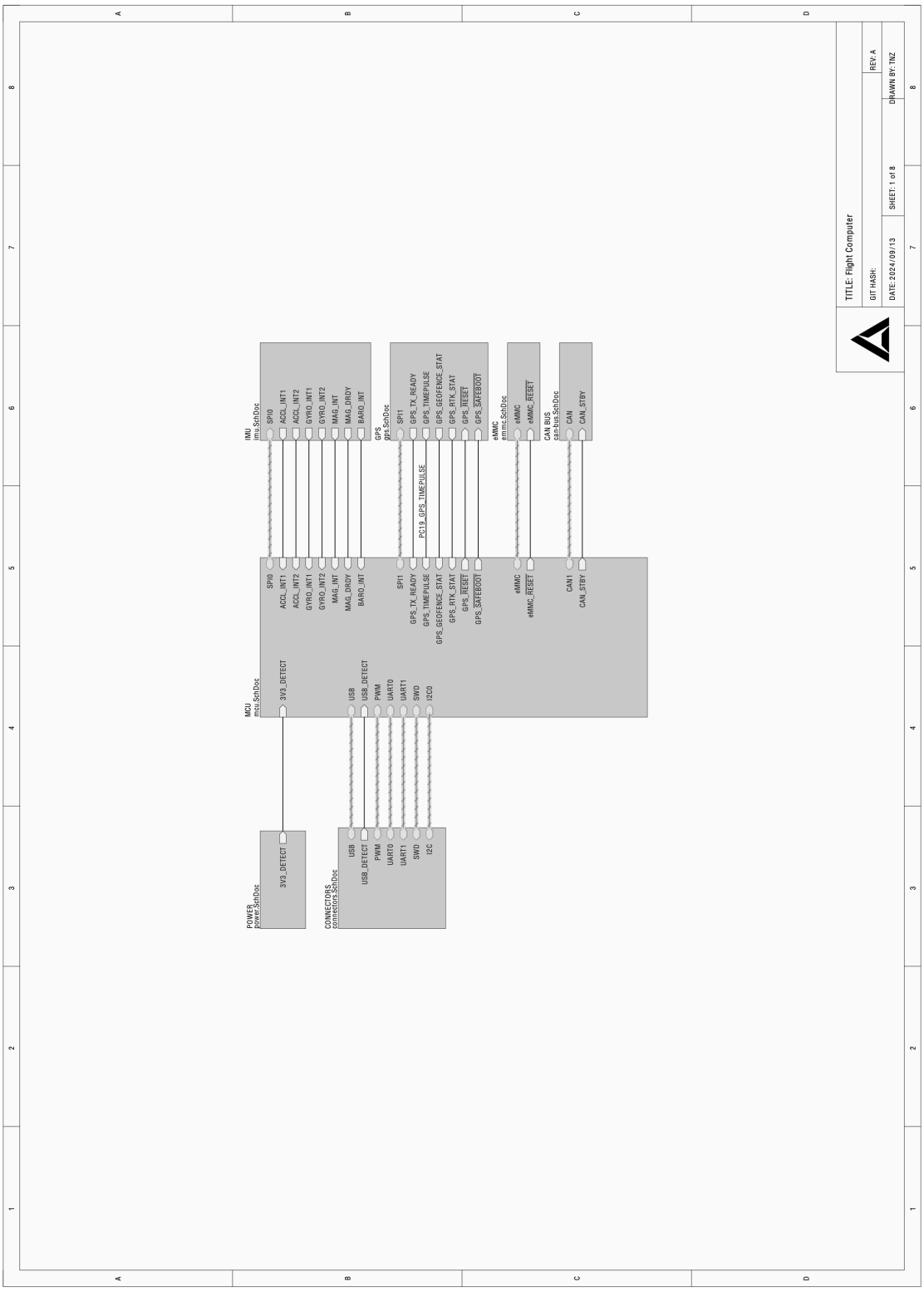

Figure A1: Top Level Electrical Schematic

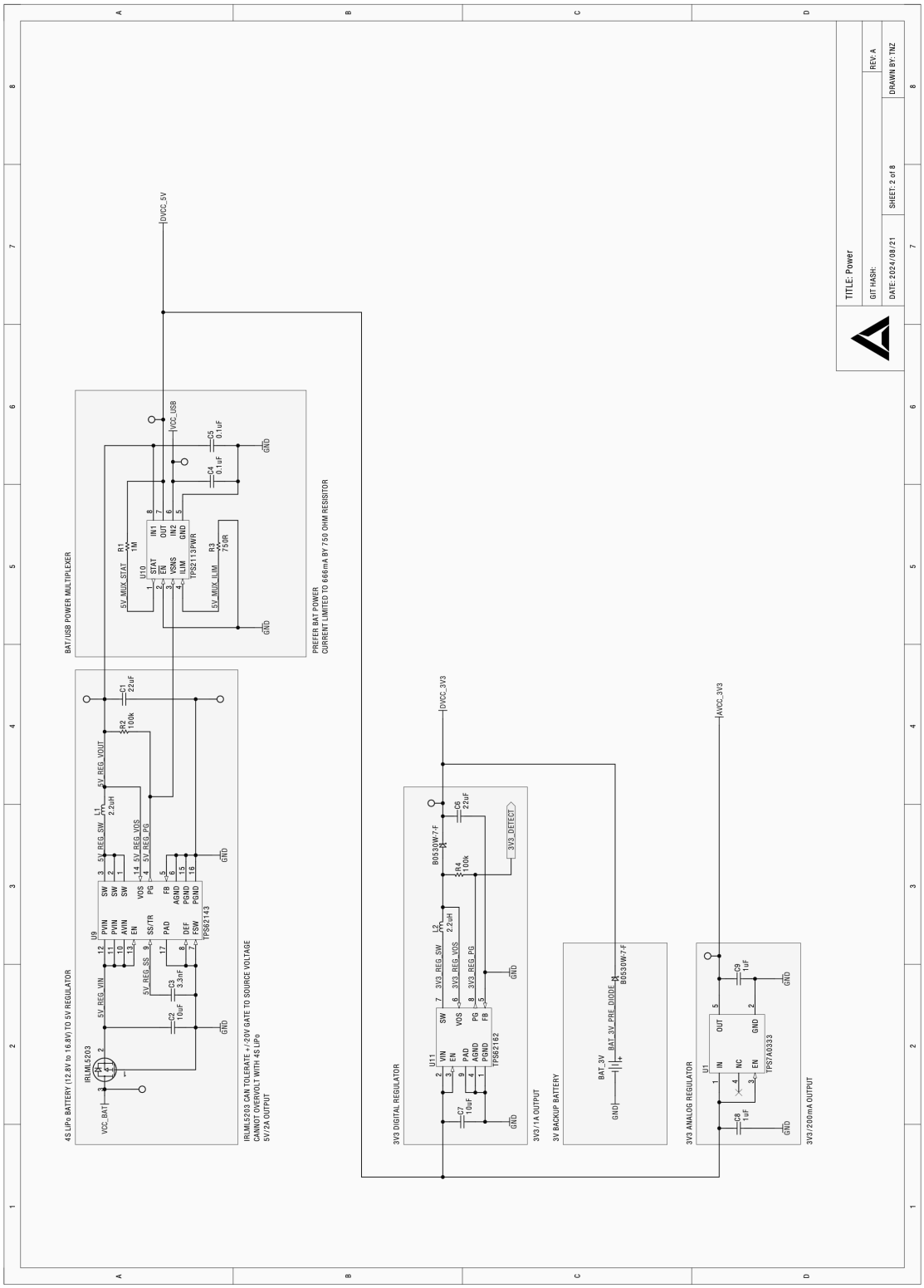

Figure A2: Power Subsystem Electrical Schematic

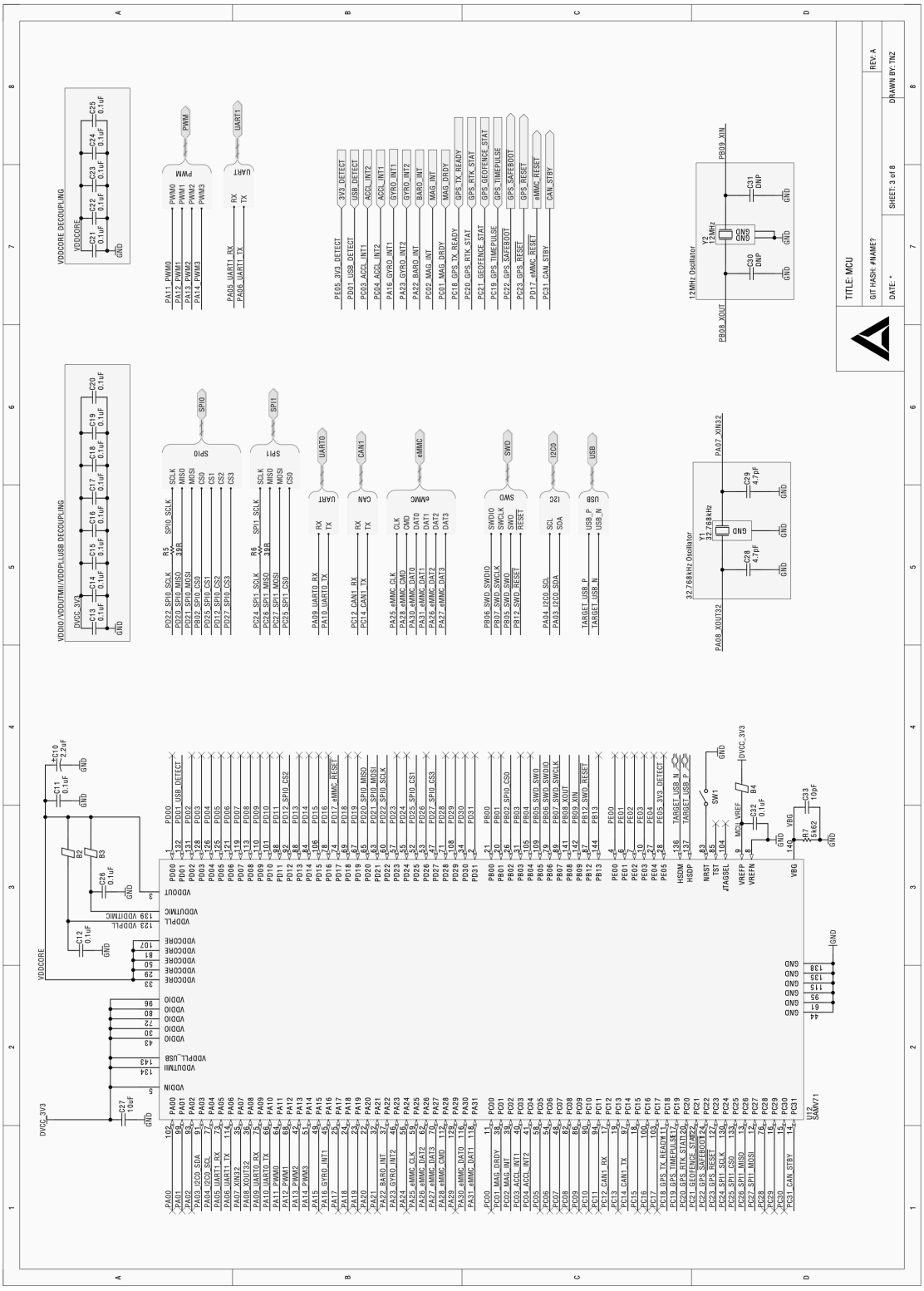

Figure A3: MCU Subsystem Electrical Schematic

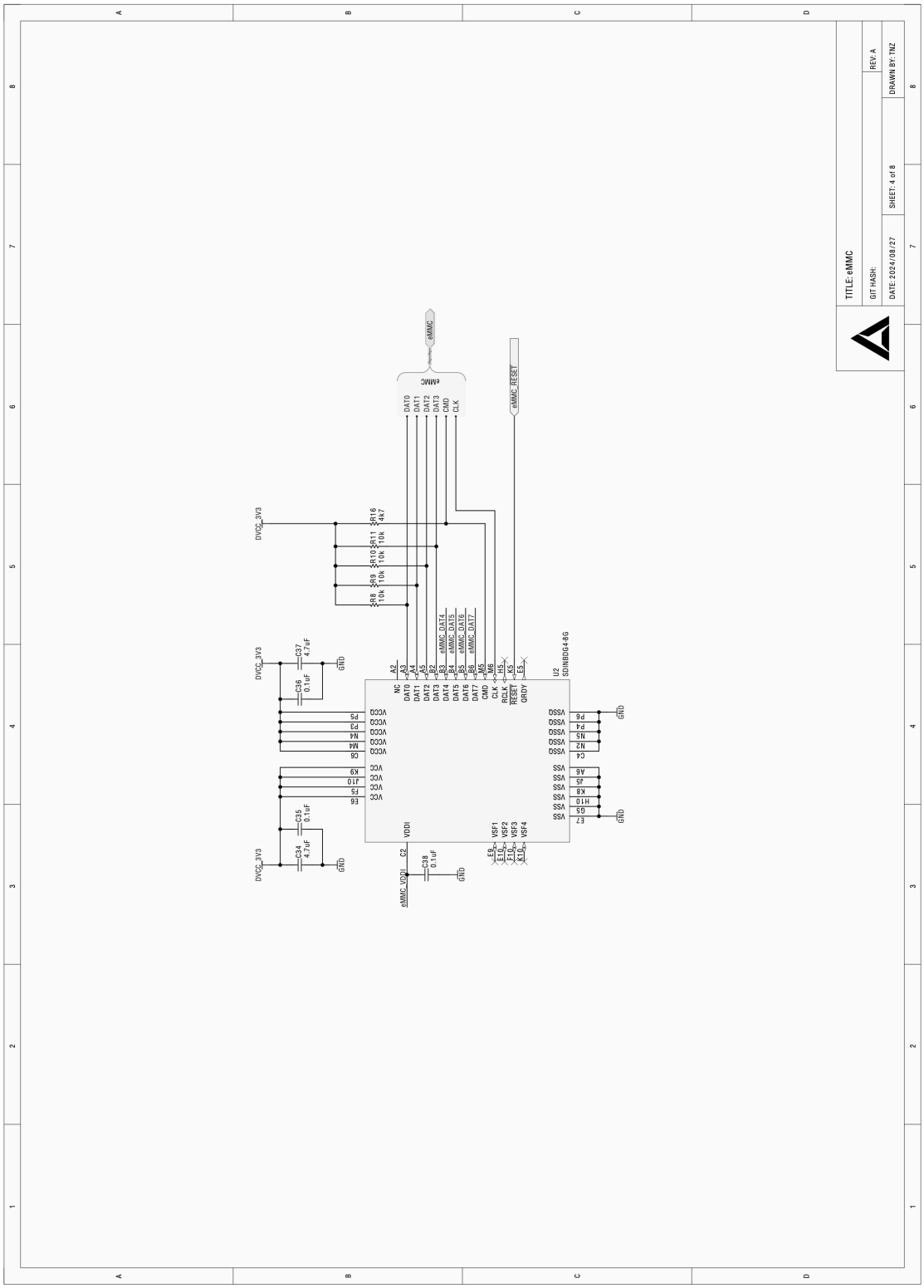

Figure A4: eMMC Electrical Schematic

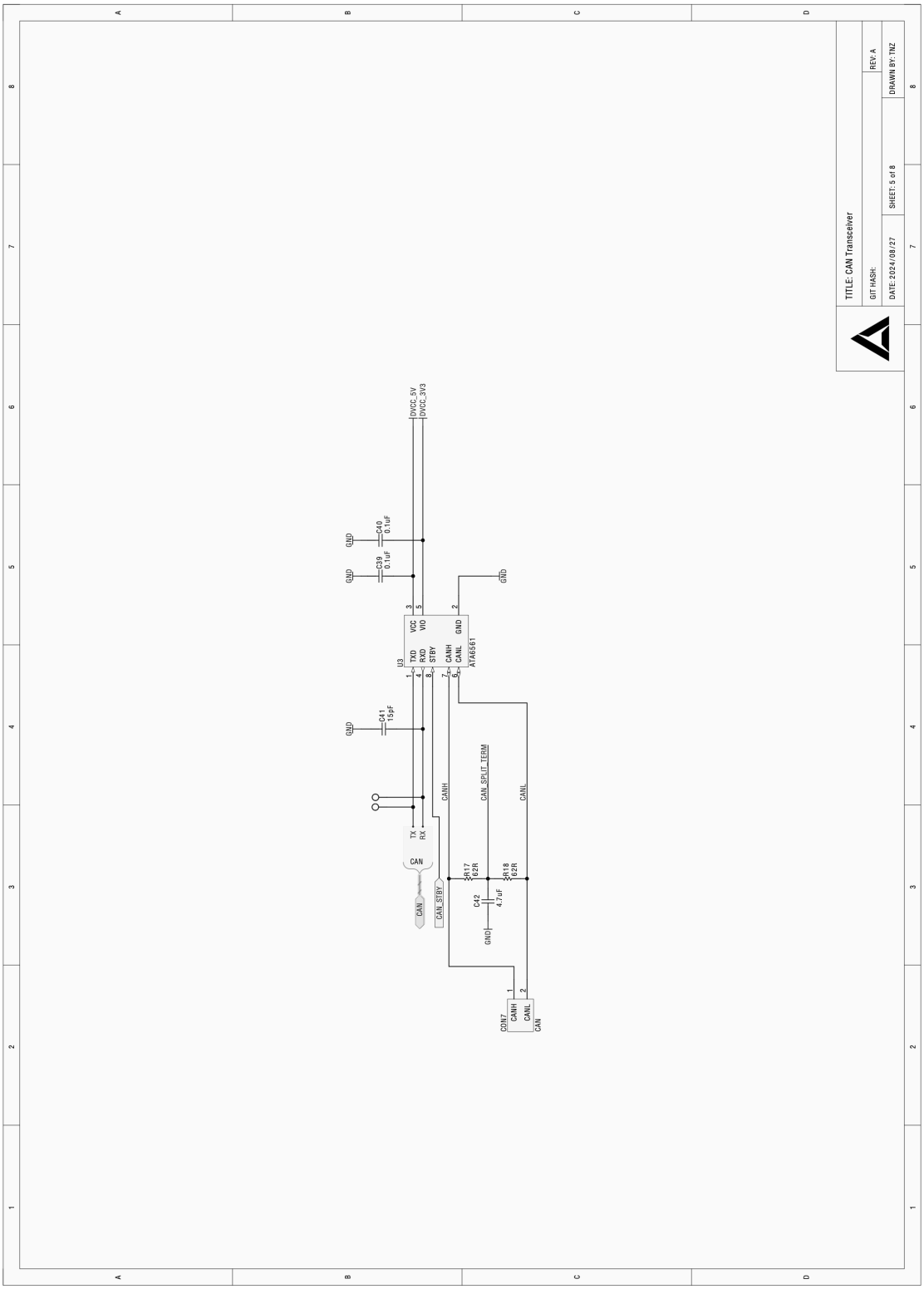

Figure A5: CAN Bus Electrical Schematic

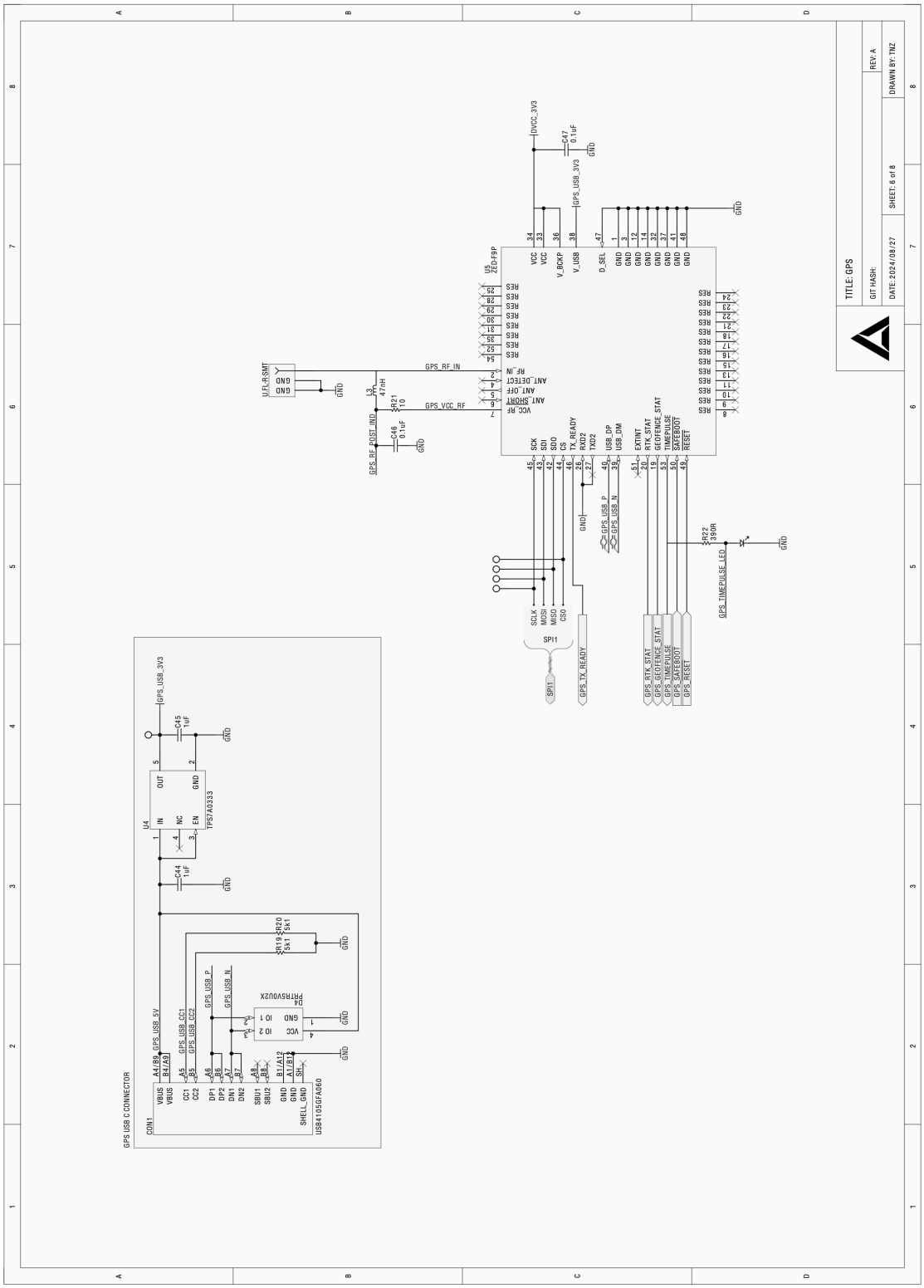

Figure A6: GPS Electrical Schematic

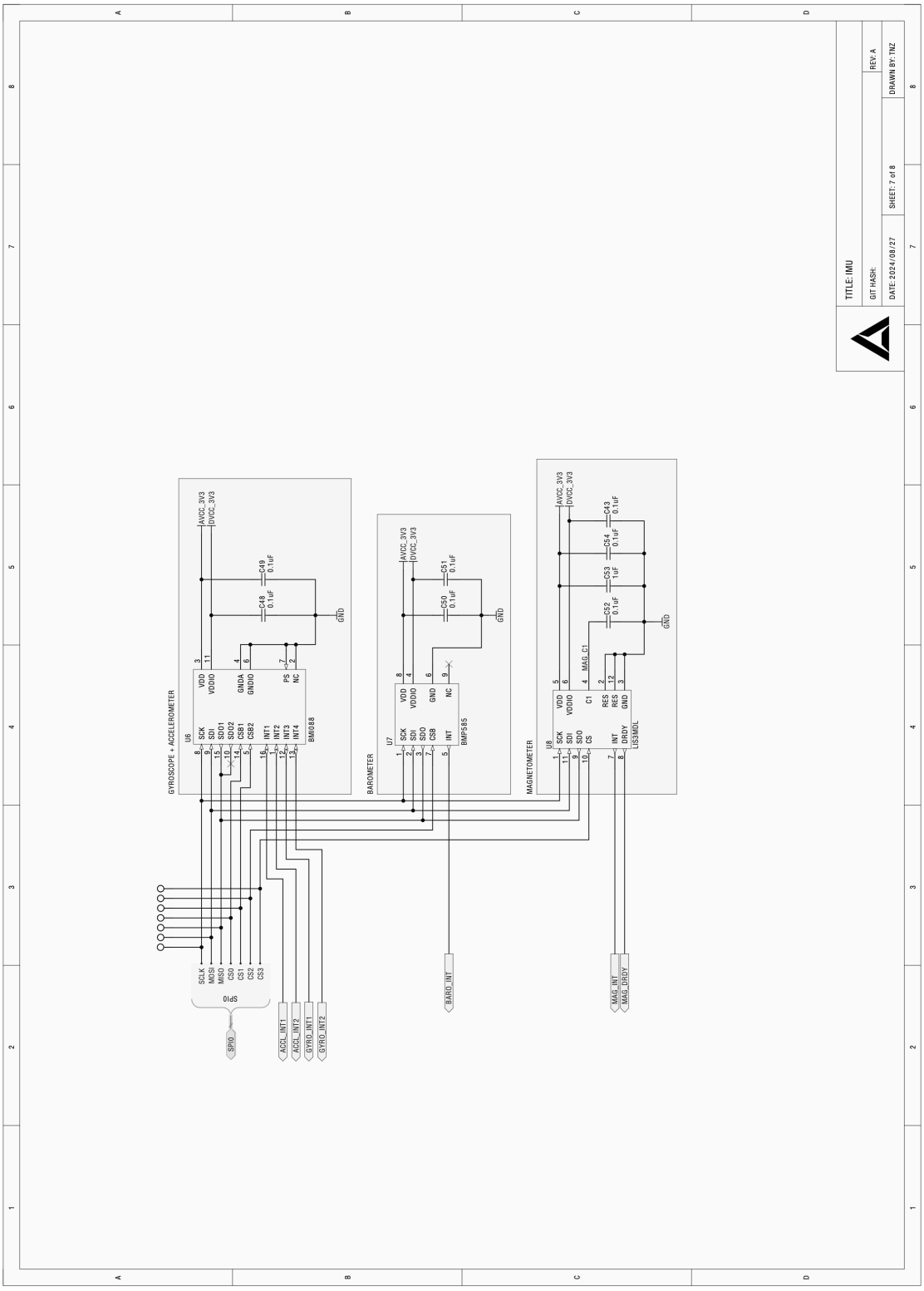

Figure A7: IMU Electrical Schematic

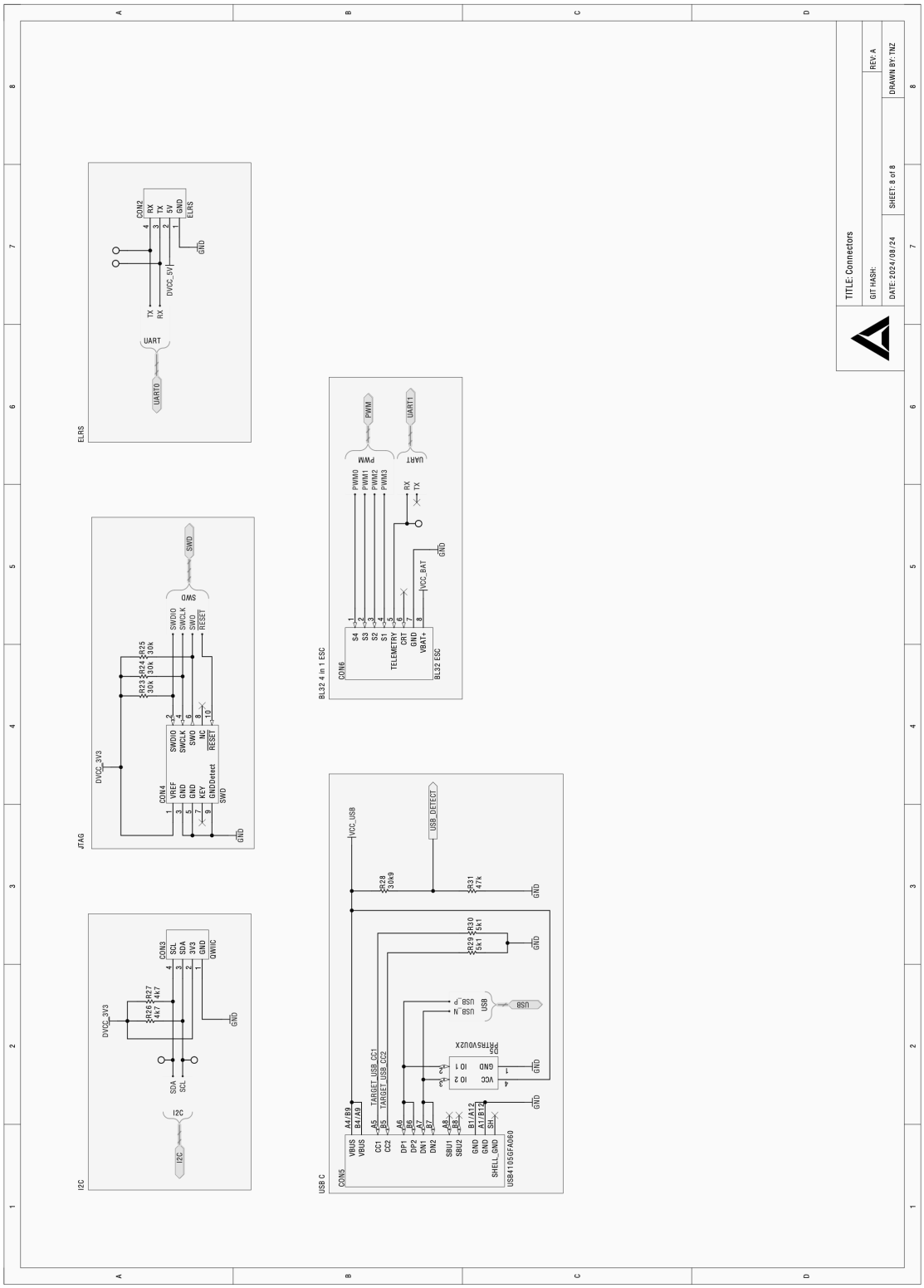

Figure A8: External Connections Electrical Schematic

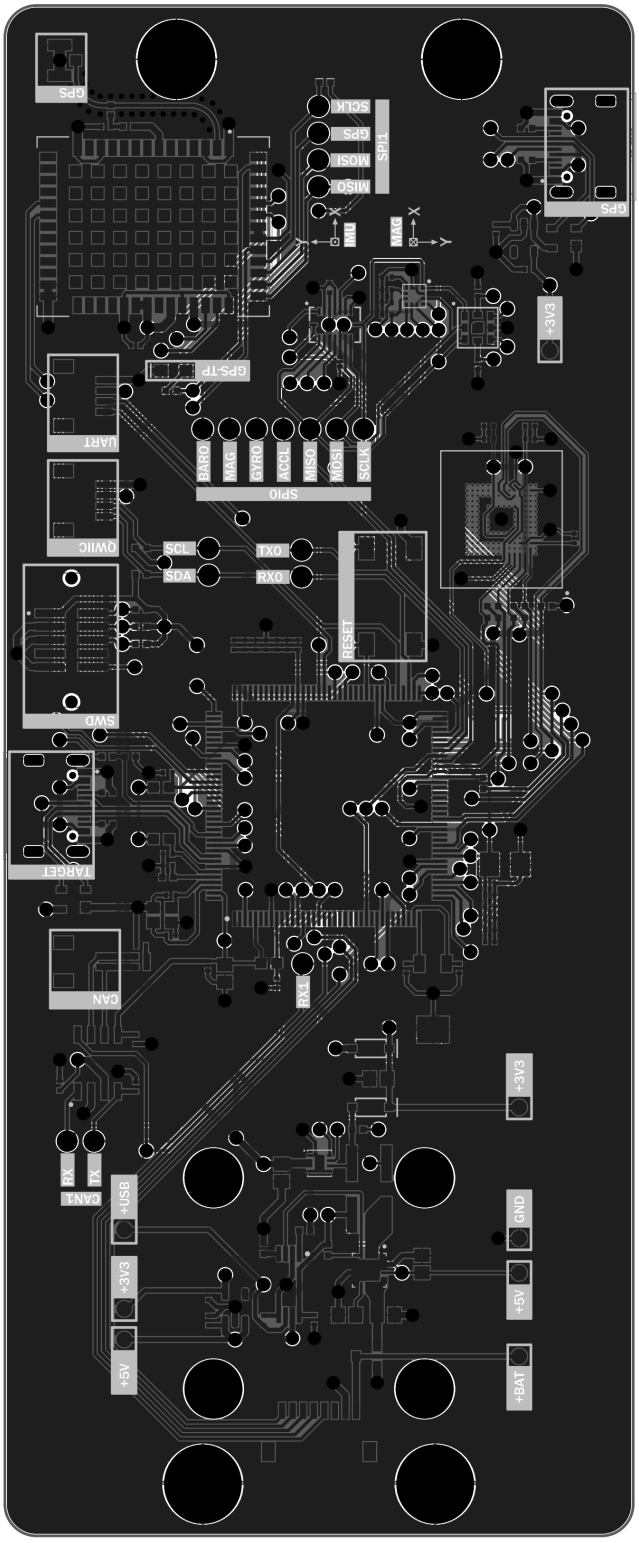

Figure A9: PCB Combined Layers

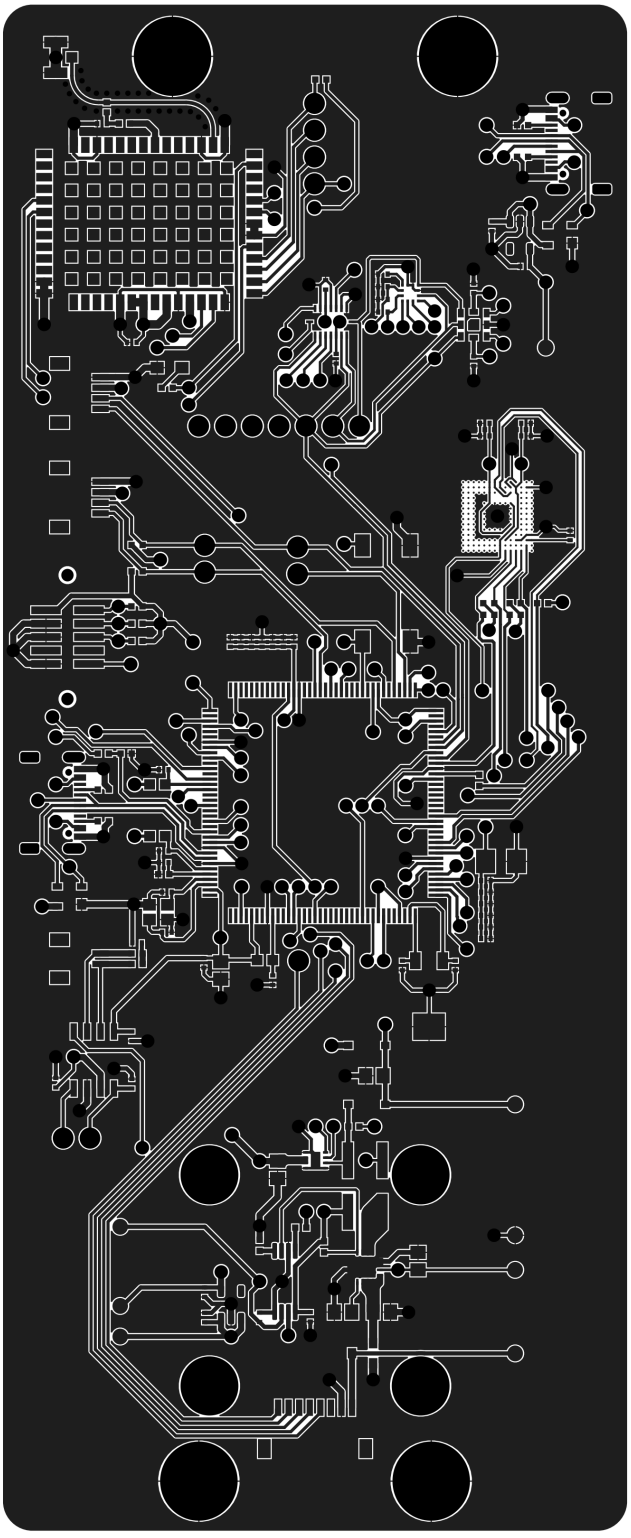

Figure A10: PCB Top Layer

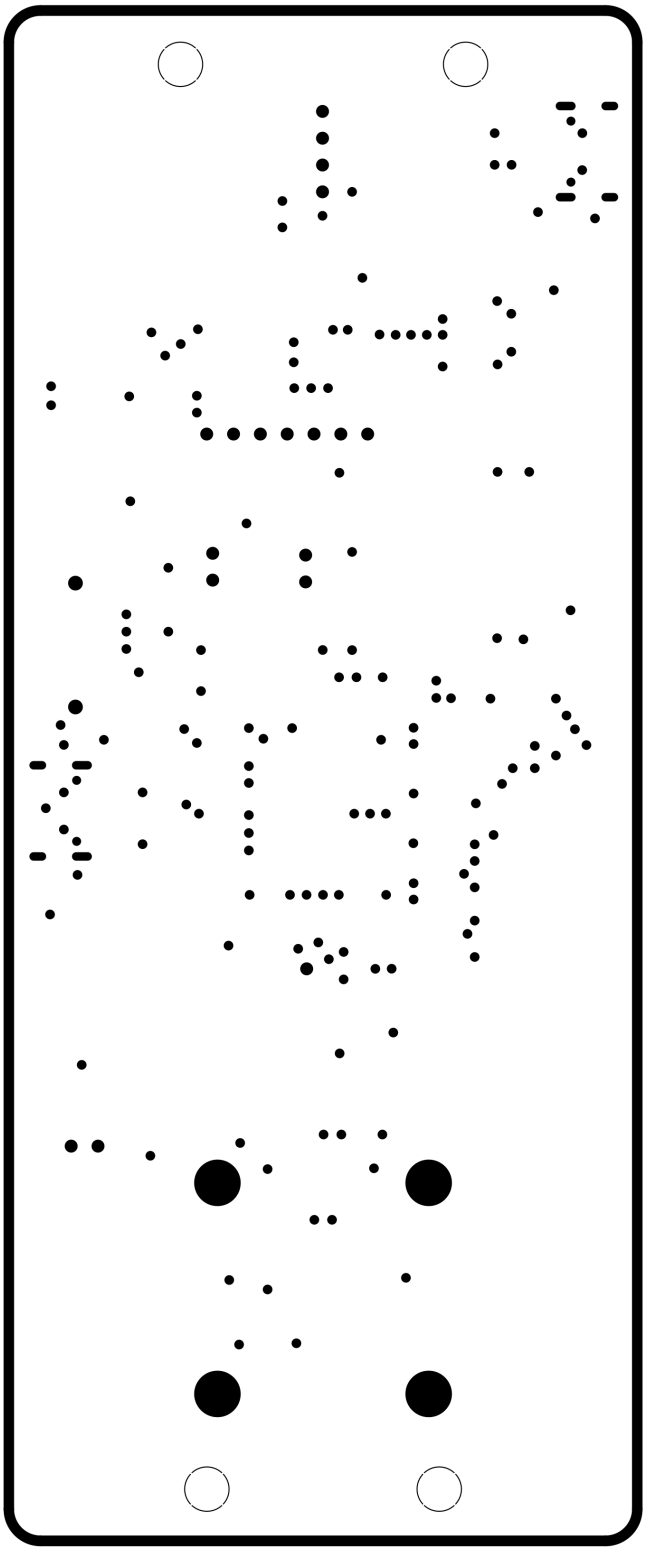

Figure A11: PCB Internal Ground Layer

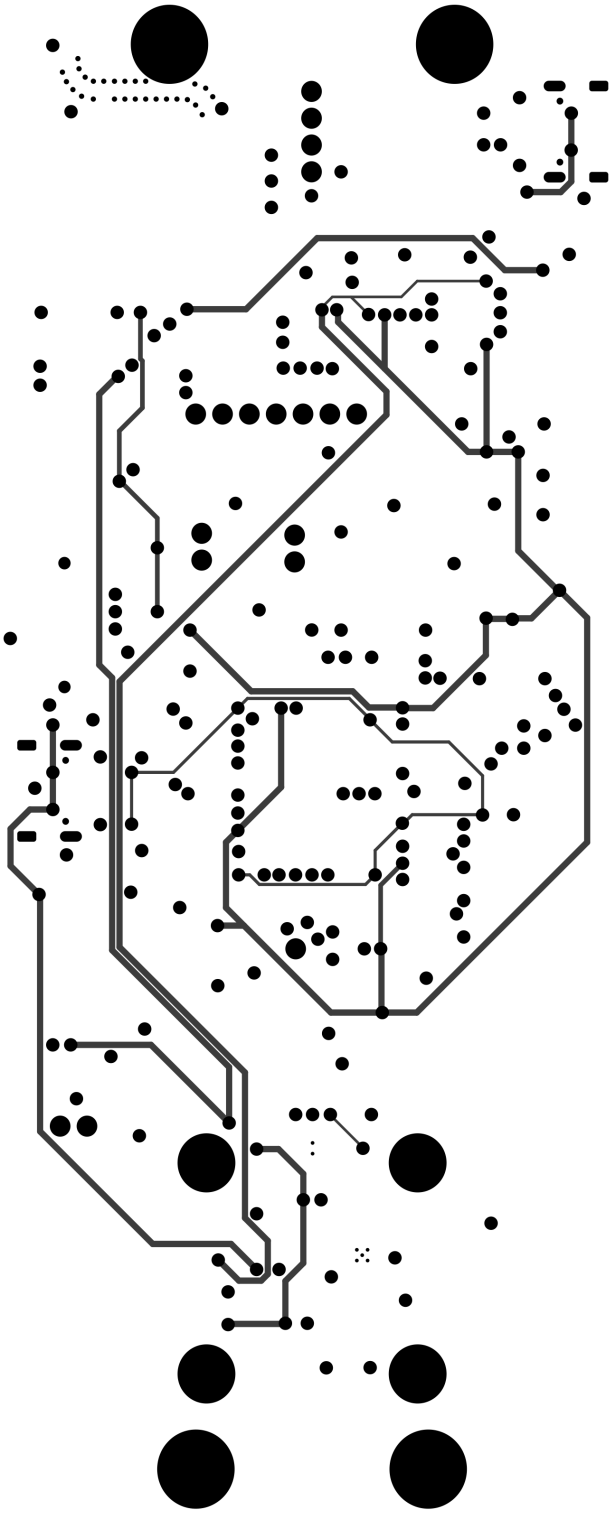

Figure A12: PCB Internal Power Layer

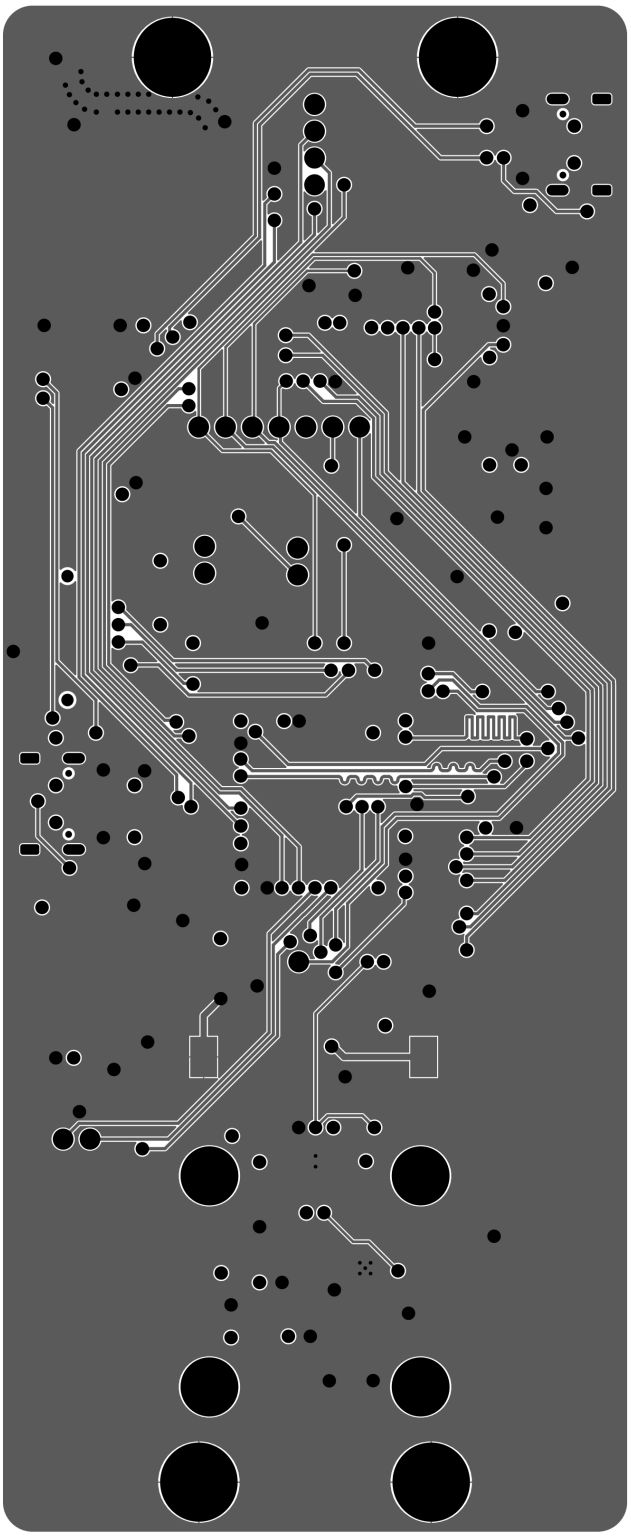

Figure A13: PCB Bottom Layer

Appendix B - Bill of Materials

Part	Schematic Reference	Quantity
BHSD-1225-SM	B1	1
BLM18PG471SN1	B2, B3, B4	3
GRM21BZ71A226ME15	C1, C6	2
GRM21BZ71E106KE15	C2	1
GCM216R71H332KA37	C3	1
GRM155R71A104KA01	C4, C5, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C32, C35, C36, C38, C39, C40, C43, C46, C47, C48, C49, C50, C51, C52, C54	33
GRM21BZ71C106KE15	C7, C27	2
GRM155Z71A105KE01	C8, C9, C44, C45, C53	5
T494A225K010AT	C10	1
GRM1555C1E4R7BA01	C28, C29	2
GRM155_0402_DNP	C30, C31	2
GRM1555C2A100JA01	C33	1
GRM155D81A475ME15	C34, C37, C42	3
GRM1555C1E150JA01	C41	1
USB4105GFA060	CON1, CON5	2
ELRS-Connector	CON2	1
QWIIC-Connector	CON3	1
Cortex-M SWD 10 Pin	CON4	1
Hobbywing-XRotorESC-45A-Connector	CON6	1
CAN-Connector	CON7	1
IRLML5203	D1	1
B0530W-7-F	D2, D3	2
PRTR5V0U2X	D4, D5	2
XEL4020-222MEC	L1, L2	2
LQG15HS47NJ02	L3	1
APT2012LVBC/D	LED1	1
CRCW04021M00FK	R1	1
CRCW0402100KFK	R2, R4	2
CRCW0402750RFK	R3	1
CRCW040239R0FK	R5, R6	2
CRCW04025K62FK	R7	1
CRCW040210K0FK	R8, R9, R10, R11	4
CRCW04024K70JN	R16, R26, R27	3
CRCW040262R0FK	R17, R18	2

CRCW04025K10FK	R19, R20, R29, R30	4
CRCW040210R0FK	R21	1
CRCW0402390RFK	R22	1
CRCW040230K0FK	R23, R24, R25	3
CRCW040230K9FK	R28	1
CRCW040247K0FK	R31	1
3-1437565-0	SW1	1
TPS7A0333	U1, U4	2
SDINBDG4-8G	U2	1
ATA6561	U3	1
ZED-F9P	U5	1
BMI088	U6	1
BMP585	U7	1
LIS3MDL	U8	1
TPS62143	U9	1
TPS2113PWR	U10	1
TPS62162	U11	1
ATSAMV71Q21B	U12	1
U.FL-R-SMT	X1	1
ProbePoint	X2, X3, X4, X5, X6, X7, X8, X10	8
ProbeHeader2	X9, X13, X14	3
ProbeHeader4	X11	1
ProbeHeader7	X12	1
ProbeHeader1	X15	1
MS1V-T1K-32.768kHz-7pF-20PPM-TA-QC-Au	Y1	1
FA-20H 12.0000MD30Z	Y2	1

Table B1: Bill of Materials