

# **GNSS Time Delay and Magnetometer Fusion for Complementary Inertial Navigation Design**

By: Punjaya Wickramasinghe  
Supervisors: Pieter van Goor & Matthew Hampsey

November 2023



**Australian  
National  
University**

## **Abstract**

Inertial Navigation Systems (INS) fuse inertial measurements to estimate a vehicle's position, velocity, and attitude. While the Extended Kalman Filter (EKF) is the industry standard for this application, it suffers from a lack of global robustness guarantees. This research extends the novel development of the Complementary Inertial Navigation System (CINS) which exploited Lie group symmetries of INS dynamics, to develop the first nonlinear, almost-globally stable state estimator. As an extension of this work, we aim to implement a magnetometer and GNSS position and velocity time-delay correction to further improve the performance of CINS. Through testing on simulations and real flight data, faster yaw convergence and reduced steady-state error were shown as compared to the original filter.

## Acknowledgements

---

First and foremost, thank you to both Pieter and Matt for having guided me through the project and dedicating so much time to explain (and re-explain) everything to me. I've learnt so much through the project and I'm truly grateful for having had you both as my supervisors. Secondly, Rob, thank you for the continued support and for pushing me every week to challenge myself. It was a pleasure getting to work in the STR lab group, and I've developed in so many ways and am grateful to have been part of such an inspiring group, so thank you. Thank you to Tridge as well for the many conversations on such short notice, you were a real source of reassurance for me throughout the project because I knew I could always come to you for help when I needed it. Also to Angela for having helped me believe in myself when I was stuck or doubting whether I could get the project done. A huge thank you to my friends and Fari, I couldn't have gotten here without you guys. And finally to my family, who have done everything in their power to make this process as easy as they could for me, so thank you, I really couldn't have done it without you.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Scope and Objectives . . . . .	2
1.2	Thesis Structure . . . . .	3
<b>2</b>	<b>Literature Review</b>	<b>4</b>
2.1	Hardware for INS . . . . .	4
2.2	Literature on Inertial Navigation Systems . . . . .	4
2.2.1	Extended Kalman Filter . . . . .	4
2.2.2	Non-Linear Observer Designs . . . . .	5
2.2.3	Complementary Filter . . . . .	6
2.3	Literature on Magnetometer Fusion for Improving Performance . . . . .	6
2.4	Literature on Time Delay Correction Techniques . . . . .	7
<b>3</b>	<b>Background</b>	<b>8</b>
3.1	Mathematical Preliminaries . . . . .	8
3.2	Lie Theory . . . . .	9
3.2.1	Specific Lie Groups: $\mathbf{SO}(3)$ , $\mathbf{SE}_2(3)$ , $\mathbf{SIM}_2(3)$ . . . . .	9
3.2.2	Corresponding Lie Algebras: $\mathfrak{so}(3)$ , $\mathfrak{se}_2(3)$ , $\mathfrak{sim}_2(3)$ . . . . .	9
3.2.3	Matrix Exponential Mapping & Tangent Spaces . . . . .	10
3.3	Coordinate Systems and Reference Frames . . . . .	11
3.3.1	Earth-Centered Earth-Fixed Coordinates . . . . .	12
3.3.2	Geodetic Coordinates . . . . .	12
3.3.3	North-East-Down Coordinates . . . . .	12
3.3.4	Reference Frames . . . . .	12
3.3.5	Conversion from Geodetic to NED Coordinates . . . . .	13
3.4	Performance Evaluation . . . . .	13
3.4.1	Root Mean Square Error . . . . .	14
3.4.2	ArduPilot – A Framework for Testing . . . . .	14
<b>4</b>	<b>Theoretical Results</b>	<b>15</b>
4.1	Problem Formulation . . . . .	15
4.2	Complementary INS Filter Architecture . . . . .	16
4.2.1	Model Dynamics . . . . .	16
4.2.2	Error Model . . . . .	17
4.2.3	State Updates . . . . .	18
4.3	Magnetometer Correction . . . . .	19
4.3.1	Problem Formulation . . . . .	19
4.3.2	Stability of Translational Error . . . . .	19
4.3.3	Lyapunov Stability of Attitude Error . . . . .	19
4.4	GNSS Position and Velocity Time Delay Correction . . . . .	21
4.4.1	Problem Formulation . . . . .	21

4.4.2	Stability of Translational Error . . . . .	23
4.4.3	Lyapunov Stability of Attitude Error . . . . .	26
4.5	Implementation . . . . .	28
4.5.1	MAG: Magnetometer Correction Filter . . . . .	28
4.5.2	DELPV: GNSS Time Delay Corrected Filter . . . . .	28
4.5.3	DELPVxMAG: Combined Magnetometer and GNSS Time Delay Corrected Filter	31
<b>5</b>	<b>Experimental Results</b>	<b>32</b>
5.1	Synthetic Simulation . . . . .	32
5.2	Real Flight Data . . . . .	35
<b>6</b>	<b>Conclusion and Future Work</b>	<b>43</b>
6.1	Summary of Findings . . . . .	43
6.2	Recommendations for Future Research . . . . .	43
6.3	Concluding Remarks . . . . .	44

## List of Figures

3.1	Geodetic, ECEF, and local NED coordinate systems [1] . . . . .	12
4.1	Isometric view of trajectory of $Y_R^\tau$ propagated in time, $t$ . . . . .	29
4.2	Side view of trajectory of $Y_R^\tau$ propagated in time, $t$ . . . . .	29
4.3	Isometric view of trajectory of $Y_R^\tau$ propagated in delay, $\tau$ . . . . .	30
4.4	Side view of trajectory of $Y_R^\tau$ propagated in delay, $\tau$ . . . . .	30
5.1	Attitude (left), velocity (centre) and position (right) estimate for CINS (black) and DELPV (red), compared to the ground truth (blue) . . . . .	33
5.2	Steady-state response (final 10 secs) of attitude (left), velocity (centre) and position (right) estimates for CINS (black) and DELPV (red), compared to the ground truth (blue) . . . . .	33
5.3	Observer error for attitude, velocity, position and the Lyapunov value for CINS (black) against DELPV (black) . . . . .	34
5.4	Trajectory of the real-flight test case comparing EKF3 (cyan) against DELPV×MAG (pink) . . . . .	36
5.5	Position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates . . . . .	37
5.6	Attitude estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) . . . . .	37
5.7	Transient response (first 90 secs) of position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates . . . . .	38
5.8	Transient response (first 160 secs) of attitude estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) . . . . .	39
5.9	Steady-state response (zoomed regions after 200 sec) of position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates . . . . .	40
5.10	Steady-state response (zoomed regions after 200 sec) of attitude estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) . . . . .	40

## List of Tables

1	Project Scope Table	3
2	System Inputs	15
3	System States	15
4	Comparison of RMSE values for CINS and DELPV×MAG	35
5	Full flight RMSE for each filter compared to EKF3 across each individual and combined state	41
6	Transient (first 60 seconds) RMSE for each filter compared to EKF3 across each individual and combined state	42
7	Steady-state (final 60 seconds) RMSE for each filter compared to EKF3 across each individual and combined state	42

# 1 Introduction

---

Navigation, tracing its origins to the dawn of civilisation, has been a cornerstone in human progression. From the stars charted by early sailors to the trade routes that helped build economies, navigation has played a significant role in shaping the development of societies throughout history [2]. With the wide adoption of systems like the Global Navigation Satellite System (GNSS) in the latter part of the 20th century [3], losing one’s way became a concern of the past. Navigation technology, through continued innovation, has evolved into a remarkably mature suite of tools across all modes of travel, where technologies like radio navigation, advanced radar systems and sonar have long aided land, aviation, and maritime navigation [4, 5]. With the rapid development of robotics and control systems, a new challenge has emerged in navigation – the development of fully autonomous navigation systems. These systems are already being used in various fields, such as agriculture, search-and-rescue, wildfire monitoring, delivery services, and military operations. Unlike its predecessors, autonomous navigation requires self-correcting mechanisms to manage errors that arise from the lack of human supervision. In the control community, this is known as a state estimation problem [6].

In the context of Unmanned Aerial Vehicles (UAVs), this problem involves estimating the *extended pose*, which includes the attitude, position and velocity by using a combination of sensor measurements and predictive models [6, 7, 8, 9]. The literature in this space showcases many different approaches for UAV state estimation, from deep learning methods that leverage sophisticated neural network architectures to optical sensor-based visual odometry techniques [10, 11, 12]. Among these techniques, one of the more rudimentary approaches is inertial navigation – the only known form of navigation entirely independent of external information [13]. This approach relies on Inertial Navigation Systems (INS) to leverage information about a vehicle’s starting location in a given reference frame, as well as its current motion to estimate its state in the same reference frame. Typically, these systems depend on inertial measurements of specific acceleration and angular velocity from an Inertial Measurement Unit (IMU) [8]. High-quality IMUs can provide more accurate and reliable navigation information, but they are often more expensive and may consume more power. State estimation that depends on the integration of these IMU signals over time alone is known as dead-reckoning, which, in the absence of noise and disturbance of measurements and dynamics, provides a sufficient result. In practice, the presence of noise leads to the rapid accumulation of error and drift in this method. The primary solution to this problem has been to employ more sophisticated algorithms that fuse exteroceptive sensor measurements to the predictive model given by navigation. These measurements include, but are not limited to those from a GNSS, magnetometer and barometers. In this regard, the industry standard of UAV state estimation is the Extended Kalman Filter (EKF), a traditional stochastic filter.

The EKF is a locally optimal controller that operates under the assumption that the noise in the pose estimates and the sensor measurements are all Gaussian in nature and that the system dynamics can be approximated by a first-order Taylor series expansion about the current best estimate [9]. These assumptions allow the EKF to provide such optimal estimates of the state in conjunction with available sensor measurements, as it assumes that the true state lies within a specific range of the current estimate. The cost of this optimality is its susceptibility to divergent behaviour when highly nonlinear dynamics and

non-Gaussian noise characteristics violate the fundamental assumptions of the EKF. Numerous algorithms have been developed to address the limitations in the robustness of the EKF. Some of these algorithms have semi-global stability guarantees, but only a few have achieved almost-global robustness stability. Van Goor et al's 2021 study proposed a novel trajectory-independent filter for INS called the Complementary Inertial Navigation System (CINS), which has almost globally asymptotically stable and locally exponentially stable error dynamics, independent of any choice of gains [14]. This filter is the first of its kind and has been tested in simulation using IMU measurements and GNSS. The filter has strong convergence properties from extreme initial errors and is a viable solution for UAVs that experience highly nonlinear motions or noise characteristics, UAVs with limited sensor availability, and UAVs in scenarios where sensor failures or reboots are possible.

## 1.1 Project Scope and Objectives

The present work aims to improve the current implementation of CINS with a magnetometer data fusion, as well as a GNSS position and velocity time delay correction. The magnetometer correction is intended to primarily aid in attitude estimation, whereas the GNSS position and velocity time delay correction aim to generally reduce the error across states in the filter. Thus, the research question for this project is

*Can a magnetometer and GNSS position and velocity time delay correction improve the overall performance of the Complementary Inertial Navigation System?*

Table 1 outlines the key expected outcomes and constraints of the project.

Category	Details
Expected Outcomes	<ol style="list-style-type: none"> <li>1. Python implementation of a magnetometer-corrected version of CINS.</li> <li>2. Python implementation of a GNSS position and velocity time delay corrected version of CINS.</li> <li>3. Evaluation of the designed filters with real-world flight data using the ArduPilot framework.</li> </ol>
Exclusions	<ol style="list-style-type: none"> <li>1. Consideration of bias in sensor data or other input sensors such as barometer or airspeed sensors.</li> <li>2. Discretisation of the system dynamics.</li> <li>3. Applicability to aerial vehicles only, excluding ground or marine autonomous vehicles.</li> </ol>

**Table 1:** Project Scope Table

## 1.2 Thesis Structure

Section 2 discusses the pivotal role of Inertial Measurement Units (IMUs) in advancing modern-day inertial navigation for smaller and cheaper vehicles. It traces the evolution from the Extended Kalman Filter (EKF) to modern nonlinear observer designs, culminating in the Complementary Filter, which underpins this study’s focus on the Complementary Inertial Navigation System (CINS). Section 3 delves into the mathematical backbone of this research, covering lie theory, coordinate frame transformations, the CINS filter itself, the benchmarks for evaluating performance, and the testing framework established to corroborate our findings. Section 4 presents the theoretical findings of this work – namely, the structure and approach for the corrections explored in this work. Thus, we define our system and its parameters and show the necessary stability guarantees of the proposed magnetometer fusion and GNSS time delay corrections. In Section 5, we detail the performance of our designed filters through three distinct test scenarios: synthetic simulation, noisy simulation, and real-world flight tests, providing a comprehensive analysis of their efficacy. Section 6 summarizes the research’s main discoveries and discusses potential directions for future research in this field.

## 2 Literature Review

---

In this section, we will examine the development of IMU hardware, with a focus on MEMS IMUs. We will also provide an overview of the evolution of INS algorithms over time, highlighting the advantages of nonlinear observer design, magnetometer data fusion as well as compensating for time delay, as discussed in the literature.

### 2.1 Hardware for INS

Almost all IMUs fit under two broad categories – stable platform and strap-down systems. The main difference between them is the reference frame with which they report their measurements with respect to. The former generates inertial measurements relative to a global frame, whereas the latter generates measurements relative to the body frame [8]. Stable platform systems have sensors mounted on a platform with gimbals that always remain in alignment with the global frame. It does this with a feedback system that rotates the gimbals to cancel any rotations to maintain this alignment [8]. On the contrary, strap-down systems are ones where the inertial sensors are rigidly attached (hence strapped down) to the body of the vehicle [15]. Unlike stable platform systems, these systems require far more computation to determine the necessary inertial measurements. Both systems rely on the same underlying principles. Stable platform systems are more mechanically complex and expensive to build, while strap-down systems are far more computationally expensive to use. As the cost of computation has decreased over time, the strap-down system has become the most common form of INS. [8] [15].

Given the enormous variety of applications that exist for the use of IMUs (aerospace, smartphones, medical equipment, etc.), there has been a significant push in the market for manufacturers to create IMUs for use in a variety of performance ranges and for a variety of application requirements [8]. These factors inherently affect the price and underlying technology used in the IMUs. For instance, for high-precision tasks such as spacecraft navigation, where consistent accuracy over long periods is crucial, or in advanced defence systems for missile guidance to ensure precise targeting, Fiber Optic Gyroscope and Ring Laser Gyroscope IMUs are some of the better options. For more low-cost, low-power, miniaturised needs, such as UAV navigation, Commercial-off-the-shelf or Micro-Electro-Mechanical Systems (MEMS), IMUs are ideal [13]. MEMS IMUs, which are built using silicon micro-machining techniques [8], in particular, have become the most popular choice of IMU for most applications due to their small form factor, low weight, reduced start-up times and relatively low cost [13].

### 2.2 Literature on Inertial Navigation Systems

#### 2.2.1 Extended Kalman Filter

Since its inception, the classical solution to many control problems, including the inertial navigation problem, has been the Extended Kalman Filter (EKF), a revolutionary nonlinear extension of the original Kalman Filter [16]. The EKF parameterises the full system on the Euclidean space and employs the first-order Taylor series expansion to linearise a nonlinear system around the current best state estimate

[16]. It is particularly effective for low to moderately non-linear systems, where the linearisation doesn't result in significant being introduced. It has been used in a wide range of applications, including those in robotics for trajectory tracking state estimation, SLAM and object detection, as well as for applications in economics and medicine. Despite its success, the EKF has a few limitations, the most prominent one being that estimation is not optimal if the system is highly nonlinear since substantial linearisation error can lead to poor filter performance. Thus, the EKF suffers with providing local trajectory-dependent stability guarantees [16]. To address these limitations, researchers have explored alternatives to the EKF [17].

The Unscented Kalman Filter (UKF) was one of the earliest modifications of the EKF that leveraged the Unscented Transform to help with the accumulation of linearisation error, particularly in the estimation of the covariance [16]. It is based on the idea that approximating a probability distribution is often more straightforward than an arbitrary linear transformation [16]. Another key development in the evolution of the EKF was the Invariant Extended Kalman Filter (IEKF). This filter seeks to retain the natural symmetries of the EKF after the filtering process. It uses geometrically adapted correction terms instead of linear correction terms to reduce linearisation error and hence, ensure convergence of the system over a much larger initialisation domain [16]. More specifically, the attitude is parameterised on the Lie group structure known as the Special Euclidian Group. While these Kalman-type filters have shown a capacity for local optimal estimation and have been shown to be able to operate over an ever-increasing domain of stability, these filters aren't able to offer global asymptotic robustness guarantees. This can lead to performance degradation of even instability. This limitation has spurred the rapid development of nonlinear observers in the research community, which conversely generally do not make assumptions of linearity.

### 2.2.2 Non-Linear Observer Designs

By modelling the system with more precision, nonlinear observers offer the advantage of typically requiring less computational power and offering greater robustness guarantees in the face of disturbances and uncertainty in initial conditions [17]. There has been extensive research on non-linear observers to provide robustness guarantees. Recent research by Fusini et al. [17] has introduced two nonlinear observers for UAV state estimation that utilise altimeter, GNSS, IMU and camera measurements as well as a novel vision algorithm that estimated velocity based on optical flow. They showcased global exponential stability when using direct sensor measurements for roll and pitch and uniform local exponential stability when these measurements are replaced with estimated feedback.

Bjorne et al. [18] developed a new nonlinear observer for UAV navigation that demonstrates semi-global asymptotic stability by leveraging measurements from a conventional sensor suite, including a camera, IMU, and GNSS. The observer's innovation rests on its capacity to concurrently estimate range and compensate for gyro bias without the need for magnetometers or altimeters, potentially mitigating environmental susceptibilities. This approach not only strengthens the robustness of UAV navigation but also simplifies the sensor complexity, marking a significant stride in autonomous UAV systems. A study done by Berkane et al. (2021) [19] implemented a novel nonlinear observer design to achieve a full state estimate of a vehicle in three-dimensional space. They integrated IMU and positional measurements to

estimate the pose and bias in the gyroscope and were able to produce a semi-globally exponentially stable filter. Despite these recent developments in the control community towards global asymptotic stability, very few work has been done that has been able to achieve this.

### 2.2.3 Complementary Filter

Mahony et al. (2005) [20] introduced a nonlinear, called the complementary filter. It was initially proposed in the context of UAV navigation, using IMU measurements for attitude and gyroscope bias estimation within the SO(3) Lie group structure. [20]. This filter, easily adaptable to quaternion representation, was later refined as the explicit complementary filter by the author [21] to enhance computational efficiency. Since then, the complementary filter has formed the foundation for work in state estimation in many applications.

A study done by William et al. (2023) [22] integrated multiple state estimators, including the explicit complementary filter, to enhance the localisation capabilities of autonomous vehicles. Their approach demonstrated improved attitude estimation and correction of sensor bias, yielding better accuracy and reliability even with basic sensors. In 2011, Vasconcelos et al. [23] developed a complementary filtering-based INS tailored for autonomous surface crafts, focusing on position and attitude estimation. The filters' low computational load allows for efficient use in oceanic applications with low-power hardware. Narkhede et al. (2021) [24] proposed a cascaded complementary filter architecture that merges aspects of nonlinear and linear filters to improve attitude estimation. This method corrects for gyroscope bias and estimates the vehicle's attitude, operating independently of the chosen gains. The aforementioned almost-globally asymptotically and locally exponentially stable filter, known as CINS, is also a filter that has its foundations in the complementary filter. CINS extends the capabilities of the original complementary filter by providing convergence guarantees for not only attitude estimates but for all states of the INS problem by integrating position and velocity estimation into the framework [25].

## 2.3 Literature on Magnetometer Fusion for Improving Performance

Magnetometers are sensors that measure the strength and direction of the magnetic field in the vicinity of a vehicle in its body frame [26]. These sensors are used to calculate the orientation of the vehicle in relation to the Earth's magnetic field. This information is particularly useful for yaw correction since it removes the absolute requirement for the persistence of excitation. Humphreys et al. (2005) [27] developed a novel magnetometer-based system to accurately determine the attitude and wire boom orientations of a spacecraft. Even under a suite of modelling errors, this system used leveraged inertial angular momentum to improve estimation accuracy. A study conducted by Renaudin et al. (2010) [28] presented a novel calibration method for magnetometers that compensates for spatially variable magnetic disturbances that cause drops in accuracy. This approach involves using multiple magnetometers in a specific geometric configuration to better account for magnetic fluctuations.

## 2.4 Literature on Time Delay Correction Techniques

GNSS measurements often suffer from a delay as a consequence of the computational load placed on the receiver to estimate the position/velocity, on top of the delays stemming from the serial data communication between the receiver and the processing computer [29]. This delay leads to outdated measurements of the position being used, inducing a systemic error in the fusion of measurements into the state estimator. Typically, GNSS delays can range in the order of a few hundred milliseconds [29]. For low-speed applications such as ship navigation, the error caused by GNSS time delay is insignificant; however, for highly dynamic vehicles like UAVs, this time delay can cause several meters of position error, which could prove fatal for the safe operation of the device [29].

Early work in handling the problem of time delay was identified and approached in a work by Jacovitti et al. (1933) where methods in discrete-time analysis were discussed [30]. Another work by Skog et al. (2011) [31] demonstrated empirical results for the impact of time synchronisation errors in GPS-aided INS, and proposed a software-based solution. This method was shown to assist in time synchronisation accurate to a few milliseconds. A work in 2015 by Khosravian et al. [32] explored nonlinear attitude estimation for highly dynamic motion with delayed, but constant, GNSS velocity and magnetometer measurements. The proposed method involved using a delayed observer to provide estimates of delayed states, in combination with a dynamic predictor to transform the previous estimation into the current state. Hansen et al. (2015) [29] proposed a semi-globally exponentially stable nonlinear observer for estimating the extended pose of a UAV that depended on IMU and delayed GNSS measurements. The authors configure the system to ensure the inertial and magnetometer measurements are delayed to overlap with the timing of the GNSS measurements, relying on a fast simulator to recover the present time state from a delayed state estimate.

### 3 Background

---

#### 3.1 Mathematical Preliminaries

This section defines several essential matrix properties and operations that provide the necessary tools to complete advanced derivations later in the thesis. We begin by introducing skew-symmetric matrix representation. Given a vector  $a = [a_1, a_2, a_3]^\top \in \mathbb{R}^3$ , its skew-symmetric matrix representation is defined as:

$$a^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Using this representation, the cross product between vectors  $a$  and  $b$  can be expressed as:

$$a \times b = a^\times b$$

We also introduce the following identities, which apply for all vectors  $a, b \in \mathbb{R}^3$ :

$$a^\times b = -b^\times a \quad (a^\times)^\top = -a^\times \quad (a \times b)^\times = ba^\top - ab^\top$$

Next, we outline some trace and matrix properties for  $A, B, C \in \mathbb{R}^{n \times n}$  and  $c \in \mathbb{R}$ :

$$\begin{aligned} \text{tr}(A) &= 0, \quad A \in \mathfrak{so}(3) \\ \text{tr}(c) &= c, \quad c \in \mathbb{R} \\ \text{tr}(ABC) &= \text{tr}(CAB) \\ \text{tr}(A^\top) &= \text{tr}(A) \\ \exp(A) &= I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots \\ \exp(A + B) &= \exp(A) \exp(B), \quad AB = BA \\ \frac{d}{dt} A^{-1} &= -A^{-1} \dot{A} A^{-1} \end{aligned}$$

Given  $M \in \mathbb{R}^{n \times n}$ , we also define the following projectors:

$$\begin{aligned} \mathbb{P}_{\text{skew}} : \mathbb{R}^{n \times n} &\rightarrow \mathfrak{so}(n), \quad \mathbb{P}_{\text{skew}}(M) = \frac{1}{2}(M - M^T) \\ \mathbb{P}_{\text{sym}} : \mathbb{R}^{n \times n} &\rightarrow \text{Sym}(n), \quad \mathbb{P}_{\text{sym}}(M) = \frac{1}{2}(M + M^T) \\ M &= \mathbb{P}_{\text{skew}}(M) + \mathbb{P}_{\text{sym}}(M) \end{aligned}$$

## 3.2 Lie Theory

A Lie group is a group,  $G$ , which is also a differentiable manifold, such that the group operations are smooth maps. A Lie algebra, on the other hand, represents the tangent space at the identity of its associated Lie group, capturing the local structure of the group. Lie groups offer a robust mathematical representation of the states of a vehicle operating in three-dimensional space. They are a tool for handling the non-linearity associated with the continuous transformation of a vehicle's pose to ensure precise estimation while providing the necessary structure for complete-state representation [33, 34]. In this application, we study Lie groups that are closed subsets of  $\mathbf{GL}(n, \mathbb{R})$ .

### 3.2.1 Specific Lie Groups: $\mathbf{SO}(3)$ , $\mathbf{SE}_2(3)$ , $\mathbf{SIM}_2(3)$

The special orthogonal group  $\mathbf{SO}(3)$  consists of all  $3 \times 3$  orthonormal matrices with a determinant of 1. These matrices represent rotations in 3D space:

$$\mathbf{SO}(3) := \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I, \det(R) = 1\}. \quad (3.1)$$

The extended special Euclidean group  $\mathbf{SE}_2(3)$  captures rigid body motions in three-dimensional space. These motions can be decomposed into rotations, translations and velocities:

$$\mathbf{SE}_2(3) := \left\{ \begin{bmatrix} R & V \\ 0_{2 \times 3} & I_2 \end{bmatrix} \mid R \in \mathbf{SO}(3), V \in \mathbb{R}^{3 \times 2} \right\}. \quad (3.2)$$

The Lie group  $\mathbf{SIM}_2(3)$  represents the automorphisms of  $\mathbf{SE}_2(3)$ :

$$\mathbf{SIM}_2(3) := \left\{ \begin{bmatrix} R & V \\ 0_{2 \times 3} & A \end{bmatrix} \mid R \in \mathbf{SO}(3), V \in \mathbb{R}^{3 \times 2}, A \in \mathbf{GL}(2) \right\}. \quad (3.3)$$

For convenience, we may denote an element of  $\mathbf{SE}_2(3)$  as  $X = (R, V)$ , where  $R \in \mathbf{SO}(3), V \in \mathbb{R}^{3 \times 2}$ . Similarly, an element of  $\mathbf{SIM}_2(3)$  may be denoted as  $Z = (R, V, A)$  where  $R \in \mathbf{SO}(3), V \in \mathbb{R}^{3 \times 2}, A \in \mathbf{GL}(2)$ .

### 3.2.2 Corresponding Lie Algebras: $\mathfrak{so}(3)$ , $\mathfrak{se}_2(3)$ , $\mathfrak{sim}_2(3)$

The Lie algebras corresponding to their associated Lie groups provide a structure for analysing the local behaviour at any point on the group's manifold through its tangent space.

For the group  $\mathbf{SO}(3)$ , its corresponding Lie algebra  $\mathfrak{so}(3)$  consists of  $3 \times 3$  skew-symmetric matrices which capture infinitesimal rotations.

$$\mathfrak{so}(3) := \{\omega^\times \in \mathbb{R}^{3 \times 3} \mid \omega^{\times T} = -\omega^\times, \omega \in \mathbb{R}^3\}. \quad (3.4)$$

The Lie algebra  $\mathfrak{se}_2(3)$  of the group  $\mathbf{SE}_2(3)$  captures infinitesimal rigid body motions in three-dimensional

space:

$$\mathfrak{se}_2(3) := \left\{ \begin{bmatrix} \omega^\times & V \\ 0_{2 \times 3} & 0_{2 \times 2} \end{bmatrix} \mid \omega^\times \in \mathfrak{so}(3), V \in \mathbb{R}^{3 \times 2} \mathbf{p} \right\}. \quad (3.5)$$

Lastly, the Lie algebra  $\mathfrak{sim}_2(3)$  of the group  $\mathbf{SIM}_2(3)$  represents infinitesimal group affine transformations in the three-dimensional space:

$$\mathfrak{sim}_2(3) := \left\{ \begin{bmatrix} \Omega^\times & V \\ 0_{2 \times 3} & S \end{bmatrix} \mid \Omega^\times \in \mathfrak{so}(3), S \in \mathfrak{gl}(2), V \in \mathbb{R}^{3 \times 2} \right\}. \quad (3.6)$$

### 3.2.3 Matrix Exponential Mapping & Tangent Spaces

The local behaviour of a Lie group around at any point on its manifold can be linearised by its associated Lie algebra. These locally linear changes in the tangent space at a given point must be integrated and mapped to the global state space (nonlinear manifold) for meaningful state representations. The exponential maps associated with the group and its algebra establish the bridge between this local and global representation. The matrix exponential provides a map from the Lie algebra to its corresponding Lie group. Specifically, for a matrix  $X$  in the Lie algebra, the matrix exponential  $\exp(X)$  produces a matrix in the Lie group:

$$\exp : \mathfrak{g} \rightarrow G. \quad (3.7)$$

The combination of global state representation on the manifold and the locally linear approximation provided by the Lie algebra provides an efficient and accurate framework for state estimation.

Below, we define two particularly useful matrix ODE properties that will be used later in the Section 4 when deriving the new filter.

**Lemma 3.1** (Right-Invariant ODE). Let  $X(t) \in \mathbb{R}^{n \times n}$  be a matrix whose dynamics are governed by  $\dot{X}(t) = DX(t)$ , where  $D \in \mathbb{R}^{n \times n}$  is a constant matrix. Assuming the matrix-valued function  $DX(t)$  is Lipschitz continuous, and given the initial condition  $X(0)$ , then the solution to the differential equation is uniquely given by:

$$X(t) = \exp(Dt)X(0). \quad (3.8)$$

*Proof.* Suppose  $X(t) = \exp(Dt)X(0)$  is a solution to the ODE. Differentiating with respect to  $t$ , we get:

$$\begin{aligned} \dot{X}(t) &= \frac{d}{dt}(\exp(Dt)X(0)), \\ &= D \overbrace{\exp(Dt)X(0)}^{X(t)}, \\ &= DX(t). \end{aligned}$$

■

**Lemma 3.2** (Left-Right Invariant Discrete Time Update). Let  $X(t) \in \mathbb{R}^{n \times n}$  be a trajectory whose dynamics are given by  $\dot{X} = X(t)A + BX(t)$ , where  $A, B$  are both constant and elements of  $\mathbb{R}^{n \times n}$ . Given

the initial condition,  $X(0)$ , and a time increment,  $\delta t$ , the discrete-time update is given by:

$$X(t + \delta t) = \exp(B\delta t)X(t)\exp(A\delta t). \quad (3.9)$$

*Proof.* Suppose  $X(t) = \exp(Bt)X(0)\exp(At)$ . Then

$$\begin{aligned}\dot{X}(t) &= \frac{d}{dt}(\exp(Bt)X(0)\exp(At)) \\ &= \frac{d}{dt}(\exp(Bt)X(0))\exp(At) + \exp(Bt)X(0)\frac{d}{dt}(\exp(At)) \\ &= B\exp(Bt)X(0)\exp(At) + \exp(Bt)X(0)\exp(At)A \\ &= X(t)A + BX(t)\end{aligned}$$

Therefore,  $X(t) = \exp(Bt)X(0)\exp(At)$  is a solution of  $\dot{X} = X(t)A + BX(t)$ . Since  $X(t)A + BX(t)$  is smooth, the uniqueness of the solution follows from *Theorem D.4* [35].

Then, we consider the discrete-time update,  $X(t + \delta t)$ :

$$\begin{aligned}X(t + \delta t) &= \exp(B(t + \delta t))X(0)\exp(A(t + \delta t)) \\ &= \exp(B\delta t + Bt)X(0)\exp(At + A\delta t)) \\ &= \exp(B\delta t)\exp(Bt)X(0)\exp(At)\exp(A\delta t) \\ &= \exp(B\delta t)X(t)\exp(A\delta t)\end{aligned}$$

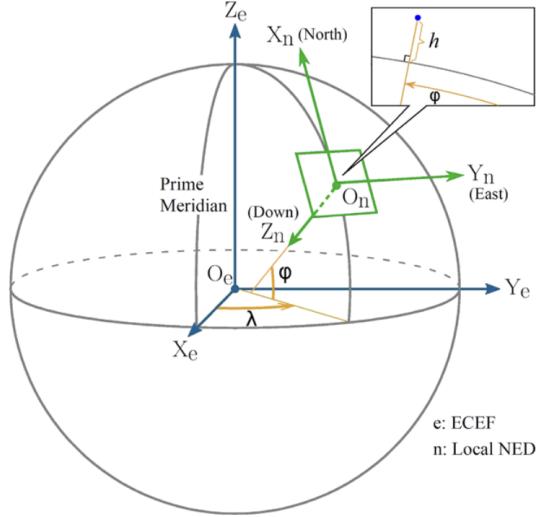
■

**Corollary 3.2.1.** Let  $X(t) \in \mathbb{R}^{n \times n}$  be a trajectory whose dynamics are  $\dot{X}(t) = X(t)A$ , where  $A \in \mathbb{R}^{n \times n}$  is a constant matrix. The equivalent system is obtained when we set  $B = 0$  for the system formulated in Lemma 3.2. Therefore, by setting  $B = 0$  in the discrete-time update from Lemma (3.2), we deduce the state update as  $X(t + \delta t) = X(t)\exp(A\delta t)$ .

**Corollary 3.2.2.** Let  $X(t) \in \mathbb{R}^{n \times n}$  be a trajectory whose dynamics are  $\dot{X}(t) = BX(t)$ , where  $B \in \mathbb{R}^{n \times n}$  is a constant matrix. The equivalent system is obtained when we set  $A = 0$  for the system formulated in Lemma 3.2. Therefore, by setting  $A = 0$  in the discrete-time update from Lemma (3.2), we deduce the state update as  $X(t + \delta t) = \exp(B\delta t)X(t)$ .

### 3.3 Coordinate Systems and Reference Frames

In order to meaningfully define the state of a UAV, it is necessary to define a coordinate system, along with a reference frame. Different coordinate systems are used in different applications depending on the requirements of the problem. Here, we introduce the three systems applicable to this work: Earth-Centered Earth-Fixed (ECEF) coordinates, North-East-Down (NED) coordinates and Geodetic coordinates.



**Figure 3.1:** Geodetic, ECEF, and local NED coordinate systems [1]

### 3.3.1 Earth-Centered Earth-Fixed Coordinates

ECEF coordinates use Cartesian coordinates  $(X, Y, Z)$  to represent position relative to the centre of the Earth. The  $X$ -axis points towards the intersection of the Prime Meridian and the Equator, the  $Y$ -axis points towards the intersection of 90 degrees East longitude and the Equator, and the  $Z$ -axis points towards the North Pole [36].

### 3.3.2 Geodetic Coordinates

A geodetic system relies on measurements of latitude ( $\varphi$ ), longitude ( $\lambda$ ) and altitude ( $h$ ) to represent position relative to the centre of the Earth. Latitude and longitude are the angular distance from the Prime Meridian and Equator, respectively, while the altitude is the height above sea level.

### 3.3.3 North-East-Down Coordinates

Unlike ECEF and geodetic systems, which are global coordinate systems, NED coordinates are a local tangent plane Cartesian coordinate system  $(x, y, z)$  with its origin at an arbitrary point on the Earth's surface. The  $x$ -axis points North, the  $y$ -axis points East, and the  $z$ -axis points towards the centre of the Earth. Figure 3.1 illustrates all three coordinate systems, with ECEF coordinates in blue, geodetic coordinates in yellow, and NED coordinates in green.

### 3.3.4 Reference Frames

Frames of reference are an essential concept in order to define any vector in space accurately - without a frame of reference, such measurements become meaningless. To transform measurements from one reference frame to another, a transformation matrix is used that includes both a rotational and translational component. The CINS filter uses two primary reference frames for the state estimation problem: an inertial NED frame and a body-fixed frame [37]. It serves as a global reference that is not affected by

the UAV's motion or orientation and is the point from which all measurements of the UAV's state can be evaluated. The body-fixed frame is positioned at the UAV's centre of mass and follows the drone's motion. The  $x$ -axis originates from the UAV's nose, the  $y$ -axis from the right wing, and the  $z$ -axis vertically downwards, pointing towards the Earth's core.

### 3.3.5 Conversion from Geodetic to NED Coordinates

It's crucial to have consistency between measurements expressed in different coordinate frames in order to compare different filter outputs effectively. In this section, we will discuss a method of converting the geodetic coordinates of a UAV position  $(\varphi, \lambda, h)$  to NED coordinates. This conversion will enable us to have all filter outputs in the NED system. We will assume a spherical Earth model with a radius of  $R = 6378100.0$  meters. However, we can use more complex models like WGS-84 to achieve higher accuracy. Additionally, we assume that the origin of the inertial frame ( $p_0$ ) is known in geodetic coordinates  $(\varphi_0, \lambda_0, h_0)$ .

The geodetic coordinates of the vehicle and origin positions are firstly converted to ECEF coordinates and are denoted as  $\mathbf{P} = (X, Y, Z)$  and  $\mathbf{P}_0 = (X_0, Y_0, Z_0)$  respectively. Given  $r = h + R$ , we use the following expression

$$\begin{aligned} X &= r \cdot \cos(\varphi) \cdot \cos(\lambda) & Y &= r \cdot \cos(\varphi) \cdot \sin(\lambda) & Z &= r \cdot \sin(\varphi) \\ X_0 &= r \cdot \cos(\varphi_0) \cdot \cos(\lambda_0) & Y_0 &= r \cdot \cos(\varphi_0) \cdot \sin(\lambda_0) & Z_0 &= r \cdot \sin(\varphi_0) \end{aligned}$$

A rotation matrix  $R_n$  is calculated to transform the ECEF coordinates to the local tangent plane defined by the reference point:

$$R_n = \begin{bmatrix} -\sin(\varphi_0) \cdot \cos(\lambda_0) & -\sin(\varphi_0) \cdot \sin(\lambda_0) & \cos(\varphi_0) \\ -\sin(\lambda_0) & \cos(\lambda_0) & 0 \\ -\cos(\varphi_0) \cdot \cos(\lambda_0) & -\cos(\varphi_0) \cdot \sin(\lambda_0) & -\sin(\varphi_0) \end{bmatrix}$$

Finally, the position of the vehicle in the NED system ( $P_{NED}$ ) is given by:

$$P_{NED} = R_n \cdot (\mathbf{P} - \mathbf{P}_0)$$

## 3.4 Performance Evaluation

In order to meaningfully assess the performance of the filter before and after the inclusion of the proposed improvements, a set of performance metrics must be established. Various evaluation methods exist in the literature.

Dam et. al (2023) implemented a filtered high-gain observer for a class of nonlinear systems subject to delayed measurements, noise measurement effects and disturbances [7]. A comparison was made with a Kalman-like observer through a direct comparison of the position, velocity and attitude estimations of both filters. Al-Absi et. al (2021) explored a method for improving the tracking performance of UAVs

with the novel implementation of the Uncertainty and Error-Aware Kalman Filter and used RMSE to compare the performance of existing particle and DMK-based filters [38]. Xiong et al. (2015) used RMSE to evaluate the performance of an optimal Kalman filter for state estimation of a quadrotor [39]. Thus, RMSE is the primary performance metric chosen for this work since the literature suggests it is a meaningful evaluator of observer performance.

### 3.4.1 Root Mean Square Error

RMSE is a fundamental metric that quantifies the average discrepancy between estimated values ( $\hat{x}$ ) and the corresponding ground truth values ( $x$ ) across a set of  $N$  samples. It is formulated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x_i)^2} \quad (3.10)$$

### 3.4.2 ArduPilot – A Framework for Testing

Ardupilot is an open-source software used for navigation and control of autonomous UAVs. The tool supports a vast range of vehicles, including copters, planes, rovers and boats [40]. In particular, ArduPilot uses what is known as ‘Replay logs’ to record all relevant information about a vehicle’s state and incoming sensor measurements throughout a flight [40]. These logs can then be ‘replayed’ to evaluate how a filter would have handled the incoming sensor data. Ultimately, they provide a means of testing new filters on real-world data without the risk of damaging any equipment. These replay logs will play a crucial role in this project.

Furthermore, the EKF3 (Extended Kalman Filter 3) is the state-of-the-art filter currently operating on ArduPilot used for state estimation. It interprets a suite of sensors, such as barometers, magnetometers, GNSS and airspeed sensors, to estimate the vehicle’s velocity, position, and attitude [41]. This filter will serve as the primary point of comparison for the filters to be proposed through this work.

## 4 Theoretical Results

---

### 4.1 Problem Formulation

We consider the state estimation problem for an unmanned aerial vehicle (UAV) with a body-fixed frame  $B$  and an inertial frame  $A$ . The state  $X \in \mathbf{SE}_2(3)$  comprises the rotation matrix  $R$ , position vector  $p$ , and velocity vector  $v$ , all measured with respect to  $A$ . Our UAV is equipped with a suite of sensors. A 3-axis accelerometer from the IMU measuring the linear acceleration,  $a(t)$ . A 3-axis gyroscope from the IMU, capturing the angular velocity,  $\Omega(t)$ . A GNSS receiver providing position and velocity measurements relative to the inertial frame  $A$ , with a delay of  $\tau$  seconds, denoted as  $y_p(t - \tau)$  and  $y_v(t - \tau)$ , respectively. A 3-axis magnetometer measuring the magnetic field,  $m(t)$ , in  $B$ . Further, we assume the reference magnetic field at any given latitude and longitude,  $m_0$  is known, as per the World Magnetic Model. The gravitational acceleration in the inertial frame is expressed as  $g \in \mathbb{R}^3$ . Since the measurement delays from the IMU sensors (accelerometers and gyroscopes) and magnetometer are significantly less impactful compared to those from GNSS, we treat the former as negligible. To further simplify our design, we will disregard the measurement noise in the magnetometer and GNSS readings.

Description	Sensor	Symbol	Units	Group
Angular acceleration	IMU	$\omega$	$\frac{rad}{s^2}$	$\mathbb{R}^3$
Linear acceleration	IMU	$a$	$\frac{m}{s^2}$	$\mathbb{R}^3$
Magnetic field strength	Magnetometer	$m$	T	$\mathbb{R}^3$
GNSS Position	GNSS	$y_p$	m	$\mathbb{R}^3$
GNSS Velocity	GNSS	$y_v$	$\frac{m}{s}$	$\mathbb{R}^3$

**Table 2:** System Inputs

The task then is to utilise the measurements  $\Omega(t)$ ,  $a(t)$ ,  $y_p(t - \tau)$ ,  $y_v(t - \tau)$  and  $m(t)$  to estimate the extended pose of the vehicle – namely the position, velocity and attitude.

Description	Group	Symbol	Units
Position	$\mathbb{R}^3$	$p$	$m$
Velocity	$\mathbb{R}^3$	$v$	$\frac{m}{s}$
Attitude	$\mathbf{SO}(3)$	$R$	rads

**Table 3:** System States

## 4.2 Complementary INS Filter Architecture

In solving the INS problem with a CINS observer, we must first understand its model architecture.

### 4.2.1 Model Dynamics

The dynamics of the INS problem can be represented as follows:

$$\dot{R} = R\Omega^\times \quad \dot{p} = v \quad \dot{v} = Ra + g$$

The definitions here are as defined in 2 where  $\dot{R} \in T_R \mathbf{SO}(3)$  represents its dynamics of the attitude. As shown in Barrau and Bonnabel (2017), since the INS dynamics are ‘group affine’ under the symmetry of  $\mathbf{SE}_2(3)$ , we can define the vehicle’s extended state as  $X \in \mathbf{SE}_2(3)$ .

$$X = \begin{bmatrix} R & V \\ 0_{2 \times 3} & I_2 \end{bmatrix}, \quad V = \begin{bmatrix} p & v \end{bmatrix}$$

Here  $V = \begin{bmatrix} p & v \end{bmatrix}$  represents an element of the tangent bundle  $T\mathbb{R}^3$  – that is, at a point  $x \in \mathbb{R}^3$  on the manifold,  $v \in T_x \mathbb{R}^3$  is the velocity at that point [14]. Van Goor et al. [14] showed that these dynamics could be manipulated by leveraging the left and right invariant properties of the extended pose group  $\mathbf{SE}_2(3)$  to arrive at the compact representation of state dynamics,  $\dot{X} \in T_X \mathbf{SE}_2(3)$ :

$$\dot{X} = X(U + D) + (G - D)X \tag{4.1}$$

$$U = \begin{pmatrix} \Omega^\times & 0 & a \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{pmatrix} \in \mathfrak{se}_2(3), \quad G = \begin{pmatrix} 0_{3 \times 3} & 0 & g \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{pmatrix} \in \mathfrak{se}_2(3),$$

$$D = \begin{pmatrix} 0_{3 \times 3} & 0_{3 \times 2} \\ 0_{2 \times 3} & S_D \end{pmatrix} \in \mathfrak{sim}_2(3), \quad S_D = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \in \mathfrak{gl}(2).$$

Consider the following observer design:

$$\dot{\hat{X}} = \hat{X}(U + D) + (G - D)\hat{X} + Z\Delta Z^{-1}\hat{X} \tag{4.2}$$

$$\dot{Z} = (G - D)Z + Z\Gamma \tag{4.3}$$

$$\Gamma = \begin{pmatrix} \Omega_\Gamma^\times & W_\Gamma \\ 0 & S_\Gamma \end{pmatrix} \in \mathfrak{sim}_2(3) \quad Z = \begin{pmatrix} R_Z & (p_Z, v_Z) \\ 0_{2 \times 3} & A_Z \end{pmatrix} \in \mathbf{SIM}_2(3) \quad \Delta = \begin{pmatrix} \Omega_\Delta^\times & w_\Delta \\ 0 & 0 \end{pmatrix} \in \mathfrak{se}_2(3)$$

Here,  $Z$  is an auxiliary state variable, with dynamics  $\dot{Z}$ . This auxillary state plays a crucial role in capturing various sources of estimation error, thereby enhancing estimation accuracy [14]. The choice of correction terms  $\Gamma$  and  $\Delta$  represents the significant theoretical focus of this research. The primary objective is the appropriate selection of these terms such that the system’s stability guarantees are preserved.

#### 4.2.2 Error Model

We define the observer error,  $\bar{E}$ , as follows:

$$\bar{E} = Z^{-1}X\hat{X}^{-1}Z = \begin{pmatrix} R_{\bar{E}} & V_{\bar{E}} \\ 0 & I_{2 \times 2} \end{pmatrix} \quad (4.4)$$

The key characteristic of this model of error is the conditional, which states that if  $\lim_{t \rightarrow \infty} \bar{E} = I$ , then it is simple to show that  $\lim_{t \rightarrow \infty} \hat{X} = X$  follows as a direct consequence. This property will be the objective of the stability analyses to come in the later sections.

Only a set of the degrees of freedom of the system defined in (4.2) and (4.3) are needed in this design. Namely, we let  $R_Z(0) = I_3$  and  $A_Z(0) = I_2$  and choose  $\Omega_\Gamma = 0$  and  $S_\Gamma = S_D$ . Given these choices, we can show  $\dot{R}_Z = 0$  and  $\dot{A}_Z = 0$ , allowing us to conclude that  $R_Z = I_3$  and  $A_Z = I_2 \forall t$  [14]. With these assumptions, we can use (4.4) to derive the following expressions for  $\bar{E}$  and  $\dot{\bar{E}}$ :

$$\begin{aligned} \bar{E} &= \begin{bmatrix} R\hat{R}^\top & V - V_Z - R_{\bar{E}}(\hat{V} - V_Z) \\ 0 & I_2 \end{bmatrix} \\ \dot{\bar{E}} &= \begin{bmatrix} -R_{\bar{E}}\Omega_\Delta^\times & V_E S_D + R_{\bar{E}}(W_\Gamma - W_\Delta) - W_\Gamma \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (4.5)$$

Then we see:

$$R_{\bar{E}} = R\hat{R}^\top \quad (4.6)$$

$$V_{\bar{E}} = V - V_Z - R_{\bar{E}}(\hat{V} - V_Z) \quad (4.7)$$

$$\dot{R}_{\bar{E}} = -R_{\bar{E}}\Omega_\Delta^\times \quad (4.8)$$

$$\dot{V}_E = V_E S_D + R_{\bar{E}}(W_\Gamma - W_\Delta) - W_\Gamma \quad (4.9)$$

The most noteworthy characteristic of the CINS filter is its  $\bar{E}$ -synchronous nature – that is, the behaviour of the error dynamics is entirely dependent on the choices of  $\Delta$  and  $\Gamma$  [14]. This characteristic offers two main advantages over more conventional nonlinear observer designs. Firstly, In the worst-case scenario where  $\Delta$  and  $\Gamma$  are set to zero, the error dynamics become  $\dot{\bar{E}} = 0$ , implying that the error dynamics will remain constant, but crucially, will not worsen.

Secondly, this feature affords the filter a ‘plug and play’ property for more straightforward integration. For a given sensor, one only needs to identify the appropriate correction terms  $\Delta$  and  $\Gamma$  that drive  $\bar{E} \rightarrow I_5$  as previously mentioned. Moreover, the filter’s linearity in combining  $\Delta$  and  $\Gamma$  from multiple sensors ensures that the performance either improves or remains stable. For instance, if we choose correction terms  $\Delta_1$  and  $\Gamma_1$  based on magnetometer readings, and  $\Delta_2$  and  $\Gamma_2$  to account for GNSS time delay, then combining  $\Delta = \Delta_1 + \Delta_2$  and  $\Gamma = \Gamma_1 + \Gamma_2$  with appropriate gain settings is likely to enhance the performance as compared to using either set of individual correction terms.

### 4.2.3 State Updates

Implementing this observer within a real-world, discretised system, requires numerically solving the associated ordinary differential equations (ODEs) – namely,  $\dot{Z}(t)$  and  $\dot{\hat{X}}(t)$  – for the purpose of advancing the state  $X(t)$  in time, subject to predefined initial conditions and control velocity inputs. Owing to the time-invariant nature of  $\dot{Z}(t)$ , its evolution in time can be solved with relative ease. More precisely, using Lemma (3.2), the state update for the ODE in (4.3) over an interval  $\delta t$  can be expressed as:

$$Z(t + \delta t) = \exp((G - D)\delta t)Z(t)\exp(\Gamma\delta t) \quad (4.10)$$

In contrast to the relatively straightforward evolution of  $\dot{Z}(t)$ , the dynamics of  $\dot{\hat{X}}$  is a time-variant function, depending on the control velocity input  $U(t)$ . This prohibits a simple analytical solution. Thus, a series of approximations are made to numerically solve this ODE. Specifically, given the high sampling rate of the MEMS IMU, each sampled velocity  $U(t)$  is treated as constant from the time it is made to the time of the next IMU measurement. This assumption facilitates the forward integration of the  $\dot{\hat{X}}$  dynamics and enables the computation of the update step over a time interval  $\delta t$ . Firstly we rearrange  $\dot{\hat{X}}$  from (4.2) as

$$\begin{aligned} \dot{\hat{X}} &= \hat{X}(U + D) + (G - D)\hat{X} + Z\Delta Z^{-1}\hat{X} \\ &= \hat{X}(U + D) + (G - D + Z\Delta Z^{-1})\hat{X}. \end{aligned}$$

Then we apply Lemma (3.2) to the equation above since  $U$  is assumed as constant for an interval of  $\delta t$ .

$$X(t + \delta t) = \exp((G - D + Z\Delta Z^{-1})\delta t)\hat{X}(t)\exp((U + D)\delta t) \quad (4.11)$$

The state update equations for  $\hat{X}$  and  $Z$ , given above in (4.10) and (4.11), define the forward propagation of the state variables over time and are therefore integral to the numerical implementation of CINS.

Now that the fundamental characteristics of the CINS architecture have been established, we seek to solve the INS problem by combining the measurements from 2. More specifically, this involves determining the corresponding correction terms,  $\Delta$  and  $\Gamma$ , that preserve the locally exponential and almost-globally asymptotic stability guarantees of the system. The problem breaks down into 3 key parts:

1. **Magnetometer Correction:** Using the magnetometer measurement  $m(t)$  and the reference magnetic field vector,  $m_0$ , to find the correction terms  $\Delta$  and  $\Gamma$  that preserve stability.
2. **GNSS Position and Velocity Delay Correction:** Using the delayed GNSS position and velocity measurement,  $y_p(t - \tau)$  and  $y_v(t - \tau)$  respectively, and a buffer of IMU measurements over the last  $\tau$  seconds, to find the correction terms  $\Delta$  and  $\Gamma$  that preserve stability.
3. **Implementation:** A brief discussion of the implementation of the proposed corrections to CINS.

## 4.3 Magnetometer Correction

### 4.3.1 Problem Formulation

Let  $m_0, m, \hat{m} \in \mathbb{R}^3$  be the reference, measured and estimated magnetic field measurements, respectively. These parameters are all measured relative to the inertial reference frame. Then

$$m = R^\top m_0 \quad (4.12)$$

$$\hat{m} = \hat{R}^\top m_0 \quad (4.13)$$

### 4.3.2 Stability of Translational Error

Magnetometer measurements are a directional measurement that helps in correcting the attitude. It does not contribute any information on the translational estimates of the state, and so  $\dot{V}_{\bar{E}} \rightarrow 0$  does not necessarily hold.

### 4.3.3 Lyapunov Stability of Attitude Error

The observers pose dynamics,  $\dot{\hat{R}}$ , are given by:

$$\dot{\hat{R}} = \hat{R}\Omega^\times + \Omega_\Delta^\times \hat{R}$$

**Lemma 4.1.** Consider the Lyapunov candidate  $\mathcal{L} = \frac{1}{2}|R_{\bar{E}} - I|^2$ . Define  $k_m \in \mathbb{R}$  as a scalar gain and the correction term,  $\Omega_\Delta^\times$  as:

$$\Omega_\Delta^\times = k_m ((\hat{R}m)^\times m_0)^\times. \quad (4.14)$$

Then, the system is Lyapunov stable.

*Proof.* To begin, we rearrange the equation (4.14) to be expressed in terms of  $R_{\bar{E}}$ .

$$\begin{aligned} \Omega_\Delta^\times &= k_m ((\hat{R}m)^\times m_0)^\times \\ &= k_m ((\hat{R}R^\top m_0)^\times m_0)^\times \\ &= k_m ((R_{\bar{E}}^T m_0)^\times m_0)^\times \end{aligned}$$

Recall the error dynamics of the attitude estimate (4.8). Substitute the expression for  $\Omega_\Delta^\times$ :

$$\begin{aligned} \dot{R}_{\bar{E}} &= -R_{\bar{E}}(k_m(R_{\bar{E}}^T m_0)^\times m_0)^\times \\ &= -k_m \cdot R_{\bar{E}}((R_{\bar{E}}^T m_0)^\times m_0)^\times. \end{aligned}$$

Manipulating the proposed Lyapunov candidate, we see:

$$\mathcal{L} = \frac{1}{2}|R_{\bar{E}} - I|^2$$

$$\begin{aligned}
&= \frac{1}{2} \langle R_{\bar{E}} - I, R_{\bar{E}} - I \rangle \\
&= \frac{1}{2} \text{tr} ((R_{\bar{E}} - I)^T (R_{\bar{E}} - I)) \\
&= \frac{1}{2} \text{tr} (R_{\bar{E}}^T R_{\bar{E}} - R_{\bar{E}}^T - R_{\bar{E}} + I) \\
&= \frac{1}{2} \text{tr} (2I - 2R_{\bar{E}}) \\
&= \text{tr}(I - R_{\bar{E}}).
\end{aligned}$$

The time derivative of this Lyapunov function is:

$$\begin{aligned}
\dot{\mathcal{L}} &= \frac{d}{dt} (\text{tr}(I - R_{\bar{E}})) \\
&= \text{tr}(-\dot{R}_{\bar{E}}) \\
&= \text{tr}(kR_{\bar{E}}((R_{\bar{E}}^T m_0)^\times m_0)^\times) \\
&= k_m \text{tr}(R_{\bar{E}}(m_0(R_{\bar{E}}^T m_0)^T - R_{\bar{E}}^\top m_0 m_0^T)) \\
&= k_m \text{tr}(R_{\bar{E}} m_0 m_0^T R_{\bar{E}} - R_{\bar{E}} R_{\bar{E}}^\top m_0 m_0^T) \\
&= k_m \text{tr}(R_{\bar{E}}^2 m_0 m_0^T - m_0 m_0^T) \\
&= k_m \text{tr}((R_{\bar{E}}^2 - I)m_0 m_0^T) \\
&= k_m \text{tr}(m_0^T (R_{\bar{E}}^2 - I)m_0) \\
&= k_m \text{tr}(m_0^T \mathbb{P}_{\text{skew}}(R_{\bar{E}}^2 - I)m_0) \\
&= -k_m m_0^T \mathbb{P}_{\text{skew}}(I - R_{\bar{E}}^2)m_0 \\
&= -k_m m_0^T \frac{1}{2} ((I - R_{\bar{E}}^2) + (I - R_{\bar{E}}^2)^T)m_0 \\
&= -\frac{k_m}{2} m_0^T (2I - R_{\bar{E}}^2 - R_{\bar{E}}^{T^2})m_0 \\
&= -\frac{k_m}{2} m_0^T ((R_{\bar{E}}^T - R_{\bar{E}})^T (R_{\bar{E}}^T - R_{\bar{E}}))m_0 \\
&= -\frac{k_m}{2} ((R_{\bar{E}}^T - R_{\bar{E}})m_0)^T ((R_{\bar{E}}^T - R_{\bar{E}})m_0) \\
&= -\frac{k_m}{2} |(R_{\bar{E}} - R_{\bar{E}}^T)m_0|^2 \leq 0
\end{aligned}$$

■

This concludes the Lyapunov stability analysis for the magnetometer attitude correction. Based on the approach shown in [14] for a similar problem, it can be shown that almost-global asymptotic and locally exponential stability follows and that  $\lim_{t \rightarrow \infty} \hat{X} \rightarrow X$ .

## 4.4 GNSS Position and Velocity Time Delay Correction

### 4.4.1 Problem Formulation

For simplicity, the notation  $f(t) = f_t$  for signal  $f$  at time  $t$  will be used interchangeably. This will particularly apply for  $X(t), U(t), y_p(t - \tau)$  and  $y_v(t - \tau)$ . For a system with dynamics (4.1) and delay  $\tau$ , denote a generic 3-axis translational measurement (GNSS position or velocity) as  $y_\alpha(t - \tau) \in \mathbb{R}^3$ . Additionally, we also consider a generic standard basis vector of two-dimensional Euclidean space as  $\mathbf{e}_\alpha \in \mathbb{R}^2$ . The value of  $y_\alpha(t - \tau)$  and  $\mathbf{e}_\alpha$  will vary depending on whether a GNSS position or GNSS velocity measurement is being considered. Furthermore, with an assumption of no noise in the delayed GNSS measurement, we may consider a measurement of position or velocity from the GNSS as the true, delayed value. Thus we defined a combined measurement vector  $\mathbf{y}_\alpha(t - \tau)$ , that relates the GNSS measurement to the delayed state,  $X_{t-\tau}$ , via  $\bar{\mathbf{O}}_\alpha = \begin{pmatrix} 0_{3 \times 1} & \mathbf{e}_\alpha \end{pmatrix}^\top$

$$\mathbf{y}_\alpha(t - \tau) = \begin{pmatrix} y_\alpha(t - \tau) \\ \mathbf{e}_\alpha \end{pmatrix} = X_\tau \bar{\mathbf{O}}_\alpha \quad (4.15)$$

We define delay matrices,  $Y_L(t, \tau), Y_R(t, \tau) \in \mathbf{SIM}_2(3)$ , in order to represent the impact of a delay  $\tau$  on the estimated state of a vehicle. As a general notation for this section, suppose that  $Y_L^\tau = Y_L(t, \tau)$  and  $Y_R^\tau = Y_R(t, \tau)$  where these are the left and right delay matrices with variables,  $t$ , and  $\tau$  as inputs for time and delay respectively. This notation applies only to  $Y_L$  and  $Y_R$ .

In order to understand how these terms evolve with both time and delay, we consider both the partial derivative of both  $Y_L^\tau$  and  $Y_R^\tau$  with respect to the delay,  $\tau$  and then consider the time derivative. Beginning with the former, we propose the following partial derivatives and initial conditions for  $Y_L$  and  $Y_R$ :

$$\frac{\partial}{\partial \tau} Y_R^\tau = (U_{t-\tau} + D) Y_R^\tau, \quad Y_R^0 = I_5, \quad (4.16)$$

$$\frac{\partial}{\partial \tau} Y_L^\tau = Y_L^\tau (G - D), \quad Y_L^0 = I_5 \quad (4.17)$$

**Lemma 4.2** (Delayed State Transformation). Consider the state dynamics,  $\dot{X}$ , from (4.1). Consider the partial derivatives of  $Y_L^\tau$  and  $Y_R^\tau$  with respect to  $\tau$  along with their respective initial conditions as given by (4.16), (4.17). Fix  $t \geq 0$ , with  $\tau \in [0, t]$ . Then

$$X_t = Y_L^\tau X_{t-\tau} Y_R^\tau \quad (4.18)$$

*Proof.* We start by showing the equivalence of the partial derivatives with respect to  $\tau$ . Starting with the right-hand side:

$$\frac{\partial}{\partial \tau} (Y_L^\tau X_{t-\tau} Y_R^\tau) = \frac{\partial}{\partial \tau} (Y_L^\tau) X_{t-\tau} Y_R^\tau + Y_L^\tau \frac{\partial}{\partial \tau} (X_{t-\tau}) Y_R^\tau + Y_L^\tau X_{t-\tau} \frac{\partial}{\partial \tau} (Y_R^\tau)$$

Given  $\frac{\partial}{\partial \tau} X_{t-\tau} = -\dot{X}_{t-\tau}$  by the chain rule, and using (4.17) and (4.16), we get:

$$\begin{aligned}
\frac{\partial}{\partial \tau} (Y_L^\tau X_{t-\tau} Y_R^\tau) &= Y_L^\tau (G - D) X_{t-\tau} Y_R^\tau - Y_L^\tau \dot{X}_{t-\tau} Y_R^\tau + Y_L^\tau X_{t-\tau} (U_{t-\tau} + D) Y_R^\tau \\
&= Y_L^\tau (G - D) X_{t-\tau} Y_R^\tau - Y_L^\tau (X_{t-\tau} (U_{t-\tau} + D) + (G - D) X_{t-\tau}) Y_R^\tau + Y_L^\tau X_{t-\tau} (U_{t-\tau} + D) Y_R^\tau \\
&= Y_L^\tau G X_{t-\tau} Y_R^\tau - Y_L^\tau D X_{t-\tau} Y_R^\tau - Y_L^\tau X_{t-\tau} U_{t-\tau} Y_R^\tau - Y_L^\tau X_{t-\tau} D Y_R^\tau \\
&\quad - Y_L^\tau G X_{t-\tau} Y_R^\tau + Y_L^\tau D X_{t-\tau} Y_R^\tau + Y_L^\tau X_{t-\tau} U_{t-\tau} Y_R^\tau + Y_L^\tau X_{t-\tau} D Y_R^\tau \\
&= 0
\end{aligned}$$

It is straightforward to see that  $\frac{\partial}{\partial \tau} X_t = 0$ . Thus we have shown the partial derivatives match.

For the initial conditions, it's clear to see that

$$X_t = X(t-0) = IX(t-0)I = Y_L^0 X_{t-0} Y_R^0. \quad (4.19)$$

Thus, the initial conditions for both sides of (4.18) are consistent. Since the initial conditions and the partial derivatives agree, we may invoke the uniqueness of ODE solutions to conclude that (4.18) holds.  $\blacksquare$

Now we show the propagation of  $Y_L^\tau$  and  $Y_R^\tau$  in time. We propose the following time derivative for  $Y_R^\tau$ :

$$\frac{d}{dt} Y_R^\tau = -(U_t + D) Y_R^\tau + Y_R^\tau (U_{t-\tau} + D) \quad (4.20)$$

**Lemma 4.3** (Time Derivative of Delay Matrices). Consider the state dynamics,  $\dot{X}$ , from (4.1). For a fixed time delay,  $\tau \geq 0$ , we can show that the time derivative for  $Y_R^\tau$  is given by (4.20) and that  $Y_L^\tau$  is constant as follows

$$Y_L^\tau = \exp(\tau(G - D)) \quad (4.21)$$

*Proof.* The left delay matrix,  $Y_L^\tau$ , can be explicitly solved using the definition from (4.16) and applying Lemma (3.1). We see that it is constant with respect to time,  $t$ .

$$Y_L(t, \tau) = \exp(\tau(G - D)) \quad (4.22)$$

A useful property of such a matrix exponential is that we get commutativity of  $Y_L^\tau$

$$(G - D) Y_L^\tau = Y_L^\tau (G - D) \quad (4.23)$$

For the time derivative of the right delay matrix,  $Y_R^\tau$ , we use the definition from (4.18) to rearrange for  $Y_R^\tau$

$$Y_R^\tau = X_t^{-1} Y_L^{-1} X_{t-\tau}$$

Taking the time derivative and using the property (4.23) from above

$$\begin{aligned}
\frac{d}{dt} Y_R^\tau &= \frac{d}{dt} (X_t^{-1} Y_L^{-1} X_{t-\tau}) \\
&= \frac{d}{dt} (X_t^{-1}) Y_L^{\tau-1} X_{t-\tau} + X_t^{-1} \frac{d}{dt} (Y_L^{\tau-1}) X_{t-\tau} + X_t^{-1} Y_L^{\tau-1} \frac{d}{dt} (X_{t-\tau}) \\
&= [-X_t^{-1}((G-D)X_t + X_t(U_t+D))X_t^{-1}] Y_L^{\tau-1} X_{t-\tau} \\
&\quad + X_t^{-1} Y_L^{\tau-1} [(G-D)X_{t-\tau} + X_{t-\tau}(U_{t-\tau}+D)] \\
&= -[X_t^{-1}(G-D) + (U_t+D)X_t^{-1}] Y_L^{\tau-1} X_{t-\tau} \\
&\quad + X_t^{-1} Y_L^{\tau-1} [(G-D)X_{t-\tau} + X_{t-\tau}(U_{t-\tau}+D)] \\
&= -X_t^{-1}(G-D)Y_L^{\tau-1} X_{t-\tau} - (U_t+D)X_t^{-1} Y_L^{\tau-1} X_{t-\tau} \\
&\quad + X_t^{-1} Y_L^{\tau-1} (G-D)X_{t-\tau} + X_t^{-1} Y_L^{\tau-1} X_{t-\tau} (U_{t-\tau}+D) \\
&= -(U_t+D)Y_R^\tau + Y_R^\tau (U_{t-\tau}+D)
\end{aligned}$$

■

Now we simply calculate the explicit solutions and the Euler integration terms for the differential equations above defined by  $\frac{\partial}{\partial t} Y_R^\tau$  and  $\frac{d}{dt} Y_R^\tau$ . Here, we temporarily use the complete (non-shortened) notation – namely,  $Y_R(t, \tau)$  instead of  $Y_R^\tau$ . First we apply Lemma (3.2) to (4.16) at a fixed time,  $t$ , with a step of  $\Delta\tau$  in the delay,  $\tau$ , and then do the same for (4.16), but with a fixed delay,  $\tau$ , with step of  $\Delta t$  in time,  $t$ :

$$Y_R(t, \tau + \Delta\tau) = \exp(\Delta\tau(U_{t-\tau} + D))Y_R(t, \tau) \quad (4.24)$$

$$Y_R(t + \Delta t, \tau) = \exp[-\Delta t(U_{t-\tau} + D)] \cdot Y_R(t, \tau) \cdot \exp[\Delta t(U_t + D)] \quad (4.25)$$

These results will be used throughout the stability analysis.

#### 4.4.2 Stability of Translational Error

Having established the delayed system model, we can now perform a stability analysis on the translational error,  $V_{\bar{E}}$ . As part of this analysis, we require a series of preliminary results, which will be explored through the following investigation, before finally proving the stability for both the GNSS position and velocity time delay corrections.

Recall the transformation of the state from delayed to present time given in (4.18). It is simple to see that this expression may be rearranged to  $X_{t-\tau} = Y_L^{\tau-1} X_t Y_R^{\tau-1}$ . Combining this expression with (4.15) we get

$$Y_L^\tau \mathbf{y}_\alpha(t - \tau) = X_t (Y_R^\tau)^{-1} \bar{\mathbf{O}}_\alpha \quad (4.26)$$

Define  $\mu_\alpha, \dot{\mu}_\alpha \in \mathbb{R}^3, C_\alpha, \dot{C}_\alpha \in \mathbb{R}^2$  where

$$\begin{pmatrix} \mu_\alpha & C_\alpha \end{pmatrix}^\top = Y_L^\tau \mathbf{y}_\alpha(t - \tau) \quad (4.27)$$

$$\begin{pmatrix} \dot{\mu}_\alpha & \dot{C}_\alpha \end{pmatrix}^\top = (Y_R^\tau)^{-1} \bar{\mathbf{O}}_\alpha \quad (4.28)$$

Substituting these definitions along with (4.2.1) into (4.26) we get

$$\begin{pmatrix} \mu_\alpha \\ C_\alpha \end{pmatrix} = \begin{pmatrix} R_t & V_t \\ 0 & I \end{pmatrix} \begin{pmatrix} \dot{\mu}_\alpha \\ \dot{C}_\alpha \end{pmatrix}$$

Therefore, we obtain  $C_\alpha = \dot{C}_\alpha$  and consequently

$$V_t C_\alpha = \mu_\alpha - R_t \dot{\mu}_\alpha, \quad (4.29)$$

which is a result that we will use later.

Now, we seek to compute an explicit solution for  $C_\alpha$  in terms of known values. We begin by expanding the explicit form of  $Y_L^\tau$  from (4.22). Suppose  $Q = \tau(G - D)$ . Compute  $Q$  and  $Q^2$  using  $G$  and  $D$  as defined in Section 3:

$$Q = \begin{pmatrix} 0_{3 \times 3} & 0_{3 \times 1} & g\tau \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & -\tau & 0 \end{pmatrix} \quad (4.30) \qquad Q^2 = \begin{pmatrix} 0_{3 \times 3} & -g\tau^2 & 0_{3 \times 1} \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{pmatrix} \quad (4.31)$$

$Q^3$  can be shown to be the zero  $3 \times 3$  matrix, meaning that  $\forall k, k > 3, Q^k = 0$ . Combining all terms into the matrix exponential power series expansion, we get:

$$\exp(Q) = Y_L^\tau = I + \begin{pmatrix} 0_{3 \times 3} & 0_{3 \times 1} & g\tau \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & -\tau & 0 \end{pmatrix} + \begin{pmatrix} 0_{3 \times 3} & -g\tau^2 & 0_{3 \times 1} \\ 0_{1 \times 3} & 0 & 0 \\ 0_{1 \times 3} & 0 & 0 \end{pmatrix} = \begin{pmatrix} I_{3 \times 3} & -\frac{g\tau^2}{2} & g\tau \\ 0_{1 \times 3} & 1 & 0 \\ 0_{1 \times 3} & -\tau & 1 \end{pmatrix} \quad (4.32)$$

Generally define  $Y_L^\tau \in \mathbf{SIM}_2(3)$  as

$$\begin{pmatrix} R_L & V_L \\ 0_{2 \times 3} & A_L \end{pmatrix} \quad (4.33)$$

Then by comparison, we see:

$$V_L = \begin{pmatrix} -\frac{g\tau^2}{2} & g\tau \end{pmatrix} \quad (4.34) \qquad A_L = \begin{pmatrix} 1 & 0 \\ -\tau & 1 \end{pmatrix} \quad (4.35)$$

Leveraging the definition in (4.27)

$$\begin{pmatrix} \mu_\alpha \\ C_\alpha \end{pmatrix} = \begin{pmatrix} R_L & V_L \\ 0_{2 \times 3} & A_L \end{pmatrix} \begin{pmatrix} 0_{3 \times 1} \\ \mathbf{e}_\alpha \end{pmatrix}$$

Finally, we see

$$C_\alpha = A_L \mathbf{e}_\alpha \quad (4.36)$$

**Lemma 4.4.** Consider the dynamics  $\dot{X}$  (4.1), translational error  $V_{\bar{E}}$  (4.7) and its dynamics  $\dot{V}_{\bar{E}}$  (4.9).

Furthermore, recall the definition of  $C_\alpha$  (4.36) and  $V_t C$  (4.29). The shorthand  $V$  will be used in place of  $V_t$ . Define  $L = \begin{pmatrix} l_1 & l_2 \end{pmatrix}$ , where  $l_1, l_2 \in \mathbb{R}$ . Choose

$$W_\Delta = (\mu_\alpha - \hat{R}\dot{\mu}_\alpha - \hat{V}C_\alpha)L \quad (4.37)$$

$$W_\Gamma = (\mu_\alpha - V_Z C_\alpha)L \quad (4.38)$$

Then we can show

$$\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - CL) \quad (4.39)$$

*Proof.* Post multiplying the expression for  $V_{\bar{E}}$  (4.29) with  $C_\alpha$  and substituting (4.29) we get:

$$\begin{aligned} V_{\bar{E}}C_\alpha &= VC - V_Z C - R_{\bar{E}}(\hat{V} - V_Z)C_\alpha, \\ &= \mu_\alpha - R\dot{\mu}_\alpha - V_Z C_\alpha - \hat{V}C_\alpha + \hat{R}V_Z C_\alpha. \end{aligned}$$

Post multiplying by  $L$  and using the definition of  $R_{\bar{E}}$  from (4.6):

$$V_{\bar{E}}C_\alpha L = \mu_\alpha L - R_{\bar{E}}\hat{R}\dot{\mu}_\alpha L - V_Z C_\alpha L - \hat{V}C_\alpha L + \hat{R}V_Z C_\alpha L.$$

We will use this result shortly. Next, we focus on  $\dot{V}_{\bar{E}}$  and substitute the choices of  $W_\Delta$  and  $W_\Gamma$ :

$$\begin{aligned} \dot{V}_{\bar{E}} &= V_{\bar{E}}S_D + R_{\bar{E}}(\mu_\alpha L - R_{\bar{E}}\hat{R}\dot{\mu}_\alpha L - V_Z C_\alpha L - \hat{V}C_\alpha L + \hat{R}V_Z C_\alpha L) \\ &= V_{\bar{E}}S_D + R_{\bar{E}}\mu_\alpha L - \mu_\alpha L + V_Z C_\alpha L. \end{aligned}$$

Considering the sum  $\dot{V}_{\bar{E}} + V_{\bar{E}}C_\alpha L$  and rearranging, we finally obtain:

$$\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - C_\alpha L).$$

■

Now, we have the necessary framework to consider the stability analysis for the GNSS position and velocity corrections.

**Lemma 4.5.** Consider the matrix  $C_\alpha$  (4.36). For the GNSS position we define  $C_p = A_L \mathbf{e}_1$ . For the GNSS velocity, we define  $C_v = A_L \mathbf{e}_2$ . Choose  $L_p = \begin{pmatrix} l_{p,1} & l_{p,2} \end{pmatrix}$  and  $L_v = \begin{pmatrix} l_{v,1} & l_{v,2} \end{pmatrix}$  such that  $l_{p,1} + l_{v,2} > \tau l_{p,2}$  and  $l_{p,1}l_{v,2} > l_{p,2}(l_{v,1} - 1)$ . Consider also the expression for  $\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - C_\alpha L)$  from (4.39). For the design where we combine the GNSS position and velocity corrections together, we see  $\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - C_p L_p - C_v L_v)$ . Then we aim to show  $\lim_{t \rightarrow \infty} V_{\bar{E}} = 0$ .

*Proof.* Firstly, let's transform the system  $\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - C_p L_p - C_v L_v)$  to a form where the Routh-Hurwitz criterion is applicable. Consider the equivalent system  $\dot{V}_{\bar{E}} = -V_{\bar{E}}(C_p L_p - C_v L_v - S_D)$  where

$C_p = \begin{pmatrix} 1 & -\tau \end{pmatrix}^\top$  and  $C_v = \begin{pmatrix} 0 & 1 \end{pmatrix}^\top$ . Expanding  $C_p L_p - C_v L_v - S_D$

$$C_p L_p - C_v L_v - S_D = \begin{pmatrix} l_{p,1} & l_{p,2} \\ l_{v,1} - \tau l_{p,1} - 1 & l_{v,2} - \tau l_{p,2} \end{pmatrix}$$

With relative simplicity, it can be shown that the trace is given by  $\text{tr}(C_p L_p - C_v L_v - S_D) = l_{p,1} + l_{p,2} - \tau l_{p,2}$  and the determinant by  $\det(C_p L_p - C_v L_v - S_D) = l_{p,1} l_{v,2} - l_{p,2} (l_{v,1} - 1)$ . To apply the Routh-Hurwitz criterion for stability, we need to ensure: i)  $\det(C_p L_p - C_v L_v - S_D) > 0$  and ii)  $\text{tr}(C_p L_p - C_v L_v - S_D) > 0$ . Therefore we require

$$\begin{aligned} l_{p,1} + l_{p,2} - \tau l_{p,2} &> 0, \\ l_{p,1} l_{v,2} - l_{p,2} (l_{v,1} - 1), \end{aligned}$$

Which can be easily rearranged to show consistency with the choices of  $l_{p,1}, l_{p,2}, l_{v,1}, l_{v,2}$ . Thus, by the Routh-Hurwitz criterion, the system  $\dot{V}_{\bar{E}} = -V_{\bar{E}}(C_p L_p - C_v L_v - S_D)$  is stable, implying asymptotic stability for the original system  $\dot{V}_{\bar{E}} = V_{\bar{E}}(S_D - C_p L_p - C_v L_v)$  and hence,  $\lim_{t \rightarrow \infty} V_{\bar{E}} = 0$  ■

Thus, under the choices of  $W_\Delta$ ,  $W_\Gamma$ ,  $L$ , and  $C_\alpha$ , it has been shown that  $V_{\bar{E}}$  is asymptotically stable for the GNSS position and velocity correction. This concludes the translational stability analysis.

#### 4.4.3 Lyapunov Stability of Attitude Error

For the attitude stability analysis, we aim to show  $\lim_{t \rightarrow \infty} R_{\bar{E}} = I_3$  with an appropriate selection of the correction term  $\Omega_\Delta$ . We use the same Lyapunov candidate as in Section 4.3 – namely,  $\mathcal{L} = \frac{1}{2}|\bar{R} - I|^2$  – and aim to show that  $\dot{\mathcal{L}} \leq 0$ .

We begin by considering (4.29) and rearranging to show  $\mu_\alpha = VC_\alpha + R\dot{\mu}_\alpha$ . We now propose two additional definitions that follow a similar structure

$$\hat{\mu}_\alpha = \hat{V}C_\alpha + \hat{R}\dot{\mu}_\alpha \quad (4.40)$$

$$\mu_{Z,\alpha} = V_Z C_\alpha \quad (4.41)$$

Using these definitions, consider the term  $\mu_\alpha - \mu_{Z,\alpha}$

$$\begin{aligned} \mu_\alpha - \mu_{Z,\alpha} &= \dot{\mu}_\alpha + V_{C_\alpha} - \mu_{Z,\alpha} \\ &= R_{\bar{E}} \hat{R} \dot{\mu}_\alpha + V_{C_\alpha} - \mu_{Z,\alpha} \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \hat{V}C_\alpha) + V_{C_\alpha} - \mu_{Z,\alpha} \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \mu_{Z,\alpha}) + R_{\bar{E}} V_Z C_\alpha - R_{\bar{E}} \hat{V} C_\alpha + V_{C_\alpha} - \mu_{Z,\alpha} \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \mu_{Z,\alpha}) - R_{\bar{E}} (\hat{V} C_\alpha - V_Z C_\alpha) + V_{C_\alpha} - \mu_{Z,\alpha} \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \mu_{Z,\alpha}) - R_{\bar{E}} (\hat{V} C_\alpha - V_Z C_\alpha) + V_{C_\alpha} - V_Z C_\alpha \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \mu_{Z,\alpha}) + (V - V_Z - R_{\bar{E}} (\hat{V} - V_Z)) C_\alpha \\ &= R_{\bar{E}} (\hat{\mu}_\alpha - \mu_{Z,\alpha}) + V_{\bar{E}} C_\alpha \end{aligned}$$

Based on the proof from Section 4.4.2 which showed that  $V_{\bar{E}}$  is asymptotically stable, we make the approximation that  $V_{\bar{E}} = 0$  for the purpose of this analysis. Therefore we get

$$\mu_\alpha - \mu_{Z,\alpha} \approx R_{\bar{E}}(\hat{\mu}_\alpha - \mu_{Z,\alpha}) \quad (4.42)$$

**Lemma 4.6.** Define the correction term,  $\Omega_\Delta = k_\alpha(\hat{\mu}_\alpha - \mu_{Z,\alpha})^\times(\mu_\alpha - \mu_{Z,\alpha})$ , where  $k_\alpha \in \mathbb{R}$  is a scalar gain to be chosen during implementation. Consider also the definition for  $\mu_\alpha - \mu_{Z,\alpha}$  from (4.42) with a Lyapunov candidate  $\mathcal{L} = \frac{1}{2}|R_{\bar{E}} - I|^2$ . Then, we can show that the system  $\dot{\mathcal{L}} \leq 0$ .

*Proof.* We firstly simplify the correction term  $\Omega_\Delta$  using (4.42) and letting  $u = \hat{\mu}_\alpha - \mu_{Z,\alpha}$ :

$$\Omega_\Delta = k_\alpha(\hat{\mu}_\alpha - \mu_{Z,\alpha})^\times(\mu_\alpha - \mu_{Z,\alpha}) \quad (4.43)$$

$$\approx k_\alpha(\hat{\mu}_\alpha - \mu_{Z,\alpha})^\times(R_{\bar{E}}(\hat{\mu}_\alpha - \mu_{Z,\alpha})) \quad (4.44)$$

$$\Omega_\Delta \approx k_\alpha u^\times R_{\bar{E}} u \quad (4.45)$$

Recall  $\dot{R}_{\bar{E}} = -R_{\bar{E}}\Omega_\Delta^\times$  from (4.8). As seen in Lemma 4.1, the Lyapunov candidate preserves the following equality and time derivative, respectively:

$$\mathcal{L} = \frac{1}{2}|R_{\bar{E}} - I|^2 = \text{tr}(I - R_{\bar{E}}) \quad (4.46)$$

$$\dot{\mathcal{L}} = \text{tr}(-\dot{R}_{\bar{E}}) \quad (4.47)$$

Evaluating  $\dot{\mathcal{L}}$  by substituting the correction term  $\Omega_\Delta$  into  $\dot{R}_{\bar{E}}$ , we see

$$\begin{aligned} \dot{\mathcal{L}} &= \text{tr}(-\dot{R}_{\bar{E}}) \\ &= \text{tr}(-R_{\bar{E}}\Omega_\Delta^\times) \\ &\approx \text{tr}(R_{\bar{E}}(k_\alpha u^\times R_{\bar{E}} u)^\times) \\ &\approx k_\alpha \text{tr}(R_{\bar{E}}(u^\times R_{\bar{E}} u)^\times) \\ &\approx k_\alpha \text{tr}(R_{\bar{E}}(R_{\bar{E}} u u^\top - u(R_{\bar{E}} u)^\top)) \\ &\approx k_\alpha \text{tr}(R_{\bar{E}}^2 u u^\top - R_{\bar{E}} u u^\top R_{\bar{E}}^\top) \\ &\approx k_\alpha \text{tr}(R_{\bar{E}}^2 u u^\top - u u^\top) \\ &\approx k_\alpha \text{tr}((R_{\bar{E}}^2 - I) u u^\top) \\ &\approx k_\alpha \text{tr}(u^\top (R_{\bar{E}}^2 - I) u) \\ &\approx k_\alpha \text{tr}(u^\top \mathbb{P}_{\text{skew}}(R_{\bar{E}}^2 - I) u) \\ &\approx -k_\alpha u^\top \mathbb{P}_{\text{skew}}(I - R_{\bar{E}}^2) u \\ &\approx -k_\alpha u^\top \frac{1}{2}((I - R_{\bar{E}}^2) + (I - R_{\bar{E}}^2)^T) u \\ &\approx -\frac{k_\alpha}{2} u^\top (2I - R_{\bar{E}}^2 - R_{\bar{E}}^{T2}) u \\ &\approx -\frac{k_\alpha}{2} u^\top ((R_{\bar{E}}^T - R_{\bar{E}})^T (R_{\bar{E}}^T - R_{\bar{E}})) u \\ &\approx -\frac{k_\alpha}{2} ((R_{\bar{E}}^T - R_{\bar{E}}) u)^T ((R_{\bar{E}}^T - R_{\bar{E}}) u) \\ &\approx -\frac{k_\alpha}{2} |(R_{\bar{E}} - R_{\bar{E}}^T) u|^2 \leq 0 \end{aligned}$$

■

This concludes the Lyapunov stability analysis for the GNSS position and velocity time delay correction. Based on the approach shown in [14] for a similar problem, it can be shown that almost-global asymptotic and locally exponential stability follows and that  $\lim_{t \rightarrow \infty} \hat{X} \rightarrow X$ .

## 4.5 Implementation

The developments in Section 4.3 and 4.4 provide the necessary framework for the practical implementation of the proposed corrections in the CINS filter. Here, we outline the high-level approach for transferring the theoretical results into a function version of CINS. The implementation was done in Python, and the input data was read from ArduPilot replay logs.

### 4.5.1 MAG: Magnetometer Correction Filter

The first filter (MAG) uses the magnetometer correction to aid in its attitude estimation based on the correction terms derived earlier. Below is the algorithm block that outlines the way in which the filter was implemented.

---

#### Algorithm 1 MAG

---

```

1: Initialise estimator with initial estimates  $\hat{X}$ ,  $\hat{Z}$  and gains
2: Load ArduPilot replay log
3: Initialise magnetometer and GPS storage variables
4: for each message in the replay log do
5:   if message is GPS data with 3D fix then
6:     Store GPS position and velocity in storage variables
7:   else if message is magnetometer data then
8:     Store magnetometer measurement in storage variable
9:   else if message is IMU data then
10:    Compute corrections using the latest GPS data
11:    Compute corrections using the latest magnetometer data
12:    Integrate dynamics with IMU data and computed corrections
13:    Update state estimate  $\hat{X}$  and auxiliary state  $\hat{Z}$ 
14:  end if
15: end for
```

---

### 4.5.2 DELPV: GNSS Time Delay Corrected Filter

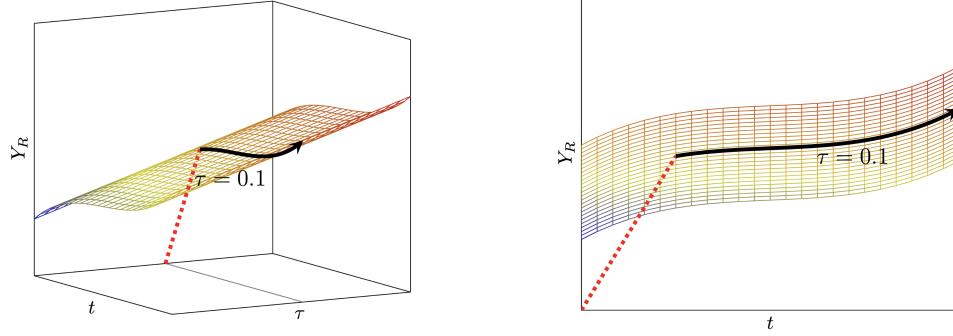
The DELPV filter incorporates the GNSS time delay corrections for both position and velocity into CINS. Note that this filter did not include the magnetometer correction.

It should be emphasized that, in contrast to the magnetometer correction terms, the correction terms discussed here exhibit time-dependent behavior. Specifically, the variables  $\Omega_\Delta$  and  $W_\Delta$  are functions of  $\dot{\mu}_\alpha$ , which is dependent upon  $Y_R^\tau$ , which itself is defined in terms of the input signal  $U_t - \tau$ .

It is important to note that unlike the magnetometer correction terms, the correction terms in this case are time-varying. Specifically,  $\Omega_\Delta$  and  $W_\Delta$  are defined in terms of  $\dot{\mu}_\alpha$ , which in turn depends on  $Y_R^\tau$ , which is further dependent on the time-varying input signal  $U_{t-\tau}$ . As such, it's essential to correctly propagate  $Y_R^\tau$  over both variables  $t$  and  $\tau$ . In our case, given the initial conditions (4.16), we can initialise the filter at any time  $t$  with  $Y_R^\tau = I_5$ , under the assumption that  $\tau = 0$ . Using this as the initial value of  $Y_R^\tau$ , we consider time propagating indefinitely, while  $\tau$  is assumed to be constant for this implementation. Suppose the desired value of  $\tau$  is  $\tau_0$ . There are two approaches for calculating  $Y_R^\tau$  properly.

### Method 1: Propagation in time at the desired delay, $\tau_0$

This method involves temporarily setting  $\tau = t$  and propagating  $Y_R(t, t)$  with respect to both arguments until  $\tau$  secs has elapsed (Step 1). The requirement  $t = \tau$  ensures that we can integrate the  $Y_R^\tau$  signal uniformly with respect to both input arguments until we reach  $\tau = \tau_0$ , where  $\tau_0$  represents the fixed delay of the GPS. From here, we relax the restriction that  $t = \tau$  and begin integrating  $Y_R^\tau$  with respect to only the time variable, allowing for  $\tau$  to remain constant at the desired value of  $\tau_0$  (Step 2).



**Figure 4.1:** Isometric view of trajectory of  $Y_R^\tau$  propagated in time,  $t$

**Figure 4.2:** Side view of trajectory of  $Y_R^\tau$  propagated in time,  $t$

*Step 1:* Consider the total derivative of  $Y_R(t, t)$  with respect to both arguments ( $x_1$  and  $x_2$ ) as denoted by  $\frac{d}{dt}Y_R(t, t)$ . Note,  $\frac{\partial}{\partial x_1}$  denotes the derivative of  $Y_R(t, t)$  with respect to the first argument and  $\frac{\partial}{\partial x_2}$  denotes the derivative of  $Y_R(t, t)$  with respect to the second argument .

$$\begin{aligned}\frac{d}{dt}Y_R(t, t) &= \frac{\partial}{\partial x_1}Y_R(t, t) + \frac{\partial}{\partial x_2}Y_R(t, t) \\ &= Y_R(t, t)[U_t + D] - [U_{t-\tau} + D]Y_R(t, t) + [U_{t-\tau} + D]Y_R(t, t) \\ &= [U_{t-\tau} + D - U_{t-\tau} - D]Y_R(t, t) + Y_R(t, t)[U_t + D] \\ &= Y_R(t, t)[U_t + D]\end{aligned}$$

We solve the explicit solution for this ODE using (3.2). Given this, and letting  $t = 0$  we see:

$$\begin{aligned}Y_R(t, t) &= Y_R(0, 0) \exp [(U(t) + D)t] \\ &= \exp [(U(t) + D)t]\end{aligned}$$

Consider  $Y_R(t, t)$  after a small step,  $\Delta t$ , with respect to both arguments.

$$\begin{aligned}Y_R(t + \Delta t, \tau + \Delta t) &= \exp [(U_t + D)(t + \Delta t)] \\ &= \exp [(U_t + D)t] \cdot \exp [(U_t + D)\Delta t] \\ Y_R(t + \Delta t, \tau + \Delta t) &= Y_R(t, t) \cdot \exp [(U_t + D)\Delta t]\end{aligned}$$

The following Euler integral is used until we have reached  $\sum_0^i (\Delta t_i) = \tau_0$  seconds.

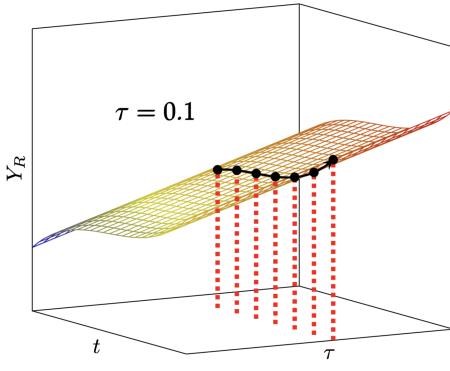
*Step 2:* Consider the Euler step from (4.25).

$$Y_R(t + \Delta t, \tau) = \exp[-\Delta t(U_{t-\tau} + D)] \cdot Y_R^\tau \cdot \exp[\Delta t(U_t + D)]$$

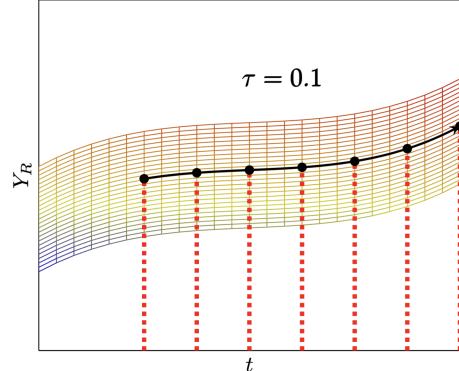
Specifically, the term to the left of  $Y_R^\tau$  above – namely  $\exp[-\Delta t(U_{t-\tau} + D)]$  – requires careful consideration in the context of real-world discretised sensor inputs. Suppose  $t' = t - \tau$  and we know  $U_{t'}$ . This signal,  $U_{t'}$ , cannot be assumed as constant over the time interval  $[t', t' + \Delta t]$ . That is, depending on how long  $\Delta t$  is, we may be required to compute this left integral with various  $U$  matrices for multiple time periods, until we have integrated exactly over a time period of exactly  $\Delta t$ . The product of these integrals will thus form the final left term. That is, for a time period  $t_k$ , with a corresponding  $U_k$ , where  $t - \tau < t_k < t - \tau + \Delta t$ , we calculate  $\exp[-t_k(U_k + D)]$  and multiply them together.

### Method 2: Propagation in delay, $\tau$ , for given time, $t_0$

This method is much simpler conceptually but requires substantially more computation. Recall that we require the calculation of  $Y_R^\tau$  at any given point in time, where  $\tau$  is a fixed value. To arrive at this value, we may simply start with the fact that  $Y_R^0 = I_5 \quad \forall t$ . As such, we may start at this value and integrate the  $\tau$  variable until we arrive at  $Y_R^\tau$ .



**Figure 4.3:** Isometric view of trajectory of  $Y_R^\tau$  propagated in delay,  $\tau$



**Figure 4.4:** Side view of trajectory of  $Y_R^\tau$  propagated in delay,  $\tau$

We know the Euler integral of  $\frac{d}{d\tau} Y_R^\tau$  is:

$$Y_R(t, \tau + \Delta\tau) = \exp(\Delta\tau(U_{t-\tau} + D)) Y_R^\tau$$

And so, at any given point in time,  $t_0$ , we start with  $Y_R^0 = I_5 \quad \forall t$  and aim to propagate this term until we arrive at  $Y_R^\tau$ . To do this, we find the velocity matrix,  $U$  at time  $t - \tau$  and we integrate it all the way till the present measurement  $U_t$ . That is, for a time period,  $t_k$ , with a corresponding  $U_k$ , where  $t - \tau < t_k < t$ , we calculate  $\exp[t_k(U_k + D)]$  and multiply them together.

While Method 1 is far superior in terms of computational efficiency, the implementation used for this work relied on Method 2. This implementation is outlined in the algorithm block below.

---

**Algorithm 2** DELPV

---

```
1: Initialize state estimator with initial estimates  $X$ ,  $Z$ , delay  $\tau$ , and gains
2: Define storage variables and initialize buffers for delayed state integration
3: Load ArduPilot replay log and YR list for delayed state propagation
4: for each message in the replay log do
5:   if message is GPS data with 3D fix then
6:     Store GPS position and velocity in storage variables
7:   else if message is IMU data then
8:     Retrieve the delayed state transformation matrix  $YR$  from YR list
9:     Compute the correction terms for both position and velocity
10:    Update the buffers with the current input and timestamp
11:    Integrate dynamics with the delayed transformation, current IMU data, and computed corrections
12:    Update state estimate  $\hat{X}$  and auxiliary state  $\hat{Z}$ 
13:   end if
14: end for
```

---

#### 4.5.3 DELPVxMAG: Combined Magnetometer and GNSS Time Delay Corrected Filter

The final filter that was produced in this work was the  $\text{DELPV}\times\text{MAG}$  filter, a hybrid solution that unifies the magnetometer corrections with the GNSS time delay corrections. The development of this filter is rooted in the  $\bar{E}$ -synchronous property of the filter, which allows for multiple correction terms to be summed for a cumulative effect. The implementation is outlined in the algorithm block below.

---

**Algorithm 3** DELPVxMAG

---

```
1: Initialise state estimator with initial estimates  $X$ ,  $Z$ , delay  $\tau$ , and gains
2: Define storage variables and initialize buffers for delayed state integration
3: Load ArduPilot replay log, YR list for delayed state propagation, and magnetometer reference value
4: for each message in the replay log do
5:   if message is GPS data with 3D fix then
6:     Update GPS position and velocity in storage variables
7:   else if message is magnetometer data then
8:     Update magnetometer measurement in storage variable
9:   else if message is IMU data then
10:    Retrieve the delayed state transformation matrix  $Y_R$  from  $Y_R$  list
11:    Compute correction terms for position, velocity, and magnetometer
12:    Update the buffers with the current input and timestamp
13:    Integrate dynamics with the delayed transformation, current IMU data, and computed corrections
14:    Update state estimate  $X$  and uncertainty estimate  $Z$  using position, velocity, and magnetometer data
15:   end if
16: end for
```

---

## 5 Experimental Results

---

The three filters – MAG, DELPV, and  $\text{DELPV} \times \text{MAG}$  – underwent evaluation in two test scenarios. These scenarios used data obtained from *i*) synthetic simulation and *ii*) real-world UAV flight log. This was a natural progression since validating a given filter’s performance for a given test scenario was important before gradually moving to a more complex one. Note that the original filter – referred to as CINS – was included in each test, allowing a point of comparison to the new filter’s performance.

### 5.1 Synthetic Simulation

In the previous work by Van Goor et al. [14], the results were based solely on simulations using synthetic input signals. Therefore, the first scenario is a similar synthetically generated flight. Here, we explicitly define a model for providing the input signals needed for the filter to run. The simulated trajectory was similar but more dynamic than the original simulation. In this case, only the IMU measurements and GNSS position and velocity were simulated, and magnetometer signals were not included in this test case. As such, this test scenario only compared CINS against DELPV.

The setup of this test case involved an extreme initial error in attitude, velocity and position as below

$$\hat{R}(0) = \exp(0.99\pi\mathbf{e}_1^\times) \quad \hat{v}(0) = \begin{pmatrix} 0.2 & 0.4 & -1.1 \end{pmatrix}^\top \text{ m/s}, \quad \hat{p}(0) = \begin{pmatrix} 3 & -2 & 2 \end{pmatrix}^\top \text{ m},$$

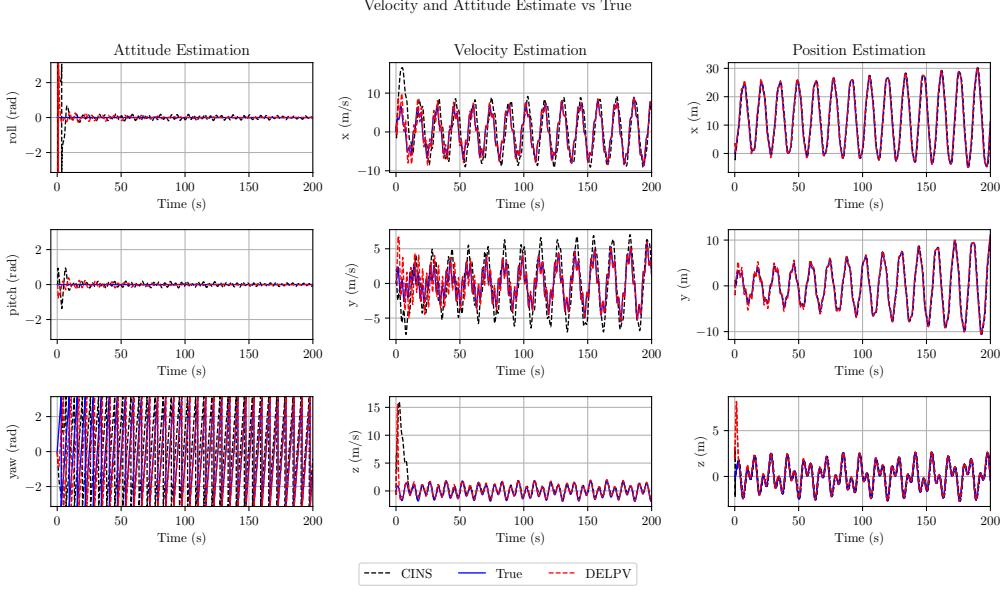
These values align with those used in the simulation from the original work on CINS. The velocity generation functions used were

$$\Omega(t) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad a(t) = \cos(t) \begin{pmatrix} 5.0 \\ 0.0 \\ 1.0 \end{pmatrix} - g - 0.2R_t^{-1}\hat{p} \quad (5.1)$$

These input controls aided in a flight that mimicked a bow-tie flight, where only the yaw angle has any angular velocity. A delay of 0.4 sec was also injected into the simulated GNSS measurements. The chosen gains for this implementation were:

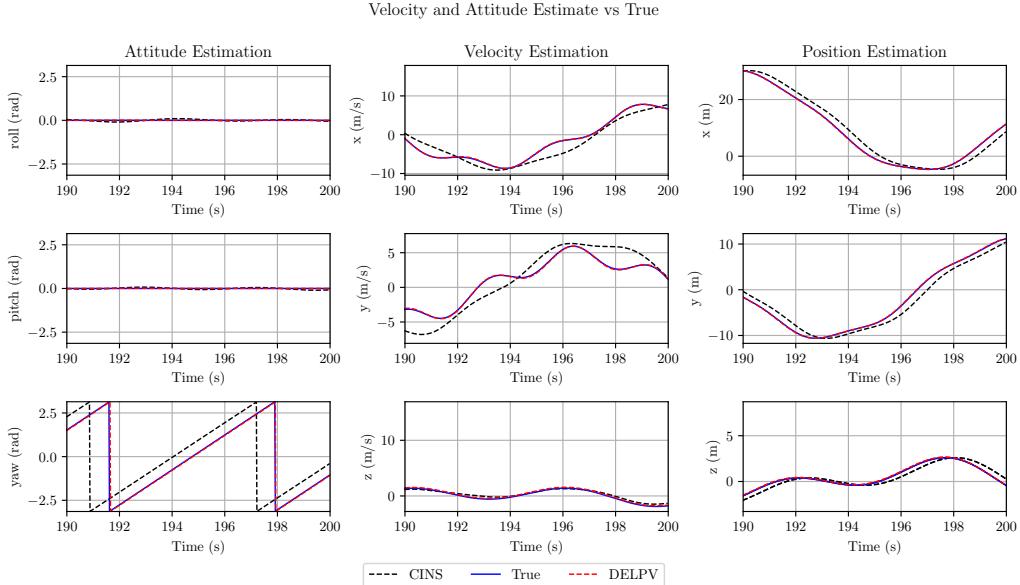
$$l_{p,1} = 2 \quad l_{p,2} = 1 \quad l_{v,1} = 0 \quad l_{v,2} = 1.1 \quad k_p = 1 \quad k_v = 10^{-5}$$

Figure 5.1 shows the estimates of CINS and DELPV against the ground truth across all states. This includes three attitude estimates (roll, pitch, yaw) in the body frame and three velocity and position estimates in the NED inertial frame. North is denoted  $x$ , East is denoted  $y$ , and Down is denoted  $z$ . This will remain consistent across the various plots to be analysed. At this scale, we observe the convergence of both filters to the ground truth, although CINS notably has more noise associated with it than DELPV, particularly across all velocity measurements, as well as position in  $z$ , roll and pitch.



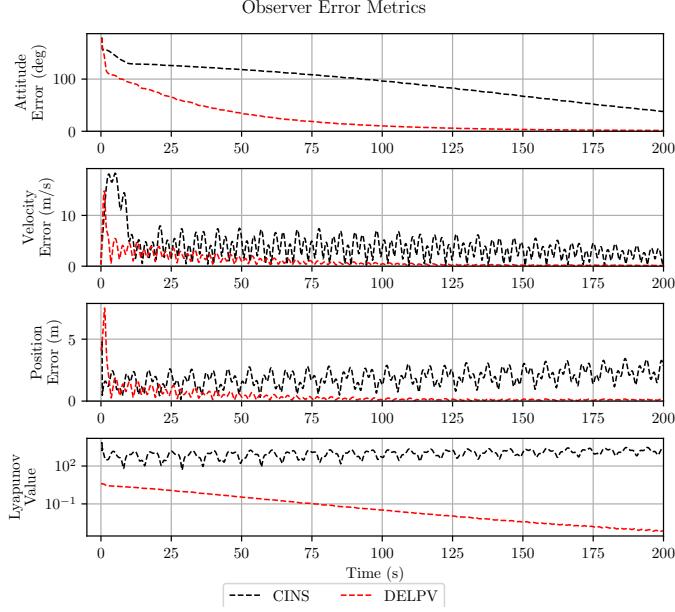
**Figure 5.1:** Attitude (left), velocity (centre) and position (right) estimate for CINS (black) and DELPV (red), compared to the ground truth (blue)

Figure 5.2 shows the steady-state response of the filters in the final 10 secs of the simulated flight. What's immediately apparent is the disparity between the CINS and the ground truth. On the contrary, DELPV shows very strong performance and is indistinguishable from the ground truth. Both these observations apply across all states. Interestingly, in the position estimates, the CINS estimates are consistently offset from the expected state by a fixed phase shift of approximately 0.4 sec, which lines up with the delay that was set.



**Figure 5.2:** Steady-state response (final 10 secs) of attitude (left), velocity (centre) and position (right) estimates for CINS (black) and DELPV (red), compared to the ground truth (blue)

Finally, Figure 5.3 shows how the error changes with time. For all subplots, except for the attitude estimate, CINS seems to flatten off or even start to increase with time as is the case with the position error. DELPV, on the other hand, seems to linearly approach 0 for all error types. This behaviour highlights the property of DELPV, which allows it to remove the steady-state error stemming from the delayed GNSS measurements. Ultimately, this allows the error to converge more accurately, especially over extended flight periods.



**Figure 5.3:** Observer error for attitude, velocity, position and the Lyapunov value for CINS (black) against DELPV (black)

Below, we present the RMSE calculated between the EKF3 and the various proposed filter designs. For this test case, the delay-corrected filter was better across all states and all periods of the flight (transient, steady state and full flight). Table 4 below showcases the RMSE for each individual state over the entire flight, as well as the combined RMSE for each category of state (position, velocity, attitude). Note the tables below use PN, PE, and PD to refer to the position in the north, east and down axes, respectively, and a similar convention for velocity with notation VN, VE, and VD. Note also the optimal result for each state is bolded. Evidently, DELPV performed better across all states.

<b>RMSE</b>	<b>CINS</b>	<b>DELPV</b>
PN (m)	1.7646	<b>0.3571</b>
PE (m)	0.8834	<b>0.3526</b>
PD (m)	0.3939	<b>0.5646</b>
VN (m/s)	3.6840	<b>0.9817</b>
VE (m/s)	2.5326	<b>1.0603</b>
VD (m/s)	2.1625	<b>1.0000</b>
Roll (deg)	0.3872	<b>0.2720</b>
Pitch (deg)	0.1718	<b>0.0969</b>
Yaw (deg)	2.7087	<b>1.4747</b>
Total Pos (m)	3.0418	<b>1.2742</b>
Total Vel (m/s)	8.3791	<b>3.0420</b>
Total Att (deg)	3.2677	<b>1.8436</b>

**Table 4:** Comparison of RMSE values for CINS and DELPV×MAG

Despite the promising results, a synthetic simulation offers the filters ideal conditions to perform without noise in the input signals, perfectly synchronous input data and the absence of any bias. Among other things, these sources of error and uncertainty are unavoidable and must be managed for a practical filter. Therefore, these synthetic results are only an initial point of comparison for the time-delay correction and will require more rigorous testing.

## 5.2 Real Flight Data

Here, we assess the filters against a dataset from a real-world flight, which inherently carries a range of sources of uncertainty: measurement noise, the possibility of not having a GPS fix in the transient, asynchronous sampling rates and irregular sampling intervals, to name a few.

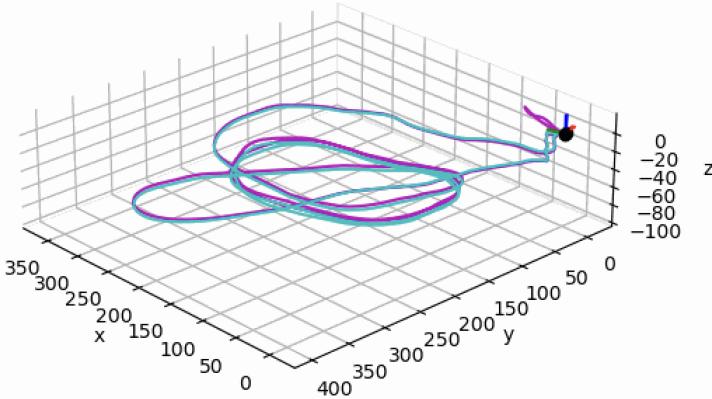
This test case used a 6-minute flight log flown by a Striver mini VTOL miniature fixed-wing aircraft at Spring Valley, Stromolo, ACT. The aircraft was boarded with a GNSS, magnetometer and IMU (among other sensors), and thus captured all the information needed for all the developed filters to function (MAG, DELPV, DELPV×MAG). A major limitation of testing the filters on real-world flight data is that there is no concept of a genuine ground truth. While alternative technologies can aid in this problem, this was outside the project’s scope. Therefore, the approach used a dataset with a relatively low-speed flight and in ideal weather conditions. Since these conditions were met for the particular dataset being used, we may assume ArduPilot’s EKF implementation, known as EKF3, as the ground truth for a point of comparison. The other challenging aspect of the real-life data is that it’s much more difficult to know the exact time delay present in the GPS. Once again, there are methods for doing this, such as iteratively testing multiple values of the delay and using the one that produces the lowest innovation in the velocity correction, but this was out of the scope. Thus, a more naive approach was taken, in that the visual discrepancy between the positional response of the CINS and EKF3 in the steady-state, was assumed as the fixed-GPS time delay,  $\tau$ . For this dataset, this was found to be 0.3 secs.

The setup for this test case involved initialising the filter was

$$\hat{R}(0) = \exp\left(\frac{\pi}{6} \begin{pmatrix} 1 & -1 & 0 \end{pmatrix}^\top \times\right) \quad \hat{v}(0) = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^\top \text{ m/s}, \quad \hat{p}(0) = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^\top \text{ m},$$

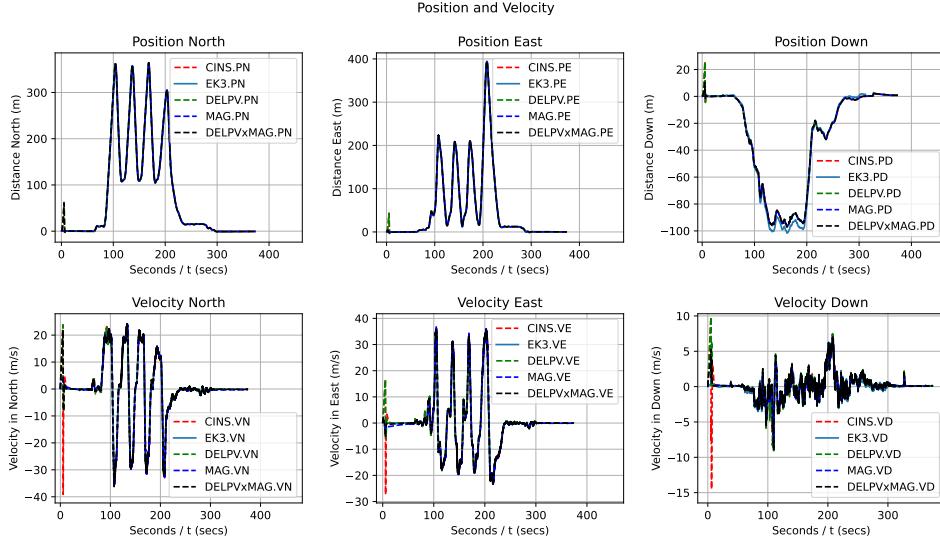
The general trajectory of this dataset is as shown in Figure 5.4, EKF3 as the cyan curve, and DELPV×MAG as pink. It consisted of a vertical take-off followed by a series of aerial loops before landing in a similar spot to where it took off. Note, the figure is in NED coordinates, which explains why the aircraft seems to move downwards based on the trajectory. The chosen gains here were the same as for the synthetic simulation

$$l_{p,1} = 2 \quad l_{p,2} = 1 \quad l_{v,1} = 0 \quad l_{v,2} = 1.1 \quad k_p = 1 \quad k_v = 10^{-5}$$

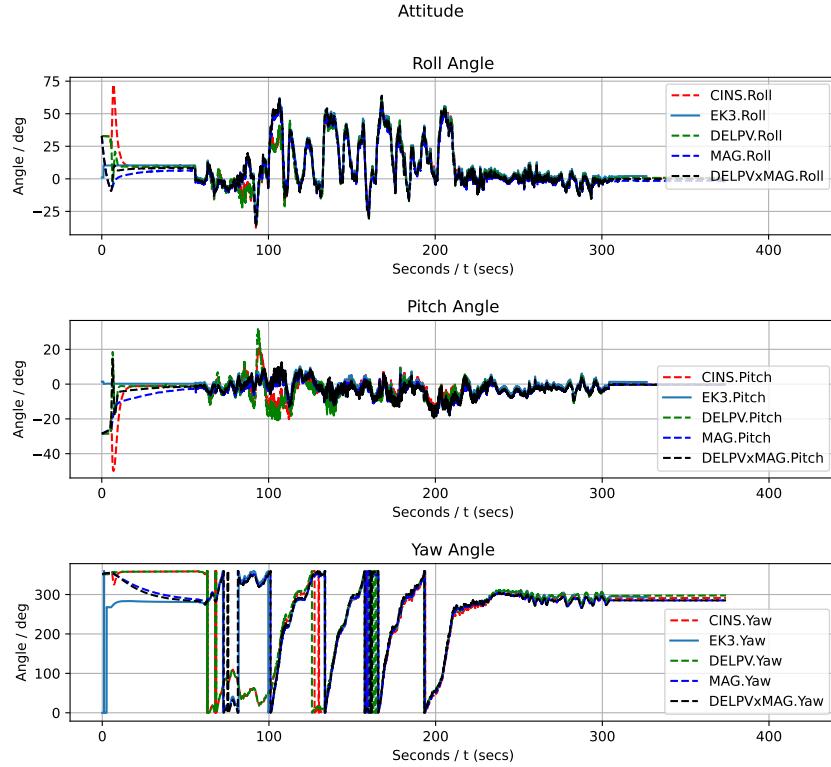


**Figure 5.4:** Trajectory of the real-flight test case comparing EKF3 (cyan) against DELPV×MAG (pink)

Figure 5.5 and Figure 5.6 present the performance of each filter with respect to estimating the translational (position and velocity) and attitude states, respectively, over the duration of the entire flight. Generally, the filters are all converging well and relatively quickly as well. At this scale, the filters have a lot of overlap in their steady-state response; however, the transient seems to trigger various strange behaviours.



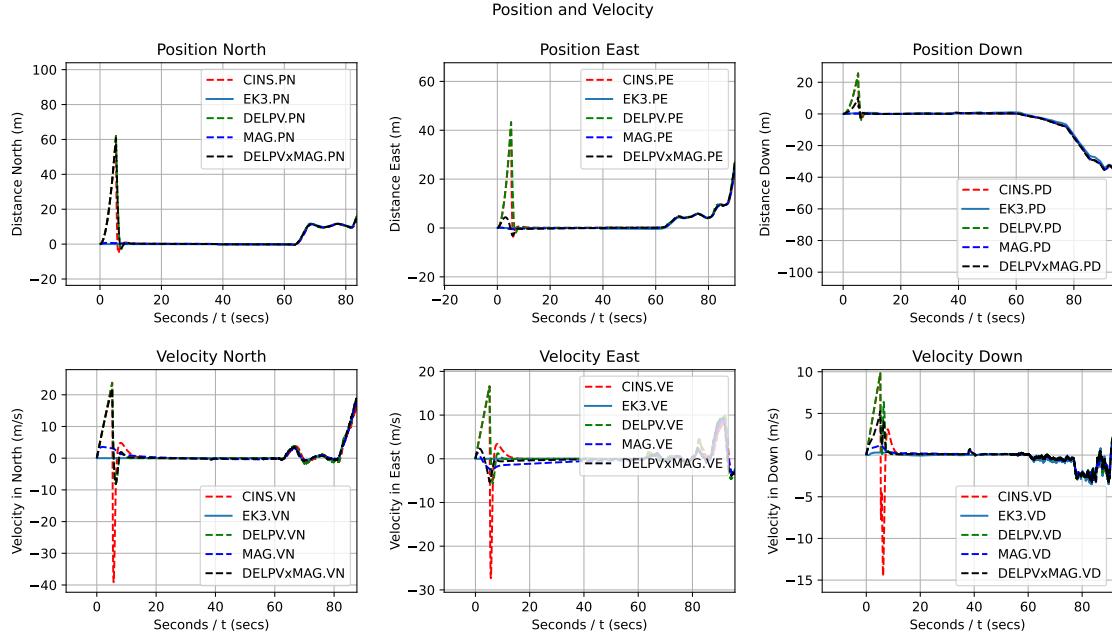
**Figure 5.5:** Position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and  $\text{DELPV} \times \text{MAG}$  (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates



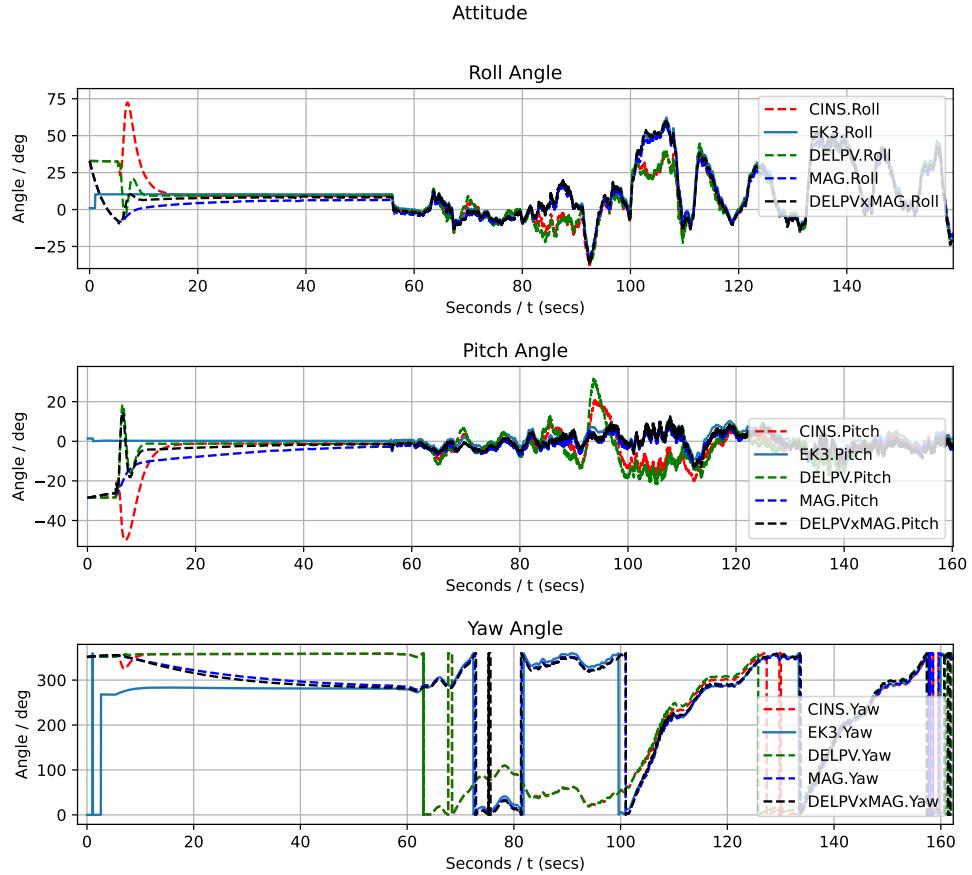
**Figure 5.6:** Attitude estimates for CINS (red), MAG (blue), DELPV (green) and  $\text{DELPV} \times \text{MAG}$  (black) against the assumed ground truth, EKF3 (light blue)

Observing the transient more closely, Figures 5.7 and 5.8 below show the transient response of each filter over the initial 100 seconds approximately. Evidently, the two magnetometer-fused filters clearly outperform the other two, especially with all the translational states. This disparity may be a result of

the heavy reliance on the non-magnetometer corrected filters (CINS and DELPV) having a proportionately higher dependence on GNSS for its state estimates. This is particularly problematic in the transient since the GPS fix takes a few seconds to establish for this particular log. Meanwhile, the magnetometer-fused filters can rely on the magnetometer measurements for external correction, which may help with faster convergence. This trend extends further when the yaw estimates are considered. The magnetometer-fused filters seem to converge from the initial error in approximately 50 sec, while the other two filters take almost three times as long to converge properly at the 140-second mark. This was an expected result. The filters, without mag, depend entirely on the persistence of excitation to determine an estimate of yaw. Notably, the initial 60 seconds of this log involve no movement. Thus, it's evident why the filters without mag only start to converge once the vehicle begins to move. On the contrary, the magnetometer provides directional information that aids in attitude estimation in a broad sense, but specifically the yaw, as can be seen. It is also worth noting that the magnetometer-fused filters converge more slowly for the roll and pitch estimates. This is likely a consequence of the typically large noise characteristics of low-cost magnetometers or potentially related to the choice of gains for the filters since the delay-corrected filters (DELPV and DELPV×MAG) are derived on an entirely different architecture to the other two filters, meaning that it's difficult to know whether one filter has been more desirably tuned.

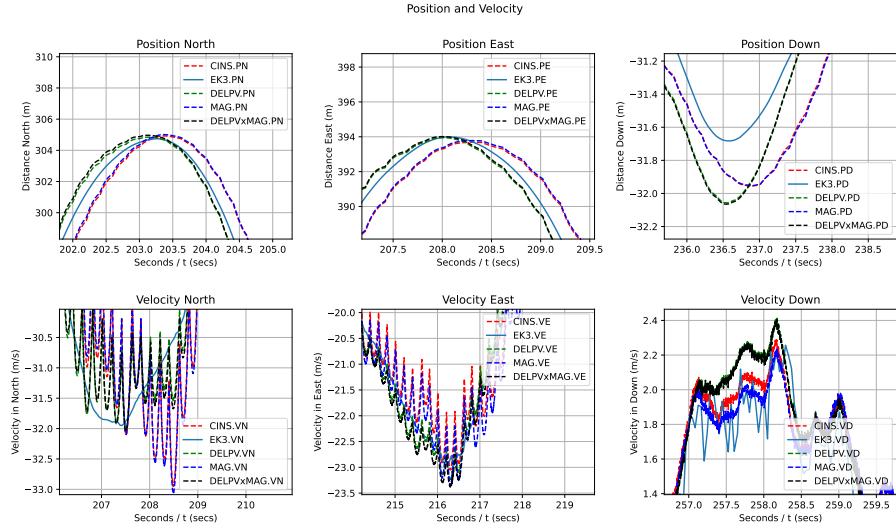


**Figure 5.7:** Transient response (first 90 secs) of position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and DELPV×MAG (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates

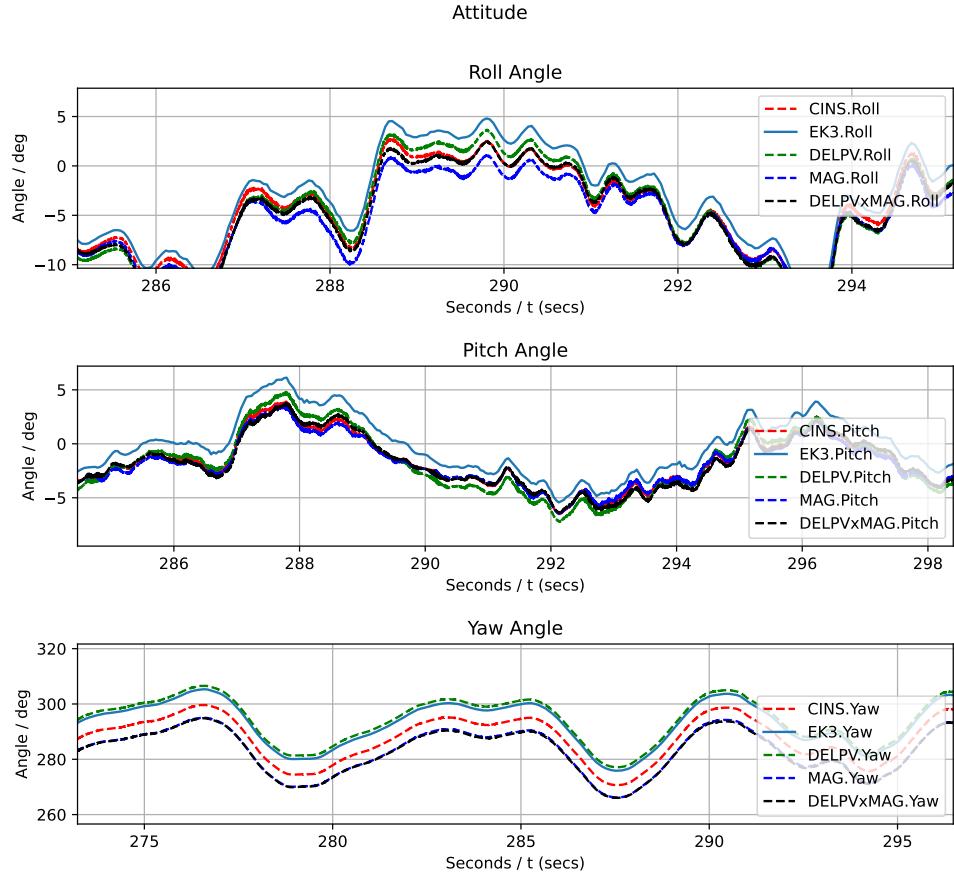


**Figure 5.8:** Transient response (first 160 secs) of attitude estimates for CINS (red), MAG (blue), DELPV (green) and DELPV $\times$ MAG (black) against the assumed ground truth, EKF3 (light blue)

The other particularly interesting aspect of the figures lies in the steady-state response, as shown in Figures 5.9 and 5.10. Particularly with reference to the position estimates in Figure 5.9, we notice a key difference between the delay-corrected filters and the rest. The delay-corrected filters maintain far superior synchrony with the timing of the EKF3's response. Conversely, the other two filters are consistently out of phase with the EKF3, as expected when fusing delayed measurements into a filter. This was a key indicator that the delay correction was working as expected. In analysing the steady-state response in the velocity and attitude estimates, such a delay is much less apparent. However, there are other noteworthy observations to be made. Firstly, the velocity measurements are highly sporadic, and there is no clear trend across the three figures for which filter may perform best. For the attitude estimates in the steady state, it is clear to see that the non-magnetometer fused filters (DELPV and CINS) are performing the best in the steady state. Once more, this may be due to the noise characteristics of magnetometers.



**Figure 5.9:** Steady-state response (zoomed regions after 200 sec) of position (top) and velocity (bottom) estimates for CINS (red), MAG (blue), DELPV (green) and  $\text{DELPV} \times \text{MAG}$  (black) against the assumed ground truth, EKF3 (light blue) in NED coordinates



**Figure 5.10:** Steady-state response (zoomed regions after 200 sec) of attitude estimates for CINS (red), MAG (blue), DELPV (green) and  $\text{DELPV} \times \text{MAG}$  (black) against the assumed ground truth, EKF3 (light blue)

The most insightful result to be drawn from this work is the performance of the **DELPV** $\times$ **MAG** filter. This filter was designed by incorporating a correction for both the magnetometer and GNSS time delay, and the cumulative effect is evident. It is the only filter to converge quickly for all of its state estimates, which also corrects for some of the steady-state error induced by delayed GNSS measurements – this was particularly apparent for the positional estimates. This behaviour is a clear reflection of the efficacy of the  $\bar{E}$ -synchrony property, which laid the foundation for this work.

Table 7 below showcases the RMSE across the entire flight for all filters. There are two stand-out filters, which are **MAG** and **DELPV** $\times$ **MAG**. Interestingly, the magnetometer-infused filters seemed to perform better across the entire duration of the flight. In particular, **MAG** performs even better than **DELPV** $\times$ **MAG**. The difference in results between the **DELPV** $\times$ **MAG** filter and the **MAG** filter may be due to the differences in their mathematical foundation and, more importantly, the choice of gains. Given the two filters have reasonably different formulations, it is likely that the gain choices are not comparable. Specifically, it may be the case the **DELPV** $\times$ **MAG** has a higher relative weight on magnetometer than GPS, compared to **MAG**. This may allow noise from the mag to affect the filter’s response. Furthermore, the fact that **DELPV** is the best-performing filter in the steady state, yet the fusion of magnetometer into **DELPV** produced a filter that wasn’t as effective is further evidence of this.

RMSE	CINS	DELPV	MAG	DELPVxMAG
PN (m)	4.3849	4.1191	<b>2.3731</b>	4.0793
PE (m)	3.4517	2.9677	2.2900	<b>1.1145</b>
PD (m)	3.0394	3.1041	<b>2.6505</b>	2.7639
VN (m/s)	2.6316	1.9195	<b>0.9410</b>	1.7977
VE (m/s)	1.9628	1.3613	0.9374	<b>0.4923</b>
VD (m/s)	1.0546	0.8426	<b>0.2851</b>	0.4432
Roll (deg)	7.4128	6.1015	4.3174	<b>3.1524</b>
Pitch (deg)	7.2316	6.5084	5.4363	<b>4.7104</b>
Yaw (deg)	110.7314	117.0238	<b>53.4966</b>	54.3321
Total Pos (m)	10.8759	10.1909	<b>7.3136</b>	7.9577
Total Vel (m/s)	5.6490	4.1235	<b>2.1635</b>	2.7333
Total Att (deg)	125.3758	129.6337	63.2503	<b>62.1948</b>

**Table 5:** Full flight RMSE for each filter compared to **EKF3** across each individual and combined state

Given the interesting characteristics of the full-flight RMSE results, we further investigate the transient and steady-state RMSE. Tables 6 and 7 below present the transient and steady-state RMSE results, respectively. The transient response demonstrates much of the same behaviour as Table 5, where the magnetometer-corrected filters dominate. This is consistent with the visual results previously discussed through Figure 5.7 and 5.8, likely due to the lack of persistence of excitation.

<b>RMSE</b>	<b>CINS</b>	<b>DELPV</b>	<b>MAG</b>	<b>DELPVxMAG</b>
PN (m)	11.6184	12.6118	<b>0.3282</b>	12.4604
PE (m)	8.1154	8.8060	<b>0.2278</b>	1.2870
PD (m)	4.6832	4.8372	<b>0.2928</b>	1.7901
VN (m/s)	7.7894	5.8742	<b>1.5144</b>	5.5988
VE (m/s)	5.4386	4.0949	1.3190	<b>1.2348</b>
VD (m/s)	3.1221	2.3761	<b>0.3592</b>	1.2029
Roll (deg)	18.1365	10.5793	9.3731	<b>7.2364</b>
Pitch (deg)	18.9628	12.7518	14.5926	<b>12.6027</b>
Yaw (deg)	121.8836	123.0303	111.0051	<b>109.9760</b>
Total Pos (m)	24.4169	26.2551	<b>0.8488</b>	15.5376
Total Vel (m/s)	16.3501	12.3452	<b>3.1926</b>	8.0366
Total Att (deg)	158.9829	146.3614	134.9707	<b>129.8151</b>

**Table 6:** Transient (first 60 seconds) RMSE for each filter compared to EKF3 across each individual and combined state

Table 7 on the other hand, shows the converse result with the non-magnetometer fused filters performing the best, particularly DELPV. Once more, as supported by the visual interpretation of Figure 5.9 and 5.10, this is likely due to the magnetometer’s noise characteristics.

<b>RMSE</b>	<b>CINS</b>	<b>DELPV</b>	<b>MAG</b>	<b>DELPVxMAG</b>
PN (m)	<b>0.0242</b>	0.0270	0.0321	0.0429
PE (m)	0.1385	0.1364	0.1345	<b>0.1304</b>
PD (m)	0.4310	<b>0.4117</b>	0.4309	<b>0.4117</b>
VN (m/s)	<b>0.0223</b>	<b>0.0201</b>	0.0860	0.0370
VE (m/s)	0.0246	<b>0.0115</b>	0.0353	0.0161
VD (m/s)	0.0521	<b>0.0486</b>	0.0526	0.0487
Roll (deg)	0.4762	<b>0.4596</b>	1.0469	0.6916
Pitch (deg)	<b>0.4740</b>	0.4950	0.4765	0.4967
Yaw (deg)	1.4557	<b>0.6011</b>	3.0700	3.1057
Total Pos (m)	0.5937	<b>0.5752</b>	0.5975	0.5850
Total Vel (m/s)	0.0990	<b>0.0802</b>	0.1739	0.1018
Total Att (deg)	2.4059	<b>1.5556</b>	4.5935	4.2940

**Table 7:** Steady-state (final 60 seconds) RMSE for each filter compared to EKF3 across each individual and combined state

## 6 Conclusion and Future Work

---

### 6.1 Summary of Findings

This research aimed to expand on the work done by Van Goor et al. (2021) [14] by improving the CINS filter with a magnetometer data fusion and a GNSS position and velocity time delay correction. In Section 4, the mathematical formulation of these corrections was discussed, and implementation approaches were proposed in Section 4.5. There were three novel filters in total being evaluated, MAG, a magnetometer fused filter, DELPV, and GNSS position and velocity delay correct filter, and  $\text{DELPV} \times \text{MAG}$  which was a combination of both, leveraging the  $\bar{E}$ -synchronous property of the CINS filter [14]. Finally, in Section 5, two thorough evaluations were conducted on the filters. Firstly, through a synthetic simulation, the delay-corrected filter, DELPV was compared against the original filter, CINS, and DELPV outperformed CINS in every possible way. We then evaluated all of the proposed filters in a real-world test scenario, where real flight data obtained from a fixed-wing VTOL aircraft was used. The results demonstrated the expected outcome of both the magnetometer and GNSS time-delay corrections. It showed that the yaw convergence in the transient was accelerated in the absence of any persistence of excitation and that the visible impact of the delay (particularly for the positional estimates) was nullified after the respective correction was made. The results also suggested that the noise characteristics of the magnetometer severely impacted the steady-state performance. This was particularly apparent in the combined filter,  $\text{DELPV} \times \text{MAG}$ , where the magnetometer's noise directly opposed the time-delay correction working to reduce the steady-state error. Nevertheless, the novel implementations of magnetometer fusion and GNSS time delay corrections in this work yielded a result showing great promise for CINS as a highly advantageous estimator to be adopted for not only its global robustness guarantees, but also its performance as evidenced by this work.

### 6.2 Recommendations for Future Research

There are many potential avenues for extending the work done through this project. The most apparent one is the need for more rigorous testing of the filters. In this work, only a single synthetic simulation and a single real-world data test were formally evaluated, which may cause the reliability of these findings to be scrutinised. Further testing in simulation, but more importantly through real-world flight logs is imperative for adding to this initial set of findings. In this regard, a limitation of this work was the lack of a genuine ground truth for comparison in the context of the real-world test case. Thus, the use of technologies, such as RTK-GPS or Vicon systems, would likely allow for more accurate performance metrics to be deduced. Further tests can be conducted through extensive gain tuning on the various gains that were defined as part of this work ( $k_m, k_p, k_v, l_{v,1}, l_{v,2}, l_{p,1}, l_{p,2}$ ), would likely lead to even better performance of the filters. Exploration of the robustness of these modified versions of CINS through testing a suite of extreme initial conditions may prove insightful.

There are also implementation-related improvements that can be made to the current filter. As mentioned in Section 4.5, this work employed ‘Method 2’ of the two approaches discussed for propagating  $Y_R^\tau$  term through time. This method is extremely computationally expensive and has no scope for utility in real-time processing applications. Therefore, a key extension to this work would be to get a working implementation

of ‘Method 1’, and compare its performance to the results from this work. Furthermore, a fundamental requirement of any widely successful state estimator is its ability to predict the bias characteristics of its sensors [42, 9], and this poses yet another avenue for meaningfully upgrading the CINS filter. This could also be extended to fusing barometer or airspeed sensor data into the filter. A further improvement to the implementation would be incorporating a dynamic GNSS time delay to remove the assumption of a constant delay.

### 6.3 Concluding Remarks

Although it is unlikely to outperform a stochastic filter like the EKF when the system remains within its linearisation domain, a filter that provides almost global asymptotic and local exponential stability poses as a highly valuable addition to modern-day avionic systems. Where safety is of paramount importance, the significance of this stability cannot be overstated. It offers a reliable alternative for inertial navigation in systems with limited sensor availability, as demonstrated in [14], while also showcasing the potential for further enhancements, as evidenced by the research presented in this work. The implications of this study extend beyond the immediate application, hinting at a future where robust, stable navigation is the cornerstone of autonomous avionic systems.

## References

---

- [1] Guowei Cai, Ben M. Chen, and Tong Heng Lee. *Coordinate Systems and Transformations*, pages 23–34. Springer London, London, 2011.
- [2] Dag Pike. *The History of Navigation*. Pen and Sword, 2018.
- [3] Richard B. Langley, Peter J.G. Teunissen, and Oliver Montenbruck. *Introduction to GNSS*, pages 3–23. Springer International Publishing, Cham, 2017.
- [4] NSW Department of Primary Industries. Echo sounder, sonar, radar operations and navigation course. Final report, NSW Department of Primary Industries, 1981.
- [5] Cezary Specht. Radio navigation systems: Definitions and classifications. *Journal of Navigation*, pages 1–10, 05 2021.
- [6] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.
- [7] Quang Truc Dam, Rihab El Houda Thabet, Sofiane Ahmed Ali, and Francois Guerin. Observer design for a class of uncertain nonlinear systems with sampled-delayed output using high-gain observer and low-pass filter: Application for a quadrotor uav. *IEEE Transactions on Industrial Electronics*, 71(1):933–942, 2023.
- [8] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom, August 2007.
- [9] Jacob Willem Langelaan. State estimation for autonomous flight in cluttered environments. *Standford University*, 2006.
- [10] Mohammad Al-Sharman, Yahya Zweiri, Mohammad Jaradat, Raghad Al-Husari, Dongming Gan, and Lakmal Seneviratne. Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation. *IEEE Transactions on Instrumentation and Measurement*, 02 2019.
- [11] Vitor Guizilini and Fabio Ramos. Visual odometry learning for unmanned aerial vehicles. pages 6213 – 6220, 06 2011.
- [12] Jinyao Zhu, Chao Yao, and Klaus Janschek. Stereo visual-inertial fusion for uav state estimation. *IFAC-PapersOnLine*, 53(2):9420–9425, 2020. 21st IFAC World Congress.
- [13] Vlad Aurelian Vaduvescu and Petre Negrea. Inertial measurement unit – a short overview of the evolving trend for miniaturization and hardware structures. In *2021 International Conference on Applied and Theoretical Electricity (ICATE)*, pages 1–5, 2021.
- [14] Pieter van Goor, Tarek Hamel, and Robert Mahony. Constructive equivariant observer design for inertial velocity-aided attitude. *IFAC-PapersOnLine*, 56(1):349–354, 2023.

- [15] Daniel Tazartes. An historical perspective on inertial navigation systems. In *2014 International Symposium on Inertial Sensors and Systems (ISISS)*, pages 1–5, 2014.
- [16] L. A. Mcgee and S. F. Schmidt. Discovery of the kalman filter as a practical tool for aerospace and industry. Technical Memorandum NASA-TM-86847 NAS 1.15:86847 REPT-85424, NASA Ames Research Center Moffett Field, CA, United States, 1985. Public Use Permitted. Work of the US Gov.
- [17] Lorenzo Fusini, Thor I. Fossen, and Tor Arne Johansen. Nonlinear observers for gnss- and camera-aided inertial navigation of a fixed-wing uav. *IEEE Transactions on Control Systems Technology*, 26(5):1884–1891, 2018.
- [18] Elias Bjørne, Edmund Førland Brekke, Torleiv Håland Bryne, and Tor Arne Johansen. *IEEE Transactions on Control Systems Technology*, (5):2279–2286, September 2021.
- [19] Soulaime Berkane, Abdelhamid Tayebi, and Simone de Marco. A nonlinear navigation observer using imu and generic position information. *Automatica*, 127:109513, 2021.
- [20] R. Mahony, T. Hamel, and J.-M. Pflimlin. Complementary filter design on the special orthogonal group  $so(3)$ . In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1477–1484, 2005.
- [21] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.
- [22] Mina W. William, Mohamed E. Shaaban, Catherine M. Elias, Omar M. Shehata, and Elsayed I. Morgan. Multiple estimation filters integration for sensor fusion localization module. In *2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 316–321, 2023.
- [23] José Vasconcelos, Bruno Cardeira, Carlos Silvestre, Paulo Oliveira, and Pedro Batista. Discrete-time complementary filters for attitude and position estimation: Design, analysis and experimental validation. *IEEE Trans. Contr. Sys. Techn.*, 19:181–198, 01 2011.
- [24] Parag Narkhede, Shashi Poddar, Rahee Walambe, George Ghinea, and Ketan Kotecha. Cascaded complementary filter architecture for sensor fusion in attitude estimation. *Sensors*, 21(6), 2021.
- [25] ArduPilot. ArduPilot DevConf 2023 - Saturday Morning, November 2023. [Online; accessed 10. Nov. 2023].
- [26] Sarvenaz Salehi, Navid Mostofi, and Gabriele Bleser-Taetz. A practical in-field magnetometer calibration method for imus. 10 2012.
- [27] Todd E. Humphreys, Mark L. Psiaki, Eric M. Klatt, Steven P. Powell, and Paul M. Jr. Kintner. Magnetometer-based attitude and rate estimation for a spacecraft with wire booms. *Journal of Guidance, Control, and Dynamics*, 28(4):584–593, 2005.
- [28] Valérie Renaudin, Muhammad Afzal, and Gérard Lachapelle. New method for magnetometers based orientation estimation. pages 348 – 356, 06 2010.

- [29] Jakob Mahler Hansen, Thor I. Fossen, and Tor Arne Johansen. Nonlinear observer for ins aided by time-delayed gnss measurements: Implementation and uav experiments. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 157–166, 2015.
- [30] G. Jacobitti and G. Scarano. Discrete time techniques for time delay estimation. *IEEE Transactions on Signal Processing*, 41(2):525–533, 1993.
- [31] Isaac Skog and Peter Handel. Time synchronization errors in loosely coupled gps-aided inertial navigation systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1014–1023, 2011.
- [32] Alireza Khosravian, Jochen Trumpf, Robert Mahony, and Tarek Hamel. Velocity aided attitude estimation on  $\text{so}(3)$  with sensor delay. In *53rd IEEE Conference on Decision and Control*, pages 114–120, 2014.
- [33] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics, 2021.
- [34] Eduardo Gallo. The  $\text{so}(3)$  and  $\text{se}(3)$  lie algebras of rigid body rotations and motions and their application to discrete integration, gradient descent optimization, and state estimation. *arXiv preprint*, 2023.
- [35] John M. Lee. *Introduction to smooth manifolds*. Springer, 2003.
- [36] Comparison of 3-D Coordinate Systems - MATLAB & Simulink - MathWorks Australia, October 2023. [Online; accessed 26. Oct. 2023].
- [37] Randal Beard. Uav coordinate frames and rigid body dynamics. *BYU ScholarsArchive, Faculty Publications*, 2004. Original Book: Beard, Randal W. and McLain, Timothy W., Small Unmanned Aircraft: Theory and Practice, Princeton University Press, 2012, ISBN: 978-06-911-4921-9.
- [38] Mohammed Abdulhakim Al-Absi, Rui Fu, Ki-Hwan Kim, Young-Sil Lee, Ahmed Abdulhakim Al-Absi, and Hoon-Jae Lee. Tracking unmanned aerial vehicles based on the kalman filter considering uncertainty and error aware. *Electronics*, 10(24), 2021.
- [39] Jing-Jing Xiong and En-Hui Zheng. Optimal kalman filter for state estimation of a quadrotor uav. *Optik*, 126(21):2862–2868, 2015.
- [40] ArduPilot Documentation, year = 2023, month = nov, note = [Online; accessed 10. Nov. 2023], url = <https://ardupilot.org/ardupilot>.
- [41] Extended kalman filter (ekf3), November 2023. [Online; accessed 10. Nov. 2023].
- [42] Magda Monteiro and Marco Costa. A note on prediction bias for state space models with estimated parameters. volume 1479, pages 2090–2093, 09 2012.