# Inertial Navigation System Drift Reduction Using Scientific Machine Learning

by

Matthew McManus

B.S. Computer Science and Engineering, Massachusetts Institute of Technology, 2024

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024



Authored by: Matthew McManus
Department of Electrical Engineering and Computer Science
May 10, 2024

Certified by: Alan Edelman
Professor of Applied Mathematics, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair
Master of Engineering Thesis Committee

# Inertial Navigation System Drift Reduction Using Scientific Machine Learning

by

Matthew McManus

Submitted to the Department of Electrical Engineering and Computer Science on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE


## ABSTRACT

Inertial Navigation Systems (INS) are crucial for accurate navigation in GPS-denied environments, but they suffer from drift errors that accumulate over time. This thesis introduces Scientific Machine Learning (SciML) as an innovative approach to mitigate INS drift by integrating physical models with machine learning algorithms. The proposed SciML architecture leverages neural networks to learn complex error patterns and relationships from simulated IMU data, outperforming conventional techniques like Kalman filtering. Utilizing a simulation-focused approach with the Julia programming language and the High-Performance Inertial Navigation Development Repository (HIDR) library, the research generates realistic datasets encompassing diverse trajectories, sensor errors, and operational conditions. The SciML methodology incorporates data generation, INS mechanization, error modeling using neural networks, and a filtering framework that integrates the Extended Kalman Filter (EKF) with batch filtering techniques. Experimental results demonstrate the superior performance of the SciML-based INS in reducing position, velocity, and attitude errors compared to a baseline Kalman filter. This pioneering approach of fusing SciML with INS physical models holds promise for revolutionizing drift error mitigation and advancing the field of navigation systems, paving the way for more accurate, reliable, and resilient navigation in GPS-denied environments, with potential applications in aviation, robotics, and autonomous vehicles.


Thesis supervisor: Alan Edelman
Title: Professor of Applied Mathematics

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Alan Edelman, from the Julia Lab, and Aaron Canciani from the Leidos team for their unwavering support throughout my MEng project. I am incredibly thankful to the Julia Lab for providing me with the opportunity to work on such a fascinating problem involving Scientific Machine Learning and for granting me the freedom to collaborate with the Leidos team. A special thank you goes to Aaron Canciani for onboarding me to this project and for his assistance in resolving any issues I encountered along the way.

Beyond the academic sphere, I have been incredibly fortunate to find two communities at MIT that I can call home: the MIT Squash Team and Phi Kappa Theta (PKT). Within the MIT Squash Team, I would like to extend a special thanks to Thierry Lincou for being one of the best leaders and coaches I know, and for always providing a fun and safe space away from the academic rigors of MIT. Playing squash throughout my undergraduate years and during my MEng has been one of the most enjoyable experiences I've had at MIT. Outside of MIT Squash, PKT was the closest community I had, and I will forever cherish the lifelong friendships I made there. Living in PKT enabled me to grow in countless ways, and the list of friends who positively impacted my time there is too long to mention. However, I would like to give special thanks to my big, Shreyas Srinivasan, and my longest roommate within PKT, Nithin Parsan, for always being there for support and sharing amazing memories together.

Finally, I would like to express my heartfelt gratitude to my Mom and Brother, whose unwavering support, and encouragement have been the foundation of my success throughout my life. Their belief in me has been a constant source of motivation, and I am forever grateful for their presence in my life.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background on Inertial Navigation Systems (INS)

Navigation plays a crucial role in various domains, including aviation, robotics, and autonomous vehicles. Accurate and reliable navigation is essential to ensure the safety, efficiency, and effectiveness of these systems. Inertial Navigation Systems (INS) have emerged as a key technology to provide self-contained high-frequency navigation solutions in challenging environments where external references, such as GPS, may be unavailable or unreliable.

In scenarios where GPS signals are unavailable, Inertial Navigation Systems (INS) become crucial. An INS utilizes inertial sensors—specifically accelerometers and gyroscopes—to independently estimate the position, velocity, and orientation of a moving object. The key advantages of INS include its self-contained operation, high short-term accuracy, and the ability to continuously provide precise navigation data at rapid update rates.

The core of an INS is the Inertial Measurement Unit (IMU) along with a sophisticated computational unit. The IMU employs accelerometers and gyroscopes to measure the specific forces and angular rates experienced by the object and these measurements are critical for calculating changes in position, velocity, and orientation without external inputs. The computational unit then processes these data points through advanced navigation algorithms to generate accurate and reliable navigation states, enabling effective operation across diverse and challenging environments.

## 1.2 Problem Statement: INS Drift and Its Impact

Despite how useful INS can be in helping navigation, INS suffer from a critical limitation known as drift. INS drift refers to the accumulation of errors in the navigation solution over time as a result of the integration of sensor errors. The main causes of drift include biases, scale factor errors, and random noise in accelerometer and gyroscope measurements.

The magnitude of drift varies between different grades of inertial sensors. Low-cost IMUs can exhibit drift rates of several meters per minute, while high-end navigation-grade sensors have drift rates on the order of kilometers per hour. The accumulation of these errors leads to increasing uncertainty in the estimated position, velocity, and orientation of the object.

Consider an autonomous drone using INS for navigation in a forested environment where GPS signals are obstructed. Over the course of a prolonged mission, the drone experiences gradual drift from its intended flight path due to inaccuracies in its low-cost IMUs. This drift could lead to navigational errors that result in the drone becoming lost or colliding with obstacles, demonstrating the critical need for improved drift mitigation techniques.

INS drift has significant consequences in real-world applications. In aviation, drift can lead to navigational errors and potential safety hazards. In autonomous vehicles, drift can cause localization errors, leading to incorrect decision making and collisions. In mobile robotics, drift can result in inaccurate mapping and inefficient path planning. This is why it is very important to try to fix the drift problem since systems may not always have access to GPS.

## 1.3 Research Questions and Hypotheses

The primary research question addressed in this thesis is: Can Scientific Machine Learning (SciML) techniques outperform traditional methods in reducing INS drift? This question leads to several sub-questions:

- How effectively can SciML learn complex error patterns and adapt to changing conditions in INS?
- What are the potential advantages and limitations of SciML in INS drift reduction compared to traditional approaches?

The hypothesis is that the proposed SciML approach will reduce the INS drift by a significant percentage (e.g., X%) over a specific time period (e.g. Y hours) compared to traditional Kalman Filtering techniques. This hypothesis will be tested through extensive simulations and experiments, but before we get to that, let us explain the Proposed Solution and expalin what Scientific Machine Learning is.

## 1.4 Proposed Solution: Scientific Machine Learning (SciML) Approach

This thesis proposes a novel approach to addressing the challenge of INS drift through the application of Scientific Machine Learning (SciML). SciML is a fusion of traditional machine learning techniques and domain-specific scientific knowledge, aimed at creating models that not only predict but also explain by adhering to the physical laws governing the systems they model. In the context of INS drift, this involves integrating machine learning techniques such as neural networks with established navigational laws to enhance prediction accuracy and model interpretability.

In tackling INS drift, the SciML approach leverages neural networks alongside Extended Kalman Filtering (EKF). Neural networks excel at identifying complex patterns and anomalies in sensor data, which might be overlooked by conventional error modeling techniques. They are designed with multiple layers and non-linear activation functions that effectively capture the dynamic behavior of sensor errors.

The integration of neural networks with EKF brings together the strengths of adaptive learning and precise state estimation. EKF provides a robust framework for updating estimates of the system's state by optimally incorporating new measurements, while neural networks adaptively improve the model by learning from ongoing data. This synergy enhances the capability of INS to maintain accuracy in environments where GPS and other navigation aids are compromised.

This combined approach not only aims to reduce the drift in INS but also enhances the system's reliability and accuracy, making it highly suitable for challenging navigation environments. By applying SciML, this research explores new frontiers in navigation technology, aiming to significantly mitigate one of the most persistent challenges in inertial navigation.

## 1.5 Significance and Contributions of the Research

This research is significant because it tackles the pervasive issues of INS drift by leveraging the capabilities of Scientific Machine Learning (SciML). It not only seeks to overcome the constraints of traditional navigation systems, but also aims to set new benchmarks in the accuracy and reliability of INS. The primary contributions of this thesis include:

1. **Innovative Approach:** Development of a SciML-based methodology for INS drift reduction, showcasing superior performance over traditional error mitigation techniques.

2. **Enhanced Learning Capabilities:** Detailed analysis of how SciML can identify complex error patterns and dynamically adapt to changing operational conditions, providing deeper insights into the behavior of inertial sensors.

3. **Practical Impact:** Empirical evidence demonstrating the enhanced accuracy and robustness of INS systems using SciML, with implications for improving real-world navigation in sectors like aviation, robotics, and autonomous vehicles.

By pushing the boundaries of what is possible with INS technology, this thesis contributes to the broader field of navigation, offering pathways to more reliable and efficient systems that can operate effectively in GPS-denied environments.

## 1.6 Thesis Structure Overview

This thesis is organized into several chapters, each dedicated to different aspects of Inertial Navigation Systems (INS) and the application of Scientific Machine Learning (SciML) for drift reduction. The structure is as follows:

- **Chapter 1: Introduction** - This chapter provides an overview of the thesis, including the background on INS, the problem statement regarding INS drift, the research questions, the proposed SciML solution and the significance of the research.
- **Chapter 2: Basics of Inertial Navigation and Operational Principles** - Covers the foundational concepts of INS, including details about sensor characteristics, navigation equations, and the sources of errors that affect INS performance.
- **Chapter 3: Redefining INS with Scientific Machine Learning** - This chapter introduces the key principles and techniques of SciML relevant to this research, including Neural Networks and Kalman Filtering.
- **Chapter 4: Simulation for INS Research** - Describes the simulation environment and the tools and techniques used to generate synthetic INS data.
- **Chapter 5: SciML Methodology for INS Drift Reduction** - Presents the detailed SciML approach and architecture developed to reduce INS drift, including the specific architectures of the system.

- **Chapter 6: Experimental Results and Analysis** - Demonstrates the effectiveness of the SciML approach through extensive simulations and experimental results. This chapter also provides a comparative analysis with traditional error mitigation methods.

- **Chapter 7: Conclusion** - Summarizes the research, its key findings, and the broader impact on advancing INS technologies. This chapter also identifies open questions and suggests areas for further investigation, inspiring future research directions in the field of intelligent navigation systems.

# Chapter 2

# Basics of Inertial Navigation and Operational Principles

## 2.1 Introduction to INS

An Inertial Navigation System (INS) is a self-contained navigation system that determines the position, velocity, and orientation of a moving object without relying on external references[1]. As I said earlier, the core operational principle of an INS is based on the concept of dead reckoning[2], which involves integrating the accelerations and angular rates measured by inertial sensors to track the object's motion relative to a known starting point.

The main components of an INS include:

1. Inertial Measurement Unit (IMU)(Figure A.1): An IMU consists of a triad of accelerometers and a triad of gyroscopes. Accelerometers measure the specific force (acceleration due to motion and gravity) along three orthogonal axes, while gyroscopes measure the angular rates about the same three axes.

2. Computational Unit: The computational unit, often a microprocessor or a dedicated navigation computer, processes the raw sensor data from the IMU using navigation algorithms. Perform coordinate transformations, integrates acceleration and angular

rates, and applies corrections based on sensor calibration parameters and error models.

3. Initialization and Alignment Unit: This unit provides the initial position, velocity, and orientation of the INS, which serve as the starting point for the dead reckoning process. Initial alignment can be achieved through various methods, such as manual input, external aiding (e.g. GPS), or gyrocompassing.

The main advantages of INS include their self-contained nature, high short-term accuracy, and the ability to provide continuous navigation information at high update rates[3]. However, the accuracy of INS tends to drift over time due to the accumulation of sensor errors, which requires periodic updates or integration with other navigation systems for long-term accuracy.

In the following sections, we will dive deeper into the components of INS, including inertial measurement units (IMUs), their error characteristics, and the operational principles of INS, such as dead reckoning, initialization, and alignment. We will also discuss error modeling and compensation techniques to mitigate the impact of sensor errors on the navigation solution.

## 2.2 Inertial Measurement Units (IMUs)

As discussed previously, an Inertial Measurement Unit (IMU) consists of a set of three accelerometers and three gyroscopes, each rigidly mounted to measure specific force and angular rate along three orthogonal axes: $x^b$, $y^b$, and $z^b$. The configuration of these sensors allows for precise measurements critical in the process of dead reckoning, detailed further in subsequent sections. The schematic representation below illustrates the orientation and alignment of accelerometers and gyroscopes within an IMU[4].

The accelerometers measure specific force, which includes both gravitational acceleration and acceleration due to motion, while the gyroscopes measure the angular rate relative to an inertial frame. These measurements were combined to track the object's movement in a three-dimensional space accurately.

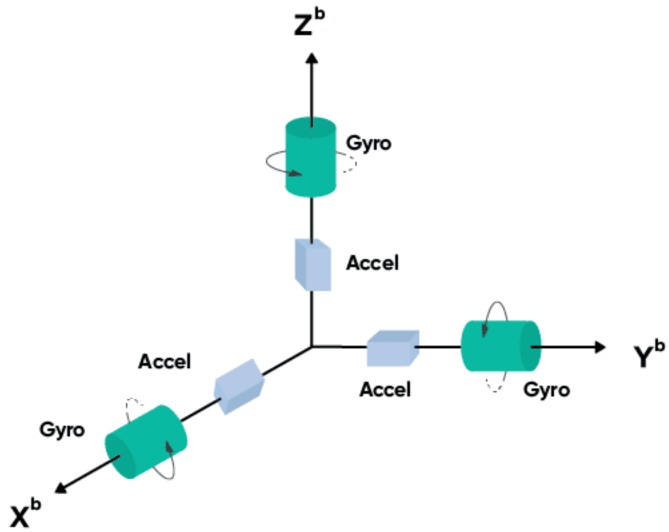


Figure 2.1: Conceptual diagram of an Inertial Measurement Unit (IMU) with three accelerometers (blue cubes) and three gyroscopes (green cylinders), oriented along the $x^b$, $y^b$, and $z^b$ axes.

### 2.2.1 Accelerometers

Accelerometers are a key component of an INS, measuring the specific force acting on the moving object. This includes both the linear acceleration due to the object's motion and the acceleration due to gravity.

In an INS, the accelerometer trio measures the accelerations along the three orthogonal axes (x,y,z) of the body frame. These acceleration measurements are integrated once to obtain the velocities and again to derive the position information of the object in the body frame[5].

Now, there are two main challenges associated with accelerometers in an INS:

- **Distinguishing Accelerations:** Accelerometers cannot inherently distinguish between linear accelerations caused by motion and the constant acceleration due to gravity. This poses a challenge for INS, as the acceleration measurements must be properly decomposed into their inertial (motion-induced) and gravitational components to accurately track the object's movement.

- **Integration Drift:** The integration process used to obtain velocities and positions from accelerometer measurements is susceptible to drift over time due to sensor errors and noise. This drift can lead to significant errors in position and velocity estimates if not properly addressed.

To mitigate these challenges, accelerometer calibration and compensation techniques are applied within the INS algorithms. This includes estimating and removing biases, applying scale factor corrections, and filtering noise through algorithmic means such as Kalman filtering. I will walk through how we can use SciML techniques to solve this problem in Chapter 3.

Furthermore, the accelerometer measurements must be rotated (resolved) from the body frame to the navigation frame using the attitude information derived from the gyroscopes. This step is crucial for consistently integrating the accelerations in the correct navigation frame and accurately tracking the object's motion. More details on the proper navigation frames are provided in Section 2.3.2.

Like gyroscopes, accelerometers are subject to similar error sources such as bias, scale factor, and noise. Proper calibration and compensation techniques are essential to compensate for these error sources and ensure accurate navigation performance.

### 2.2.2 Gyroscopes

Gyroscopes are another essential component of an INS, measuring the angular rate (i.e., how quickly the object rotates around its axes). This information is crucial for determining the object's changing orientation over time.

In an INS, the gyroscope triad measures the rotation rates about the three orthogonal axes (x, y, z) of the body frame. These angular rate measurements are integrated to track the changing attitude (orientation) of the object relative to a reference frame, typically an inertial navigation frame.

There are two main reasons why the information from gyroscopes is vital in an INS:

- **Resolving Accelerations:** The accelerations measured by the accelerometers are initially expressed in the body frame. To properly integrate these accelerations and obtain velocities and positions in a navigation frame, the accelerations must be rotated (resolved) from the body frame to the navigation frame using the attitude information derived from the gyroscopes. This process ensures that the accelerations are correctly represented in the desired reference frame for accurate navigation.

- **Tracking Orientation Changes:** As the object maneuvers, its orientation relative to the navigation frame changes continuously. The gyroscope measurements enable the INS to account for these orientation changes, ensuring that the accelerations are consistently resolved and integrated in the correct navigation frame. By tracking the object's attitude, the INS can maintain an accurate representation of its motion in three-dimensional space.

However, like accelerometers, gyroscopes are subject to various error sources, such as bias, scale factor, and noise. These errors can accumulate over time, leading to drift in the attitude estimates. Proper calibration and compensation techniques are essential to mitigate these error sources and maintain accurate orientation tracking.

Bias refers to a constant offset in the gyroscope measurements, while scale factor errors cause a proportional deviation in the angular rate readings. Noise, on the other hand, introduces random fluctuations in the measurements. These errors, if left uncompensated, can result in significant drift in the attitude estimates over time.

To address these challenges, advanced calibration and compensation techniques are employed within the INS algorithms. These techniques aim to estimate and correct for the gyroscope errors, minimizing their impact on the navigation solution. We will delve into these techniques in more detail in Chapter 3, where we will explore various approaches to mitigate the effects of gyroscope errors and enhance the overall accuracy of the INS.

In summary, gyroscopes play a vital role in an INS by providing the necessary attitude information to resolve accelerations and track orientation changes. However, gyroscope errors pose significant challenges that must be addressed through proper calibration and compensation techniques. By effectively managing these errors, the INS can maintain accurate attitude estimates and ensure reliable navigation performance.

## 2.3 INS Operational Principles

The operational principles of an Inertial Navigation System (INS) are based on the concept of dead reckoning, which involves the continuous integration of acceleration and angular rate measurements to determine the position, velocity, and orientation of a moving object relative to a known starting point. The INS uses measurements from the Inertial Measurement Unit (IMU), consisting of accelerometers and gyroscopes, to perform this integration process.

To understand the inner workings of an INS, it is essential to explore the mathematical basis of dead reckoning, the strapdown mechanization algorithms, and the impact of noise on the system's performance. In addition, the initialization and alignment processes play a crucial role in establishing the initial conditions and reference frames for accurate navigation.

In the following subsections, we will delve into the details of dead reckoning and strapdown mechanization, examining the velocity and position update equations, error propagation, and the challenges posed by noise. We will also discuss the importance of initialization and alignment in INS operations and the methods used to achieve them.

### 2.3.1 Dead Reckoning and Strapdown Mechanization

Building upon the basic concepts introduced in the previous section, we will now delve deeper into the math and key steps that govern the operation of Inertial Navigation Systems (INS)[6][7]. The core principle of an Inertial Navigation System (INS) is dead reckoning, which involves the continuous integration of acceleration and angular rate measurements to determine the position, velocity, and orientation of a moving object relative to a known starting point. The mathematical basis of dead reckoning relies on the following velocity and position update equations: The velocity at a future time $t + T$ is calculated by integrating the acceleration $a$ over the time interval from $t$ to $t + T$ and adding this to the velocity at time $t$, as shown in the equation:

$$v(t+T) = \int_{t}^{t+T} a \cdot dt + v(t) \tag{2.1}$$

Similarly, the position at time $t + T$ is determined by integrating the velocity $v$ over the same interval and adding it to the position at time $t$ :

$$p(t+T) = \int_{t}^{t+T} v \cdot dt + p(t) \tag{2.2}$$

In these formulas, $v(t+T)$ and $p(t+T)$ represent the velocity and position in the future $t + T$, respectively. $v(t)$ is the known velocity at the current time $t$, and $p(t)$ is the current position. This method updates each new position and velocity based on its previous state, using the measured accelerations and velocities.

The paths we are integrating are discretized because a real inertial measurement unit will give discrete sampled values, commonly between one hundred and six hundred hertz. But the main problem with dead reckoning and IMU measurements is the noise within the system.

If there is any amount of noise, $n$, in our sensor readings from the inertial measurement unit,

$$v(t+T) = \int_t^{t+T} (a + n) \cdot dt + v(t) \tag{2.3}$$

our calculated position will begin to diverge from the truth (Fig. 2.2).

This propagation of errors is a significant challenge in INS, as it causes an unbounded drift in the navigation solution over time.

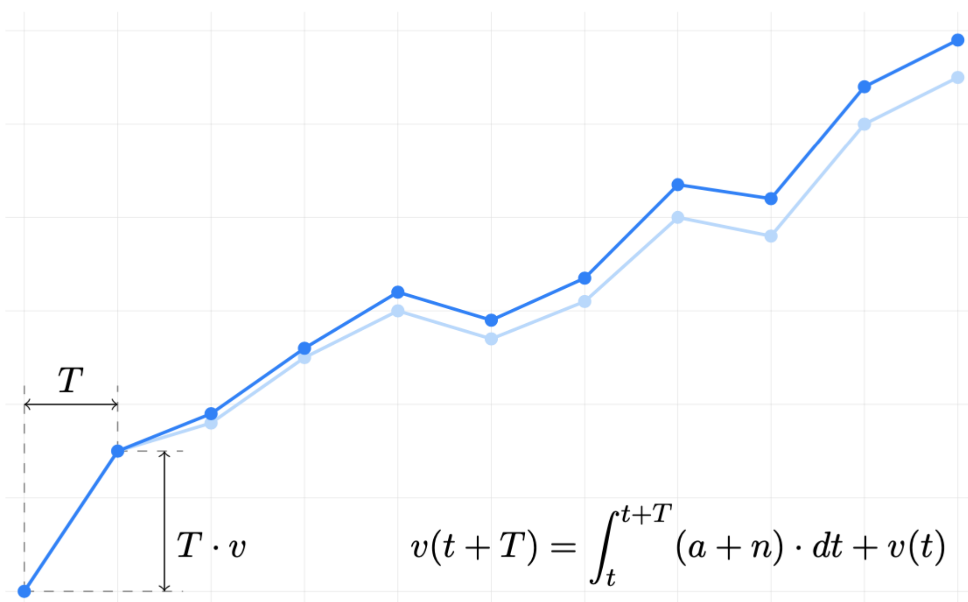


Figure 2.2: Dead reckoning (path integration) in one dimension. The light blue shows the correct path without noise. The dark blue shows the calculated path with noise. The noise error, which we are notating here as $n$, accumulates.

### 2.3.2 Navigation Frames and Coordinate Transformations

In an Inertial Navigation System (INS), the choice of navigation frames and the ability to perform coordinate transformations are fundamental to the accurate representation and processing of motion data. Navigation frames provide a reference for expressing the position, velocity, and orientation of the INS, while coordinate transformations allow the conversion of measurements and quantities between different frames.

Below is a table summarizing the commonly used navigation frames in INS: the Earth-Centered Earth-Fixed (ECEF) frame, the Sensor frame, and the Body frame. Each frame is accompanied by a reference to its illustrative image.

| **Frame** | **Image Reference** | **Description** |
|---|---|---|
| ECEF | Figure A.3 | The Earth-Centered Earth-Fixed (ECEF) Frame is a global reference frame fixed relative to the Earth with its origin at the Earth's center of mass. Axes are aligned as follows: X-axis points towards the prime meridian, Y-axis points eastward, and Z-axis aligns with the Earth's rotation axis. Used for global positioning and satellite tracking. |
| Sensor | Figure A.4 | The Sensor Frame is a type of reference frame that is fixed to the sensor. On VectorNav sensors, the sensor frame is aligned as shown in the referenced figure. The individual sensor element measurement axes within the instrument are aligned to the sensor frame as part of the calibration process. The "Misalignment" or "Alignment Error" specification indicates how closely the measurement axes are aligned with the indicated sensor frame. |
| Body | Figure A.5 | The Body Frame is fixed to the moving platform where the INS is mounted. Axes are aligned with the platform's forward direction, right side, and downward direction, forming a right-handed coordinate system. Essential for onboard control and dynamics analysis. |

Table 2.1: Overview of Navigation Frames Used in INS

Coordinate transformations between these frames are facilitated by rotation matrices or Direction Cosine Matrices (DCMs), which describe the orientation of one frame relative to another. These matrices are critical for maintaining the accuracy and performance of INS systems. For instance, the DCM from the body frame to the navigation frame, denoted as

$C_b^n$, is used to transform acceleration measurements from the body frame to the navigation frame:

$$a^n = C_b^n \cdot a^b \tag{2.4}$$

where $a^n$ is the acceleration vector in the navigation frame, $a^b$ is in the body frame, and $C_b^n$ represents the DCM from the body to the navigation frame.

DCMs are 3x3 orthogonal matrices that fulfill orthogonality and unit determinant properties. They can be parameterized using Euler angles, quaternions, or rotation vectors, with quaternions often preferred in INS due to their computational efficiency and non-susceptibility to the gimbal lock issue associated with Euler angles.

Further, the accurate determination and regular maintenance of DCMs are crucial for optimal INS performance. These matrices are typically initialized using the initial alignment process and continuously updated with angular rate measurements from gyroscopes. This update process involves integrating these angular rates and applying necessary corrections for phenomena like Earth's rotation and transport rate.

Errors in coordinate transformations, arising from factors such as misalignment between the IMU and the platform or errors in initial alignment, can significantly degrade navigation accuracy. To mitigate these errors, advanced techniques like Kalman filtering and error modeling are employed, enhancing the reliability and precision of the INS.

In conclusion, navigation frames and their accurate coordinate transformations constitute core components of an INS, enabling consistent representation and integration of motion data. The maintenance of precise DCMs is essential, not only for accurate data transformation but also for the overall reliability and functionality of navigation systems.

## 2.4 Error Modeling and Compensation Techniques

As I have discussed throughout this chapter, we have seen that error comes in various ways in Inertial Navigation. To further illustrate this problem here is a real-life example:

Consider an autonomous firefighting robot equipped with an inertial navigation system. As the robot navigates through a burning building, it deploys a water hose that introduces several sources of noise into its guidance system. The recoil of the water pumping out of the nozzle causes vibrations in the hose that translate to the robot itself, creating erroneous readings in the IMU. Additionally, the hose dragging on the ground or getting caught on debris creates friction and resistance forces that subtly change the robot's path but are not fully registered by the IMU. Finally, as the water-filled hose gets lighter, the IMU's acceleration readings become less reliable because of the changing mass. These types of noise can cause the inertial navigation system's calculations to become inaccurate, leading to errors in the robot's estimated position and orientation.

Now that we can see how large of a problem error can be let's briefly go over what this error looks like and then a more in-depth view of ways we can mitigate the affects of error in IMUs.

### 2.4.1 Error Propagation

The errors in the inertial sensor measurements, such as biases, scale factor errors, and noise, accumulate over time due to the integration process in the INS mechanization equations, as we saw in equation 2.3. For example, a constant bias in the accelerometer measurements will result in a linearly growing error in velocity and a quadratically growing error in position. Similarly, a constant bias in the gyroscope measurements will cause a linearly growing error in orientation, which in turn affects the transformation of accelerations from the body frame to the navigation frame.

The propagation of error in an INS can be modeled using error state equations, which describe the evolution of errors in navigation states (position, velocity, and orientation) as functions of errors in inertial sensor measurements. These error state equations form the basis for many error compensation techniques, such as Kalman filtering, which aim to estimate and correct errors in the navigation solution.

### 2.4.2 Kalman Filtering Overview

Kalman filtering is a powerful technique widely used in INS for estimating and compensating for the errors in the navigation states. The Kalman filter is a recursive algorithm that combines knowledge of the system dynamics, sensor measurements, and their associated uncertainties to provide an optimal estimate of the system states [8].

In the context of INS, the Kalman filter is used to estimate the errors in the navigation states (position, velocity, and orientation) and inertial sensor errors (biases, scale factors, and misalignments). The filter operates by performing two main steps: prediction and update.

In the prediction step, the filter uses the INS mechanization equations and the error state equations to propagate the estimated navigation states and their associated uncertainties forward in time. This step relies on the knowledge of the system dynamics and the statistical properties of the inertial sensor errors.

In the update step, the filter incorporates external measurements, such as GPS position and velocity, to correct the predicted navigation states and sensor errors. The external measurements are compared with the predicted measurements, and the difference (known as the measurement residual) is used to update the estimated states and their uncertainties. The update step helps to limit the growth of errors in the navigation solution and maintains the accuracy of the INS over time.

Figure A.2 [9] illustrates the Kalman filtering process in practice, showing the predict, measure, and update steps.

Kalman filtering is a vast topic, and its application to INS involves various aspects, such as the choice of the error state model, the tuning of the filter parameters, and the handling of different types of external measurements. A deeper discussion of Kalman filtering can be found in Section 3.3.

# Chapter 3

# Redefining INS with Scientific Machine Learning

As INS technologies grapple with inherent limitations such as error accumulation and reliance on external aids, SciML emerges as a potent solution capable of redefining traditional methodologies. This chapter lays the groundwork for SciML in INS that leverages the capabilities of machine learning infused with rigorous physical laws.

## 3.1 Motivation for SciML in INS

This section explores the compelling reasons behind integrating SciML into INS frameworks, highlighting the deficiencies of conventional error mitigation strategies and illustrating how SciML's advanced capabilities can lead to more reliable, adaptive, and robust navigation systems. The insights provided here set the stage for a deeper appreciation & understanding of the methodology & results presented later on in Section 5 and Section 6.

### 3.1.1 Limitations of Traditional INS Error Mitigation

Inertial Navigation Systems (INS) have been instrumental in providing autonomous guidance for various applications, from aviation to autonomous vehicles. This subsection explores the significant limitations of conventional INS error mitigation techniques, shedding light on why advancements such as Scientific Machine Learning (SciML) are necessary for the evolution of navigation technologies. Here are just a few major issues with traditional INS error mitigation:

**Simplified Error Models:** Traditional error mitigation in INS is largely based on models that simplify the complex dynamics of sensor errors. These models typically assume that errors such as biases and scale factor inaccuracies remain constant or vary linearly over time. Such assumptions do not account for the stochastic and non-linear nature of real-world sensor errors, which can fluctuate due to a variety of factors such as temperature changes, mechanical stresses, or electromagnetic interference. The inadequacy of these models becomes particularly evident in scenarios requiring high precision over long durations, where even minor unmodeled error dynamics can accumulate, leading to significant drifts in navigation accuracy.

**Dependence on External Aiding:** A common practice to counteract the inherent drift in INS is to integrate external aiding sources such as GPS or other sensors. Although effective in reducing long-term drift, this reliance introduces new vulnerabilities, including dependency on the availability and integrity of the external signals. In environments where GPS signals are weak or spoofed, such as urban canyons or under dense foliage, the INS might be deprived of necessary corrections, thus degrading its reliability. In addition, the integration of these external sources often adds complexity to the system architecture, increasing the potential for failure points and the computational load.

**Inflexibility to Sensor Variability:** Conventional INS systems utilize static error models calibrated under specific conditions. These models are not designed to adapt to changes in sensor characteristics that may occur due to aging, wear, or environmental changes. As a result, the system's ability to accurately compensate for errors degrades unless frequent recalibrations are performed, a process that is often impractical during normal operations. This inflexibility can be detrimental in applications where long-term deployment without maintenance is essential, such as in space navigation or underwater explorations.

These limitations highlight the need for more sophisticated approaches capable of addressing the dynamic and complex nature of INS errors. The next section explores how Scientific Machine Learning (SciML) not only addresses these issues but also offers enhanced adaptability and robustness, making it a formidable successor to traditional methods in the field of inertial navigation.

### 3.1.2 Advantages of SciML Approaches

Traditional INS error mitigation techniques rely on mathematical models to characterize and compensate for inertial sensor errors. These models, such as bias and scale factor models, attempt to capture the systematic and stochastic behavior of the errors. However, these approaches have limitations that can lead to suboptimal error compensation and reduced INS accuracy.

Firstly, traditional error models often make simplifying assumptions about the nature of sensor errors, assuming constant biases or linear scale factor errors. In reality, sensor errors exhibit complex, nonlinear, and time-varying characteristics that are difficult to accurately model using conventional techniques. Secondly, traditional approaches often rely on external aiding sources, such as GPS, to mitigate long-term INS drift. While providing periodic corrections, these sources introduce vulnerabilities, as GPS signals can be jammed, spoofed, or blocked in certain environments. Lastly, adapting pre-calibrated error models to changing operational conditions or aging sensors requires frequent recalibration, which can be impractical in many applications.

In light of these limitations, Scientific Machine Learning (SciML) has emerged as a promising paradigm that combines data-driven learning with the incorporation of physical knowledge. SciML techniques offer several advantages for INS error mitigation:

Data-driven learning allows SciML models to capture complex, nonlinear error patterns directly from sensor data. By training on large datasets of inertial sensor measurements, these models can learn intricate error patterns and correlations that may be difficult to model explicitly using traditional techniques. This leads to a more accurate representation of sensor errors and improved INS accuracy.

One of the key components of SciML for INS error mitigation is the use of neural networks. Neural networks are powerful machine learning models that can learn complex, nonlinear relationships between inputs and outputs. In the context of INS error mitigation, neural networks can be trained on large datasets of inertial sensor measurements to learn the intricate error patterns and correlations present in the data. By leveraging the ability of neural networks to capture these complex relationships, SciML models can provide more accurate and robust error compensation compared to traditional techniques.

The flexibility and adaptability of neural networks make them particularly well-suited for handling the time-varying and dynamic nature of sensor errors. Neural networks can be designed with various architectures and hyperparameters to optimize their performance for specific INS applications. For example, recurrent neural networks (RNNs) can be employed to capture the temporal dependencies in sensor error patterns, while convolutional neural networks (CNNs) can be used to extract spatial features from multi-axis sensor data. The choice of neural network architecture and training methodology plays a crucial role in the effectiveness of SciML models for INS error mitigation.

SciML models have the potential to adapt to changing sensor characteristics and operational conditions without requiring explicit recalibration or manual intervention. By continuously learning from incoming sensor data, these models can autonomously adjust their parameters to capture evolving error patterns. This adaptability is particularly valuable in scenarios where sensor errors drift over time or when the INS operates in dynamic environments with varying conditions. The motivation for applying SciML techniques to INS error mitigation stems from their potential to overcome the limitations of traditional approaches and deliver more accurate, robust, and adaptable solutions. By leveraging data-driven learning while incorporating physical knowledge, SciML offers a promising avenue for advancing INS performance and enabling more reliable navigation in challenging scenarios.

## 3.2 Neural Networks in Sensor Error Calibration

Neural networks have emerged as a powerful tool in the field of Scientific Machine Learning (SciML), offering new possibilities for enhancing the accuracy and reliability of Inertial Navigation Systems (INS). By leveraging the ability of neural networks to learn complex patterns and relationships from data, we can develop more sophisticated and adaptive approaches for sensor error calibration. In this section, we will explore the role of neural networks in INS, focusing on their potential to improve error modeling and correction. We will discuss the advantages of neural networks over traditional calibration methods and delve into the specific architectures and configurations that are well-suited for INS applications. By understanding the strengths and adaptability of neural networks, we can harness their power to mitigate sensor errors and enhance the overall performance of inertial navigation systems.

### 3.2.1 Neural Networks in INS

Using neural networks in the field of Inertial Navigation Systems (INS) marks a shift towards more data-driven approaches for sensor error calibration. As a core component of Scientific Machine Learning (SciML), neural networks bring to the table an unparalleled ability to discern and model complex patterns in data, which traditional statistical methods might miss or oversimplify. This section introduces the rationale behind integrating neural networks into INS, emphasizing their potential to enhance the system's accuracy and reliability through advanced error modeling and correction.

Neural networks, with their deep learning capabilities, are particularly adept at handling the nonlinear and complex error characteristics inherent in inertial sensors. Unlike traditional error mitigation techniques, which often rely on simplified assumptions and static models, neural networks learn directly from the data. This learning enables them to adapt to the intricacies and dynamics of sensor errors, offering a more nuanced understanding and correction of these errors.

Moreover, the flexibility of neural networks allows them to be tailored specifically to the needs of INS. This dynamic adaptability is crucial for applications where INS must perform reliably over long periods or in varying environments.

In comparing neural networks to traditional calibration methods, several advantages become clear:

- **Dynamic Error Modeling**: Neural networks continually refine their understanding of sensor errors as they process more data. This ongoing learning process contrasts sharply with traditional methods, which typically use fixed parameters that may not reflect changes in sensor behavior over time or under different environmental conditions.

- **Generalization Capabilities**: Well-trained neural networks can generalize from their training data to handle unseen scenarios, making them more robust to the kinds of novel situations that often arise in real-world navigation tasks.

- **Integration Flexibility**: Neural networks can be seamlessly integrated into existing INS frameworks, enhancing their error correction capabilities without necessitating a complete overhaul of the system.

This introduction sets the stage for a deeper exploration into the specific architectures and configurations of neural networks that are optimized for INS applications. Following sections will detail the network structures, training processes, and implementation strategies that harness the full potential of neural networks to recalibrate and enhance the accuracy of inertial navigation systems.

### 3.2.2 Neural Network Architecture

The architecture of neural networks used in sensor error calibration for Inertial Navigation Systems (INS) is crucial for effectively processing and learning from complex sensor data. Figure 3.1 illustrates a typical neural network architecture used in this context, designed to handle the nonlinear and complex error characteristics inherent in inertial sensors.

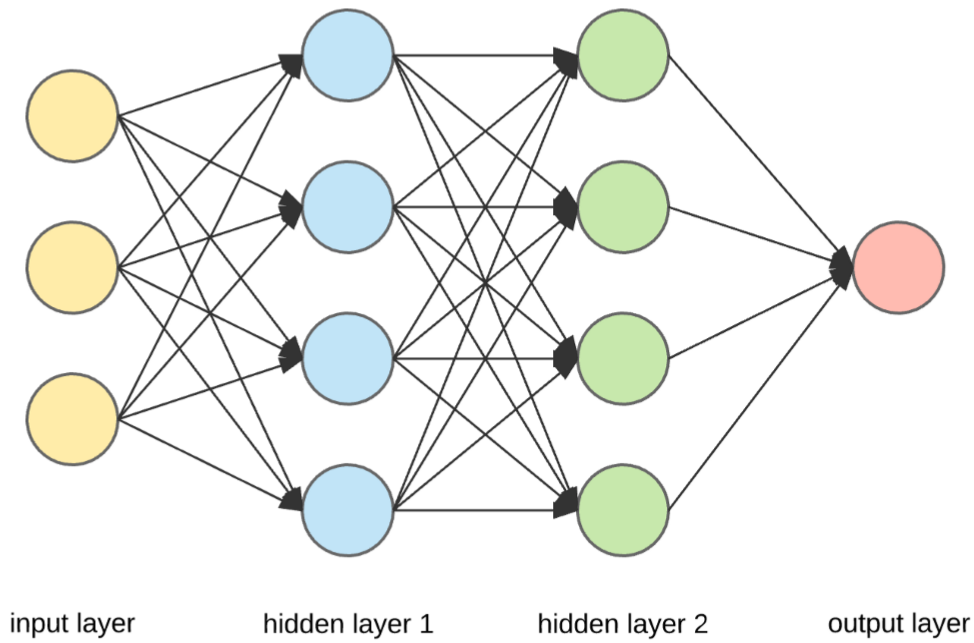


Figure 3.1: Example of a neural network architecture used in sensor error calibration for INS. This network includes an input layer, multiple hidden layers, and an output layer.

**Basic Architecture Overview:** Neural networks for INS typically consist of several components, each playing a critical role in processing sensor data:

- **Input Layer**: This layer receives raw data from sensors, such as accelerometer and gyroscope readings. It distributes the data to subsequent layers without modification.
- **Hidden Layers**: As shown in Figure 3.1, the hidden layers perform the majority of computational tasks, processing features extracted from the input data. These layers use activation functions like ReLU (Rectified Linear Unit) or tanh (hyperbolic tangent) to introduce non-linearity, enabling the network to capture complex patterns.

- **Output Layer**: The final layer outputs the error corrections or calibrated data. Depending on the calibration task, a linear activation function may be employed for regression outputs, which are typical in error calibration scenarios.

**Example Configuration:** A typical configuration for a neural network in INS applications might include:

- **Input Layer**: Takes three-dimensional data from each sensor (e.g., x, y, z coordinates from an accelerometer).
- **First Hidden Layer**: Comprises 100 neurons with ReLU activation.
- **Second Hidden Layer**: Consists of 50 neurons, also with ReLU activation.
- **Output Layer**: Utilizes a linear activation function to output continuous values that represent the corrected sensor errors.

Adjustments to this architecture can be made based on specific needs, such as the complexity of error patterns and available computational resources. The flexibility and adaptability of neural networks make them highly effective for dynamic error modeling and correction in INS.

This configuration is just a starting point. The actual architecture can be adjusted based on specific requirements such as the complexity of the sensor error characteristics and the computational resources available.

**Adaptations for INS:** In the context of INS, neural networks might incorporate specific adaptations to better handle the characteristics of sensor data:

- **Recurrent Neural Networks (RNNs)**: For time-series data like that from INS sensors, RNNs or their more advanced variants like LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Units) networks are often used. These models are capable of capturing temporal dependencies and dynamics in the sensor data, which are crucial for accurate error modeling.

- **Convolutional Neural Networks (CNNs)**: Although more commonly associated with image data, CNNs can also be applied to time-series sensor data. They can extract spatial-temporal features from multichannel sensor data, which can be particularly useful in complex dynamic environments.

**Training and Tuning:** The effectiveness of a neural network significantly depends on proper training and parameter tuning. This involves selecting the right number of layers and neurons, choosing appropriate activation functions, and setting the hyperparameters like the learning rate and the number of epochs. Moreover, techniques such as cross-validation and regularization (e.g., dropout, L2 regularization) are essential to prevent overfitting and ensure that the network generalizes well to new, unseen data.

The architecture described here provides a robust framework for developing neural network models capable of addressing the specific challenges posed by sensor error calibration in INS. By leveraging such advanced architectures, INS can achieve higher accuracy and reliability, critical for applications ranging from autonomous vehicles to aerospace navigation.

## 3.3 Extended Kalman Filter for State Estimation

In this section, we will further investigate what was talked about earlier in Section 2.4. We know that the Extended Kalman Filter (EKF) is an extension of the Kalman Filter (KF), which is optimal for linear systems with Gaussian noise. What Extended Kalman Filtering provides over traditional Kalman Filtering is that it can process nonlinear systems due to the addition of the Taylor series expansion module. With this addition EKF is often used in state estimation problems for linear systems with unknown parameters, making it ideal for solving estimation issues in SLAM, Navigation suystems, and GPS.

In the following sections I will briefly go over the inner workings of EKF and what it means for INS. If you would like to investigate the math behind EKF you can find further details in these papers [8] and [10].

### 3.3.1 Extended Kalman Filter Overview

The Extended Kalman Filter is a powerful tool for state estimation in nonlinear systems, making it particularly well-suited for INS applications. The EKF extends the traditional Kalman Filter by linearizing the nonlinear system dynamics and measurement models around the current state estimate using a first-order Taylor series expansion. This linearization allows the EKF to handle the nonlinearities present in INS, such as the nonlinear relationship between the sensor measurements and the system states.

The EKF operates in a two-step process: prediction and update. In the prediction step, the filter uses the system model to predict the state estimate and its associated uncertainty (covariance) forward in time. The update step occurs when new measurements become available, and the filter uses these measurements to correct the predicted state estimate and covariance.

### 3.3.2 EKF Formulation for INS

To apply the Extended Kalman Filter (EKF) to an Inertial Navigation System (INS), we need to define two key components: the system dynamics model and the measurement model.

The system dynamics model describes how the states of the system, such as position, velocity, and orientation, change over time based on the measurements from the inertial sensors (accelerometer and gyroscope). This model is nonlinear due to the complex relationships between the sensor measurements and the system states.

The measurement model relates the observed measurements from external sensors, such as GPS, magnetometer, or other aiding sensors, to the system states. This model is also nonlinear because the relationship between the measurements and the states is often complex and cannot be described by simple linear equations.

To handle these nonlinearities, the EKF linearizes the system dynamics and measurement models around the current state estimate using a first-order Taylor series expansion.

This linearization process involves computing Jacobian matrices, which are essentially the derivatives of the nonlinear functions with respect to the system states.

The EKF operates in two main steps: prediction and update. In the prediction step, the filter uses the linearized system dynamics model to predict the state estimate and its associated uncertainty (covariance) forward in time. This step takes into account the inertial sensor measurements and any control inputs applied to the system.

When new measurements from external sensors become available, the EKF performs the update step. It uses the linearized measurement model to correct the predicted state estimate and covariance based on the difference between the actual measurements and the predicted measurements. This correction is weighted by the Kalman gain, which determines how much trust the filter places in the new measurements compared to the predicted state estimate.

The EKF iteratively repeats the prediction and update steps, continuously refining the state estimate and its covariance as new sensor measurements arrive. This process allows the EKF to effectively estimate the system states and their uncertainties, even in the presence of nonlinear system dynamics and measurement models.

By applying the EKF to INS, we can obtain a more accurate and reliable estimate of the system states, such as position, velocity, and orientation, by fusing information from multiple sensors and accounting for the nonlinearities inherent in the system. This improved state estimate can then be used for various applications, such as navigation, guidance, and control of vehicles or robots equipped with inertial sensors.

### 3.3.3 Benefits and Limitations

The Extended Kalman Filter offers several benefits for state estimation in INS:

1. It can handle the nonlinear system dynamics and measurement models present in INS.

2. It provides a recursive framework for estimating the system states and their uncertain-

ties.

3. It allows for the fusion of multiple sensor measurements to improve the overall state estimate.

However, the EKF also has some limitations:

1. The linearization of the nonlinear models can introduce errors, especially if the system is highly nonlinear or the state estimate is far from the true state.

2. It assumes that the process and measurement noises are Gaussian, which may not always be the case in real-world scenarios.

3. It can be computationally expensive, particularly for high-dimensional state spaces or frequent measurement updates.

Despite these limitations, the Extended Kalman Filter remains a widely used and effective technique for state estimation in INS. Its ability to handle nonlinearities and fuse multiple sensor measurements makes it a valuable tool for improving the accuracy and reliability of inertial navigation systems.

In summary, the Extended Kalman Filter is a powerful extension of the traditional Kalman Filter that enables state estimation in nonlinear systems, such as Inertial Navigation Systems. By linearizing the system dynamics and measurement models around the current state estimate, the EKF can effectively estimate the system states and their uncertainties, while incorporating multiple sensor measurements. While the EKF has some limitations, such as the potential for linearization errors and the assumption of Gaussian noise, it remains a widely used and valuable technique for enhancing the performance of INS.

# Chapter 4

# Simulation for INS Research

Simulation plays a vital role in the development and evaluation of Scientific Machine Learning (SciML) techniques for Inertial Navigation Systems (INS) drift reduction. Using simulated environments, researchers can generate diverse datasets, control error sources, and explore challenging scenarios that may be difficult to replicate in real-world experiments. This chapter focuses on the rationale for adopting a simulation-focused approach, the advantages it offers, and the challenges associated with real-world datasets. We will also delve into the specifics of our simulation environment, including the tools and software used, the approach to realistic physics modeling, and the process of simulated data generation.

## 4.1 Rationale for Simulation-Focused Approach

In this section, we will explore the reasons behind choosing a simulation-focused approach to develop SciML techniques for INS drift reduction. We will discuss the advantages of simulation, such as the ability to control error sources, generate large and diverse datasets, and investigate challenging scenarios. In addition, we will highlight the limitations and complexities of real-world datasets, which further motivate the use of simulations in the research and development phase.

### 4.1.1 Advantages of Simulation

Simulation offers several key advantages in the research and development of Scientific Machine Learning (SciML) techniques for Inertial Navigation Systems (INS). Firstly, simulations allow researchers to precisely manipulate sensor biases and noise characteristics, providing control over error sources and their magnitudes. This control is critical, as described in Chapter 2, where sensor errors such as biases and noise in accelerometers and gyroscopes are detailed, highlighting their impact on INS accuracy.

Secondly, simulations enable the generation of large and diverse datasets, which are crucial for training and validating SciML techniques. Simulations provide a means of generating large amounts of synthetic data that cover a wide range of trajectories, motion profiles, and error conditions. This ability to create extensive and varied datasets is essential for improving the robustness and generalization capabilities of SciML models. By training and testing on a rich set of simulated examples, researchers can develop models that are better equipped to handle real-world variations and uncertainties.

Although the ultimate goal is to transfer and validate these models on real-world hardware, the simulation-focused approach offers a valuable foundation for understanding the behavior and performance of SciML techniques in a controlled and repeatable environment.

### 4.1.2 Challenges of Real-World Datasets

While real-world datasets are essential for validating and fine-tuning SciML models, they present several limitations and complexities that motivate the use of simulations in the research and development phase. One major challenge is the limited availability of high-quality labeled INS data. Collecting real-world high-quality data requires expensive equipment, time-consuming experiments, and precise ground truth measurements, which can be difficult to obtain in practice.

Real-world datasets often suffer from sensor misalignments, time synchronization issues, and unexpected behaviors in dynamic environments. Sensor misalignments occur when the sensors are not perfectly aligned with the vehicle's axes, introducing errors in the INS measurements that are difficult to identify and compensate for during data collection. Time synchronization between different sensors and the ground truth system is another challenge, as even small timing errors can lead to significant inaccuracies in the INS solution.

To illustrate these challenges, I can share my personal experience working with the Oxford Inertial Navigation dataset[11], a well-known real-world INS dataset. While the Oxford dataset provides valuable data from a real vehicle, I encountered several difficulties during my research. The dataset lacked detailed information about the sensor characteristics, such as noise levels and biases, making it challenging to develop accurate error models. Additionally, the ground truth measurements were not always reliable, particularly in areas with poor GPS coverage, which made it difficult to evaluate the performance of the SciML techniques accurately. These challenges highlight the limitations of relying solely on real-world datasets for the development and testing of SciML algorithms.

## 4.2 Simulation Environment

The simulation environment is built using Julia, a high-performance programming language well-suited for numerical analysis and computational science. Julia's efficiency in handling complex mathematical operations and its rich ecosystem of packages make it an ideal choice for simulating sophisticated INS scenarios and implementing SciML algorithms.

At the core of our simulation environment is the HIDR (High-Performance Inertial Navigation Development Repository) library, a proprietary collection of packages and tools specifically designed for simulating INS. HIDR integrates several essential components, including:

- IMU simulation package: Generates realistic sensor data, incorporating noise, biases, and other error characteristics.
- Extended Kalman Filter (EKF) solver: Implements the EKF algorithm for state esti-

mation and error correction.

- Plotting and comparison tools: Facilitates the visualization and analysis of simulation results.

These packages, seamlessly integrated within the Julia ecosystem, streamline the development process and enable efficient experimentation and iteration. The modular structure of HIDR allows for easy customization and extension, accommodating a wide range of INS configurations and scenarios.

By leveraging the capabilities of Julia and HIDR, our simulation environment provides a robust platform for developing and evaluating SciML techniques for INS drift reduction. The realistic modeling of sensor errors, vehicle dynamics, and environmental factors ensures that the simulated data closely resembles real-world conditions, enhancing the transferability of the developed algorithms to practical applications.

In the following sections, we will delve into the specifics of simulated data generation and discuss how this simulation environment contributes to the advancement of SciML methodologies for INS drift reduction.

## 4.3 Simulated Data Generation

The process of simulated data generation involves several key steps, each designed to ensure that the data reflects realistic navigation conditions as closely as possible. Starting from the generation of a trajectory path, moving through the calculation of corresponding velocities and gravitational effects, and culminating in the simulation of sensor outputs, each step is crucial. This section will delve into the specifics of these steps, illustrated by the flowchart below that outlines the sequence from trajectory generation to the final output of simulated INS data.

Leveraging the operational principles of INS outlined in Chapter 2, including the integration of acceleration and angular rates, our simulation environment mimics these processes

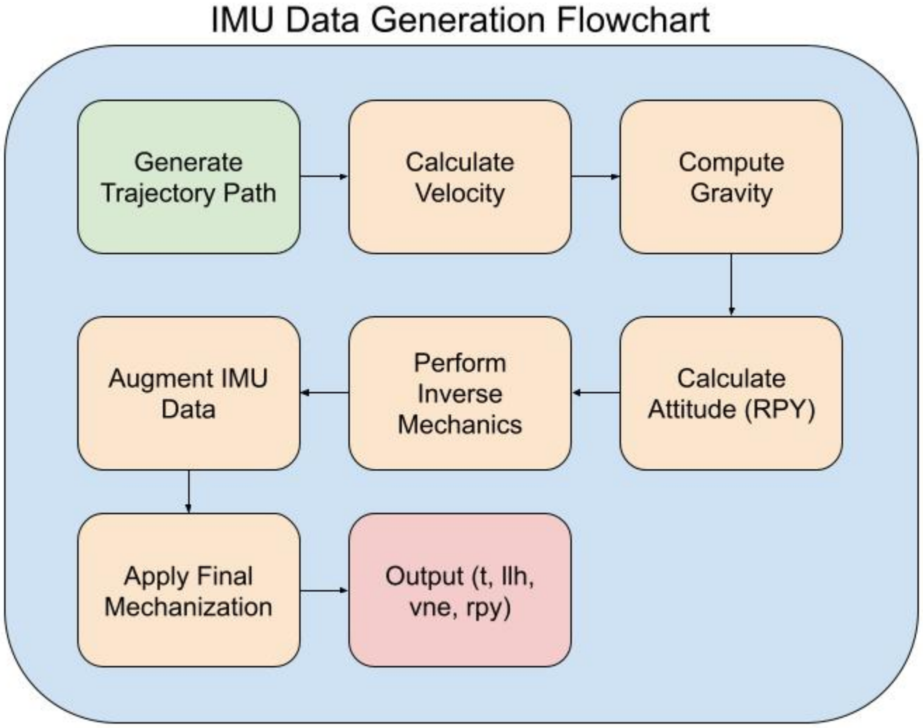


Figure 4.1: Overview of the IMU Data Generation Process

to generate realistic navigational paths. The data generation begins with the creation of a trajectory path, defining latitude, longitude, and altitude to establish a baseline for navigational computations. From this we can generate various paths including simple linear trajectories or more complex figures like a figure eight, simulating different driving or flying conditions.

Following trajectory generation, velocities in the north, east, and down(vne) directions are computed. This calculation translates the geographic changes into directional velocities, which are crucial for the next step of the process, namely the calculation of gravity. Gravity values are calculated at different trajectory points to accurately simulate the gravitational influence that affects the accelerometer measurements.

The roll, pitch, and yaw (RPY) are then determined from the velocity and gravity data. These orientation angles are essential for aligning the vehicle's movement within its environment, providing fundamental data for the subsequent inverse mechanics process.

Inverse mechanics play a critical role in simulating an ideal INS's response[12]. This step involves:

- Simulating perfect sensor outputs by theoretically calculating the specific forces and angular rates that inertial sensors would record in an error-free scenario.
- Back-calculating raw IMU outputs to determine what the accelerometers and gyroscopes should ideally measure based on the vehicle's movements.

Once the "perfect" IMU data is generated, it is augmented with realistic errors, such as biases, noise, and random drifts. These imperfections simulate the actual conditions under which INS algorithms must operate, adding complexity to the data set.

The final mechanization applies these augmented data through algorithms that mimic actual INS operations, factoring in all introduced errors to compute the navigation states. This step tests the robustness of the SciML techniques in correcting inaccuracies and improving overall navigation accuracy.

The output from this process includes time-stamped data sets of positions (llh), velocities (vne), and orientations (rpy), serving as the simulated ground truth for further validation of INS algorithms.

Through this detailed and systematic approach, the simulation environment supports comprehensive testing and validation of INS algorithms, ensuring that the SciML techniques are robust and effective in real-world navigation scenarios.

## 4.4 Conclusion

In this chapter, we have discussed the simulation-focused approach adopted in our research for developing SciML techniques for INS drift reduction. Simulation offers numerous benefits, such as the ability to control error sources, generate large and diverse datasets, and investigate challenging scenarios that may be difficult to replicate in real-world experiments.

We have highlighted the advantages of simulation, which include precise manipulation of sensor biases and noise characteristics, as well as the generation of extensive and varied datasets. These capabilities are essential for improving the robustness and generalization of SciML models, allowing researchers to develop techniques that can better handle real-world variations and uncertainties.

Additionally, we have addressed the challenges associated with real-world datasets, such as limited availability, sensor misalignments, and time synchronization issues. While real-world data is crucial for validating the developed techniques, the simulation-focused approach serves as a vital intermediary step, enabling us to mitigate these challenges and develop robust algorithms.

The simulation environment, built using Julia and leveraging the proprietary HIDR library, provides a powerful platform for simulating sophisticated INS scenarios and implementing SciML algorithms. HIDR integrates essential components, including an IMU simulation package, an Extended Kalman Filter (EKF) solver, and plotting and comparison tools. The seamless integration of these packages within the Julia ecosystem streamlines the development process and enables efficient experimentation and iteration.

Moreover, the simulation environment emphasizes realistic modeling of sensor errors, vehicle dynamics, and environmental factors, ensuring that the simulated data closely resembles real-world conditions. This enhances the transferability of the developed algorithms to practical applications.

The process of simulated data generation, as outlined in Section 4.3, involves a systematic approach that mimics the operational principles of INS. By generating realistic trajectories, computing velocities and gravitational effects, and simulating sensor outputs with realistic errors, the simulation environment supports comprehensive testing and validation of INS algorithms.

In the following chapter, we will delve into the specifics of the SciML methodology for INS drift reduction, building upon the foundation established by our simulation-focused approach. By leveraging the capabilities of our simulation environment, we aim to develop robust and effective techniques that can significantly improve the accuracy and reliability of inertial navigation systems.

# Chapter 5

# SciML Methodology for INS Drift Reduction

## 5.1 Introduction

Building upon the foundational knowledge of Data Generation, Inertial Navigation Systems (INS), and Scientific Machine Learning (SciML) discussed in the previous chapters, this chapter focuses on the integration of these components to address the critical challenge of drift reduction in INS.

The proposed SciML architecture integrates neural network models with traditional INS methods, aiming to harness the power of machine learning to capture complex error patterns and relationships from data. By combining the strengths of both domains, this approach seeks to overcome the limitations of conventional error correction methods and provide a more robust and accurate navigation solution.

## 5.2 Proposed SciML Architecture

Figure 5.1 presents a high-level conceptual diagram of the proposed SciML architecture for INS drift reduction. The architecture integrates neural network models with traditional INS

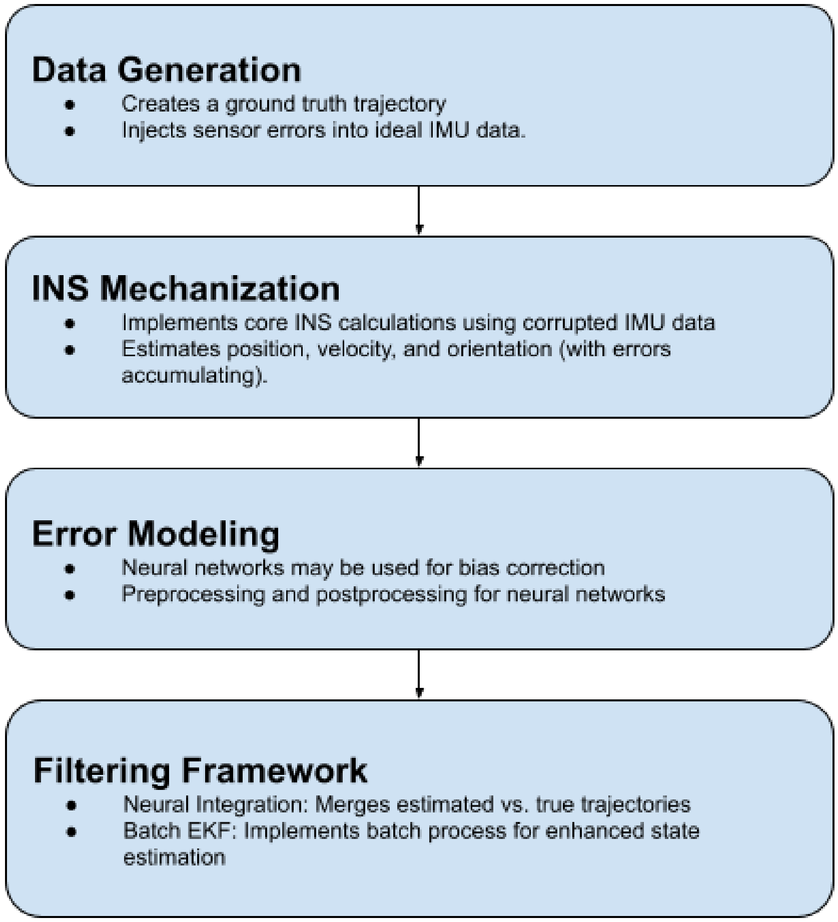


Figure 5.1: High-level conceptual diagram of the proposed SciML architecture for INS drift reduction

components to enhance the error correction capabilities of the system. The key components of the architecture and their roles are as follows:

- **Data Generation**: This component focuses on creating a ground truth trajectory and injecting sensor errors into the ideal IMU data. It provides the necessary datasets for training and validating the SciML models. The data generation process leverages a simulation environment to create realistic scenarios and control the characteristics of the injected errors. The details of this process are discussed in Section 4.3.
- **INS Mechanization**: The INS mechanization component implements the core INS calculations using the corrupted IMU data. It estimates the position, velocity, and

orientation of the system, considering the errors accumulating over time. The mechanization code forms the backbone of the INS and provides the foundation for integrating the SciML models. While the foundations of INS mechanization were laid out in Chapter 2, a high-level overview will be provided in this chapter to illustrate its role in the proposed architecture.

- **Error Modeling**: This component leverages neural networks to model and correct sensor biases effectively. Separate models are tailored for accelerometer and gyroscope data, addressing the distinct error characteristics of each sensor type. Through training, these neural networks learn to associate the corrupted sensor data with accurate bias corrections, thereby enhancing error compensation capabilities.

- **Filtering Framework**: This framework integrates neural network-derived models with the dynamics of the Inertial Navigation System (INS) to perform sophisticated state estimation. It employs a batch filtering approach that incorporates bias corrections provided by the neural networks into the fundamental INS dynamics model. An Extended Kalman Filter (EKF) is utilized to refine state estimation, processing batches of data to enhance the accuracy of the estimated trajectories significantly.

The rationale behind incorporating neural networks into the INS framework lies in their ability to learn complex error patterns and relationships from data. By leveraging the power of machine learning, the proposed architecture aims to enhance the error correction capabilities of the INS and mitigate the impact of sensor biases on the overall navigation accuracy.

By integrating the SciML components with the traditional INS mechanization and filtering techniques, the proposed architecture aims to achieve superior navigation accuracy and robustness in the presence of sensor errors and drift. The subsequent sections will delve into the details of each component, explaining their implementation and the underlying principles that govern their operation.

## 5.3 Data Generation

Data generation is a critical component in the development and evaluation of the proposed SciML methodology for INS drift reduction. As detailed in Chapter 4, the simulation environment, built using Julia and the High-Performance Inertial Navigation Development Repository (HIDR), enables the creation of realistic and diverse datasets that encompass a wide range of trajectories, sensor errors, and operational conditions.

The data generation process involves several key steps, including the generation of trajectory paths, calculation of corresponding velocities and gravitational effects, and simulation of sensor outputs. Importantly, the systematic augmentation of sensor data with various types of errors, such as biases, noise, and random walks, is performed to represent a wide range of inaccuracies encountered in real-world scenarios. This ensures that the SciML models are trained on datasets that closely resemble real-world conditions, enhancing their ability to learn robust error correction strategies.

For a comprehensive understanding of the data generation process, its components, and the underlying principles, readers are encouraged to refer to Chapter 4 and more specifically look at Fig. 4.1 to see how the system was built. The detailed exploration of the simulation environment, tools, software setup, and modeling techniques provided in that chapter forms the foundation for the successful application of SciML techniques in enhancing INS performance.

## 5.4 INS Mechanization

INS mechanization plays a crucial role in the proposed SciML architecture for INS drift reduction. It involves the implementation of the core INS calculations using the corrupted IMU data to estimate the position, velocity, and orientation of the system. The mechanization process forms the backbone of the INS and provides the foundation for integrating the

SciML models. This section builds upon the foundations of INS mechanization laid out in Chapter 2.

The INS mechanization process is implemented using the HIDR library, which provides a comprehensive set of tools for INS-related computations. The mechanization code takes the following inputs:

- Corrupted IMU data: Specific force measurements from accelerometers and rotation rate measurements from gyroscopes.
- Initial conditions: Initial position (latitude, longitude, and altitude), velocity (in the North-East-Down frame), and attitude (roll, pitch, and yaw angles).
- Sampling period: The time step between consecutive IMU measurements.

The mechanization process applies the INS mechanization equations to propagate the navigation states forward in time. It performs several key steps, including:

- Coordinate transformations: Converting the IMU measurements from the body frame to the navigation frame using rotation matrices such as the direction cosine matrix (DCM).
- Integration: Integrating the accelerometer and gyroscope data to estimate the vehicle trajectory, computing the updated position, velocity, and orientation at each time step.
- Accounting for Earth's rotation and gravity: Incorporating the effects of the Earth's rotation and the local gravity vector for accurate INS mechanization.

The output of the INS mechanization, including the estimated position, velocity, and orientation, serve as critical inputs for the SciML components of the proposed architecture. These mechanization outputs are compared with ground truth data to compute errors in navigation states. The errors calculated are then used to train neural network models (modelA for accelerometer data and modelG for gyroscope data) to learn the error characteristics and generate appropriate bias corrections. The corrected navigation states obtained from the neural networks are then fed back into the mechanization process to update the INS solution, allowing for dynamic correction of errors and minimization of drift over time.

INS mechanization poses several challenges, particularly when dealing with non-linear dynamics, error accumulation, and proper initialization and alignment of the system. These challenges were briefly introduced in Chapter 2. To address these challenges, the mechanization code employs advanced numerical methods, such as high-order integration schemes and adaptive time-stepping, to minimize numerical errors and maintain the stability of the solution. The integration of SciML techniques, such as neural network-based error correction, helps dynamically compensate for the accumulated errors and reduce overall drift in the navigation solution.

The mathematical foundations of the INS mechanization process, including the key equations for position, velocity, and attitude updates, were covered in detail in Chapter 2. Readers are encouraged to refer to that chapter for a comprehensive understanding of the underlying mathematics and concepts.

In summary, INS mechanization is a vital component of the proposed SciML architecture for INS drift reduction. The mechanization process accurately propagates the navigation states forward in time, taking into account the corrupted IMU data and the necessary coordinate transformations. The integration of SciML techniques, such as neural network-based error correction, enhances the accuracy and robustness of the INS, enabling superior navigation performance in the presence of sensor errors and drift.

## 5.5 Error Modeling and Correction

Building upon the foundations of INS Mechanization within the SciML Architecture, we now explore the system's approach to learning and correcting the noise and bias present in the sensor data. The Error Modeling and Correction component plays a crucial role in addressing sensor biases, which are a significant contributor to navigational drift. This component employs neural network models, namely ModelA for accelerometer data and ModelG for gyroscope data, to tackle this critical issue.

Figure 5.2 provides an overview of the error modeling and correction pipeline within the SciML architecture.

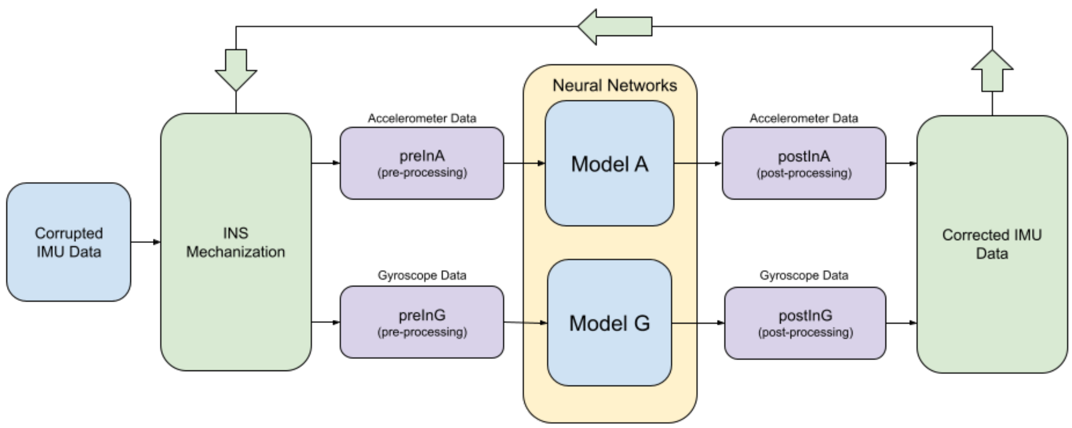


Figure 5.2: Error Modeling and Correction Pipeline

As illustrated in Figure 5.2, the pipeline begins with the input of corrupted IMU data, which is fed into the INS Mechanization block. The INS Mechanization process separates the data into two parallel paths: one for accelerometer data and another for gyroscope data.

Each path includes preprocessing (preInA and preInG), neural network models (ModelA and ModelG), and postprocessing (postInA and postInG) steps. The preprocessing functions normalize the sensor data, including biases and noise, to ensure consistency across training samples. ModelA and ModelG are sophisticated neural network architectures that incorporate multiple layers with tanh activation functions to effectively capture the non-linear error characteristics inherent in sensor data. These networks are trained to minimize a loss function that quantifies the discrepancy between the predicted corrections and the actual error values, employing optimization techniques such as stochastic gradient descent.

Once trained, ModelA and ModelG provide real-time bias corrections, which are applied to the preprocessed sensor data. The corrected data from both paths is then combined and passed through the postprocessing functions, which convert the neural network outputs back to their original scale. The resulting corrected IMU data is fed back into the INS Mechanization block, dynamically enhancing the navigation estimates.

In addition to error correction, Jacobian matrices are computed for each neural network model. These matrices enable the linear approximation of the impact of neural networks and facilitate the integration of machine learning-based corrections into the Extended Kalman Filter (EKF), which will be discussed in the next section.

Figure 5.3 illustrates the error evaluation process, which is an essential component of the Error Modeling and Correction module.

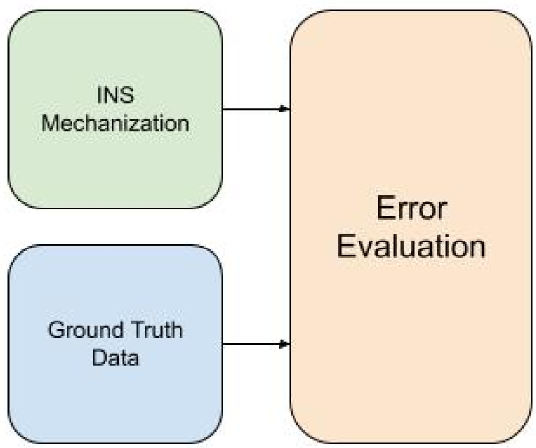


Figure 5.3: Error Evaluation Process

As shown in Figure 5.3, the effectiveness of the error correction approach is continuously evaluated by comparing the INS Mechanization output with ground truth data. This comparison allows for ongoing refinement of the neural network models and the processing steps involved, ensuring the system remains accurate and reliable.

The comprehensive integration of SciML components with traditional INS mechanisms in the Error Modeling and Correction module significantly advances INS error correction capabilities, enhancing the robustness and accuracy of the system. The diagrams provided offer a clear visual representation of the key steps involved in the error modeling and correction process, highlighting the interplay between the neural network models, preprocessing and postprocessing functions, and the integration with the INS Mechanization process.

With the Error Modeling and Correction component in place, the SciML architecture is well-equipped to mitigate the impact of sensor biases and improve overall navigation performance. However, to fully realize the potential of the SciML approach, the corrected sensor data must be effectively integrated into the state estimation process. This is where the Filtering Framework and State Estimation component comes into play, leveraging the corrected sensor data and Jacobian matrices to achieve high-accuracy and reliable navigation solutions.

In the next section, we will explore the Filtering Framework and State Estimation component in detail, examining how it integrates traditional filtering methods with SciML enhancements to optimize the state estimation process and revolutionize the field of inertial navigation.

## 5.6 Filtering Framework and State Estimation

The Filtering Framework and State Estimation component is a crucial element of the SciML architecture, responsible for achieving high-accuracy and reliable navigation solutions by integrating traditional filtering methods with SciML enhancements. This component leverages the strengths of both approaches to optimize the state estimation process and overcome the limitations of conventional INS techniques.

At the core of the filtering framework lies the Extended Kalman Filter (EKF), a well-established technique for estimating the states of non-linear systems. The EKF relies on linear approximations of the system dynamics and measurement models to effectively handle the complexities of the integrated system and adapt to the non-linear characteristics of sensor errors.

Figure 5.4 presents a simplified architecture diagram of the Filtering Framework and

State Estimation component, highlighting the integration of the EKF and batch filtering approaches.

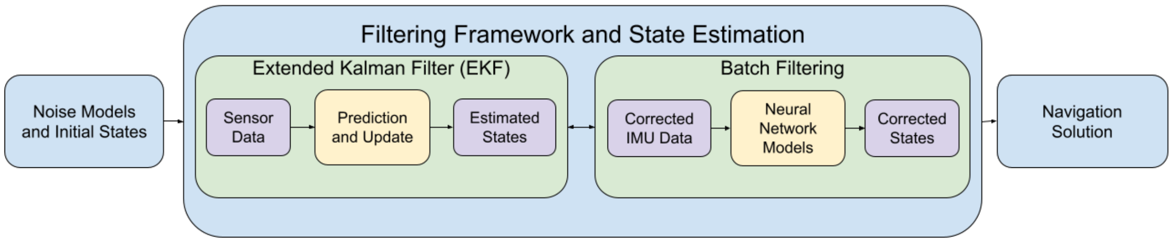


Figure 5.4: Simplified architecture of the Filtering Framework and State Estimation component in the SciML methodology.

As depicted in Figure 5.4, the filtering process receives noise models and initial states as inputs, which are meticulously tuned to ensure optimal filter performance and accurate state estimation. The careful selection and adjustment of these parameters are critical to the overall effectiveness of the filtering framework.

The EKF operates in an iterative manner, alternating between prediction and update steps. During the prediction step, the filter employs the system dynamics model to estimate the current state based on the previous state and control inputs. The update step incorporates measurements from external sensors, such as GPS, to correct the predicted state estimates. By fusing the information from external sensors with the predicted states, the filter significantly improves the accuracy and reliability of the navigation solution.

One of the key innovations of the SciML architecture is the integration of batch filtering, which harnesses the capabilities of neural network models to provide corrections to the state estimates. The batch filtering process involves processing data in batches, enabling the exploitation of temporal correlations and patterns in the sensor data for more effective error correction and state estimation.

The neural network models play a vital role in the batch filtering approach. By learning complex error patterns and adapting to evolving environmental conditions, these models enhance the accuracy and robustness of the state estimation process. The corrected IMU data obtained from the neural networks is fed into the batch filtering component, which further refines the state estimates.

The integration of the EKF and batch filtering approaches is represented by the bidirectional arrow in Figure 5.4. This integration allows for the seamless exchange of information between the two components, leveraging the strengths of both traditional filtering techniques and SciML enhancements.

The performance of the Filtering Framework and State Estimation component is rigorously evaluated using a range of metrics, and the SciML architecture incorporates mechanisms for continuous monitoring and adaptation of the filter parameters. This adaptability is essential for maintaining high accuracy and reliability in real-world applications, where environmental conditions and sensor characteristics may vary over time.

In conclusion, the Filtering Framework and State Estimation component of the SciML architecture represents a significant advancement over traditional INS methods. By integrating the EKF with batch filtering and leveraging the power of neural network models, the system gains the ability to learn and adapt to complex error patterns, resulting in more accurate and reliable navigation solutions. The simplified architecture diagram (Figure 5.4) provides a clear visual representation of the key components and their interactions, highlighting the potential of SciML in revolutionizing the field of inertial navigation. Through the careful design and integration of the Filtering Framework and State Estimation component, the SciML architecture paves the way for a new era of high-performance, adaptive, and reliable navigation systems.

# Chapter 6

# Experimental Results and Analysis

## 6.1 Overview of Experiment

This chapter presents the results of experiments designed to test the effectiveness of the Scientific Machine Learning (SciML) methodology for Inertial Navigation System (INS) drift reduction, as introduced in Chapter 5. The experiments aimed to address the research questions outlined in Chapter 1 and evaluate the performance of the proposed SciML architecture in mitigating INS drift under realistic conditions.

The SciML methodology, as discussed in Chapter 5, integrates neural network models with traditional INS components to enhance the error correction capabilities of the system. The architecture incorporates key components such as data generation, INS mechanization, error modeling, and a filtering framework. By leveraging the power of machine learning, the SciML approach seeks to capture complex error patterns and relationships from data, enabling more accurate and robust navigation solutions.

In the experiments, a simulated figure-8 trajectory, as shown in Figure 6.1, was used to generate realistic motion data. The data generation process, detailed in Section 4.3, involved the creation of ground truth trajectories and the injection of sensor errors to simulate real-world scenarios. This approach allowed for a comprehensive evaluation of the SciML methodology's performance under diverse and challenging conditions.

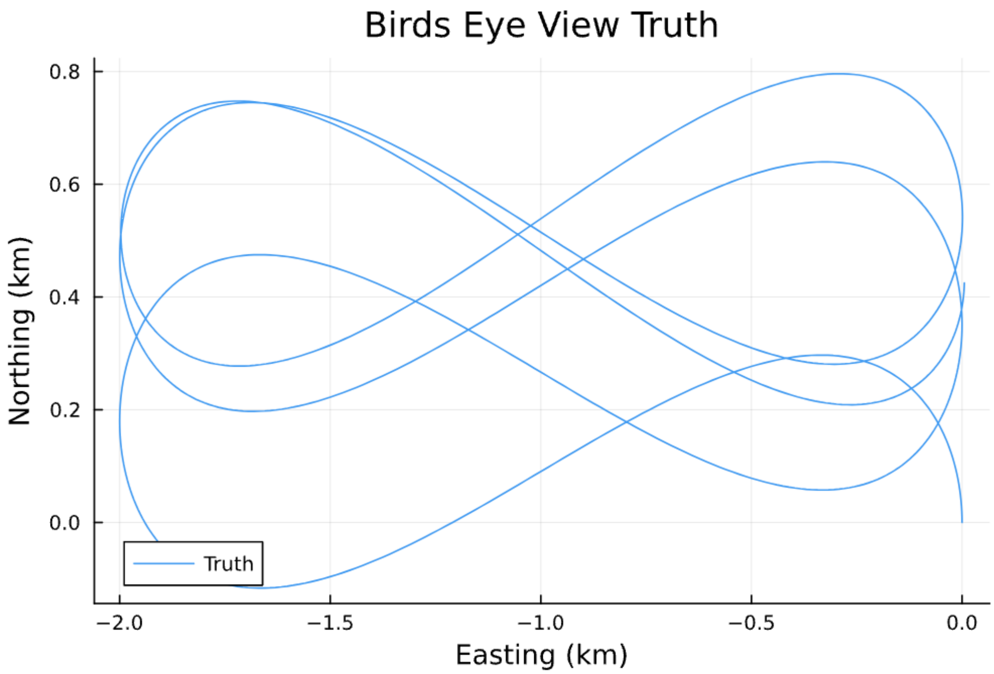


Figure 6.1: Birds Eye View of Truth Data

The outcomes of these experiments are expected to shed light on the potential of SciML approaches in revolutionizing the field of inertial navigation. By demonstrating the superior performance of the SciML-based INS compared to traditional methods, this study contributes to the growing body of knowledge supporting the integration of machine learning techniques with classical navigation algorithms. The findings presented in this chapter serve as a foundation for further research and development efforts aimed at advancing the state-of-the-art in INS drift reduction and enabling more reliable and accurate navigation in a wide range of applications.

### 6.1.1 SciML vs. Traditional Baselines

To demonstrate the advantages of the SciML approach, the performance of the SciML-based INS was compared against traditional INS solutions, such as those employing Kalman filtering. The comparison was conducted using a simulated figure-8 trajectory, which provides a realistic and challenging scenario for evaluating the drift reduction capabilities of the different approaches.

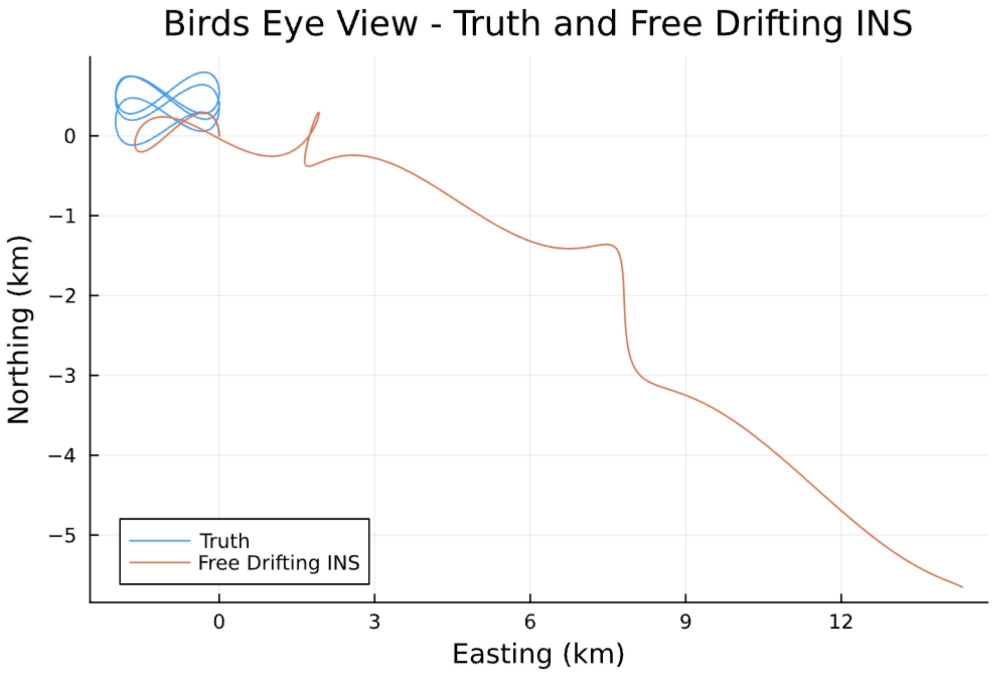


Figure 6.2: Comparison between Truth and Free Drifting INS

Figure 6.2 illustrates the problem of drift in traditional INS systems. The blue line represents the ground truth trajectory, while the orange line shows the trajectory estimated by a free-drifting INS without any error correction. The significant deviation between the two trajectories highlights the need for effective drift reduction techniques to maintain navigation accuracy over time.

To address the drift problem, traditional approaches such as Kalman filtering have been employed. Figure 6.3 compares the performance of a baseline Kalman filter (orange line) against the ground truth (blue line). While the baseline filter provides some improvement over the free-drifting INS, there is still a noticeable discrepancy between the estimated and true trajectories, indicating the limitations of traditional filtering methods in handling complex error patterns and non-linear motion dynamics.

The introduction of the SciML approach marks a significant advancement in INS drift reduction. Figure 6.4 showcases the superior performance of the SciML filter (dotted green

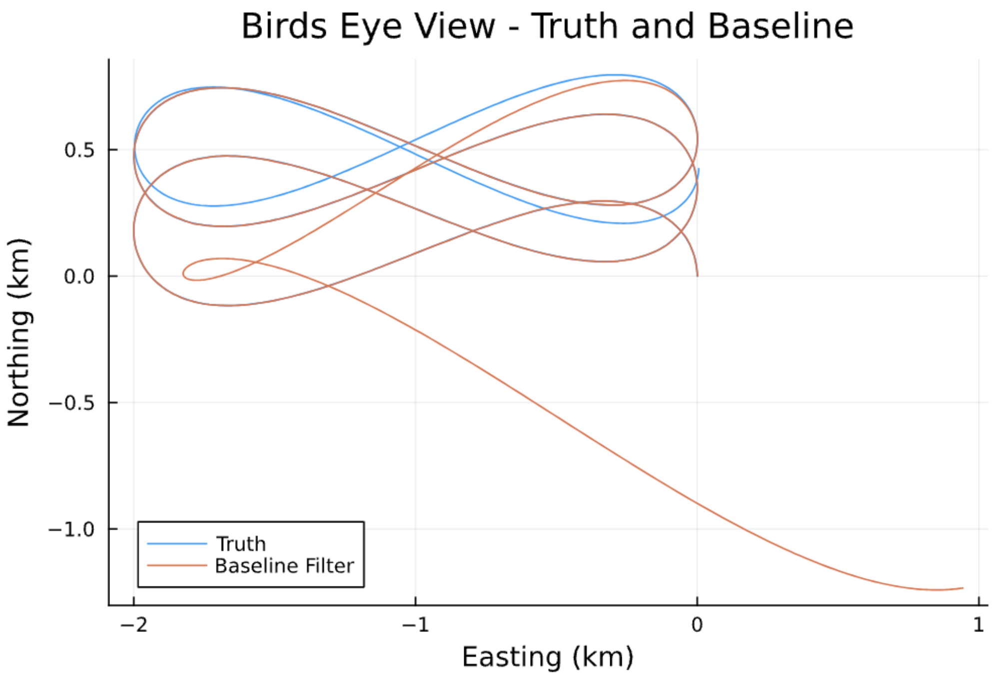


Figure 6.3: Comparison between Truth and Baseline Filter

line) compared to both the baseline filter (orange line) and the ground truth (blue line). The SciML filter demonstrates a remarkable ability to closely track the true trajectory, effectively compensating for the complex error patterns and non-linearities present in the INS measurements.

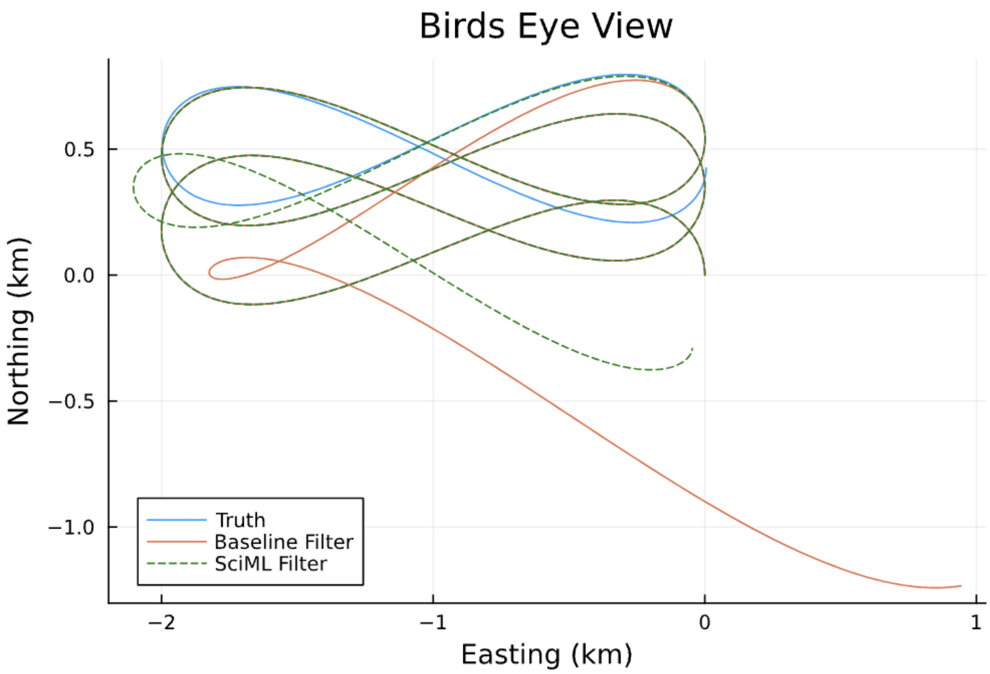


Figure 6.4: Comparison between Truth, Baseline Filter, and SciML Filter

The enhanced performance of the SciML filter can be attributed to its ability to learn and adapt to the specific error characteristics of the INS through the integration of neural network models. By leveraging the power of machine learning, the SciML approach can capture intricate error patterns and dynamically adjust the error correction based on the learned models. This adaptability enables the SciML filter to provide more accurate and robust drift reduction compared to traditional filtering techniques.

The visual comparison of the trajectories in Figure 6.4 clearly demonstrates the superiority of the SciML approach in minimizing drift and maintaining a high level of navigation accuracy. The close alignment between the SciML filter output and the ground truth trajectory validates the effectiveness of integrating machine learning techniques with traditional INS methods.

These diagrams combine the most important information from our Simulation Results. A breakdown of all the individual comparisons between Northing, Easting, and other important IMU measurments can be found in the Appendix B.

In the next subsection, we will breakdown the numerical performance of the SciML-based

INS and see what kind of improvement we ended up achieving.

### 6.1.2 Numerical Summary of Results

Table 6.1 summarizes the key numerical metrics from the experiments, highlighting the percentage improvements achieved by the SciML-based system over the baseline.

| **Metric** | **Baseline** | **SciML-based INS** | **Improvement (%)** |
|---|---|---|---|
| 3D RMSE | 516.55 | 191.21 | 62.99 |
| Max 3D Error | 1914.55 | 718.74 | 62.47 |
| Std 3D Error | 466.22 | 171.50 | 63.21 |
| Median 3D Error | 1.02 | 0.46 | 54.98 |
| MAE Roll Error (Deg) | 0.081 | 0.034 | 57.98 |
| MAE Pitch Error (Deg) | 0.127 | 0.040 | 68.59 |
| MAE Yaw Error (Deg) | 1.923 | 1.447 | 24.75 |

Table 6.1: Comparative performance metrics for baseline and SciML-based INS systems.

The significant reductions in error metrics and the percentage improvements highlight the potential of integrating Scientific Machine Learning techniques with traditional INS solutions. The SciML-based INS achieved a remarkable 62.99% reduction in 3D RMSE (Root Mean Square Error) compared to the baseline system. This substantial improvement indicates that the SciML approach effectively minimizes the overall position error, providing a more accurate navigation solution.

Furthermore, the SciML-based INS demonstrated a 62.47% reduction in the maximum 3D error and a 63.21% reduction in the standard deviation of 3D error. These metrics underscore the consistency and reliability of the SciML approach in mitigating large errors and maintaining a stable navigation performance throughout the trajectory.

The median 3D error, which represents the typical error magnitude, was reduced by 54.98% in the SciML-based system. This significant reduction suggests that the SciML approach effectively handles the majority of error scenarios, providing a robust navigation solution even in the presence of outliers or extreme error conditions.

In terms of attitude estimation, the SciML-based INS achieved impressive improvements in roll, pitch, and yaw errors. The Mean Absolute Error (MAE) for roll and pitch was reduced by 57.98% and 68.59%, respectively, indicating a substantial enhancement in the accuracy of attitude determination. The MAE for yaw error showed a more modest improvement of 24.75%, highlighting the challenges associated with estimating heading in inertial navigation systems.

These numerical results demonstrate the superior performance of the SciML-based INS across various error metrics, showcasing its ability to effectively reduce drift, improve position accuracy, and enhance attitude estimation. The consistent and significant improvements over the baseline system underscore the potential of SciML techniques in addressing the limitations of traditional INS solutions.

Moreover, the percentage improvements achieved by the SciML-based INS highlight its adaptability to complex error patterns and dynamic environmental conditions. The substantial reductions in error metrics suggest that the SciML models can effectively learn and compensate for the intricate error characteristics of the INS, leading to a more robust and reliable navigation solution.

The numerical results presented in Table 6.1 provide a comprehensive assessment of the SciML-based INS performance and serve as a strong foundation for further analysis and discussion. These results not only validate the effectiveness of the SciML approach but also motivate continued research and development efforts to further enhance the accuracy, robustness, and scalability of SciML techniques in inertial navigation applications.

## 6.2 Analysis and Discussion

The experimental results robustly prove the numerous advantages of the SciML-based INS over the traditional baseline system in terms of drift reduction and superior navigation performance. SciML consistently outperforms the baseline, and its overall error reduction varied from 24.75% to 68.59% in comparison with various metrics. The robustness of the theoretical assumption represented in the initial hypothesis can be clearly observed in these outcomes, which also demonstrate the vast opportunities of SciML use in practice. The significant advantage of learning and fitting scientific machine learning models to complex patterns of error compensates for the substantial shortcomings of the traditional INS approach, which largely guarantees its high effectiveness in the environment of challenging conditions.

The SciML models demonstrated exceptional performance in scenarios characterized by non-linear and unpredictable motion patterns, aligning with the inherent strengths of machine learning in capturing intricate relationships and adapting to dynamic environments. The models' behavior closely matched theoretical predictions, validating the underlying assumptions and design choices of the SciML architecture. This alignment reinforces the soundness of the SciML methodology and its potential for further enhancements. However, it is important to acknowledge that some deviations from the expected behavior were observed during the experiments. These deviations provide valuable insights into the models' capabilities and limitations, highlighting the need for further research and refinement to address specific challenges and improve robustness in diverse operating conditions.

The experimental results also shed light on the scalability and generalization capabilities of the SciML models. The models exhibited consistent performance improvements across different trajectory types and error magnitudes, indicating their ability to adapt to various scenarios. This adaptability is crucial for real-world applications, where navigation systems must handle a wide range of operating conditions. However, further investigation is necessary to assess the performance of the SciML approach in larger-scale and more complex navigation tasks, such as extended missions or highly dynamic environments. These investigations will provide a more comprehensive understanding of the models' capabilities and guide future development efforts.

The insights gained from the analysis of the SciML models' behavior and performance trends have far-reaching implications for future research and development in the field of inertial navigation. The identified strengths and limitations of the SciML approach can inform the refinement of the models, the optimization of training procedures, and the exploration of novel techniques to further enhance drift correction capabilities. Moreover, the successful application of SciML methods opens up new avenues for innovation and advancement in the development of next-generation navigation technologies. By leveraging the power of machine learning, researchers and engineers can push the boundaries of what is possible in terms of navigation accuracy, reliability, and adaptability.

In conclusion, the analysis and discussion of the experimental results provide strong evidence for the effectiveness and potential of the SciML approach in reducing INS drift. The significant performance improvements, the ability to adapt to complex error patterns, and the alignment with theoretical predictions all contribute to the growing body of knowledge supporting the integration of SciML techniques with traditional INS solutions. The insights gained from this study lay the foundation for further research and development efforts aimed at revolutionizing the field of inertial navigation and unlocking new possibilities for accurate and reliable navigation in a wide range of applications.

# Chapter 7

# Conclusion

## 7.1 Summary of Research and Key Findings

This thesis addressed the challenges faced by traditional Inertial Navigation Systems (INS), specifically the accumulation of errors over time and the reliance on external aids for correction. The introduction chapter highlighted these issues and their impact on the accuracy and reliability of INS in various applications. To overcome these challenges, this research proposed the integration of Scientific Machine Learning (SciML) techniques with traditional INS methods.

The SciML approach presented in this thesis offers a new way of modeling and correcting INS errors. By using machine learning, particularly neural network models, the proposed architecture allows INS to learn and adapt to complex error patterns in real-time. This dynamic error compensation capability distinguishes SciML from conventional INS error mitigation techniques, which often rely on simplified models and static parameters. The effectiveness of the SciML approach was thoroughly tested through simulations, as detailed in the experimental results chapter. The simulation outcomes demonstrate significant improvements in drift reduction and overall navigation accuracy achieved by the SciML-based INS. The comparative analysis against traditional baseline systems revealed substantial reductions in position, velocity, and attitude errors, with the SciML approach consistently outperforming its counterparts.

The findings of this research contribute to the broader field of navigation and positioning systems. By showing the feasibility and benefits of integrating SciML with INS, this research opens up new possibilities for developing more reliable, adaptable, and resilient navigation technologies. The successful application of SciML in this context suggests its potential extension to other domains, such as autonomous vehicles, robotics, and aerospace systems, where accurate and robust navigation is essential. Furthermore, the research presented in this thesis aligns with the ongoing efforts in the navigation community to make INS more self-sufficient and capable of operating in challenging environments. The SciML approach enables INS to handle complex error dynamics and adapt to changing conditions without heavily relying on external aids, which is particularly valuable in scenarios where external references may be unavailable, unreliable, or vulnerable to interference.

## 7.2 Final Remarks

While this thesis has made progress in addressing the challenges of INS drift and error accumulation, it also opens up new research questions and opportunities for further exploration. Future research could focus on optimizing the SciML architecture, investigating advanced neural network structures, training strategies, and data preprocessing techniques to further improve performance. Additionally, the scalability and generalization capabilities of the SciML approach could be tested in more diverse and complex navigation scenarios, such as extended missions, multi-sensor fusion, and highly dynamic environments.

Moreover, the integration of SciML with other emerging technologies, such as deep learning, reinforcement learning, and transfer learning, presents exciting prospects for the future of INS. These combinations could lead to the development of even more advanced and intelligent navigation systems that can learn from past experiences, adapt to new situations, and make optimal decisions in real-time.

In conclusion, this thesis demonstrates the effectiveness of SciML as an approach for improving INS performance and paves the way for a new generation of navigation systems that are more intelligent, adaptable, and resilient. The research presented herein lays the foundation for future advancements in the field, inspiring researchers and practitioners to explore the potential of integrating machine learning with traditional navigation techniques. As we continue to push the boundaries of what is possible in navigation and positioning systems, the insights and contributions of this thesis will serve as a valuable resource and catalyst for further innovation.

I would like to acknowledge the use of AI language models, specifically for assisting in the proofreading and grammatical improvement of the final draft of this thesis, in accordance with the EECS Department guidelines effective April 20, 2023.

# Appendix A

# List of Figures for Understanding Inertial Navigation

This appendix includes a comprehensive list of figures in order to understand Inertial Navigation and its Operational Principles.

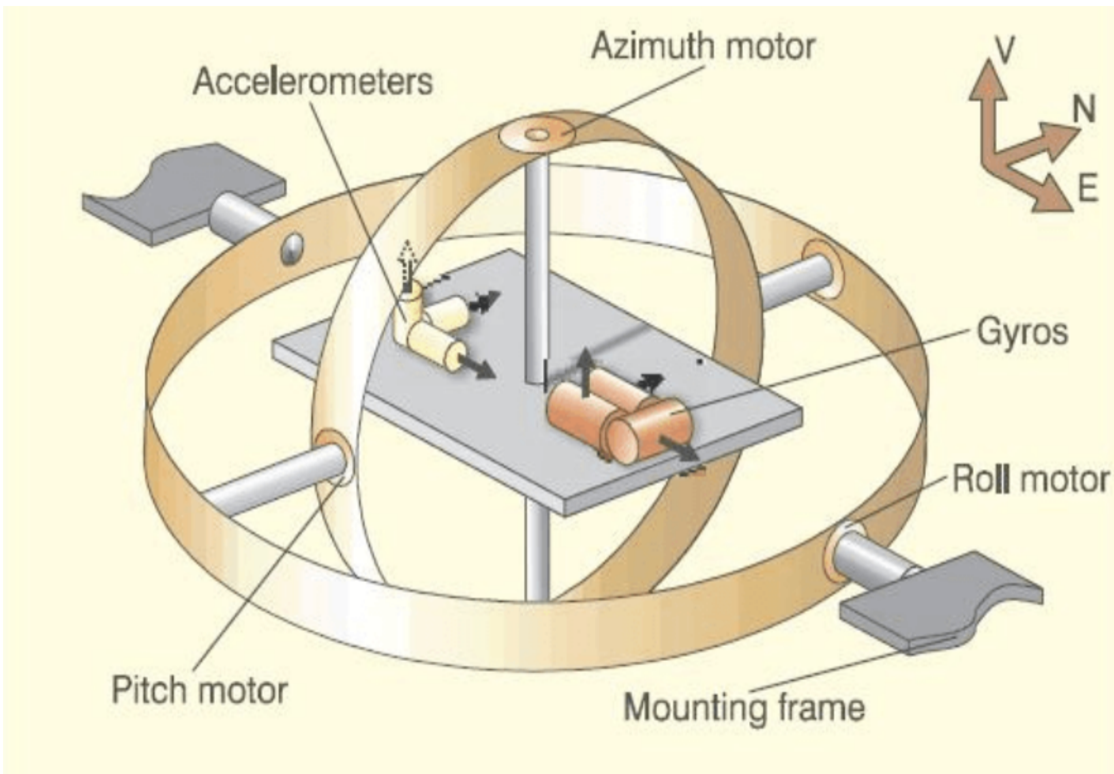


Figure A.1: Inertial Measurement Unit

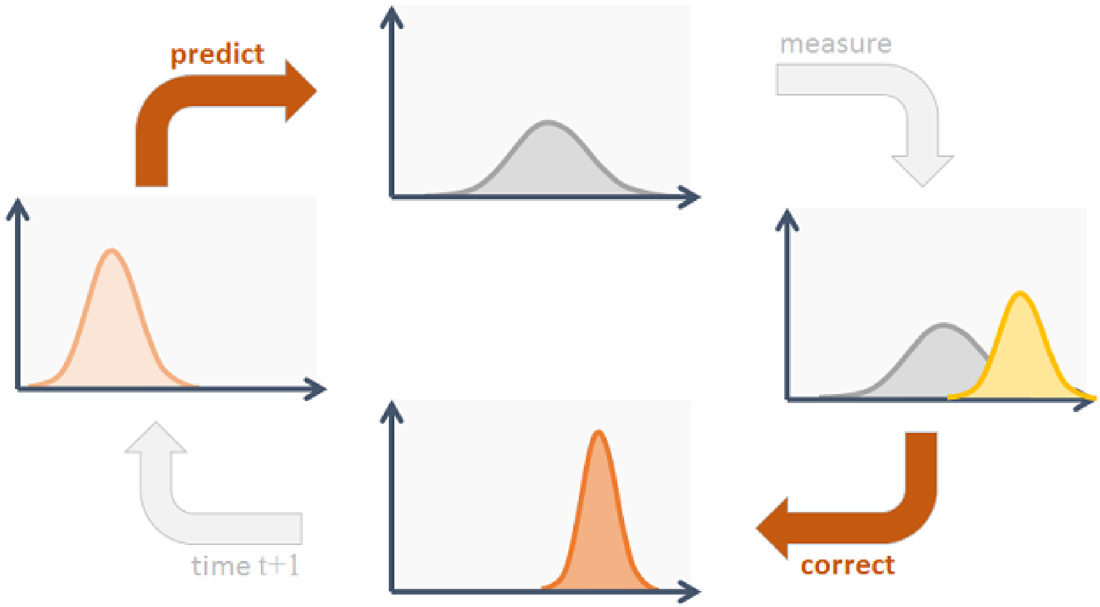


Figure A.2: Kalman Filtering Process

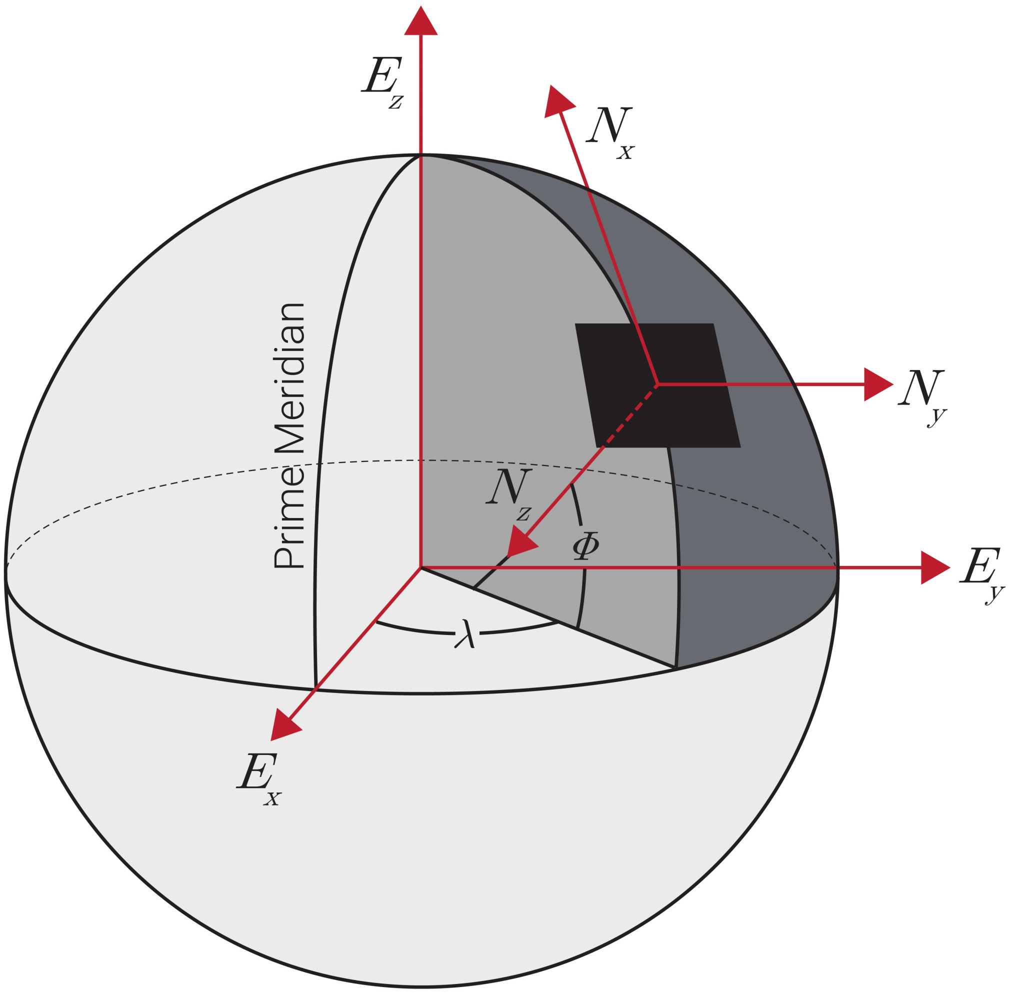


Figure A.3: Earth-Centered, Earth-Fixed (ECEF) Frame

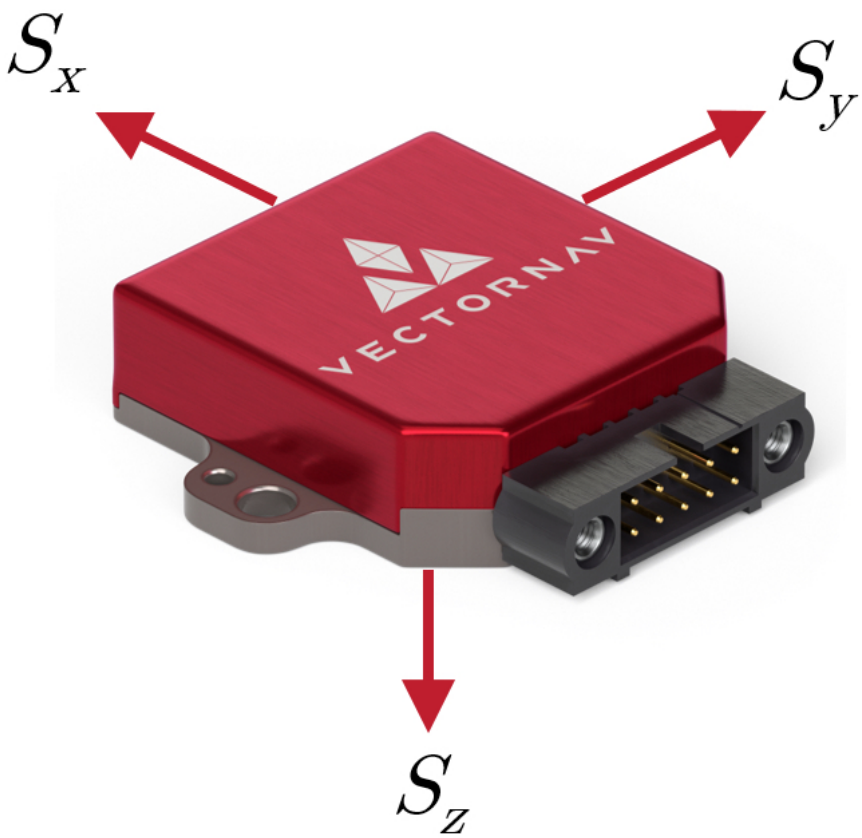


Figure A.4: Sensor Frame

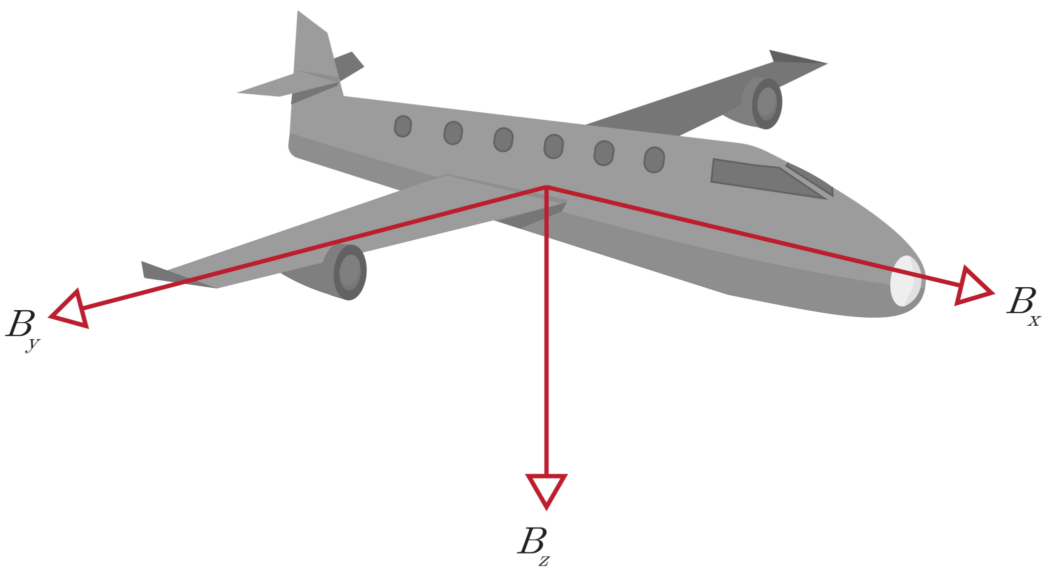


Figure A.5: Body Frame

# Appendix B

# List of Figures for INS Drift Reduction Analysis

This appendix includes a comprehensive list of figures generated during the analysis of INS drift reduction using traditional methods and SciML-enhanced approaches. Each figure is aimed at comparing specific aspects of the navigation system performance under different analysis conditions.

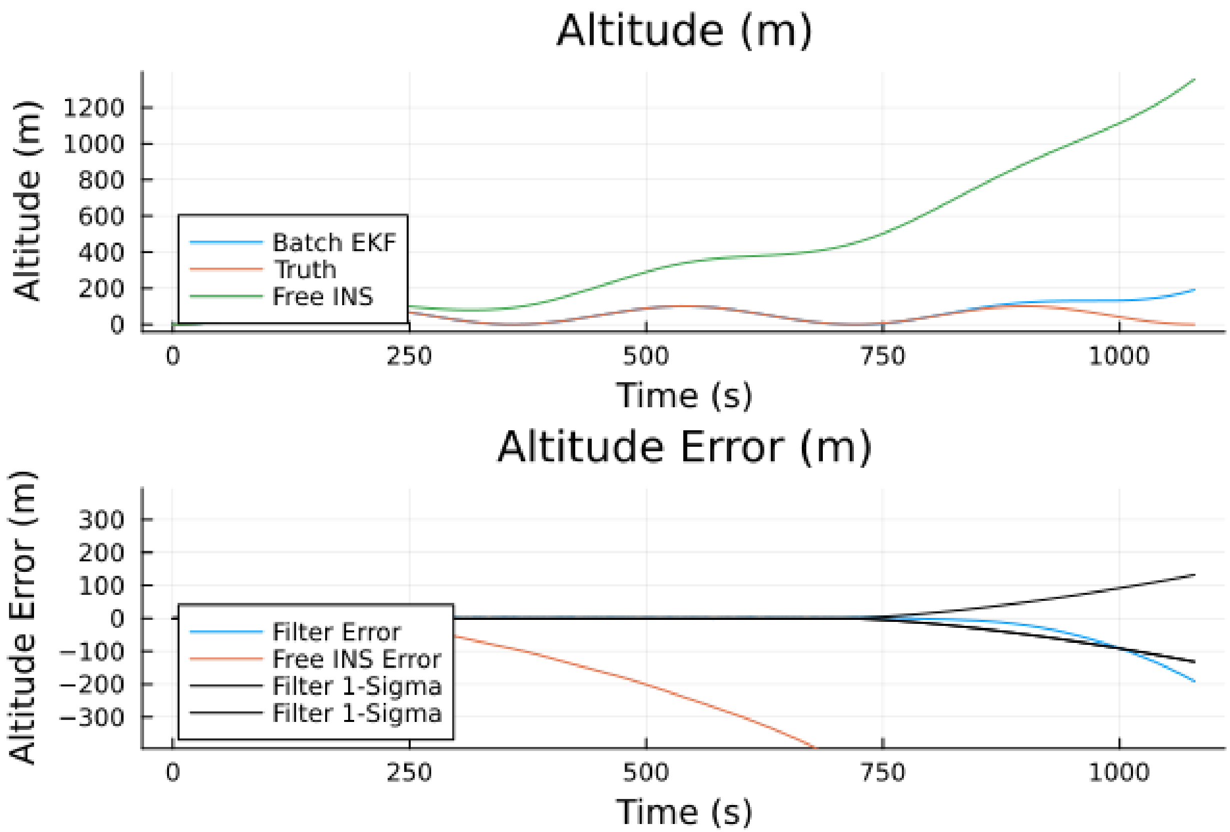


Figure B.1: Baseline altitude over time

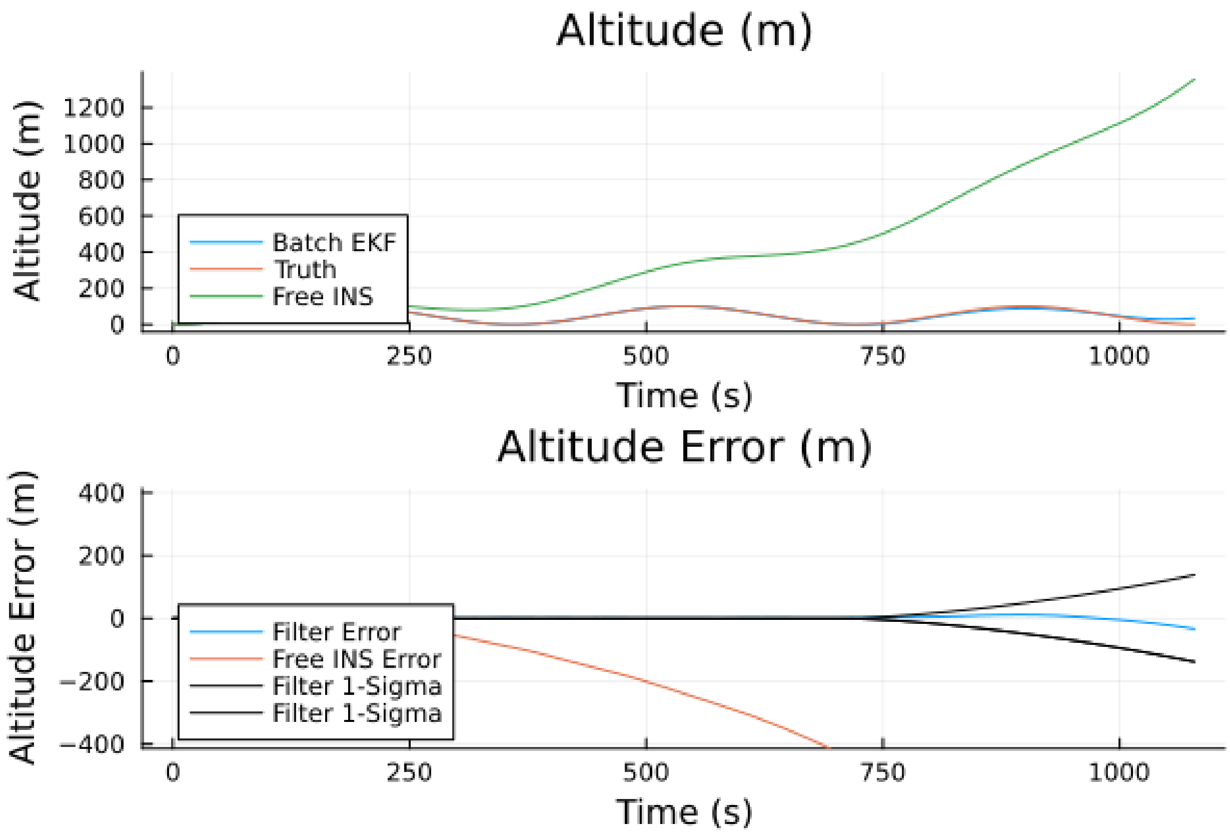


Figure B.2: Altitude comparison using the SciML approach.

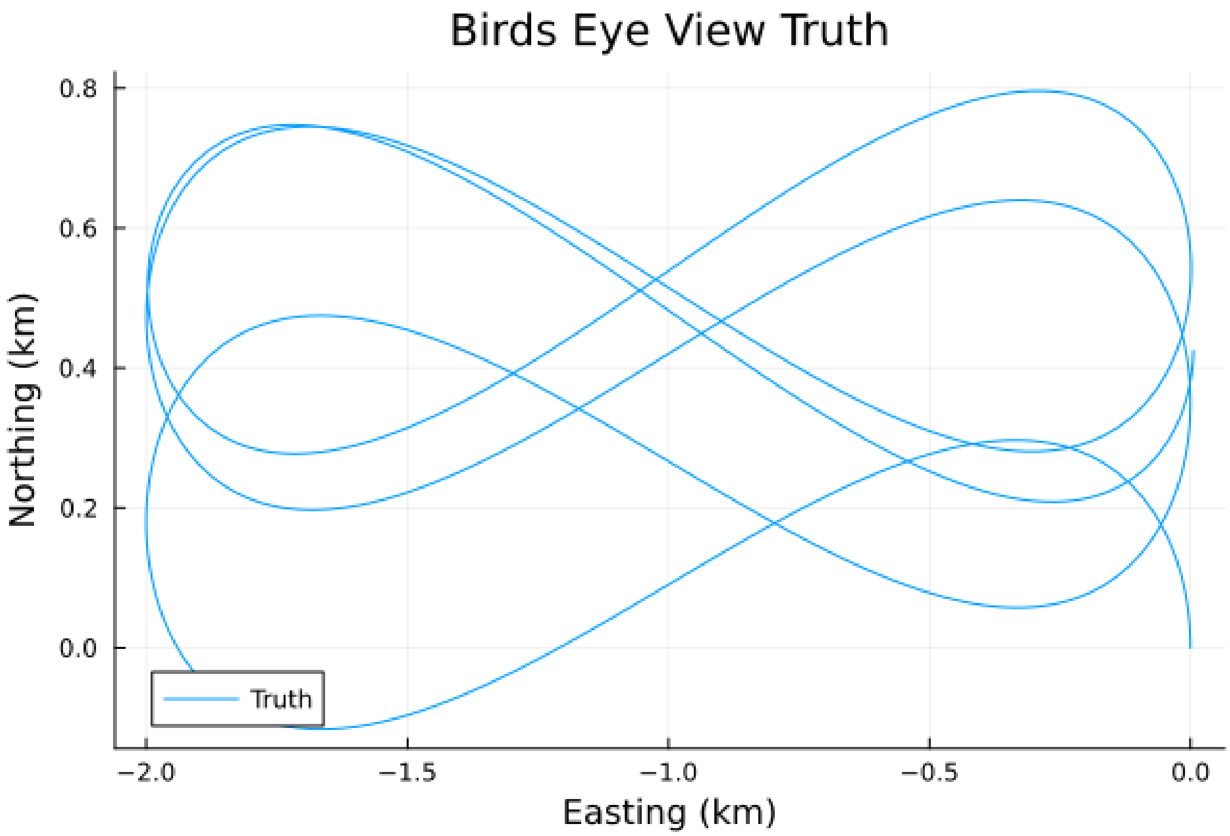


Figure B.3: Bird's eye view of truth data


Figure B.4: Birds Eye view of Truth and Free Drifting INS

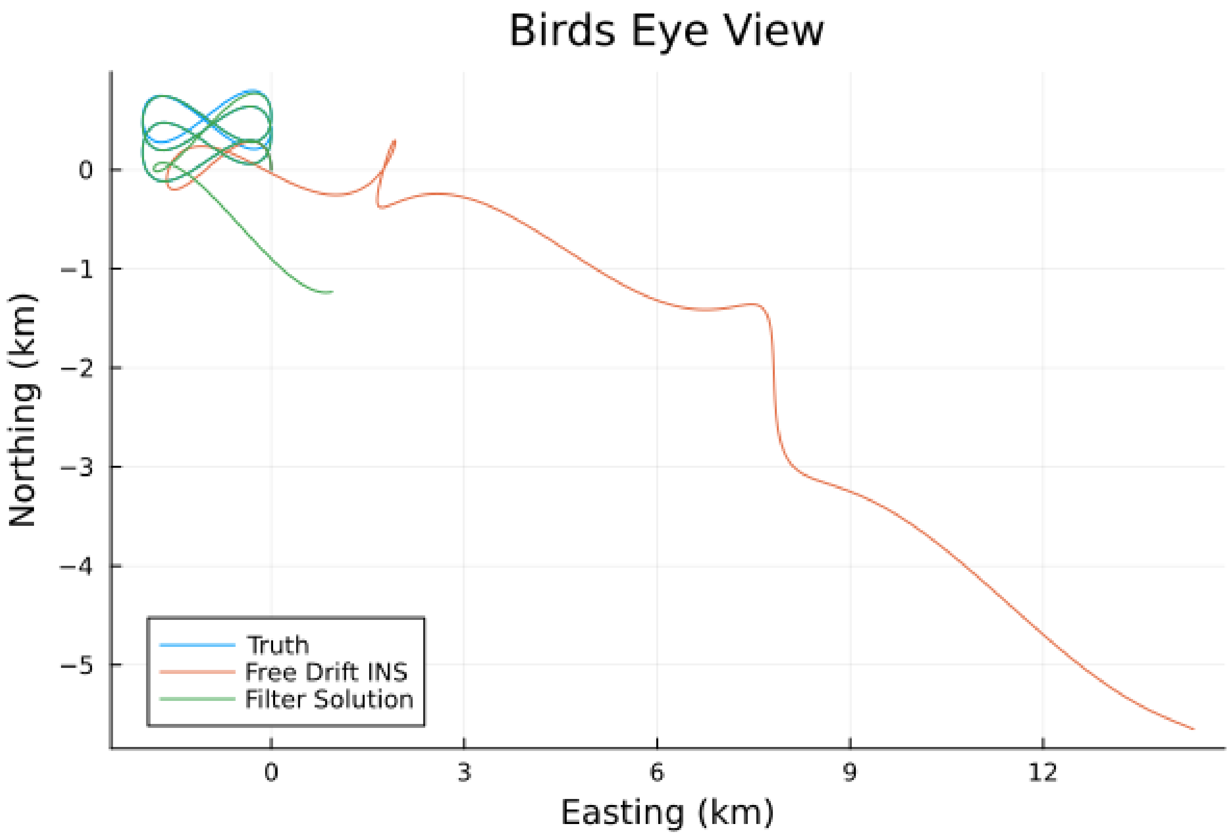


Figure B.5: Bird's eye view of the baseline trajectory.

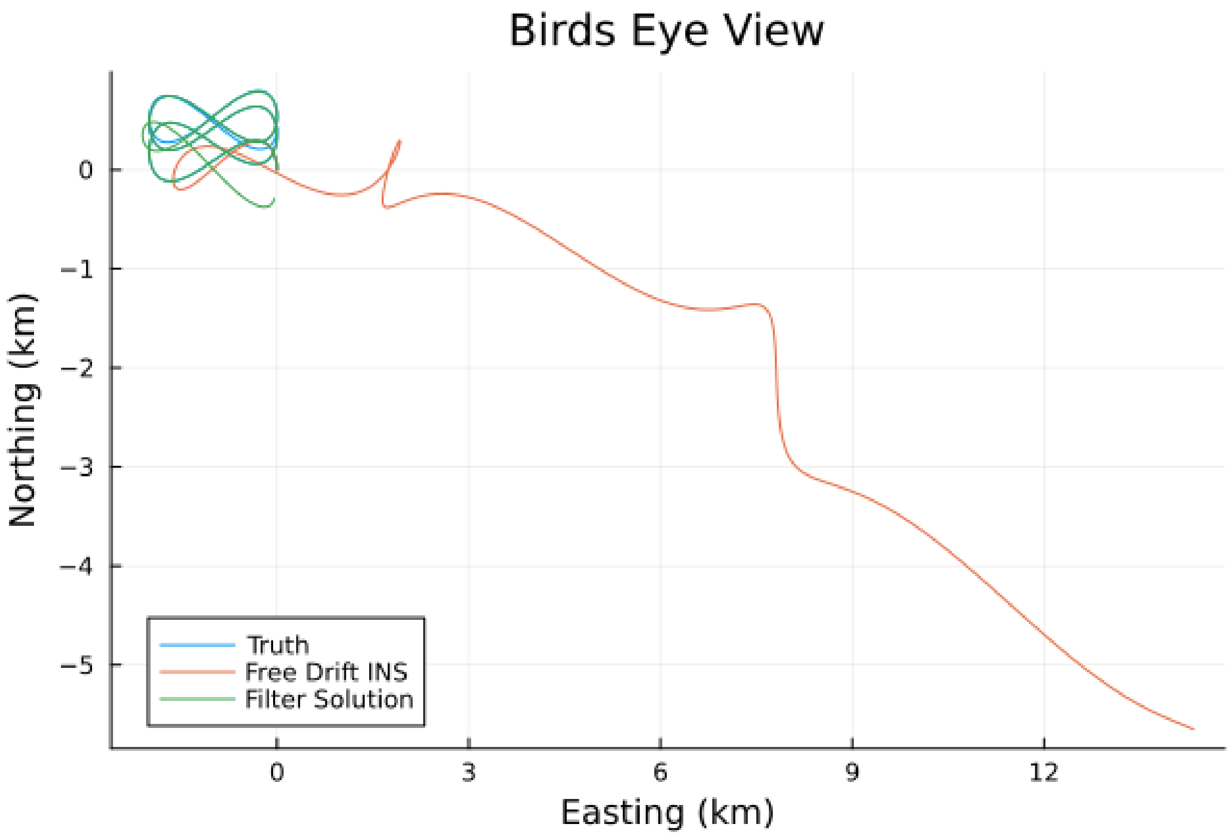


Figure B.6: Bird's eye view using the SciML model.

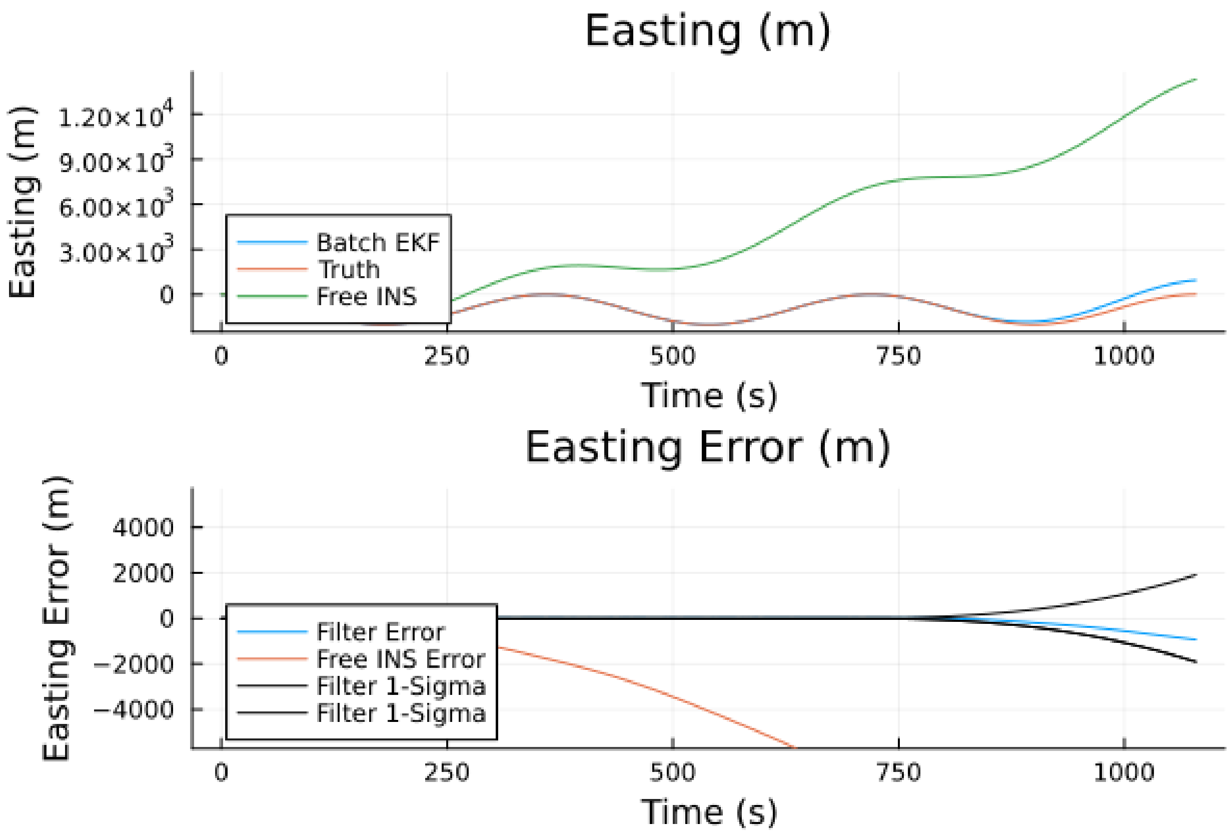


Figure B.7: Easting comparison over time for the baseline scenario.

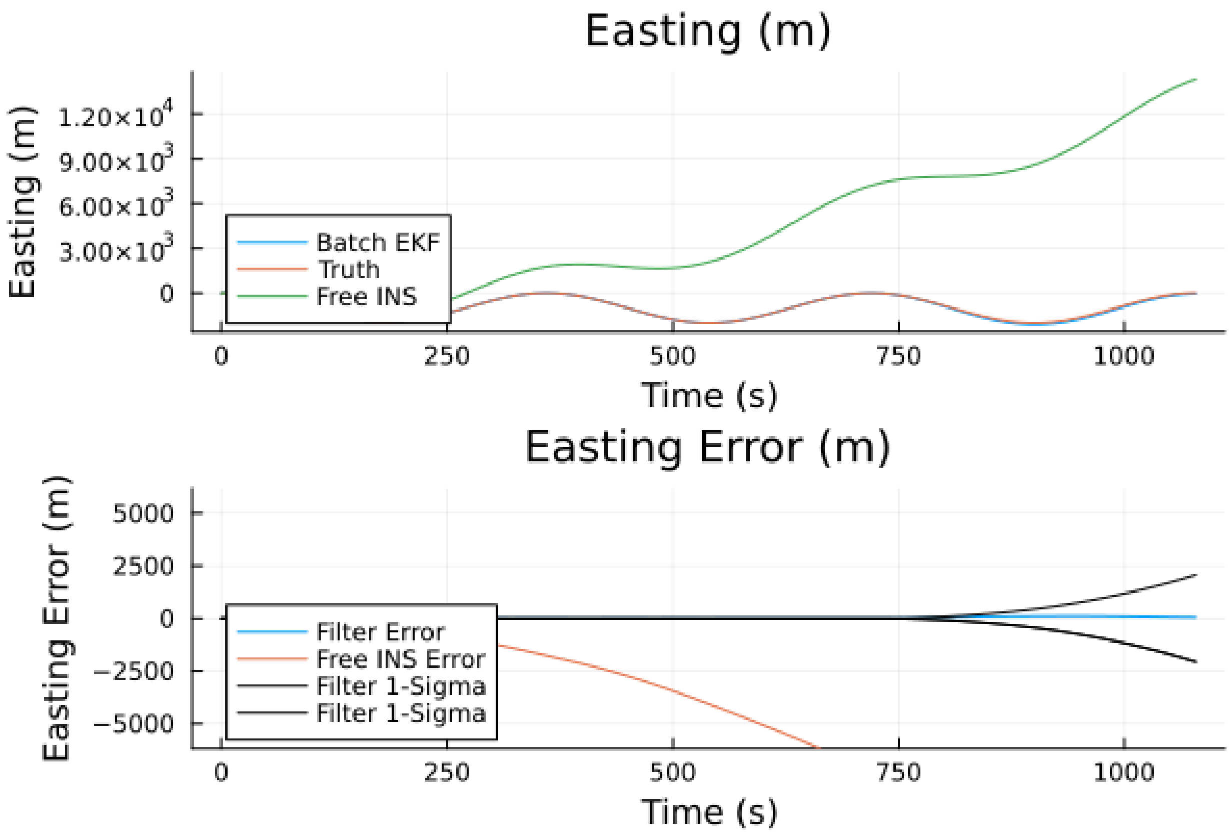


Figure B.8: Easting comparison using the SciML approach over time.

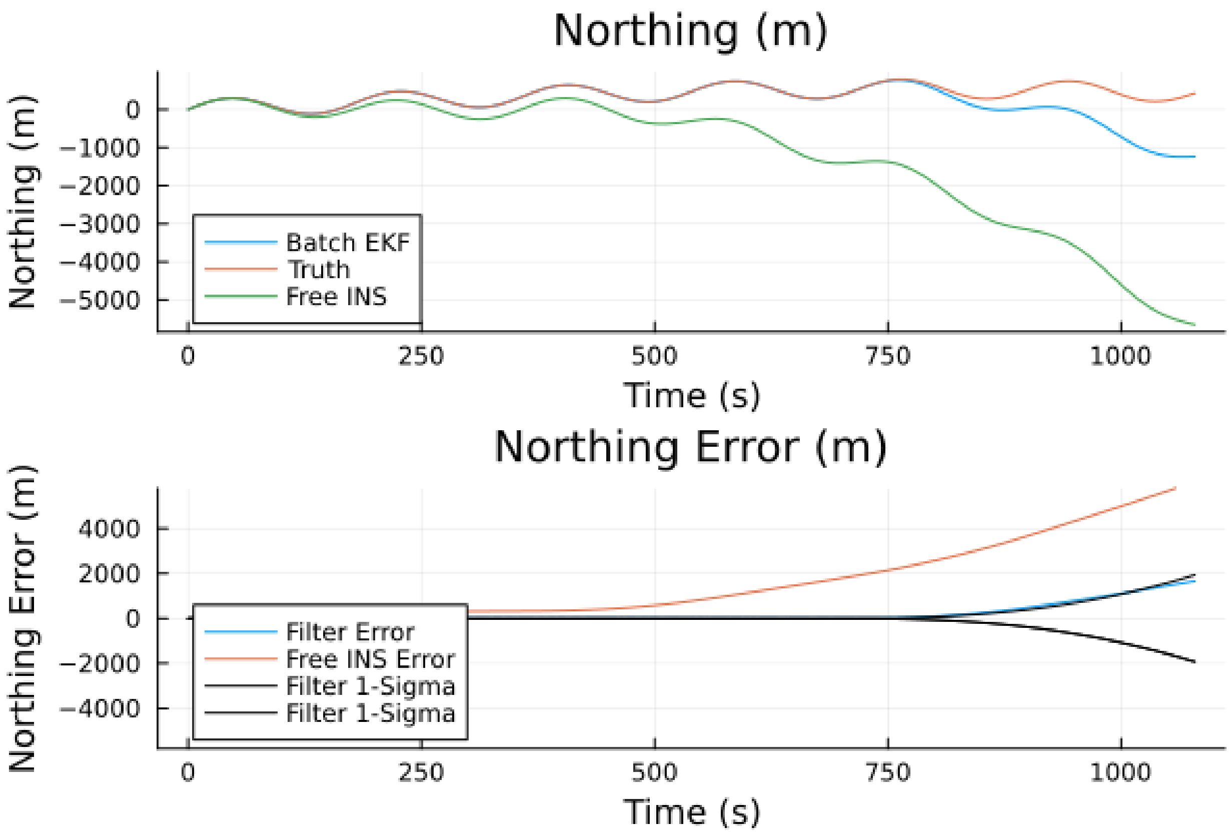


Figure B.9: Northing comparison over time for the baseline scenario.

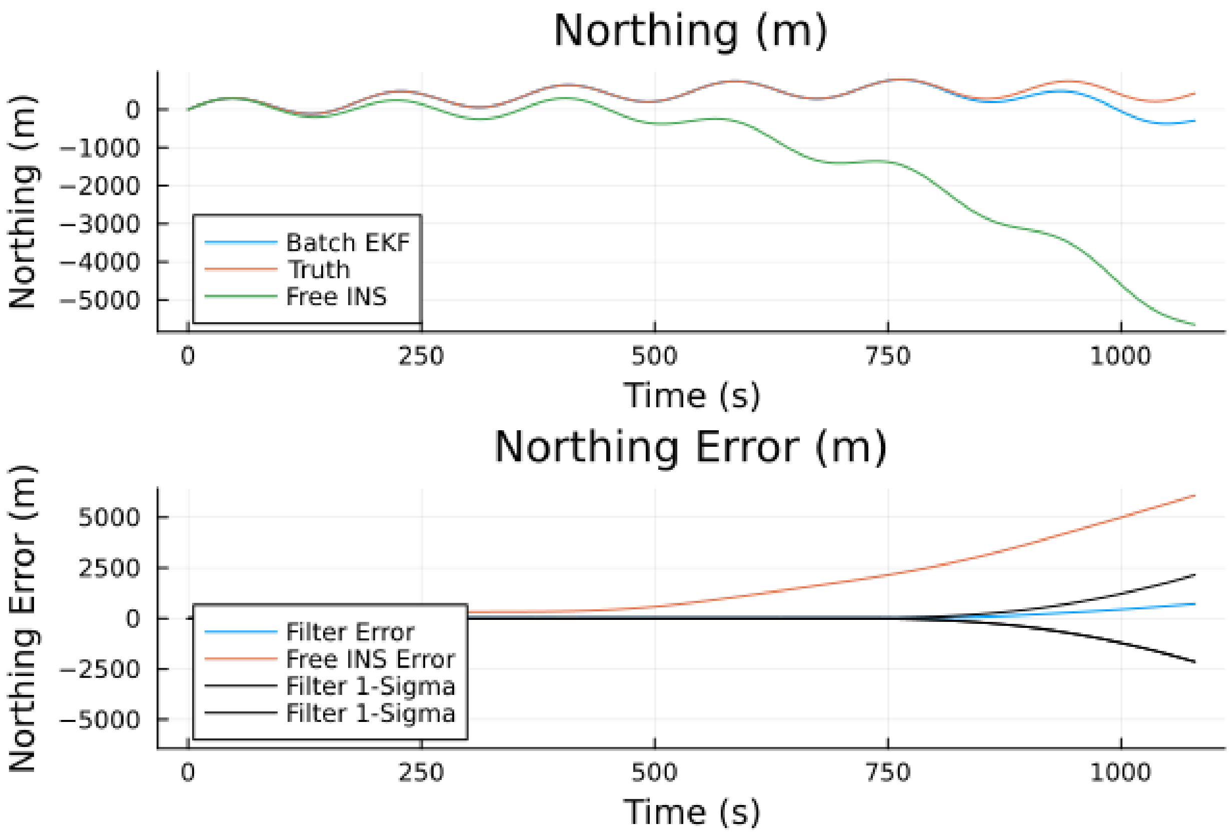


Figure B.10: Northing comparison using the SciML approach over time.

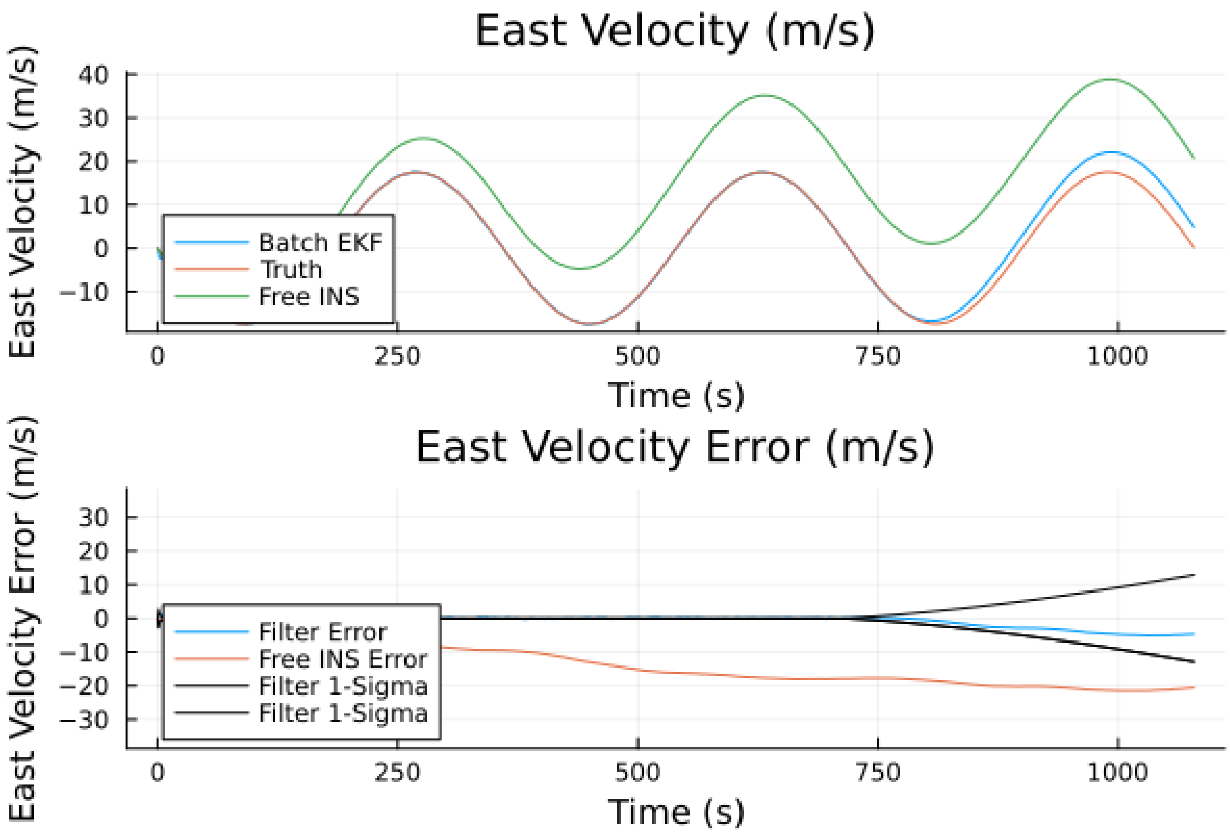


Figure B.11: East velocity error over time in the baseline model.

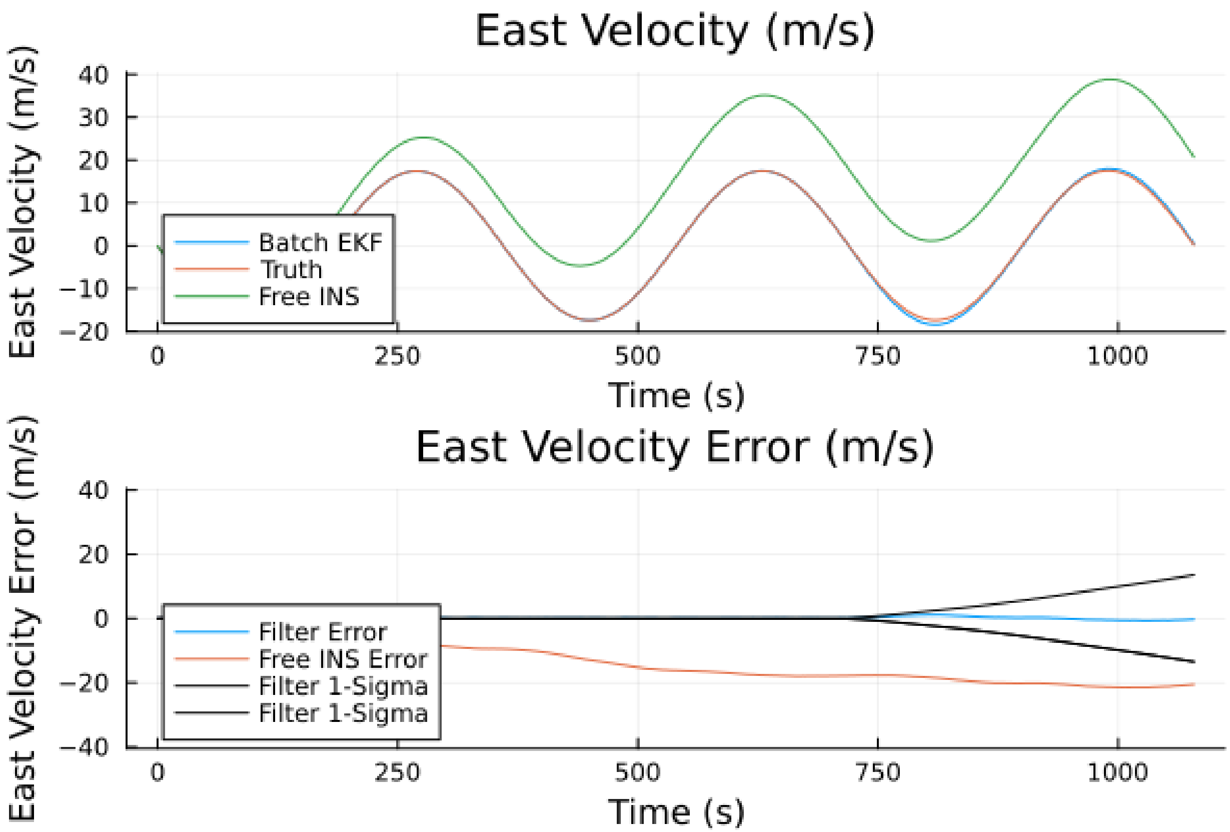


Figure B.12: East velocity error over time in the SciML model.

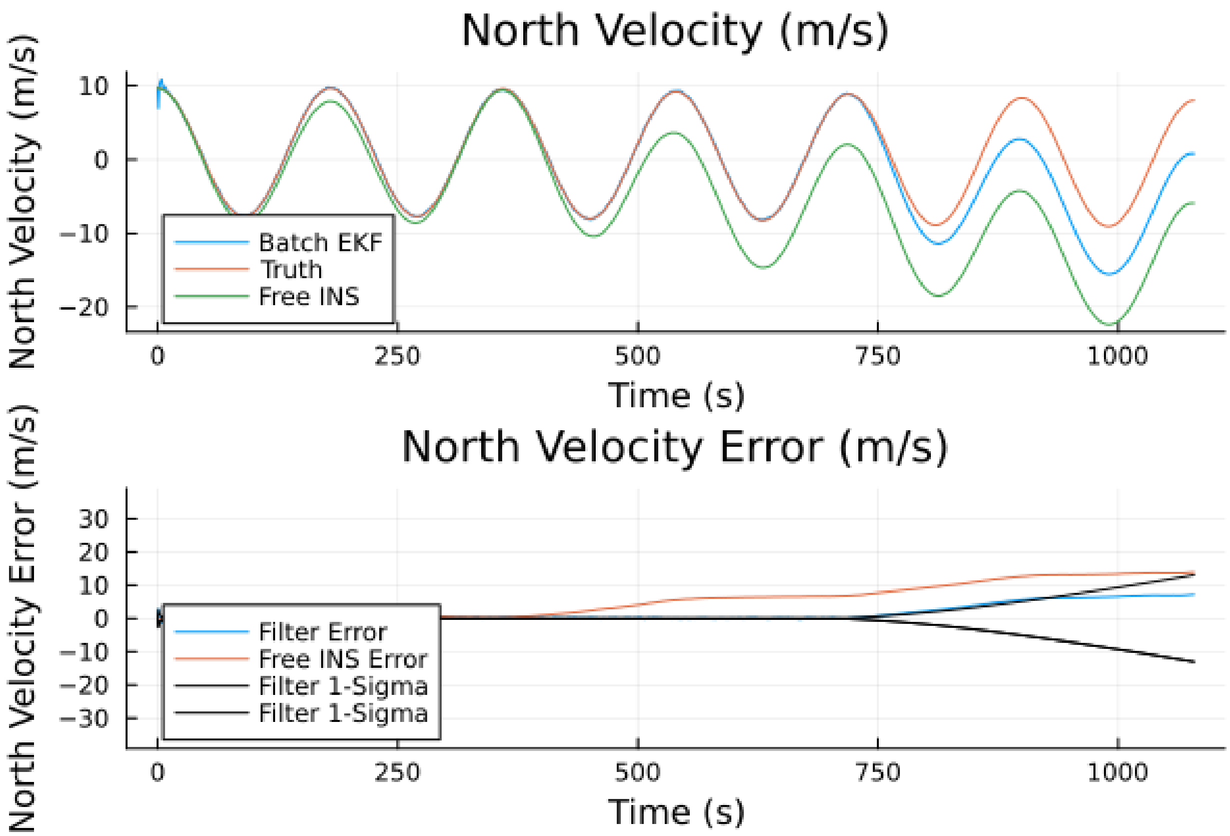


Figure B.13: North velocity error over time in the baseline model.

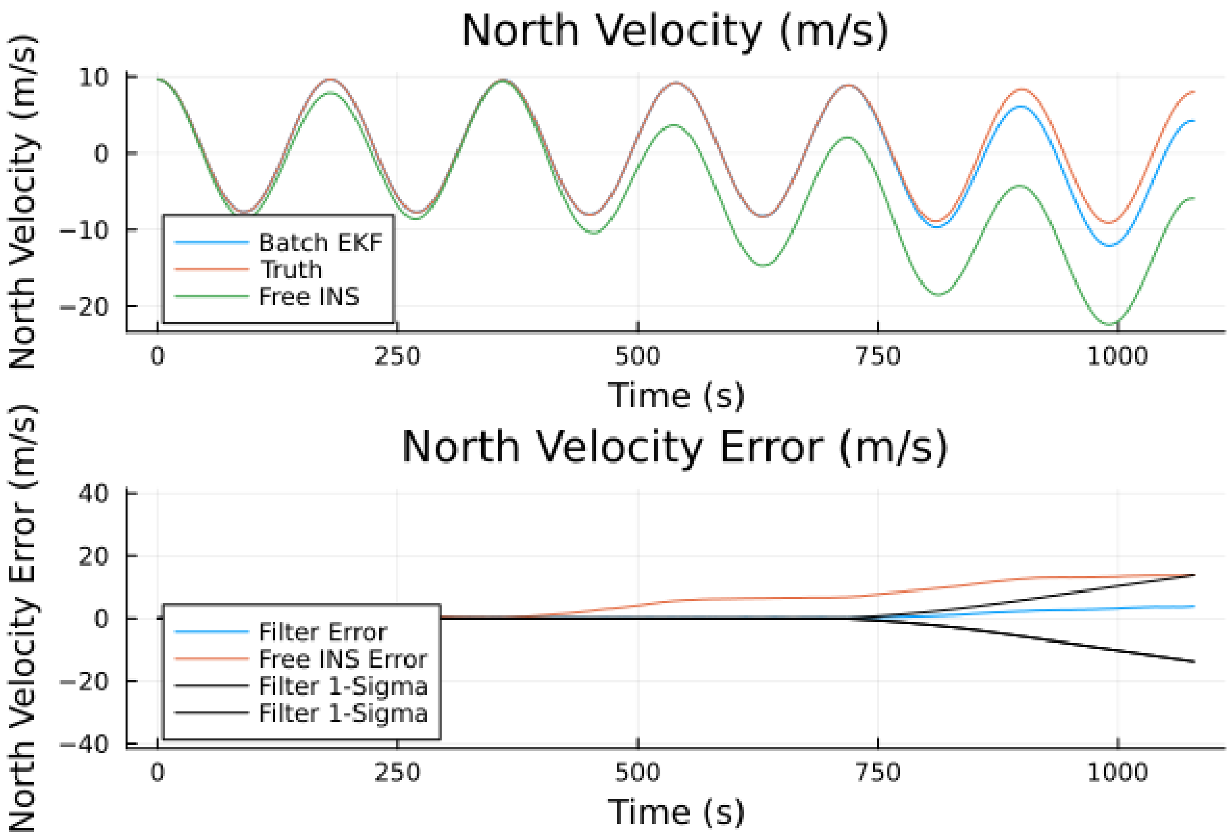


Figure B.14: North velocity error over time in the SciML model.

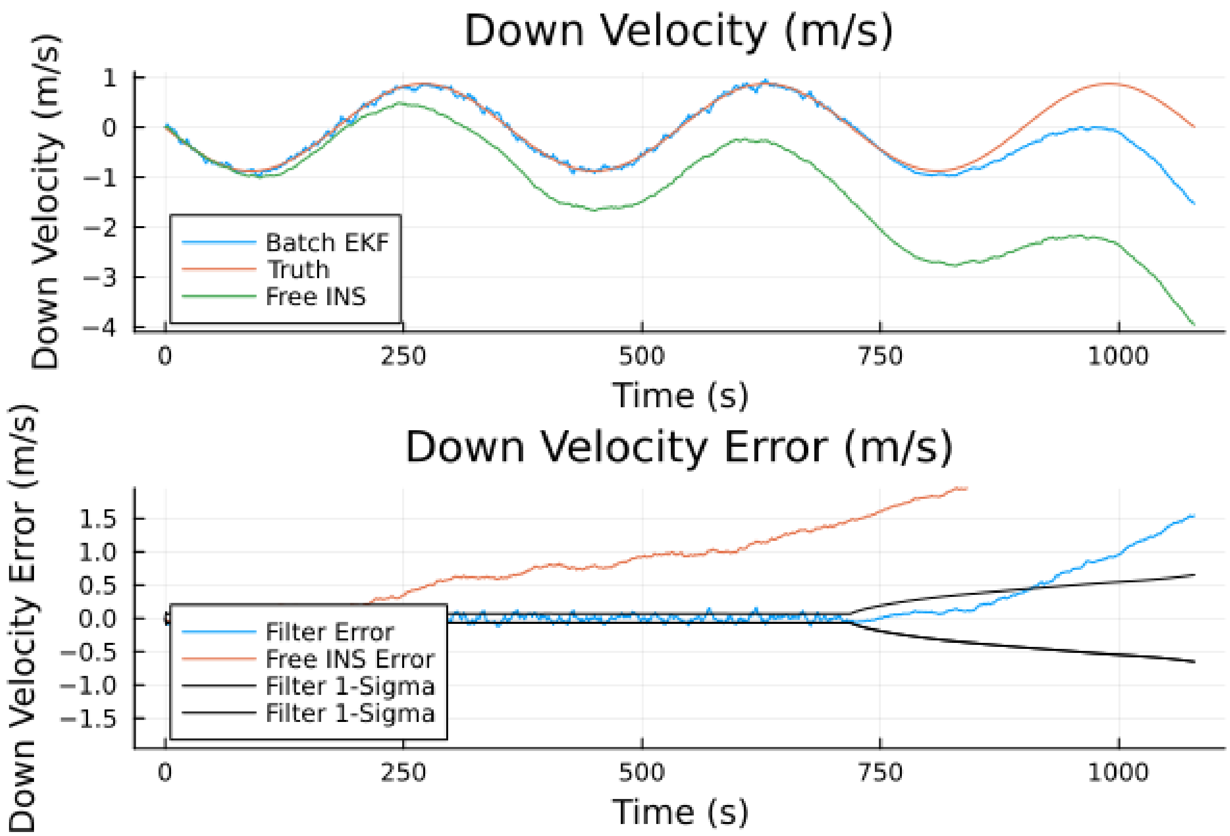


Figure B.15: Down velocity error over time in the baseline model.

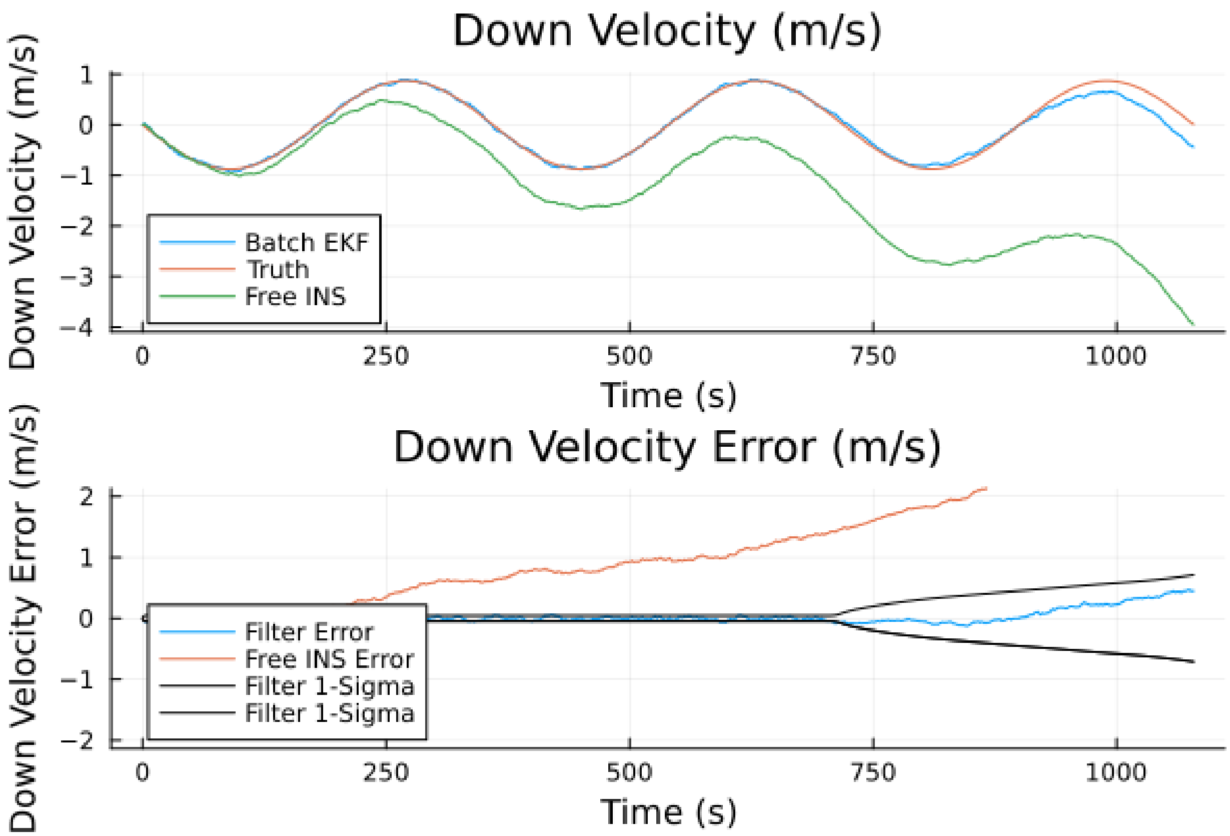


Figure B.16: Down velocity error over time in the SciML model.

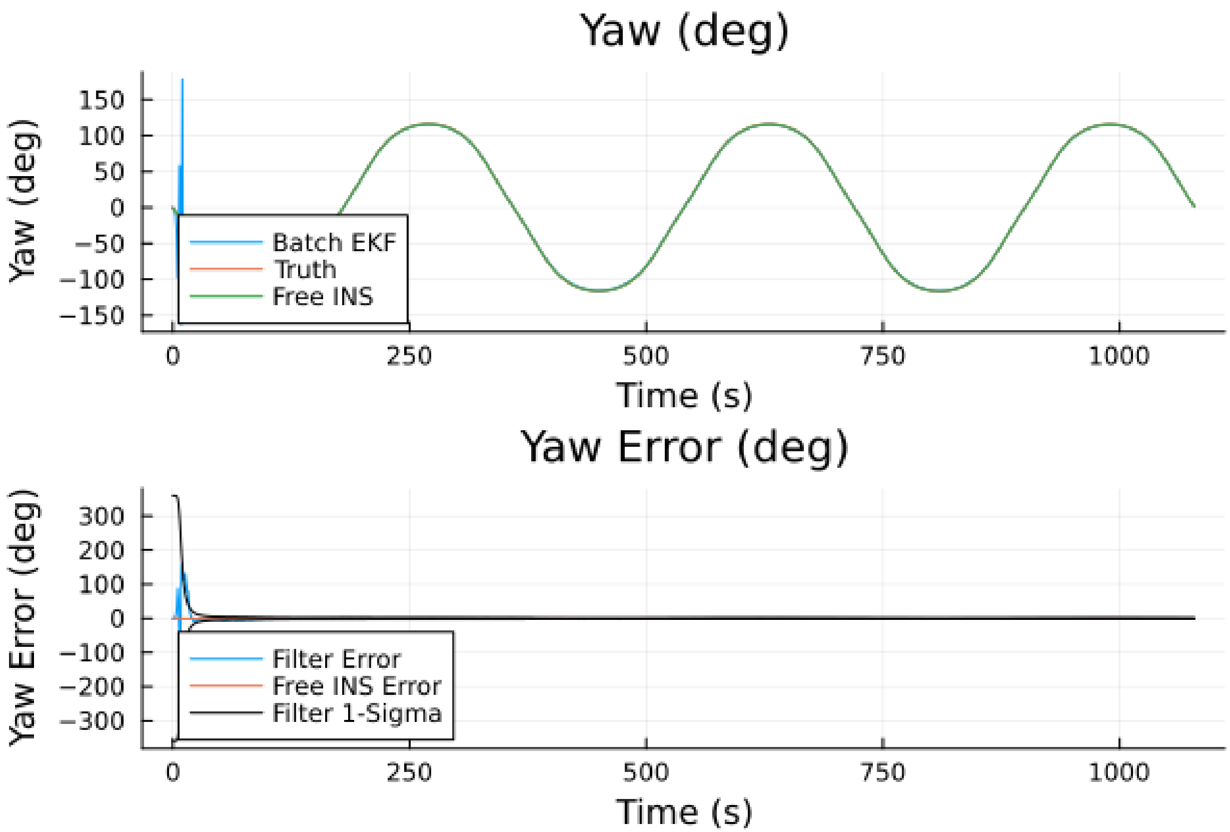


Figure B.17: Yaw comparison over time for the baseline scenario.

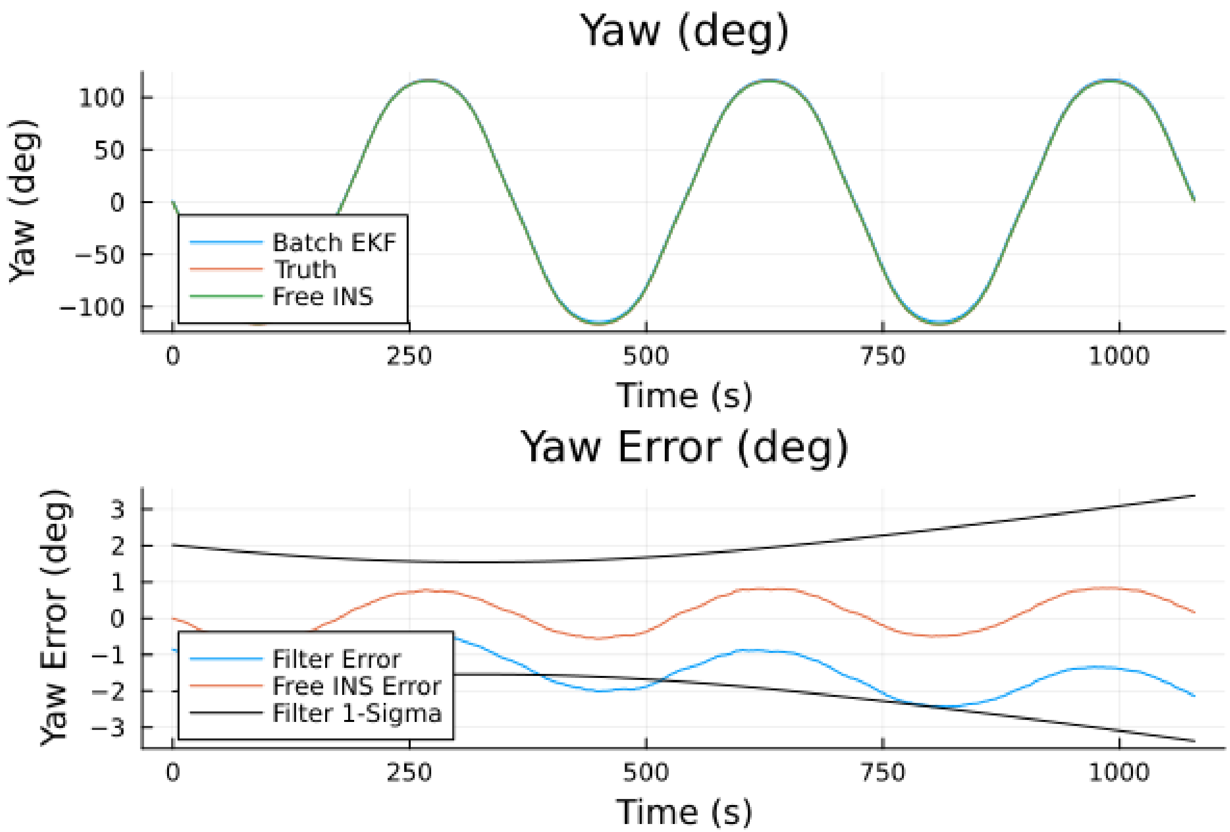


Figure B.18: Yaw comparison using SciML approach over time.

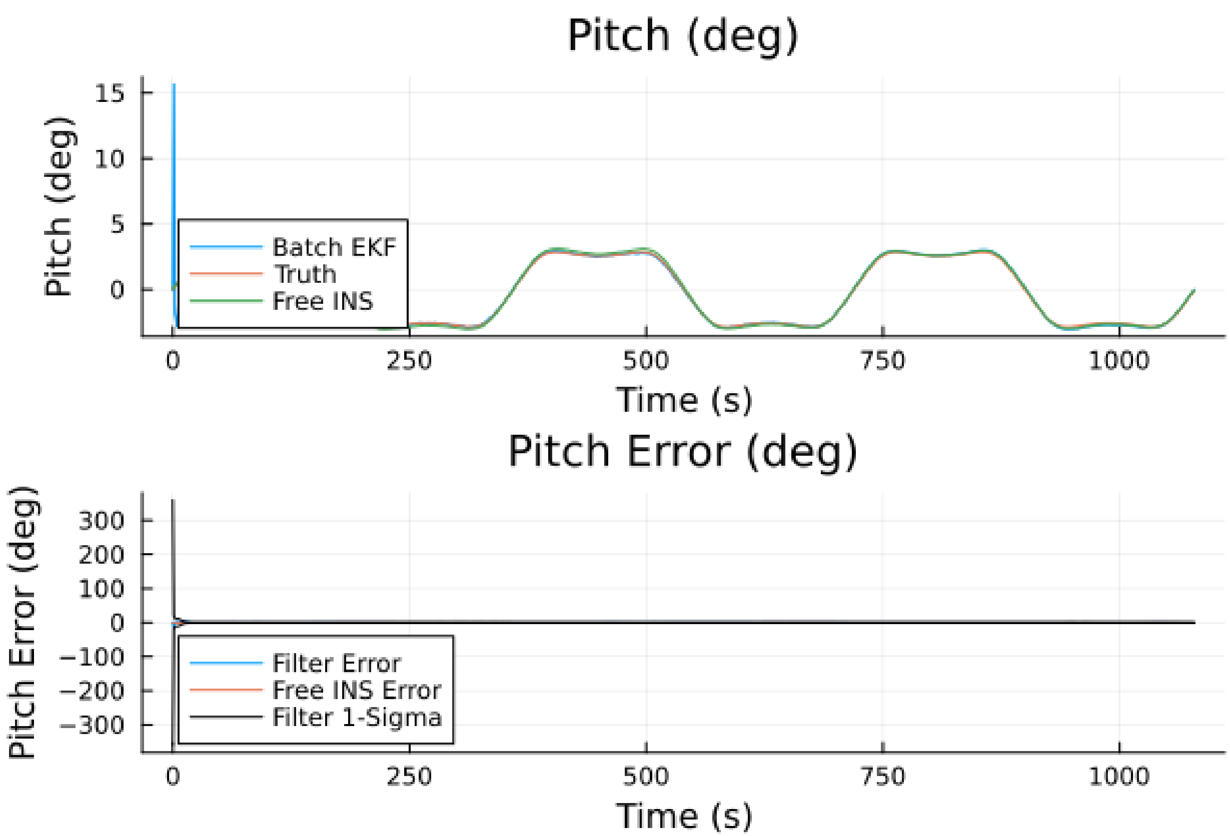


Figure B.19: Pitch comparison for the baseline INS model.

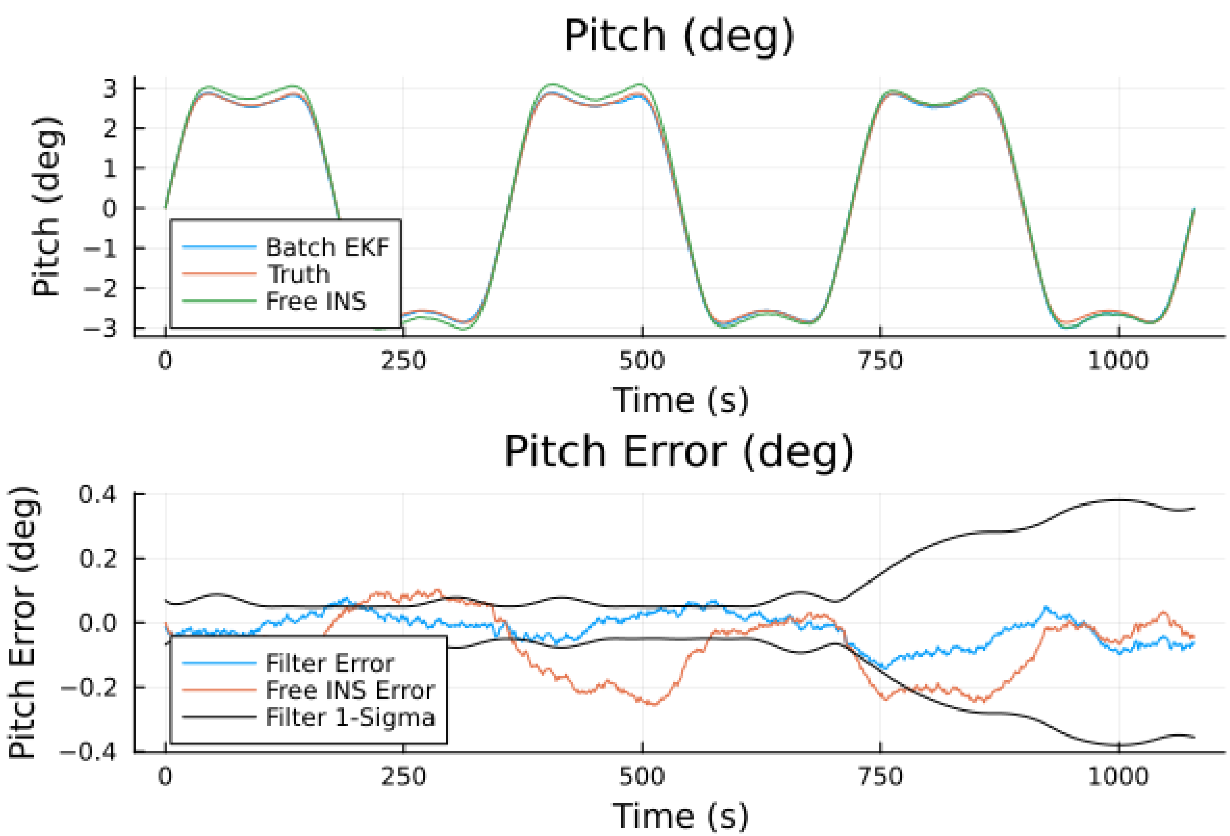


Figure B.20: Pitch comparison using the SciML approach.

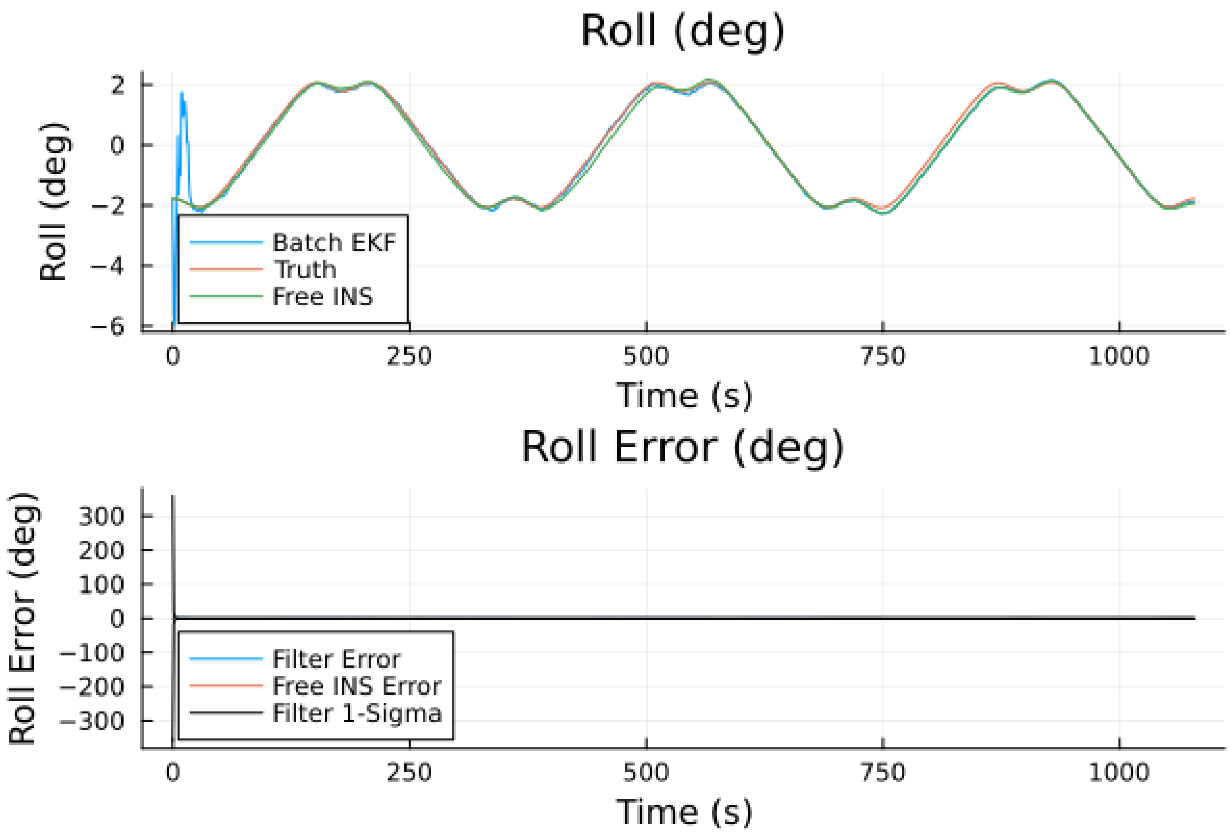


Figure B.21: Roll comparison for the baseline INS model.

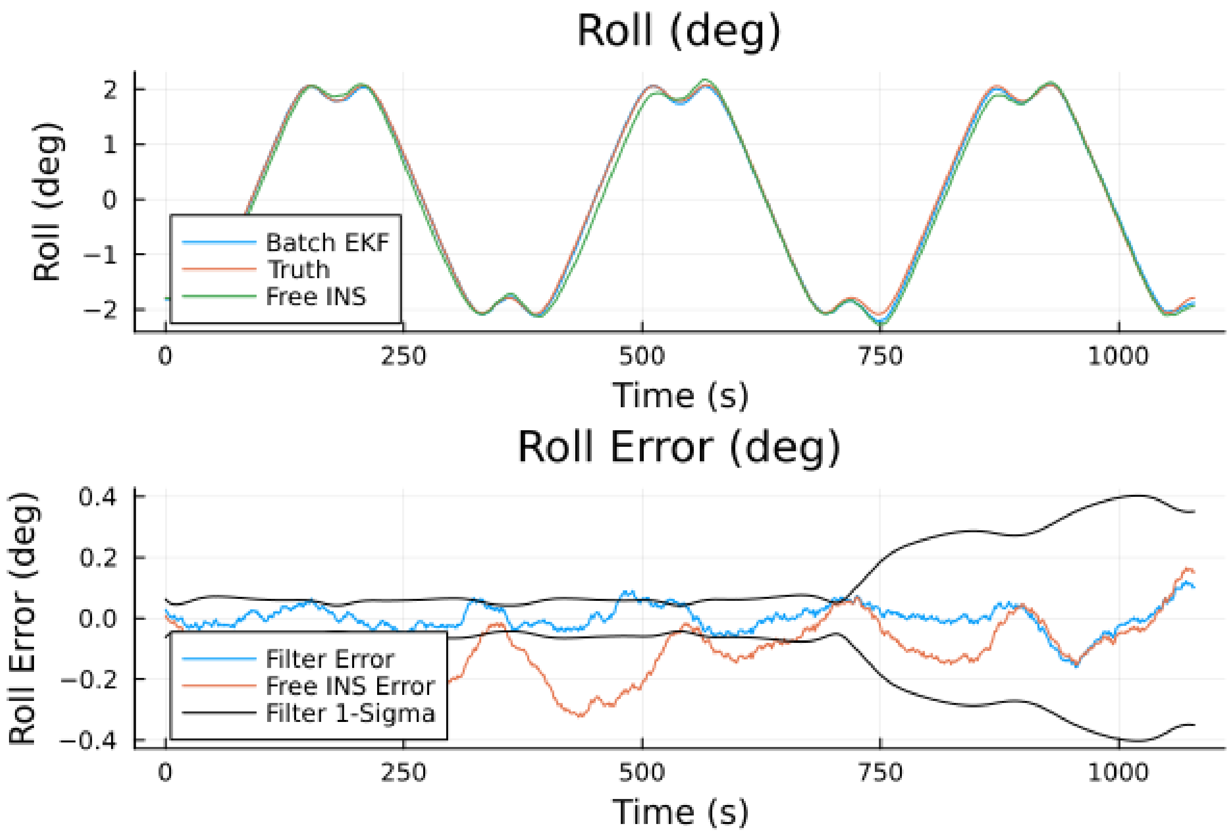


Figure B.22: Roll comparison using the SciML approach.

# References


[1] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Research report 696, Aug. 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.7402.

[2] Z. Berman and J. Powell, "The role of dead reckoning and inertial sensors in future general aviation navigation," in *IEEE 1998 Position Location and Navigation Symposium (Cat. No.98CH36153)*, 1998, pp. 510–517. DOI: 10.1109/PLANS.1998.670206.

[3] M. Kuritsky, M. Goldstein, I. Greenwood, H. Lerman, J. McCarthy, T. Shanahan, M. Silver, and J. Simpson, "Inertial navigation," *Proceedings of the IEEE*, vol. 71, no. 10, pp. 1156–1176, 1983. DOI: 10.1109/PROC.1983.12744.

[4] Advanced Navigation, *Inertial measurement unit (imu): An introduction*, Accessed: 05-09-2024, 2024. URL: https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction.

[5] R. Zanetti and C. D'Souza, "Inertial navigation," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Cham: Springer International Publishing, 2021, pp. 993–999, ISBN: 978-3-030-44184-5. DOI: 10.1007/978-3-030-44184-5_100036. URL: https://doi.org/10.1007/978-3-030-44184-5_100036.

[6] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems (GNSS Technology and Applications)*, English, Hardcover. Artech House on

Demand, Apr. 1, 2013, p. 776, ISBN: 978-1608070053. URL: https://lead.to/amazon/com/?op=bt&la=en&cu=usd&key=1608070050.

[7] Noureldin, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*, English, Hardcover. Springer, Oct. 28, 2012, p. 332, ISBN: 978-3642304651. URL: https://lead.to/amazon/com/?op=bt&la=en&cu=usd&key=3642304656.

[8] F. E. Daum, "Extended kalman filters," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. London: Springer London, 2013, pp. 1–3, ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_62-2. URL: https://doi.org/10.1007/978-1-4471-5102-9_62-2.

[9] D. Jurić. "Object tracking: Kalman filter with ease." Accessed: 2024-05-09, CodeProject. (Jan. 2015), URL: https://www.codeproject.com/Articles/865935/Object-Tracking-Kalman-Filter-with-Ease (visited on 05/09/2024).

[10] G. A. Terejanu, *Extended kalman filter tutorial*, https://homes.cs.washington.edu/~todorov/courses/cseP590/readings/tutorialEKF.pdf, Accessed: [insert date here], Buffalo, NY 14260, 2008.

[11] C. Chen, C. X. Lu, A. Markham, and N. Trigoni. "Deep learning based pedestrian inertial navigation: Methods and dataset." (2018), URL: http://deepio.cs.ox.ac.uk/ (visited on 05/07/2024).

[12] D. Woodburn, "Tutorial on inverse mechanization," in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, Denver, Colorado, Sep. 2023, pp. 23–37. DOI: 10.33012/2023.19180. URL: https://doi.org/10.33012/2023.19180.