Estimating the State of a Dynamically Evolving System

by Kabir Samlal

Student Number: 4869621
Academic Supervisor: Dr. R.C. Kraaij
Thesis Committee: Dr. R.C. Kraaij, Dr. F. Mies

1 Abstract

Estimating the state of dynamically evolving systems is a fundamental challenge across diverse fields such as robotics, navigation, economics, and environmental monitoring. This thesis explores and compares three prominent state estimation methods: the Kalman Filter (KF), the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF), each tailored to handle specific complexities encountered in real-world applications.

The foundational Kalman Filter is rigorously examined first, deriving its algorithm through Bayesian inference and the fusion of multiple estimates. A comparative analysis of these approaches highlights the KF's robustness in linear systems while acknowledging limitations in nonlinear environments.

The thesis then transitions to the Extended Kalman Filter, which extends the KF to nonlinear systems by linearizing state equations. Detailed mathematical derivation and comparative studies underscore the EKF's enhanced capabilities in handling complex dynamics, yet reveal challenges in accuracy and computational cost.

Moving further, the Unscented Kalman Filter is introduced as a non-linear state estimation method utilizing the Unscented Transform. Detailed exploration and mathematical formulation demonstrate its effectiveness in addressing uncertainties, presenting a viable alternative to both KF and EKF in scenarios where linearization proves inadequate.

To validate these methodologies, simulations are conducted using real-world data from the KITTI dataset, comprising of GPS and IMU measurements. Ground truth trajectories and non-linear variables such as yaw rates and forward velocities are utilized, showcasing each filter's ability to estimate and track dynamic system states accurately.

Results from simulations are analyzed using performance metrics including Normalized Estimation Error Squared (NEES) and Root Mean Squared Error (RMSE), providing quantitative insights into filter performance relative to ground truth. These evaluations emphasize the strengths and limitations of each method across various application domains, supporting informed decisions on filter selection based on specific system dynamics and measurement characteristics.

In conclusion, this thesis contributes to a comprehensive analysis and comparative study of state estimation methods essential for navigating the complexities of dynamically evolving systems. By bridging theoretical advancements with practical insights, it lays a foundation for future research and application in fields requiring precise state estimation amidst dynamic change.

2 Lay Summary

Imagine you're driving with the GPS guiding you through unfamiliar streets. The GPS is supposed to tell you where you are, but sometimes it's not quite right. Maybe it loses signal in a tunnel or gives confusing directions in a dense city. In reality, there are many reasons why GPS measurements can be wrong – bad connections, satellite positions or atmospheric conditions.

Now, imagine if you couldn't trust the GPS at all times. What if it suddenly tells you that you're in another country, even though you know you're not? This is where the challenge lies: how do we accurately estimate where we are or where we're going, even when our measurements are noisy or incomplete?

This problem isn't just about GPS in cars. It's everywhere. In robotics, economics, environmental monitoring – in almost every field, measurements can be imprecise due to errors, noise, or unexpected events. Mathematicians and scientists have developed tools called filters and estimators to tackle this challenge.

For example, in your car, there are sensors like accelerometers that measure your car's acceleration. Mathematicians can use these measurements in filters and estimators to better provide a location estimate, even if the GPS signal is lost or inaccurate. These tools help by combining predictions with measurements to give accurate estimates of a system's state.

This thesis explores one of the most widely used filters, called the Kalman Filter. It's a powerful tool that combines predictions with measurements to give accurate estimates of a system's state, even when the measurements are flawed. By understanding and evaluating the Kalman Filter, we gain insights into how to handle uncertainty and make better decisions based on imperfect information.

Contents

3 Introduction

3.1 Background

In many real-world applications, accurately estimating the state of a system is crucial for effective decision-making, control, and analysis. Whether it is tracking the position of a moving object, predicting financial market trends, or controlling the dynamics of a robotic system, having a reliable estimate of the system's state is essential. However, obtaining precise measurements is often challenged by noise, uncertainties, and incomplete information. This is where filters and estimators come into play.

Filters and estimators are mathematical tools designed to extract useful information from noisy and uncertain data. They aim to provide an accurate estimate of the true state of a system by combining available measurements with prior knowledge about the system's dynamics. These tools are fundamental in fields such as engineering, economics, and natural sciences, where dealing with uncertainties and making predictions based on incomplete information are common tasks.

Consider the everyday example of using a GPS navigation system while driving. Most of us have experienced moments where the GPS signal becomes unreliable due to factors such as tall buildings, tunnels, or atmospheric conditions. Despite its occasional inaccuracies, GPS navigation generally helps us navigate through unknown territories with reasonable accuracy. However, in scenarios where GPS measurements fail or produce extreme outliers (like placing us on the wrong road), relying solely on GPS data can lead to erroneous conclusions. This illustrates the broader challenge: how can we accurately estimate the true position of a vehicle, or any system, in the presence of imperfect, noisy, or incomplete information?

Modern vehicles are equipped with additional sensors such as accelerometers and Inertial Measurement Units (IMUs). For instance, accelerometers measure the car's acceleration, and IMUs provide data on orientation and angular velocities. By integrating these measurements with GPS data, we can improve the accuracy of the vehicle's location estimate. These sensors can help predict the car's position even when GPS data is unreliable or lost, demonstrating how multiple sources of information can be combined to enhance estimation.

This is where mathematicians and engineers apply advanced estimation techniques like the Kalman filter (KF). Introduced by Rudolf Kalman in 1960 [2], the Kalman filter is a powerful tool that combines predictions and measurements to estimate the state of a system, even in the presence of noise and uncertainty. It iteratively refines its estimates through prediction and update steps, effectively integrating prior knowledge with current measurements to provide optimal state estimates.

This thesis provides an intuitive explanation of the Kalman filter (KF) and its applications, followed by a mathematically rigorous exploration of its underlying Bayesian principles and the method of fusing estimates. In *Section 4*, we delve into the Extended Kalman Filter (EKF), exploring its algorithm, mathematical derivation, and comparative analysis with the KF, while also examining its limitations.

Next, *Section 5* addresses the limitations of the EKF by introducing the Unscented Kalman Filter (UKF). We provide a detailed explanation of its algorithm, key mathematical formulations, and comparative analysis with the EKF, highlighting its advantages and addressing its limitations.

Finally, in *Section 6*, we perform a comparative analysis of the KF, EKF, and UKF using real-world data from the KITTI dataset. Performance metrics such as Normalized Estimation Error Squared (NEES) and Mean Squared Error (MSE) provide quantitative insights into the filters' performance relative to ground truth.

By understanding the principles and mechanics of the Kalman filter and its variants, as well as their respective strengths and limitations, we gain valuable insights into modern estimation theory and their applications across diverse fields of science and engineering.

3.2 Prerequisites

In many scientific and engineering disciplines, accurately estimating the state of a dynamic system is a fundamental task. A dynamic system is typically represented by a set of equations that describe its evolution over time. However, these systems are often subject to uncertainties and noise, making it challenging to obtain precise measurements. Estimators and filters are used to mitigate the effects of noise and provide accurate state estimates.

3.2.1 Problem Setup: State Space Model

A common framework for modeling dynamic systems is the state space model. In this model, the system is described by two main equations: the state equation and the measurement equation.

State Equation

The state equation describes how the state of the system evolves over time. Ideally, from a deterministic standpoint, it can be represented as:

$$x_k = A_k x_{k-1}, \tag{1}$$

where x_k is the state vector at time k, and A_k is the state transition matrix that describes how the state evolves from time $k-1$ to k.

From a physics perspective, A_k represents the idealized evolution encapsulates the fundamental laws governing the system dynamics, assuming no external influences or uncertainties.

Example

Consider tracking the position of an object in motion where x_k could represent the position and velocity at time k. Here, A_k would encapsulate the laws of motion, such as constant velocity or acceleration.

In practice, however, real-world systems are subject to various influences and uncertainties that are not accounted for in the idealized model. Therefore noise must be added to the idealized model:

$$x_k = A_k x_{k-1} + (noise), \tag{2}$$

To accurately reflect these complexities, we revise the state equation to incorporate additional factors:

$$x_k = A_k x_{k-1} + B_k u_k + w_k, \tag{3}$$

where:

- $B_k u_k$ represents the impact of control inputs u_k on the state evolution. These inputs can include forces, commands, or interventions applied to the system.
- w_k denotes process noise, which accounts for unmodeled dynamics, disturbances, or random fluctuations affecting the system. It is assumed to follow a Gaussian distribution with zero mean and covariance matrix Q_k.

Continuing with the vehicle tracking example, $B_k u_k$ could represent the influence of acceleration due to varying road conditions, while w_k could account for unpredictable factors like wind gusts or measurement inaccuracies from GPS sensors.

Measurement Equation

The measurement equation relates the state of the system to the measurements taken at time k:

$$y_k = C_k x_k + v_k, \tag{4}$$

where:

- y_k is the measurement vector at time k, capturing observations or sensor readings.
- C_k is the measurement matrix that maps the state vector to the measurement vector. It defines how the state variables contribute to the observed measurements.
- v_k is the measurement noise, assumed to be Gaussian with zero mean and covariance matrix R_k. It accounts for errors or disturbances in the measurement process.

The state space model, defined by equations (3) and (4), provide a comprehensive framework for modeling dynamic systems, accounting for both the evolution of the system state and the relationship between the state and measurements.

3.2.2 Intuitive Examples

To illustrate the state space model, consider the following examples:

- **Tracking a Moving Vehicle**: The state vector x_k could represent the position and velocity of a vehicle at time k. The control input u_k might be the acceleration, and the measurements y_k could be obtained from GPS sensors. The matrices A_k, B_k, and C_k would be determined based on the physical laws of motion and the sensor characteristics.
- **Economics**: In an economic system, the state vector x_k might include variables such as GDP, inflation rate, and unemployment rate. The control input u_k could be government policies, and the measurements y_k could come from economic indicators. The state transition and measurement matrices would be derived from economic models.

3.2.3 Mathematical Prerequisites

Understanding the Kalman filter and its derivations requires familiarity with several mathematical concepts. Below are some key definitions and theorems:

Gaussian Distributions

A random variable x is said to be Gaussian distributed with mean μ and covariance Σ, denoted as $x \sim \mathcal{N}(\mu, \Sigma)$, if its probability density function is given by:

$$f(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right). \tag{5}$$

Bayesian Inference

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability distribution of a random variable as more information becomes available. For a given set of measurements y, the posterior distribution of the state x is given by:

$$f(x|y) = \frac{f(y|x)f(x)}{f(y)}, \tag{6}$$

where:

- $f(x|y)$ is the posterior distribution.
- $f(y|x)$ is the likelihood.

- $f(x)$ is the prior distribution.
- $f(y)$ is the marginal likelihood.

Linear Algebra and Matrix Calculations

Several linear algebra concepts are essential for understanding the Kalman filter, including:

- **Matrix Multiplication**: The product of two matrices A and B is denoted by AB and is defined if the number of columns in A matches the number of rows in B.
- **Transpose**: The transpose of a matrix A, denoted A^T, is obtained by swapping its rows and columns.
- **Inverse**: The inverse of a square matrix A, denoted A^{-1}, satisfies $AA^{-1} = A^{-1}A = I$, where I is the identity matrix.
- **Covariance Matrices**: A covariance matrix Σ is a symmetric, positive semi-definite matrix that describes the covariance between elements of a random vector.

3.2.4 Conclusion

By understanding the problem setup and the essential mathematical prerequisites, a strong foundation is laid for comprehending the Kalman filter. The Kalman filter leverages the state space model and Bayesian inference to provide accurate state estimates from noisy measurements. In the subsequent section, this thesis will delve deeper into the Kalman filter's operations, including the prediction and update steps, and explore its derivations and applications.

4 Kalman Filter (KF)

The Kalman Filter is a powerful algorithm for estimating the state of a dynamic system from noisy measurements. Originally developed for aerospace engineering applications, it has since found widespread use in various fields such as robotics, finance, and signal processing.

To understand the Kalman Filter, consider the problem of tracking the position and velocity of a car over time using GPS measurements. The goal is to estimate the position of the car over time given noisy measurements of its position.

4.1 State Variables

Define the state and measurement variables as follows:

- **State vector:** $\mathbf{x}_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$, where x_k is the position and v_k is the velocity at time k.
- **Measurement vector:** $\mathbf{z}_k = z_k$, where z_k is the measured position at time k.

4.2 Model Parameters

The system is described using the following parameters:

- **State transition matrix:** $\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, where Δt is the time step duration.
- **Control input matrix:** $\mathbf{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ (no control input in this example).
- **Measurement matrix:** $\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.
- **Process noise covariance:** $\mathbf{Q} = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$, where σ_w^2 and σ_v^2 are the variances of the process noise in position and velocity, respectively.
- **Measurement noise covariance:** $\mathbf{R} = \sigma_z^2$, where σ_z^2 is the variance of the measurement noise.
- **Initial state estimate:** $\hat{\mathbf{x}}_0 = \begin{bmatrix} \hat{x}_0 \\ \hat{v}_0 \end{bmatrix}$.
- **Initial estimate covariance:** $\mathbf{P}_0 = \begin{bmatrix} \sigma_{x_0}^2 & 0 \\ 0 & \sigma_{v_0}^2 \end{bmatrix}$, where $\sigma_{x_0}^2$ and $\sigma_{v_0}^2$ are the variances of the initial state estimates.

4.3 The Kalman Filter's algorithm

The objective is to estimate the car's position and velocity over time, given a sequence of noisy GPS measurements. The Kalman Filter accomplishes this through a two-step process: prediction and update.

4.3.1 Predict Step

In the predict step, the process model is used to forecast the state of the system at the next time step. Given the current state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and the process model parameters, the predicted state is:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \tag{7}$$

Since there is no control input in this example, $\mathbf{B}_k \mathbf{u}_k = 0$.

Also predict the error covariance matrix to account for process noise:

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k \tag{8}$$

4.3.2 Update Step

In the update step, the actual measurement $\mathbf{z}_k$ is used to refine the state estimate. The innovation or residual, representing the discrepancy between the predicted and actual measurements, is computed as:

$$\mathbf{e}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{9}$$

Next, compute the Kalman gain $\mathbf{K}_k$, which determines the weighting between the predicted state and the measurement:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{10}$$

Using the Kalman gain, update the state estimate:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{11}$$

Finally, update the error covariance matrix to reflect the reduction in uncertainty due to the new measurement:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{12}$$

These equations constitute the core of the Kalman Filter algorithm. The filter iteratively performs the predict and update steps, continuously refining the state estimate based on incoming measurements.

4.4 Complete Mathematical Formulation

The Kalman filter can be derived using a Bayesian approach, which assumes that the state of the system is a random variable with a prior distribution. The measurement is also a random variable, and it is related to the state by a linear equation. The goal is to update the estimate of the state using Bayes' rule, which states that the posterior distribution of the state is proportional to the product of the prior distribution and the likelihood function of the measurement. In the case of the Kalman filter, the likelihood function is Gaussian, and the prior and posterior distributions are also Gaussian.

Theorem 1 (Prediction Step of the Kalman Filter)**.** *Given the state transition model*

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \mathbf{w}_k, \tag{13}$$

where F is the state transition matrix and $\mathbf{w}_k$ is the process noise, which is Gaussian with zero mean and covariance Q, the predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $P_{k|k-1}$ are given by:

$$\hat{\mathbf{x}}_{k|k-1} = F\hat{\mathbf{x}}_{k-1|k-1}, \tag{14}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q. \tag{15}$$

Proof. The proof follows directly from the state transition model. Given the prior state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and its covariance $P_{k-1|k-1}$, the state prediction is obtained by applying the state transition model. The process noise $\mathbf{w}_k$ introduces additional uncertainty, represented by its covariance matrix Q. Thus, the predicted state estimate is:

$$\hat{\mathbf{x}}_{k|k-1} = F\hat{\mathbf{x}}_{k-1|k-1}.$$

The predicted covariance is the sum of the transformed prior covariance and the process noise covariance:

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q.$$

□

Theorem 2 (Update Step of the Kalman Filter)**.** *Given the measurement model*

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k, \tag{16}$$

where H is the measurement matrix and $\mathbf{v}_k$ is the measurement noise, which is Gaussian with zero mean and covariance R, the updated state estimate $\hat{\mathbf{x}}_{k|k}$ and the updated covariance $P_{k|k}$ are given by:

$$P_{k|k} = (H^T R^{-1} H + P_{k|k-1}^{-1})^{-1}, \tag{17}$$

$$\hat{\mathbf{x}}_{k|k} = P_{k|k}(H^T R^{-1}\mathbf{z}_k + P_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1}). \tag{18}$$

Proof. Begin with Bayes' theorem:

$$f(\mathbf{x}|\mathbf{z}) = \frac{f(\mathbf{z}|\mathbf{x})f(\mathbf{x})}{f(\mathbf{z})}, \tag{19}$$

where $\mathbf{x}$ is the state, $\mathbf{z}$ are the measurements, $f(\mathbf{x}|\mathbf{z})$ is the posterior distribution, $f(\mathbf{z}|\mathbf{x})$ is the likelihood, $f(\mathbf{x})$ is the prior, and $f(\mathbf{z})$ is the marginal likelihood.

The Kalman filter assumes Gaussian distributions for the state and measurement noise:

$$f(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; H\mathbf{x}, R), \tag{20}$$

$$f(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}). \tag{21}$$

Substituting these into Bayes' theorem:

$$f(\mathbf{x}|\mathbf{z}) = \frac{\mathcal{N}(\mathbf{z}; H\mathbf{x}, R)\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1})}{f(\mathbf{z})}, \tag{22}$$

where $f(\mathbf{z})$ is the marginal likelihood of the measurements, which can be computed using the law of total probability:

$$f(\mathbf{z}) = \int f(\mathbf{z}|\mathbf{x})f(\mathbf{x})d\mathbf{x}. \tag{23}$$

Substituting the likelihood and prior, it follows that:

$$f(\mathbf{z}) = \int \mathcal{N}(\mathbf{z}; H\mathbf{x}, R)\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1})d\mathbf{x}. \tag{24}$$

Expanding the Gaussian distributions:

$$\begin{aligned} f(\mathbf{z}) = {} & \frac{1}{\sqrt{(2\pi)^n|R|}} \int \exp\left[-\frac{1}{2}(\mathbf{z} - H\mathbf{x})^T R^{-1}(\mathbf{z} - H\mathbf{x})\right] \\ & \frac{1}{\sqrt{(2\pi)^n|P_{k|k-1}|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})^T P_{k|k-1}^{-1}(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})\right] d\mathbf{x}. \end{aligned} \tag{25}$$

Simplifying the exponent, it follows that:

$$\begin{aligned} f(\mathbf{z}) = {} & \frac{1}{\sqrt{(2\pi)^n|R|}} \int \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{k-1|k})^T R^{-1}(\mathbf{x} - \hat{\mathbf{x}}_{k-1|k})\right] \\ & \exp\left[-\frac{1}{2}(\mathbf{z} - H\mathbf{x})^T Q^{-1}(\mathbf{z} - H\mathbf{x})\right] d\mathbf{x}. \end{aligned} \tag{26}$$

Expanding the exponent:

$$\begin{aligned} & -\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{k-1|k})^T R^{-1}(\mathbf{x} - \hat{\mathbf{x}}_{k-1|k}) - \frac{1}{2}(\mathbf{z} - H\mathbf{x})^T Q^{-1}(\mathbf{z} - H\mathbf{x}) \\ & = -\frac{1}{2}\mathbf{x}^T R^{-1}\mathbf{x} + \frac{1}{2}\hat{\mathbf{x}}_{k-1|k}^T R^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T R^{-1}\hat{\mathbf{x}}_{k-1|k} - \frac{1}{2}\hat{\mathbf{x}}_{k-1|k}^T R^{-1}\hat{\mathbf{x}}_{k-1|k} \\ & \quad - \frac{1}{2}\mathbf{z}^T Q^{-1}\mathbf{z} + \frac{1}{2}\mathbf{z}^T Q^{-1}H\mathbf{x} + \frac{1}{2}H\mathbf{x}^T Q^{-1}\mathbf{z} - \frac{1}{2}H\mathbf{x}^T Q^{-1}H\mathbf{x}. \end{aligned} \tag{27}$$

Complete the square for the quadratic terms involving $\mathbf{x}$:

$$
\begin{aligned}
-\frac{1}{2}\mathbf{x}^T R^{-1}\mathbf{x} + \frac{1}{2}\hat{\mathbf{x}}_{k-1|k}^T R^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T R^{-1}\hat{\mathbf{x}}_{k-1|k} - \frac{1}{2}H\mathbf{x}^T Q^{-1}H\mathbf{x} \\
+ \frac{1}{2}\mathbf{z}^T Q^{-1}H\mathbf{x} + \frac{1}{2}H\mathbf{x}^T Q^{-1}\mathbf{z} - \frac{1}{2}H\mathbf{x}^T Q^{-1}H\mathbf{x} \\
= -\frac{1}{2}(\mathbf{x} - K\mathbf{z})^T (P_{k|k-1}^{-1} + H^T R^{-1}H)(\mathbf{x} - K\mathbf{z}),
\end{aligned} \tag{28}
$$

where K is the Kalman gain given by:

$$
K = P_{k|k-1}H^T (HP_{k|k-1}H^T + R)^{-1}. \tag{29}
$$

Thus, the posterior distribution is Gaussian with mean and covariance:

$$
P_{k|k} = (H^T R^{-1}H + P_{k|k-1}^{-1})^{-1}, \tag{30}
$$

$$
\hat{\mathbf{x}}_{k|k} = P_{k|k}(H^T R^{-1}\mathbf{z}_k + P_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1}). \tag{31}
$$

□

Thus, combining the prediction and update steps, the full Kalman filter algorithm is derived.

4.5 Fusing Two Estimates

In some cases, it may be necessary to combine two estimates of the state that are obtained from different sources. For example, in multi-sensor systems, each sensor may provide an estimate of the state with its own error covariance matrix. The task is to combine these estimates into a single estimate with a smaller error covariance matrix. This can be done using the fusing two estimates method, which is a non-Bayesian approach that does not assume that the state or the measurements are Gaussian.

Suppose there are two estimates of the state, $\hat{x}_1$ and $\hat{x}_2$, with associated error covariance matrices P_1 and P_2, respectively. The goal is to combine these two estimates into a single estimate with a smaller error covariance matrix.

Definition 1 (Mahalanobis Distance)**.** *The Mahalanobis distance between two estimates $\hat{x}_1$ and $\hat{x}_2$ is defined as:*

$$
D^2(\hat{x}_1, \hat{x}_2) = (\hat{x}_1 - \hat{x}_2)^T (P_1 + P_2)^{-1}(\hat{x}_1 - \hat{x}_2). \tag{32}
$$

Definition 2 (Weights)**.** *The weights w_1 and w_2 are computed as:*

$$
w_1 = \frac{P_2}{P_1 + P_2}\exp\left(-\frac{1}{2}D^2(\hat{x}_1, \hat{x}_2)\right), \tag{33}
$$

$$
w_2 = \frac{P_1}{P_1 + P_2}\exp\left(-\frac{1}{2}D^2(\hat{x}_1, \hat{x}_2)\right). \tag{34}
$$

Theorem 3. *Given two estimates $\hat{x}_1$ and $\hat{x}_2$ with error covariance matrices P_1 and P_2, the fused estimate $\hat{x}_f$ and its error covariance matrix P_f are given by:*

$$\hat{x}_f = w_1\hat{x}_1 + w_2\hat{x}_2, \tag{35}$$

$$P_f = w_1(P_1 + (\hat{x}_1 - \hat{x}_f)(\hat{x}_1 - \hat{x}_f)^T) + w_2(P_2 + (\hat{x}_2 - \hat{x}_f)(\hat{x}_2 - \hat{x}_f)^T), \tag{36}$$

where w_1 and w_2 are the weights defined above.

Proof. First, we establish the weights based on the Mahalanobis distance. The Mahalanobis distance accounts for the relative error covariances of the estimates, ensuring that the fused estimate minimizes the combined error covariance.

To find the fused estimate $\hat{x}_f$, we compute the weighted average:

$$\hat{x}_f = w_1\hat{x}_1 + w_2\hat{x}_2,$$

where the weights w_1 and w_2 are chosen such that they minimize the error covariance of the fused estimate.

Next, we calculate the error covariance matrix P_f. The error covariance of the fused estimate takes into account the individual error covariances P_1 and P_2, as well as the deviation of the individual estimates from the fused estimate:

$$P_f = w_1(P_1 + (\hat{x}_1 - \hat{x}_f)(\hat{x}_1 - \hat{x}_f)^T) + w_2(P_2 + (\hat{x}_2 - \hat{x}_f)(\hat{x}_2 - \hat{x}_f)^T).$$

This accounts for the contribution of each estimate to the total error covariance, ensuring that the fused estimate has a smaller error covariance than either individual estimate. □

The intuition behind these weights is that if the Mahalanobis distance between the two estimates is small relative to their error covariance matrices, then the weights will be close to 0.5, indicating that the two estimates are equally reliable. If the Mahalanobis distance is large relative to the error covariance matrices, then one estimate will be given more weight than the other, depending on which one is more reliable.

The fusing two estimates method can be viewed as a simplified version of the Kalman filter, where the estimate of the state is obtained by fusing two independent estimates. In this method, there is no explicit modeling of the system dynamics or the measurement equations, and the weights are computed based on the error covariance matrices and the Mahalanobis distance between the estimates.

Theorem 4. *If the error distributions of the estimates $\hat{x}_1$ and $\hat{x}_2$ are assumed to be Gaussian, and the Mahalanobis distance between the estimates is small (which is not always the case), then the fusing two estimates method is equivalent to the Kalman filter.*

Proof. Consider the update step of the Kalman filter. Recall that the update step involves computing the Kalman gain K_k based on the predicted state estimate $\hat{x}_{k|k-1}$ and predicted error covariance matrix $P_{k|k-1}$, and using it to update the state estimate and error covariance matrix based on the new measurement z_k:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1},$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k \hat{x}_{k|k-1}),$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}.$$

In the fusing two estimates method, the update step can be seen as a weighted average of the two estimates, where the weights are determined by the error covariance matrices and the Mahalanobis distance:

$$w_1 = \frac{P_2}{P_1 + P_2} \exp\left(-\frac{1}{2} D^2(\hat{x}_1, \hat{x}_2)\right),$$
$$w_2 = \frac{P_1}{P_1 + P_2} \exp\left(-\frac{1}{2} D^2(\hat{x}_1, \hat{x}_2)\right),$$
$$\hat{x}_f = w_1 \hat{x}_1 + w_2 \hat{x}_2,$$
$$P_f = w_1(P_1 + (\hat{x}_1 - \hat{x}_f)(\hat{x}_1 - \hat{x}_f)^T) + w_2(P_2 + (\hat{x}_2 - \hat{x}_f)(\hat{x}_2 - \hat{x}_f)^T).$$

Comparing these equations to the update step of the Kalman filter, one can see that the weights w_1 and w_2 play the role of the Kalman gain K_k, and the fused estimate $\hat{x}_f$ corresponds to the updated state estimate $\hat{x}_{k|k}$. Furthermore, the error covariance matrix P_f can be seen as a weighted combination of the predicted error covariance matrix $P_{k|k-1}$ and the measurement error covariance matrix R_k.

Thus, under Gaussian assumptions and small Mahalanobis distance, the fusing two estimates method and the Kalman filter are equivalent. □

The predict step of the fusing two estimates method can be seen as a simple weighted average of the two predicted state estimates:

$$\hat{x}_{k|k-1} = w_1 \hat{x}_{k-1|k-1} + w_2 \hat{x}_{k-1|k-1}. \tag{37}$$

Thus, the predict step of the fusing two estimates method is equivalent to the predict step of the Kalman filter, which simply propagates the state estimate forward in time based on the system dynamics.

In summary, if the error distributions of the estimates are assumed to be Gaussian and the Mahalanobis distance is small, the fusing two estimates method is equivalent to the Kalman filter. The weights used in the fusing two estimates method are similar to the Kalman gain matrix used in the Kalman filter. However, the fusing two estimates method does not require matrix inversions and multiplications, making it computationally less expensive than the Kalman filter.

The fusing two estimates method can be seen as a simplified version of the Kalman filter. The predict step in the fusing two estimates method corresponds to the propagation of the state estimate and error covariance matrix using the process model in the Kalman filter. The update step in the fusing two estimates method corresponds to the correction of the state estimate and error covariance matrix using the measurement and its associated error covariance matrix in the Kalman filter.

4.6 Comparison and Limitations

Both the Kalman filter and the fusing two estimates method are used for state estimation, but they have different assumptions and properties. The Kalman filter assumes that the state and the measurements are Gaussian and that the system dynamics are linear. It provides a probabilistic interpretation of the estimate, and it is optimal in the sense of minimizing the mean squared error. However, it requires the computation of matrix inversions and multiplications, which can be computationally expensive for large systems.

The fusing two estimates method does not assume that the state or the measurements are Gaussian and does not require the computation of matrix inversions and multiplications. It is computationally less expensive than the Kalman filter, but it does not provide a probabilistic interpretation of the estimate, and it is not optimal in the sense of minimizing the mean squared error. It is useful when there are multiple estimates of the state from different sources and the goal is to combine them into a single estimate.

In general, the choice between the Kalman filter and the fusing two estimates method depends on the properties of the system and the requirements of the application. If the system dynamics are linear and the state and measurements are Gaussian, the Kalman filter is the appropriate method to use. If the system dynamics are nonlinear or the state and measurements are non-Gaussian, other methods such as the extended Kalman filter or particle filters may be more appropriate. If the goal is to combine multiple estimates of the state, the fusing two estimates method may be useful.

In conclusion, the Kalman filter is a powerful tool for state estimation in linear systems with Gaussian noise. It provides a probabilistic interpretation of the estimate and is optimal in the sense of minimizing the mean squared error. However, it has limitations in its assumptions and can be computationally expensive for large systems. The fusing two estimates method is a useful alternative when there are multiple estimates of the state from different sources, but it does not provide a probabilistic interpretation of the estimate and is not optimal in the sense of minimizing the mean squared error.

In the following section the Extended Kalman Filter (EKF) will be explored, which will address some key limitations of the KF, specifically its inability to effectively estimate non-linear systems.

5 Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF), introduced and formalized by Rudolf Kalman in [2] and [3] is an algorithm that extends the Kalman filter to nonlinear systems. The basic idea of the EKF is to linearize the nonlinear system by taking a first-order Taylor expansion around the current estimate of the state. The linearized system can then be used with the Kalman filter equations to estimate the state.

5.1 EKF Algorithm

Consider a nonlinear system described by the following state-space model:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h(x_k, v_k) \end{aligned}$$

where x_k is the state of the system at time step k, u_k is the input, w_k is the process noise, y_k is the measurement, v_k is the measurement noise, and f and h are nonlinear functions. The EKF estimates the state x_k using the measurements $y_{1:k}$.

To apply the Kalman filter to this nonlinear system, the system needs to be linearized. The purpose of the Taylor expansion is to approximate the nonlinear functions f and h by their first-order linear approximations around the current estimate of the state. This linearization is necessary because the Kalman filter is designed for linear systems, and by linearizing, it can be applied to nonlinear systems. Linearizing around the current estimate helps in capturing the local behavior of the nonlinear functions, making the problem tractable with linear algebra. Without linearization, the nonlinearities would complicate the estimation process, making it computationally infeasible.

Write the first-order Taylor expansion of f around $\hat{x}_{k-1}$ as:

$$f(x_{k-1}, u_{k-1}, w_{k-1}) \approx f(\hat{x}_{k-1}, u_{k-1}, \mathbf{0}) + F_{k-1}(x_{k-1} - \hat{x}_{k-1})$$

where $\hat{x}_{k-1}$ is the estimate of the state at time step $k-1$, and F_{k-1} is the Jacobian matrix of f evaluated at $\hat{x}_{k-1}$. Similarly, the first-order Taylor expansion of h around $\hat{x}_k$ can be written as:

$$h(x_k, v_k) \approx h(\hat{x}_k, \mathbf{0}) + H_k(x_k - \hat{x}_k)$$

where $\hat{x}_k$ is the estimate of the state at time step k, and H_k is the Jacobian matrix of h evaluated at $\hat{x}_k$.

The assumption of replacing the process and measurement noise with zero-mean Gaussian random variables simplifies the mathematical treatment of the noise, enabling the use of the Kalman filter framework, which is optimal for linear systems with Gaussian noise. This assumption allows the derivation of recursive equations for the mean and covariance of the state estimate. Gaussian noise is used because it has desirable properties, such as being fully described by its mean and covariance, and because the Kalman filter is optimal for linear systems with Gaussian noise.

The prediction step is given by:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}, \mathbf{0})$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}$$

where $\hat{x}_{k|k-1}$ is the predicted state, $P_{k|k-1}$ is the predicted error covariance, and Q_{k-1} is the covariance of the process noise.

The update step is given by:

$$\tilde{y}_k = y_k - h(\hat{x}_{k|k-1}, \mathbf{0})$$
$$S_k = H_k P_{k|k-1} H_k^T + R_k$$
$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

where $\tilde{y}_k$ is the measurement residual, R_k is the covariance of the measurement noise, S_k is the innovation covariance, K_k is the Kalman gain, $\hat{x}_{k|k}$ is the updated state, and $P_{k|k}$ is the updated error covariance.

Each term in the update equation has a specific role:

- $\tilde{y}_k$ represents the difference between the actual measurement and the predicted measurement.

- S_k combines the predicted error covariance and the measurement noise covariance to determine the uncertainty in the innovation.

- K_k is the Kalman gain, which determines how much the predictions should be corrected based on the new measurements. The Kalman gain balances the uncertainty between the prediction and the measurement, ensuring that the update is weighted appropriately. A higher Kalman gain implies a greater reliance on the measurements, while a lower gain implies greater reliance on the model.

5.2 Mathematical Derivation of EKF

To derive the EKF mathematically, start with the nonlinear system described by:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}),$$
$$y_k = h(x_k, v_k).$$

The Kalman filter assumes that the system is linear and that the noise is Gaussian. Therefore, first linearize the nonlinear functions around the current estimate of the state.

Theorem 5 (Linearization of Nonlinear Functions)**.** *Consider the nonlinear functions f and h describing the system. The first-order Taylor expansion of f around $\hat{x}_{k-1}$ is given by:*

$$f(x_{k-1}, u_{k-1}, w_{k-1}) \approx f(\hat{x}_{k-1}, u_{k-1}, \mathbf{0}) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + \mathcal{O}((x_{k-1} - \hat{x}_{k-1})^2),$$

where F_{k-1} is the Jacobian matrix of f evaluated at $\hat{x}_{k-1}$. Similarly, the first-order Taylor expansion of h around $\hat{x}_k$ is:

$$h(x_k, v_k) \approx h(\hat{x}_k, \mathbf{0}) + H_k(x_k - \hat{x}_k) + \mathcal{O}((x_k - \hat{x}_k)^2),$$

where H_k is the Jacobian matrix of h evaluated at $\hat{x}_k$.

Proof. The first-order Taylor expansion of a multivariable function f around a point $\hat{x}$ is given by:

$$f(x) \approx f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) + \mathcal{O}((x - \hat{x})^2).$$

Applying this to $f(x_{k-1}, u_{k-1}, w_{k-1})$ around $\hat{x}_{k-1}$ and $h(x_k, v_k)$ around $\hat{x}_k$, we obtain the stated results. □

To justify why higher-order terms are not needed, consider that the first-order Taylor expansion provides a linear approximation which is sufficient if the nonlinearity is not too strong. The conditions needed for the linear approximation to hold include:

- The state estimates must be close to the true states.
- The nonlinear functions f and h must be differentiable.

Theorem 6 (Linearized System Equations)**.** *Substituting the Taylor expansions into the nonlinear system equations, we obtain:*

$$\begin{aligned} x_k &= f(\hat{x}_{k-1}, u_{k-1}, \mathbf{0}) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + w_{k-1}, \\ y_k &= h(\hat{x}_k, \mathbf{0}) + H_k(x_k - \hat{x}_k) + v_k. \end{aligned}$$

Proof. Substitute the Taylor expansions derived in Theorem 5 into the original nonlinear system equations. □

Theorem 7 (EKF Prediction and Update Steps)**.** *Applying the standard Kalman filter equations to the linearized system, the prediction step is given by:*

$$\begin{aligned} \hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}, u_{k-1}, \mathbf{0}), \\ P_{k|k-1} &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1}, \end{aligned}$$

and the update step is given by:

$$\begin{aligned} \tilde{y}_k &= y_k - h(\hat{x}_{k|k-1}, \mathbf{0}), \\ S_k &= H_k P_{k|k-1} H_k^T + R_k, \\ K_k &= P_{k|k-1} H_k^T S_k^{-1}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k, \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}. \end{aligned}$$

Proof. Given the linearized system:

$$\begin{aligned} x_k &= f(\hat{x}_{k-1}, u_{k-1}, \mathbf{0}) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + w_{k-1}, \\ y_k &= h(\hat{x}_k, \mathbf{0}) + H_k(x_k - \hat{x}_k) + v_k, \end{aligned}$$

the goal is to apply the Kalman filter to this system.

Prediction Step:

- Predict the state estimate:

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbb{E}[x_k|\mathcal{Y}_{k-1}] \\ &= \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1})|\mathcal{Y}_{k-1}] \\ &\approx f(\hat{x}_{k-1|k-1}, u_{k-1}, \mathbf{0}), \end{aligned}$$

  where $\mathcal{Y}_{k-1}$ denotes the information set available at time $k-1$.

- Predict the error covariance:

$$\begin{aligned} P_{k|k-1} &= \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T|\mathcal{Y}_{k-1}] \\ &= \mathbb{E}[(F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + w_{k-1})(F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + w_{k-1})^T|\mathcal{Y}_{k-1}] \\ &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1}, \end{aligned}$$

  assuming $\mathbb{E}[w_{k-1}w_{k-1}^T] = Q_{k-1}$ and w_{k-1} is uncorrelated with x_{k-1}.

Update Step:

- Compute the innovation (measurement residual):

$$\begin{aligned} \tilde{y}_k &= y_k - \mathbb{E}[y_k|\mathcal{Y}_{k-1}] \\ &= y_k - h(\hat{x}_{k|k-1}, \mathbf{0}). \end{aligned}$$

- Compute the innovation covariance:

$$\begin{aligned} S_k &= \mathbb{E}[\tilde{y}_k\tilde{y}_k^T|\mathcal{Y}_{k-1}] \\ &= \mathbb{E}[(H_k(x_k - \hat{x}_{k|k-1}) + v_k)(H_k(x_k - \hat{x}_{k|k-1}) + v_k)^T|\mathcal{Y}_{k-1}] \\ &= H_kP_{k|k-1}H_k^T + R_k, \end{aligned}$$

  assuming $\mathbb{E}[v_kv_k^T] = R_k$ and v_k is uncorrelated with x_k.

- Compute the Kalman gain:

$$K_k = P_{k|k-1}H_k^TS_k^{-1}.$$

- Update the state estimate:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k.$$

- Update the error covariance:

$$P_{k|k} = (I - K_kH_k)P_{k|k-1}.$$

Thus, the EKF prediction and update steps for the linearized system are rigorously derived. □

5.2.1 Key Assumptions

Assumption 1 (Gaussian Noise Models)**.** *The process and measurement noise are assumed to be Gaussian. This assumption is justified by the Central Limit Theorem, which states that the sum of a large number of independent and identically distributed random variables tends towards a Gaussian distribution, regardless of the original distribution of the variables. In practical systems, process and measurement noise can often be modeled as Gaussian due to the aggregation of multiple independent noise sources.*

Assumption 2 (Convergence Properties)**.** *The convergence of the EKF is more challenging to prove than for the linear Kalman filter due to the nonlinearities. However, under certain conditions, such as small nonlinearities and accurate initial state estimates, the EKF can converge to the true state. Specifically, the EKF converges if the system dynamics are sufficiently smooth (differentiable) and the linearization errors are small.*

5.3 Comparison to KF and Limitations of EKF

The Extended Kalman Filter (EKF) adapts the Kalman filter methodology to nonlinear systems by linearizing the system dynamics using first-order Taylor expansions. While the EKF approximates the true nonlinear system, its accuracy hinges on effective linearization and accurate modeling of noise.

1. **Advantages of KF:**
    - **Optimal Performance in Linear Systems:** The Kalman Filter excels in linear systems with Gaussian noise, delivering optimal state estimates without the need for approximation. It ensures the best possible estimate due to its direct application to linear models.
2. **Advantages of EKF:**
    - **Suitability for Nonlinear Systems:** EKF shines in systems where nonlinear dynamics can be adequately represented by a first-order Taylor expansion. Unlike the KF, which falters in nonlinear scenarios, the EKF provides more accurate state estimates by effectively capturing nonlinear behavior.
3. **Limitations of EKF:**
    - **Impact of Strong Nonlinearities:** When nonlinearities are pronounced, the EKF's linear approximation may introduce errors, leading to suboptimal state estimates. The Taylor expansion only accurately approximates the system locally.
    - **Computational Demands:** EKF requires the computation of Jacobian matrices at each step, which can be computationally intensive, particularly in high-dimensional systems.

- **Convergence Challenges:** Convergence of the EKF depends on accurate initial estimates and noise models. Inaccuracies can hinder convergence, impacting the filter's effectiveness.

In summary, while the EKF extends the Kalman Filter to nonlinear systems, offering enhanced applicability, it introduces complexities and limitations. Careful consideration of system characteristics and trade-offs between accuracy, computational complexity, and nonlinear effects guide the choice between KF and EKF in practical applications.

6 Unscented Kalman Filter (UKF)

The Unscented Kalman Filter (UKF) is a recursive estimator designed for nonlinear systems, using a deterministic sampling technique to better capture the true mean and covariance of the state distribution under nonlinear transformations. Unlike the Extended Kalman Filter (EKF), the UKF does not require the explicit calculation of Jacobians, making it more robust in handling nonlinearities.

The UKF is based on the Unscented Transformation (UT), which is a method for calculating the statistics of a random variable undergoing a nonlinear transformation. The core idea is to use a set of carefully chosen sample points, known as sigma points, to parameterize the mean and covariance of the state distribution. This thesis utilizes the UKF's formalization from [4].

6.1 UKF Algorithm

1. **Sigma Point Generation**:

   For an n-dimensional state vector $\mathbf{x}_k$, generate $2n+1$ sigma points $\mathbf{X}_k^{[i]}$:

$$\mathbf{X}_k^{[0]} = \hat{\mathbf{x}}_k, \tag{38}$$

$$\mathbf{X}_k^{[i]} = \hat{\mathbf{x}}_k + \sqrt{(n+\lambda)\mathbf{P}_k}_i, \quad i = 1, \ldots, n, \tag{39}$$

$$\mathbf{X}_k^{[i+n]} = \hat{\mathbf{x}}_k - \sqrt{(n+\lambda)\mathbf{P}_k}_i, \quad i = 1, \ldots, n, \tag{40}$$

   where $\sqrt{(n+\lambda)\mathbf{P}_k}_i$ denotes the i-th column of the matrix square root of $(n+\lambda)\mathbf{P}_k$, and $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter.

2. **Propagation of Sigma Points through the Process Model**:

$$\mathbf{X}_{k+1|k}^{[i]} = f(\mathbf{X}_k^{[i]}, \mathbf{u}_k), \quad i = 0, \ldots, 2n. \tag{41}$$

3. **Computation of Predicted Mean and Covariance**:

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n} W_i^{[m]} \mathbf{X}_{k+1|k}^{[i]}, \tag{42}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{2n} W_i^{[c]} \left(\mathbf{X}_{k+1|k}^{[i]} - \hat{\mathbf{x}}_{k+1|k}\right)\left(\mathbf{X}_{k+1|k}^{[i]} - \hat{\mathbf{x}}_{k+1|k}\right)^T + \mathbf{Q}_k, \tag{43}$$

   where $W_i^{[m]}$ and $W_i^{[c]}$ are weights for the mean and covariance, respectively.

4. **Propagation of Sigma Points through the Measurement Model**:

$$\mathbf{Y}_{k+1|k}^{[i]} = h(\mathbf{X}_{k+1|k}^{[i]}), \quad i = 0, \ldots, 2n. \tag{44}$$

5. **Computation of Predicted Measurement Mean and Covariance**:

$$\hat{\mathbf{y}}_{k+1} = \sum_{i=0}^{2n} W_i^{[m]} \mathbf{Y}_{k+1|k}^{[i]}, \tag{45}$$

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} W_i^{[c]} \left(\mathbf{Y}_{k+1|k}^{[i]} - \hat{\mathbf{y}}_{k+1}\right) \left(\mathbf{Y}_{k+1|k}^{[i]} - \hat{\mathbf{y}}_{k+1}\right)^T + \mathbf{R}_k, \tag{46}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} W_i^{[c]} \left(\mathbf{X}_{k+1|k}^{[i]} - \hat{\mathbf{x}}_{k+1|k}\right) \left(\mathbf{Y}_{k+1|k}^{[i]} - \hat{\mathbf{y}}_{k+1}\right)^T. \tag{47}$$

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

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}, \tag{48}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_k (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}), \tag{49}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_k \mathbf{P}_{yy} \mathbf{K}_k^T. \tag{50}$$

6.1.1 Sigma Point Selection and Unscented Transformation

The choice of sigma points is crucial to the accuracy of the UKF. Sigma points are chosen to capture the mean and covariance of the underlying distribution accurately up to the second order. The selection is as follows:

$$\mathbf{X}^{[0]} = \hat{\mathbf{x}}, \tag{51}$$

$$\mathbf{X}^{[i]} = \hat{\mathbf{x}} + \sqrt{(n+\lambda)\mathbf{P}}_i, \quad i = 1, \ldots, n, \tag{52}$$

$$\mathbf{X}^{[i+n]} = \hat{\mathbf{x}} - \sqrt{(n+\lambda)\mathbf{P}}_i, \quad i = 1, \ldots, n, \tag{53}$$

Common values for parameters are $\alpha = 10^{-3}$, $\kappa = 0$, and $\beta = 2$ for Gaussian distributions.

6.1.2 Higher-Order Moments and Advanced Sigma Point Selection

1. Incorporating Higher-Order Moments: To capture skewness and kurtosis, use higher-order sigma points.

$$\mathbf{X}_k^{[i]} = \hat{\mathbf{x}}_k + \eta_i \sqrt{(n+\lambda)\mathbf{P}_k} + \gamma_i (\text{higher-order terms}), \quad i = 1, \ldots, n, \tag{54}$$

6.2 Proofs of Mean and Covariance Propagation

The Unscented Transformation (UT) is a method to approximate the mean and covariance of a random variable $\mathbf{y} = g(\mathbf{x})$, where $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$. This section will present the detailed derivation of the mean and covariance propagation using the UT.

Definition 3. *The Unscented Transformation (UT) generates sigma points and weights to approximate the mean and covariance of a transformed random variable. Given $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$, the steps are:*

1. ***Generate Sigma Points:***

$$
\begin{aligned}
\mathbf{X}^{[0]} &= \hat{\mathbf{x}}, \\
\mathbf{X}^{[i]} &= \hat{\mathbf{x}} + \sqrt{(n+\lambda)\mathbf{P}_x}_i, \quad i = 1, \ldots, n, \\
\mathbf{X}^{[i+n]} &= \hat{\mathbf{x}} - \sqrt{(n+\lambda)\mathbf{P}_x}_i, \quad i = 1, \ldots, n,
\end{aligned}
$$

where $\sqrt{(n+\lambda)\mathbf{P}_x}_i$ denotes the i-th column of the matrix square root of $(n+\lambda)\mathbf{P}_x$, and $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter.

2. ***Propagate Sigma Points through Nonlinear Function:***

$$
\mathbf{Y}^{[i]} = g(\mathbf{X}^{[i]}), \quad i = 0, \ldots, 2n.
$$

3. ***Calculate Mean:***

$$
\hat{\mathbf{y}} = \sum_{i=0}^{2n} W_i^{[m]} \mathbf{Y}^{[i]},
$$

where $W_i^{[m]}$ are weights for the mean.

4. ***Calculate Covariance:***

$$
\mathbf{P}_y = \sum_{i=0}^{2n} W_i^{[c]} (\mathbf{Y}^{[i]} - \hat{\mathbf{y}})(\mathbf{Y}^{[i]} - \hat{\mathbf{y}})^\top,
$$

where $W_i^{[c]}$ are weights for the covariance.

6.2.1 Mean Propagation

Theorem 8. *Given $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$ and a nonlinear function $\mathbf{y} = g(\mathbf{x})$, the mean $\hat{\mathbf{y}} = \mathbb{E}[\mathbf{y}]$ can be approximated using the Unscented Transformation as:*

$$
\hat{\mathbf{y}} \approx \sum_{i=0}^{2n} W_i^{[m]} \mathbf{Y}^{[i]},
$$

where $W_i^{[m]}$ are the weights for the mean.

Proof. The mean of $\mathbf{y}$ is defined as:

$$\hat{\mathbf{y}} = \mathbb{E}[\mathbf{y}] = \mathbb{E}[g(\mathbf{x})].$$

Given $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$, we use the sigma points $\mathbf{X}^{[i]}$ to approximate the expectation:

$$\hat{\mathbf{y}} \approx \sum_{i=0}^{2n} W_i^{[m]} g(\mathbf{X}^{[i]}).$$

The weights $W_i^{[m]}$ are defined as:

$$W_i^{[m]} = \begin{cases} \frac{\lambda}{n+\lambda}, & \text{for } i = 0, \\ \frac{1}{2(n+\lambda)}, & \text{for } i = 1, \dots, 2n. \end{cases}$$

These weights ensure that the approximation retains the properties of the mean and covariance of the Gaussian distribution. □

6.2.2 Covariance Propagation

Theorem 9. *Given $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$ and a nonlinear function $\mathbf{y} = g(\mathbf{x})$, the covariance $\mathbf{P}_y$ can be approximated using the Unscented Transformation as:*

$$\mathbf{P}_y \approx \sum_{i=0}^{2n} W_i^{[c]} (\mathbf{Y}^{[i]} - \hat{\mathbf{y}})(\mathbf{Y}^{[i]} - \hat{\mathbf{y}})^\top,$$

where $W_i^{[c]}$ are the weights for the covariance.

Proof. The covariance of $\mathbf{y}$ is defined as:

$$\mathbf{P}_y = \mathbb{E}[(\mathbf{y} - \hat{\mathbf{y}})(\mathbf{y} - \hat{\mathbf{y}})^\top] = \mathbb{E}[(g(\mathbf{x}) - \hat{\mathbf{y}})(g(\mathbf{x}) - \hat{\mathbf{y}})^\top].$$

Given $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_x)$, we use the sigma points $\mathbf{X}^{[i]}$ and their transformations $\mathbf{Y}^{[i]}$ to approximate the expectation:

$$\mathbf{P}_y \approx \sum_{i=0}^{2n} W_i^{[c]} (\mathbf{Y}^{[i]} - \hat{\mathbf{y}})(\mathbf{Y}^{[i]} - \hat{\mathbf{y}})^\top.$$

The weights $W_i^{[c]}$ are defined as:

$$W_i^{[c]} = \begin{cases} \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), & \text{for } i = 0, \\ \frac{1}{2(n+\lambda)}, & \text{for } i = 1, \dots, 2n. \end{cases}$$

These weights are chosen to ensure that the approximation maintains the properties of the covariance of the Gaussian distribution, including the incorporation of higher-order moments through the terms $(1 - \alpha^2 + \beta)$. □

6.3 Comparison to EKF and Limitations of UKF

The UKF differs from the EKF in handling nonlinearities. The EKF linearizes the nonlinear functions using Jacobians, while the UKF uses sigma points to capture the mean and covariance directly.

1. **EKF: Linearization via Jacobians**

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k, \mathbf{u}_k), \tag{55}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k, \tag{56}$$

where $\mathbf{F}_k$ is the Jacobian of $f(\cdot)$.

$$\mathbf{K}_k = \mathbf{P}_{k+1|k} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k+1|k} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \tag{57}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_k (\mathbf{y}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k})), \tag{58}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k+1|k}, \tag{59}$$

Compared to the EKF, the UKF does not require the computation of the Jacobian of the nonlinear functions, which can be a difficult and time-consuming task. Instead, it approximates the mean and covariance of the state distribution using a set of sigma points, which captures the nonlinearities in a more accurate way.

However, the UKF can suffer from two main limitations: (1) the choice of the scaling parameter λ can affect the performance of the filter, and (2) the propagation of the sigma points through the nonlinear functions can introduce numerical errors, especially if the functions are highly nonlinear or if the spread of the sigma points is not properly chosen.

In summary, the EKF and the UKF are two widely used nonlinear filtering techniques that can be used to estimate the state of a dynamic system. The EKF is based on the linearization of the nonlinear functions around the current estimate of the state, while the UKF approximates the mean and covariance of the state distribution using a set of sigma points. Both filters have their own strengths and weaknesses, and the choice between them depends on the specific requirements of the application at hand.

7 Performance Comparison of KF and EKF: A Case Study Using KITTI Dataset

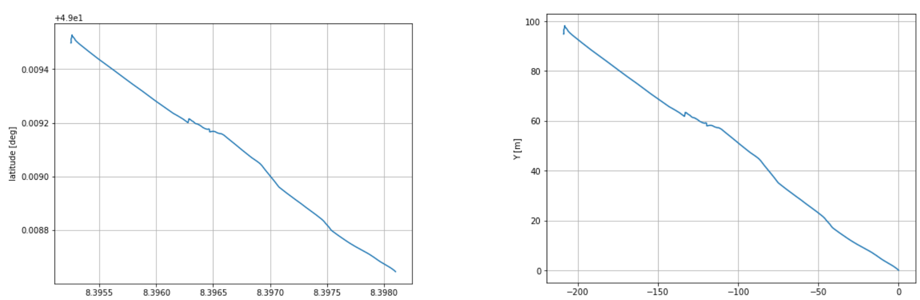
The KITTI dataset [1] is a benchmark for evaluating computer vision algorithms, particularly those related to autonomous driving. It provides a diverse range of sensor data, including high-resolution images, GPS, IMU measurements, and ground truth annotations. For this study, the raw GPS and IMU data from drive 71 on 29-09-2011 of the KITTI dataset is leveraged. The GPS and IMU data are essential for estimating vehicle trajectories and dynamic parameters using state estimation methods like the Kalman Filter (KF) and Extended Kalman Filter (EKF).

7.1 Data Setup

This section describes the setup of ground truth and observed data from the KITTI dataset, which forms the basis for the state estimation simulations. First, the ground truth data will be discussed, which provides the benchmark for evaluating the performance of these algorithms, followed by the observed data that will be fed into the KF and EKF models.

7.1.1 Ground Truth Data

Figure 1 presents the ground truth trajectories derived from the KITTI dataset. These trajectories serve as the baseline against which the performance of the state estimation algorithms is evaluated.



(a) Longitude-Latitude ground truth trajectory of the car

(b) Ground truth trajectory of the car converted to Cartesian coordinates

Figure 1: Ground truth trajectories

- **Figure 1a:** This plot shows the ground truth trajectory of the vehicle in longitude-latitude coordinates. It provides a precise path of the vehicle's movement over time, crucial for evaluating the accuracy of position estimation.

- **Figure 1b:** Here, the ground truth trajectory is converted to Cartesian coordinates. This transformation facilitates easier integration with state estimation algorithms like KF and EKF, which typically operate in Cartesian space.

In addition to the trajectory data, the ground truth yaw rates, yaw angles, and forward velocities are also extracted from the KITTI dataset. These dynamic parameters are essential for the EKF, which can handle non-linear data more effectively than the KF.

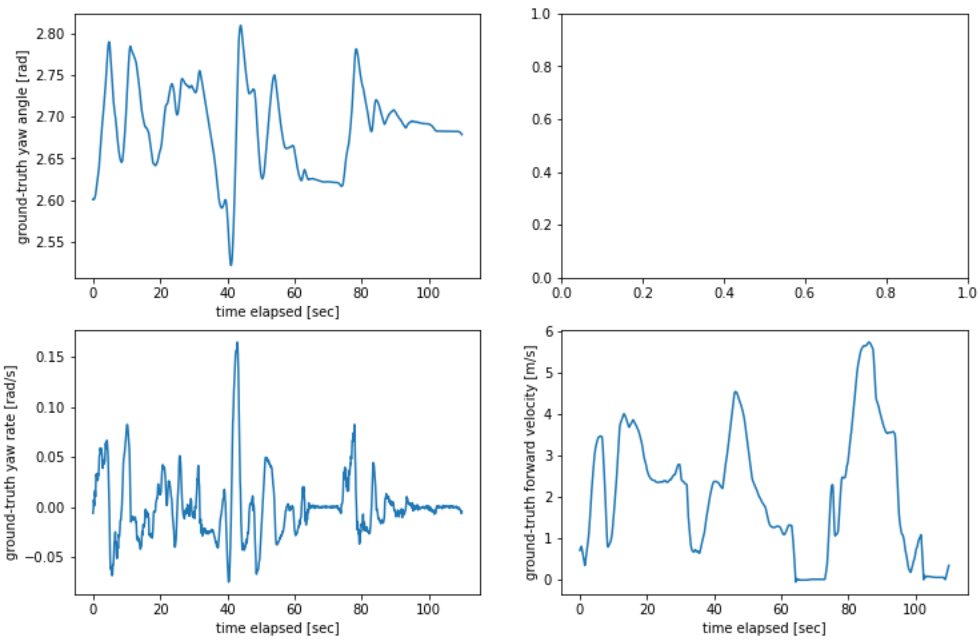
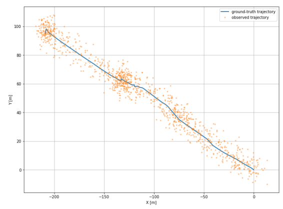


Figure 2: Ground truth yaw rates, yaw angles, forward velocities

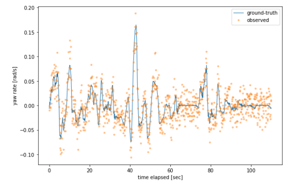
Figure 2 presents the ground truth yaw angles, yaw rates, and forward velocities. This data is crucial for the EKF simulation as it incorporates non-linear dynamics in its state estimation process. The KF, on the other hand, will assume constant yaw rates, yaw angles, and forward velocities and will focus primarily on estimating the vehicle's trajectory.

7.1.2 Observed Data

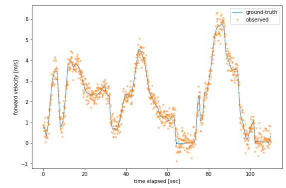
The observed data is derived by adding noise to the ground truth data, simulating real-world sensor measurements. This data will be used as inputs to the KF and EKF models to test their performance in state estimation.



(a) Observed trajectory



(b) Observed yaw rates



(c) Observed forward velocity

Figure 3: Observed data

- **Figure 3a:** This plot shows the observed trajectory, which includes noise added to the ground truth positions. This simulates the real-world GPS data that the KF and EKF will process.
- **Figure 3b:** The observed yaw rates, which are noisy measurements of the ground truth yaw rates. These will be particularly important for the EKF.
- **Figure 3c:** The observed forward velocity, also derived by adding noise to the ground truth forward velocities. This will be used to simulate the vehicle's motion in both the KF and EKF.

The observed data provides a realistic input for testing the state estimation algorithms. By comparing the estimated states from the KF and EKF with the ground truth data, one can evaluate their performance in handling real-world noise and non-linear dynamics.

7.2 KF results

The Kalman Filter (KF) is applied to estimate the trajectory of a moving object using the KITTI dataset. The KF operates in two main steps: prediction and update. In the prediction step, the filter predicts the state of the system based on its previous state and a dynamic model. In the update step, it corrects this prediction using noisy measurements. For the KITTI dataset, the state vector includes position and velocity information, and the measurements come from the GPS sensor. The non-linear IMU data for the yaw rates, yaw angles and forward velocities was unable to be effectively integrated, and so those were made constant, leaving the Kalman Filter to estimate solely on the x and y position of the car.

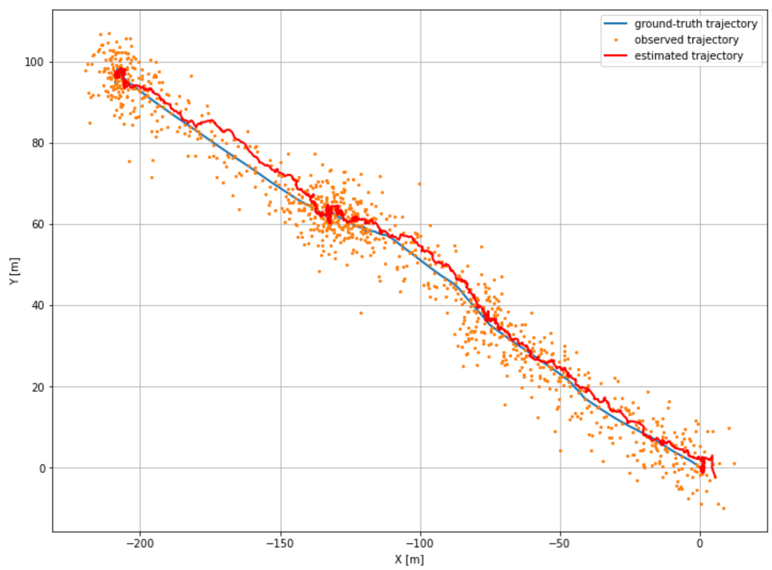
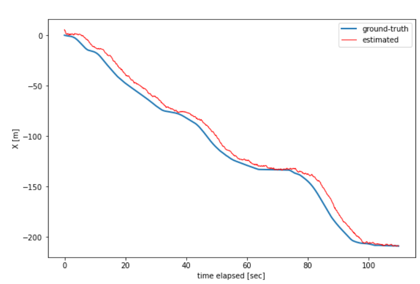
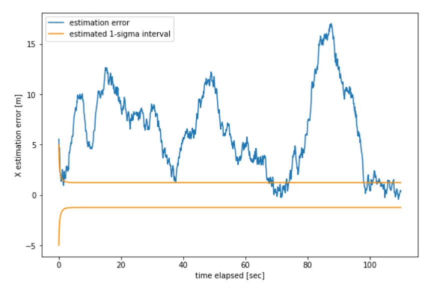


Figure 4: KF estimated trajectory

Figure 4 shows the estimated trajectory of the object obtained from the Kalman Filter. The trajectory demonstrates how the filter tracks the object's position over time, incorporating both prediction and measurement updates.



(a) KF estimated x-trajectory

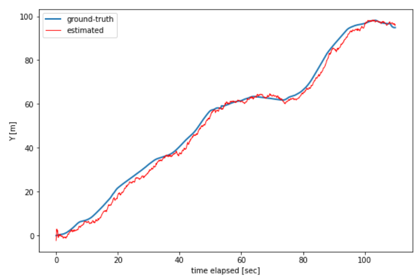


(b) KF x-trajectory error

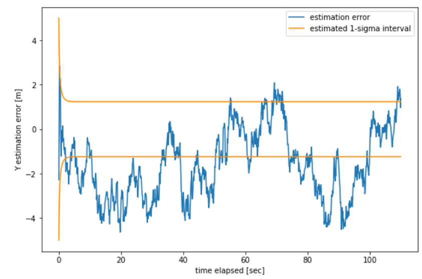
Figure 5: KF x-trajectory performance

Figure 5 presents the performance of the Kalman Filter in estimating the X-coordinate of the object. Subfigure 5a shows the estimated X-trajectory, demonstrating how accurately the filter tracks the X-position. Subfigure 5b depicts the error in the X-trajectory estimation, highlighting deviations between the estimated and ground truth X-coordinates.

Figure 6 illustrates the performance of the Kalman Filter in estimating the Y-coordinate of the object. Subfigure 6a displays the estimated Y-trajectory, showing how well the filter tracks the Y-position. Subfigure 6b portrays the error



(a) KF estimated y-trajectory



(b) KF y-trajectory error

Figure 6: KF y-trajectory performance

in the Y-trajectory estimation, indicating discrepancies between the estimated and ground truth Y-coordinates.

These plots collectively demonstrate the effectiveness of the Kalman Filter in accurately estimating the trajectory of a moving object using the KITTI dataset, highlighting both the trajectory estimation and the associated errors.

7.3 EKF results

The Extended Kalman Filter (EKF) represents an advancement over the traditional Kalman Filter by accommodating nonlinearities in the system dynamics. Unlike the KF, which assumes linear state transitions and measurements, the EKF can handle nonlinearities by linearizing them around the current estimate. This capability allows integration of additional sensor data such as IMU measurements, including yaw rates, yaw angles, and forward velocities. By incorporating these IMU data, the EKF enhances the estimation process by modeling not only the position but also the orientation (yaw angle). It can also use the extra velocity dynamics information to better define and estimate the initial state. In the EKF, the state vector includes position (x and y), orientation (θ).

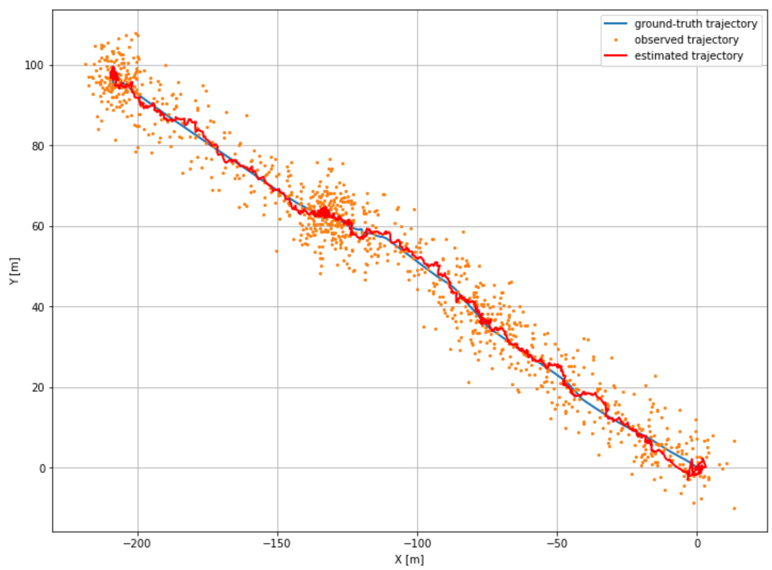
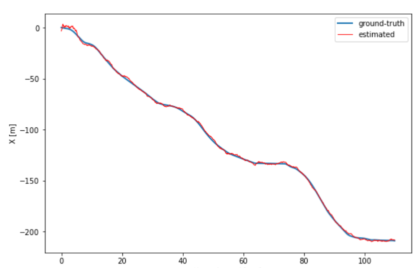
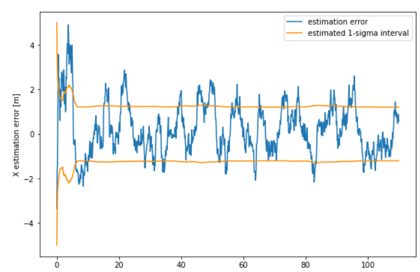


Figure 7: EKF estimated trajectory

Figure 7 shows the estimated trajectory obtained from the Extended Kalman Filter. This trajectory reflects the enhanced capability of the EKF to track both position and orientation (θ) of the moving object.



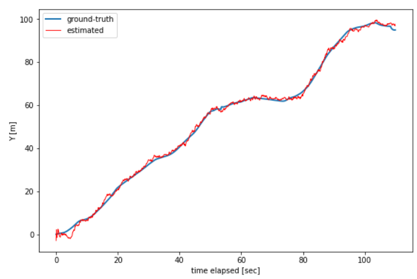
(a) EKF estimated x-trajectory



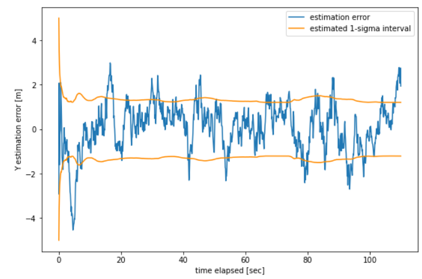
(b) EKF x-trajectory error

Figure 8: EKF x-trajectory performance

Figure 8 presents the performance of the Extended Kalman Filter in estimating the X-coordinate of the object. Subfigure 8a shows the estimated X-trajectory, demonstrating how accurately the filter tracks the X-position. Subfigure 8b depicts the error in the X-trajectory estimation, highlighting deviations between the estimated and ground truth X-coordinates.



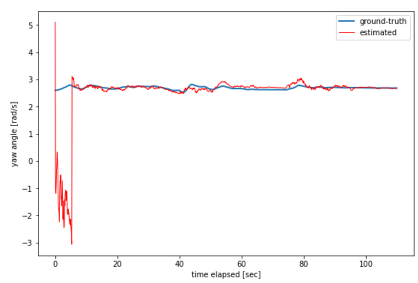
(a) EKF estimated y-trajectory



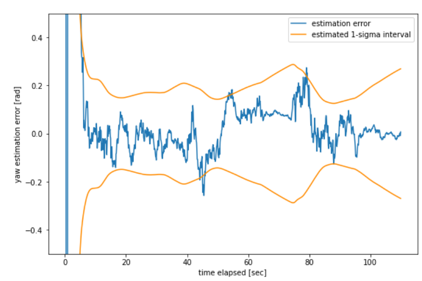
(b) EKF y-trajectory error

Figure 9: EKF y-trajectory performance

Figure 9 illustrates the performance of the Extended Kalman Filter in estimating the Y-coordinate of the object. Subfigure 9a displays the estimated Y-trajectory, showing how well the filter tracks the Y-position. Subfigure 9b portrays the error in the Y-trajectory estimation, indicating discrepancies between the estimated and ground truth Y-coordinates.



(a) EKF estimated yaw-angles



(b) EKF yaw error

Figure 10: EKF theta performance

Figure 10 depicts the performance of the Extended Kalman Filter in estimating the yaw angles (θ) of the object. Subfigure 10a shows the estimated yaw angles, demonstrating how well the filter tracks the orientation of the object. Subfigure 10b portrays the error in the yaw angle estimation, indicating discrepancies between the estimated and ground truth yaw angles.

These plots collectively demonstrate the effectiveness of the Extended Kalman Filter in accurately estimating the trajectory of a moving object using the KITTI dataset, showcasing the integration of IMU data to improve both position and orientation estimates.

7.4 Performance Comparison

In this section, the performance of the Kalman Filter (KF) and Extended Kalman Filter (EKF) is rigorously analyzed using NEES (Normalized Estimation Error Squared) and RMSE (Root Mean Square Error).

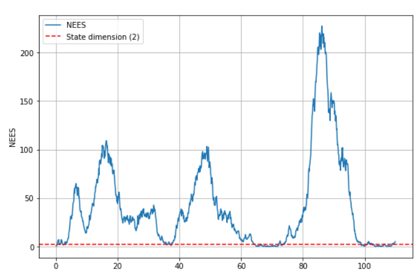
7.4.1 Normalized Estimation Error Squared (NEES)

NEES measures the consistency of a filter's covariance predictions relative to the actual measurement noise. For a state estimation $\hat{\mathbf{x}}$ with covariance $\mathbf{P}$, and true state $\mathbf{x}$, NEES is defined as:

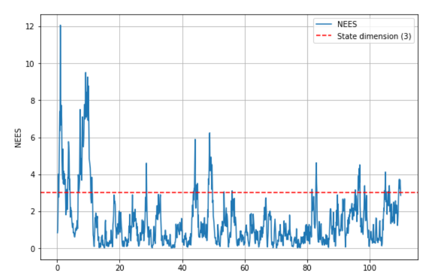
$$\text{NEES} = (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{P}^{-1} (\mathbf{x} - \hat{\mathbf{x}})$$

- **Mathematical Rigor**: NEES follows a chi-squared distribution with degrees of freedom equal to the state dimension n when the filter is correctly specified. This property makes NEES a robust measure of filter consistency.
- **Interpretation**: A NEES value close to the state dimension n suggests the filter is well-calibrated to the noise statistics. Values significantly higher than n indicate overconfidence (underestimation of noise), while values lower suggest underconfidence (overestimation of noise).
- **KF vs EKF Expectations**: In the simulation using KITTI data:
  - KF: Expected NEES around 2, reflecting its assumptions of linear state transitions and Gaussian noise.
  - EKF: Expected NEES around 3 due to its ability to handle nonlinearities, albeit with higher computational complexity.

The following figure 11 illustrates the NEES plots for KF and EKF:



(a) KF NEES



(b) EKF NEES

Figure 11: KF vs EKF NEES

In Figure 11a, KF's NEES plot may show periods of higher values, indicating potential issues with noise estimation, specifically the position's standard deviation which was used (in this case 5). For EKF in Figure 11b, NEES initially starts higher due to initial uncertainty propagation but stabilizes below the state dimension as the filter adapts to nonlinearities. When lowering the position's standard deviation, the KF's NEES immediately drops down. However unlike the EKF's NEES which seems to stabilize quickly, the volatility of the KF stays consistent no matter the initial standard deviation used. This showcases how the inability to integrate the necessary non-linear data of the KITTI drive causes the KF to remain volatile, even after several iterations of the filter.

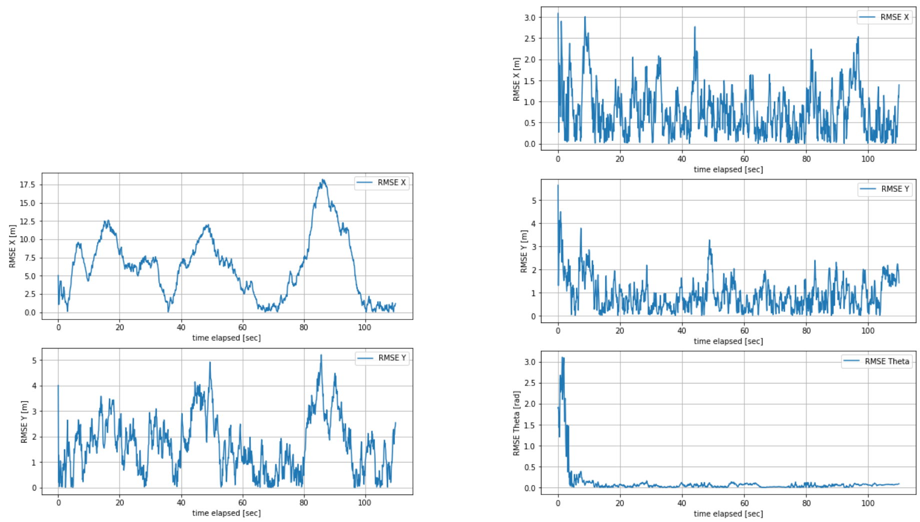
7.4.2 Root Mean Squared Error

RMSE provides an average measure of the deviation between estimated and true values over time. It is defined as:

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

- **Mathematical Rigor**: RMSE provides a straightforward metric to assess overall estimation accuracy. Lower RMSE values indicate better estimation performance.
- **Utility in KITTI Dataset**: With access to ground truth data, RMSE allows direct comparison of estimated positions and velocities with actual values.
- **KF vs EKF Performance**: In the analysis:
  - KF: Exhibits higher RMSE values, particularly in x-direction (0-12.5) due to limitations in handling nonlinearities and higher noise.
  - EKF: Shows improved RMSE performance, with values lower and more consistent across all dimensions (X, Y, θ), reflecting its ability to model nonlinear dynamics more accurately.

Figure 12 displays the RMSE plots for KF and EKF:



(a) KF RMSE

(b) EKF RMSE

Figure 12: KF vs EKF RMSE

In Figure 12a, KF's higher RMSE values suggest less accurate estimation, especially evident in x-direction variability. EKF's improved performance in Figure 12b highlights its effectiveness in reducing estimation errors across all evaluated dimensions.

These metrics collectively provide a comprehensive evaluation of KF and EKF performance, highlighting the advantages of EKF in handling nonlinear dynamics and improving overall estimation accuracy compared to KF.

8 Discussion and Conclusion

8.1 Discussion

The comparison between the Kalman Filter (KF) and Extended Kalman Filter (EKF) presented a nuanced evaluation of their performance in state estimation using the KITTI dataset. The KF, operating under the assumption of linear dynamics and Gaussian noise, demonstrated robust trajectory estimation capabilities but exhibited limitations in handling non-linearities present in real-world systems. Conversely, the EKF, through its linearization of non-linear dynamics, provided enhanced trajectory and orientation estimates by incorporating IMU data, showcasing its adaptability to more complex scenarios.

The metrics of Normalized Estimation Error Squared (NEES) and Root Mean Squared Error (RMSE) were instrumental in quantifying the performance of each filter. The KF, while effective in scenarios with linear dynamics, showed occasional overconfidence due to its simplified noise model. This was reflected in NEES values occasionally exceeding the expected range, indicating a potential underestimation of measurement noise variability. RMSE analysis further underscored this, particularly in the X-direction, where deviations from ground truth were more pronounced.

In contrast, the EKF's NEES values aligned more closely with theoretical expectations, reflecting its ability to handle non-linearities more adeptly. RMSE results consistently showed superior accuracy across all dimensions (X, Y, θ), emphasizing its capability to mitigate errors arising from non-linear system dynamics and noisy measurements.

A logical extension of the simulation involves extending the simulation to include the Unscented Kalman Filter (UKF) to provide a comprehensive comparison among the three major Kalman Filter variants. The UKF, which avoids linearization by directly propagating the mean and covariance through a deterministic set of points, could potentially offer a middle ground between computational complexity and accuracy, especially in scenarios with highly non-linear dynamics or when accurate modeling of noise statistics is critical.

Moreover, expanding the simulation across multiple drives of the KITTI dataset would validate the robustness and generalizability of conclusions drawn from the current study. Variability across different driving conditions and environments could provide further insights into the filters' performance under diverse real-world scenarios.

Further enhancing the EKF by incorporating additional sensor modalities such as LiDAR, which provides precise spatial information, could improve state estimation accuracy. LiDAR's ability to directly measure distance and reflectivity could complement GPS and IMU data, particularly in scenarios where GPS signal may be degraded (e.g., urban canyons).

On the theoretical side, beyond the KF, EKF, and UKF, exploring other Kalman Filter variants like the Particle Filter (PF) for non-linear and non-Gaussian state estimation could offer insights into alternative methodologies. PF's ability to represent the posterior distribution using a set of particles could overcome limitations of linearization inherent in EKF and UKF, albeit at the cost of increased computational complexity.

Theoretical research to enhance specific Kalman Filters could focus on improving the EKF's performance in scenarios with highly non-linear dynamics. One potential approach involves integrating more sophisticated non-linear models directly into the state transition and measurement functions. For instance, leveraging higher-order Taylor series expansions or Gaussian process models could provide more accurate approximations of non-linear system behavior.

8.2 Conclusion

By providing a comprehensive and complete framework for the KF, EKF and UKF, while also demonstrating the effectiveness of the KF vs EKF in a simulation using the KITTI dataset, this thesis contributes to advancing the understanding and application of Kalman Filters in autonomous systems, paving the way for more accurate and robust state estimation methodologies.

References

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[2] E Rudolf. Kalman. a new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[3] E Rudolf. New results in linear filtering and prediction theory. *Journal of basic Engineering*, 1961.

[4] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.