CHALMERS
UNIVERSITY OF TECHNOLOGY

Robust Self-Localization in the Urban Driving Environment

Mitigating the Impact of Anomalous GPS Behavior in a Last-Mile Delivery Platform Using an Adaptive Unscented Kalman Filter and the Normalized Innovation Squared Test

Master's thesis in Systems, Control, and Mechatronics

OTTO ERIKSSON
ALEXANDER ODHNER

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Robust Self-Localization in the Urban Driving Environment

Mitigating the Impact of Anomalous GPS Behavior in a Last-Mile Delivery Platform Using an Adaptive Unscented Kalman Filter and the Normalized Innovation Squared Test

OTTO ERIKSSON
ALEXANDER ODHNER

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Robust Self-Localization in the Urban Driving Environment
Mitigating the Impact of Anomalous GPS Behavior in a Last-Mile Delivery Platform Using an Adaptive Unscented Kalman Filter and the Normalized Innovation Squared Test

OTTO ERIKSSON
ALEXANDER ODHNER

Supervisor: Ludvig Hazard, Aptiv
Supervisor: Rickard Nilsson, Aptiv
Supervisor: Jonas Sjöberg, Department of Electrical Engineering
Examiner: Jonas Sjöberg, Department of Electrical Engineering

Master's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 10 00

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Robust Self-Localization in the Urban Driving Environment
Mitigating the Impact of Anomalous GPS Behavior in a Last-Mile Delivery Platform Using an Adaptive Unscented Kalman Filter and the Normalized Innovation Squared Test
OTTO ERIKSSON, ALEXANDER ODHNER
Department of Electrical Engineering
Chalmers University of Technology

Abstract

A critical component to safe autonomous driving is an accurate state estimate of the host vehicle. This estimation relies on fusing data from onboard sensors, where a self-localization algorithm utilizes available sensory information to produce an informed estimate of the vehicle's state. However, the accuracy and availability of GPS signals, which heavily influences the self-localization algorithm, can be compromised in the urban driving environment due to various factors.

This thesis addresses the self-localization problem that arises when sensor accuracy is non-stationary and affected by external factors, with particular focus on GPS-related faults. An Unscented Kalman Filter with asynchronous updates is employed, enabling the fusion of sensors operating at different sampling rates. The measurement model considers inputs from GPS, IMU, rotary encoders, and a pose estimate obtained from a visual SLAM algorithm. During filtering, the measurements are subjected to a normalized innovation squared test in order to detect and isolate faulty measurements. Furthermore, the system adapts to non-stationary sensor accuracy by a residual-based update of the expected measurement noise.

The developed self-localization algorithm is evaluated using real-world sensor data collected by a research vehicle, where simulated GPS-related errors are injected in post-processing. The results demonstrate that the algorithm provides a robust state estimate, even in the presence of faulty GPS signals. The developed state estimator is proven to provide a more accurate positional estimate than the current state estimator of the research vehicle, which utilizes an Extended Kalman Filter, and also outperforms a conventional Unscented Kalman Filter. Additionally, subsets of the developed algorithm are evaluated to showcase the collective contribution of each component in achieving overall resilience to faulty sensor measurements. It is observed that the algorithm effectively mitigates the impact of three common error types: dropout, varying noise, and offset. However, it is also noted that not all types of drifting measurement errors can be effectively mitigated.

Keywords: Self-localization, Adaptive Unscented Kalman Filter, Fault Detection and Isolation, GPS dropout, GPS multi-path

Acknowledgements

First and foremost, we would like to extend our gratitude to our company supervisors, Ludvig Hazard and Rickard Nilsson, for all their help and guidance throughout this thesis. We are also grateful to Simon Björklund at Aptiv for his assistance with hardware-related matters and data collection. Lastly, we thank Professor Jonas Sjöberg for examining our thesis.

Otto Eriksson & Alexander Odhner, Gothenburg, June, 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADAS	Advanced Driver Assistance System
AKF	Adaptive Kalman Filter
AUKF-FDI	Adaptive Unscented Kalman Filter with Fault Detection and Isolation
CMOD	City Mobility On Demand
DF	Degrees of Freedom
ECEF	Earth-centered, Earth-fixed coordinate system
EKF	Extended Kalman Filter
ENU	East, North, Up
FDI	Fault Detection and Isolation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
MEMS	Micro Electro-Mechanical System
NIS	Normalized Innovation Squared
PF	Particle Filter
RL	Rear Left
RMSE	Root Mean Square Error
ROS	Robot Operating System
RPM	Revolutions Per Minute
RR	Rear Right
RTK	Real Time Kinematic
SLAM	Simultaneous Localization And Mapping
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter

Contents

List of Figures

List of Tables

1

Introduction

This chapter begins with a brief background and motivation for the thesis. A summary of the key contributions made throughout the study is then provided, along with the limitations of the thesis. A brief literature review explores various self-localization methods and examines how to identify and handle faulty sensor behavior. A concise overview of the developed self-localization algorithm is provided, and lastly, the chapter concludes with an outline of the upcoming chapters.

1.1 Background

Autonomous Vehicles are a central part of today's automotive industry and lower-level autonomous behavior is already common in newer vehicles, such as Adaptive Cruise Control, Lane Keep Assistance, and Autonomous Emergency Braking. Such Advanced Driving Assistance Systems (ADAS) aims to improve road safety where further development aspires to eventually extend ADAS to full autonomy.

The development of autonomous driving in public traffic emphasizes improving driving accuracy and reliability. A key concept of autonomous driving is self-localization; the system's ability to determine its position in regard to the surrounding environment. Self-localization is performed with the aid of sensors that measures the internal and external conditions of the vehicle. A self-localization algorithm produces an informed estimate of the vehicle's state by considering several different types of sensor measurements, this is known as sensor fusion. Driving in public traffic will subject a vehicle to several challenging environments, such as harsh weather, poor lighting conditions, or areas without reliable GPS coverage, all of which can inhibit the performance of one or several sensors. Varying sensor reliability highlights the importance of redundant systems operating sufficiently on a subset of information, as even a minor offset from the true state can cause collisions or otherwise hazardous situations. During sensor dropout it is essential the the self-localization algorithm still provides a reliable estimate based on its other available sensor readings and that the system is aware of the uncertainty of its current estimate. Similarly, should a sensor become unreliable, it is crucial that the system is able to determine the measurement quality and adjust the trust accordingly to prevent degradation of the state estimate.

1.2 Contributions

This thesis contributes to the field of autonomous navigation by developing a robust self-localization algorithm for ground moving vehicles. The algorithm addresses scenarios with faulty sensor measurements, with particular focus on GPS-related faults. The key contributions of this research can be summarized as follows:

- The thesis performs a modeling of a research vehicle provided by the automotive company Aptiv. The kinematic bicycle model is employed to describe the behaviour of the vehicle, and a measurement model is defined to account for the multiple onboard sensors of the vehicle.
- A self-localization algorithm is developed to estimate a vehicle's state by utilizing information from multiple onboard sensors operating at different sampling rates. The algorithm accounts for the possible presence of faulty sensor signals and aims to consistently provide a reliable and accurate state estimate.
- The algorithm uses a statistical test for detecting faulty sensor readings during the self-localization process. By analyzing the real-time data from onboard sensors, the algorithm can identify and exclude faulty measurements, ensuring robustness and reliability of the state estimate.
- The algorithm dynamically assigns varying degrees of trust to each sensor signal based on their individual noise characteristics at any given time, effectively mitigating the impact of non-stationary sensor reliability.

The performance of the developed algorithm is assessed using real-world data collected by the research vehicle. The evaluation involves injecting three types of GPS-related errors into the data stream: dropout, varying noise, and multi-path errors. A comparison is made between the performance of the developed algorithm and the currently implemented state estimator of the research vehicle, an Extended Kalman Filter. Furthermore, subsets of the developed algorithm are evaluated to demonstrate how each component of the solution collectively contributes to achieving overall resilience to faulty sensor measurements.

1.3 Limitations

This thesis has certain limitations, which are outlined as follows:

- The thesis does not consider computational and memory requirements for integration of the solution in a real-time scenario, testing and evaluation of the developed algorithm is performed offline.
- Currently implemented methods for processing the raw camera data in the research vehicle are utilized as is; alternative data processing is not considered, i.e., investigating various SLAM algorithms.
- The evaluation is based on data gathered at a test track, as the research vehicle is not permitted to operate in public traffic.

- The evaluation relies on GPS measurements as ground truth; it is only possible to inject GPS-related errors in post-processing of the data.

1.4 Previous and Related Research

The topic of the thesis can be split into two distinct areas of interest; self-localization based on the input of multiple sensor readings, and how to handle faulty sensor readings. Section 1.4.1 examines state-of-the-art techniques for self-localization, and Section 1.4.2 explores sensor fault detection and isolation methods that can be applied as an extension to self-localization.

1.4.1 Self-Localization

Self-localization based on multiple sensor readings is an extensively researched subject. Rudolf Emil Kálmán was one of the primary developers of the Kalman Filter in 1960, a method based on Bayesian probability theory which fuses several uncertain measurements to estimate unknown variables at a higher accuracy than each individual measurement [1]. The Kalman filter utilizes a linear model to describe the system dynamics and measurement mapping and is thus not ideal for many real-world applications. The Extended Kalman Filter (EKF) is frequently utilized in vehicle localization, it handles non-linearities in both system dynamics and measurement mappings by linearizing the system around its current state and applying the linear Kalman filtering equations. An example of the EKF can be found in the work of Rezaei et al. who successfully applied an Extended Kalman Filter to fuse measurements from the GPS, gyroscope, and wheel encoders of a car [2].

The linearization of the EKF requires analytical derivation and well-defined Jacobians that generally require simplifications and assumptions to be made of the system dynamics. Furthermore, the EKF will generalize poorly in situations where the system experiences heavy non-linearities. Julier and Uhlmann proposes the Unscented Kalman Filter (UKF) to better generalize a non-linear system [3]. The UKF utilizes a set of carefully chosen sample points to parameterise the mean and covariance of a probability distribution. In the works by Merwe and Wan [4] [5], and also by Ullah et al. [6], it is proven that the UKF is superior to the EKF under heavy non-linearity, and that the filters perform equally otherwise. Merwe and Wan compared the positional estimate of an Unmanned Aerial Vehicle (UAV) using the EKF and UKF, their experiments found that the UKF improved the positional estimate by 30% compared to the EKF [5].

The performance of the Kalman filtering solution is highly dependent on how accurately any process and measurement noise can be modeled. The noise covariances are typically found through ad-hoc methods and manually adjusted through trial and error. Improper modelling can even cause the filter to diverge [7]. Akhlaghi et al. states that the ad-hoc modelling approaches is not adequate to achieve the best filtering performance and instead proposes a method that adapts the noise covariances in real-time [8]. The adaptive filtering technique has seen many implementation across literature, especially in scenarios where a sensor's accuracy is

environmentally dependent. Hu et al. employed the Adaptive Kalman Filter (AKF) in vehicle navigation to dynamically estimate GPS covariances where the standard KF had a root mean square positional error of 1.56 m, and the AKF 0.97 m under GPS measurements with added noise [9].

1.4.2 Fault Detection and Isolation

Fault Detection and Isolation (FDI) considers methods used to monitor the quality of sensor signals to allow the system to exclude erroneous signal readings. Kobayashi et al. proposes a solution using a Bank of Kalman filters to perform fault detection for sensors in aircraft engines [10]. A Kalman Filter is constructed for each possible sensor pairing where each filter is designed to detect a specific sensor or actuator fault. By comparing the state covariance matrices of all possible filters running in parallel, one can isolate the sensor that provides faulty readings.

Mori et al. consider FDI in self-localization of a vehicle; the base filtering solution is the Unscented Kalman Filter, and each sensor's measurements are evaluated by Hotelling's T^2-test which compares the difference between the predicted and true measurements. The test considers the correlation between each sensor's data and computes the probability that the measurements are reliable. By computing a probability regarding each sensor's reliability and checking this against a predefined statistical threshold, faulty sensor values can be excluded from the filtering process. Fault detection methods that compares statistics of the expected measurements and the true measurements is an established method, and similar solutions are used in in the works of Guo et al. [11] for localization of mobile robots, and by Bae et al. [12] for satellite attitude estimation.

Learning-based approaches have also been widely used for fault detection, Safavi et al. states that with the increase in available computational power, memory capacity, and sensor technology, deep learning methods have become the state-of-the-art frameworks for sensor health monitoring and fault detection [13]. Van Wyk et al. uses a combination of a Convolutional Neural Network(CNN) and a traditional Kalman filter for anomaly detection, to detect and identify anomalous sensor behavior in real time in connected and automated vehicles [14]. The proposed solution is general with regards to what type of sensors that can be evaluated and uses time-series data collected from several sensors as input to the network. To simulate sensor malfunctions they inject sensor errors into previously collected driving data. The simulated errors fall into four major anomaly types; instant, constant, gradual drift, and bias. According to their results, the combination of a CNN and a Kalman filter approach provides a higher accuracy in fault detection because of the inherent differences between the CNN and Kalman filtering solutions.

1.5 High Level Structure of Developed Self-Localization Algorithm

The developed self-localization algorithm presented in this thesis fuses the information from an IMU, GPS receiver, wheel encoders, and a visual SLAM algorithm to provide a robust state estimate. The modeling of the research vehicle's behavior and the model of the fused measurements are utilized in an Unscented Kalman Filter (UKF) that computes the state estimate. The functionality of the state estimator has been extended with a Fault Detection and Isolation (FDI) component and an adaptive update of the expected measurement noise.

The FDI component utilizes the normalized innovation squared test to evaluate each sensor measurement and determine if measurements are outside an expected statistical range. The FDI evaluation is based on the difference between the predicted and actual measurements. Faulty measurements are isolated to assure that they do not deteriorate the quality of the state estimate. The adaptive update estimates the expected measurement noise covariance at each time step of the filter by considering current and past noise characteristics. The state estimate of the filter update is then corrected using the new expected noise covariance before it is passed to the next iteration of the filter. The adaptive component provides a method of utilizing sensor data that has degraded in quality or have noise characteristics that vary over time. A high-level flowchart of the developed self-localization algorithm and how the components interact can be seen in Figure 1.1.

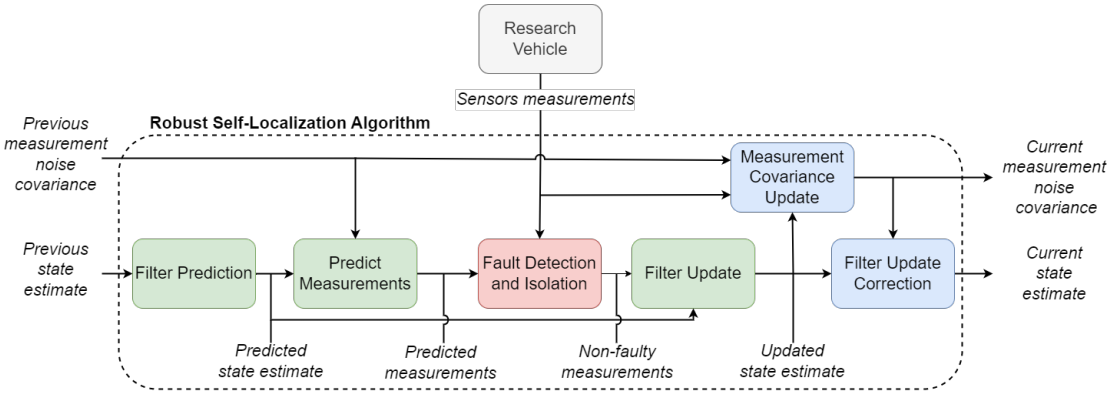


Figure 1.1: High-level structure of the developed self-localization algorithm. The green blocks constitutes functions of the UKF, the red block is the FDI component, and the blue blocks denote the adaptive noise update and its correction of the filter update.

It should be noted that the computations performed in the block "Predict Measurements" is conventionally part of the filter update process of an UKF. However, in the developed self-localization algorithm, these functions are separated because the predicted measurements are first passed to the FDI component before updating the state estimate. The algorithm is divided into three main components, each repre-

sented by a different color in Figure 1.1. The following sections cover each of the components:

- **UKF (green):** Section 3.1.4 presents the filtering equations, while Sections 4.1 and 4.2 derives the process and measurement models used by the filter.
- **FDI (red):** Section 3.2 covers the fundamental theory of the FDI method, while Section 4.3.1 presents how the FDI is incorporated in the self-localization algorithm.
- **Adaptive Update (blue):** Section 3.3 introduces the theoretical foundation of the adaptive update technique, while Section 4.3.2 presents the specific equations utilized in the self-localization algorithm.

The complete algorithm is described in Section 4.3, accompanied by a more detailed flowchart in Figure 4.4, and the corresponding pseudocode of the complete algorithm can be found in Algorithm 1.

1.6 Thesis Outline

The remainder of the thesis is structured as follows:

- Chapter 2 provides an overview of the research vehicle that is used for data collection and explains how the sensors work, what measurements each onboard sensor provides, and potential sources of erroneous sensor readings.
- Chapter 3 presents the theoretical background to Bayesian filtering and provides the equations of the UKF. The chapter also covers the foundations of the methods used for FDI and the adaptive update of the measurement noise covariance. Additionally, the chapter defines the coordinate systems and transforms relevant to the self-localization algorithm.
- Chapter 4 describes the modeling of the research vehicle and onboard sensors, presents the complete self-localization algorithm, and explains how the algorithm's components interact to achieve a robust state estimate.
- Chapter 5 covers the evaluation of the developed self-localization algorithm under four scenarios recorded by the research vehicle. The chapter concludes with a summary and discussion of the obtained results.
- Chapter 6 provides a conclusion of the work conducted in this thesis and proposes future areas for which to improve and extend the solution.

2

Research vehicle

This chapter provides an overview of the research vehicle that has been utilized for collecting sensor data. By familiarizing the reader with the research vehicle and its onboard sensors, this chapter establishes the necessary context for the mathematical modeling of the system and the evaluation strategy presented in subsequent chapters. The specifications of the research vehicle and its onboard sensors are provided in Section 2.1. Next, the technical background of each sensor and the potential causes of faulty sensor readings are covered in Section 2.2.

2.1 City Mobility On Demand Research Vehicle

The thesis is carried out in partnership with Aptiv, an automotive company that has provided the research vehicle used for collecting sensor data. Aptiv is conducting a project titled City Mobility On Demand (CMOD), with the intention of developing a fully autonomous last-mile delivery vehicle intended for the urban driving environment. The testing and development platform is shown in Figure 2.1; it is a motorized three-wheeled rickshaw equipped with an IMU, a dual-antenna GNSS receiver connected to the GPS, a front-facing camera, and rotary encoders measuring steering angle and wheel speed.

Figure 2.1: The City Mobility On Demand research vehicle utilized by this thesis for collecting sensor data.

A currently implemented Extended Kalman Filter (EKF) provides self-localization by fusing the readings of the IMU, the GPS position, the wheel speed, and the steering angle; a SLAM algorithm processes the camera feed, but this information is not fused with the other sensors. If the quality of a sensor reading unexpectedly decreases, the EKF is unable to detect and account for this, causing the accuracy of the state estimate to deteriorate. Furthermore, the EKF is imported from an open source library [15] designed to provide a generalized self-localization solution for a broad range of robot applications; this hinders customization of the filter.

The top speed of the vehicle is about 5 m/s and it has a maximum steering angle of $\pm 32.5°$. During data collection, the vehicle is operated remotely using a PlayStation 4 controller. The vehicle utilizes the Robot Operating System (ROS) framework for integration of the vehicle and its sensors.

2.1.1 Onboard Sensor Specifications

Trimble BX992 Dual-Antenna GPS Receiver

The CMOD vehicle utilizes the Trimble BX992 Dual-Antenna GPS Receiver which is shown in Figure 2.2. There is one antenna mounted to each rear corner of the vehicle. The Trimble BX992 is connected to the GPS and provides geodetic coordinates, heading with respect to true north, and absolute velocity, the specifications of the Trimble BX992 is provided in Table 2.1.

Figure 2.2: Trimble BX992 Dual-Antenna GPS Receiver.

Table 2.1: Trimble BX992 Dual-Antenna GPS Receiver specifications and output.

Sampling rate	[Hz]	5
Position accuracy	[m]	0.25
Velocity accuracy	[m/s]	0.007
Heading accuracy	[deg]	~ 0.30
Measurements		Position, Velocity, Heading

HG1120 MEMS Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is mounted over the rear-axle of the CMOD research vehicle and provides measurements of linear acceleration and angular rate.

The HG1120 MEMS IMU, seen in Figure 2.3, is a high performance micro-electro-mechanical system (MEMS) IMU designed for applications such as unmanned vehicles, industrial equipment, robotics and transportation. The HG1120 includes gyroscopes, accelerometers and magnetometers, providing 9 degrees of freedom. The specifications of the HG1120 MEMS IMU is provided in Table 2.2.

Figure 2.3: HG1120 MEMS Inertial Measurement Unit.

Table 2.2: HG1120 MEMS IMU specifications and output.

Sampling rate	[Hz]	600
Accel bias instability	$[mg, 1\sigma]$	0.15
Accel output noise	$[m/s]$	7.62×10^{-5}
Velocity random walk	$[m/s/\sqrt{hr}]$	0.09
Gyro bias instability	$[^\circ/hr, 1\sigma]$	24
Gyro output noise	$[rad, 1\sigma]$	0.001
Angular random walk	$[^\circ/\sqrt{hr}]$	0.4
Measurements		Linear acceleration, Angular rate

Rotary Encoders

The CMOD vehicle uses Sevcon continuous rotation servo motors with build-in encoders that provides the individual RPM of the rear wheels, and a Bosch steering-angle sensor to measure the steering angle. The exact specifications of the wheel encoder in the servo is unavailable, and the sampling rate provided in Table 2.3 has been manually measured.

Table 2.3: Rotary encoders specifications and output.

Sampling rate	[Hz]	~ 200
Steering angle resolution	[deg]	0.10
Measurements		Rear right RPM, Rear left RPM, Steering angle

Intel RealSense Depth Camera D455

The CMOD vehicle has an Intel RealSense Depth Camera D455, see Figure 2.4, mounted above the windshield. The camera provides RGB images at a maximum resolution of 1280×800 pixels at 30 frames per second, using the camera at lower resolution allows for a higher frame rate of up to 90 fps. The camera is also capable of measuring the depth at a maximum resolution of 1280×720 with a range of $0.4 - 6.0$ meters (varies with lighting conditions). Processing images in real time is computationally demanding, therefore the CMOD vehicle operates the camera at a lower resolution and frame rate. Table 2.4 presents the settings used by the CMOD vehicle and the measurements that processing the image feed provides.

Figure 2.4: Intel RealSense Depth Camera D455.

Table 2.4: Intel RealSense Depth Camera D455 settings and ORB SLAM3 output.

RGB resolution	640×480
Depth resolution	640×480
Fps	15
Sampling rate [Hz]	15
Measurements	Relative position, Relative heading

2.2 Technical Background to Onboard Sensors

The following subsections presents a short summary of the technical background behind the sensors that are utilized in this thesis. The summary considers a high-level explanation of how the measurements are extracted and some common causes to faulty sensor behavior.

2.2.1 Global Navigation Satellite System

The Global Navigation Satellite System (GNSS) is a network of satellites that provides the global coordinates, the heading with respect to true north, and the absolute velocity of the receiver in all weather conditions to users on or near the Earth's surface. There are four available GNSS networks with global coverage [16].

- The Global Positioning System (GPS), operated by the United States.
- The Global Navigation Satellite System (GLONASS), operated by Russia.

- The Galileo system, operated by the European Union.
- The BeiDou system, operated by China.

The GNSS receiver on the ground must have an unobstructed line of sight to a minimum of three satellites to compute the position and velocity of the receiver. Each satellite transmits an electromagnetic wave signal to the receiver of the satellite's position and the exact time the signal was emitted based on its own atomic clock [17]. The timing of each signal is then used to determine a distance measurement known as psuedorange between the receiver and each satellite, and the longitudinal and lateral positions of the receiver is determined by trilateration of the psuedoranges of three satellites. If a fourth satellite signal is available the GNSS uses it to synchronize the timing information which improves accuracy, and to determine the altitude. If there are more than four satellites available, the receiver can triangulate different constellations of satellites to find its position by averaging the resulting positions. An illustration of the concept of trilateration is provided in Figure 2.5, note that the union of three spheres has two points of intersection, hence the need for the fourth satellite to reliably determine altitude.

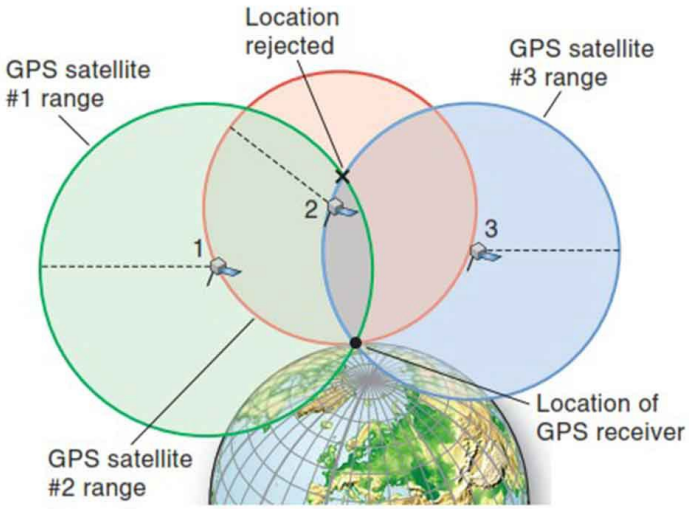


Figure 2.5: Example of trilateration with three satellites where the GNSS selects the point closest to Earth's surface. Figure credit: [18] CC3.0.

The velocity of the receiver is determined by measuring the Doppler frequency shifts of the received signal, this yields a psuedorange rate between the receiver and satellite. When the psuedorange rate between the receiver and each of the satellites is known, the receiver's velocity can also be determined by trilateration. Some low-end GNSS receivers compute the velocity by considering the change in position over time. In addition, the GNSS can also determine the heading of the receiver, this is possible by having a receiver pair sufficiently far apart on a vehicle and evaluating the position of both receivers. If only one receiver is utilized, the heading is determined by considering the change in position over time.

A common type of GPS-related error is the multi-path error. A multi-path error

occurs when the GNSS receiver receives reflected signals from surfaces such as buildings, trees, or other obstacles [19]. These reflected signals can cause the receiver to calculate incorrect psuedoranges, which can result in a significant positional error. Multi-path errors may also occur in scenarios where the receiver is unable to distinguish between a reflected signal and the direct signal. Multi-path errors can occur over short instances that momentarily scatters the position, and over longer periods of time which offsets the position. Illustrations of the multi-path error are presented in Figures 2.6 and 2.7. In summary, the performance of the GNSS receiver depends on satellite availability and the surrounding environment.

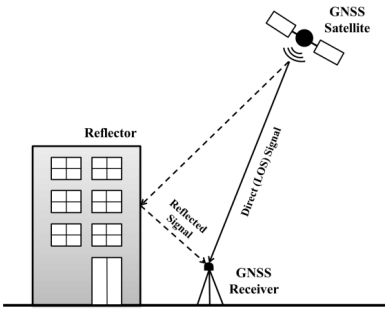


Figure 2.6: Illustration of the GNSS multi-path error. Figure credit: [20] CC4.0.

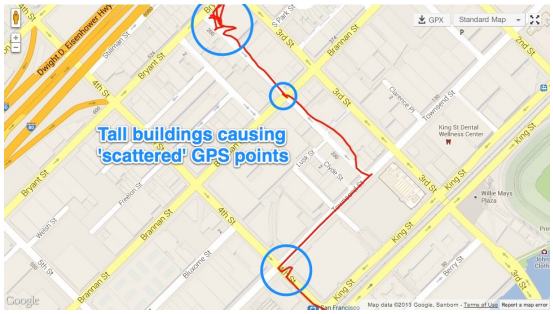


Figure 2.7: Distorted GPS trajectory when the GNSS receiver is subjected to multi-path errors in the urban environment. Figure credit: [21].

2.2.2 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is a self-contained electronic device that conventionally consists of three accelerometers and three gyroscopes which provides the body-frame accelerations and angular rates. Some IMU's are also equipped with magnetometers which provides magnetic field measurements used to determine the global yaw, pitch, and roll. A magnetometer measures the Earth's magnetic field intensity, it is easily distorted by other electromagnetic fields and thus not reliable in an automotive application [22]. The accelerometers and gyroscopes of an IMU are prone to errors due to misalignment, scale factor, bias, and random noise [23]. Both the accelerometer and gyroscope measurements are susceptible to shocks, high frequency vibrations, and temperature variations.

2.2.2.1 Accelerometer

An accelerometer measures the external forces acting on the sensor. There are several kinds of accelerometers but the principal concept is to measure the deflection of a small mass that is suspended by springs. By measuring the displacement δ of the mass, and knowing the spring constant k and mass m, we can find the acceleration a acting on the mass by Newton's second law of motion, $a = k\delta/m$. By using three 1D accelerometers mounted on an orthogonal triad, it is possible to measure the linear acceleration in all three axes of the sensor coordinate frame. The gravitational force is always acting on the sensor which means that while the system is at rest, the sensor should only output $-9.81 m/s^2$ along the z-axis.

2.2.2.2 Gyroscope

A gyroscope provides the angular rate around an axis of the sensor coordinate system by measuring the displacement of a mass through the Coriolis effect [24]. The Coriolis effect occurs when an angular rate is applied to a moving mass, the reacting force will displace the mass perpendicularly to the velocity direction. The resulting displacement is coupled to the applied angular rate. An IMU is able to measure the angular rate around the x, y, and z-axis by having three gyroscopes mounted on an orthogonal triad.

2.2.3 Rotary Encoders

A common type of rotary encoder considers a magnetic pickup and a toothed steel gear which is mounted on a shaft. The pickup measures the change in the magnetic field as the teeth of the gear pass by to find either the angular speed or the shaft angle. Rotary encoders can be mounted to a driving axle in order to find the wheel speed, or the steering axle to find the steering angle. Wheel speed is thus computed by multiplying the angular rate with the wheel circumference, and the steering angle can be measured directly. The measurements from a rotary encoder will only account for the applied shaft rotation, i.e., it does not account for wheel slippage.

2.2.4 Camera

The concept of computer vision through the processing of a camera feed is a heavily researched subject and there exist several techniques to derive the pose of the camera in regards to its surrounding environment [25] [26]. The CMOD vehicle utilizes a visual Simultaneous Localization And Mapping (SLAM) algorithm to estimate the camera pose, titled ORB SLAM3 [27]. It is outside the scope of the thesis to research alternative solutions for camera pose estimation, or to perform an in-depth evaluation of ORB SLAM3. A high-level summary of the algorithm is provided below.

SLAM is an algorithm that performs mid- to long-term data association by building a 3D point cloud of the surrounding environment through detecting, matching, and saving feature points in each consecutive image of a video feed. When SLAM is processing a new image, it will attempt to match the features in that image to the point map it has built up of the surrounding environment to determine the camera pose. An example of the simultaneous localization and mapping procedure is illustrated in Figure 2.8. ORB corresponds to a technique used to for feature detection and matching, and ORB SLAM3 is the third generation of the algorithm proposed by Campos et al. [27].

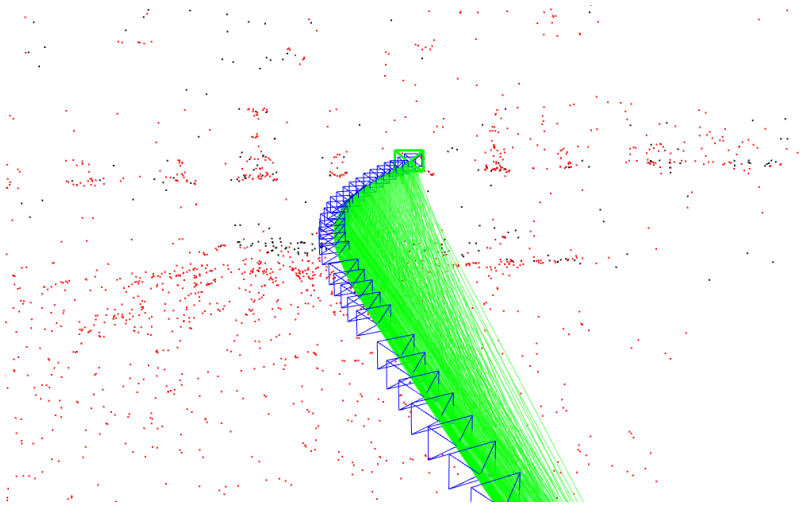


Figure 2.8: Example showing the resulting camera poses, seen in blue, over time as provided by the ORB SLAM3 algorithm. The red dots illustrates the point cloud of the environment, black dots are 3D positions of the features found in the the current image, and the green lines are vectors from each camera pose in relation to the origin.

ORB SLAM3 provides the position and orientation of the camera in regards to the initialization point, meaning the position of the first image, and the relative position between each frame is then computed which provides the pose of the camera over time. A common issue with vision-based navigation is that information from images alone only provides a relative solution, the scale of the environment is unknown. Providing ORB SLAM3 with the depth information by using a depth camera mitigates the aforementioned issue with scale, but does not solve it reliably in scenes containing distances over the maximum depth-range. Another issue is accumulative error over time, if one pose is estimated incorrectly, this error will carry over into all successive estimations causing drift. Furthermore, there are three conditions that must be met for effective visual navigation; the lighting must be adequate, the scene must contain many static features, and there needs to be a sufficient overlap between consecutive images to match said static features. Should the aforementioned conditions not be met, ORB SLAM3 is likely to loose tracking and will need to reinitialize the mapping.

3

Theoretical Background

This chapter aims to provide the necessary preliminaries to understand the functionalities of the developed self-localization algorithm. The chapter covers the theoretical background of the three main components of the algorithm: the Unscented Kalman Filter, fault detection using the normalized innovation squared test, and the adaptive update of the measurement noise covariance. The first section introduces Bayesian filtering techniques that form the basis of the Unscented Kalman Filter. Subsequently, the following section explores fault detection based on the Mahalanobis distance and its application in detecting faulty measurements using the normalized innovation squared test. The next section covers the adaptive measurement noise covariance update, a method of adapting the state estimator in real-time to non-stationary sensor accuracy. Furthermore, the coordinate systems and coordinate system transforms that are relevant to the self-localization algorithm are defined at the end of this chapter.

3.1 Statistical Filtering for Self-Localization

The following sections introduces the background and algorithms to Bayesian filters. Section 3.1.1 briefly introduces the general Bayesian framework, the foundation for the type of optimal filtering techniques that this thesis considers. Section 3.1.2 covers the Linear Kalman Filter, a recursive algorithm than provides an optimal state estimate in linear and Gaussian systems. The Extended Kalman Filter and the Unscented Kalman Filter are modifications of the linear Kalman Filter that are capable of handling non-linear systems, these filters are presented in Section 3.1.3 and 3.1.4, respectively.

3.1.1 Bayesian Filtering

Bayesian filtering is a well-established framework for estimating the probability distribution of an unknown state vector $\boldsymbol{x}_k$ at the discrete time instance k, based on a set of observations collected up to the current time, $\boldsymbol{y}_{0:k} = \{y_0, y_1, ..., y_k\}$. The state sequence is assumed to be a Markov process, meaning that the present state $\boldsymbol{x}_k$ is only dependent on the previous state $\boldsymbol{x}_{k-1}$ and conditionally independent of all earlier states, such that

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{3.1}$$

In addition, a measurement at time k is assumed to be independent of all other states and measurements, which means that

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k) \tag{3.2}$$

The assumed relations between the states and measurements are visualised in Figure 3.1.

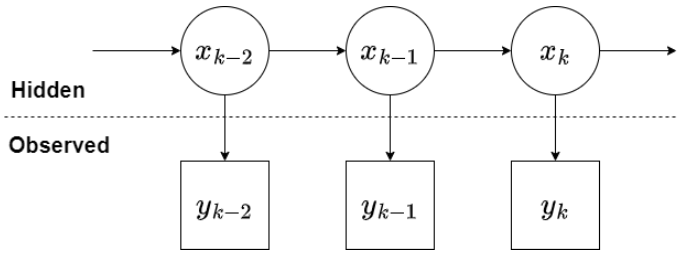


Figure 3.1: A visualization of the assumed dependencies between state and measurements for a hidden-Markov process utilized in recursive Bayesian filtering.

With the aforementioned assumptions in mind, the objective of Bayesian filtering can be regarded as recursively estimating the posterior probability density function $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ in each time-instance while regarding newly incoming measurements. The posterior density is then used to determine an estimate $\hat{\boldsymbol{x}}$ of the current state based on all measurements up to and including time instance k. To show how this can be achieved the measurements in the posterior can be split up as

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{y}_{1:k-1}) \tag{3.3}$$

By applying Baye's rule and the assumed Markov process property, the posterior can be described as a combination of conditional probabilities

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \tag{3.4}$$

Where $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is known as the likelihood and describes the probability density function of the measurements at time k based on the current state. The term $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$ describes the predicted density of the current state based on all previous measurements, up to time $k-1$, this is known as the prior. The denominator $p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})$ of Equation (3.4) is a normalization factor that ensure that the posterior density integrates to one and only affects the amplitude of the probability density function. Bayesian filtering only considers the mean and covariance of the probability density function, meaning that the normalization factor can be ignored. The resulting posterior can therefore be expressed as

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood} \tag{3.5}$$

Which states that the posterior is proportional to the prior times the likelihood, and therefore tells us that the posterior can be found by combining the information from the current measurement and the predicted state based on previous measurements.

To find the predicted distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$, the Chapman-Kolmogorov equation can be utilized, which marginalizes over the state at the previous time step

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) &= \int p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1} \\ &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1} \end{aligned} \tag{3.6}$$

The term $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$ can be seen as the posterior at the previous time instance $k-1$, and the term $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ describes how the state evolves over time and is called the state transition density. When filtering, the state transition density $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is described by a process model, and the likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is provided by a measurement model. The models can therefore be expressed as

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \Leftrightarrow \boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{q}_{k-1}) \tag{3.7}$$

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) \Leftrightarrow \boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{r}_k) \tag{3.8}$$

Where $\boldsymbol{q_{k-1}}$, $\boldsymbol{r}_k$ is the process and measurement noise and describes the error or uncertainty in the models. $f_{k-1}(\cdot)$, $h_k(\cdot)$ is the state transition and measurement function, respectively. Using the methods provided in Equations (3.5)-(3.8), one can see that it is possible to recursively compute the posterior density of Equation (3.4) as new measurements are provided. This forward recursion step in each time instance is split into a prediction using the process model, followed by a update using the measurement model, as illustrated in Figure 3.2.

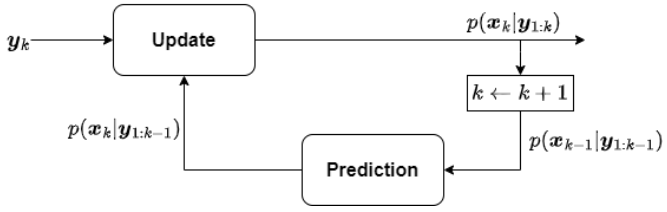


Figure 3.2: Structure of the recursive Bayesian filter that estimates the posterior density at each time-instance.

3.1.2 The Kalman Filter

If the process and measurement models are linear, with the process noise $\boldsymbol{q}_{k-1}$ and measurement noise $\boldsymbol{r}_k$ being additive, normally distributed, and independent over time, the filtering problem can be solved optimally using the linear Kalman filter. In this case Equations (3.7) and (3.8) are expressed in the following way

$$\boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{q}_{k-1} \tag{3.9}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{r}_k \tag{3.10}$$

where $\boldsymbol{q}_{k-1} \sim \mathcal{N}(0, \boldsymbol{Q}_{k-1})$ and $\boldsymbol{r}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$, $\boldsymbol{Q}_{k-1}$ is the process noise covariance matrix and $\boldsymbol{R}_k$ is the measurement noise covariance matrix. $\boldsymbol{A}$ and $\boldsymbol{H}$ are linear matrix descriptions of the process and measurement models, respectively. As long as the initial prior density $p(\boldsymbol{x_0})$ is Gaussian, the posterior density in each step will also

be Gaussian, and the filtering can be performed optimally such that it minimizes the expected error in each state estimate. However, optimality is only guaranteed when the filter models accurately mirror the system's actual behavior.

The Kalman filter operates like the Bayesian filtering equations of Section 3.1.1 and consists of two steps, the prediction and the update. In the prediction step, the predicted mean $\hat{\boldsymbol{x}}_{k|k-1}$ and covariance $\boldsymbol{P}_{k|k-1}$ of the current state is computed by propagating the information from the previous time-instance through the process model. The prediction equations are defined as

Prediction

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} \tag{3.11}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_{k-1} + \boldsymbol{P}_{k-1|k-1}\boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1} \tag{3.12}$$

The calculated mean and covariance of the prediction step is then combined with the information from the gathered measurements in the update step of the filter. Resulting in the computation of the estimated mean and covariance that describe the posterior density as

Update

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\boldsymbol{v}_k \tag{3.13}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{P}_k^{yy}\boldsymbol{K}_k^T \tag{3.14}$$

where

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T(\boldsymbol{P}_k^{yy})^{-1} \tag{3.15}$$

$$\boldsymbol{v}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{3.16}$$

$$\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} \tag{3.17}$$

$$\boldsymbol{P}_k^{yy} = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k \tag{3.18}$$

$\boldsymbol{K}_k$ is the Kalman gain and determines how much weight the measurements should have in the final estimate of the posterior density. $\boldsymbol{v}_k$ is known as the innovation and constitutes the difference between the predicted measurements $\hat{\boldsymbol{y}}_{k|k-1}$ and the actual measurements $\boldsymbol{y}_k$, and $\boldsymbol{P}_k^{yy}$ represents the estimated innovation covariance.

3.1.3 The Extended Kalman Filter

The optimality of the linear Kalman filter breaks when the models describing the system are non-linear. When a Gaussian density is propagated through a non-linear model, the resulting density is no longer Gaussian. One natural solution to perform filtering with non-linear models is to first linearize the models and then apply the linear Kalman equations to perform the prediction and update, this type of filter is known as the Extended Kalman filter (EKF) [28]. In order to linearize the process model, a Taylor expansion around the previous state is performed, where the process model for the EKF is defined as

$$\begin{aligned} \boldsymbol{x}_{k|k-1} &= \boldsymbol{f}\left(\boldsymbol{x}_{k-1|k-1}\right) \\ &\approx \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}) + \boldsymbol{f'}(\hat{\boldsymbol{x}}_{k-1|k-1})(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k-1}) + \boldsymbol{q}_{k-1} \end{aligned} \tag{3.19}$$

Where $\boldsymbol{f'}(\hat{\boldsymbol{x}}_{k-1|k-1})$ is the Jacobian matrix of $\boldsymbol{f}\left(\boldsymbol{x}_{k-1|k-1}\right)$, evaluated around $\hat{\boldsymbol{x}}_{k-1|k-1}$. Furthermore, a Taylor expansion is also applied to the non-linear measurement model, where the linearization point is chosen as the predicted mean, which gives the measurement model as

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{r}_k) \\ &\approx \boldsymbol{h}_k(\hat{\boldsymbol{x}}_{k|k-1}) + \boldsymbol{h}'(\hat{x}_{k|k-1})(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) + \boldsymbol{r}_k \end{aligned} \tag{3.20}$$

The linearized models can then be applied to the equations presented for the linear Kalman filter, providing the prediction and update equations for the EKF as

Prediction EKF

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1|k-1}\right) \tag{3.21}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{f}'\left(\hat{\boldsymbol{x}}_{k-1|k-1}\right)\boldsymbol{P}_{k-1|k-1}\boldsymbol{f}'\left(\hat{\boldsymbol{x}}_{k-1|k-1}\right)^T + \boldsymbol{Q}_{k-1} \tag{3.22}$$

Update EKF

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\boldsymbol{v}_k \tag{3.23}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{P}_k^{yy}\boldsymbol{K}_k^T \tag{3.24}$$

Where the Kalman gain $\boldsymbol{K}_k$, innovation $\boldsymbol{v}_k$, predicted measurements $\hat{\mathbf{y}}_{k|k-1}$ and innovation covariance $\boldsymbol{P}_{k|k-1}^{yy}$ are computed as

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{h}'\left(\hat{\boldsymbol{x}}_{k|k-1}\right)^T(\boldsymbol{P}_k^{yy})^{-1} \tag{3.25}$$

$$\boldsymbol{v}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{3.26}$$

$$\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{k|k-1}\right) \tag{3.27}$$

$$\boldsymbol{P}_k^{yy} = \boldsymbol{h}'\left(\hat{\boldsymbol{x}}_{k|k-1}\right)\boldsymbol{P}_{k|k-1}\boldsymbol{h}'\left(\hat{\boldsymbol{x}}_{k|k-1}\right)^T + \boldsymbol{R}_k \tag{3.28}$$

Unlike the linear Kalman filter which is optimal in the linear case, the EKF is not an optimal estimator when the models are non-linear. The EKF provides a reasonable approximation under non-linearities that can be approximated well by the linearization, but the filter will likely diverge in the presence of heavy non-linearities.

3.1.4 The Unscented Kalman Filter

Another type of filter that can handle non-linear models, and the one used as the base filtering solution of the self-localization algorithm presented in this thesis, is the Unscented Kalman Filter (UKF). Like the EKF, the UKF approximates the desired densities as Gaussian. However, while the EKF approximates the predicted

and posterior densities by linearizing the models, the UKF does this by propagating a set of chosen points through the models, known as sigma points. By giving each sigma point a weight, the propagated points can be used to approximate the mean and covariance of the resulting densities. This way of propagating a set of sample points is called the Unscented Transform and builds upon the idea that it is easier to approximate a distribution than it is to approximate an arbitrary nonlinear function or transformation.

Like the linear Kalman filter and the EKF, the UKF can be divided into a prediction step and an update step. The first step of both the prediction and update is to generate a set of sigma point based on the previous or predicted state estimates, respectively. Wan and Merve presents the UKF for non-linear state estimation and selects the weight and sigma points based on a scaling parameter $\lambda = \alpha^2(L+\kappa) - L$ [29]. Where L is the number of states and κ, α and β are all tuning parameters. α determines the spread of the of the sigma points, and is usually set to a small positive value, typically $\alpha = 10^{-3}$, β is used to incorporate prior knowledge about the distribution, for Gaussian distributions $\beta = 2$ is ideal. κ is a secondary scaling parameter, and is typically set to zero [29]. The UKF weights and sigma points are computed as follows

UKF weights

$$\begin{aligned} W_0^{(m)} &= \lambda / (L+\lambda) \\ W_0^{(c)} &= \lambda / (L+\lambda) + \left(1 - \alpha^2 + \beta\right) \\ W_i^{(m)} &= W_i^{(c)} = \lambda / \{(2(L+\lambda)\} \quad i = 1, ..., 2L \end{aligned} \tag{3.29}$$

UKF Sigma points calculation

$$\begin{aligned} \mathcal{X}_0 &= \hat{\boldsymbol{x}} \\ \mathcal{X}_i &= \hat{\boldsymbol{x}} + \left(\sqrt{(L+\lambda)\,\boldsymbol{P}^{xx}}\right)_i \qquad i = 1, ..., L \\ \mathcal{X}_i &= \hat{\boldsymbol{x}} - \left(\sqrt{(L+\lambda)\,\boldsymbol{P}^{xx}}\right)_{i-l} \qquad i = L+1, ..., 2L \end{aligned} \tag{3.30}$$

Where $\hat{\boldsymbol{x}}$ is the mean of the distribution that is to be propagated through any of the non-linear models, and $\boldsymbol{P}^{xx}$ is its covariance. As one can see from Equation (3.29), the UKF have different weights for the state estimation and covariance estimation, denoted by $W_i^{(m)}$ and $W_i^{(c)}$, respectively. By propagating the generated sigma points through the process model, the mean and covariance for the prediction part of the UKF can calculated through a weighted sum of the propagated points as

Prediction UKF

$$\hat{\boldsymbol{x}}_{k|k-1} \approx \sum_{i=1}^{2n} \boldsymbol{f}\left(\boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}\right) W_i^{(m)} \tag{3.31}$$

$$\boldsymbol{P}_{k|k-1}^{xx} \approx \boldsymbol{Q}_{k-1} + \sum_{i=0}^{2n} \left(\boldsymbol{f}(\boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\boldsymbol{f}(\boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1}\right)^T W_i^{(c)} \tag{3.32}$$

For the update part of the filter, a new set of sigma points are generated using the predicted mean and covariance. The sigma points are then propagated through the measurement model to update the state estimate through the equations

Update UKF

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{v}_k \tag{3.33}$$

$$\boldsymbol{P}^{xx}_{k|k} = \boldsymbol{P}^{xx}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{P}^{yy}_{k|k-1} \boldsymbol{K}^T_k \tag{3.34}$$

Where the Kalman gain $\boldsymbol{K}_k$, innovation $\boldsymbol{v}_k$, predicted measurements $\hat{\mathbf{y}}_{k|k-1}$, cross covariance $\boldsymbol{P}^{xy}_{k|k-1}$ and innovation covariance $\boldsymbol{P}^{yy}_{k|k-1}$ are calculated as

$$\boldsymbol{K}_k = \boldsymbol{P}^{xy}_{k|k-1} \left(\boldsymbol{P}^{yy}_{k|k-1} \right)^{-1} \tag{3.35}$$

$$\boldsymbol{v}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{3.36}$$

$$\hat{\mathbf{y}}_{k|k-1} \approx \sum_{i=0}^{2n} \boldsymbol{h} \left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} \right) W_i^{(m)} \tag{3.37}$$

$$\boldsymbol{P}^{xy}_{k|k-1} \approx \sum_{i=0}^{2n} \left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} - \hat{\boldsymbol{x}}_{k|k-1} \right) \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}) - \hat{\boldsymbol{y}}_{k|k-1} \right) W_i^{(c)} \tag{3.38}$$

$$\boldsymbol{P}^{yy}_{k|k-1} \approx \boldsymbol{R}_k + \sum_{i=1}^{2n} \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}) - \hat{\boldsymbol{y}}_{k|k-1} \right) \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}) - \hat{\boldsymbol{y}}_{k|k-1} \right)^T W_i^{(c)} \tag{3.39}$$

such that the predicted measurements and covariances are calculated through a weighted sum of the propagated sigma points. Figure 3.3 illustrates the difference in how the linearization of the EKF and the Unscented Transform of the UKF approximates an arbitrary non-linear distribution, which shows why it could be beneficial to utilize a UKF when the models describing the system are highly non-linear. In addition, the UKF have the benefit of not requiring the Jacobians of the process and measurement function to be calculated, which can significantly lower the implementation complexity.

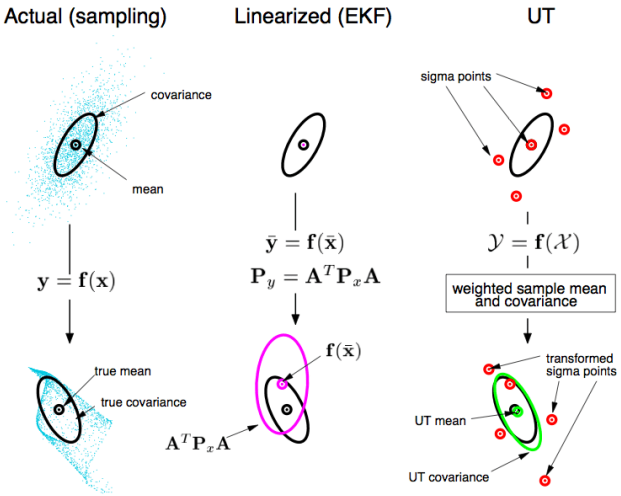


Figure 3.3: Example showing the difference of approximating an arbitrary distribution using the linearization technique utilized by the EKF and the unscented transform in the UKF. Figure credit: [29] CC4.0.

3.2 Fault Detection - The Normalized Innovation Squared Test

The Mahalanobis distance is a measure of how far a point $\bar{\boldsymbol{x}} = \{x_1, x_2, ..., x_k\}$ is from a multivariate distribution D with mean $\bar{\boldsymbol{\mu}} = \{\mu_1, \mu_2, ..., \mu_k\}$ [30]. The Euclidean distance between $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{\mu}}$ assumes that the data is isotropically Gaussian, meaning that its variance is uniformly distributed across all dimensions. The Mahalanobis distance, on the other hand, takes into account the possible correlation between the variables in the distribution and instead assumes it to be anisotropic Gaussian, where the variance can vary and have correlation across the dimensions. Therefore, the Mahalanobis distance can better represent how far away the point is from the expected values of the distribution when the variables are significantly correlated or have different scaling. The Mahalanobis distance, d, is defined as

$$d = \sqrt{(\bar{\boldsymbol{x}} - \bar{\boldsymbol{\mu}})^T \boldsymbol{S}^{-1} (\bar{\boldsymbol{x}} - \bar{\boldsymbol{\mu}})} \tag{3.40}$$

Where $\boldsymbol{S}$ is a positive-definite covariance matrix describing the expected covariance of the distribution. Naturally, a larger distance would indicate that the point is farther away from the expected values of the distribution.

When the underlying variables are Gaussian distributed, the square of the Mahalanobis distance will approximate to a chi-square distribution with k degrees of freedom, $d^2 = \chi_k^2$. The probability density function of a chi-square distribution and how it varies based on the number of variables can be seen in Figure 3.4.

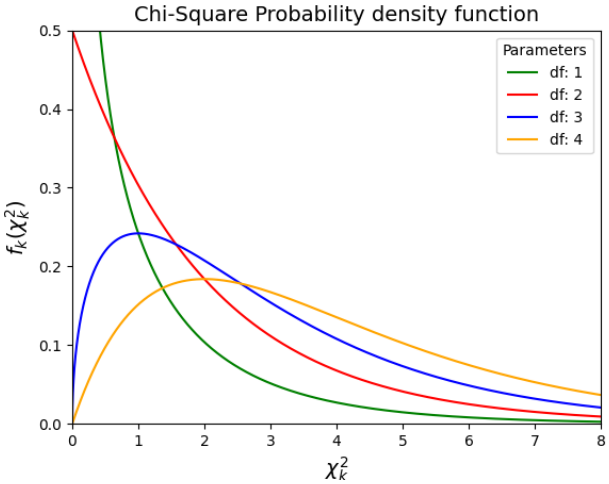


Figure 3.4: Probability density function for the chi-square distributions with k degrees of freedom (df), the plot visualizes how the probability of the squared Mahalanobis distance will vary depending on the dimension of the distribution.

In data analysis, the Mahalanobis distance is often used to find outliers in large data samples or in techniques such as classification and clustering [31]. Outlier detection is achieved by determining the probability that a certain point belongs to

the expected distribution based on the calculated distance and the corresponding chi-square probability density function.

To determine when a point should be rejected, one can compute a critical value based on a selected significance level α and the number of degrees of freedom of the data point. Should the Mahalanobis distance be above the critical value, the data point is considered an outlier and can be handled accordingly based on the application. The significance level is defined as the accepted probability that a point is falsely regarded as an outlier. The critical value for a certain significance level is either calculated directly from the inverse cumulative distribution function $F_k^{-1}(1-\alpha)$ of the chi-squared distribution, or the same value can be determined using a Chi-Square cumulative distribution table as seen in Table 3.1.

Table 3.1: Chi-square cumulative distribution table determining the critical value of the Mahalanobis distance based on the degrees of freedom of the distribution and the desired significance level.

Degrees of	**Significance level**							
freedom(DF)	**.99**	**.975**	**.95**	**.9**	**.1**	**.05**	**0.025**	**.01**
1	-	0.001	0.004	0.016	2.706	3.841	5.024	6.635
2	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210
3	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345
4	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277

For example, a selected significance level of $\alpha = 0.01$ would indicate that a point has a 1% chance of being falsely regarded as an outlier. Furthermore, with $\alpha = 0.01$ and the distribution having two degrees of freedom, the critical value would be 9.210 according to Table 3.1.

In Kalman filtering, this type of test can be utilized to perform what is known as a Normalized Innovation Squared (NIS) test to check when sensors are deemed to give non-expected values [32][33]. The NIS test can be performed both on the complete measurement vector to check the consistency of the complete filter, or on a subset of the measurement vector. The test utilizes the innovation vector computed in the update part of the Kalman filter, which is the difference between the observed measurements and predicted measurements, as derived in Section 3.1.2. The innovation computed in the filter is assumed to obey a Gaussian distribution with covariance $\boldsymbol{P}_{k|k-1}^{yy}$, and the distance is given as given as

$$d_k^{NIS} = (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1})^T (\boldsymbol{P}_{k|k-1}^{yy})^{-1} (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}) \tag{3.41}$$

By setting a critical value as a threshold for the distance in the same way as for the general outlier detection utilized in the data analysis domain, one can conclude when a measurement or a set of measurements does not coincide with the predicted measurements. A distance over the threshold would indicate that either the measurements can be regarded as faulty or that something about the underlying models and parameters in the filter is incorrect. The degrees of freedom of the resulting Chi-square distribution will be equal to the number of measurements being tested.

3.3 Adaptive Estimation of Measurement Noise Covariance

The filtering methods covered in Section 3.1 conventionally requires prior knowledge of the statistical noise characteristics the system is subjected to. Estimating the noise characteristics can prove difficult for large or complex systems, and improper noise modeling can cause a Bayesian filter to be biased or even diverge. Furthermore, there are also applications where the noise statistics are non-stationary due to changes in the surrounding environment, e.g. the GNSS presented in Section 2.2.1.

Adaptive Kalman Filtering is a technique which evaluates the noise characteristics at each time step to estimate the process and measurement noise covariance during runtime of the filter. The methods for adapting the process and measurement noise covariances are similar but the area of interest to the thesis only considers adapting the measurement noise covariance. Mohamed and Schwarz proposes utilizing the the innovation of the Kalman filter update to estimate the measurement noise covariance [34]. The innovation-based measurement noise covariance estimation is provided by the equation

$$\boldsymbol{R}'_k = \mathbb{E}[\boldsymbol{v}_k \boldsymbol{v}_k^T] - \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T \tag{3.42}$$

Here, $\boldsymbol{v}_k$ represents the innovation, $\boldsymbol{P}_{k|k-1}$ is the predicted state covariance, and $\boldsymbol{H}_k$ is the measurement function. It should be noted that the innovation-based approach does not guarantee a positive semi-definite $\boldsymbol{R}'_k$ because of the subtraction of two positive definite matrices. Should any value on the diagonal of $\boldsymbol{R}'_k$ become negative, $\boldsymbol{R}'_k$ is no longer a covariance matrix and the state estimate will diverge. A covariance matrix is defined as a square, symmetric, and positive semi-definite matrix with real and non-negative eigenvalues. Wang proposes an alternative method based on the residual $\boldsymbol{\varepsilon}_k$, the difference between the estimated and the actual measurements, to assure a positive semi-definite $\boldsymbol{R}_k$ [35]. The equations for calculating the measurement noise covariance using the residual based approach are

$$\boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k} \tag{3.43}$$

$$\boldsymbol{R}'_k = \mathbb{E}[\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T] + \boldsymbol{H}_k \boldsymbol{P}_{k|k} \boldsymbol{H}_k^T \tag{3.44}$$

The expected value of the squared residual can be found by averaging the residual values over time. A method of finding the average of the residuals is to employ a sliding window that carries the previous N residuals over each iteration of the filter

$$\boldsymbol{R}'_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k} \boldsymbol{H}_k^T + \frac{1}{N} \sum_{i=0}^{N-1} (\boldsymbol{y}_{k-i} - \hat{\boldsymbol{y}}_{k-i|k-i})(\boldsymbol{y}_{k-i} - \hat{\boldsymbol{y}}_{k-i|k-i})^T \tag{3.45}$$

As is evident from Equation (3.45), the sliding window results in the expected value of the squared residual being a positive semi-definite covariance matrix. The sliding window enables the computation of a theoretically suitable $\boldsymbol{R}'_k$ at each time step. However, utilizing a sliding window will inflate the memory requirements of a filter and furthermore, outlier datapoints will impair the estimation accuracy for as many time steps as the window size N.

A more memory-efficient alternative to the sliding window method is to instead use a forgetting factor $0 < \gamma \leq 1$ to retain the information of the preceding residual values [8]. The measurement noise covariance estimation using the forgetting factor is calculated as

$$\boldsymbol{R}'_k = (1-\gamma)\boldsymbol{R}_k + \gamma(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T + \boldsymbol{H}_k\boldsymbol{P}_{k|k}\boldsymbol{H}_k^T) \tag{3.46}$$

The forgetting factor γ will diminish the effect of preceding residuals over time, where the more recent residuals holds greater weight. A large γ causes less fluctuation of R_k over time but it also means that it will adapt slower to changes.

There are two alternative approaches on how to utilize the estimated measurement noise covariance $\boldsymbol{R}'_k$ [36]. One alternative is to just pass the updated $\boldsymbol{R}'_k$ as $\boldsymbol{R}_{k+1}$ and utilize it in the next iteration of the filter. The other alternative is to first perform an adjustment of the filter update based on the new covariance at the current time step k and then passing $\boldsymbol{R}'_k$ as $\boldsymbol{R}_{k+1}$. The adjustment is achieved by treating the acquired state $\hat{\boldsymbol{x}}_{k|k}$ as the predicted state at time k, using the newly estimated $\boldsymbol{R}'_k$, and re-applying the filter update.

3.4 Coordinate Systems

There are several coordinate systems to consider in self-localization of a ground moving vehicle. The measurements of a sensor is provided from the perspective of the sensor's location and orientation, this is referred to as the sensor frame. Fusing data of several different sensors requires transforming each sensor frame into a single joint frame that originates from the object of interest; this is referred to as the body frame of the host vehicle. Furthermore, some measurements require additional transform, e.g., between the body frame and a fixed local Cartesian coordinate system, the navigation frame. The navigation frame describes the position and orientation of objects with respect to a Cartesian coordinate system that originates from a fixed point. Subsections 3.4.1 - 3.4.4 describe each of the relevant coordinate systems and Subsection 3.4.5 derives the necessary transforms between coordinate systems.

3.4.1 Geodetic Coordinate System

The GPS receiver utilized in the thesis provides positional measurements in geodetic coordinates. The coordinates are defined as the geodetic longitude λ, the geodetic latitude φ, and the geodetic height h. The spherical coordinates of the geodetic coordinate system provide a global position on the Earth's surface by approximating the Earth as a smooth ellipsoid. The longitude is measured with respect to the prime meridian and ranges from $-180°$ to $180°$, the latitude with respect to the equator and ranges of $-90°$ to $90°$, and the geodetic height is the distance measured orthogonally from the ellipsoid's surface [37]. The dimensions of the reference ellipsoid can be defined in a number of ways, depending on application. The most commonly used reference ellipsoid for the GPS is the WGS-84 [38]. The geodetic coordinate system is illustrated in orange in Figure 3.5.

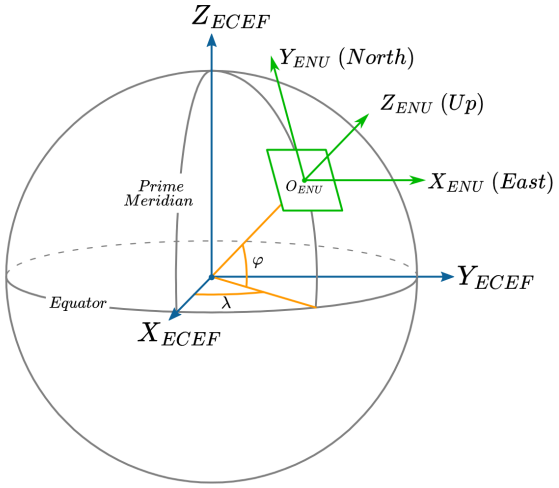


Figure 3.5: The Geodetic coordinate system (λ, φ), the Earth-Centered Earth-Fixed coordinate system (ECEF), and the East-North-Up (ENU) coordinate system.

3.4.2 Earth-Centered Earth-Fixed Coordinate System

An alternative Cartesian representation of the spherical coordinates of the geodetic coordinate system is the Earth-Centered Earth-Fixed (ECEF) coordinate system. The ECEF coordinate system is fixed to the earth, meaning that it follows the Earth's rotation around its spin axis. The coordinate system originates from Earth's center of mass and is defined by the Z-axis intersecting the north pole, the X-axis intersecting 0° longitude and 0° latitude of the geodetic coordinate system, and the Y-axis follows the right-hand rule pointing orthogonally from the Z- and X-axis [37]. The ECEF coordinate system axes are illustrated in blue in Figure 3.5.

3.4.3 Local East-North-Up Coordinate System

The local East-North-Up (ENU) coordinate system is also known as the navigation frame, or coordinate system, and it originates from an arbitrary fixed point on the Earth's surface. The Y-axis points towards the north pole, the X-axis towards East, and the Z-axis points upwards. In ground-moving vehicle applications, the origin of the navigation coordinate system can be selected as the initial starting point of the vehicle. The local ENU coordinate system is defined as a local plane tangent to the Earth's surface, the coordinate system is illustrated in green in Figure 3.5.

3.4.4 Body Coordinate System

The body coordinate system describes the position, orientation, and motion of an object or vehicle relative to itself. The coordinate system is fixed to the host object and originates from a selected point on the vehicle, such as the center of gravity. The X-axis points in the forwards direction of the vehicle, the Y-axis to port, and the Z-axis upwards. In self-localization, the body coordinate system carries measurements directly related to the ego vehicle such as wheel speed, steering angle, the velocity

and acceleration vectors. An illustration of the relations between the Local ENU and the Body coordinate systems are provided in Figure 3.6. Furthermore, the sensor coordinate system has the same characteristics as the body coordinate system and is exemplified by a camera in the figure.

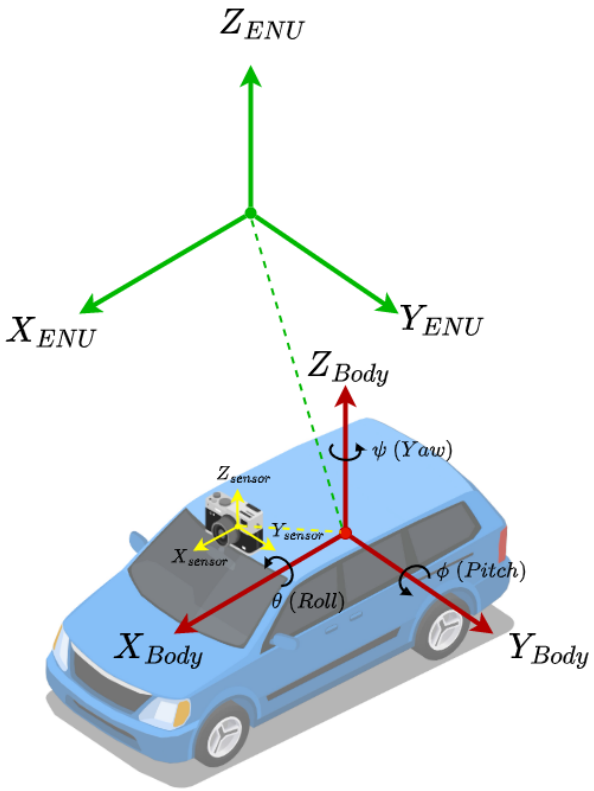


Figure 3.6: Illustration of the Local ENU, body, and sensor coordinate systems that are regarded in vehicle self-localization. Figure credit: [39]

3.4.5 Coordinate Transforms

There are two types of transforms that are of interest to the thesis; the sensor to body transform which aligns the sensor frame measurements to the body frame, and the Geodetic to ENU transform that coverts GPS coordinates to positions relative to a local reference point.

3.4.5.1 Sensor to Body Coordinate Transform

As aforementioned, all sensor measurements must be translated and rotated such that the measurements are provided with respect to the origin of the body coordinate system. Figure 3.6 illustrates a camera sensor where the measurements must be translated to coincide with the body coordinate system to allow for sensor fusion. It is possible to find the required transforms by manually measuring the placement of each sensor on the vehicle. The rotations are based on Euler angles and the

single-axis rotations for each axis is given as

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \; R_y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}, \\ R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.47}$$

To perform a full three-axis rotation we need the yaw, pitch, and roll angles and a rotation order, e.g. xyz; this would yield the rotation matrix $R_{sensor\to body}(\theta, \phi, \psi) = R_x(\theta)R_y(\phi)R_z(\psi)$.

3.4.5.2 Geodetic to ENU Coordinate Transform

In order to use the geodetic coordinates provided by the GPS for the task of self-localization, the geodetic coordinates need to be converted to a local Cartesian coordinate system. The conversion is achieved by projecting the global coordinates to a local plane tangent to the Earth's surface, fixed around a chosen reference point. Any projection of global spherical coordinates onto a local plane will result in distortions over large distances, but such distortions are usually negligible for applications that consider slow-moving vehicles. The local Cartesian coordinate system utilized in the thesis is the ENU coordinate system described in Section 3.4.3, which is also denoted as the navigation frame. To achieve the conversion to the local ENU coordinate system, the geodetic coordinates are first converted to the ECEF coordinate system as an intermediate step by [38]

$$\begin{aligned} x_{ECEF} &= (N + h)\cos\varphi\cos\lambda \\ y_{ECEF} &= (N + h)\cos\varphi\sin\lambda \\ z_{ECEF} &= \left(\frac{b^2}{a^2}N + h\right)\sin\varphi \\ N &= \frac{a^2}{\sqrt{a^2\cos^2\varphi + b^2\sin^2\varphi}} \end{aligned} \tag{3.48}$$

Where a and b are the semiaxes of the chosen ellipsoid. For WGS-84

$$a = 6378137.0 \text{ m}, \quad b = 6356752.31425 \text{ m} \tag{3.49}$$

In order to convert coordinates of the ECEF coordinate systems to a point in the local ENU coordinate system, the following translation and rotation is applied

$$\begin{bmatrix} x_{ENU} \\ y_{ENU} \\ z_{ENU} \end{bmatrix} = \begin{bmatrix} -\sin\lambda_0 & \cos\lambda_0 & 0 \\ -\sin\varphi_0\cos\lambda_0 & -\sin\varphi_0\sin\lambda_0 & \cos\varphi_0 \\ \cos\varphi_0\cos\lambda_0 & \cos\varphi_0\sin\lambda_0 & \sin\varphi_0 \end{bmatrix} \begin{bmatrix} x_{ECEF} - x_{0,ECEF} \\ y_{ECEF} - y_{0,ECEF} \\ z_{ECEF} - z_{0,ECEF} \end{bmatrix} \tag{3.50}$$

Where $\lambda_0, \varphi_0, h_0$ is the geodetic coordinates for the desired origin of the local ENU coordinate system and $(x_{0,ECEF}, y_{0,ECEF}, z_{0,ECEF})$ is the position of the origin in the ECEF coordinate system.

4

Design of the Self-Localization Algorithm

This chapter presents the modeling of the research vehicle and the onboard sensors introduced in Chapter 2, and covers the developed algorithm for robust self-localization, building upon the methods presented in Chapter 3. The beginning of the chapter considers the mathematical modeling of the research vehicle through a kinematic bicycle model, which constitutes the process model of the state estimator. The following section defines the measurement equations for each sensor mounted on the research vehicle that together builds the filter measurement model. The last section of the chapter presents the complete self-localization algorithm, how each of the components are combined to form a self-localization algorithm that is resilient to faulty sensor measurements.

4.1 Vehicle Modeling

Defining a model of the vehicle is fundamental to sensor fusion as it provides a mathematical representation of the vehicle's expected behaviour. Vehicle modeling involves describing the system dynamics using mathematical formulations, and the resulting model serves as the process model in Kalman filtering, see Section 3.1.2. When modeling the dynamic behavior of a ground moving vehicle, two approaches are commonly used: kinematic and dynamic motion models. In this thesis, a kinematic model is adopted due to the slow-moving nature of the intended application. The kinematic model assumes a lumped mass at the center of gravity, no tire slippage, and does not consider any lateral forces acting on the vehicle. Furthermore, slow-moving and stop-and-go scenarios can cause mathematical singularity issues when modeling tire forces in dynamic models [40].

As stated in Chapter 2, the research vehicle is a motorized three-wheeled rickshaw; it has two fixed rear wheels and a single front steering wheel. A bicycle model can be applied to any two-axle vehicle by approximating each wheeled axle as a single center-placed wheel. A fitting way to model the tricycle system is therefore to approximate the rear axle as a single fixed wheel and utilize the Kinematic bicycle model which is illustrated in Figure 4.1. It should be noted that the motion of the kinematic bicycle model is approximated to a 2D plane, meaning that pitch and roll are assumed to be zero.

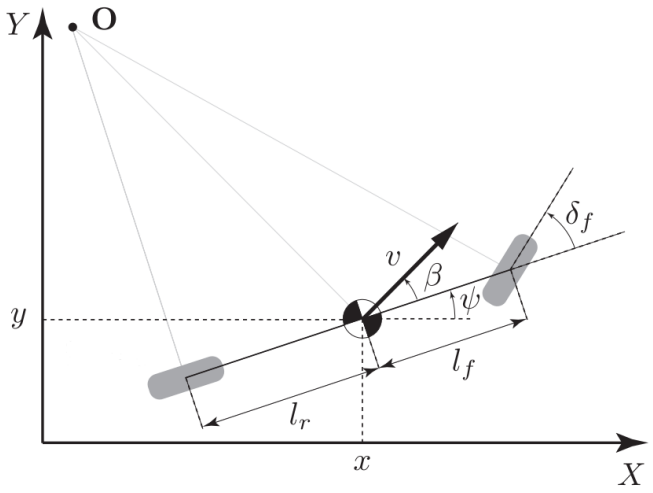


Figure 4.1: The kinematic bicycle model utilized as process model in the self-localization algorithm. Figure credit: [40] CC4.0.

The non-linear equations that describes the kinematic bicycle model in continuous time are

$$\dot{x} = v\cos(\psi + \beta) \tag{4.1}$$

$$\dot{y} = v\sin(\psi + \beta) \tag{4.2}$$

$$\dot{\psi} = \frac{v}{l_r + l_f}\tan(\delta_f)\cos(\beta) \tag{4.3}$$

$$\dot{v} = a \tag{4.4}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta_f)\right) \tag{4.5}$$

Where

- (x, y) are the coordinates of the chosen reference point in the (X, Y) navigation frame.
- v is the vehicle speed.
- a is the vehicle acceleration.
- ψ is the heading angle in the navigation frame.
- β is the angle of the velocity with respect to the longitudinal axis of the vehicle.
- δ_f is the steering angle of the front wheel.
- **O** is the instantaneous center of rotation.
- l_r and l_f are the rear- and front axles respective distance from the reference point along the longitudinal axis of the vehicle.

The reference point in Figure 4.1 and for the Equations (4.1) - (4.5) is placed at the bicycle's center of gravity. The center of gravity of the research vehicle is unknown

but assumed to lie close to the rear axle since the electric motors of the rear wheels make up a large part of the vehicle's weight. By placing the reference point at the rear axle, the equations of motion can be further simplified as the velocity is applied in the direction of the fixed rear tire, such that $\beta = 0$, $l_r = 0$, and $l_f = L$. Furthermore, as the IMU and the GPS receivers are mounted above the rear axle they will not require any translation to fit the body coordinate system of the vehicle. The mathematical representation of the rear axle kinematic bicycle model in continuous time is given by

$$\dot{x} = v\cos(\psi) \tag{4.6}$$
$$\dot{y} = v\sin(\psi) \tag{4.7}$$
$$\dot{\psi} = \frac{v}{L}\tan(\delta) \tag{4.8}$$
$$\dot{v} = a \tag{4.9}$$

Where the subscript of the front steering angle δ_f have been dropped since the back wheel is fixed. The continuous system equations can be discretized using the backward Euler method, $\dot{x} \approx \frac{x_k - x_{k-1}}{\Delta t}$, this yields the state equations of the discrete rear axle kinematic bicycle model

$$\begin{bmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \psi_k \\ \delta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1}\cos(\psi_{k-1})\Delta t \\ y_{k-1} + v_{k-1}\sin(\psi_{k-1})\Delta t \\ v_{k-1} + a_{k-1}\Delta t \\ a_{k-1} \\ \psi_{k-1} + v_{k-1}\frac{\tan(\delta_{k-1})}{L}\Delta t \\ \delta_{k-1} \end{bmatrix} \tag{4.10}$$

Where the position x_k, y_k, ψ_k are defined in the navigation coordinate system described in Section 3.4.3, while v_k, a_k, δ_k are given in the body coordinate system from Section 3.4.4.

4.2 Measurement Modeling

The measurement model is crucial to sensor fusion because it enables the integration of multiple sensor measurements and accounts for the noise characteristics of each measurement. A measurement model is a set of equations which define the mathematical relation between the state vector of the process model and the incoming measurements. This section provides the measurement models for each of the sensors of the research vehicle, see Section 2.1, and the necessary pre-processing of each sensor measurement. All measurements are assumed to be affected by an individual additive white Gaussian noise, denoted by r_k in the measurement equations. Figure 4.2 shows the different sensors and what type of information they provide to the self-localization algorithm. A simple illustration showing the sensor locations on the research vehicle is provided in Figure 4.3.

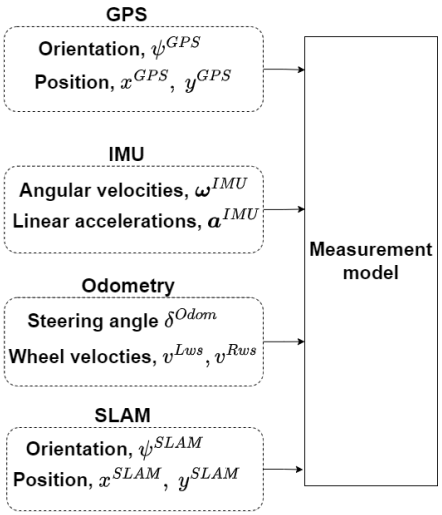


Figure 4.2: Types of information each sensor provides to the state estimator.

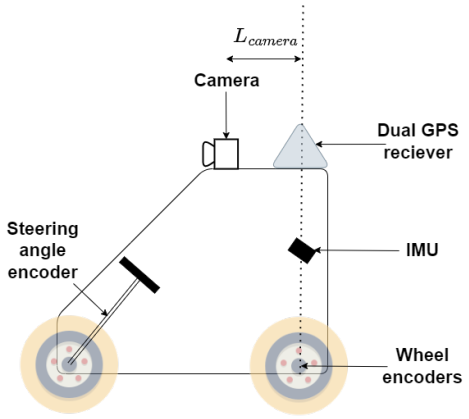


Figure 4.3: Simplified overview of the sensors and their locations on the research vehicle.

4.2.1 GPS Measurement Model

After the incoming geodetic coordinates from the GPS have been converted to local Cartesian coordinates as described in Section 3.4.5, they can can be fused directly in the filter such that

$$x_k^{GPS} = x_k + r_k^{x,GPS} \tag{4.11}$$

$$y_k^{GPS} = y_k + r_k^{y,GPS} \tag{4.12}$$

The same relation can be seen between the velocity of the vehicle and the velocity provided from the GPS system

$$v_k^{GPS} = v_k + r_k^{v,GPS} \tag{4.13}$$

The GPS provides a heading that is positive in the counter-clockwise direction around the z-axis of the navigation coordinate system with with zero at true north, the y-axis in the navigation coordinate system. As simple conversion is therefore performed before the heading value is utilized in the filter, such that $\psi_k^{GPS} \leftarrow -\psi_k^{GPS} + \frac{\pi}{2}$. The measurement equation of the GPS heading is defined as

$$\psi_k^{GPS} = \psi_k + r_k^{\psi,GPS} \tag{4.14}$$

which gives a linear relation between the heading measurements and the estimated heading of the vehicle.

4.2.2 IMU Measurement Models

The IMU, containing the accelerometers and gyroscopes, is located at angles θ_{IMU}, ϕ_{IMU}, ψ_{IMU} from the body coordinate system presented in Section 3.4.4. In order

to fuse the measurement values received from the sensors, they firstly need to be rotated to the body coordinate system. The rotation is achieved by utilizing the rotational transformations presented in Section 3.4.5, such that

$$R_{Sensor \to Body} = R_x(\theta_{IMU}) R_y(\phi_{IMU}) R_z(\psi_{IMU}) \tag{4.15}$$

We can note from Figure 4.2 that the IMU is mounted above the rear-axle of the vehicle and therefore only require rotation to coincide with the body coordinate system. The resulting rotational matrix of Equation (4.15) is applied to the sensor measurements of the accelerometer and gyroscope as follows

$$\boldsymbol{\omega}^{IMU,Body} = R_{Sensor \to Body} \boldsymbol{\omega}^{IMU,Sensor} \tag{4.16}$$

$$\boldsymbol{a}^{IMU,Body} = R_{Sensor \to Body} \boldsymbol{a}^{IMU,Sensor} \tag{4.17}$$

where $\boldsymbol{\omega}^{IMU,Sensor}$ is the angular velocities of the gyroscope and $\boldsymbol{a}^{IMU,Sensor}$ the linear accelerations of the accelerometer in the sensor coordinate system originating from the IMU.

4.2.2.1 Gyroscope Measurement Model

The relation between the rotated angular rate of the gyroscope $\boldsymbol{\omega}^{IMU,Body}$ and the Euler angles of the body coordinate system $\dot{\phi}, \dot{\theta}, \dot{\psi}$ need to be regarded in order for the values to be correctly applied in the filter. The necessary transform between the Euler angles and gyroscope measurements are given by

$$\begin{aligned} \boldsymbol{\omega}^{IMU,Body} = \begin{bmatrix} \omega^{IMU,x} \\ \omega^{IMU,y} \\ \omega^{IMU,z} \end{bmatrix} &= \begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + R_x(\theta) \begin{bmatrix} 0 \\ \dot{\phi} \\ 0 \end{bmatrix} + R_x(\theta) R_y(\phi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & \sin(\phi) \\ 0 & \cos(\theta) & -\sin(\theta)\cos(\phi) \\ 0 & \sin(\theta) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{4.18}$$

However, since the kinematic bicycle model assumes that the vehicle navigates a two-dimensional plane, the relationship is a simple one-to-one mapping since $\phi = \theta = 0$

$$\Rightarrow \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \omega^{IMU,x} \\ \omega^{IMU,y} \\ \omega^{IMU,z} \end{bmatrix} \tag{4.19}$$

Furthermore, the gyroscope values have an additive and slowly varying bias, denoted b^{gyro}. The characteristics of the biases of the research vehicle's IMU is provided in Table 2.2. The gyroscope bias is incorporated in the state vector and estimated alongside the other system states. By estimating the gyroscope bias during filtering the impact is mitigated, resulting in the following measurement equation for the gyroscope

$$\omega_k^{IMU,z} = \dot{\psi}_k + b_k^{gyro} + r_k^{gyro} = \frac{v_k}{L} \tan \delta_k + b_k^{gyro} + r_k^{gyro} \tag{4.20}$$

where only the z-component of the gyroscope measurements are utilized because of the aforementioned assumptions in the kinematic bicycle model.

4.2.2.2 Accelerometer Measurement model

The kinematic bicycle model assumes that there is no sideways motion for the back-wheels of the vehicle. Since the accelerometer is located in the origin of the body coordinate system, we only fuse the x-component of the rotated accelerometer values, $a_k^{IMU,x}$. Similarly to the gyroscope values, the accelerometer values are affected by a slowly varying bias that can be estimated by the filter, denoted as b^{acc}, resulting in the following accelerometer measurement equation

$$a_k^{IMU,x} = a_k + b_k^{acc} + r_k^{acc} \tag{4.21}$$

4.2.3 Wheel Encoder Measurement Models

As described in Section 2.2.3, the wheel encoders measure the number of full revolutions per minute for each back wheel, denoted Ω^{RPM}. The notation for the rear wheels are Rear Right (RR) and Rear Left (RL). From these sensor readings, the wheel hub linear velocities over the ground, v_k^{RL} and v_k^{RR}, can be calculated based one the radius of the wheels, R_{wheel}, such that

$$v_k^{RL} = \frac{\Omega_k^{RPM,RL}}{60}(2\pi R_{wheel}) \tag{4.22}$$

$$v_k^{RR} = \frac{\Omega_k^{RPM,RR}}{60}(2\pi R_{wheel}) \tag{4.23}$$

Although the vehicle model approximates the vehicle as having a single rear wheel, the wheel encoders provide the individual velocity of each rear wheel. Because of their location on the vehicle, the resulting individual velocities will not only be dependent on the forward velocity of the vehicle, but also its turning rate. A simple solution is to average the rear wheel speed of the vehicle, thus approximating a single rear wheel. Averaging the wheel speed will however, ignore the heading rate information that can be computed from individual rear wheel velocity. Furthermore, in order to effectively monitor the quality of measurements for each individual sensor and identify any potential faults, it is beneficial to separate each wheel within the measurement model. Hence, the individual wheel speeds are approximated as a function of the linear velocity v_k and the heading rate $\dot{\psi}_k$ of the vehicle

$$v_k^{RL} = v_k - \dot{\psi}_k l_w + r_k^L = v_k - \frac{v_k}{L}\tan(\delta_k) l_w + r_k^L \tag{4.24}$$

$$v_k^{RR} = v_k + \dot{\psi}_k l_w + r_k^L = v_k + \frac{v_k}{L}\tan(\delta_k) l_w + r_k^R \tag{4.25}$$

Where l_w is the lateral distance between the center of the car and the wheels, and L is the wheelbase, the distance between the front and rear axles. The radius of each wheel have been manually measured. However, the wheel radius is an approximation and may slowly vary over time depending on wear of the tires and change in air pressure. Therefore, a correction factor, c_k^{RR}, c_k^{RL} is added for each wheel radius and is estimated during filtering by expanding the state vector, in the same way as the gyroscope and accelerometer bias, such that the measurement equations for the

wheel velocities becomes

$$v_k^{RL} = c_k^{RL}(v_k - \frac{v_k}{L}\tan(\delta_k)l_w) + r_k^{RL} \tag{4.26}$$

$$v_k^{RR} = c_k^{RR}(v_k + \frac{v_k}{L}\tan(\delta_k)l_w) + r_k^{RR} \tag{4.27}$$

meaning that any change in wheel radius will be corrected during filtering.

4.2.4 Steering Angle Measurement model

A steering angle ratio, η, is used to describe the relation between the angle of the steering wheel and the angle of the front wheel of the vehicle. The steering angle measurements are therefore represented by the measurement equation

$$\delta_k^{odom} = \eta\delta_k + r_k^{odom} \tag{4.28}$$

By manually measuring the relation on the test vehicle, it was deemed to be a 1:1 ratio such that $\eta = 1$.

4.2.5 SLAM Measurement Model

The camera is mounted above the front of the vehicle, at a distance L_{camera} from the rear axle, see Figure 4.3. The positional and heading estimate given by the SLAM module in each time-instance is given in a local coordinate frame with its origin in the initial camera position. The origin of this coordinate system will be translated and rotated compared to the navigational frame of reference. Therefore, the SLAM positions and heading will have to be transformed before they can be utilized in the filter. Firstly the heading is corrected by adding the difference between the initial heading from the GPS system and the current SLAM heading, this assumes that the GPS measurements are available and reliable at the initialization phase of the vehicle

$$\psi_k^{SLAM} \leftarrow \psi_k^{SLAM} + \psi_0 \tag{4.29}$$

Where ψ_0 is the initial difference between the GPS heading and the SLAM heading when the first SLAM reading is received. The process of transforming the positional estimates can be described in two steps, first the positions are transformed to the navigational frame of reference. Secondly, the SLAM positions are translated from the front of the vehicle to the rear axle, by subtracting the corresponding distance dependent on the current heading of the vehicle.

$$\begin{bmatrix} x_k^{SLAM} \\ y_k^{SLAM} \end{bmatrix} = \begin{bmatrix} \cos\psi_0 & -\sin\psi_0 \\ \sin\psi_0 & \cos\psi_0 \end{bmatrix} \begin{bmatrix} x_k^{SLAM} + L_{camera} \\ y_k^{SLAM} \end{bmatrix} - \begin{bmatrix} \cos\psi_k^{SLAM} & -\sin\psi_k^{SLAM} \\ \sin\psi_k^{SLAM} & \cos\psi_k^{SLAM} \end{bmatrix} \begin{bmatrix} L_{camera} \\ 0 \end{bmatrix} \tag{4.30}$$

After the measurements have been transformed, they can be utilized in the filter through the measurement equations

$$x_k^{SLAM} = x_k + r_k^{x,SLAM} \tag{4.31}$$

$$y_k^{SLAM} = y_k + r_k^{y,SLAM} \tag{4.32}$$

$$\psi_k^{SLAM} = \psi_k + r_k^{\psi,SLAM} \tag{4.33}$$

which gives a direct relation between the SLAM measurements and the estimated states.

4.2.6 Complete Process and Measurement Models

The derived process model from equation (4.10) in Section 4.1 is expanded with the additive biases and wheel scale correction factors introduced in the derivation of the measurement equations. The complete process model utilized by the self-localization algorithm is therefore defined as

$$\boldsymbol{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \psi_k \\ \delta_k \\ b_k^{gyro} \\ b_k^{acc} \\ c_k^{RL} \\ c_k^{RR} \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1}\Delta t \cos\psi_{k-1} \\ y_{k-1} + v_{k-1}\Delta t \sin\psi_{k-1} \\ v_{k-1} + a_{k-1}\Delta t \\ a_{k-1} \\ \psi_{k-1} + v_{k-1}\Delta t \dfrac{\tan\delta_{k-1}}{L} \\ \delta_{k-1} \\ b_{k-1}^{gyro} \\ b_{k-1}^{acc} \\ c_{k-1}^{RL} \\ c_{k-1}^{RR} \end{bmatrix} + \boldsymbol{q}_{k-1} \tag{4.34}$$

where $\boldsymbol{q}_{k-1} \sim \mathcal{N}(0, \boldsymbol{Q}_{k-1})$. The measurement model that is used in the state estimator at each time-step is dependent on the availability of sensor measurements at time k. If a sensor is passing a message at time k, the measurement equations of that sensor is concatenated to the complete measurement model of the state estimator. The measurement model at full sensor availability is provided by

$$\boldsymbol{y}_k = \begin{bmatrix} x_k^{GPS} \\ y_k^{GPS} \\ v_k^{GPS} \\ \psi_k^{GPS} \\ a_k^{IMU,x} \\ \omega_k^{IMU,z} \\ v_k^{RL,\ Odom} \\ v_k^{RR,\ Odom} \\ \delta_k^{Odom} \\ x_k^{SLAM} \\ y_k^{SLAM} \\ \psi_k^{SLAM} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ v_k \\ \psi_k \\ a_k + b_k^{acc} \\ v_k \dfrac{\tan\delta_k}{L} + b_k^{gyro} \\ c_k^{RL} v_k (1 - \dfrac{l_w \tan\delta_k}{L}) \\ c_k^{RR} v_k (1 + \dfrac{l_w \tan\delta_k}{L}) \\ \delta_k \\ x_k \\ y_k \\ \psi_k \end{bmatrix} + \boldsymbol{r}_k \tag{4.35}$$

where $\boldsymbol{r}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$.

4.3 Self-Localization Algorithm

The state estimator of the self-localization algorithm is based on the Unscented Kalman Filter. The UKF is chosen in favor of the EKF due to its proven superiority at handling local non-linearities [6]. Additionally, the UKF offers the advantage of ease of implementation as it does not require the computation of Jacobians. The way that the UKF propagates sigma points through the process and measurement models enables straightforward integration of additional sensors or changes to the process model, which presents potential benefits for future development. The UKF is implemented with an asynchronous update because the utilized sensors operate at different sampling rates, see Table 4.1. An asynchronous update means that the filter runs the update based on the available sensor measurements at each time step, this is also paramount because the filter must be able to operate during instances of sensor dropout; where a sensor does not deliver a message for extended periods of time. The implemented UKF operates at 100 Hz, faster sensors are therefore restricted to this sampling rate. The tuning and filter settings are provided in Appendix A. The tuning of the process and measurement noise covariance assumes no correlation and are therefore defined as diagonal matrices.

Table 4.1: Measurement frequency of the different sensors.

Message	Frequency
GPS	5 Hz
IMU	100 Hz
Rotary encoders	100 Hz
ORB SLAM3	15 Hz

A flow-chart of the developed self-localization algorithm is presented in Figure 4.4. The conventional UKF algorithm is covered in the theoretical background, Section 3.1.4, and considers the following boxes of the figure; “Filter Prediction”, “Predict Measurements”, and “Filter Update”. There are three additional function blocks in the figure that supplements the UKF algorithm, these are “Fault Detection and Isolation”, “Measurement Covariance Update”, and “Filter Update Correction”. The filter is presented in pseudocode in Algorithm 1 with references to the equations utilized in each block.

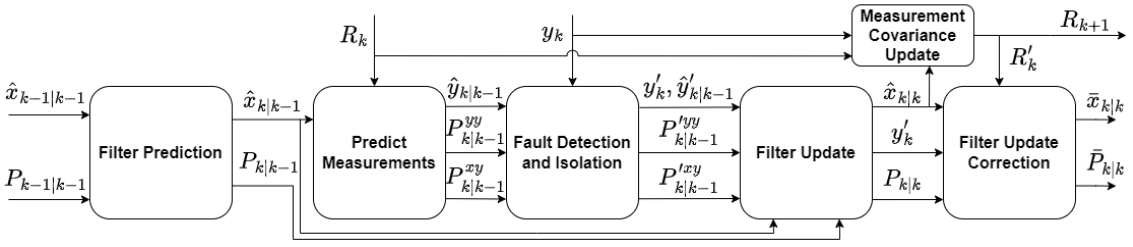


Figure 4.4: Flow-chart of the developed self-localization algorithm, the Adaptive Unscented Kalman Filter with Fault Detection and Isolation.

Algorithm 1 Adaptive Unscented Kalman Filter with Fault Detection and Isolation

Input: $f(\cdot)$, $h(\cdot)$, $\hat{x}_0$, P_0, Q, R_1, α, β, κ

Initialize UKF state weights $W_i^{(m)}$ and covariance weights $W_i^{(c)}$ ▷ Eq. (3.29)
for $k \in \{1, ..., \infty\}$ **do**
  Filter Prediction
  Compute Sigma points $\mathcal{X}_{k-1|k-1}$ for $\hat{x}_{k-1|k-1}, P^{xx}_{k-1|k-1}$ ▷ Eq. (3.30)
  Use $\hat{x}_{k-1}$, P^{xx}_{k-1} to obtain $\hat{x}_{k|k-1}$, $P^{xx}_{k|k-1}$ ▷ Eq. (3.31), (3.32)
  Predict measurements
  Compute Sigma points $\mathcal{X}_{k|k-1}$ for $\hat{x}_{k|k-1}, P^{xx}_{k|k-1}$ ▷ Eq. (3.30)
  Compute $\hat{y}_{k|k-1}$, $P^{xy}_{k|k-1}$, $P^{yy}_{k|k-1}$ ▷ Eq. (3.37), (3.38), (3.39)
  Fault Detection and Isolation
  for each $y_i \in y_k$ **do**
    Perform NIS-test: compute d_i^{NIS} ▷ Eq. (4.36)
    if $d_i^{NIS} < threshold^*$ **then** ▷ *See Table 3.1
      Append inlier measurement y_k^i to y'_k
    end if
  end for
  Extract $P'^{xy}_{k|k-1}, P'^{yy}_{k|k-1}, \hat{y}'_{k|k-1}$ with respect to y'_k
  Filter Update
  Use $y'_k, P'^{xy}_{k|k-1}, P'^{yy}_{k|k-1}, \hat{y}'_{k|k-1}$ to obtain $\hat{x}_{k|k}$, $P^{xx}_{k|k}$ ▷ Eq. (3.33), (3.34), (3.36)
  Measurement Covariance Update
  Compute Sigma points $\mathcal{X}_{k|k}$ for $\hat{x}_{k|k}, P^{xx}_{k|k}$ ▷ Eq. (3.30)
  Compute individual forgetting factor γ_k^i ▷ Eq. (4.37)
  Compute R'_k using the forgetting factor γ_k. ▷ Eq. (4.38) - (4.41)
  Filter Update Correction
  Compute $\bar{x}_{k|k}$, $\bar{P}^{xx}_{k|k}$ using y'_k, R'_k ▷ Eq. (4.42) - (4.47)
  Save to next iteration:
  $\hat{x}_{k|k} \leftarrow \bar{x}_{k|k}$
  $P^{xx}_{k|k} \leftarrow \bar{P}^{xx}_{k|k}$
  $R_{k+1} \leftarrow R'_k$
end for

4.3.1 Fault Detection and Isolation

The Fault Detection and Isolation (FDI) of Figure 4.4 is based on the Mahalanobis distance and the NIS-test as described in Section 3.2. The FDI algorithm isolates

faulty measurements based on a chi-squared distribution threshold with a selected significance level of 0.01. The significance level is a tuning parameter that was selected empirically to provide a well balanced compromise between sensitivity to faulty measurements and the probability of correct measurements being excluded. There is one degree of freedom for all sensor measurements except the GPS and SLAM positions, a position considers two measurements and therefore has two degrees of freedom. The NIS-test is performed on each measurement or measurement pair $\boldsymbol{y}_i \in \boldsymbol{y}_k$ and checked against the threshold, see Table 3.1. The Mahalanobis distance is computed as

$$d_i^{NIS} = (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^T (\boldsymbol{P}_i^{yy})^{-1} (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i) \tag{4.36}$$

Note that the covariance $\boldsymbol{P}_i^{yy}$ considers the diagonal element or block of $\boldsymbol{P}_{k|k-1}^{yy}$ that corresponds to $\boldsymbol{y}_i$. All valid measurements are saved as $\boldsymbol{y}_k'$ and the corresponding information is extracted from the computed covariance matrices, $\boldsymbol{P}_{k|k-1}^{xy}, \boldsymbol{P}_{k|k-1}^{yy}$ to form $\boldsymbol{P}_{k|k-1}^{\prime xy}, \boldsymbol{P}_{k|k-1}^{\prime yy}$. The FDI method thus removes outlier measurements which would likely degrade the state estimate and passes only the valid measurements and corresponding covariance to the filter update.

4.3.2 Adaptive Update of Measurement Noise Covariance

After the FDI and UKF filter update, the next step of the self-localization algorithm is to apply the measurement noise covariance update. The update is applied to all currently available measurements, including those that have been discarded by the FDI, at each iteration of the UKF using the residual-based approach and a forgetting factor γ which was presented in Section 3.3. The thesis has developed a method of utilizing an individual forgetting factor for each sensor measurement. Each measurement-specific forgetting factor γ_k^i is computed as a linear function dependent on the time passed since the measurement was last received t_k^i. The function for calculating the each sensors forgetting factor is

$$\gamma_k^i = \begin{cases} 0.5 t_k^i & \text{if } 0.5 t_k^i < 0.20 \\ 0.20 & \text{otherwise} \end{cases} \tag{4.37}$$

Where both the upper limit and the gradient has been selected empirically to yield a covariance update that quickly reacts to changes in the measurement noise, while not being too heavily influenced by individual measurements. Using an individual and varying forgetting factor for each sensor was found to be necessary because of the asynchronous update, where a global and constant γ diminished the trust in less frequent sensors. Additionally, if a sensor is unavailable for a long period of time it is beneficial to place a larger weight on current measurement characteristics which is possible by correlating the forgetting factor with the time passed since the measurement was last received. The measurement noise covariance update using the forgetting factor for an UKF is adopted from Equations (3.43) - (3.46), the resulting

equations becomes

$$\boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k}) \tag{4.38}$$

$$\hat{\boldsymbol{y}}_{k|k} = \sum_{i=0}^{2n} \boldsymbol{h}(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}) W_i^{(m)} \tag{4.39}$$

$$\boldsymbol{S}_{k|k}^{yy} = \sum_{i=0}^{2n} \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}) - \hat{\boldsymbol{y}}_{k|k}\right) \left(\,\cdot\,\right)^T W_i^{(c)} \tag{4.40}$$

$$\boldsymbol{R'}_k = (1-\gamma)\boldsymbol{R}_k + \gamma(\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T + \boldsymbol{S}_{k|k}^{yy}) \tag{4.41}$$

The last step of the self-localization algorithm is to correct the filter update based on the measurements that passed through the FDI and the new measurement noise covariance. The update correction is performed to adjust the state $\hat{\boldsymbol{x}}_{k|k}$ and state covariance $\boldsymbol{P}_{k|k}^{xx}$ in accordance to the new estimated measurement noise covariance $\boldsymbol{R'}_k$. The filter update correction is similar to the UKF filter update but considers the state $\hat{\boldsymbol{x}}_{k|k}$ as the predicted state. Additionally, since the state transition function of an UKF is non-stationary we also need to compute a new the state covariance $\bar{\boldsymbol{P}}_{k|k}^{xx}$, the correction of the filter update is defined as

$$\boldsymbol{P}_{k|k}^{xx} = \boldsymbol{Q}_{k-1} + \sum_{i=0}^{2n} \left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)} - \hat{\boldsymbol{x}}_{k|k}\right) \left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)} - \hat{\boldsymbol{x}}_{k|k}\right)^T W_i^{(c)} \tag{4.42}$$

$$\boldsymbol{P}_{k|k}^{xy} = \sum_{i=0}^{2n} \left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)} - \hat{\boldsymbol{x}}_{k|k}\right) \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}) - \hat{\boldsymbol{y}}'_{k|k}\right)^T W_i^{(c)} \tag{4.43}$$

$$\boldsymbol{P}_{k|k}^{yy} = \boldsymbol{R'}_k + \sum_{i=0}^{2n} \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}) - \hat{\boldsymbol{y}}'_{k|k}\right) \left(\boldsymbol{h}(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}) - \hat{\boldsymbol{y}}'_{k|k}\right)^T W_i^{(c)} \tag{4.44}$$

$$\bar{\boldsymbol{K}}_k = \boldsymbol{P}_{k|k}^{xy} (\boldsymbol{P}_{k|k}^{yy})^{-1} \tag{4.45}$$

$$\bar{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k} + \bar{\boldsymbol{K}}_k \left(\boldsymbol{y}'_k - \hat{\boldsymbol{y}}'_{k|k}\right) \tag{4.46}$$

$$\bar{\boldsymbol{P}}_{k|k}^{xx} = \boldsymbol{P}_{k|k}^{xx} - \bar{\boldsymbol{K}}_k \boldsymbol{P}_{k|k}^{yy} (\bar{\boldsymbol{K}}_k)^T \tag{4.47}$$

where $\bar{\boldsymbol{x}}_{k|k}$ and $\bar{\boldsymbol{P}}_{k|k}^{xx}$ is the corrected state and state covariance that are passed to the next iteration of the algorithm, along with the updated measurement noise covariance $\boldsymbol{R}'_k$.

4.3.3 Pairing FDI and Adaptive Measurement Noise Covariance Update

Together, FDI and the adaptive measurement noise covariance algorithm cooperates to provide resilience to faulty measurements. When a measurement is excluded from the state update by the FDI algorithm, the measurement is still passed to the measurement noise covariance update which allows the system to expect the next measurements at a larger uncertainty. Furthermore, if the system has adapted the measurement noise covariance to expect a larger discrepancy of a measurement, the innovation covariance $\boldsymbol{P}_{k|k-1}^{yy}$ is altered which affects the Mahalanobis distance, see Equations (3.39) and (3.41). This allows the system to utilize any useful information from a degraded measurement in future time steps. On the other hand, a

measurement whose quality improves will in turn reduce the expected measurement noise covariance, thus increasing the trust in the measurement. An example that visualizes how the FDI and adaptive update handle a scenario where the noise affecting the GPS measurements suddenly increases can be seen in Figure 4.5. As the increased noise is added at $t = 61$, the FDI excludes the initial GPS measurements, seen in red. In the following time-instances, the expected noise covariance of the GPS positional measurement is increased by the adaptive update, resulting in the filter incorporating the later measurements but with a severely decreased trust. The decreased trust is visualized by the larger shaded blue region, showing the $\pm 3\sigma$-level for each of the corresponding diagonal elements in R_k representing the expected GPS noise variance.

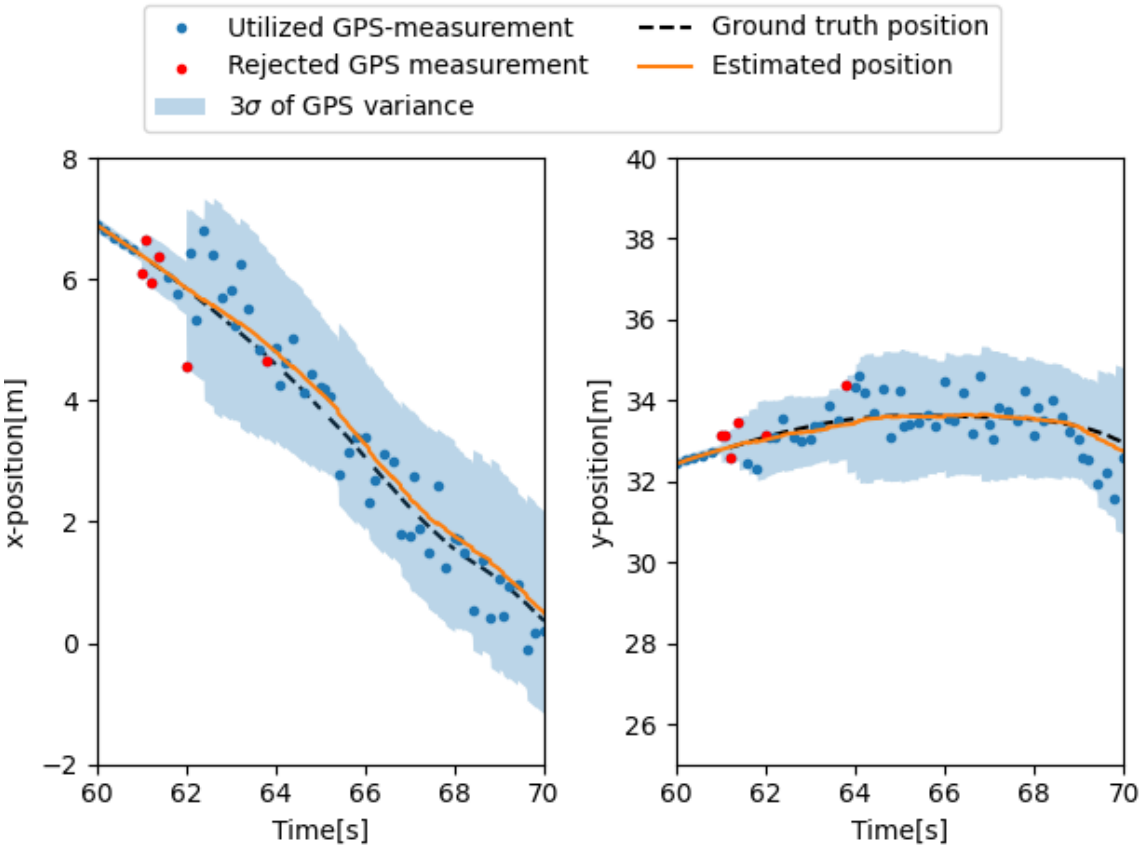


Figure 4.5: Positional estimates and GPS measurement uncertainty over time during a scenario where a white Gaussian noise is added to the GPS position measurements starting at $t = 61$. The plots visualizes how the FDI and adaptive components of the filter reacts to an increased noise in the GPS measurements by excluding the initial measurements and then raising the expected uncertainty in the GPS.

A similar response can be seen for the 2-dimensional case when the GPS sensor suddenly experiences a constant offset over several consecutive time steps, as depicted in Figure 4.6. The initial GPS offsets at $(x, y) \approx (-12.5, 33.0)$ are rejected before the expected uncertainty of the measurements is increased. In the plot, the 2-dimensional 3σ-ellipses of the expected GPS noise covariance are presented by the

dotted blue ellipses of every third received GPS measurement.

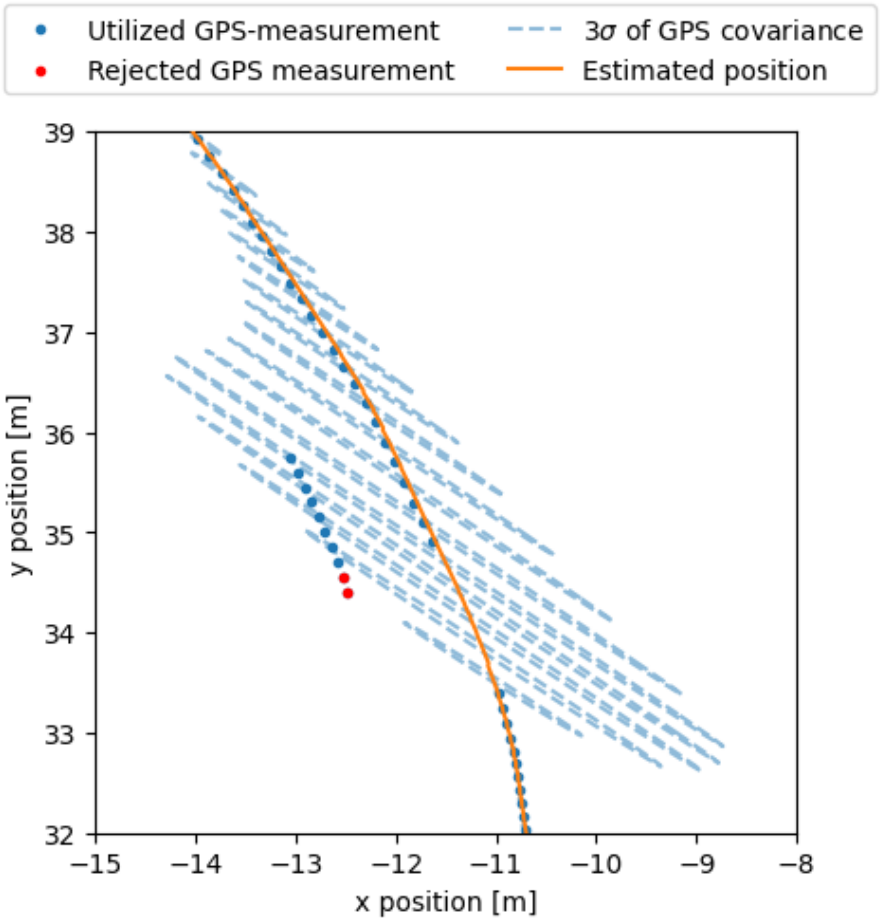


Figure 4.6: 2D positional estimates and GPS measurement uncertainty during a scenario where a sudden offset is added to the GPS positional measurements. The visualization demonstrates how the FDI and adaptive components of the filter reacts to the offset and alters the trust in the GPS; rejecting the initial offset measurements and increasing the expected covariance such that the state estimate is not influenced by the offset.

5

Evaluation of Self-Localization Algorithm

This chapter presents the performance of the developed self-localization algorithm, the Adaptive Unscented Kalman Filter with Fault Detection and Isolation (AUKF-FDI). The AUKF-FDI is compared to subsets of the algorithm, such as a conventional UKF, and also to the currently implemented Extended Kalman Filter of the research vehicle. The chapter begins by describing the test setting and the collected evaluation data. The performance of the state estimators are evaluated on four test scenarios, which mainly considers simulated GPS errors; GPS dropout, varying noise, and multi-path errors. Additionally, a mechanical error is introduced in the form of a loosely mounted IMU, and the research vehicle is rocked sideways a few times during one test. Next, the positional uncertainty of the AUKF-FDI is compared to the actual positional error under selected periods of the test scenarios. The end of the chapter summarizes the obtained results and discusses the robustness of the developed algorithm.

5.1 Test Setting

The evaluation data has been collected at a test track facility, the area considered an open and flat surface of approximately $6000m^2$, see Figure 5.1. As mentioned in Section 1.3, the evaluation utilizes the GPS as ground truth meaning that the true signal is logged throughout the testing scenario and injected with faults in post-processing. The GPS signal is either not provided to the state estimators at selected periods to simulate GPS dropout, or altered before it is passed to the state estimators to simulate varying noise or a multi-path error. The research vehicle was maneuvered in a way that was intended to mirror an urban driving environment that included both straight sections and turns at different radii. The state estimators are evaluated in four scenarios designed to assess their resilience to faulty sensor data. Each test is described in detail in its corresponding subsection, following is a brief description of each test:

- **Scenario 1 - GPS Dropout:** The state estimators are deprived of the GPS signal for a majority of the scenario, this is intended to replicate the vehicle entering a garage or any other GPS-denied environment.
- **Scenario 2 - GPS Noise:** The GPS signal is distorted by a Gaussian noise

applied at separate instances with different variance. The scenario demonstrates poor GPS conditions, such as environments with crowded building or trees.

- **Scenario 3 - GPS Multi-path Error:** The GPS signal is offset from its true position at a few instances. The multi-path error scenario simulates the vehicle driving close to a reflective surface, such as a large building, which would distort the provided GPS position as a mirroring to said surface.
- **Scenario 4 - GPS Dropout, Loosely Mounted and Disturbed IMU:** The GPS signal is removed, the IMU is loosened from its mounting, and the vehicle is rocked at three instances while driving. The scenario is intended to simulate poor road conditions in a GPS denied environment.

Figure 5.1: Birds-eye view of the test track where the research vehicle was driven to collect sensor data used for evaluation of the developed self-localization algorithm.

5.1.1 Slam Accuracy in the Test Environment

During data gathering at the test track it became evident that ORB SLAM3 did not provide a reliable positional estimate. The testing area did not contain a sufficient amount of stationary objects as can be seen in Figure 5.1. The lack of nearby and well defined static objects caused the feature extraction to favor poorly defined features such as the treeline and clouds. Several object were spread out across the test area in an attempt to improve the performance of SLAM; a few parked cars, pallets, cones, and boxes. The objects did improve the feature extraction but the positional estimate still exhibited a varying scaling error due to the dominance of

features far beyond the capacity of the depth camera. The most successful SLAM positional estimate is illustrated in Figure 5.2, note that the data was acquired by manually scaling the position in post-processing which is unrealistic for real-time application. It was deemed infeasible to utilize the current method of SLAM to obtain a positional estimate in an open environment, the positional estimate from SLAM is therefore excluded from the measurement model during the evaluation. The heading estimate however, proved to have a much higher accuracy with minimal drift compared to the referenced GPS heading, as can be seen in Figure 5.2.

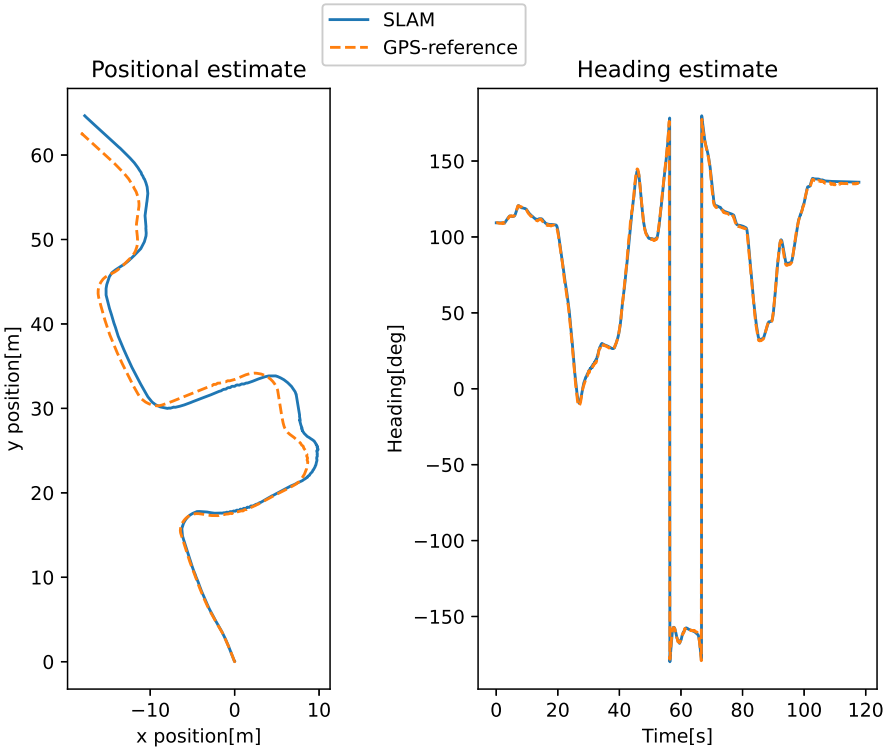


Figure 5.2: The position and heading measurements provided by ORB SLAM3 and the GPS, the sensor data was collected while driving on the test track. The SLAM positional estimate has been manually scaled in post-processing but still exhibits a large accumulative error.

5.2 Evaluation of Developed Self-Localization algorithm

In order to demonstrate the robustness of the developed self-localization algorithm, the algorithm's performance is compared to both the EKF of the research vehicle, and subsets of the developed algorithm. The subsets are evaluated to demonstrate the collective contributions of each component to robust self-localization, and to determine if simplifications of the complete self-localization algorithm are adequate in specific scenarios. The evaluated filters are denoted as follows:

- **EKF**: The currently implemented filter in the research vehicle, a standard EKF as derived in Section 3.1.3.
- **UKF**: A standard UKF, as derived in Section 3.1.4.
- **UKF-FDI**: An UKF with the Fault Detection and Isolation component presented in Section 4.3.1.
- **AUKF**: An UKF with the adaptive update of the measurement noise covariance component presented in Section 4.3.2.
- **AUKF-FDI**: The complete self-localization algorithm as presented in Section 4.3 and Algorithm 1.

The tuning parameters and filter settings of the UKF-based filters are provided in Appendix A, note that the adaptive update will alter the measurement noise covariance matrix during the filtering process for the AUKF and AUKF-FDI, whereas the UKF and UKF-FDI has a set tuning that does not change. In all scenarios, the starting position of the vehicle is at $x = 0$, $y = 0$ in the local East-North-Up coordinate system. Some key plots and metrics are visualized for each scenario described in Section 5.1, these are ass follows:

- The estimated 2D trajectory of the AUKF-FDI is compared to the trajectory of the EKF currently implemented in the test vehicle, and the standard UKF.
- The heading and positional errors between the estimates and the GPS ground truth are plotted over time for the AUKF-FDI, EKF, and UKF.
- For scenarios 2-4, the diagonal elements of the AUKF-FDI's measurement noise covariance matrix is plotted over time for the affected sensors.
- The Root-Mean-Squared Error (RMSE) and maximum error for the 2D positions and heading estimations are presented for all five state estimators.

5.2.1 Scenario 1 - GPS Dropout

Figure 5.3 shows the resulting trajectory of the state estimators as they are deprived of the GPS signal between $t = 45s$ and $t = 120s$. Both the UKF and the AUKF-FDI remain fairly close to the ground truth position while the EKF exhibits an increasing drift over time. Figure 5.4 provides the divergence from the GPS ground truth over time, note that the EKF error is clipped by the y-axis limit to illustrate the error of the better-performing state estimators. The scenario highlights the difference between the EKF currently implemented in the research vehicle, and the UKF-based state estimators. It is evident that the custom modeling of the research vehicle and the tuning of the UKF provides a more accurate state estimate compared to the EKF under GPS dropout.

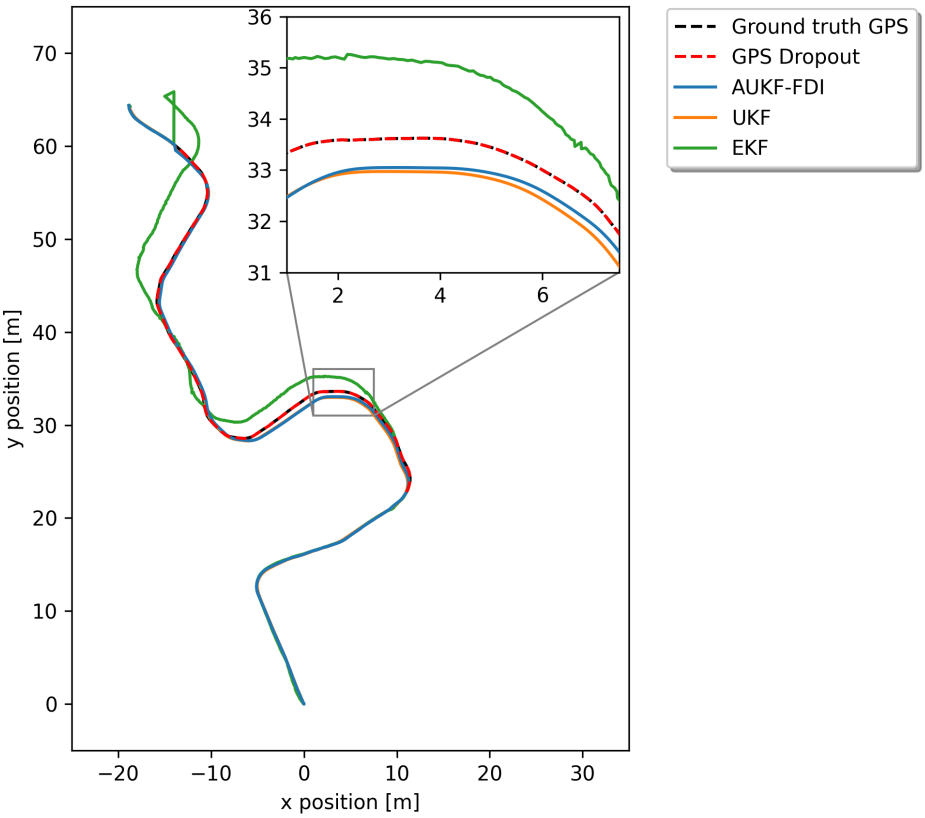


Figure 5.3: Positional estimates of the EKF, UKF, and AUKF-FDI for Scenario 1: GPS dropout. The GPS ground truth is visualized in red during dropout.

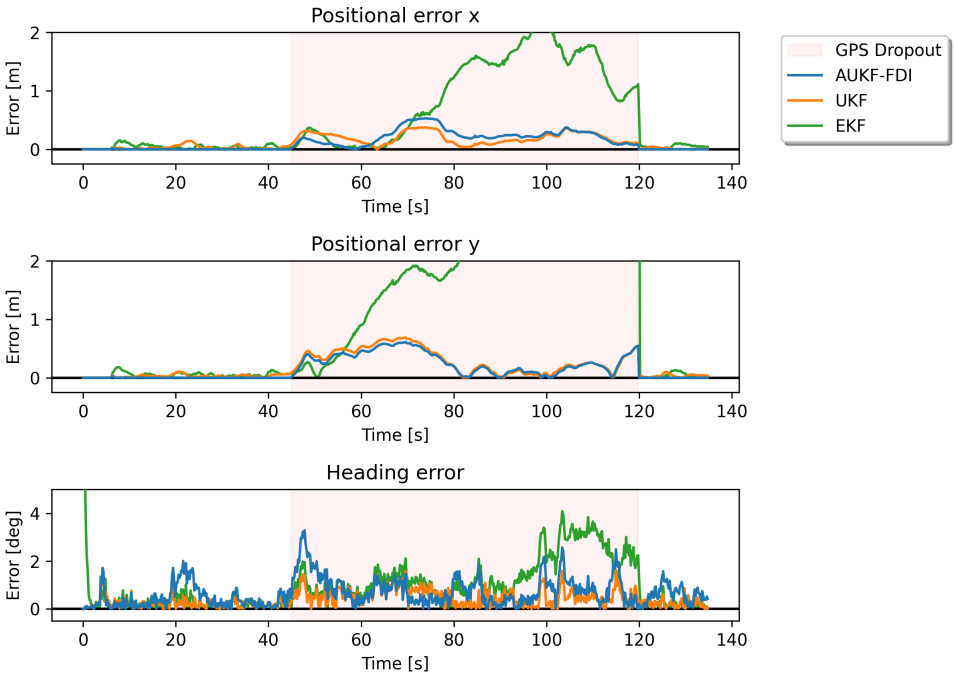


Figure 5.4: Positional and heading errors of the EKF, UKF, and AUKF-FDI for Scenario 1: GPS dropout.

The positional error for all UKF-based state estimators is very similar as can be seen in Table 5.1. The main difference between the UKF-based solutions is the impact of the adaptive measurement noise update which introduces a $\sim 0.40°$ larger RMSE heading error and a $\sim 1.50°$ larger maximum heading error. The larger heading RMSE of the adaptive state estimators is caused by the slight difference between the SLAM heading and the ground truth GPS heading. The SLAM heading exhibits a low degree of noise which means that the adaptive update alters the expected variance such that the system generally trusts the SLAM heading over the other measurements when determining the heading estimate. Additionally, the SLAM heading has a faster sampling rate than the GPS heading which further skews the trust towards the faster sensor when the GPS is available. The aforementioned heading issues are present in all scenarios for all filters with the adaptive measurement noise covariance update component, the issues will be discussed further in Section 5.4.

Table 5.1: Performance of state estimators for Scenario 1: GPS dropout.

	RMSE		Max error	
	Position $[m]$	Heading $[deg]$	Position $[m]$	Heading $[deg]$
EKF	2.3596	1.3808	5.5293	8.9528
UKF	0.3190	0.5284	**0.7623**	**1.6317**
UKF-FDI	0.3232	**0.4858**	0.7850	1.7976
AUKF	0.3147	0.9181	0.7812	3.3573
AUKF-FDI	**0.3120**	0.8874	0.7747	3.2880

5.2.2 Scenario 2 - GPS Noise

The second scenario considers three time-periods where zero-mean Gaussian noise was added to the GPS positional measurements. The noise was applied to the GPS positions at the following time intervals and variances:

- $t \in [20, 45],\ \sigma_x^2 = \sigma_y^2 = 1.00$
- $t \in [62, 82],\ \sigma_x^2 = \sigma_y^2 = 0.25$
- $t \in [97, 117],\ \sigma_x^2 = \sigma_y^2 = 0.5625$

As mentioned in Section 2.2.1, the GPS accuracy is dependent on the surrounding environment. The scenario is intended to exemplify how the GPS position is influenced by a sub-optimal operating environment; where many different and small disturbances or issues with accuracy are approximated as an increased Gaussian noise. We can note from Figures 5.5 and 5.6 that both the EKF and the conventional UKF adhere to the noisy GPS positions which results in their irregular trajectories. The AUKF-FDI exhibits a smoother trajectory and remains close to the ground truth even under the influence of increased noise.

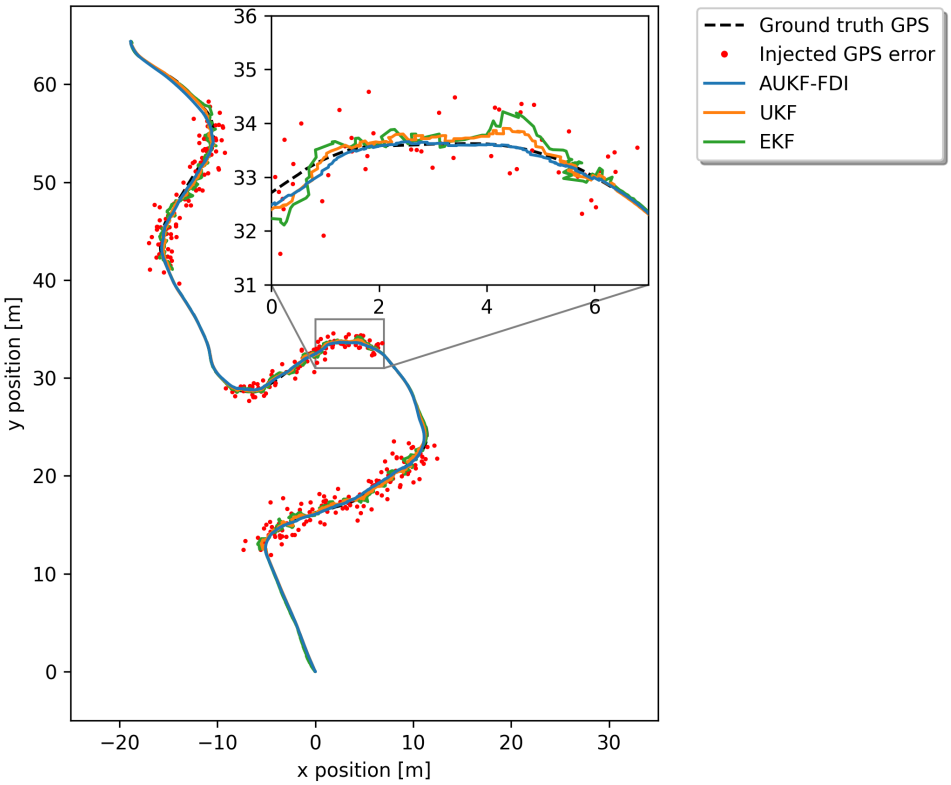


Figure 5.5: Positional estimates of the EKF, UKF, and AUKF-FDI for Scenario 2: injected noise of shifting variance in the GPS position.

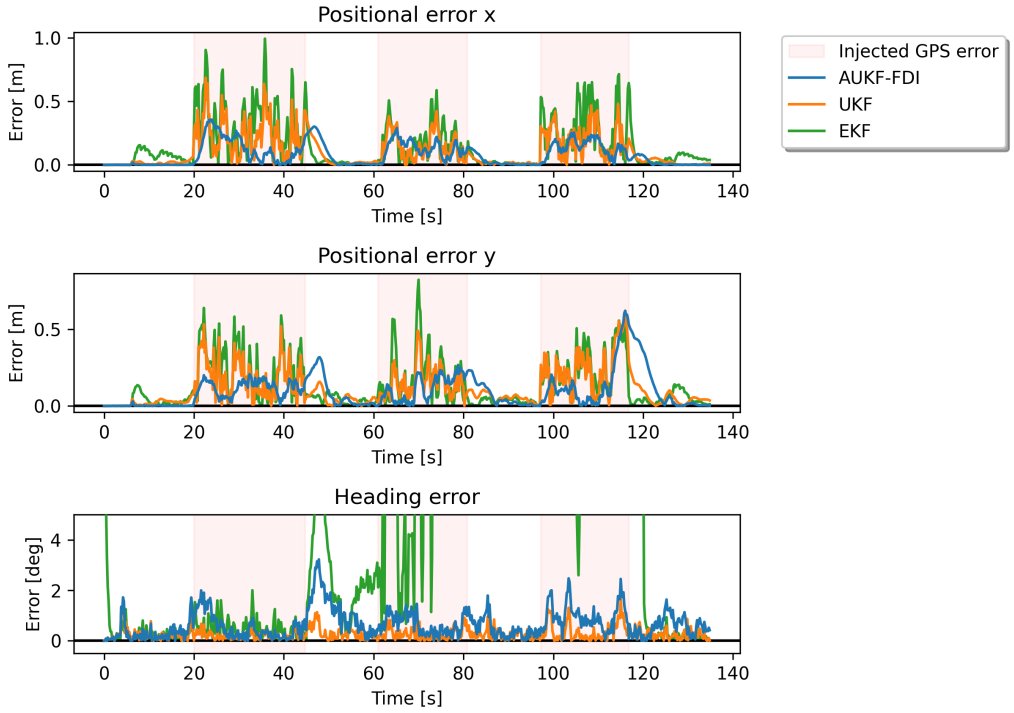


Figure 5.6: Positional and heading errors of the EKF, UKF, and AUKF-FDI for Scenario 2: injected noise of shifting variance in the GPS position.

Figure 5.7 considers the AUKF-FDI, and plots the change in measurement variance over time for each individual GPS measurement. We can note that the system quickly adapts to expect the GPS positions at a larger variance. Furthermore, we can see that the adaptive measurement noise covariance update results in a variance with a mean value close to the applied noise as the initial GPS positional variance is $R_{initial}^{GPS,x} = R_{initial}^{GPS,y} = 0.01$. The benefits of incorporating both the adaptive update and the FDI is evident if we compare the performance of the state estimators provided by Table 5.2, where the AUKF-FDI has the lowest positional RMSE. The AUKF has the largest positional RMSE because the first noisy GPS position will offset the positional estimate and the adaptive component will subsequently diminish the trust in the sensor, causing the error to propagate throughout the entire noisy section. By instead determining that the GPS position is suddenly outside an expected range by also incorporating FDI, the offset is never introduced since the first noisy measurement is excluded, and then the adaptive component causes the system to anticipate the upcoming measurements at a higher variance.

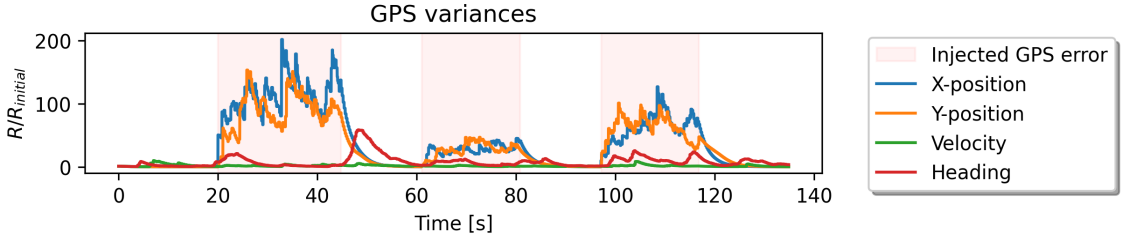


Figure 5.7: Change in expected GPS variance over time as computed by the adaptive component of the AUKF-FDI for Scenario 2: injected noise of shifting variance in the GPS position.

Table 5.2: Performance of the state estimators for Scenario 2: injected noise of shifting variance in the GPS position.

	RMSE		Max error	
	Position $[m]$	Heading $[deg]$	Position $[m]$	Heading $[deg]$
EKF	0.3143	12.1013	0.9974	44.6507
UKF	0.2406	**0.3751**	0.8037	**1.6558**
UKF-FDI	0.3218	0.4099	0.6873	1.8177
AUKF	0.4358	0.8978	1.8071	3.3369
AUKF-FDI	**0.1922**	0.8573	**0.6332**	3.2240

5.2.3 Scenario 3 - GPS Multi-path Error

In the third scenario, an offset to GPS positions has been added to the GPS positional measurements on four occasions to simulate a multi-path error of varying magnitude. The offset error is introduced by adding a fixed distance to the GPS positions under a selected time interval, the offsets are applied as follows:

- $t \in [20, 30]$: $(x^{GPS},\ y^{GPS}) \leftarrow (x^{GPS} + 3.00,\ y^{GPS} - 2.00)$
- $t \in [57, 62]$: $(x^{GPS},\ y^{GPS}) \leftarrow (x^{GPS} + 1.00,\ y^{GPS} + 2.00)$
- $t \in [67, 74]$: $(x^{GPS},\ y^{GPS}) \leftarrow (x^{GPS} + 2.00,\ y^{GPS} - 3.00)$
- $t \in [88, 90]$: $(x^{GPS},\ y^{GPS}) \leftarrow (x^{GPS} - 1.50,\ y^{GPS} + 1.00)$

In this scenario, the AUKF-FDI provides a significantly more accurate positional estimate during the the simulated errors compared to the standard UKF and EKF, as demonstrated by the positional errors in Figure 5.9, and with a positional RMSE that is 87% lower than that of the UKF, see Table 5.2.3. From Figure 5.8, it is evident that the UKF and EKF trusts the GPS measurements during the offsets and quickly converges towards the faulty GPS positions. The AUKF-FDI, however, detects and excludes the initial offset and quickly increases the GPS measurement variances as seen in Figure 5.10, resulting in the filter relying more on the other sensors and producing a more accurate estimate.

By comparing the positional- and heading RMSE for the state estimators in Table 5.2.3, it is clear that the FDI component of the filter provides the most benefits to this scenario, with the UKF-FDI having a positional RMSE similar to the AUKF-FDI. In contrast, the AUKF has the most significant error of all filters. This indicates that there is likely no benefit in still utilizing the offset GPS measurements rather can excluding them completely for this type of error, since the offset will still have a small impact in the update of the filter. We can note from Figure 5.9 that the error of the AUKF-FDI slowly increases over the offset periods because the estimate moves towards the faulty positions as time progresses.

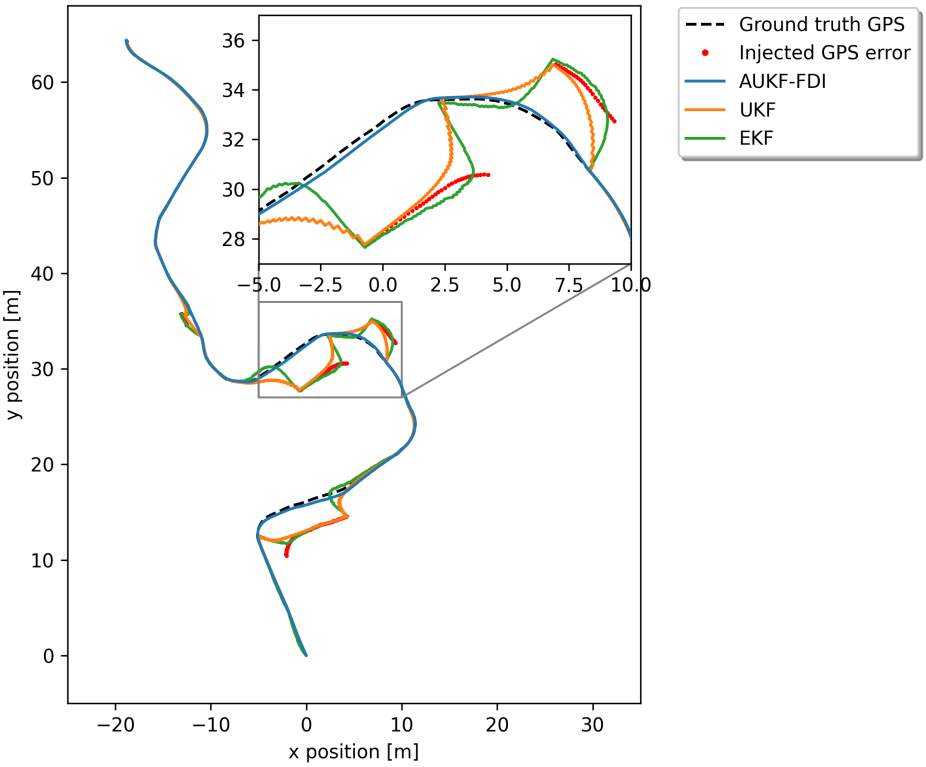


Figure 5.8: Positional estimates of the EKF, UKF, and AUKF-FDI for Scenario 3: simulated multi-path GPS errors.

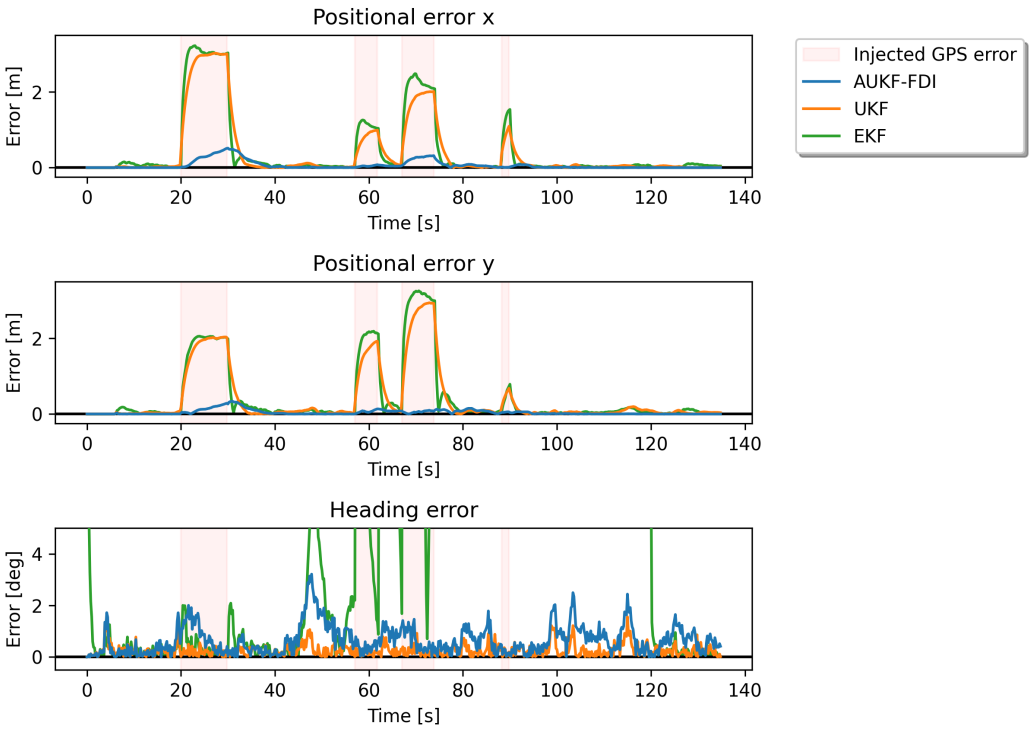


Figure 5.9: Positional and heading errors of the EKF, UKF, and AUKF-FDI for Scenario 3: simulated multi-path GPS errors.

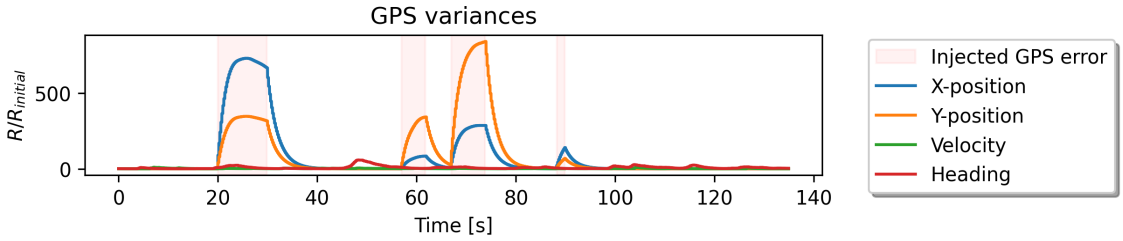


Figure 5.10: Change in expected GPS variance over time as computed by the adaptive component of the AUKF-FDI for Scenario 3: simulated multi-path GPS errors.

Table 5.3: Performance of the state estimators for Scenario 3: simulated multi-path GPS errors.

	RMSE		Max error	
	Position $[m]$	Heading $[deg]$	Position $[m]$	Heading $[deg]$
EKF	1.3815	18.3814	4.0854	84.4976
UKF	1.2455	**0.3535**	3.6329	**1.5594**
UKF-FDI	0.1582	0.3953	**0.4968**	1.7975
AUKF	1.4708	0.8975	3.6169	3.3307
AUKF-FDI	**0.1528**	0.8563	0.5877	3.2128

5.2.4 Scenario 4 - GPS Dropout, Loosely Mounted and Disturbed IMU

In the fourth scenario, the IMU is loosely mounted, and the vehicle have been rocked in its lateral direction three times while the data was collected, at approximately $t = 32$, $t = 44$, and $t = 60$. In addition, a GPS dropout has been simulated by removing the measurements for the time period $t = 42s$ until $t = 118s$. The scenario was intended to evaluate the performance of the AUKF-FDI in the presence of faulty data, not limited to GPS measurements. However, the GPS data is also removed to emphasize the impact of the IMU disturbances. The resulting positional estimates are presented in Figure 5.11.

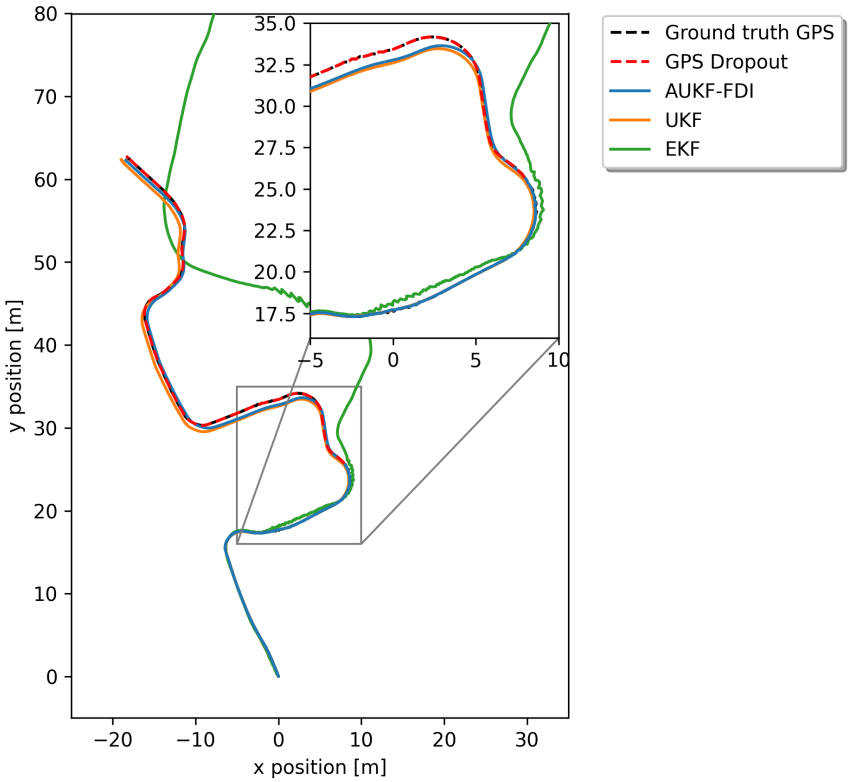


Figure 5.11: Positional estimates of the EKF, UKF, and AUKF-FDI for Scenario 4: GPS dropout, loosely mounted and disturbed IMU.

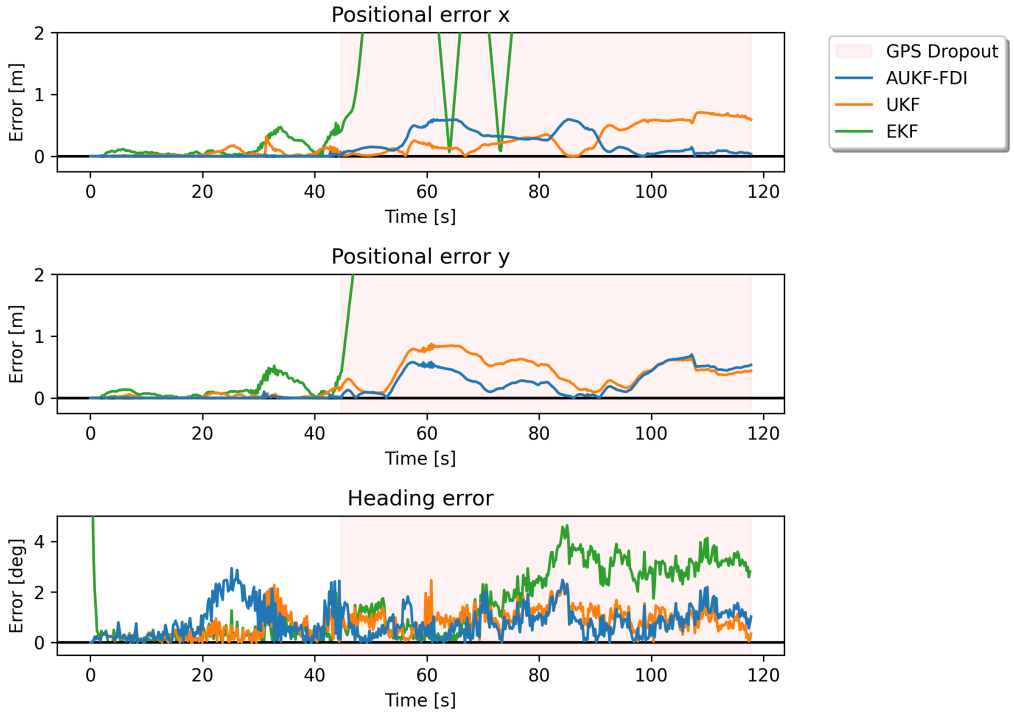


Figure 5.12: Positional and heading errors of the EKF, UKF, and AUKF-FDI for Scenario 4: GPS dropout, loosely mounted and disturbed IMU.

In this scenario, the EKF quickly diverges when the GPS measurements are removed, while the UKF-based filters do not. Although this could indicate that the linearization of the EKF results in a less stable filter in this type of scenario, the behavior can likely also be attributed to less-than-ideal tuning and setup of the EKF. Both the AUKF-FDI and UKF provide a more stable and accurate estimate during the whole scenario, with the AUKF-FDI providing the lower positional RMSE of the two, see Table 5.4. We can see from the gyroscope variance in Figure 5.13 that there are three large spikes when the vehicle is being rocked, indicating that the AUKF-FDI accurately lowers its trust in the sensor. This adaption likely leads to the AUKF-FDI having the lower positional RMSE, a conclusion which is further strengthened by the fact that AUKF provides the lowest RMSE for both the position. It should be noted that the loose mounting of the IMU did not have a notable impact on the measurements when the vehicle was not being disturbed.

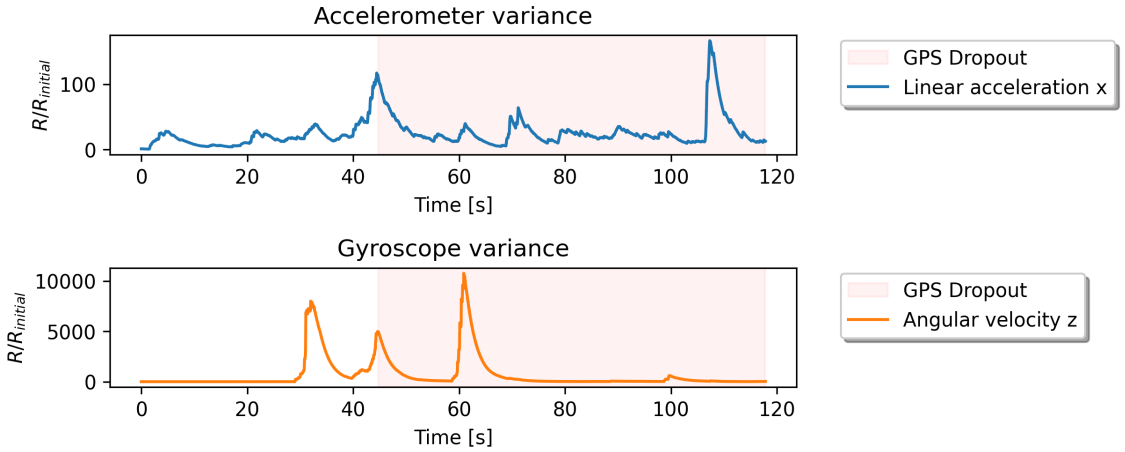


Figure 5.13: Change in expected IMU variance over time as computed by the adaptive component of the AUKF-FDI for Scenario 4: GPS dropout, loosely mounted and disturbed IMU.

Table 5.4: Performance of the state estimators for Scenario 4: GPS dropout, loosely mounted and disturbed IMU.

	RMSE		Max error	
	Position $[m]$	Heading $[deg]$	Position $[m]$	Heading $[deg]$
EKF	36.6647	2.0140	91.8671	9.0044
UKF	0.5044	**0.9266**	0.8945	**2.4593**
UKF-FDI	0.5847	0.9358	1.0241	2.8768
AUKF	**0.3153**	1.0311	**0.7818**	2.9443
AUKF-FDI	0.3747	1.0330	0.8232	2.9361

5.3 State Uncertainty

To demonstrate how well the estimated positional uncertainty of the AUKF-FDI corresponds to the actual deviation from the ground truth when faulty sensor data is present, the positional estimate, its estimated covariance, and the GPS ground truth are presented for a portion of each scenario. The resulting plots allow for a visual comparison to the standard UKF. The positional uncertainty of the state estimators is represented by the $\pm 3\sigma$-ellipses plotted at every three second interval. Figures 5.14-5.17 showcases the estimated positional uncertainty for scenario 1-4, respectively.

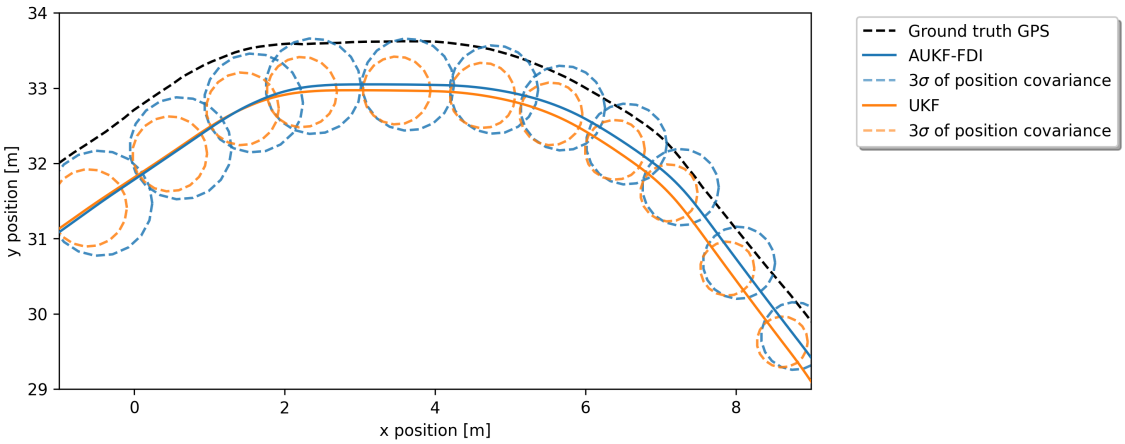


Figure 5.14: Positional estimates and their associated positional covariance represented as $\pm 3\sigma$-ellipses for both the UKF and the AUKF-FDI, the estimates have been obtained under GPS dropout.

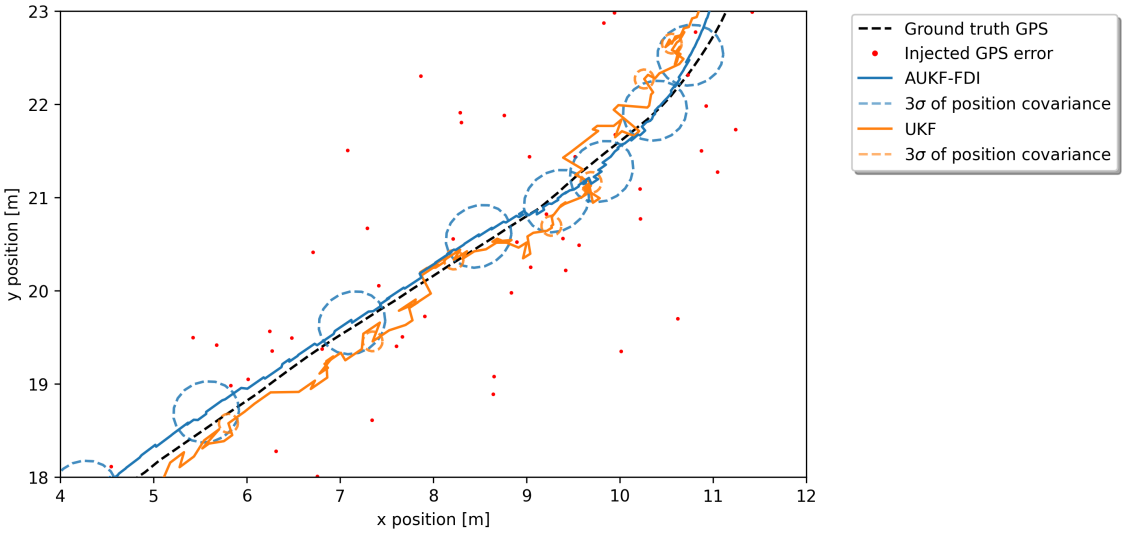


Figure 5.15: Positional estimates and their associated positional covariance represented as $\pm 3\sigma$-ellipses for both the UKF and the AUKF-FDI, the estimates have been obtained under noisy GPS measurements.

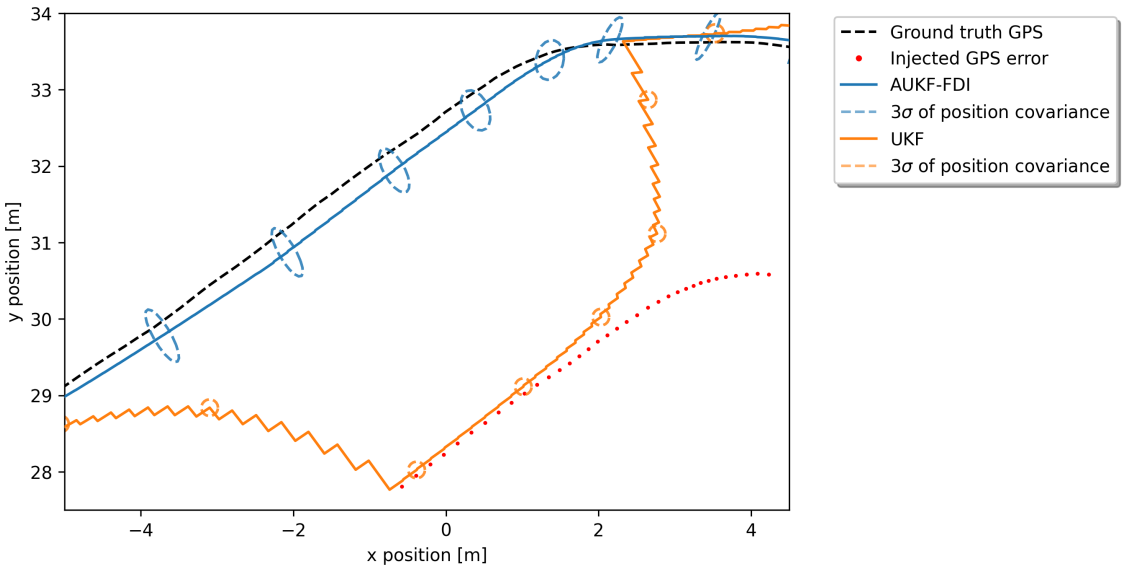


Figure 5.16: Positional estimates and their associated positional covariance represented as $\pm 3\sigma$-ellipses for both the UKF and the AUKF-FDI, the estimates have been obtained under GPS multi-path errors.

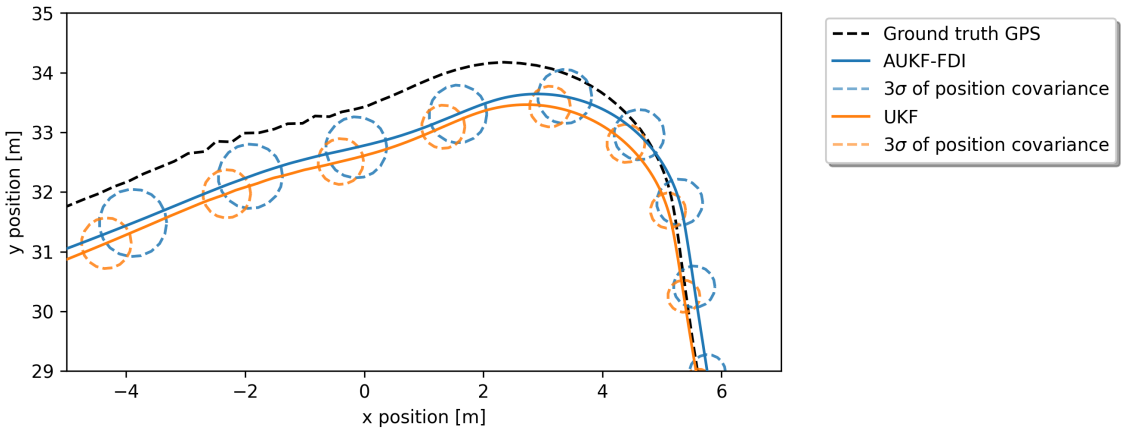


Figure 5.17: Positional estimates and their associated positional covariance represented as $\pm 3\sigma$-ellipses for both the UKF and the AUKF-FDI, the estimates have been obtained during a scenario with a loosely mounted IMU and GPS dropout.

In all four scenarios, it is evident that the AUKF-FDI has an increased uncertainty in its positional estimates compared to the UKF, indicated by the increased size of the the sigma-ellipses. When comparing the uncertainty to the actual deviation between the estimated positions of the filters and the GPS ground truth, the uncertainty of the AUKF-FDI corresponds better to the true deviation from the GPS ground truth, and the sigma-ellipses contains the ground truth for a more prominent part of the scenarios. This is especially true for the scenarios with added noise and offset to the GPS measurements, seen in Figures 5.15 and 5.16. The results indicate that the AUKF-FDI can adapt its uncertainty to provide a more accurate estimation under

several types of faulty measurements. Although the AUKF-FDI provides a more accurate uncertainty estimate compared to the UKF, the errors are still somewhat larger than estimated for the scenarios with GPS dropout, primarily for the case with the loosely mounted IMU, shown in Figure 5.17.

5.4 Summary and Discussion

Summarizing the results from all four scenarios, we can observe that the AUKF-FDI demonstrates an overall resilience to faulty sensor readings, and that it is the only evaluated filter with a consistent performance across all scenarios. The modeling and tuning of the AUKF-FDI provides accurate positional estimates under periods of GPS dropout. Furthermore, the AUKF-FDI adapts well to an increase in measurement noise, and it is capable of handling an offset GPS position. It is evident that the interplay between adaptive measurement noise covariance and FDI delivers the resiliency towards the spectrum of faulty sensor readings that were evaluated, meaning that both of these two components are crucial for robust self-localization. Although another of the showcased state estimators may provide a more accurate state estimate than the AUKF-FDI in a specific scenario, the alternative filter handles another scenario worse. We can also note that the currently implemented EKF of the research vehicle provides the worst state estimate in all four scenarios, proving the superiority of the custom modeling and the tuning of the UKF-based filters.

Although the AUKF-FDI consistently provides a smaller positional error compared to the UKF, it should be noted that the adaptive filters provides a larger error between the estimated heading and the GPS heading under all scenarios. As stated in Section 5.2.1, the reason for this is that the noise in the SLAM headings during the scenarios is lower than that of the initial tuning for the filters, which the adaptive update accurately estimates. However, since the SLAM heading has a small constant offset compared to the GPS heading, the adaptive state estimators will naturally have a larger heading error compared to the UKF which trusts the ground truth GPS heading to a much larger degree than the SLAM heading. The behaviour of the adaptive state estimators shows that the adaptive update of the noise covariance correctly estimates the noise of the measurements but that it can lead to errors when there are small constant offsets between the measurements that is not behaving like a Gaussian noise over time. The upcoming subsections discusses the evaluation strategy and highlights a few problematic areas of the developed self-localization algorithm.

5.4.1 GPS Ground Truth

The same GPS measurements are utilized as both the ground truth and relayed to the state estimators. Each state estimator therefore has access to the “true” position at all instances that does not consider an injected GPS fault, this means that the RMSE is a somewhat misleading performance indicator as it considers the heading and positional error over the entire scenario. A better way to evaluate the state estimators would be to utilize an independent positional reference, such as

an additional GPS with Real Time Kinematic (RTK) correction. A GPS receiver capable of RTK correction utilizes signals from both satellites and base stations to achieve centimeter precision [41]. Unfortunately, this type of reference system was not within the available resources of the thesis. Another alternative that was considered for evaluation was to use a manually measured reference track. However, the imprecise maneuvering of the research vehicle, due to wear of the steering servo, meant that trying to follow a reference track provided a less precise ground truth than the GPS.

5.4.2 Simulated Sensor Error

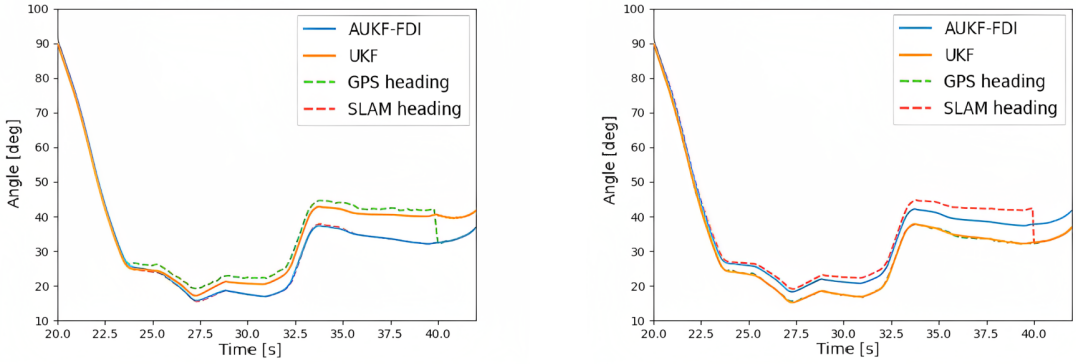
The study of the onboard sensors, covered in Section 2.2, provided a baseline understanding of the causes to faulty sensor behaviour. The main focus of the thesis considers GPS-related errors, but as aforementioned, the thesis also utilizes the GPS as ground truth for evaluation. The contradictory nature of using GPS as both ground truth and the main area of interest for faulty behaviour meant that evaluation only considered simulated GPS faults. Since the AUKF-FDI was developed and tuned using simulated sensor faults, it might be biased towards such "ideal" faulty measurements. Furthermore, the simulated GPS faults might not accurately reflect the complexity and behaviour of real-world GPS faults.

5.4.3 The Adaptive Measurement Noise Covariance Update

The adaptive update will tend to favor sensor measurements provided at a faster sampling rate if two measurements of the same type are contradictory. If a sensor measurement is applied more often, the state estimate will likely move towards the faster sensor while the slower measurement is unavailable. The resulting drift towards the faster sensor will therefore cause the adaptive update to increase the trust in the faster measurement, and when the slower measurement becomes available, the drift will imply that the slower measurement should be trusted less. The varying and measurement-specific forgetting factor γ was proposed in an attempt to reduce the dominance of faster sensors by assuring that faster sensors receives a lesser weighting in the adaptive update. The method of selecting the forgetting factor shows promise in reducing the favoring of faster sensors but the larger heading RMSE of the adaptive state estimators indicate that it did not completely solve the issue.

5.4.4 Drifting Sensor Errors

It is difficult for the AUKF-FDI to detect or adapt to a drifting measurement error. The adaptive update will consider the disturbance as an increased Gaussian noise which does not accurately describe a constant drift. Furthermore, the difference between actual and expected measurement will be small to begin with and the measurement will appear reasonable to the FDI. A measurement that exhibits a drifting bias will therefore be trusted in the Kalman update which will skew the state estimate, thus not increasing the Mahalanobis distance over time as the bias grows. The developed algorithm estimates the IMU and wheel scale biases to counteract



(a) Drift added to GPS heading.

(b) Drift added to SLAM heading.

Figure 5.18: Heading estimate over time during an introduced drift to the GPS (a), and SLAM (b) heading measurements. Showcasing how the estimates of the UKF and AUKF-FDI will vary depending on which measurement is experiencing the added drift.

this type of error, but should another sensor measurement slowly start to drift, it can result in a degraded state estimate. Drifting errors of other sensors than the IMU and the rotary encoders should be noted as a fault that the AUKF-FDI is not capable of handling. A solution to handle this type of error could be to increase the measurement redundancy. With more sensors providing the same type of measurements, it would be possible to extend the FDI component to not only evaluate the difference between the predicted measurements and received measurements, but also evaluate the difference between redundant measurements.

An example showing how the AUKF-FDI behaves when a simulated drifting error is introduced to two heading measurements with different sampling frequencies is presented in Figure 5.18, a simulated drifting error has been applied to the GPS and SLAM heading, respectively. The drift is introduced between $t \in [20, 40]$ and increases linearly from zero up to 10°. In Figure 5.18a, where the drift is introduced to the GPS, we can see that the AUKF-FDI estimate correctly conforms to the SLAM measurements, while the UKF exhibits a large error and follows the GPS heading closely. However, when the drift is introduced to the SLAM heading, as seen in Figure 5.18b, the opposite response occurs; The estimate of the AUKF-FDI follows the faulty SLAM heading, while the UKF conforms to the GPS heading.

6

Conclusion

The objective of the thesis was to develop and evaluate a self-localization algorithm that could accurately estimate the state of the research vehicle during scenarios with faulty sensor measurements. To achieve this task, an Unscented Kalman filter with fault detection and an adaptive update of the measurement noise covariance was implemented. The filter was evaluated under four scenarios with different simulated GPS measurement errors and a fourth scenario where the mounting of the IMU was loosened, the GPS was removed, and the vehicle was rocked sideways a few times during the test.

The developed self-localization algorithm, the Adaptive Unscented Kalman Filter with Fault Detection and Isolation, shows promise in providing a robust state estimate in scenarios with varying types of sensor dropouts or faults. It is proven that the modeling and tuning of the UKF-based filters provides a superior state estimate in comparison to the currently implemented EKF of the research vehicle, and that the AUKF-FDI is more resilient to the evaluated sensor faults in its positional estimate than a conventional UKF that utilizes the same modeling and settings as the AUKF-FDI.

The combination of the FDI component and the adaptive update provides a method of excluding initial sensor faults and then enabling the system to expect subsequent measurements at a larger variance. As a result, faulty measurements with a persistent error will still be regarded by the filter over time. According to the evaluation scenarios, it is clear that the benefit of adapting the noise covariance instead of excluding all faulty sensor measurements will depend on the type of error that the sensor experiences. With an increased Gaussian noise in the measurements, the benefit is apparent with the AUKF-FDI providing a less fluctuating positional estimate and a lower RMSE than the conventional filters. On the other hand, in the case of a sudden offset in the measurement that remains for several consecutive time steps, the benefit of keeping the measurement is diminished, with the erroneous measurements slowly starting to influence the state estimate as time progresses. Still, in the offset scenario, the AUKF-FDI provides a lower positional error than the EKF and UKF. Considering that constructing a filter solution that is ideal for all types of errors is a difficult task, the developed self-localization algorithm shows promise in providing a general and robust solution that can handle several types of errors.

It is, however, concluded that there are scenarios where the developed solution may lead to a more inaccurate state estimate compared to filters without the adaptive

update and fault detection components. For example, in a scenario where two sensors providing the same type of measurements start to slowly diverge from each other, with the faulty sensor having a faster update frequency and a lower noise in the measurements, the filter will likely trust the faulty sensor, resulting in significant errors. This type of scenario will, of course, result in estimate errors for the conventional Kalman filters as well since they assume the noise in the measurements to be Gaussian. However, the effect of the inaccurate measurement can likely be reinforced by the adaptive component of the AUKF-FDI.

6.1 Future Work

With regards to the results of the AUKF-FDI, the next natural step in development is to implement it in the research vehicle and to evaluate the robustness more rigorously. The developed self-localization algorithm should be tested over longer distances and in more diverse environments. Future work should also include a deeper analysis of sensor faults common to the urban driving environment and the state estimator should be evaluated on real GPS errors with an independent ground truth position.

A considerable amount of time was spent tuning the filter but further tuning should be considered for optimal performance. There are several parameters that could be subject to more extensive tuning, such as; the process covariance, how to define the forgetting factor dependent on sensor availability and sampling rates, and the FDI thresholds. Furthermore, an adaptive update of the process noise covariance could be implemented to estimate all noise characteristics of the system.

It would be valuable to test if it is possible to achieve better results with a more complex process model. The dynamic bicycle model or the Ackermann model could be evaluated as an alternative to the kinematic bicycle model that was utilized in the thesis.

Aptiv is planning to equip the research vehicle with a solid state LiDAR sensor. A LiDAR-based SLAM algorithm could potentially replace the camera-based ORB SLAM3 and provide a more accurate and reliable pose measurement in an open environment.

The literature study suggested that fault detection of sensory data using neural networks has seen success in similar projects. A neural network could be trained on labeled data to classify which type of sensory error is present. If the type of sensory error is classified, case-specific tuning or settings could be applied to the state estimator to produce the best possible state estimate. Training such a neural network requires large amounts of labeled data which was outside the resources allocated for the thesis but could be considered for future work.

Bibliography

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. [Online]. Available: https://doi.org/10.1115/1.3662552.

[2] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 1080–1088, 2007. DOI: 10.1109/TCST.2006.886439.

[3] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Defense, Security, and Sensing*, 1997.

[4] E. Wan and R. 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)*, 2000, pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.

[5] R. van der Merwe, E. A. Wan, and S. J. Julier, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," 2004.

[6] I. Ullah, X. Su, X. Zhang, D. Choi, Z. Hou, and J. Zhu, "Evaluation of localization by extended kalman filter, unscented kalman filter, and particle filter-based techniques," *Wireless Communications and Mobile Computing*, vol. 2020, p. 15, Oct. 2020. DOI: 10.1155/2020/8898672.

[7] "Evaluating the performances of adaptive kalman filter methods in gps/ins integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33–40, 2010.

[8] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *2017 IEEE Power & Energy Society General Meeting*, 2017, pp. 1–5. DOI: 10.1109/PESGM.2017.8273755.

[9] C. Hu, W. Chen, Y. Chen, and D. Liu, "Adaptive kalman filtering for vehicle navigation," *Journal of Global Positioning Systems*, vol. 02, pp. 42–47, 2003.

[10] T. Kobayashi and D. Simon, "Application of a bank of kalman filters for aircraft engine fault diagnostics," vol. 1, Sep. 2003. DOI: 10.1115/GT2003-38550.

[11] P. Guo, H. Kim, N. Virani, J. Xu, M. Zhu, and P. Liu, "Roboads: Anomaly detection against sensor and actuator misbehaviors in mobile robots," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 574–585. DOI: 10.1109/DSN.2018.00065.

[12] J. Bae, S. Yoon, and Y. Kim, "Attitude estimation for satellite fault tolerant system using federated unscented kalman filter," in Feb. 2011, vol. 11, ISBN: 978-953-307-551-8. DOI: 10.5139/IJASS.2010.11.2.080.

[13] S. Safavi, M. A. Safavi, H. Hamid, and S. Fallah, "Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles," *Sensors*, vol. 21, no. 7, 2021, ISSN: 1424-8220. DOI: 10.3390/s21072547. [Online]. Available: https://www.mdpi.com/1424-8220/21/7/2547.

[14] F. van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1264–1276, 2020. DOI: 10.1109/TITS.2019.2906038.

[15] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*, Springer, Jul. 2014.

[16] Lantmäteriet. "Gps och andra gnss." (2023), [Online]. Available: https://www.lantmateriet.se/sv/geodata/gps-geodesi-och-swepos/GPS-och-satellitpositionering/GPS-och-andra-GNSS/#qry=GPS%5C%20och (visited on 02/25/2023).

[17] R. B. Langley, P. J. Teunissen, and O. Montenbruck, "Introduction to gnss," in *Springer Handbook of Global Navigation Satellite Systems*, P. J. Teunissen and O. Montenbruck, Eds. Cham: Springer International Publishing, 2017, pp. 3–23, ISBN: 978-3-319-42928-1. DOI: 10.1007/978-3-319-42928-1_1. [Online]. Available: https://doi.org/10.1007/978-3-319-42928-1_1.

[18] M. Xie, "Application and exploration of computer technology in modern enterprise logistics management," *Journal of Physics: Conference Series*, vol. 1992, p. 032 061, Aug. 2021. DOI: 10.1088/1742-6596/1992/3/032061.

[19] M. S. Braasch, "Multipath," in *Springer Handbook of Global Navigation Satellite Systems*, P. J. Teunissen and O. Montenbruck, Eds. Cham: Springer International Publishing, 2017, pp. 443–468, ISBN: 978-3-319-42928-1. DOI: 10.1007/978-3-319-42928-1_15. [Online]. Available: https://doi.org/10.1007/978-3-319-42928-1_15.

[20] A. Pirsiavash, A. Broumandan, G. Lachapelle, and K. O'Keefe, "Detection and de-weighting of multipath-affected measurements in a gps/galileo combined solution," in *2019 European Navigation Conference (ENC)*, 2019, pp. 1–11. DOI: 10.1109/EURONAV.2019.8714191.

[21] [Online]. Available: https://support.strava.com/hc/en-us/articles/216917707-Bad-GPS-Data (visited on 03/16/2023).

[22] L. Teschler, *Inertial measurement units will keep self-driving cars on track*, 2018. [Online]. Available: https://www.microcontrollertips.com/inertial-measurement-units-will-keep-self-driving-cars-on-track-faq/ (visited on 03/14/2023).

[23] A. Quinchia, G. Falco, E. Falletti, F. Dovis, and C. Ferrer, "A comparison between different error modeling of mems applied to gps/ins integrated systems," *Sensors (Basel, Switzerland)*, vol. 13, pp. 9549–88, Aug. 2013. DOI: 10.3390/s130809549.

[24] "Coriolis force." (2023), [Online]. Available: https://www.britannica.com/science/Coriolis-force (visited on 03/14/2023).

[25] M. Xu *et al.*, *A critical analysis of image-based camera pose estimation techniques*, Jan. 2022. [Online]. Available: https://arxiv.org/abs/2201.05816.

[26] Y. Wu, F. Tang, and H. Li, *Image-based camera localization: An overview - visual computing for industry, biomedicine, and art*, Sep. 2018. [Online]. Available: https://vciba.springeropen.com/articles/10.1186/s42492-018-0008-z#citeas.

[27] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. DOI: 10.1109/TRO.2021.3075644.

[28] S. Särkkä and L. Svensson, *Bayesian filtering and smoothing*. Cambridge university press, 2023, vol. 17.

[29] E. Wan and R. 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)*, 2000, pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.

[30] C. C. Aggarwal and C. C. Aggarwal, *An introduction to outlier analysis*. Springer, 2017.

[31] H. Ghorbani, "Mahalanobis distance and its application for detecting multivariate outliers," *Facta Universitatis, Series: Mathematics and Informatics*, pp. 583–595, 2019.

[32] Y. Morales, E. Takeuchi, and T. Tsubouchi, "Vehicle localization in outdoor woodland environments with sensor fault detection," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 449–454. DOI: 10.1109/ROBOT.2008.4543248.

[33] P. Sundvall and P. Jensfelt, "Fault detection for mobile robots using redundant positioning systems," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 3781–3786. DOI: 10.1109/ROBOT.2006.1642280.

[34] A. H. Mohamed and K. P. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of Geodesy*, vol. 73, no. 4, pp. 193–203, May 1999, ISSN: 1432-1394. DOI: 10.1007/s001900050236. [Online]. Available: https://doi.org/10.1007/s001900050236.

[35] J. WANG, "Stochastic modeling for real-time kinematic gps/glonass positioning," *NAVIGATION*, vol. 46, no. 4, pp. 297–305, 1999. DOI: https://doi.org/10.1002/j.2161-4296.1999.tb02416.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.1999.tb02416.x.

[36] B. Zheng, P. Fu, B. Li, and X. Yuan, "A robust adaptive unscented kalman filter for nonlinear estimation with uncertain noise covariance," *Sensors*, vol. 18, no. 3, 2018, ISSN: 1424-8220. DOI: 10.3390/s18030808. [Online]. Available: https://www.mdpi.com/1424-8220/18/3/808.

[37] G. Cai, B. M. Chen, and T. H. Lee, "Coordinate systems and transformations," in *Unmanned Rotorcraft Systems*. London: Springer London, 2011, pp. 23–34, ISBN: 978-0-85729-635-1. DOI: 10.1007/978-0-85729-635-1_2. [Online]. Available: https://doi.org/10.1007/978-0-85729-635-1_2.

[38] S. True, "Planning the future of the world geodetic system 1984," in *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*, 2004, pp. 639–648. DOI: 10.1109/PLANS.2004.1309054.

[39] [Online]. Available: https://www.freepik.com/free-vector/most-common-auto-accidents-car-types-angle-collisions-isometric-set-with-sedan-saloon-hatchback-van-vector-illustration_26761382.htm (visited on 04/28/2023).

[40] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830.

[41] Lantmäteriet. "Metoder för gnss mätning." (2023), [Online]. Available: https://www.lantmateriet.se/sv/geodata/gps-geodesi-och-swepos/GPS-och-satellitpositionering/Metoder-for-GNSS-matning/RTK/ (visited on 05/09/2023).

A

Filter Settings and Parameters

The appendix considers the tuning parameters of the proposed solution, the AUKF with FDI. Furthermore, the other UKF-based filtering solutions that are evaluated utilizes the same tuning parameters.

Weight and sigma point parameters:

$$\alpha = 10^{-3} \tag{A.1}$$
$$\beta = 2 \tag{A.2}$$
$$\kappa = 0 \tag{A.3}$$

Measurement noise covariance:

$$\boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{GPS}} = \begin{bmatrix} 0.10^2 & 0 & 0 & 0 \\ 0 & 0.10^2 & 0 & 0 \\ 0 & 0 & 0.07^2 & 0 \\ 0 & 0 & 0 & \left(\frac{0.3\pi}{180}\right)^2 \end{bmatrix} \tag{A.4}$$

$$\boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{IMU}} = \begin{bmatrix} 0.07848^2 & 0 \\ 0 & \left(\frac{0.083\pi}{180}\right)^2 \end{bmatrix} \tag{A.5}$$

$$\boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{Odom}} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times 10^{-3} \tag{A.6}$$

$$\boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{SLAM}} = \begin{bmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & \left(\frac{2.5\pi}{180}\right)^2 \end{bmatrix} \tag{A.7}$$

$$\boldsymbol{R}_{\mathbf{1}} = \begin{bmatrix} \boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{GPS}} & & & \\ & \boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{IMU}} & & \\ & & \boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{Odom}} & \\ & & & \boldsymbol{R}_{\mathbf{1}}^{\boldsymbol{SLAM}} \end{bmatrix} \tag{A.8}$$

Measurement noise covariance update - forgetting factor:

$$\gamma_k^i = \begin{cases} 0.5t_k^i & \text{if } 0.5t_k^i < 0.20 \\ 0.20 & \text{otherwise} \end{cases} \tag{A.9}$$

Fault detection and isolation thresholds:

Table A.1: Chi-square distribution table determining the critical value of the distribution based on the degrees of freedom and the desired significance level.

Degrees of freedom(df)	**Significance level**							
	.99	**.975**	**.95**	**.9**	**.1**	**.05**	**0.025**	**.01**
1	-	0.001	0.004	0.016	2.706	3.841	5.024	6.635
2	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210
3	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345
4	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277

Table A.2: Selected thresholds for each sensor measurement based on a significance level of 0.01 and the number of degrees of freedom.

Measurement	Threshold
(x^{GPS}, y^{GPS})	9.210
v^{GPS}	6.635
ψ^{GPS}	6.635
$a^{IMU,x}$	6.635
$\omega^{IMU,z}$	6.635
$v^{RL,Odom}$	6.635
$v^{RR,Odom}$	6.635
δ^{Odom}	6.635
(x^{SLAM}, y^{SLAM})	9.210
ψ^{SLAM}	6.635

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY