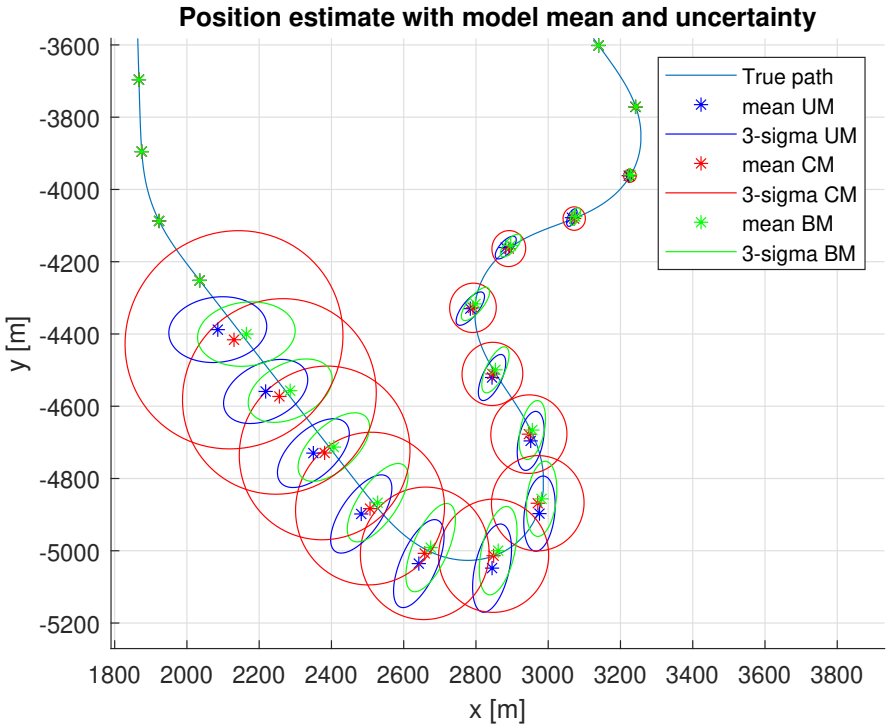


Enhancing INS Accuracy in GNSS-Denied Environments

Incorporating Vehicle Dynamics Motion Models and Slip Estimation in INS Algorithm to Improve Positioning Accuracy

Master's thesis in Systems, Control, and Mechatronics

JOHANNES JOHANSSON
LINUS WESTLUND

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis in Systems, Control, and Mechatronics

Enhancing INS Accuracy in GNSS-Denied Environments

Incorporating Vehicle Dynamics Motion Models and Slip Estimation in INS Algorithm to Improve Positioning Accuracy

JOHANNES JOHANSSON
LINUS WESTLUND

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Gothenburg, Sweden 2025

Enhancing INS Accuracy in GNSS-Denied Environments
Incorporating Vehicle Dynamics Motion Models and Slip Estimation in INS Algorithm to Improve Positioning Accuracy
JOHANNES JOHANSSON
LINUS WESTLUND

Supervisor: Hanna Jonsson, CPAC Systems AB
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2025
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Visualizing the 3-sigma confidence region of the models during GNSS-outage.

Typeset in LaTeX
Gothenburg, Sweden 2025

Enhancing INS Accuracy in GNSS-Denied Environments
Incorporating Vehicle Dynamics Motion Models and Slip Estimation in INS Algorithm to Improve Positioning Accuracy
JOHANNES JOHANSSON
LINUS WESTLUND
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Accurate vehicle positioning is critical for navigation systems, and one approach to gain high positioning accuracy is to fuse the measurements from a global navigation satellite system (GNSS) and an inertial measurement unit (IMU). However, certain challenges remain, particularly in scenarios involving vehicle slip and GNSS-denied environments.

This thesis investigates how using vehicle-specific motion models and slip estimation can enhance the performance of an inertial navigation system (INS). Three motion models are developed and evaluated in both simulated and real-world scenarios. These are a constant acceleration model (CM), a unicycle model (UM), and a bicycle model (BM). A slip estimation method is proposed, using a Kalman filter to adaptively estimate slip parameters based on GNSS and IMU data.

Results show that the UM and BM outperform the CM in most scenarios, with the BM demonstrating superior accuracy in the presence of slip. Real-world tests show potential for the UM and BM as they seem to be able to follow the expected shape of the trajectory. However, limitations due to 2D assumptions result in incorrect values for the velocity, leading to inadequate positioning accuracy. The results indicate that integrating advanced motion models and slip estimation into INS algorithms can significantly improve positioning accuracy. However, further testing and extension to 3D implementations are necessary to validate these results in real-world applications.

Keywords: INS, GNSS, Kalman Filter, Slip Estimation, Motion Model, Vehicle Dynamics, Sensor Fusion, Dead Reckoning, Unicycle Model, Bicycle Model.

Preface

This report presents the outcome of our master's thesis project carried out at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology during the spring of 2025. The project was carried out in collaboration with CPAC Systems AB, which proposed the thesis topic and provided a supportive working environment. We were given access to office space at the company and benefited greatly from insightful discussions with our company supervisor and other employees. Throughout the project, we gained valuable knowledge and practical experience in vehicle positioning, particularly inertial navigation, motion modeling, and slip estimation.

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Hanna Jonsson at CPAC Systems AB, for her continuous guidance, insightful feedback, and encouragement throughout the work on this thesis. We also wish to thank Mikael Tulldahl at CPAC Systems AB for his support with theoretical insights and coding advice. Our thanks extend to all employees at CPAC Systems AB for their welcoming and curious attitude toward the project, as well as for including us in enjoyable company activities such as group training sessions and lunchtime beach volleyball. Finally, we would like to thank our examiner, Peter Forsberg at the Department of Mechanics and Maritime Sciences, Chalmers University of Technology, for his valuable advice and for examining our thesis.

Johannes Johansson & Linus Westlund, Gothenburg, June 2025

Thesis advisor: Hanna Jonsson, CPAC Systems AB
Thesis examiner: Peter Forsberg, Mechanics and Maritime Sciences

List of Acronyms

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

AI	Artificial intelligence
BM	Bicycle model
CM	Constant acceleration model
CPAC	CPAC Systems AB
CST	Continuous time state transition function
DST	Discrete time state transition function
EKF	Extended Kalman filter
fl	Front left
fr	Front right
GNSS	Global navigation satellite system
IMU	Inertial measurement unit
INS	Inertial navigation system
LC	Loose coupling
rl	Rear left
RMSE	Root mean square error
rr	Rear right
RTK	Real-time kinematic positioning
TC	Tight coupling
UM	Unicycle model

Nomenclature

Below is the nomenclature of indices, parameters, variables and functions that have been used throughout this thesis.

Indices

$k,\ i,\ j$	Simulation timestep index
n	Simulation length
Δt	Timestep size

Parameters

L_r	Distance from rear axis to center of gravity
L_f	Distance from front axis to center of gravity
L	Distance between front and rear axis
m	mass of the vehicle
h	The height of center of gravity over ground
R	Rotation matrix
$\mathbf{a}_b$	Accelerometer measurement bias
$\mathbf{a}_\epsilon$	Accelerometer measurement noise
σ_a	Accelerometer measurement noise standard deviation
ω_ϵ	Gyroscope measurement noise
σ_ω	Gyroscope measurement noise standard deviation
$\mathbf{p}_\epsilon$	GNSS position measurement noise
σ_p	GNSS position measurement noise standard deviation
θ_ϵ	GNSS heading measurement noise
σ_θ	GNSS heading measurement noise standard deviation
a_{mv}	Accelerometer measurement variance

ω_{mv}	Gyroscope measurement variance
p_{mv}	Position measurement variance
θ_{mv}	Heading measurement variance
$\boldsymbol{Q}_{CM}$	Process noise covariance matrix for constant acceleration model
$\boldsymbol{R}_k^{GNSS}$	GNSS measurement noise covariance matrix
$\boldsymbol{Q}_{UM}$	Process noise covariance matrix for unicycle model
$\boldsymbol{Q}_{BM}$	Process noise covariance matrix for bicycle model

Variables

$\hat{x}_{k\|k-1}$	Predicted state estimate
F_{k-1}	State transition model
$P_{k\|k-1}$	Predicted estimate covariance
Q_{k-1}	Process noise
$\hat{h}_k$	Kalman prediction
H_k	Observation matrix
y_k	Innovation
z_k	Measurement
S_k	Innovation covariance
R_k	Measurement noise
K_k	Kalman gain
$\hat{x}_{k\|k}$	Updated state estimate
$P_{k\|k}$	Updated estimate covariance
α	Slip angle at center of gravity
ω_z	Angular velocity around the z-axis
V	Velocity forward
K_r	Rear wheel cornering power
K_α	Relation between $m,\ L_f,\ L$ and K_r
$v_i^d,\ v_j^d$	Derived velocity
$\omega_i^d,\ \omega_j^d$	Derived angular velocity
$a_i^d,\ a_j^d$	Derived acceleration
$p_i,\ p_j$	Position
v_i^{IMU}	Derived velocity in IMU frame
a_i^{IMU}	Derived acceleration in IMU frame

ω_i^{IMU}	Derived angular velocity in IMU frame
$\mathbf{a}_i$	Generated accelerometer measurement
ω_i	Generated gyroscope measurement
$\mathbf{p}_k$	Generated position measurement
θ_k	Generated heading measurement
θ	Attitude
$\dot{x}$	Continuous x-position derivative
$\dot{y}$	Continuous y-position derivative
ω_x	Roll rate, angular velocity around x-axis
ω_z	Yaw rate, angular velocity around z-axis
F_y	Lateral forces
F_x	Longitudinal forces
$F_{rl}, F_{rr}, F_{fl}, F_{fr}$	Forces at each tire, rear left (rl), rear right (rr), front left (fl), and front right (fr).
θ_z	Heading angle, (yaw angle)
θ_x	Roll angle
δ	Steering angle
x	x-coordinate
V_x	Velocity in x direction
y	y-coordinate
V_y	Velocity in y direction
a_x	x-component of acceleration
a_y	y-component of acceleration
$\boldsymbol{a}_c$	Centripetal acceleration
$\boldsymbol{a}_\alpha$	Acceleration in slip direction
$\boldsymbol{a}_{\alpha,x}$	x-component of $\boldsymbol{a}_\alpha$
v_{est}	Estimated velocity form velocity estimator
$\hat{k},\ \hat{m}$	Parameters of velocity estimator
v_{pred}	Predicted velocity from models
x_{vel}	Velocity estimator states
v_{gnss}	GNSS measurement for velocity estimator
y_{vel}	Velocity prediction innovation
x_{slip}	Slip estimator states
$\hat{L}_r,\ \hat{K}_\alpha$	Slip estimation parameters

Functions

$f(\cdot)$	Function describing motion of nonlinear model
$F(\cdot)$	Jacobian of $f(\cdot)$
$h(\cdot)$	Function for nonlinear measurement model
$H(\cdot)$	Jacobian of $h(\cdot)$

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xvii**

List of Tables **xix**

1 Introduction **1**
1.1 Background . . . 1
1.2 Purpose . . . 2
1.2.1 Research questions . . . 3
1.3 Limitations . . . 3
1.4 Thesis outline . . . 3

2 Theory **5**
2.1 Kalman filter . . . 5
2.1.1 Extended Kalman filter (EKF) . . . 6
2.2 Motion models . . . 6
2.2.1 Slip . . . 7
2.3 Global navigation satellite system (GNSS) . . . 8
2.4 Inertial navigation system (INS) . . . 8
2.5 INS/GNSS coupling . . . 9

3 Methods **11**
3.1 Simulation environment . . . 11
3.1.1 Manual path simulation without slip . . . 12
3.1.2 Model based path simulation with slip dynamics . . . 13
3.1.3 Sensor simulation . . . 14
3.1.4 Simulation loop . . . 15
3.2 Motion models . . . 16
3.2.1 Constant acceleration model . . . 16
3.2.2 Unicycle model . . . 18
3.2.3 Bicycle model . . . 19
3.3 Velocity estimation . . . 21
3.4 Slip estimation . . . 22
3.5 Evaluation metrics . . . 23

4 Results **25**
4.1 Estimating the path in different simulation environments 25
4.1.1 Tests in the environment with manual path simulation and no slip 25
4.1.2 Tests in the model based simulation environment 29
4.2 Velocity estimation 33
4.3 Slip estimation 34
4.4 Robustness 36
4.5 Uncertainty comparison 38
4.6 Test with real data 39

5 Discussion **47**
5.1 Simulation environment 47
5.2 Accuracy of the models for different scenarios 48
5.3 Slip estimation 49
5.4 Ethical and sustainability aspects 50
5.5 Future work 50
5.5.1 Real world expansion and testing 50
5.5.2 IMU misalignment compensation 51

6 Conclusion **53**

Bibliography **55**

List of Figures

2.1 Different trajectories resulting from the same steering input and different road conditions. A) the road has high friction meaning the heading will follow the velocity trajectory. B) slip is introduced and there is a deviation from the velocity trajectory, which is the slip angle α. C) the road has low friction meaning the heading will deviate more, i.e the slip angle α will increase. 7

3.1 Defining the world frame (red) and the IMU frame (blue). 11
3.2 Showing the 4 degrees of freedom model including forces affecting the slip and the slip angles. The vehicle turns around the center of rotation point P. 14
3.3 Accelerations measured from the IMU (red) and the assumed true accelerations during slip (blue). 20

4.1 Ground truth for the environment with manual path simulation and no slip. 26
4.2 Results from tests in the environment with manual path simulation and no slip. 26
4.3 Results from the models using GNSS measurements and dead reckoning in the environment with manual path simulation and no slip. . 28
4.4 Ground truth for model based simulation environment without slip. . 29
4.5 Results from the models using GNSS measurements and dead reckoning in the model based simulation environment without slip. 30
4.6 Ground truth for model based simulation environment with slip. . . . 31
4.7 Results from the models using GNSS measurements and dead reckoning in the model based simulation environment with slip. 32
4.8 Velocity estimation using parameters trained by a Kalman filter, simulated with no added GNSS noise. 33
4.9 Results from the slip estimator with a reference velocity of 20 m/s. . 34
4.10 Results from the slip estimator with a reference velocity of 60 m/s. . 34
4.11 Parameters of slip estimator with a reference velocity of 20 m/s. . . . 35
4.12 Position estimates for the models, showing their estimate means and 3-sigma ellipses at an interval of 1000 time-steps, visualizing the uncertainty of their estimates. Having a GNSS outage through the curves results in larger ellipses since there is no GNSS update reducing the covariance. 38
4.13 Collected data from a test log, using sensors mounted on a car. 39

4.14 Collected IMU and GNSS data in the parking garage. 40
4.15 Estimated positions and velocities from parking garage data. 41
4.16 Path estimate from INS log in the parking garage. 42
4.17 Results of the slip estimator in the parking garage. 42
4.18 Collected data for the full path, using CPAC:s sensors on testing equipment. 43
4.19 Estimated positions and velocities for the full path. 44
4.20 Results of the slip estimator for the full path. 45

List of Tables

3.1 Added sensor noise to simulate realistic behavior. 15

4.1 RMSE values for 10 simulations in the environment with manual path simulation and no slip, without GNSS. 27
4.2 RMSE values for 10 simulations in the environment with manual path simulation and no slip, with GNSS. 27
4.3 RMSE values for 10 simulations using the model based simulation environment without slip. 30
4.4 RMSE values for 10 simulations using the model based simulation environment with slip. 32
4.5 Position RMSE values for the models with different accelerometer noise standard deviation. 36
4.6 Position RMSE values for the models with different gyroscope noise standard deviation. 36
4.7 Position RMSE values for the models with different GNSS position noise standard deviation. 37
4.8 Position RMSE values for the models with different GNSS heading noise standard deviation. 37
4.9 GNSS quality translation. 40

1

Introduction

In today's society, many products and systems are being digitalized and automated, contributing to safety and productivity. Vehicles are among these products being implemented with advancing aiding systems to guide the user, increase safety, and reduce environmental impact while also maximizing efficiency and profit. One such system is positioning, where precise and reliable navigation has become essential.

CPAC Systems AB (CPAC) has developed an in-house inertial measurement unit (IMU), a product that combines gyroscopes and accelerometers specifically designed for the automotive industry. This IMU, when integrated with global navigation satellite system (GNSS) solutions, fuses sensor data to provide highly accurate positioning for vehicles. However, certain challenges remain, particularly in scenarios involving vehicle slip, a deviation between the heading and velocity direction, and GNSS-denied environments. Therefore, this thesis aims to improve the system's accuracy by incorporating motion models and slip estimation, enhancing overall reliability of vehicle navigation.

1.1 Background

Multiple factors can disturb navigation when driving in an urban environment. The system's position can be determined accurately when GNSS is available. However, when not receiving a signal, the system relies on dead reckoning from the inertial navigation system (INS). This method integrates IMU measurements to estimate the position based on a previously known location. The accelerometer introduces sensor noise and disturbances, which cause drift in the estimations when integrated to determine velocities and position.

Previous works have used a simple motion model and data from an IMU/GNSS fusion to estimate slip angle and later the vehicle position [1], [2]. The results from the studies showed that cheaper IMU sensors and GNSS receivers received similar accuracy to expensive ones. However, Takikawa et al. state that some points during testing had higher slip angle estimation errors since the used model was insufficient [1]. Research done by Li et al. regarding sideslip angle estimation has utilized a more complex motion model [3]. They have, however, exploited more measurements than those from IMU and GNSS, such as steering angle measurements, for the estimations. Research by Eriksson et al. showed how faulty sensor readings could

be filtered to achieve higher accuracy in the self-positioning algorithm [4]. Similar to the previously mentioned work, a simple vehicle model was used. The authors remark in future work that researching more complex vehicle models could lead to improved accuracy. The slip had not been considered in their research, and implementing this in the algorithm could potentially increase the accuracy of the positioning.

The vehicle dynamics can be modeled in multiple ways. Previous research by Patil compares a more straightforward and more advanced model for a truck [5]. The simpler model represented the truck as a two-wheeled bicycle model, while the more advanced model included all six wheels and the forces from the cabin. The results showed that the advanced model achieved better accuracy for most cases.

Having an accurate system model is necessary to determine the position. However, errors can be introduced when the dynamics and measurement models are implemented. A proposed method by Zhang et al. used a modified adaptive robust extended Kalman filter to increase adaptability to system nonlinearity and model errors [6]. Research by Fransson et al. investigated how a neural network filter compared to ordinary Kalman filters [7]. They concluded that it was possible to utilize machine learning approaches, but they were limited to the quality of the training data. Other research has furthermore explored the use of artificial intelligence (AI) to estimate sensor noise and disturbances, as well as to predict the GNSS signal during outages [8], [9], [10]. Additionally, studies have investigated the integration of INS with supplementary measurements such as odometry and steering angle to enhance positioning accuracy [11], [12], [13]. However, the sensors available for this project are only IMU and GNSS solutions.

The research by Takikawa et al. used numerous data points for linear regression to determine the coefficient used for slip estimation [1]. In other research by Huang et al., the slip estimation was more adaptive since the parameters were trained by an error state Kalman filter [2]. To extend the research in this area, it is of interest to investigate how slip estimation can be achieved by parameter training using states instead of error states in a Kalman filter. Comparing different motion models and the effect of slip estimation can show how motion and slip modeling affect the accuracy of an INS solution.

1.2 Purpose

The purpose of this thesis is to develop an INS algorithm that compares various motion models and estimates normal vehicle slip. The thesis further investigates how these models affect vehicle navigation accuracy during GNSS signal loss. One objective is to demonstrate an INS algorithm that utilizes both a motion model and slip estimation for increased accuracy.

1.2.1 Research questions

The aim is to answer the following research questions in this Master's thesis:

- How can the existing INS solution be enhanced with a specific vehicle motion model for a robust fusion of GNSS and IMU data for accurate positioning?
- How can the addition of slip estimation improve the INS solution performance for accurate positioning?

1.3 Limitations

One limitation is that development was done in a two-dimensional environment. The thesis approach was to work in an agile manner, developing all parts simply first and advancing each part of the solution thereafter. Focusing the work on ensuring simulation accuracy in two dimensions led to not extending to all three dimensions within the time frame.

Another limitation of the thesis is the accuracy of the simulated sensors. The disturbances were simulated as normal distributions, resulting in some lost realistic sensor behavior. For example, in reality the GNSS noise follows a random walk process due to satellite movement when real-time kinematic positioning (RTK) is not available. Simulating the disturbances as normal distributions was, however, considered sufficient for this thesis.

The research was also limited to only considering lateral slip that can occur in normal driving conditions, hence disregarding total loss of friction. Furthermore, since the IMU can not measure the movement of the wheels, the longitudinal slip can not be determined.

1.4 Thesis outline

The thesis is structured as follows, beginning with a theory Chapter 2, introducing significant theoretical background to the project, such as how Kalman filters work, motion models, and slip modeling. Thereafter is Chapter 3 describing the methods used, building the simulation environment, and the different motion models evaluated in the thesis. This chapter also explains the velocity and slip estimation.

The results are presented in Chapter 4, where two simulation environments are evaluated. The first environment is based on a point-generated path, from which the simulated sensor data is derived. The second utilizes a vehicle model driven by velocity and steering inputs to generate the path and sensor data. For each simulation environment, the different motion models have been tested and compared. Additionally, the results from the velocity and slip estimations are presented. The final results are from running the simulations on realistic data collected from a sensor log. A detailed discussion of the results is provided in Chapter 5 and concluded in Chapter 6.

2

Theory

This chapter presents necessary concepts for this project, such as Kalman filters used to estimate the states of dynamic systems. Additionally, it explains motion and slip modeling to describe the vehicle movement. Lastly, the chapter explains how GNSS and IMU measurements are used to determine the position and to navigate.

2.1 Kalman filter

A Kalman filter can be used to combine data from different sensors and calculate an estimate from the information [4], [14]. The filter is based on two steps: the prediction step estimates the states and covariance based on previous knowledge, and the update step calculates an update to the states using measurements from the sensors. For the Kalman filter to work, the system needs to be linear.

Prediction step:

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} \tag{2.1}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1} \tag{2.2}$$

The equations calculate the predicted state estimate $\hat{x}_{k|k-1}$ and the predicted estimate covariance $P_{k|k-1}$. The calculations use the state transition model F_{k-1} and previous known state and covariance. The predicted estimate covariance also includes the process noise Q_{k-1}.

Update step:

$$\hat{h}_k = H_k\hat{x}_{k|k-1} \tag{2.3}$$

$$y_k = z_k - \hat{h}_k \tag{2.4}$$

$$S_k = H_kP_{k|k-1}H_k^T + R_k \tag{2.5}$$

$$K_k = P_{k|k-1}H_k^TS_k^{-1} \tag{2.6}$$

Where $\hat{h}_k$ is the prediction, combining H_k, the observation matrix, and $\hat{x}_{k|k-1}$, predicted state estimate. The innovation y_k is calculated as the difference between the measurement z_k and the prediction $\hat{h}_k$. The covariance of the innovation S_k is calculated using the current estimate of the state covariance $P_{k|k-1}$ with added

measurement noise R_k. These are used to calculate the Kalman gain matrix K_k and thereafter update the filter using Equations 2.7 and 2.8.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \tag{2.7}$$
$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{2.8}$$

Where $\hat{x}_{k|k}$ is the updated state estimate and $P_{k|k}$ is the updated estimate covariance.

2.1.1 Extended Kalman filter (EKF)

The extended Kalman filter (EKF) extends the regular Kalman filter to handle nonlinear problems [15]. This is done by changing the filter equations in the following way:

Prediction step:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) \tag{2.9}$$
$$P_{k|k-1} = F(\hat{x}_{k-1|k-1}) P_{k-1|k-1} F^T(\hat{x}_{k-1|k-1}) + Q_{k-1} \tag{2.10}$$

Update step:

$$\hat{h}_k = h(\hat{x}_{k|k-1}) \tag{2.11}$$
$$y_k = z_k - \hat{h}_k \tag{2.12}$$
$$S_k = H(\hat{x}_{k|k-1}) P_{k|k-1} H^T(\hat{x}_{k|k-1}) + R_k \tag{2.13}$$
$$K_k = P_{k|k-1} H^T(\hat{x}_{k|k-1}) S_k^{-1} \tag{2.14}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \tag{2.15}$$
$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{2.16}$$

Where $f(\cdot)$ is the function for the dynamic model, $F(\cdot)$ is its Jacobian, $h(\cdot)$ is the function for the measurement model and $H(\cdot)$ is its Jacobian.

2.2 Motion models

Motion models are models that describe a vehicle's dynamic behavior, and they can be used to predict the future position of the vehicle [16]. There exist a lot of different models with different levels of complexity. One of the simplest models is the constant velocity model, being a linear model that assumes a constant velocity. Here, the states are described by the position and velocity, where the position is updated using the velocity. The problem with this simple model is that it presumes the motion to be straight and can not account for rotations. To be able to handle more complex cases, more complex models that describe that kind of behavior have been created.

One of the key tasks when working with motion models is estimating the system's states, as discussed in [16]. This process typically relies on sensor data and is often

implemented using variants of the Kalman filter, which provides a recursive solution for estimating the state of a dynamic system in the presence of noise.

The advantage of having the vehicle motion model in the INS algorithm is that in the case of GNSS outage, it is still possible to do a complete update step in the Kalman filter [17]. However, this requires that the vehicle dynamics are measured and observable. For instance, using wheel speed and steering angle sensors, it is possible to calculate the vehicle movement and estimate the relative positioning with the proposed method by D. Salmon [17].

2.2.1 Slip

When a vehicle turns it occurs a difference between the heading angle and the velocity angle, which is called the slip angle [1]. Depending on the road conditions and velocity, the same steering input from the driver will result in different trajectories [18]. Illustrated in Figure 2.1 is an example where the vehicle has been given the same steering angle but different road conditions.

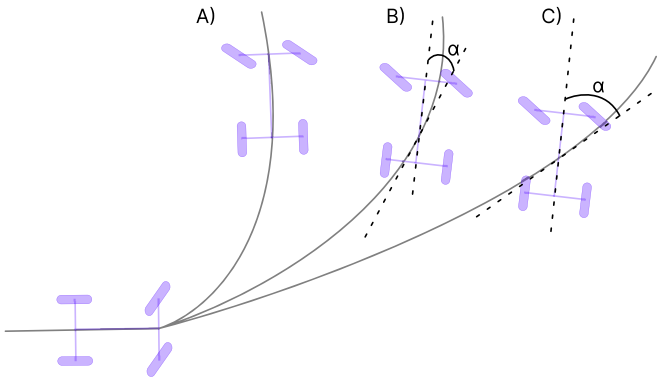


Figure 2.1: Different trajectories resulting from the same steering input and different road conditions. A) the road has high friction meaning the heading will follow the velocity trajectory. B) slip is introduced and there is a deviation from the velocity trajectory, which is the slip angle α. C) the road has low friction meaning the heading will deviate more, i.e the slip angle α will increase.

To estimate the slip, Takikawa et al. derived Equation 2.17 for the bicycle model [1]. Where α is the slip angle at the center of gravity, L_r is the distance from the rear axis to the center of gravity, L_f is the distance from the center of gravity to the front axis, and L is the total distance between the front and rear axis. The vehicle mass is m, the angular velocity around the z-axis is ω_z, the forward velocity is V, and K_r is the rear wheel cornering power. In the report by Takikawa et al., the equation is expanded to include the roll angle as well. However, since this study is limited to two dimensions, this part is omitted.

$$\alpha = \omega_z \left(\frac{L_r}{V} - \frac{m \cdot L_f}{2 \cdot L \cdot K_r} \cdot V \right) \tag{2.17}$$

The components of Equation 2.17 are not constants, and they are not always easy to measure [1]. For example, the position of center of gravity and the vehicle mass will change depending on the load conditions. Therefore, Equation 2.17 can be simplified to Equation 2.18, where $\frac{m \cdot L_f}{2 \cdot L \cdot K_r}$ have been combined into K_α.

$$\alpha = \omega_z \left(\frac{L_r}{V} - K_\alpha \cdot V \right) \tag{2.18}$$

2.3 Global navigation satellite system (GNSS)

GNSS can provide accurate positioning and velocity to the user. The system calculates the user's position by combining the information from multiple satellites, which transmit signals using radio frequency [19]. This solution allows worldwide positioning using satellites. The existing systems use shorter wavelengths for transmission to increase accuracy, which limits the user to be within line-of-sight to receive the signals. In an urban environment, this can be challenging.

One way to increase GNSS accuracy is by utilizing RTK, which uses a reference station that transmits correction data to the vehicle. The vehicle additionally receives GNSS signals from the satellites, and having both these measurements makes it possible to correct the position estimate, increasing accuracy to a centimeter-level [20], [21].

2.4 Inertial navigation system (INS)

An INS provides positional data with the help of IMU measurements [22]. This allows the INS to provide positional information between the slower updates from the GNSS and positional estimates in areas where the GNSS signal is unavailable. The IMU is used to get measurements for accelerations and angular velocities, which are processed and rotated from the coordinate frame for the measurements to the coordinate frame for the navigation. With the help of integrating the values, it is possible to get the position, velocity, and attitude.

However, when integrating the measurements, bias errors and white Gaussian noise errors lead to a drift in the position estimation, as errors accumulate [23], [24]. The accuracy of the sensors differs between types and is categorized in precision grades. A higher grade IMU has higher accuracy, less noise, and thereby better position estimation [25], [26]. Generally, the gyroscope has a lower noise magnitude compared to the accelerometer. The IMU used in the test log for the real-world data evaluation can be found in [27].

2.5 INS/GNSS coupling

There are multiple different ways to fuse the information from GNSS with INS. Two of them are loose coupling (LC) and tight coupling (TC) [28]. When LC is used, the GNSS and INS independently calculate the position and heading. Afterward, the GNSS measurements are used to correct the INS errors. This approach is relatively straightforward to implement but has issues when operated in a GNSS-challenging environment. The error accumulates because the INS error can not be corrected without GNSS measurements. For TC, the GNSS no longer calculates the position and heading directly. Raw GNSS data is used instead with the INS in a Kalman filter to determine the position. This method is more demanding to implement than LC, but it can still utilize the GNSS data in challenging environments and achieve higher accuracy. More complex coupling methods exist, discussed by Mouyan et al. in [29] and by GAO et al. in [30].

3

Methods

This chapter presents the methodology used in the thesis. Two simulation environments were created: one that manually defined the vehicle path and one that used a vehicle model to generate a path. Three motion models were developed to estimate the vehicle motion: a constant acceleration model, a unicycle model, and a bicycle model. The chapter also introduces the implemented velocity and slip estimators.

Two different frames were used: the world frame and the IMU frame. The world frame followed standard cartesian coordinates, with the x-axis pointing to the right, the y-axis pointing up, and the z-axis pointing out from the paper. The IMU frame had the IMU as the origin, the x-axis pointing in the vehicle heading direction, the y-axis pointing to the left of the x-axis, and the z-axis pointing out from the paper. Both frames followed the right-hand coordinate system and are shown in Figure 3.1, where the world frame is shown in red and the IMU frame in blue. Point P is the center of rotation for the vehicle turn.

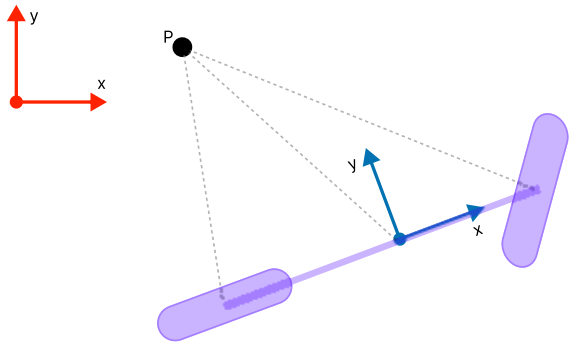


Figure 3.1: Defining the world frame (red) and the IMU frame (blue).

3.1 Simulation environment

A simulation was developed using MATLAB, allowing comparison of different models and methods in a controlled environment. The simulation was used to generate accelerometer, gyroscope, and GNSS data, thereby enabling testing and comparison of different motion models. Furthermore, it facilitated the exploration of various paths and scenarios without needing hardware and a test vehicle. In the initial approach, the path was manually defined, as explained in Section 3.1.1, for straightfor-

ward development and verification of the motion models. Thereafter, the simulation environment was further extended to generate a model-based path that accounts for vehicle motion and includes the lateral slip in Section 3.1.2.

3.1.1 Manual path simulation without slip

To simulate the data signals of a gyroscope, accelerometer, and GNSS measurements, the initial task was to create a simulated path that the vehicle had driven. The first approach was to generate discrete points along a path from which velocity, acceleration, and angular velocity were derived. To produce the points, two functions were created: one to generate points in a straight line and one to generate curves.

The function used to generate straight segments took the start position, heading, initial velocity, final velocity, duration, and timestep as inputs. The distance between the points was calculated using the MATLAB function linspace to linearly space the points, given the initial and final velocity. Thereafter, the positions were updated based on the starting position and the calculated distances. The function outputted the discrete points in the entire path with the corresponding heading for each point, together with the final velocity. The function for curves worked similarly, taking start position, radius, turn direction, initial heading, initial velocity, final velocity, duration, and timestep as inputs. Moreover, it outputted the same variables as the function generating straight segments. Using the initial position, turn direction, radius, and heading, the center coordinates of the turn were calculated to update the position and heading. These two functions were utilized to generate desired paths. The first function used in a path received initial velocity and initial position as input, and thereafter, the next function received the last point and final velocity from the previous function used and continued the path.

Given the discrete points p of the path, the velocity v and angular velocity ω were calculated using the central difference method, as shown in Equation 3.1, where n was the number of points, θ was the heading and Δt was the time between two time steps.

$$v_i^d = \frac{p_{i+1} - p_{i-1}}{2 \cdot \Delta t}, \qquad \omega_i^d = \frac{\theta_{i+1} - \theta_{i-1}}{2 \cdot \Delta t}, \qquad i \in \{2, 3, \ldots, n-1\} \tag{3.1}$$

The last point was calculated using backward difference, as shown in Equation 3.2.

$$v_j^d = \frac{p_j - p_{j-1}}{\Delta t}, \qquad \omega_j^d = \frac{\theta_j - \theta_{j-1}}{\Delta t}, \qquad j \in \{n\} \tag{3.2}$$

Thereafter, using the calculated velocities, the acceleration was derived using the methods above, as shown in Equation 3.3.

$$a_i^d = \frac{v_{i+1} - v_{i-1}}{2 \cdot \Delta t}, \qquad a_j^d = \frac{v_j - v_{j-1}}{\Delta t}, \qquad i \in \{2, 3, \ldots, n-1\},\ j \in \{n\} \tag{3.3}$$

The generated data was combined into a matrix that represented the ground truth for the simulation in the world frame. To transform the data into the local IMU

frame, an inverse rotation was applied based on the heading angle. The rotation matrix was defined according to Equation 3.4. Thereafter, sensor data was generated as shown in Equation 3.5.

$$R = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$\text{Generated sensor data} = \begin{cases} v_i^{IMU} = R^T \cdot v_i^d \\ a_i^{IMU} = R^T \cdot a_i^d \\ \omega_i^{IMU} = R^T \cdot \omega_i^d \end{cases}, \qquad i \in \{1, 2, \ldots, n\} \tag{3.5}$$

This approach to data generation provided both a predefined path and simulated sensor data used to present the vehicle's movement. It served as a simplified model for understanding vehicle behavior. However, forces and slip effects were disregarded.

3.1.2 Model based path simulation with slip dynamics

The second approach for path generation utilized a motion model to produce the position points and derive the sensor data like the previous method. This approach employed a nonlinear four-degrees-of-freedom vehicle dynamics model developed by Mendes et al. [31], which took steering angle and longitudinal force as inputs. Steering inputs were generated using a basic steering controller that updated the angle based on the previous value and a predefined sequence, given the simulation timestep. Force inputs were controlled via a proportional controller designed to track a specified reference velocity.

The model had position, yaw angle, roll angle, velocity, slip angle, yaw rate, and roll rate as states. A MATLAB solver was used by the simulator to integrate the differential equations and to calculate each timestep. By having the position for each timestep, the sensor data was generated identically to the path generation without a model. This method was used to create the positions, influenced by the vehicle dynamics and slip modeling since the positions were updated using the slip angle and the velocity, shown in equation 3.6 and 3.7. In the equation, V is the velocity state, α is the slip angle, and θ_z is the heading.

$$\dot{x} = V \cdot \cos(\alpha + \theta_z) \tag{3.6}$$

$$\dot{y} = V \cdot \sin(\alpha + \theta_z) \tag{3.7}$$

The velocity, V, was updated using the lateral and longitudinal forces. The lateral forces F_y, which affected the slip, were derived using the Pacejka magic formula [31],

while the longitudinal forces F_x were produced by the velocity controller. The force dependency at each tire, denoted as rear left (rl), rear right (rr), front left (fl), and front right (fr), is illustrated in Figure 3.2. To allow for testing the motion without slip, the slip update was set to zero when slip, α, was supposed to be disregarded. The slip angles are also visualized in Figure 3.2. The model had four degrees of freedom, where it could move in the xy-plane, rotate around the z-axis (yaw), and rotate around the x-axis (roll).

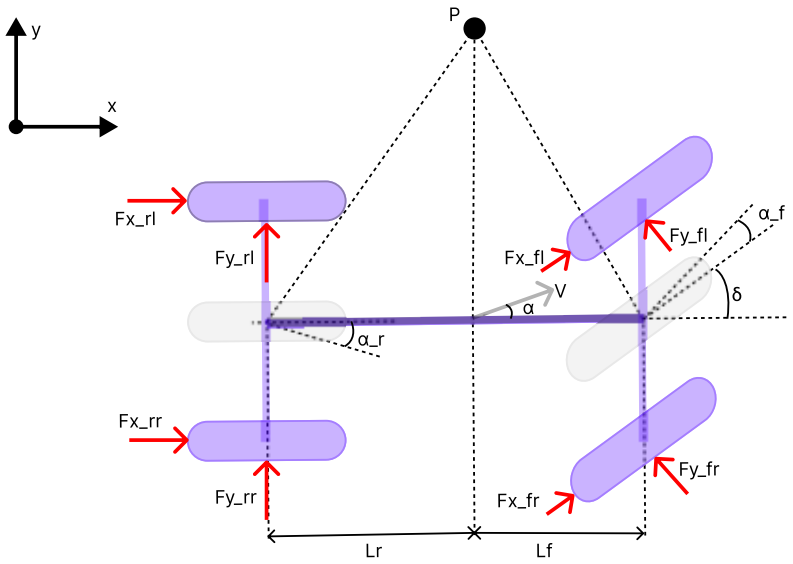


Figure 3.2: Showing the 4 degrees of freedom model including forces affecting the slip and the slip angles. The vehicle turns around the center of rotation point P.

The vehicle position, represented by the x and y states, was defined in the center of mass, and the distances from the center of mass to the front and rear axles were specified in the model. Sampling the positions at the center of mass for each timestep generated the path needed to create the sensor data.

3.1.3 Sensor simulation

The data generated using either method provided the accelerations, angular velocities, and positions given the vehicle movement. This was considered the ideal sensor data free from any noise, biases, or interference. Simulating with this sensor data allowed the models to be validated. Measurement noise was added to the sensor data to simulate the disturbances typically produced by physical sensors, thereby making the simulations more realistic.

The added noise was defined as zero-mean Gaussian with standard deviations specified in Table 3.1. A bias term, $\mathbf{a}_b$, and noise, $\mathbf{a}_\epsilon$, were added to the accelerometer data. For most tests, the accelerometer bias was set to zero, as handling bias was outside the scope of this project. However, it was used to investigate if the velocity estimator could compensate for bias in its estimation. The gyroscope and GNSS data included only noise without any added bias. GNSS data were sampled at a

lower frequency than the IMU sensors to reflect the slower update rate of GNSS readings in reality [32]. The accelerometer and gyroscope were sampled at 100Hz, while GNSS was sampled at 1Hz. The noise addition can be seen in Equations 3.8 to 3.11, where $\mathbf{a}$ represents the accelerometer measurements, $\boldsymbol{\omega}$ the gyroscope measurements, $\mathbf{p}$ the GNSS position, and θ the GNSS heading.

$$\mathbf{a}_i = a_i^{IMU} + \mathbf{a}_b + \mathbf{a}_\epsilon, \qquad \mathbf{a}_\epsilon \sim \mathcal{N}(0, \sigma_a),\ \mathbf{a}_b \in \{0, 0.01\},\ i \in \{1, 2, 3, \ldots, n\} \tag{3.8}$$

$$\omega_i = \boldsymbol{\omega}_i^{IMU} + \boldsymbol{\omega}_\epsilon, \qquad \boldsymbol{\omega}_\epsilon \sim \mathcal{N}(0, \sigma_\omega) \tag{3.9}$$

$$\mathbf{p}_k = p_k^{\text{true}} + \mathbf{p}_\epsilon, \qquad \mathbf{p}_\epsilon \sim \mathcal{N}(0, \sigma_p),\ k \in \{1, 100, 200, \ldots, n\} \tag{3.10}$$

$$\theta_k = \theta_k^{\text{true}} + \theta_\epsilon, \qquad \theta_\epsilon \sim \mathcal{N}(0, \sigma_\theta) \tag{3.11}$$

Table 3.1: Added sensor noise to simulate realistic behavior.

Sensor	**Type of noise**	**Standard deviation**, σ	**Unit**
Accelerometer	Acceleration	0.1	m/s^2
Gyroscope	Angular velocity	0.0017	rad/s
GNSS	Position X	0.05	m
	Position Y	0.05	m
	Altitude	0	m
	Heading	0.0175	rad

The standard deviations listed in Table 3.1 were selected in consultation with engineers in the field. The altitude deviation was set to zero to constrain the simulation to the XY-plane, simplifying the analysis while maintaining the flexibility to expand to three dimensions in future work. GNSS noise levels were chosen to approximate the performance of RTK-enabled systems and were therefore modeled as normal distributed white noise. In contrast, GNSS measurements without RTK typically exhibit random walk behavior. However, in this work, only RTK-based GNSS measurements were considered.

3.1.4 Simulation loop

The simulation loop is shown in Algorithm 1. It utilized the generated sensor data to evaluate different motion models. After initializing the system with predefined initial conditions, it iterated through the sensor data using an EKF to update the vehicle dynamics and state estimates.

At each timestep, the prediction step was performed using the motion models to predict how the states were expected to evolve. The update step was executed only when GNSS measurements were available, incorporating the measurements to correct the predicted states. Additionally, a slip angle measurement was calculated, and the parameters used in the slip estimation were updated accordingly, as described in Section 3.4.

Algorithm 1 Simulation loop

Initialize motion models: CM, UM, BM
Initialize state vectors x_{CM}, x_{UM}, x_{BM} and covariance matrices P_{CM}, P_{UM}, P_{BM}
Initialize slip estimator and related variables
for $i = 2$ to simulation length **do**
    Predict Step (EKF) for each model:
    for each model in {CM, UM, BM} **do**
        Compute predicted state f, Jacobian F, and process noise Q
        $[x, P] \leftarrow \text{EKFPrediction}(f, F, P, Q)$
    end for
    if GNSS measurement available and not in outage **then**
        Update Step (EKF) for each model:
        for each model in {CM, UM, BM} **do**
            Obtain measurement z, Compute expected measurement h,
            Compute Jacobian H, and noise R
            $[x, P] \leftarrow \text{EKFUpdate}(x, P, z, h, H, R)$
        end for
        Estimate velocity heading from GNSS measurements
        Compute slip angle α_{meas}
        Update slip estimator with α_{meas} and vehicle dynamics
        Estimate slip angle $\hat{\alpha}$
    else
        Estimate slip angle $\hat{\alpha}$
    end if
end for

3.2 Motion models

In order for the prediction step to be as accurate as possible, a reliable model of the vehicle was considered essential. Therefore, three different motion models were developed and tested. These models included the constant acceleration model (CM), the unicycle model (UM), and the bicycle model (BM).

3.2.1 Constant acceleration model

A commonly used model is the CM. In 2D, it was defined with the states shown in Equation 3.12, where x and y represented the position, V_x and V_y were the velocities and θ_z was the heading angle. All states were expressed in the world frame, and the model allowed the vehicle to move freely in all directions.

$$\boldsymbol{X}_{CM} = \begin{bmatrix} x \\ y \\ V_x \\ V_y \\ \theta_z \end{bmatrix} \tag{3.12}$$

The discrete-time state transition function (DST) for the model is presented in Equation 3.13, and the continuous time state transition function (CST) is provided in Equation 3.14.

$$\mathrm{DST}(\boldsymbol{X}_{CM}) = \begin{cases} x + V_x \cdot \Delta t + 0.5 \cdot a_x \cdot \cos(\theta_z) - a_y \cdot \sin(\theta_z) \cdot \Delta t^2 \\ y + V_y \cdot \Delta t + 0.5 \cdot a_x \cdot \sin(\theta_z) + a_y \cdot \cos(\theta_z) \cdot \Delta t^2 \\ V_x + a_x \cdot \cos(\theta_z) - a_y \cdot \sin(\theta_z) \cdot \Delta t \\ V_y + a_x \cdot \sin(\theta_z) + a_y \cdot \cos(\theta_z) \cdot \Delta t \\ \theta_z + \omega_z \cdot \Delta t \end{cases} \tag{3.13}$$

$$\mathrm{CST}(\boldsymbol{X}_{CM}) = \begin{cases} V_x \\ V_y \\ a_x \cdot \cos(\theta_z) - a_y \cdot \sin(\theta_z) \\ a_x \cdot \sin(\theta_z) + a_y \cdot \cos(\theta_z) \\ \omega_z \end{cases} \tag{3.14}$$

These equations describe the temporal evolution of the system states based on prior state values and the corresponding measured sensor data. The measurements were the accelerations a_x, a_y, and the angular velocity around the z-axis ω_z. The reason both the DST and CST were used was because the DST was needed to calculate the Jacobian in Equation 3.15, as the EKF expects a Jacobian based on a discrete system.

$$\boldsymbol{F} = \frac{\partial(DST(X_{cm}))}{\partial X_{cm}} \tag{3.15}$$

However, it was considered more accurate to simulate in real-time, thereby, the CST was used in Equation 3.16 to simulate the system, where ode15s is a differential equation solver from MATLAB [33]. The corresponding Equations 3.15 and 3.16 were the same for all models and will not repeated.

$$\boldsymbol{f} = \mathrm{ode15s}(CST(X_{cm}), [0, \Delta t], X_{cm}) \tag{3.16}$$

The EKF requires a process noise covariance matrix, $\boldsymbol{Q}_{CM}$, in order to calculate the certainty of the prediction. To handle different levels of noise, $\boldsymbol{Q}_{CM}$ was defined based on the impact of sensor noise on the model states. Therefore, it depended on the noise level as an input. The covariance matrix, $\boldsymbol{Q}_{CM}$, was automatically tuned when different noise levels were used. It was assumed that the states were independent and therefore, $\boldsymbol{Q}_{CM}$ was defined as a diagonal matrix as shown in Equation 3.17, where a_{mv} was the accelerometer measurement variance and ω_{mv} was the gyroscope measurement variance.

$$\boldsymbol{Q}_{CM} = \text{diag}\left(\begin{bmatrix} 0.5 \cdot a_{mv} \cdot \Delta t^2 \\ 0.5 \cdot a_{mv} \cdot \Delta t^2 \\ a_{mv} \cdot \Delta t \\ a_{mv} \cdot \Delta t \\ \omega_{mv} \cdot \Delta t \end{bmatrix}\right) \tag{3.17}$$

The measurement noise covariance matrix for the GNSS measurements, $\boldsymbol{R}_k^{GNSS}$, was created in a similar way, where the position measurement variance was used for the x and y measurements, and the heading measurement variance was used for the θ_z measurement. The resulting diagonal matrix $\boldsymbol{R}_k^{GNSS}$ can be seen in Equation 3.18, where p_{mv} was the position measurement variance and θ_{mv} was the heading measurement variance.

$$\boldsymbol{R}_k^{GNSS} = \text{diag}\left(\begin{bmatrix} p_{mv} \\ p_{mv} \\ \theta_{mv} \end{bmatrix}\right) \tag{3.18}$$

This method was used for the measurement noise covariance matrix for all models' GNSS measurements and will, therefore, not be repeated.

3.2.2 Unicycle model

Another implemented model was the UM. In 2D, it was defined with the four states, shown in Equation 3.19, where x and y represented the position, V was the velocity, and θ_z was the heading angle. The position and heading were expressed in the world frame, while the velocity was defined as a scalar describing the vehicle velocity in the heading direction.

$$\boldsymbol{X}_{UM} = \begin{bmatrix} x \\ y \\ V \\ \theta_z \end{bmatrix} \tag{3.19}$$

Although the UM did not incorporate detailed vehicle dynamics, it assumed that velocity was constrained to the vehicle's heading direction. This constraint was considered a more realistic vehicle behavior compared to the CM, which allowed unrestricted movement in all directions. The DST for the model is presented in Equation 3.20, and the CST is provided in Equation 3.21. As evident from the equations, only the x-axis component of the accelerometer was used to update the velocity and, in turn, the position.

$$\mathrm{DST}(\boldsymbol{X}_{UM}) = \begin{cases} x + V \cdot \cos(\theta_z) \cdot \Delta t + 0.5 \cdot a_x \cdot \cos(\theta_z) \cdot \Delta t^2 \\ y + V \cdot \sin(\theta_z) \cdot \Delta t + 0.5 \cdot a_x \cdot \sin(\theta_z) \cdot \Delta t^2 \\ V + a_x \cdot \Delta t \\ \theta_z + \omega_z \cdot \Delta t \end{cases} \tag{3.20}$$

$$\mathrm{CST}(\boldsymbol{X}_{UM}) = \begin{cases} V \cdot \cos(\theta_z) \\ V \cdot \sin(\theta_z) \\ a_x \\ \omega_z \end{cases} \tag{3.21}$$

Similar to the CM, the process noise covariance matrix, $\boldsymbol{Q}_{UM}$, was derived based on the accelerometer measurement variance, a_{mv}, and the gyroscope measurement variance, ω_{mv}. The diagonal matrix $\boldsymbol{Q}_{UM}$ can be seen in Equation 3.22.

$$\boldsymbol{Q}_{UM} = \mathrm{diag}\left(\begin{bmatrix} 0.5 \cdot a_{mv} \cdot \cos(\theta_z)^2 \cdot \Delta t^2 \\ 0.5 \cdot a_{mv} \cdot \sin(\theta_z)^2 \cdot \Delta t^2 \\ a_{mv} \cdot \Delta t \\ \omega_{mv} \cdot \Delta t \end{bmatrix} \right) \tag{3.22}$$

A limitation of this model was its inability to account for slip. When slip occurred, the model continued to assume motion in the heading direction, leading to inaccurate position estimates. Additionally, centripetal acceleration was misinterpreted as longitudinal acceleration, resulting in an incorrect velocity estimate. Due to these shortcomings, a more advanced model was required to accurately capture the vehicle's dynamics under such conditions.

3.2.3 Bicycle model

To compensate for the slip issue, a BM was created. The states for this model are provided in Equation 3.23 and were the same as for the UM.

$$\boldsymbol{X}_{BM} = \begin{bmatrix} x \\ y \\ V \\ \theta_z \end{bmatrix} \tag{3.23}$$

To further clarify the issues with the UM, Figure 3.3 shows the measured acceleration from the IMU in red and the accelerations that the center of gravity was assumed to experience in blue.

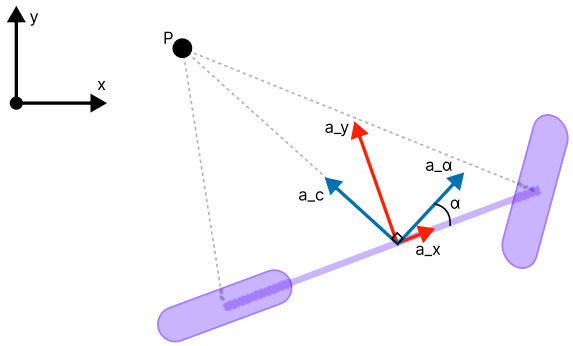


Figure 3.3: Accelerations measured from the IMU (red) and the assumed true accelerations during slip (blue).

It was clear that the IMU measurements needed to be compensated for slip in order to gain the correct values. This was done according to Equation 3.24, where α is the slip angle and ω_z is the angular velocity around the z-axis.

$$\boldsymbol{a}_\alpha = \boldsymbol{a}_x + \boldsymbol{a}_y - \boldsymbol{a}_c = \begin{bmatrix} a_x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix} - \begin{bmatrix} V \cdot \omega_z \cdot \cos(\frac{\pi}{2} + \alpha) \\ V \cdot \omega_z \cdot \sin(\frac{\pi}{2} + \alpha) \\ 0 \end{bmatrix} \tag{3.24}$$

The velocity state of the BM was in the direction of the slip. However, it was still only updated with the longitudinal component of $\boldsymbol{a}_\alpha$ as can be seen in the DST in Equation 3.25 and the CST in Equation 3.26. The assumption has been made that the centripetal acceleration was the product of the velocity multiplied by the angular velocity in order to calculate $\boldsymbol{a}_\alpha$. In the ideal case with no slip, this is true, but with slip, it became an approximation due to the slip dynamics.

$$\mathrm{DST}(\boldsymbol{X}_{BM}) = \begin{cases} x + V \cdot \cos(\alpha + \theta_z) \cdot \Delta t \\ y + V \cdot \sin(\alpha + \theta_z) \cdot \Delta t \\ V + a_{\alpha,x} \cdot \Delta t \\ \theta_z + \omega_z \cdot \Delta t \end{cases} \tag{3.25}$$

$$\mathrm{CST}(\boldsymbol{X}_{BM}) = \begin{cases} V \cdot \cos(\alpha + \theta_z) \\ V \cdot \sin(\alpha + \theta_z) \\ a_{\alpha,x} \\ \omega_z \end{cases} \tag{3.26}$$

Relying solely on the longitudinal component of $\boldsymbol{a}_\alpha$ removed some of the actual dynamics. If both components of $\boldsymbol{a}_\alpha$ were used to update the velocity, the lateral displacement would have been included. Since this velocity does not contribute to centripetal acceleration, it would have introduced errors in its calculation. For small slip angles, the acceleration from the vehicle was expected to have its primary part

in the longitudinal direction and only a minor part in the lateral direction. With the current method, omitting the lateral part was expected to build up small errors, but it was still estimated to be more accurate than including it.

Simply looking at the Equations 3.25 and 3.26, it appeared that this model did not contain more information about the vehicle than the UM, but this was not the case because slip angle α was estimated based on a vehicle model. The slip estimation will be described later in Section 3.4.

The process noise covariance matrix for the BM, $\boldsymbol{Q}_{BM}$, was very similar to the one for the UM, with the only difference being how the accelerometer measurement variance, a_{mv}, was rotated for the position covariance. The diagonal matrix $\boldsymbol{Q}_{BM}$ can be seen in Equation 3.27.

$$\boldsymbol{Q}_{BM} = \text{diag}\left(\begin{bmatrix} 0.5 \cdot a_{mv} \cdot \cos(\alpha + \theta_z)^2 \cdot \Delta t^2 \\ 0.5 \cdot a_{mv} \cdot \sin(\alpha + \theta_z)^2 \cdot \Delta t^2 \\ a_{mv} \cdot \Delta t \\ \omega_{mv} \cdot \Delta t \end{bmatrix}\right) \tag{3.27}$$

3.3 Velocity estimation

Relying on the IMU sensors to navigate when the GNSS measurements were unavailable led to one of the main challenges of dead reckoning: error accumulation. Integrating the accelerometer measurements to determine velocity built up an error due to the accelerometer noise. To mitigate this issue, a parameter-trained velocity estimator was developed. The idea was to recursively train parameters with GNSS measurements in a Kalman filter that could be used to linearly approximate the velocity during GNSS outages. A linear equation was used as a simple approach to estimating the velocity, v_{est}, as shown in Equation 3.28.

$$v_{est} = \hat{k} \cdot v_{pred} + \hat{m} \tag{3.28}$$

The parameters $\hat{k}$ and $\hat{m}$ were defined as the states of the Kalman filter and initialized to one and zero, respectively, as in Equation 3.29.

$$x_{\text{vel}} = \begin{bmatrix} \hat{k} \\ \hat{m} \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.29}$$

For the filter, the matrices shown in Equation 3.30 were used.

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 0.5^2, \quad F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} v_{pred} & 1 \end{bmatrix} \tag{3.30}$$

When GNSS was available, the velocity was updated in the INS Kalman filter using the GNSS measurement. The updated velocity was used in the parameter estimation Kalman filter as the measurement, which resulted in the innovation from

the difference between GNSS measured velocity and parameter estimated velocity prediction, as seen in Equation 3.31.

$$y_{vel} = v_{gnss} - (v_{pred} \cdot k + m) \tag{3.31}$$

Following the Kalman filter update step described in Section 2.1, the parameters of x_{vel} were updated. The idea was thereafter to use the trained parameters when GNSS is unavailable to improve the velocity estimate from the prediction.

3.4 Slip estimation

In order to estimate the slip of the vehicle, Equation 2.18 was rewritten to Equation 3.32, assuming that a_y could be approximated by $\omega_z \cdot V$ when in reality this was not entirely accurate when there was slip. This assumption was, however, considered to be a better approximation than using the estimated velocity like in Equation 2.18. The equation contained two unknown variables, L_r and K_α, the estimate for V from the model was used, and ω_z and a_y were measured.

$$\alpha = \omega_z \cdot \frac{L_r}{V} - K_\alpha \cdot a_y \tag{3.32}$$

In the report by Takikawa et al. [1], the equation was rewritten to describe the slip at the rear wheel instead of the center of gravity by having the L_r component removed from Equation 2.17, however, it was kept in this thesis. To determine the K_α parameter, collected data were used by Takikawa et al. in order to perform a linear regression to find the K_α that resulted in the best performance. However, with that method, K_α was kept constant and was not adapted to current conditions. Furthermore, a large amount of data was required.

To avoid those issues, a Kalman filter was used to tune the parameters $\hat{L}_r$ and $\hat{K}_\alpha$ when GNSS measurements were available, and the parameters were then used for slip estimation when GNSS was unavailable. The matrices for the filter are provided in Equation 3.33.

$$x_{\text{slip}} = \begin{bmatrix} \hat{L}_r \\ \hat{K}_\alpha \end{bmatrix}, \qquad F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad H = \begin{bmatrix} \frac{\omega}{V} & -a_y \end{bmatrix} \tag{3.33}$$

When GNSS was available, a measurement of the slip angle α was required in order to tune the parameters. However, the slip angle could not be directly measured by any of the sensors used. Considering that the slip angle is the difference between the heading angle and the velocity direction, a method to estimate the direction of the velocity based on previous GNSS measurements was developed. A selected amount of previous GNSS points was taken, and a line-fitting method from MATLAB was

used to fit a line to these points to estimate the course over ground. The line-fitting method was chosen depending on the measured angular velocity. For straight lines, piecewise cubic Hermite interpolating polynomial was used to ensure that the fitted line passed through the points without oscillations [34]. For curved lines, cubic spline interpolation was applied to provide a smooth curve that accommodates changes in direction [35]. When a line had been fitted to the points, the angle between the last point and the previous point was calculated and selected as the estimate for velocity direction.

The difference between the estimated velocity direction and the measured GNSS heading was used as a measurement for α. This value was compared to the estimated α from equation 3.32 and the parameters $\hat{L}_r$ and $\hat{K}_\alpha$ were tuned to make the measurement and prediction as similar as possible.

3.5 Evaluation metrics

To assess the performance of the models and estimators for the different simulations, the root mean square error (RMSE) was calculated and compared. By running the simulation 10 times, the mean of the RMSE could be calculated for the models, which was used to show the general performance rather than the performance for a specific noise sequence. The uncertainty of the models was also compared to investigate how the variance of the predicted position from the models grew. The covariance matrix for the models was obtained from the EKF, and three standard deviations (3-sigma) was plotted with the obtained covariance for the x- and y-position. This showed the region where approximately 99.7% of the true positions were expected to be within.

The robustness of the models was also compared to investigate how much they were affected by different sensor noises. To check this, the noise for one of the sensors was changed between 0.01 and 100, increasing the noise by an order of magnitude for each simulation. The results highlighted the models' ability to maintain performance under increasing noise and revealed which measurements had the most significant impact on model stability and accuracy.

The slip estimator was evaluated by investigating the difference between the estimated and the true slip angle for simulations with different amounts of slip. To introduce varying levels of slip, the reference velocity in the simulation was adjusted. This provided a simple method for inducing more slip. However, some of the reference velocities used were not representative of typical everyday driving conditions. The velocity estimator was evaluated through observation of the difference between the true velocity, the predicted velocity from the model, and the velocity estimated from the estimator.

Finally, the models were tested with real data, where the performance was evaluated by looking at the predicted paths and evaluating their reasonability. Here, there was no ground truth, and thus, no values for accuracy could be obtained. This test was done to investigate how the models were affected by real data.

4

Results

This chapter presents the results of testing and comparing the motion models. Tests were done in the two environments presented in Section 3.1 and with real-world data. The robustness of the models was also tested by varying the noise levels for the sensors and evaluating their performance. Furthermore, an uncertainty comparison was made for the models to investigate how well the estimated accuracy reflected reality. In all the tests, the models received the same sensor data. The tests were evaluated using the evaluation metrics described in Section 3.5.

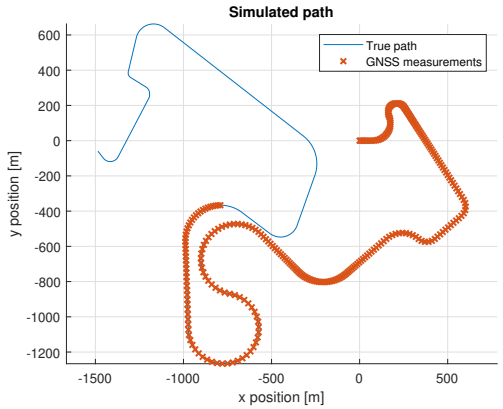
4.1 Estimating the path in different simulation environments

To evaluate the performance of the different models, the accuracy of their respective position estimates was compared in two different simulation environments. To enhance the reliability and validity of the tests, 10 different simulations were run for each scenario, after which the mean position RMSE across these simulations was calculated. The mean provided a more comprehensive measure of overall model performance involving more than just a single noise sequence. The simulation with RMSE values closest to the mean RMSE was plotted to illustrate the most general performance.

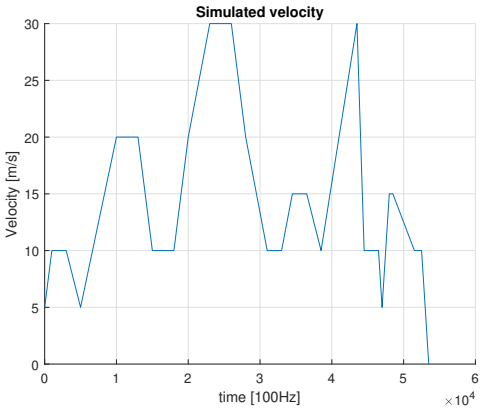
In all tests with access to GNSS measurements, the GNSS data was available until timestep 30000 at a frequency of 1 Hz, and thereafter, the positions were estimated using dead reckoning.

4.1.1 Tests in the environment with manual path simulation and no slip

To test the models in the environment with manual path simulation and no slip, the path shown in Figure 4.1a was created with velocities shown in Figure 4.1b. For the first test, the models ran the entire path with pure dead reckoning and only had knowledge about the initial position.



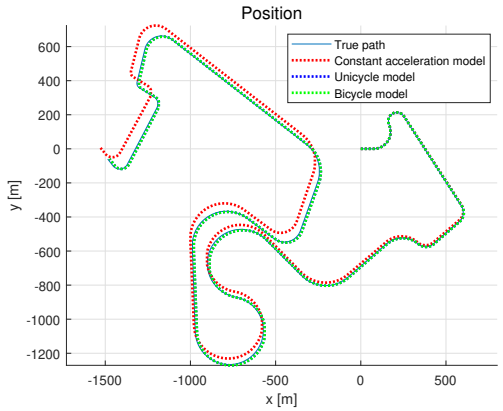
(a) Simulated true path.



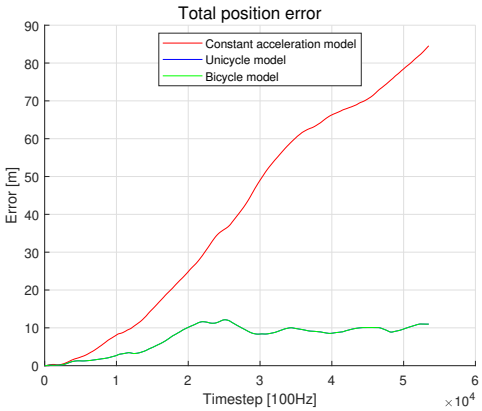
(b) Simulated true velocity.

Figure 4.1: Ground truth for the environment with manual path simulation and no slip.

The RMSE values for the first test can be seen in Table 4.1, and the estimated path is shown in Figure 4.2a, with the position errors at each timestep shown in Figure 4.2b. The UM and BM overlap in the figures. By looking at the estimated paths, it was apparent that the CM performed worse than the other models in this case. This was also clear when looking at the position error, where the CM error increased to a final magnitude approximately eight times the error of the UM and BM. The RMSE table also showed that the mean RMSE for the CM was approximately eight times larger than the other models. Note that compared to the rest of the tests presented later, this test gave a larger RMSE in general. The error had more time to build up since there were not 30000 timesteps with very low errors, reducing the RMSE like in the case when there were GNSS measurements available.



(a) Predictions from the models using dead reckoning.



(b) Position errors for the models using dead reckoning.

Figure 4.2: Results from tests in the environment with manual path simulation and no slip.

Table 4.1: RMSE values for 10 simulations in the environment with manual path simulation and no slip, without GNSS.

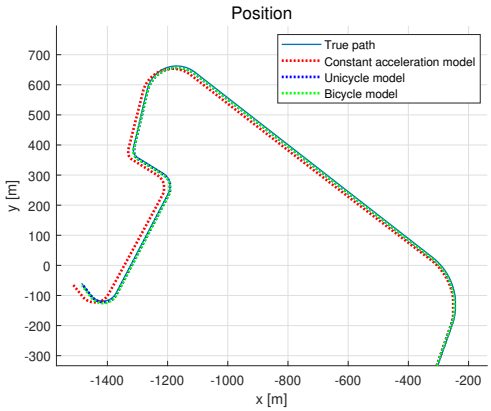
	RMSE										Mean
UM	8.2691	11.7832	12.8554	7.4013	5.1544	3.2058	6.9136	5.5017	9.905	13.273	7.7344
CM	48.497	27.9747	37.7592	29.7204	58.205	83.3432	71.6662	35.6219	72.7688	47.8945	48.0579
BM	8.2691	11.7832	12.8554	7.4013	5.1544	3.2058	6.9136	5.5017	9.905	13.273	7.7344

It can be seen in Table 4.1 that the UM and the BM performed the same, and the CM had a more pronounced error. This was because the UM and BM only had one velocity state and turns with the help of the gyroscope instead of using the centripetal acceleration. Since a gyroscope has a higher accuracy than an accelerometer in general, this was less affected by noise, as described in Section 2.4. When driving along a straight path, the slight drift from the gyroscope gave a reduced error compared to the resulting lateral velocity error for the CM from the accelerometer noise. Since the GNSS measurements were unavailable, the slip parameters were never trained, and thereby the estimated slip was always zero. Looking at the CST for the models in Equations 3.21 and 3.26, they became the same when the slip was zero.

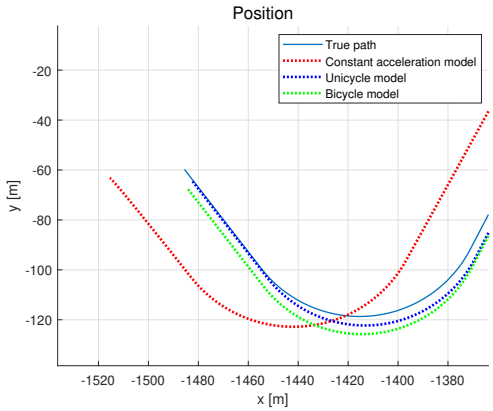
In the second test, the models were given the available GNSS measurements shown in Figure 4.1a. This gave the RMSE values shown in Table 4.2 and the estimated paths during the dead reckoning shown in Figure 4.3a, with a zoomed-in view at the end of the path shown in Figure 4.3b. The position errors are shown in Figure 4.3c. Looking at the estimated paths, especially at the end of the path, it was shown that for this noise sequence, the prediction from the UM was closest to the true path, followed by the BM and the CM. The errors showed that UM and BM had a similar performance most of the time, and the UM achieved a better performance towards the end. The CM had a significantly higher error compared to the other models. The RMSE table reflected the results from the plots where the UM had the lowest mean RMSE, followed by the BM, and the CM had the highest mean.

Table 4.2: RMSE values for 10 simulations in the environment with manual path simulation and no slip, with GNSS.

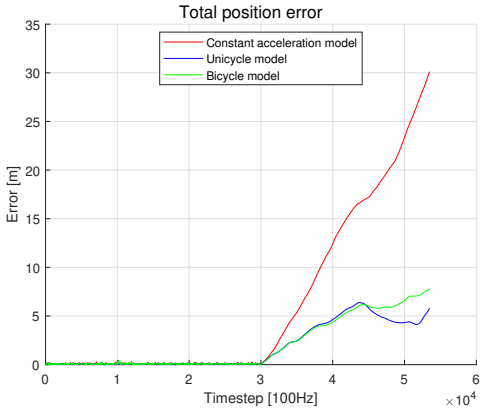
	RMSE										Mean
UM	4.0339	2.863	2.9639	5.4657	4.8885	1.9872	2.3817	1.4041	2.8911	1.5086	2.7691
CM	18.8277	10.7125	13.0168	6.6721	14.5658	7.549	14.5046	5.1185	8.0895	4.7414	9.4171
BM	2.2499	3.321	13.0992	4.0987	12.2408	6.7788	2.5178	1.6958	18.6536	11.9389	5.6239



(a) Predictions from the models during the time of GNSS outage.



(b) Predictions from the models at the end of the path.



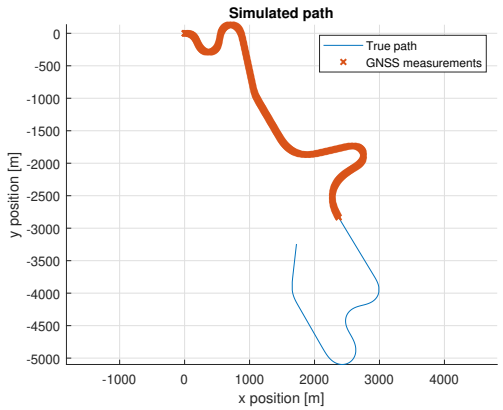
(c) Position errors for the models.

Figure 4.3: Results from the models using GNSS measurements and dead reckoning in the environment with manual path simulation and no slip.

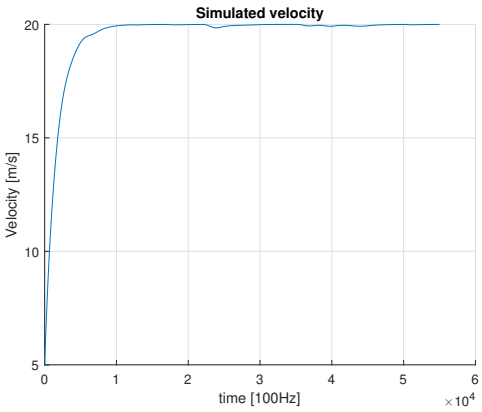
The most significant difference between the results in the two scenarios was the BM and UM comparison. In the first test, they were the same since the slip was never updated when GNSS was unavailable. However, in the second test, the UM performed better in general. In this scenario, the measurement for slip came from the line-fitting method, which tended to produce noisy results when the GNSS measurements were noisy. The measurements caused training of the slip parameters. Although the values remained small, this still led to slip being estimated in the corners, which in turn increased the error.

4.1.2 Tests in the model based simulation environment

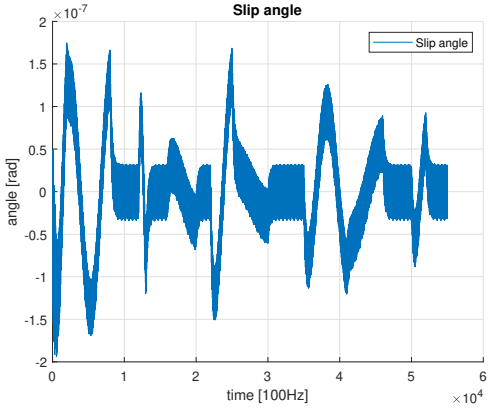
For the model-based simulation environment, the models received the available GNSS measurements and were tested with and without slip. In the case when there was no slip, the path can be seen in Figure 4.4a with velocities shown in Figure 4.4b and slip angles shown in Figure 4.4c. It can be noted that despite the scenario not supposed to contain any slip, some still remained. Due to the complexity of the model, it was not easy to completely eliminate the slip. However, the remaining slip was assumed to be negligible.



(a) Simulated true path.



(b) Simulated true velocity.



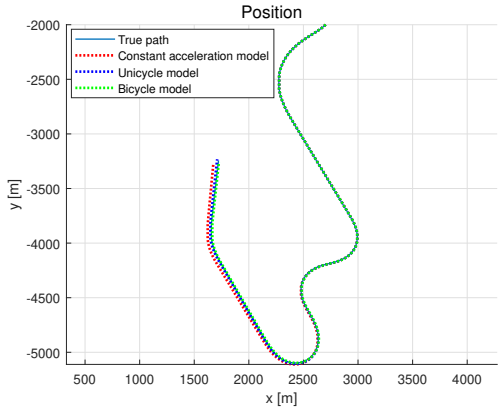
(c) Simulated true slip angle.

Figure 4.4: Ground truth for model based simulation environment without slip.

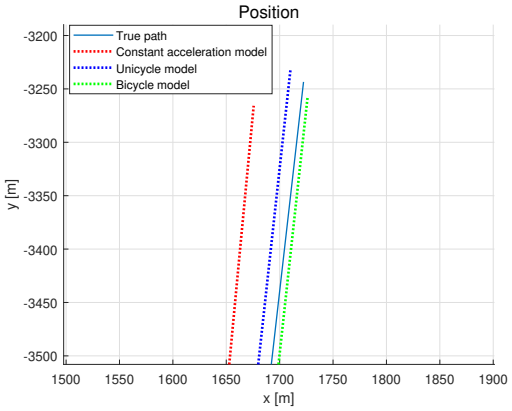
The resulting RMSE values from the simulations can be seen in Table 4.3, where the UM had a slightly lower RMSE than the BM, and both of them had less than half the RMSE compared to the CM. During dead reckoning, the estimated paths can be seen in Figure 4.5a, and the estimated path at the end of the simulation can be seen in Figure 4.5b. The errors for the models during the simulation can be seen in Figure 4.5c. The results showed that both the BM and the UM had a better positioning accuracy than the CM.

Table 4.3: RMSE values for 10 simulations using the model based simulation environment without slip.

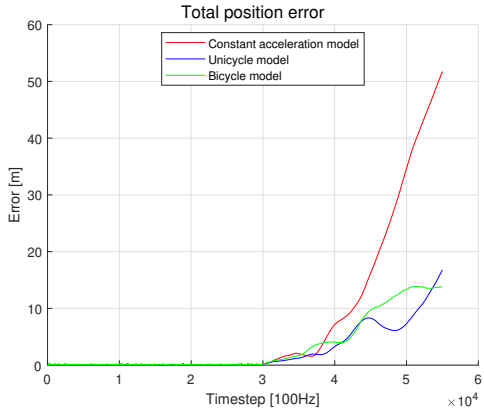
	RMSE										Mean
UM	2.5907	1.5925	5.091	6.3835	4.593	3.5102	5.4995	1.8044	5.4283	3.3833	3.6260
CM	13.6519	12.1028	15.6652	22.802	15.6565	8.3409	14.7757	9.1003	6.5052	10.3417	12.1433
BM	2.5304	1.5945	10.7773	14.7259	5.8206	6.6704	7.3203	1.2365	12.9813	9.2197	5.5339



(a) Predictions from the models during the time of GNSS outage.



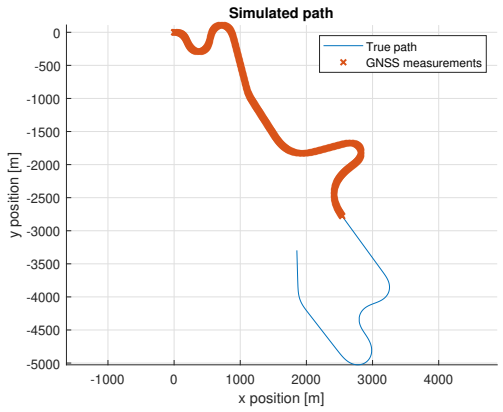
(b) Predictions from the models at the end of the path.



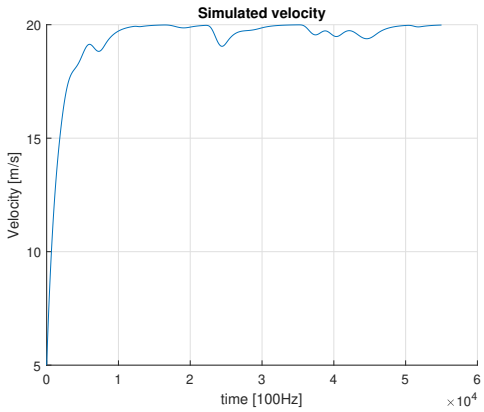
(c) Position errors for the models.

Figure 4.5: Results from the models using GNSS measurements and dead reckoning in the model based simulation environment without slip.

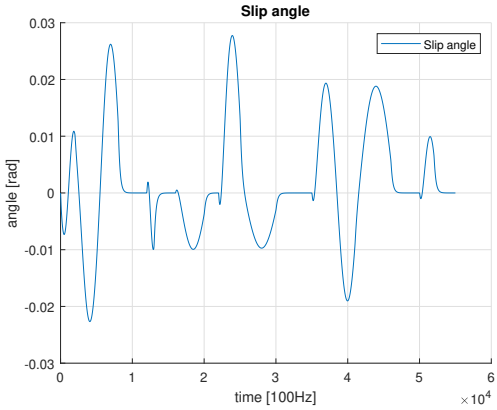
In the case when there was slip, the path can be seen in Figure 4.6a with velocities shown in Figure 4.6b and slip angles shown in Figure 4.6c.



(a) Simulated true path.



(b) Simulated true velocity.



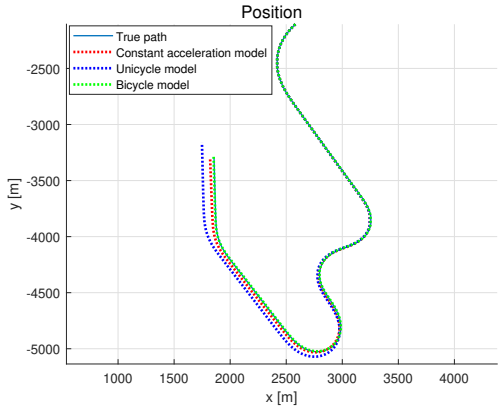
(c) Simulated true slip angle.

Figure 4.6: Ground truth for model based simulation environment with slip.

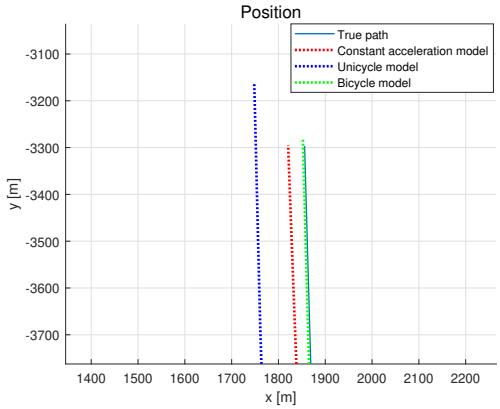
The resulting RMSE values from the simulations can be seen in Table 4.4, the estimated paths during dead reckoning can be seen in Figure 4.7a, and the estimated path at the end of the simulation can be seen in Figure 4.7b. The errors for the models can be seen in Figure 4.7c. The results showed that in this simulation environment, the BM had a better positioning accuracy than the CA model, while the UM had a poor accuracy compared to the other models. The reason the UM had worse accuracy than the others, despite having been the best in the previous cases, was that it assumed no slip. When there are slip dynamics in the simulation, the model obtained an incorrect velocity direction and thus estimated the path in the wrong direction. Furthermore, as explained in Section 3.2.3, the IMU measurements needed to be compensated for the slip angle, this did not happen in the UM, and thus, it incorrectly estimated the velocity.

Table 4.4: RMSE values for 10 simulations using the model based simulation environment with slip.

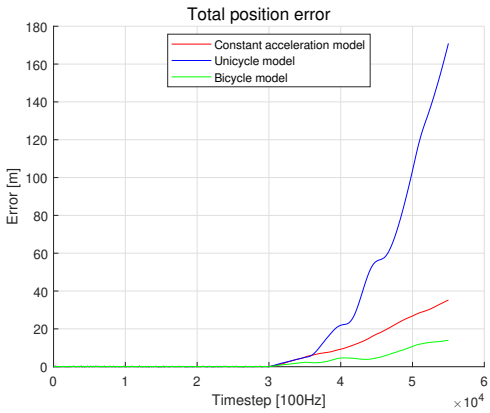
	RMSE										Mean
UM	48.7846	48.9787	53.2886	45.031	52.824	52.5694	52.3073	50.838	44.5091	50.7859	49.8993
CM	14.2075	12.365	15.5216	22.8522	15.7396	8.6012	14.2847	9.2282	6.7522	10.2162	12.2599
BM	3.6928	4.7665	11.9997	13.0779	3.4129	8.6503	10.5275	2.8981	10.9659	10.7866	7.0288



(a) Predictions from the models during the time of GNSS outage.



(b) Predictions from the models at the end of the path.



(c) Position errors for the models.

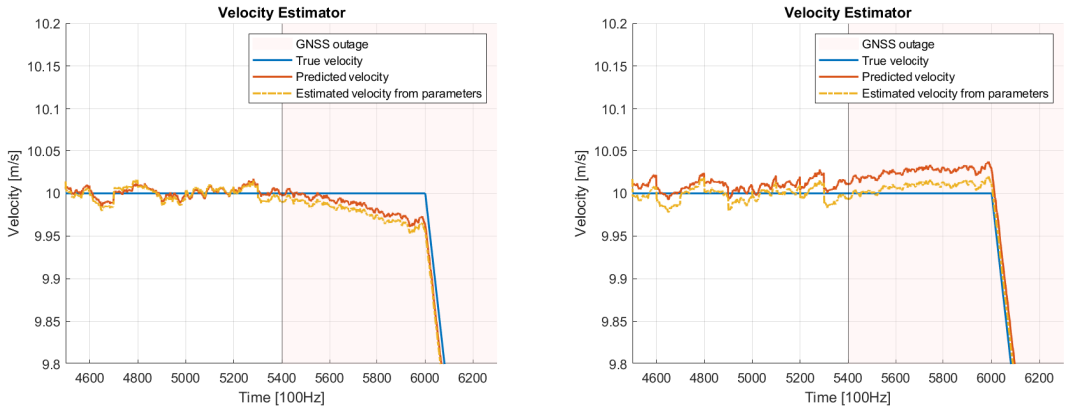
Figure 4.7: Results from the models using GNSS measurements and dead reckoning in the model based simulation environment with slip.

Comparing Table 4.3 and Table 4.4, the CM had approximately the same mean RMSE for both scenarios. The mean RMSE for the UM and BM increased when the slip was added. This showed that when more information about the vehicle was incorporated into the motion model, the predictions were more affected by environmental changes, such as slip addition.

4.2 Velocity estimation

One idea investigated during the project was to create a velocity estimate based on the measurements and Kalman filter trained parameters. This method aimed to update the velocity of the vehicle even without GNSS measurements, described in Section 3.3.

Initial observations demonstrated that the parameters could be trained to compensate for velocity drift when the GNSS signal was lost. However, running the simulations for multiple noise sequences, it became evident that the performance relied on when the GNSS outage occurred and in what direction the velocity had changed before the outage. As illustrated in Figure 4.8a, the estimated velocity was below the predicted velocity. This would be the case until new GNSS measurements were received and the parameters were updated. In the simulation shown in Figure 4.8a, this behavior resulted in a worse velocity estimation than the predicted one.



(a) No sensor noise and no added bias. **(b)** No sensor noise and added bias.

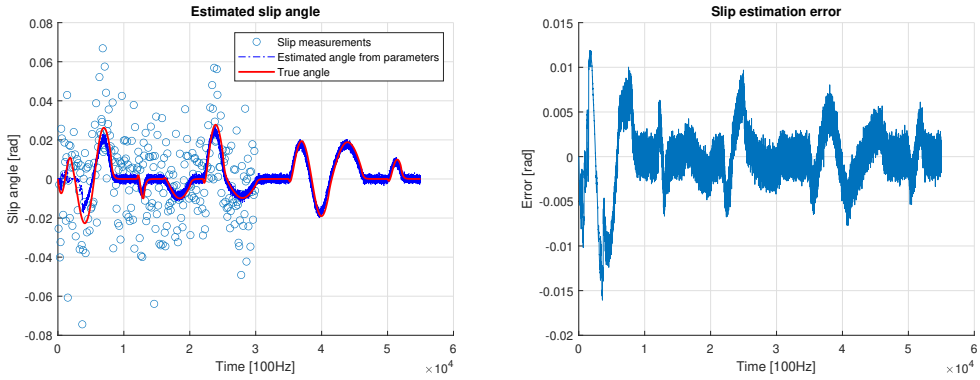
Figure 4.8: Velocity estimation using parameters trained by a Kalman filter, simulated with no added GNSS noise.

When adding a bias to the accelerometer measurements, the velocity estimator was able to compensate, resulting in the prediction being closer to the true velocity for longer GNSS outages, as seen in Figure 4.8b. The estimated velocity always lay below the predicted velocity, causing it to be closer to the true velocity after the GNSS outage occurred. Looking at the time when there were GNSS measurements, however, when the predicted velocity got updated by a GNSS measurement, it initially was better than the estimated velocity. When the error built up from the noise and bias, the estimation became more accurate.

Compensating for biases was outside the scope of this project, and the data was assumed to be filtered and bias compensated. In this case, the velocity estimator did not improve prediction accuracy and was not further investigated.

4.3 Slip estimation

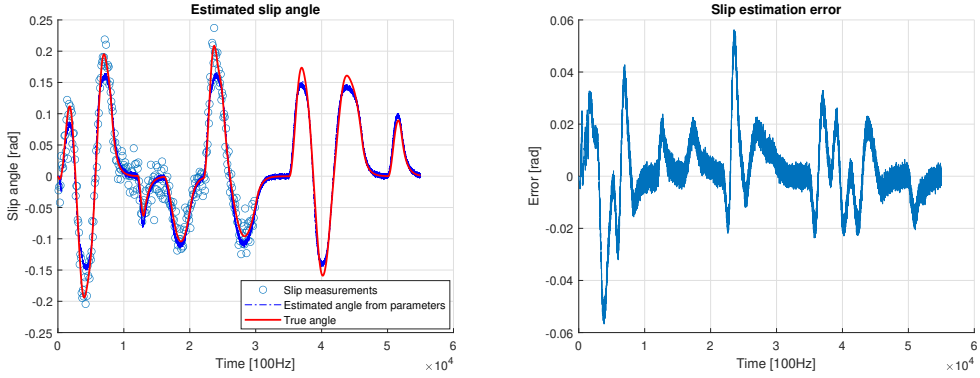
To evaluate the slip estimator, the simulator was provided with the same steering sequence as described in Section 4.1.2, but with varying reference velocities. Although this resulted in different paths across the tests, the consistent steering inputs allowed for a straightforward comparison of the slip angles. The estimator was given GNSS measurements until timestep 30000, after which it relied on the trained parameters for estimation. The first test had 20 m/s as the reference velocity, which gave the results shown in Figure 4.9. At the initial time, the error magnitude was approximately 0.015 radians. However, the parameters had not yet adapted to the slip dynamics, resulting in poorer initial performance when compared with the post-training parameters. After the initial stage, the slip angle could be estimated with an accuracy of approximately 0.01 radians.



(a) Estimated slip angles. **(b)** Error of estimated slip angles.

Figure 4.9: Results from the slip estimator with a reference velocity of 20 m/s.

Simulating with a velocity of 60 m/s gave the estimated slip angles shown in Figure 4.10a with the error shown in Figure 4.10b. The slip angle could be estimated with an accuracy of between 0.05 and 0.06 radians. As mentioned in Section 3.5, this velocity was not reasonable. However, it was selected to vary the amount of slip.



(a) Estimated slip angles. **(b)** Error of estimated slip angles.

Figure 4.10: Results from the slip estimator with a reference velocity of 60 m/s.

The parameters used in the slip estimator for the case with a reference velocity of 20 m/s are illustrated in Figure 4.11. It was evident that the K parameter became certain quickly and remained relatively constant. The L_r parameter, however, had a higher uncertainty that continued to increase. This was because the measurements used to train L_r were at a scale 10^3 smaller than measurements for K, making it learn slower.

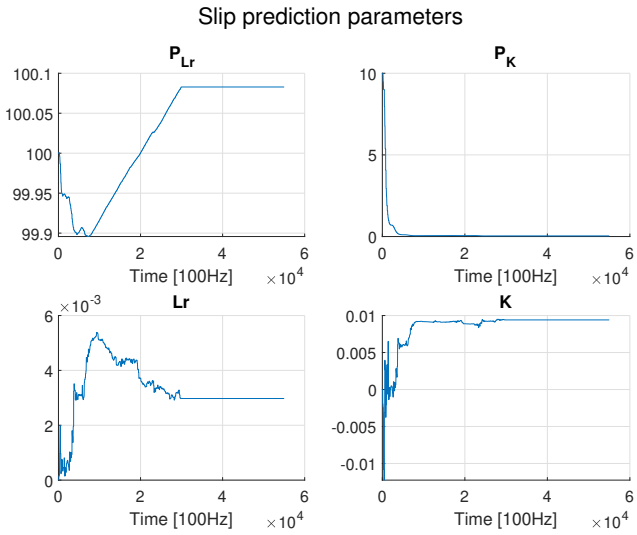


Figure 4.11: Parameters of slip estimator with a reference velocity of 20 m/s.

Initially, K and L_r were set to zero since the estimator should assume no slip in the initial condition. Subsequently the estimator learned about the current road conditions instead of assuming a certain level of initial slip by giving the parameters initial values. Testing having initially set L_r to the specified distance from the rear axis to the center of gravity produced nearly identical results to those initially setting L_r to zero under slip conditions. Testing with no slip, the estimator's performance deteriorated when L_r was not explicitly initialized to zero. This supported the decision to initialize L_r at zero to maintain robustness across varying conditions.

4.4 Robustness

To assess model robustness, the standard deviations of the sensor noise were varied, and the resulting position RMSE from the simulations was compared across models. For all the tests, only the noise for a single sensor was varied, while the rest of the sensors had the noises in Table 3.1. The different models had the standard deviation for the noises as input, which automatically tuned the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices. This was not the case for the slip estimator that had been tuned manually. The simulation used for these tests was the same as in Section 4.1.2, containing slip, and employed the simulation seed that previously yielded RMSE values closest to the mean.

The first test was to vary the accelerometer standard deviation and observe how the models behaved. The results are provided in Table 4.5. It showed that the UM and BM were more robust to accelerometer noise than the CM.

Table 4.5: Position RMSE values for the models with different accelerometer noise standard deviation.

	Acccelerometer noise standarad deviation [m/s^2]				
	0.01	0.1	1	10	100
UM	49.2908	48.9787	50.2532	199.1507	1988.1587
CM	3.8597	12.365	83.8991	814.9634	8127.6296
BM	2.6409	4.7665	32.4117	199.2094	1987.6726

In the next test, the gyroscope noise was instead varied, and the results are shown in Table 4.6. As can be seen in the Table, the UM and BM were more robust than the CM in this case as well.

Table 4.6: Position RMSE values for the models with different gyroscope noise standard deviation.

	Gyroscope noise standard deviation [deg/s]				
	0.01	0.1	1	10	100
UM	50.1199	48.9787	43.3537	93.6772	965.7179
CM	11.6859	12.365	41.1012	423.1073	3117.5251
BM	4.8857	4.7665	7.6461	105.5432	1037.9891

The third test was to vary the GNSS position measurement accuracy, and the results can be seen in Table 4.7. In this scenario, the UM and CM were relatively unaffected, while the BM was more sensitive to high noise levels.

Table 4.7: Position RMSE values for the models with different GNSS position noise standard deviation.

	GNSS position noise standard deviation [m]				
	0.01	0.1	1	10	100
UM	48.3594	49.0928	48.4674	44.0957	62.7025
CM	9.2182	14.1771	8.4947	25.4397	49.1952
BM	4.3011	5.3224	13.1184	33.0739	1310.8877

Finally, a test was conducted in which the accuracy of the GNSS heading measurement was varied, and the results are provided in Table 4.8. For this test, the BM was once more the least robust to higher noise levels. This was because the GNSS heading measurements were used to determine the slip measurements, and when the heading measurements were inaccurate, the slip estimation parameters were incorrectly trained.

Table 4.8: Position RMSE values for the models with different GNSS heading noise standard deviation.

	GNSS heading noise standard deviation [deg]				
	0.01	0.1	1	10	100
UM	45.3821	47.0482	48.9787	48.9742	48.9686
CM	12.7505	12.417	12.365	17.3996	14.8042
BM	6.8143	5.629	4.7665	60.908	657.7465

The results for the different robustness tests show that the CM was the least robust model to IMU disturbances while the BM was the most affected by GNSS disturbances.

4.5 Uncertainty comparison

A test was done to investigate how the uncertainties of the model estimates changed during a period of GNSS outage. The simulation was run with slip, and the results are shown in Figure 4.12. The figure shows that during GNSS outages, the uncertainties for the position estimates from the models increased, and this was because there was no update step in the EKF reducing the covariance.

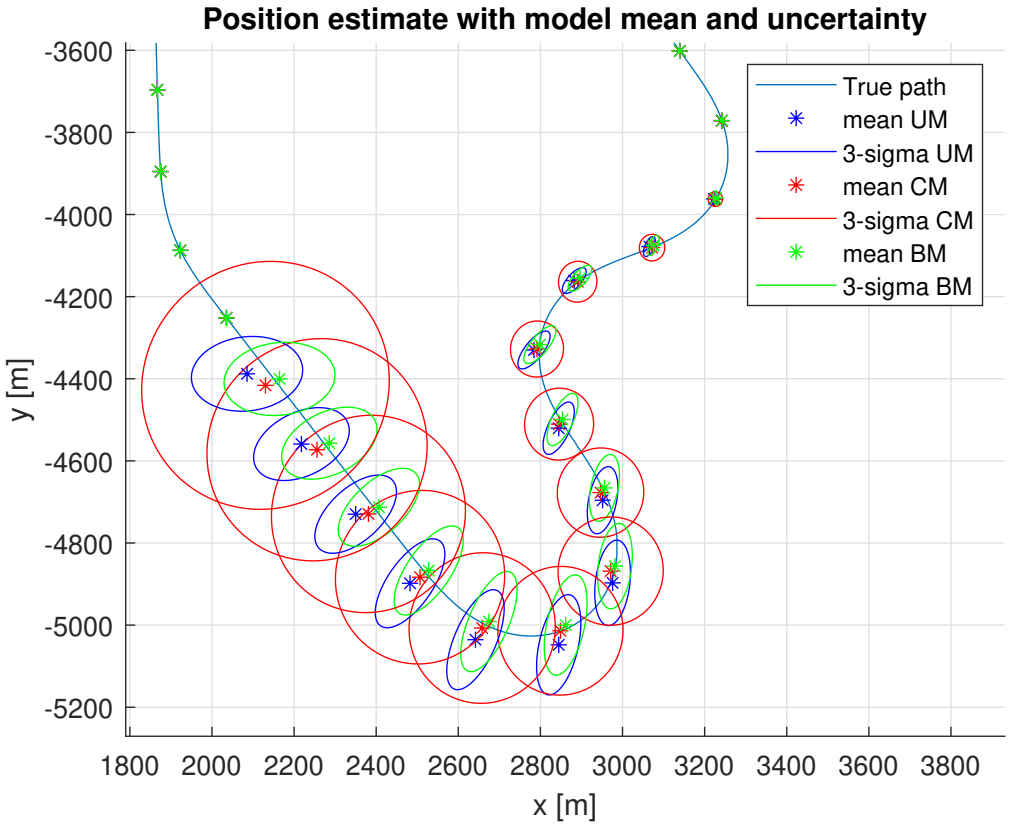
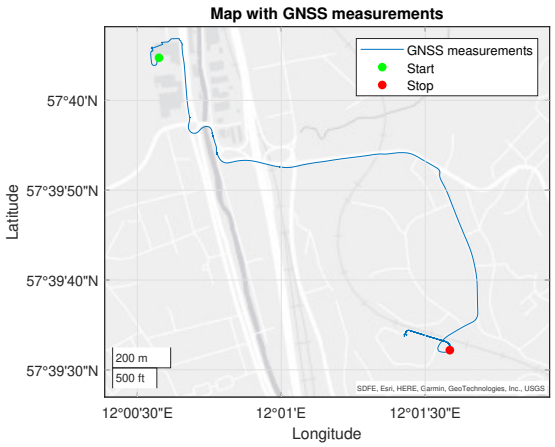


Figure 4.12: Position estimates for the models, showing their estimate means and 3-sigma ellipses at an interval of 1000 time-steps, visualizing the uncertainty of their estimates. Having a GNSS outage through the curves results in larger ellipses since there is no GNSS update reducing the covariance.

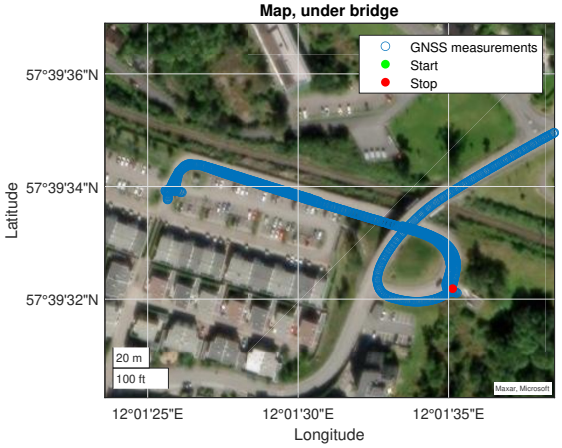
The ellipses in Figure 4.12 show three standard deviations (3-sigma) of the position estimates. This corresponded to a confidence region covering 99.7% of the area in which the models expected the true position to be. The middle point plotted was the mean, which was the estimated position from the model. The CM uncertainty region became a circle around the mean, while the UM and BM uncertainty regions became ellipses. This was because the confidence region of the CM expanded equally in all directions due the accelerometer noise. The UM and BM, however, were only affected by the accelerometer noise in the direction of travel and the gyroscope noise. This caused their confidence region to become smaller than the one for the CM.

4.6 Test with real data

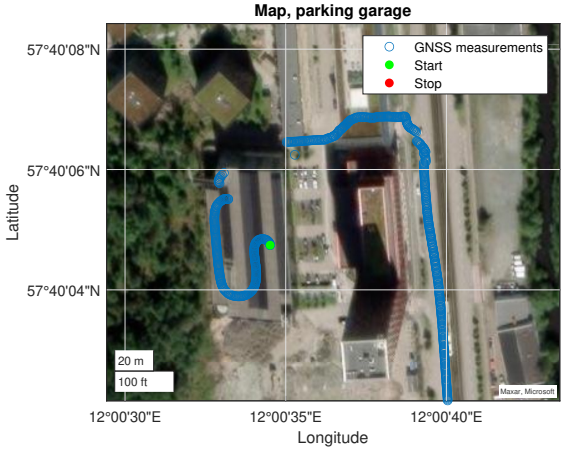
Despite the models being developed in a 2D environment, it was interesting to see how they handled real sensor data compared to artificial. Therefore, a data log containing GNSS, accelerometer, and gyroscope measurements from a car was obtained, where the path from GNSS measurements is illustrated in Figure 4.13a. The test consisted of driving from the roof of a parking garage, through the city, and three times under a bridge. The GNSS measurements were disturbed in two areas: when driving through the parking garage, as illustrated in Figure 4.13c, and when driving under the bridge, as seen in Figure 4.13b.



(a) The entire path used to collect GNSS measurements.



(b) Driving under a bridge three times to collect measurement disturbance.

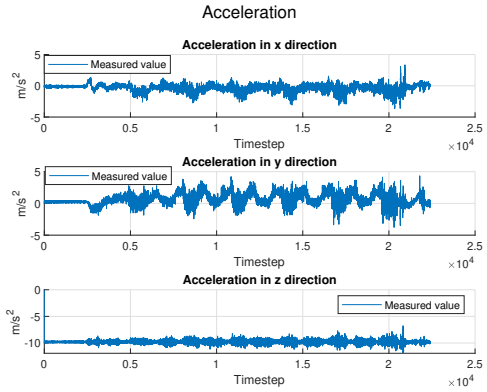


(c) Received GNSS measurements when driving from the roof to the ground through the parking garage.

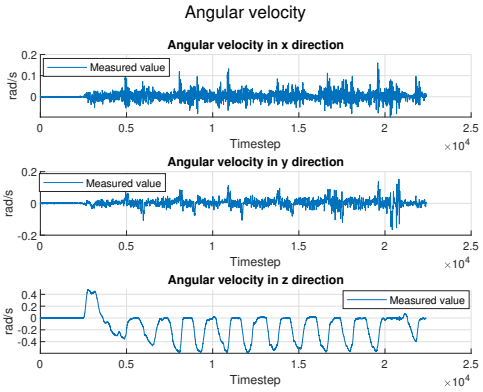
Figure 4.13: Collected data from a test log, using sensors mounted on a car.

An INS log was obtained for the test drive, where the path had been estimated based on a constant acceleration model. However, it differed from the CM model used for comparison in the previous tests because the INS was developed for a real 3D

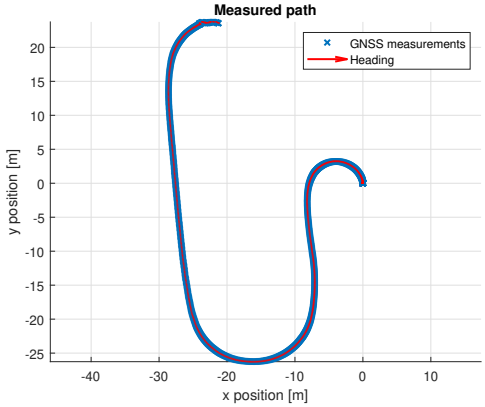
space. Therefore, it took more measurements from the sensors into consideration and also filtered the data. The 2D models received data that had already been filtered through the INS, since that part was not in the scope of this project. The first test conducted only considered the data and path estimation in the parking garage. The IMU data used in this test can be seen in Figures 4.14a and 4.14b, and the GNSS data used can be seen in Figures 4.14c and 4.14d. The meaning of the GNSS quality level seen in Figure 4.14d can be seen in Table 4.9, where the different levels correspond to different position standard deviations.



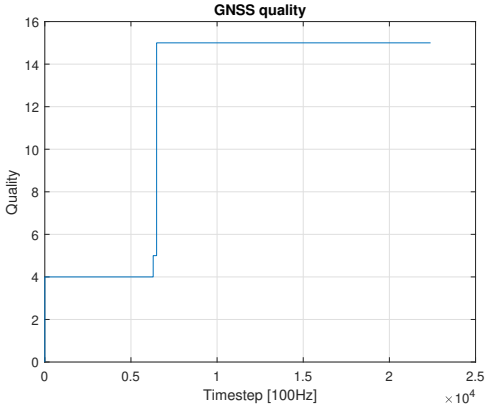
(a) Accelerometer data.



(b) Gyroscope data.



(c) GNSS measurements.



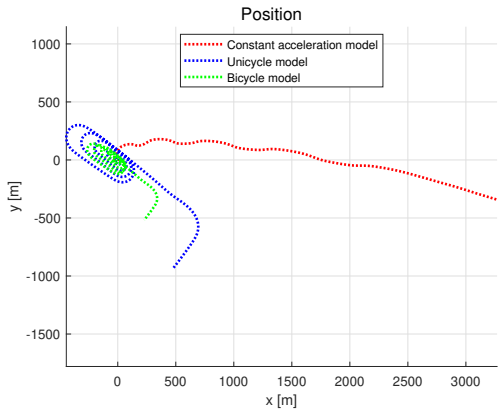
(d) GNSS quality.

Figure 4.14: Collected IMU and GNSS data in the parking garage.

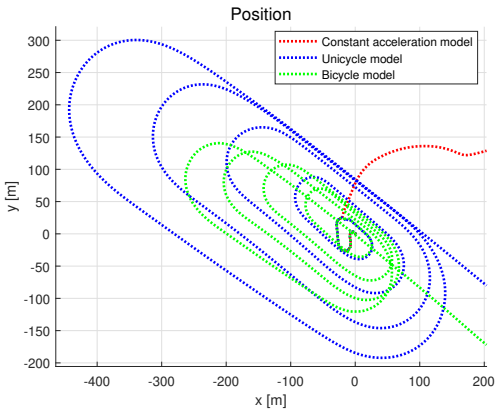
Table 4.9: GNSS quality translation.

GNSS quality	Position standard deviation [m]
1	30
2	15
3	2
4	0.02
5	1
Otherwise	Not usable

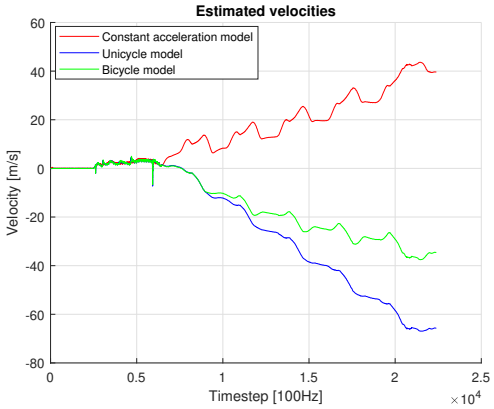
Giving this data to the three models gave the estimations shown in Figure 4.15. Comparing the models, the UM and BM followed the expected path in a garage closer than the CM. However, the scale was unrealistic, and the heading deviated from the expected one based on the GNSS data. It can also be seen that although the BM had a more reasonable scale, it had a slight drift sideways each lap, while the UM had a smaller drift. Further, Figure 4.15c shows that all models gained unreasonable velocities, especially the BM and UM, which gained large negative velocities.



(a) Estimated paths in the parking garage.



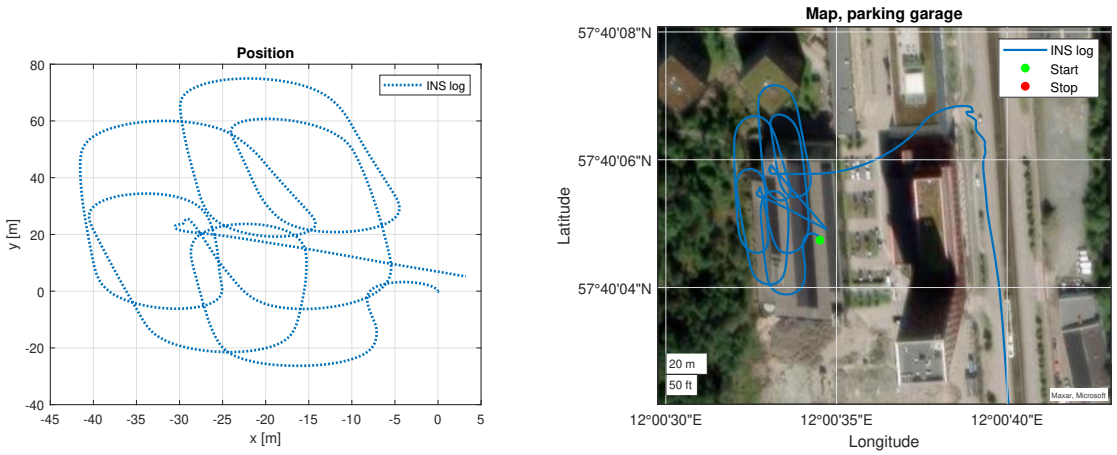
(b) Estimated paths in the parking garage zoomed in at the estimated spiral down.



(c) Estimated velocities in the parking garage.

Figure 4.15: Estimated positions and velocities from parking garage data.

Figure 4.16 shows the estimated path in the parking garage from the INS log, with the same data used for the 2D models. Because this solution was designed for the real world, it compensated for the roll and pitch in the measurements, which caused the scale to remain more realistic. Apart from that aspect, it could be seen that the INS log experienced movement sideways each lap. Therefore, having trouble following the real path in the parking garage, where the path was expected to overlap closely each lap.

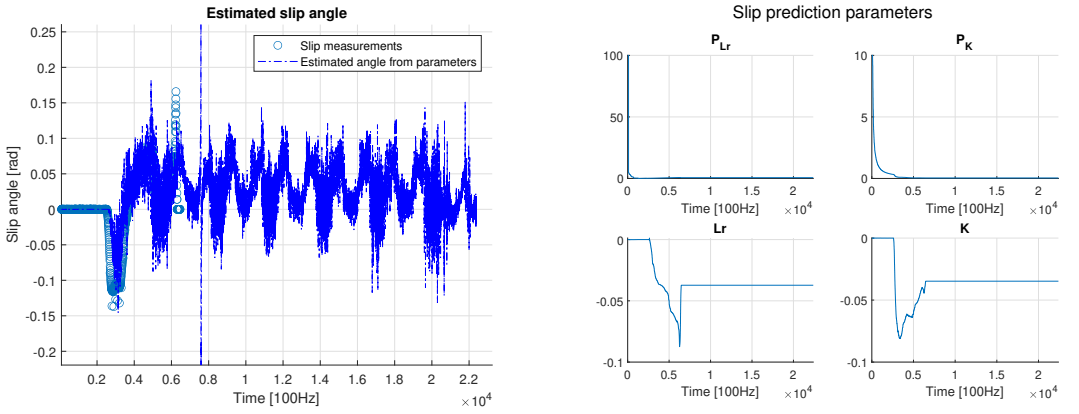


(a) Estimated path in cartesian coordinates.

(b) Estimated path in geographic coordinates.

Figure 4.16: Path estimate from INS log in the parking garage.

Looking at Figure 4.17a, it is possible to see that the slip estimation did not perform optimally and had a spike in the estimation. The figure was scaled to capture the magnitude of the majority of the estimation, and thus, the values of the peak cannot be seen; they were approximately from -1 to 1.8 rad. Since the estimated slip angle depended on the velocity, which had unrealistic values in the estimation, these results did not accurately reflect the performance of the slip estimator.



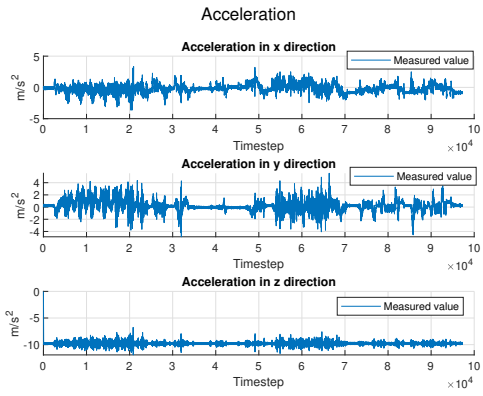
(a) Estimated slip angles in the parking garage.

(b) Slip estimation trained parameters for garage.

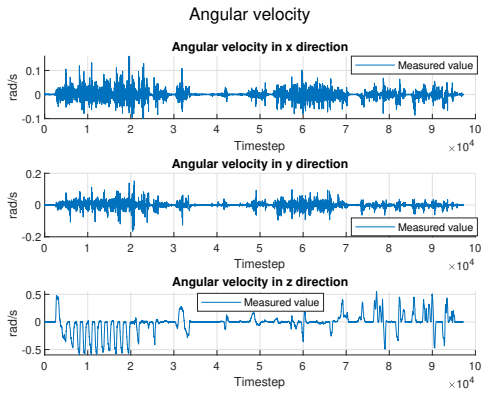
Figure 4.17: Results of the slip estimator in the parking garage.

In Figure 4.17b, the parameters of the slip estimator can be seen. They did not obtain the expected result as both parameters had a low covariance, and especially L_r was far from the expected value. This was because L_r was the distance from the rear axis to the center of gravity and was therefore expected to be positive with a value of scale 10^0, not negative and of scale 10^{-2}.

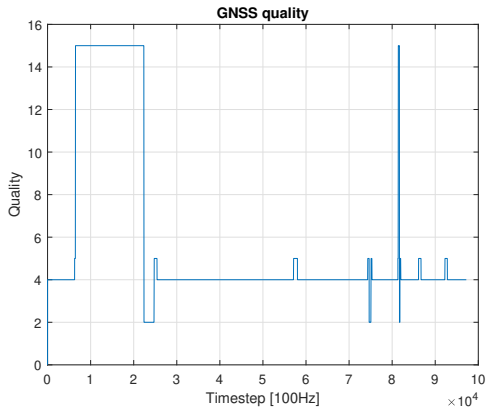
When looking at the entire measured path shown in Figure 4.13a, the accelerometer and gyroscope data used for the estimations can be seen in Figure 4.18, as well as the GNSS quality.



(a) Accelerometer data for full path.



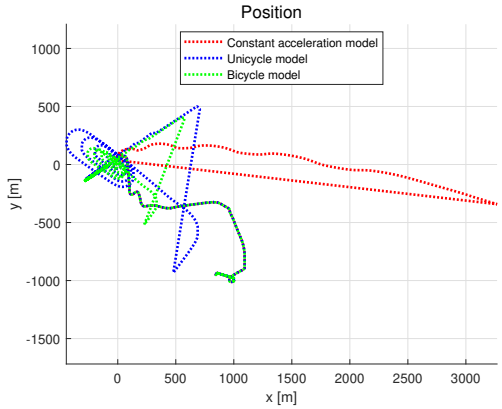
(b) Gyroscope data for full path.



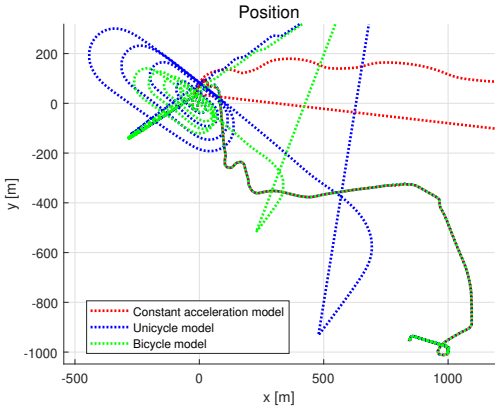
(c) GNSS quality for full path.

Figure 4.18: Collected data for the full path, using CPAC:s sensors on testing equipment.

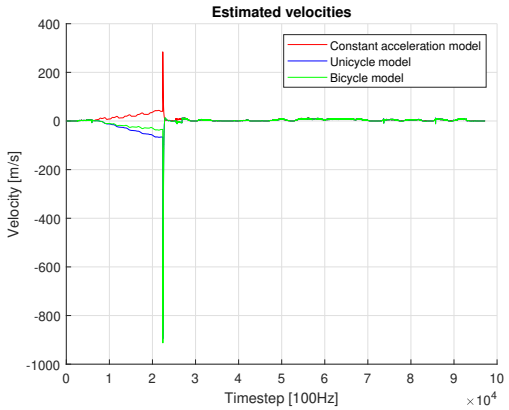
Using this data, the models gave the estimated paths and velocities shown in Figure 4.19 and the estimated slip angles shown in Figure 4.20. When the models regained the GNSS signal after the parking garage, a large jump in position and velocity was observable in Figures 4.19a and 4.19c. After this, the models demonstrated the potential to follow the measured positions, but there were smaller spikes in the velocity where the GNSS signal was once more either fully blocked or had worse precision than before. This was where the car drove under the bridge multiple times, shown in Figure 4.13b. Compared to the parking garage, there was not a considerable deviation from the estimated position when the GNSS signal returned since the time frame for the GNSS outage was significantly shorter. However, there were still smaller velocity deviations that can be seen in Figure 4.19d. In this case, all the models behaved comparably.



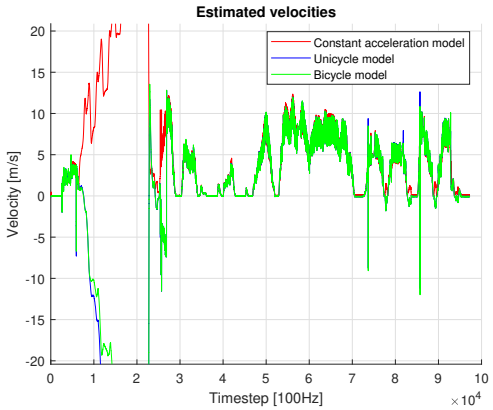
(a) Estimated full path.



(b) Estimated full path zoomed in.



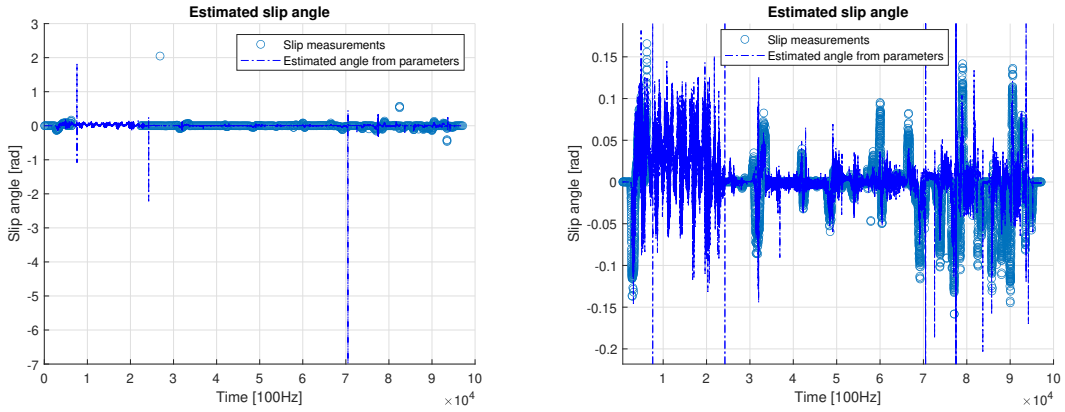
(c) Estimated velocities full path.



(d) Estimated velocities full path zoomed in to more clearly see the values other than the spike.

Figure 4.19: Estimated positions and velocities for the full path.

Looking at Figure 4.20, it was possible to see that there were some peaks where the slip estimator gave considerably unrealistic slip angles. Visualized in Figure 4.20b, the largest peaks were disregarded, and the estimator could be seen to follow the measurements to a certain degree. However, these results should be further investigated as there was no ground truth knowledge. The peaks indicated that the estimator did not perform optimally. However, the lack of 3D compensation might have been one reason for this.



(a) Estimated slip angles for the full path.

(b) Estimated slip angles for the full path zoomed in to remove peaks.

Figure 4.20: Results of the slip estimator for the full path.

5

Discussion

This chapter discusses the results obtained from the simulation environments, focusing on the accuracy of the models across various scenarios and the performance of the slip estimator. It also discusses some ethical aspects of the thesis as well as suggest future works.

5.1 Simulation environment

Two different simulation environments were used, as described in Section 3.1. Both of them had their respective advantages and served different purposes.

The simple environment had the advantage of being completely independent of the vehicle when generating the motion, and therefore, did not contain bias towards a specific type of model. The more advanced simulation was based on a model driven by inputs to generate a path, and therefore, might have been biased towards similar models to the one used to generate the path. In the simple environment, it was also easier to do initial testing as there were no advanced dynamics affecting the motion.

One issue with the simple environment was that implementing realistic slip was not straightforward, since the slip depended on vehicle dynamics. The simulation needed to be extended to a more advanced simulation environment using vehicle and slip dynamics to incorporate realistic slip. Subsequently, the advanced, dynamics-enabled environment was employed to develop and test the models under more complex conditions.

Using the differently complex environments to develop the models made it possible to test and verify them with known ground truth. This approach facilitated spotting errors in the models early and simplified the debugging process, knowing the changes made and comparing the model estimates to the ground truth. It was thereby always possible to verify that modifications produced the intended behavior. Although developing the simulations required a significant time investment, it was likely a more efficient approach than working directly with real-world data. Real-world data contains disturbances and lacks a known ground truth, making it difficult to verify and debug the models during development. In cases where the model produced incorrect predictions, extensive analysis would be needed to determine whether the errors were caused by model inaccuracies or effects from the

disturbances. Moreover, using real-world data would necessitate that the models account for the full complexity of the 3D environment from the beginning. This would increase the initial model complexity, complicate debugging, and significantly expand the range of potential error sources compared to the controlled simulation environment.

5.2 Accuracy of the models for different scenarios

As observed in the results from Section 4.1, the accuracy of the UM and BM depended much on the simulation environment. The UM performed well when there was no slip, whereas the BM underperformed since it estimated minor incorrect slip angles. However, when simulating including slip, the UM performance deteriorated while the BM excelled. This change in performance was because now the BM could estimate and compensate for the slip angles, while the UM could not. It is important to note that none of the simulation environments perfectly replicates real-world conditions. Therefore, the models' real-world accuracy cannot be determined based only on simulation results. Certain assumptions embedded in the simulations may have enhanced the BM's performance, which may not hold in practical applications. If these assumptions fail in real-world scenarios, the BM could experience degraded performance, similar to how the UM's accuracy declined when the no-slip assumption was violated in the simulation.

One advantage of the CM in this case is that it does not rely on any assumptions about the vehicle to determine the position since it simply integrates the accelerometer measurements. This can be seen clearly in Section 4.1.2 where the RMSE for CM was close to the same for both the cases with and without slip. However, both the UM and BM had a lower RMSE in the case without slip, and the BM had a lower RMSE also in the case with slip. These results indicate that a more specific model can give higher positioning accuracy, if it describes reality well, compared to a general model.

When the models were tested with real data, the results shown in Figure 4.15 indicate that the UM and BM might achieve higher accuracy than the CM in real-world applications as well. The comparison is, however, not entirely fair as the accelerometer data was not compensated for the attitude of the vehicle, and the CM is more affected by the issues that arise from that, as shown in Section 4.4. However, when looking at the INS log in Figure 4.16, the results further indicate that the BM and UM might achieve a higher accuracy if they were compensated for 3D, as they seem to follow the expected shape of the garage better. No definitive conclusions can, however, be drawn from this as the scales of the estimated paths differ significantly, making them difficult to compare accurately. It is also unknown how the models would behave when receiving the correctly compensated measurements.

5.3 Slip estimation

As shown in Section 4.3, the slip estimator could estimate the slip with a relatively small error when there were small slip angles, but it has some issues with more considerable angles. The reason for this was probably that the equation describing the slip in the estimator did not reflect a very complex behavior. For the small slip angles, the slip seemed straightforward and feasible to describe with this equation. For larger angles, however, it seemed like the slip gained a more complex behavior that the current equation for the estimation appeared unable to capture.

Looking at Figure 4.20b, the slip measurements of the real-world tests had a maximum magnitude of approximately 0.15 radians. However, when compared to the results from Takikawa et al., where the slip angle was measured with a high accuracy GNSS/IMU system for a 5 km drive in Tokyo, the measured slip angle had a maximum magnitude of approximately 0.03 radians [1]. The difference in magnitude is substantial, which indicates that the method to obtain slip measurements in this thesis might need to be further investigated. Based on the results from Section 4.3, the slip estimator would probably have some issues estimating the slip correctly if the magnitude of the slip measurements from the line-fitting method are correct. If the actual magnitude is close to the one measured by Takikawa et al., however, the simulations indicate that this should not be a problem.

Since there was no reference in the test with real-world data, it was difficult to draw any conclusion about the accuracy of the estimations. However, it was possible to see some spikes in the estimations that indicate that the estimator currently does not perform optimally. This might be an effect of the model not being adapted to 3D, and further investigation is needed to evaluate the current estimator's real-world performance fully.

Another aspect worth examining more closely is the training of the slip parameters. In the simulated environment, the parameter L_r did not appear to converge to the correct value, as shown in Figure 4.11. It can be seen further in its covariance plot, where L_r had a high degree of uncertainty and appears significantly underestimated. However, since the slip estimation did seem to be able to follow the actual slip with relatively high accuracy, the L_r parameter does not seem as significant for the estimation as the K parameter. This could also be attributed to limitations in the simulation environment, which may not provide a perfect representation of reality. In a real-world setting, the L_r parameter might have a noticeable impact and can be further investigated. In the tests with real-world data, it was possible to see the results of the trained parameters in the parking garage in Figure 4.17b. Both L_r and K gained a low covariance, and L_r gained an even more unrealistic value than before, as it was negative. One reason for this was probably that there was not much time for testing with log data, and thus, the slip estimator has not been tuned to fit the real-world conditions. The lack of 3D compensation also likely resulted in the parameters being incorrectly trained due to the incorrect measurements, giving them unrealistic values.

5.4 Ethical and sustainability aspects

In sustainability aspects, this improved positioning algorithm can be used in many different areas, as the position of a vehicle is essential information for many tasks.

The increased position accuracy will make it possible to take more accurate routes. One example is driving in urban areas where the GNSS signal can be disturbed. This might cause a delay in the navigation and result in a longer path than the optimal one taken due to missing an exit. An improved robust positioning algorithm increases the possibility of following the optimal path and reduces the emissions.

The improvement of a positioning and navigation algorithm will not have a direct impact on the land environment, which is goal fifteen in the UN sustainability goals to protect, restore, and promote sustainable use of the land environment. However, the product might be used on heavy-duty vehicles that can cause a great impact. Driving these vehicles will emit pollution and can damage the ground. Having a higher accuracy in positioning can lead to better navigation and more conscious driving, meaning less ground damage and impact on the land environment.

The project does not raise any ethical concerns directly. However, it is not always good to be able to accurately determine the position of a vehicle. The more accurate positioning algorithm could potentially be used for military purposes, for example, automated tanks and military vehicles.

5.5 Future work

From the results presented in Chapter 4, there is some future work needed in order to evaluate the real-world performance of the models. Further research can also be done to investigate the possibility of IMU misalignment compensation using the slip estimator.

5.5.1 Real world expansion and testing

It is clear from the tests with real-world data in Section 4.6 that the current models cannot yet be used for accurate real-world positioning. Therefore, future work needs to be done to expand the models to three dimensions to evaluate real-world performance. As discussed in Section 5.2, there might be some simplifications made to the simulation environment that cause the UM and BM to lose accuracy in the real world compared to the CM, even though the simulations suggest that they might be more accurate.

The data log used from the real-world test contained slopes that the model could not compensate for, and thus, it performed inadequately. However, it would be interesting to further test the models in a flat environment and investigate if the 2D assumption might be close enough to give increased performance compared to the INS log. One such environment could be the relatively flat ground in a storage facility, where, with some initial calibration, the IMU would experience close to a

2D world. In that case, the models might be helpful to enhance the precision of autonomous trucks, for example.

There has been no testing done on the efficiency of the code, and if it were to be used in real-time applications, it would require a short runtime. Especially the line-fitting method is probably not very efficient, and therefore, it would be beneficial to investigate the runtime in real applications. Based on those results, a faster method to obtain slip measurements might be required.

5.5.2 IMU misalignment compensation

An additional potential benefit of slip estimation, which was not investigated, is its possible use for misalignment compensation. If the IMU has been mounted at an angle, it might be possible to estimate this angle with the slip estimator and compensate the predictions and measurements. The slip measurements are able to detect a potential misalignment angle. However, the current slip estimator equations cannot estimate misalignment as they require either lateral acceleration or angular velocity to give an estimation. Further investigation is needed to determine if this is possible. A simple version of the misalignment compensation might be easy to develop, potentially requiring only the addition of a constant term to the current slip estimator equations.

6

Conclusion

This report examined the impact of motion models on the INS algorithm and whether specific vehicle motion models can increase the accuracy. It has also investigated whether slip estimation can affect the performance of the estimations, particularly in GNSS-denied environments.

Three different models have been developed and evaluated: the unicycle model (UM), the bicycle model (BM), and the constant acceleration model (CM). Results demonstrated that the choice of model has a significant impact on positioning accuracy. Both the UM and BM, which incorporate vehicle-specific information, generally outperformed the CM across most of the simulation scenarios. However, when slip dynamics were introduced, the UM performed poorly, while the BM maintained higher accuracy than the CM. Since the only difference between the UM and BM is that the latter includes slip estimation, this indicates that slip estimation can have a significant impact on positioning accuracy.

Slip estimation was implemented by using a Kalman filter to estimate parameters used to find the relation between sensor measurements and the vehicle's slip angle. The estimator was evaluated in different scenarios, and the results indicated that it could reliably estimate slip angles. However, the estimator struggled to capture the dynamics of more pronounced slip angles in the simulation environment.

Although the models were developed in 2D, they were evaluated using real-world 3D data. While this mismatch led to significant errors due to the models' limited adaptability, both the UM and BM demonstrated promising results. These findings suggest that extending the models to a 3D implementation could potentially achieve higher positioning accuracy compared to the estimates from the INS log data.

To answer the research questions, the results show that the INS solution could be enhanced with a vehicle model to increase position accuracy. In the simulations, the BM performed most effectively. The second research question, whether the addition of slip estimation improves the INS solution performance for accurate positioning, could be answered by comparing the UM and BM in the simulations with slip dynamics. Having slip estimations improved the position accuracy, in turn answering the research question.

In conclusion, integrating vehicle-specific motion models and slip estimation into the INS framework has demonstrated clear potential for enhancing positioning accuracy in 2D simulations. To fully assess their applicability in real-world scenarios, future work should focus on extending these models to operate effectively in 3D environments.

Bibliography

[1] K. Takikawa, Y. Atsumi, A. Takanose, and J. Meguro, "Vehicular trajectory estimation utilizing slip angle based on gnss doppler/imu," *ROBOMECH Journal*, vol. 8, 2021, article 5. DOI: 10.1186/s40648-021-00195-4. [Online]. Available: https://doi.org/10.1186/s40648-021-00195-4.

[2] W. Huang, Z. Zhao, X. Zhu, and S. Lei, "An enhanced gnss/mins integrated navigation system with a wheel velocity predictor based on vehicle dynamics model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 11 946–11 960, 2024. DOI: 10.1109/TITS.2024.3363488.

[3] J. Li, B. Feng, L. Zhang, and J. Luo, "Research on vehicle stability control based on a union disturbance observer and improved adaptive unscented kalman filter," *Electronics*, vol. 13, no. 16, 2024, ISSN: 2079-9292. DOI: 10.3390/electronics13163220. [Online]. Available: https://www.mdpi.com/2079-9292/13/16/3220.

[4] O. Eriksson and A. Nordgren Odhner, "Robust Self-Localization in the urban driving environment," M.S. thesis, Chalmers University of Technology, Systems, control and mechatronics (MPSYS), MSc, Gothenburg, 2023. [Online]. Available: http://hdl.handle.net/20.500.12380/306199.

[5] V. Patil, "Generic and complete vehicle dynamic models for open-source platforms," M.S. thesis, Chalmers University of Technology, Applied mechanics (MPAME), MSc, 2017. [Online]. Available: https://hdl.handle.net/20.500.12380/251558.

[6] X. Zhang and J. Yang, "An adaptive robust ekf based on mahalanobis distance and non-holonomic constraints for enhancing vehicle positioning accuracy," *IEEE Sensors Journal*, vol. 24, no. 9, pp. 14 586–14 595, 2024. DOI: 10.1109/JSEN.2024.3373828.

[7] A. Fransson and O. Lundström, "Learning a better attitude: A recurrent neural filter for orientation estimation," M.S. thesis, Chalmers University of Technology, Systems, control and mechatronics (MPSYS), MSc, Gothenburg, 2020. [Online]. Available: https://hdl.handle.net/20.500.12380/300918.

[8] N. Al Bitar and A. Gavrilov, "A new method for compensating the errors of integrated navigation systems using artificial neural networks," *Measurement*, vol. 168, p. 108 391, 2021, ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2020.108391. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026322412030926X.

[9] H. Zhang, Y. Wang, S. Shan, *et al.*, "An enhanced gnss/ins navigation compensation method using lstm-fpn for bridging gnss outages," *Measurement Science and Technology*, vol. 36, no. 1, p. 016 339, Dec. 2024. DOI: 10.1088/1361-6501/ad9cab. [Online]. Available: https://dx.doi.org/10.1088/1361-6501/ad9cab.

[10] S. Chen, M. Xin, F. Yang, *et al.*, "Error compensation method of gnss/ins integrated navigation system based on at-lstm during gnss outages," *IEEE Sensors Journal*, vol. 24, no. 12, pp. 20 188–20 199, 2024. DOI: 10.1109/JSEN.2024.3395009.

[11] J. Wang, L. Yi, and B. Wang, "Research on integrated navigation system based on strapdown inertial component/odometer," in *2024 International Conference on Energy and Electrical Engineering (EEE)*, 2024, pp. 1–7. DOI: 10.1109/EEE59956.2024.10709747.

[12] X. Lyu, X. Wang, G. Xu, and B. Hu, "An inertial-based integrated navigation robust filtering algorithm based on vehicle model for gnss failure," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024. DOI: 10.1109/TIM.2024.3484536.

[13] S. Fang, H. Li, M. Yang, and Z. Wang, "Inertial navigation system based vehicle temporal relative localization with split covariance intersection filter," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5270–5277, 2022. DOI: 10.1109/LRA.2022.3155827.

[14] S. Särkkä and L. Svensson, "Batch and recursive bayesian estimation," in *Bayesian Filtering and Smoothing* (Institute of Mathematical Statistics Textbooks), Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2023, pp. 27–43.

[15] S. Särkkä and L. Svensson, "Extended kalman filtering," in *Bayesian Filtering and Smoothing* (Institute of Mathematical Statistics Textbooks), Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2023, pp. 108–130.

[16] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th International Conference on Information Fusion*, 2008, pp. 1–6.

[17] D. Salmon, "An experimental exploration of low-cost sensor and vehicle model solutions for precision ground vehicle navigation," eng, Auburn University, 2015. [Online]. Available: http://hdl.handle.net/10415/4912.

[18] D. Piyabongkarn, R. Rajamani, J. A. Grogg, and J. Y. Lew, "Development and experimental evaluation of a slip angle estimator for vehicle stability control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 78–88, 2009. DOI: 10.1109/TCST.2008.922503.

[19] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Norwood, MA, UNITED STATES: Artech House, 2017, ch. 1, ISBN: 9781630814427. [Online]. Available: http://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=5430709.

[20] L. Wanninger, "Introduction to network rtk," *IAG Working Group*, vol. 4, no. 1, pp. 2003–2007, 2004.

[21] C. Rizos and S. Han, "Reference station network based rtk systems-concepts and progress," *Wuhan University Journal of Natural Sciences*, vol. 8, no. 2, pp. 566–574, 2003, ISSN: 1993-4998. DOI: 10.1007/BF02899820. [Online]. Available: https://doi.org/10.1007/BF02899820.

[22] K. J. Walchko and P. A. Mason, "Inertial navigation," in *Florida conference on recent advances in robotics*, 2002, pp. 1–9.

[23] A. Harindranath and M. Arora, "Mems imu sensor orientation algorithms-comparison in a simulation environment," in *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, 2018, pp. 1–6. DOI: 10.1109/ICNEWS.2018.8904029.

[24] Q. Zhang, X. Niu, and C. Shi, "Impact assessment of various imu error sources on the relative accuracy of the gnss/ins systems," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 5026–5038, 2020. DOI: 10.1109/JSEN.2020.2966379.

[25] Z. Gao, M. Ge, W. Shen, *et al.*, "Evaluation on the impact of imu grades on bds+gps ppp/ins tightly coupled integration," *Advances in Space Research*, vol. 60, no. 6, pp. 1283–1299, 2017, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2017.06.022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117717304489.

[26] V. Suvorkin, M. Garcia-Fernandez, G. González-Casado, M. Li, and A. Rovira-Garcia, "Assessment of noise of mems imu sensors of different grades for gnss/imu navigation," *Sensors*, vol. 24, no. 6, 2024, ISSN: 1424-8220. DOI: 10.3390/s24061953. [Online]. Available: https://www.mdpi.com/1424-8220/24/6/1953.

[27] CPAC SYSTEMS AB, *IMU 2.0 inertial measurement unit*, https://cpacsystems.se/wp-content/uploads/2024/09/IMU-Datasheet.pdf, Accessed: 2025-5-19.

[28] Z. Ji, Y. Zhao, H. Zhao, and X. Li, "A gnss rtk/ins tight coupling method under challenging scenarios," in *2024 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2024, pp. 1–6. DOI: 10.1109/ICSPCC62635.2024.10770540.

[29] M. Wu, J. Ding, L. Zhao, Y. Kang, and Z. Luo, "An adaptive deep-coupled gnss/ins navigation system with hybrid pre-filter processing," *Measurement Science and Technology*, vol. 29, no. 2, p. 025 103, Jan. 2018. DOI: 10.1088/1361-6501/aa9672. [Online]. Available: https://dx.doi.org/10.1088/1361-6501/aa9672.

[30] W. Gao, R. Yang, J. Huang, and X. Zhan, "Quasi-deep integration for dpe/ins in gnss navigation domain: Framework design and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–21, 2025. DOI: 10.1109/TAES.2025.3532891.

[31] A. de Souza Mendes, D. De Rizzo Meneghetti, M. Ackermann, and A. de Toledo Fleury, "Vehicle dynamics - lateral: Open source simulation package for matlab," in *25th SAE BRASIL International Congress and Display*, SAE International, Oct. 2016. DOI: https://doi.org/10.4271/2016-36-0115. [Online]. Available: https://doi.org/10.4271/2016-36-0115.

[32] S. Zhao, Y. Zhou, and T. Huang, "A novel method for ai-assisted ins/gnss navigation system based on cnn-gru and ckf during gnss outage," *Remote Sensing*, vol. 14, p. 4494, Sep. 2022. DOI: 10.3390/rs14184494.

[33] The MathWorks, Inc., *Solve stiff differential equations and daes — variable order method*, https://se.mathworks.com/help/matlab/ref/ode15s.html, Accessed: 2025-5-15, 2020.

[34] The MathWorks, Inc., *Piecewise cubic hermite interpolating polynomial (PCHIP)*, https://se.mathworks.com/help/matlab/ref/pchip.html, Accessed: 2025-5-15, 2020.

[35] The MathWorks, Inc., *Cubic spline data interpolation*, https://se.mathworks.com/help/matlab/ref/spline.html, Accessed: 2025-5-15, 2020.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY