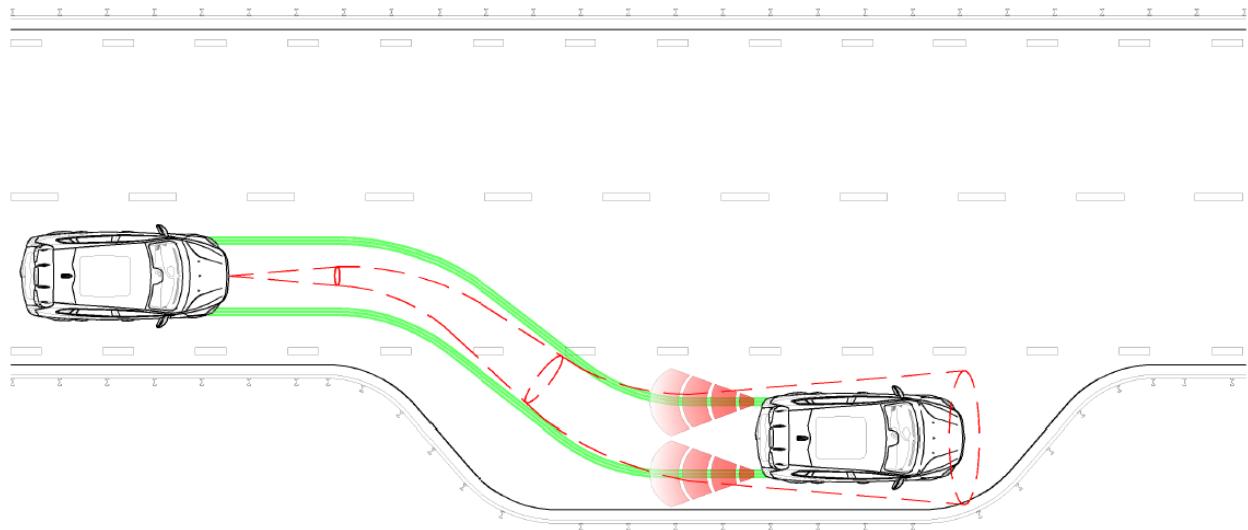




CHALMERS
UNIVERSITY OF TECHNOLOGY



Dead Reckoning during Safe Stop of Autonomous Vehicles

Master's Thesis in Systems, Control and Mechatronics

**Charlotte Lanfelt
Åsa Rogenfelt**

MASTER'S THESIS EX035/2017

Dead Reckoning during Safe Stop of Autonomous Vehicles

Charlotte Lanfelt
Åsa Rogenfelt



Department of Signals and Systems
Division of Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Dead Reckoning during Safe Stop of Autonomous Vehicles
Charlotte Lanfelt
Åsa Rogenfelt

© Charlotte Lanfelt, Åsa Rogenfelt 2017.

Supervisor: Mats Jonasson, Volvo Cars, Department of Active Safety
Examiner: Jonas Fredriksson, Chalmers University of Technology, Department of Signals and Systems

Master's Thesis EX035/2017
Department of Signals and Systems
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An example of how a safe stop manoeuvre could be performed including a reference trajectory and estimated uncertainties.

Typeset in L^AT_EX
Gothenburg, Sweden 2017

Dead Reckoning during Safe Stop of Autonomous Vehicles

Charlotte Lanfelt

Åsa Rogenfelt

Department of Signals and Systems

Chalmers University of Technology

Abstract

This Master thesis considers the problem of position estimation for autonomous vehicles during a so called safe stop, where it cannot be assumed that the vehicles GPS, camera or other environmental sensor data are accessible. The position estimation will therefore instead be done with the use of inertia measurement units, odometers, a pinion angle sensor and vehicle dynamical models. The sensors are initially characterized and modeled to be used in the filtering process and in the simulation environment created with CarMaker. These sensor models are then fused together with vehicle dynamical models in a filtering process using an Extended Kalman filter. The designed filter concepts are used to estimate the position during different scenarios in the simulation environment and using gathered data collected with an actual vehicle. The results from the filtering process using both simulated and gathered data shows similar trends, that the size of the estimation errors mostly depend on the time of the filtering process and the size of the sensor biases. When traveling with the highest allowed initial velocity on a straight road in the simulation environment the errors are approximately 1.3 [m] in longitudinal direction and approximately 1 [m] in lateral direction.

Keywords: *Dead Reckoning, Safe Stop, Autonomous Vehicle, Allan Variance, Extended Kalman Filter, Inertia Measurement Unit, Bicycle Model of Lateral Vehicle Dynamics, CarMaker*

Acknowledgements

We would like to thank the Vehicle Motion & Control group at Volvo Cars for giving us the opportunity to carry out this thesis and giving us all support needed. A special thanks to our supervisor at Volvo Cars Mats Jonasson for all help during the thesis process. We would also like to thank Martin Idegren for all technical support and Niklas Ohlsson for the graphical help. Last we would like to thank our supervisor and examiner Jonas Fredriksson at Chalmers University for the guidance during the thesis work.

Charlotte Lanfelt, Åsa Rogenfelt, Gothenburg, June 2017

Contents

List of Figures	xiii
List of Tables	xvii
Nomenclature	xix
1 Introduction	1
1.1 Background	1
1.2 Purpose	2
1.3 Objective	2
1.4 Scope	2
1.4.1 User Cases	3
1.5 Method	4
2 Sensors	5
2.1 Inertial Measurement Unit	5
2.1.1 MEMS Inertial Sensors	6
2.1.2 Multiple IMU	7
2.1.3 Error Characterization	7
2.1.3.1 Impact on Angular Estimates	7
2.1.3.2 Impact on Position Estimates	8
2.1.4 Allan Variance	9
2.1.5 Characterization of the IMU	11
2.1.6 Modeling of the IMU	14
2.1.6.1 Model Validation	16
2.2 Odometer	18
2.2.1 Characterization of the Odometers	19
2.2.2 Modeling of the Odometers	20
2.3 Pinion Angle Sensor	21
2.3.1 Characterization of the Pinion Angle Sensor	21
2.3.2 Modeling of the Pinion Angle Sensor	23
3 Vehicle Kinematics and Dynamics	25
3.1 Kinematic Model of a Vehicle	25
3.1.1 Rotational Movement	26
3.1.2 Translational Movement	27
3.1.3 Gravitational Effect on the Accelerometer	28

3.1.4	Angular Velocity to Euler Angular Rate	29
3.1.5	Summarizing of the Kinematic Processing	30
3.1.6	Local to Global Coordinates	31
3.1.7	Chassi Angle	31
3.2	Dynamical Model of a Vehicle	33
3.2.1	Lateral Bicycle Models	33
3.2.1.1	Steering-Angle Based Bicycle Model	34
3.2.1.2	Acceleration Based Bicycle Model	36
3.2.2	Simulation of Dynamical Models	37
4	Filtering Design	39
4.1	Bayesian Filtering	39
4.1.1	Process and Measurement Model	41
4.1.2	Kalman Filter	41
4.1.3	Extended Kalman Filter	42
4.2	The Filtering Process	43
4.3	Filtering Concept 1	44
4.3.1	Process Model	45
4.3.2	Measurement Model	45
4.4	Filtering Concept 2	46
4.4.1	Process Model	46
4.4.2	Measurement Model	48
4.5	Treatment of Sensor Data	48
4.6	Filter Evaluation Process	49
4.6.1	Defined Estimation Errors	49
4.6.2	Mean Squared Error	51
5	Simulation Results	53
5.1	Case 1	55
5.2	Case 2	66
5.3	Case 3	72
5.4	Summary of the Simulation Results	80
6	Experimental Tests and Results	83
6.1	Data Gathering	83
6.1.1	Post Treatment of Sensor Data	83
6.2	Results	86
6.2.1	Straight Forward	86
6.2.2	Straight Forward in Slope	91
6.2.3	Right Turn	93
6.2.4	Lane Change	97
6.3	Summary of Experimental Tests	99
7	Discussion and Future Work	101
7.1	Simulation Results	101
7.2	Experimental Tests	103
7.3	Concept Evaluation	104

7.4	Improvements and Future Work	105
8	Conclusion	107
	Bibliography	109
A	Appendix	I
B	Appendix	III
	B.1 IPG CarMaker	III
C	Appendix	V
D	Appendix	VII
E	Appendix	IX
	E.1 Concept 1	IX
	E.2 Concept 2	IX
F	Appendix	XI
	F.1 Simulation Parameters	XI
	F.2 Gathered Data Parameters	XI
G	Appendix	XIII
	G.1 Simulation	XIII
	G.1.1 Concept 1	XIII
	G.1.2 Concept 2	XIII
	G.2 Real data	XIV
	G.2.1 Concept 1	XIV
	G.2.2 Concept 2	XIV
H	Appendix	XV
	H.1 Case 2	XVI
	H.2 Case 3	XVII
I	Appendix	XIX
	I.1 Straight Forward Downhill 20 % 50 [km/h]	XIX
	I.2 Lane Change 90 [km/h]	XXI

Contents

List of Figures

2.1	A conceptual illustration of an IMU with 6-degrees of freedom	5
2.2	Inertial navigation algorithm	6
2.3	Allan deviation for a measurement with an accelerometer during 40 minutes	10
2.4	The time domain data for the IMU with visible discretization levels .	12
2.5	Histograms for the time domain data of the IMU	13
2.6	Allan deviation plot for the accelerometers	14
2.7	Allan deviation plot for the gyroscopes	14
2.8	Conceptual model of the sensor modeling process	15
2.9	The time domain data for the simulated IMU using CarMaker	16
2.10	Histograms for the simulated IMU	17
2.11	Allan deviation plot for the simulated accelerometer X-axis compared to the accelerometer in the XC90	18
2.12	Illustration of the odometer technique	18
2.13	The velocity of the front wheels of the XC90 when traveling at constant velocity forward	19
2.14	Histogram for the odometer data from the front left wheel minus the front right wheel	20
2.15	The rotational velocity and a histogram of the simulated odometers when traveling at constant velocity forward	21
2.16	The actual ratio between the pinion angle and the front wheel angle compared to a linear approximation of 16.75	22
2.17	The angles created when traveling forward with constant velocity keeping the trajectory as straight as possible	22
2.18	Processing of the steering wheel angle obtained in CarMaker to achieve the front wheel angle	23
3.1	The vehicle body fixed coordinate system	25
3.2	Euler angles describing rotations in space	26
3.3	Two points placed on the same rotating rigid body	27
3.4	Illustration of how the gravitational acceleration affects the accelerometer readings	29
3.5	Summarizing of the signal processing performed on the data given from the IMU	30
3.6	A vehicle roll angle due to the lateral forces acting on it	32
3.7	Chassi angle due to accelerations	32

3.8	A visualization of the conceptual bicycle model	33
3.9	Illustration of how the bank angle affects the sign on F_{bank}	34
3.10	The angles created between the front tire and the longitudinal axis X	35
3.11	The relation between the lateral force on the rear tires and the rear tire slip angle	36
3.12	Simulated result for the different bicycle models during case 1	38
3.13	Simulated result for the different bicycle models during case 3	38
4.1	A Bayesian network describing a state space model	39
4.2	The wheels numbering seen from above	48
4.3	The defined estimation errors	50
4.4	Trigonometry to calculate the errors	50
5.1	The estimated X- and Y- positions for case 1	55
5.2	Estimated global position states with standard deviations for case 1 . .	56
5.3	Estimated velocity and acceleration states for case 1	57
5.4	Estimated Euler angles and respective rates case 1	58
5.5	The errors of e_{lon} , e_{lat} and e_ψ for case 1	59
5.6	The distribution of the errors over 1000 runs for case 1	61
5.7	Squared error of e_{lon} , e_{lat} and e_ψ using different numbers of IMU:s for case 1	62
5.8	Squared error illustrated against growing constant bias of the IMU:s for case 1	63
5.9	Squared error illustrated against growing initial velocity for case 1 . .	65
5.10	The estimated X- and Y- positions for case 2	66
5.11	Estimated global position states with standard deviations for case 2 .	67
5.12	Estimated velocity and acceleration states for case 2	68
5.13	Estimated Euler angles and respective rates for case 2	69
5.14	The errors of e_{lon} , e_{lat} and e_ψ for case 2	70
5.15	Squared error of e_{lon} , e_{lat} and e_ψ using different numbers of IMU:s for case 2	71
5.16	The estimated X- and Y- positions for case 3	72
5.17	Estimated global position states with standard deviations for case 3 .	73
5.18	Estimated velocity and acceleration states for case 3	74
5.19	Estimated Euler angles and respective rates for case 3	75
5.20	The errors of e_{lon} , e_{lat} and e_ψ for case 3	76
5.21	Squared error of e_{lon} , e_{lat} and e_ψ using different numbers of IMU:s for case 3	77
5.22	Squared error illustrated against growing constant bias of the IMU:s for case 3	78
5.23	Squared error illustrated against growing initial velocity for case 3 .	80
6.1	The normal forces acting on the vehicle	84
6.2	The estimated X- and Y- positions when traveling straight forward .	87
6.3	Estimated global position states with standard deviations when traveling straight forward	88

6.4	Estimated velocity and acceleration states when traveling straight forward	89
6.5	Estimated Euler angles and their rates when traveling straight forward	90
6.6	The estimated X- and Y- positions traveling straight forward downhill	91
6.7	Estimated global position states with standard deviations traveling straight forward downhills	92
6.8	The estimated X- and Y- positions for a right turn	93
6.9	Estimated global position states with standard deviations for a right turn	94
6.10	Estimated velocity and acceleration states for a right turn	95
6.11	Estimated Euler angles and their rates for a right turn	96
6.12	The estimated X- and Y- positions for a lane change	97
6.13	Estimated global position states with standard deviations during a lane change	98
A.1	Definition of the estimation errors in lateral-, longitudinal position and heading	I
D.1	Allan deviations of the simulated accelerometer and gyroscope for 30 runs compared with the Allan deviation for the vehicle IMU	VII
H.1	The distribution of the errors over 1000 runs for case 2	XVI
H.2	The distribution of the errors over 1000 runs for case 3	XVII
I.1	Estimated velocity and acceleration states traveling straight forward downhills	XIX
I.2	Estimated Euler angles and their rates traveling straight forward downhills	XX
I.3	Estimated velocity and acceleration states during a lane change	XXI
I.4	Estimated Euler angles and their rates during a lane change	XXII

List of Figures

List of Tables

2.1	Noise peak to peak specifications on the sensors given by the data sheet, see Appendix C	11
2.2	Calculated peak to peak values for the IMU	12
2.3	Calculated mean and variance for the XC90 IMU	12
2.4	<i>RW</i> and <i>BI</i> read from the Allan deviation plot in Figure 2.6, 2.7 and Appendix D	15
2.5	The calculated mean and variance for the simulated IMU	17
5.1	Mean and standard deviations when traveling straight forward	60
5.2	Filtering result from simulated data with absolute values of the errors for case 1	64
5.3	Mean and standard deviation for case 2	70
5.4	Mean and standard deviations for case 3	76
5.5	Filtering result from simulated data with absolute value of the errors for case 3	79
6.1	Filtering result from gathered data when traveling straight forward . .	86
6.2	Filtering result from gathered data when traveling up- and downhill . .	91
6.3	Filtering result from gathered data when performing a turn	93
6.4	Filtering result from gathered data during lane change	97
F.1	Vehicle parameters used in filtering process for the simulated data . .	XI
F.2	Vehicle parameters used in filtering process for the gathered data . .	XI

List of Tables

Nomenclature

Abbreviations

- C1 Concept 1
C2 Concept 2
CAN Controller Area Network
COG Center of Gravity
EKF Extended Kalman Filter
EMC Electromagnetic Compatibility
FOG Fibre Optic Gyroscope
GPS Global Positioning System
GUI Graphical User Interface
IMU Inertial Measurement Unit
KF Kalman Filter
MEMS Micromachined Electro Mechanical Systems
MIMU Multiple Inertial Measurement Unit
MSE Mean Squared Error
PDF Probability Density Function
PSD Power Spectral Density
SAW Surface Acoustic Wave
UKF Unscented Kalman Filter

Vehicle Kinematics

- $\ddot{\psi}$ Yaw acceleration, [rad/s²]
 $\ddot{\theta}$ Pitch acceleration, [rad/s²]

$\ddot{\varphi}$ Roll acceleration, [rad/s²]

$\dot{\psi}$ Yaw rate, [rad/s]

$\dot{\theta}$ Pitch rate, [rad/s]

$\dot{\varphi}$ Roll rate, [rad/s]

ψ Yaw, [rad]

θ Pitch, [rad]

φ Roll, [rad]

$R_{x,y,z}$ Rotational matrices

X, Y Global coordinate system axis

Other Symbols

e_ψ The angular heading error, [rad]

e_{lat} The lateral position error, [m]

e_{lon} The longitudinal position error, [m]

g Gravitational acceleration constant, [m/s²]

t_s Sampling interval, [s]

Sensors

$\omega_{x,y,z}$ Angular velocities from gyroscope, [rad/s]

$a_{x,y,z}$ Inertial frame accelerations from accelerometer, [m/s²]

Vehicle Dynamics

α_f Slip angle, front wheel, [rad]

α_r Slip angle, rear wheel, [rad]

δ Steering angle of the front wheel, [rad]

\dot{v}_x Vehicle longitudinal acceleration, [m/s²]

\dot{v}_y Vehicle lateral acceleration, [m/s²]

\dot{v}_z Vehicle vertical acceleration, [m/s²]

ϕ Road bank angle, [rad]

\mathbf{V} Vehicle velocity vector, [m/s]

θ_{V_f} Angle between velocity vector and longitudinal axis, front wheel, [rad]

θ_{V_r}	Angle between velocity vector and longitudinal axis, rear wheel, [rad]
C_f	Cornering stiffness, front wheels, [N/rad]
C_r	Cornering stiffness, rear wheels, [N/rad]
F_{bank}	Lateral force due to road bank angle, [N]
F_{xf}	Longitudinal tire force, front wheel, [N]
F_{xr}	Longitudinal tire force, rear wheel, [N]
F_{yf}	Lateral tire force, front wheel, [N]
F_{yr}	Lateral tire force, rear wheel, [N]
I_z	Yaw moment of inertia, [kgm ²]
l_f	Distance from COG to front wheel axis, [m]
l_r	Distance from COG to rear wheel axis, [m]
m	Vehicle mass, [kg]
R^{whl}	Vehicle wheel radius, [m]
v_x	Vehicle longitudinal velocity, [m/s]
v_y	Vehicle lateral velocity, [m/s]
v_z	Vehicle vertical velocity, [m/s]

1

Introduction

In this chapter the background of the Master thesis is presented with some fact about self-driving vehicles and what previously have been done within the research field of dead reckoning. The purpose and scope with the thesis is also presented together with the main objective. Last the method of the work is described together with the outline of the report.

1.1 Background

Vehicles are a big part of today's society and a lot of people depend on them in their everyday life, for example in transport, work and hobbies. Recently the technology in vehicles has developed rapidly and today there even exists some self-driving vehicles that are circulating the streets. One example is *Drive Me*, [1], a project by Volvo Cars that allows 100 regular people to use self-driving vehicles in a specified 50 [km] long path in Gothenburg. These vehicles still require humans that can actively interact with the vehicle and reclaim the control (in case something unexpected happens on the street or if the vehicle suffers from a severe error, [2], [3]).

For many of the companies in the automotive industry the long term goal is to have fully autonomous vehicles without any human interaction, see [4], [5] and [6]. To achieve this, self-driving vehicles needs to be able to handle all possible situations and circumstances, which means that a lot of special cases needs to be solved in the area of self-driving vehicles. One example is that the lateral and longitudinal control of self-driving vehicles are normally done with environmental sensors such as cameras, radar and GPS. However during severe failures, such as power black-out, EMC disturbance, communication shut down, etc., the self-driving vehicle must be able to be controlled to a safe stop without any environmental sensors or human interaction.

It is not possible to directly measure the position of the vehicle without using GPS, camera or other environmental sensors, instead measurements like velocity, acceleration and angular velocity might be accessible during severe failures. This kind of navigation without sensors that measures the actual position is called dead reckoning and is often used at seas or by aircrafts where GPS signals are limited. Vehicular

navigation experiments has been done at places where GPS tracking is not possible, or limited as for example in tunnels or narrow streets surrounded with skyscrapes, see [7] and [8]. This kind of navigation has though only been practised during very limited time sequences since when not using exact measurements the position estimates easily drift away due to biases on the sensors. As shown in [8] the error grows almost exponential when the time period of dead reckoning increases.

So for self-driving vehicles to be possible on a long term goal it is important to solve among others, the problem in case the readings from the environmental sensors gets non trustworthy. This for the vehicles to be as safe as possible and potential to use during all circumstances.

1.2 Purpose

The purpose with this Master thesis is to explore different approaches to estimate the vehicles position in case of a severe failure (when the environmental sensors are blacked-out or not trustworthy). To perform position estimation during such severe failure the vehicle has to be equipped with additional EMC resistant backup-sensors such as IMU:s and odometers. The position estimate is then supposed to be used during a safe stop which means that the vehicle decelerates to stop while following a safe path to for example the roadside.

1.3 Objective

The main objective of this Master thesis is to estimate the position of the vehicle with minimized position error using dead reckoning. A part of the objective is also to investigate how small the tracking errors, see Appendix A, can be during different conditions. Reasonable position estimation errors for the dead reckoning process are 3 [m] longitudinal and 0.75 [m] lateral and the objectives are therefore set to fulfill those with 95% confidence (2σ standard deviations) with an initial velocity of 120 [km/h].

1.4 Scope

Self-driving vehicles are assumed to be equipped with extra sensors that will be used during severe failures. The sensors that are assumed to always be available are a six degrees of freedom MEMS IMU, an odometer for each wheel and a sensor that measures the pinion angle, (which is possible to estimate the front wheel angle from). The vehicle will be equipped with three MEMS IMU:s, but it can not be assumed that all three will be available during a severe failure. However it will be

investigated how the number of IMU:s affect the result. Different kinds of IMU:s and odometers will not be investigated as a part of this thesis.

In this thesis a Volvo XC90, henceforth only referred as XC90, will be used for data gathering therefore equivalent parameters as the vehicle will be used in simulation as well. However the XC90 that is used for data gathering is not equipped with exactly the same sensors as the self-driving cars will have. The only significant difference is that the IMU only has five degrees of freedom instead of six. This will be solved by adding the extra degree of freedom, the pitch rate from a reference system that will be placed within the vehicle when gathering the data.

For the safe stop it is assumed that a safe stop path (buffered before the occurrence of the fault) will be available. The vehicle will also continuously estimate and save all data given from the environmental sensors. This will give access to the vehicles actual position, velocity, acceleration and orientation at the time of the black-out.

It is assumed that the vehicle will not have a higher velocity than 120 [km/h] when traveling straight forward which limits the time for the death reckoning to approximately 10 [s]. This given a braking power of approximately 5 [m/s²] which is a limit due to passenger comfort. When traveling in a curve the lateral acceleration also needs to be limited due to passenger comfort. The limit is set to $|a_y| < 2$ [m/s²] which then limits the velocity in curves to $v_x < \sqrt{a_y R}$, where R is the curve radius.

1.4.1 User Cases

Different safe stop user cases has been defined to test the dead reckoning process. These will be referred to as 1, 2, 3 throughout the report.

- **Case 1**

The vehicle starts at 120 [km/h] (which is the maximum velocity self-driving vehicles will be assumed to drive with) and decelerates with 5 [m/s²] on a straight road until it stands still.

- **Case 2**

The vehicle starts at 120 [km/h] on a straight road with a longitudinal slope of 20 % and decelerates with 5 [m/s²] until it stands still.

- **Case 3**

The vehicle travels on a road with a constant radius of 100 [m]. The velocity is set to the maximum velocity allowed that will not exceed the limitation $v_x < \sqrt{a_y R}$, see Section 1.4, which in this case will be 50 [km/h] for $|a_y| < 2$ [m/s²]. The vehicle decelerates with 5 [m/s²] until it stands still.

1.5 Method

The thesis work starts with characterization of the IMU sensors performance and analysis of the sensor data using for example Allan variance. Next the odometer and pinion angle sensor are characterized and modeled. After characterizing and modeling all the sensors vehicle kinematics and dynamics are derived and validated using sensor data from CarMaker, see Appendix B. Two different vehicle lateral bicycle models are investigated and tested to represent the dynamical behaviour of the vehicle. With the characterized sensors and validated vehicle kinematics and dynamics two different sensor fusion filters are designed for filtering the sensor data to retain position estimates. The filters are tuned and compared with each other using the different user cases to achieve the most correct position estimate.

The thesis outline is structured in the same order as the method is performed i.e. first sensor analysis, then vehicle modeling and last the filtering process.

2

Sensors

In this chapter the sensors used during the dead reckoning process are presented. It is also described how they are characterized and modeled for the development of a realistic simulation environment.

2.1 Inertial Measurement Unit

An IMU consist of an accelerometer, a gyroscope and a magnetometer or a subset of those. Accelerometers measure the acceleration force applied to the IMU in X-, Y- and Z-direction. Gyroscopes measure the angular velocity of the sensor's axis, see Figure 2.1 and magnetometers measure the magnetic fields in the same axis as the accelerometer.

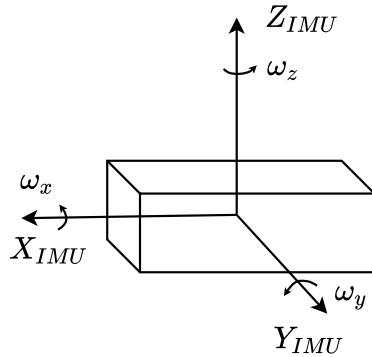


Figure 2.1: A conceptual illustration of an IMU with 6-degrees of freedom

There exists IMU:s with everything from 1- to 9-degrees of freedom, the simplest ones only containing one accelerometer or a gyroscope and the most advanced one all three sensors in all three axis. In the scope for this thesis the focus will lie on 6-degrees of freedom IMU:s since it is assumed that it is required to perform a safe stop.

With a 6-degrees of freedom IMU the angular velocity can be used to estimate the orientation of an object given the initial orientation. Accelerations can together with the estimated angles be used to estimate the position of the object, see Figure 2.2.

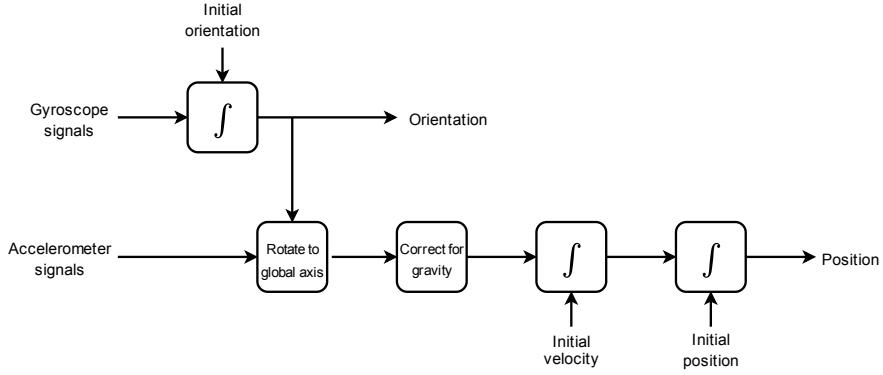


Figure 2.2: Inertial navigation algorithm

There exists different types of gyroscopes, for example; mechanical, FOG, and MEMS, see [9]. Mechanical gyroscopes consists of a spinning wheel mounted on two gimbals that allows it two rotate in all three axis. A mechanical gyroscope measures the objects orientation and not the angular velocity as the more modern gyroscope does. A FOG measuring the angular velocity by firing two light beams into a coil of optical fibre in opposite directions. By using the Sagnac effect, see [10], it is possible to measure the angular velocity by measuring the intensity of the combined beam. Both the mechanical gyroscope and the FOG is expensive to produce. There also exists different types of accelerometers, for example mechanical; solid state and MEMS. The mechanical accelerometer consists of a mass attached by springs and the acceleration is then measured by the displacement of the mass. Solid state accelerometers can be divided into sub-groups one example is the SAW accelerometer. A SAW sensor measure the change in frequency of mass caused by a change in tension which occurs by accelerations, see [9].

2.1.1 MEMS Inertial Sensors

MEMS inertial sensors are a type of accelerometers and gyroscopes built using silicon-micro-machining techniques. This technique makes it possible for the sensors to have a low number of parts and relatively cheap to produce. Other advantages with this type of sensors are for example; small size, low weight, low power consumption, short start up time, high reliability and low maintenance according to [9]. The disadvantages is that the performance can not match the accuracy of traditional accelerometers and gyroscopes such as mechanical accelerometers and optical gyroscopes.

2.1.2 Multiple IMU

By using multiple IMU:s, (MIMU), the confidential of measurements can be increased drastically. As shown in [11] the variance of the sensor data is decreased to var/N , where N is the number of IMU:s, this during the assumption that the measurement errors are independent. According to [12] the navigation errors using 4 independent IMU:s reduces by approximately 40%. Using several IMU:s does not only decrease the standard deviation it also makes it easier to detect failures in the sensors which makes it possible to exclude incorrect measurements.

2.1.3 Error Characterization

To achieve reliable results for position and orientation using IMU data the disturbances of the signals have to be characterized. This has to be done quite accurately since possible errors will have a big impact when integrating the signals. The impacts that affect the readings of MEMS accelerometers and gyroscopes are quite similar. The disturbances with the most influence are; an addition of a constant bias, a perturbation of thermo-mechanical noise which due to the fast fluctuations (faster than the sampling rate) causes the samples to be perturbed by a white noise sequence. Random flicker noise in the components also contribute and causes the bias to wander over time. These error sources do however affect the integrated signals in different ways since the samples from the accelerometer are integrated twice to achieve the position while the samples from the gyroscope are only integrated once to achieve the angles.

2.1.3.1 Impact on Angular Estimates

The angular error due to the constant bias, ϵ , on the integrated signal will grow linearly due to the integration process, during a given time t it is given by

$$\int_0^t \epsilon(\tau) d\tau = \epsilon \cdot t. \quad (2.1)$$

This can result in a quite big error if not compensated for in a correct way. The white noise perturbation due to the thermo-mechanical noise will then add a zero mean random variable at every sample. These random variables are assumed to be Gaussian distributed with a given variance, $N_i \sim \mathcal{N}(0, \sigma_w^2)$. The integral of a sequence, $w(\tau)$, of random variables N_i with sampling interval t_s , during a specified time $t = k \cdot t_s$ is given by

$$\int_0^t w(\tau) d\tau = t_s \sum_{i=1}^k N_i. \quad (2.2)$$

This will lead to an additional error on the angles that is described by a random walk. To determine the properties of the random walk the mean and variance can be calculated by

$$\mathbb{E} \left(t_s \sum_{i=1}^k N_i \right) = k \cdot t_s \mathbb{E}(N) = 0 \quad (2.3)$$

$$\text{Var} \left(t_s \sum_{i=1}^k N_i \right) = \underbrace{k \cdot t_s}_{t} \cdot t_s \text{Var}(N) = t \cdot t_s \cdot \sigma_w^2. \quad (2.4)$$

Which shows that the variance of the random walk grows linearly with time and that the standard deviation grows with the square root of time. A common unit to specify this noise angular random walk is therefore in ${}^\circ/\sqrt{h}$ (ARW). Other commonly used units are PSD, (${}^\circ/h)^2/Hz$) and FFT noise density, (${}^\circ/h/\sqrt{Hz}$). It is possible to convert between these units according to:

$$\begin{aligned} ARW({}^\circ/\sqrt{h}) &= \frac{1}{60} \cdot \sqrt{PSD(({}^\circ/h)^2/Hz)} \\ ARW({}^\circ/\sqrt{h}) &= \frac{1}{60} \cdot FFT({}^\circ/h/\sqrt{Hz}). \end{aligned} \quad (2.5)$$

The wandering of the bias caused by the flicker noise can be modeled as a random walk creating a second order random walk on the integrated angular signal. However this assumption is only valid for short periods of time since otherwise the angular error would grow unbounded proportionally to the time.

When estimating velocity based on accelerometer data the error affect will be very similar to this case, however there will be a velocity random walk instead of angular random walk.

2.1.3.2 Impact on Position Estimates

Just as the angular estimate errors the position estimate errors due to a constant bias error grows with time, and is given by

$$\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau = \frac{\epsilon \cdot t^2}{2}, \quad (2.6)$$

this time the error grows not linearly but quadratically with time.

The white noise sequence does also this time affect in the same way as for the gyroscope but this time due to the double integration the error in position will be a second order random walk, as follows

$$\mathbf{X}_w = \int_0^t \int_0^t w(\tau) d\tau d\tau = t_s \sum_{j=1}^k t_s \sum_{i=1}^j N_i. \quad (2.7)$$

If assuming that the sampling interval is small it is possible to characterize the properties of the second order random walk by calculating its mean and variance. According to [9] they are given by

$$\begin{aligned}\mathbb{E}(\mathbb{X}_w) &= \mathbb{E}\left(\int_0^t \int_0^\tau \epsilon(\tau) d\tau d\tau\right) \\ &= t_s^2 \sum_{i=1}^n (n-i+1) \mathbb{E}(N_i) = 0\end{aligned}\tag{2.8}$$

and

$$\begin{aligned}\text{Var}(\mathbb{X}_w) &= \text{Var}\left(\int_0^t \int_0^\tau w(\tau) d\tau d\tau\right) \\ &= t_s^4 \sum_{i=1}^n (n-i+1)^2 \text{Var}(N_i) = \frac{t_s^4 n(n+1)(2n+1)}{6} \text{Var}(N) \\ &\approx \frac{1}{3} \cdot t^3 \cdot t_s \cdot \sigma_w^2.\end{aligned}\tag{2.9}$$

Analyzing these values shows that this second order random walk in position has a standard deviation that grows in proportion to $t^{3/2}$.

The wandering of the bias is modeled as a random walk for the accelerometer as well as for the gyroscope which make the error estimation also in this case only valid for a short period of time. However due to the double integration the affect on the position estimate is now a third order random walk with a standard deviation that grows proportionally to $t^{5/2}$.

2.1.4 Allan Variance

To characterize and identify possible underlying white noise of a signal and also to identify possible bias instability (which causes the bias to wander over time) a technique called Allan variance can be used. The Allan variance is different from the regular variance, for example instead of being a function of time it is a function of averaging time. This means that instead of giving just a value of the variance at a specific time it describes how the variance of a signal changes over different time intervals. These time intervals can be viewed of as windows (or bins) which then are slided over the whole signal during one iteration of the calculation process. The average is calculated for the samples in each window at each position. In the next iteration this window (or bin) will be enlarged with a predetermined size and slided over the signal again. This iterating process which reuses the samples in each iteration then removes the time dependency and instead the Allan variance is as mentioned a function of averaging time.

More specifically described the Allan variance is calculated dividing the signal into a sequence of bins for each iteration. The size of the bins, m , increases for each iteration with $m = 2^j$ where j is the number of iteration. According to [13] the

number of bins, n , should not be smaller than $n < N/2$, where N is the number of samples. The averaging time, T , is then updated in each iteration as $T = m \cdot t_s$, where t_s is the sampling interval. For each iteration, each bin should be averaged such that list of averages, $(a_1(T), a_2(T), \dots, a_n(T))$, is obtained. According to [9] the Allan variance for each iteration is then given by

$$\text{AVAR}(T) = \frac{1}{2(n-1)} \sum_{i=1}^n (a_{i+1}(T) - a_i(T))^2 \quad (2.10)$$

and the Allan deviation is given by

$$\text{AD}(T) = \sqrt{\text{AVAR}(T)}. \quad (2.11)$$

The resulting Allan deviation can be plotted as a function of the averaging time T on a log-log scale. By analyzing the Allan deviation plot it is possible to identify underlying disturbances on the signal. In Figure 2.3 the Allan deviation for a measurement with an accelerometer during 40 minutes is shown. The underlying white noise of the signal appears on the plot as a gradient with slope -0.5 . The variance of the white noise can be deduced from the Allan deviation plot by identifying the random walk constant, (RW). RW can be identified by placing a tangent in the plot where the slope is -0.5 , it is then deduced where the tangent crosses $T = 1$. When the slope flattens out the bias instability appears and the constant, BI , can be identified at the minimum value of the slope. BI describes how the bias of the signal will wander over time. This two constants can then be used to recreate a signal with the same characteristics as the original signal.

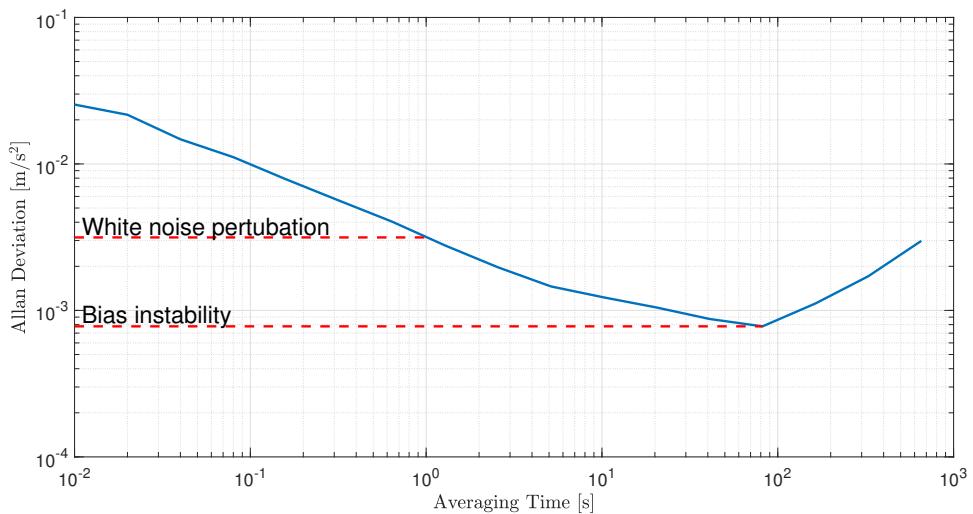


Figure 2.3: Allan deviation for a measurement with an accelerometer during 40 minutes

2.1.5 Characterization of the IMU

The IMU sensors used throughout the project are a 3-axis MEMS accelerometer and a 2-axis MEMS gyroscope placed within the XC90. However in simulation a 3-axis gyroscope will be modeled instead since the IMU in the self-driving vehicles are assumed to have 6-degrees of freedom. The sampling frequency for the XC90 is set to 100 Hz. To be able to find the characteristics of the sensors, sensor data had to be gathered during a longer period of time (40-60 minutes). This was done when the vehicle was not exposed to any forces or disturbances, (except for the gravitational force), and was standing on a flat surface. The gathered data could then be analyzed in MATLAB to be able to find characteristics for each axis of each sensor. With the help of MATLAB the mean and variance for each sensor was calculated and conclusions regarding if the sensor noise was Gaussian distributed or not could also be drawn. Comparisons with the given data sheets for the XC90 IMU, see Appendix C, was also done to find out if the results was reasonable and within the specifications given from the distributors, see Table 2.1.

Table 2.1: Noise peak to peak specifications on the sensors given by the data sheet, see Appendix C

	Signal Noise (peak to peak)
Accelerometer	$<30 \text{ [mg]} \approx 0.2946 \text{ [m/s}^2\text{]}$
Gyroscope	$<1.5^\circ/\text{s} \approx 0.0262 \text{ [rad/s]}$

2. Sensors

The time domain data for the accelerometer and gyroscopes different axis are visible in Figure 2.4. The plots of the signals shows clear discrete levels, this is due to the quantization of the signal when it goes through the CAN bus. The quantization level for the accelerometer is established to $0.0085 \text{ [m/s}^2]$ and for the gyroscope it is $0.000244140625 \text{ [rad/s]}$.

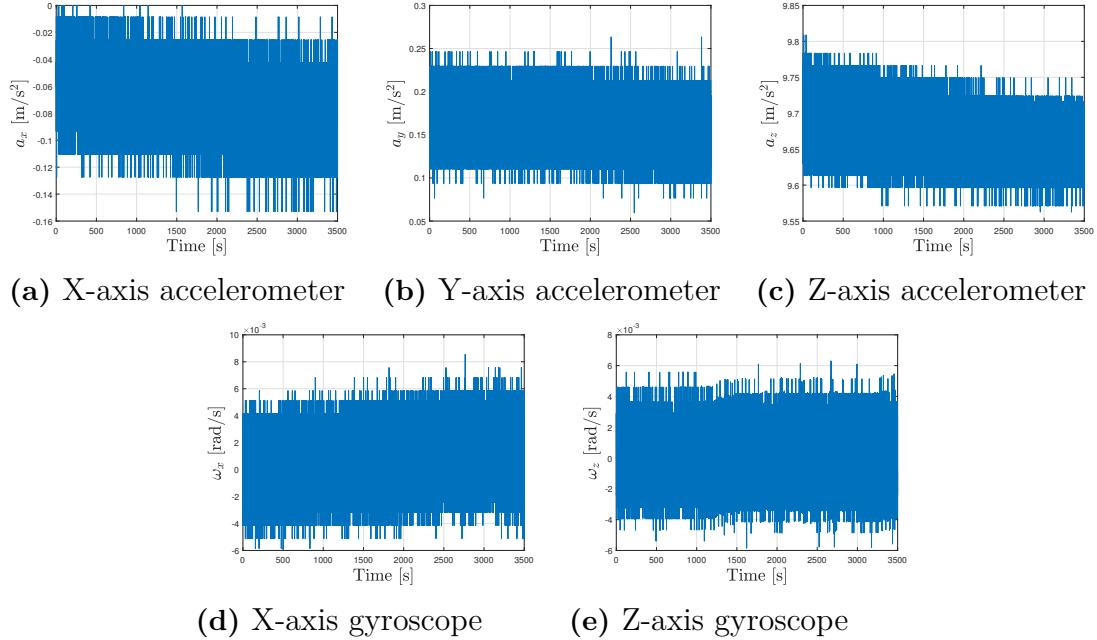


Figure 2.4: The time domain data for the IMU with visible discretization levels

In Section 2.1.3 it was stated that if the signal was perturbed by random flicker noise within the component this would cause the bias to wander over time. This behaviour is visible for some of the measured signals, however all signals are still within the specified peak to peak value given by the distributor, see Table 2.1 and 2.2.

Table 2.2: Calculated peak to peak values for the IMU

	a_x	a_y	a_z	ω_x	ω_z
Peak to peak value	$0.1530 \text{ [m/s}^2]$	$0.2040 \text{ [m/s}^2]$	$0.2465 \text{ [m/s}^2]$	0.0144 [rad/s]	0.0122 [rad/s]

Relevant means and variances are calculated using MATLAB and the results are presented in Table 2.3. These values are used later to validate the sensor model and are then compared to the mean and variances from the simulated data.

Table 2.3: Calculated mean and variance for the XC90 IMU

	a_x	a_y	a_z	ω_x	ω_z
Mean	$-0.0788 \text{ [m/s}^2]$	$0.1618 \text{ [m/s}^2]$	$9.6730 \text{ [m/s}^2]$	0.0014 [rad/s]	$-4.3099 \cdot 10^{-4} \text{ [rad/s]}$
Variance	$2.9100 \cdot 10^{-4} \text{ [m/s}^2]$	$5.1419 \cdot 10^{-4} \text{ [m/s}^2]$	$9.0586 \cdot 10^{-4} \text{ [m/s}^2]$	$2.2123 \cdot 10^{-6} \text{ [rad/s]}$	$1.8874 \cdot 10^{-6} \text{ [rad/s]}$

To see if the distributions for each sensor axis resembled a Gaussian distribution the MATLAB command `hist` was used and the results are visible in Figure 2.5. As visible the histograms consist of several peaks which is due to the quantization levels. This fact makes it hard to compare them to a Gaussian distribution curve, however the shape of the peaks for the accelerometer in Y- and Z-direction seems to be as if the distributions are normal. In X-direction the peaks does not exactly resemble a Gaussian distribution curve, however the noise is assumed to be Gaussian anyway. As visible the histogram for the gyroscope data in Z-direction seems to have been processed in some way after the quantization for example with a LP filter. This is unfortunately the only signal available during the data gathering and will therefore be used anyway since this will probably not affect the result.

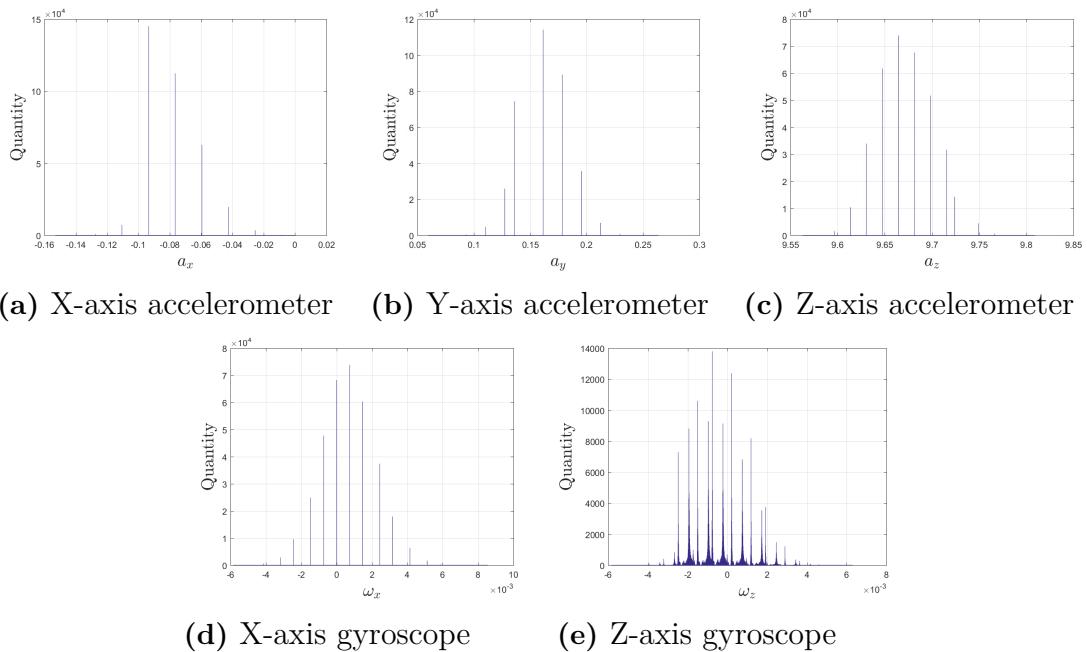


Figure 2.5: Histograms for the time domain data of the IMU

Allan variances and Allan deviation plots for each axis of each sensor was also constructed to find the more specific character of the sensors as well as to simplify the modeling of the sensors. The resulting plots with the Allan deviation (described in Section 2.1.4) calculated for two independent set of measurements are visible in Figure 2.6 and 2.7. By analyzing the Allan deviation plot it is possible to see that the random walk bias starts to affect the variance when the window sizes are larger than 30 seconds for the accelerometers and 80 seconds for the gyroscopes. From this it can be concluded that the random walk bias will probably not have an affect on the bias for this application since the safe stop will not overstep approximately 10 seconds. Another conclusion that can be made is that, to have as small bias affect as possible the offset compensation should be performed within 30 seconds interval for the accelerometers and 80 seconds interval for the gyroscopes during normal usage given that the sampling frequency is 100 Hz.

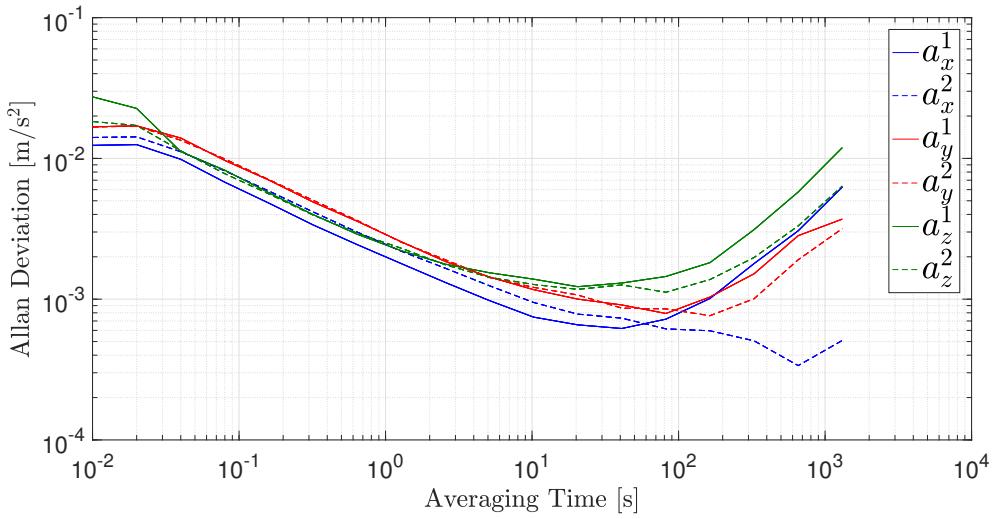


Figure 2.6: Allan deviation plot for the accelerometers

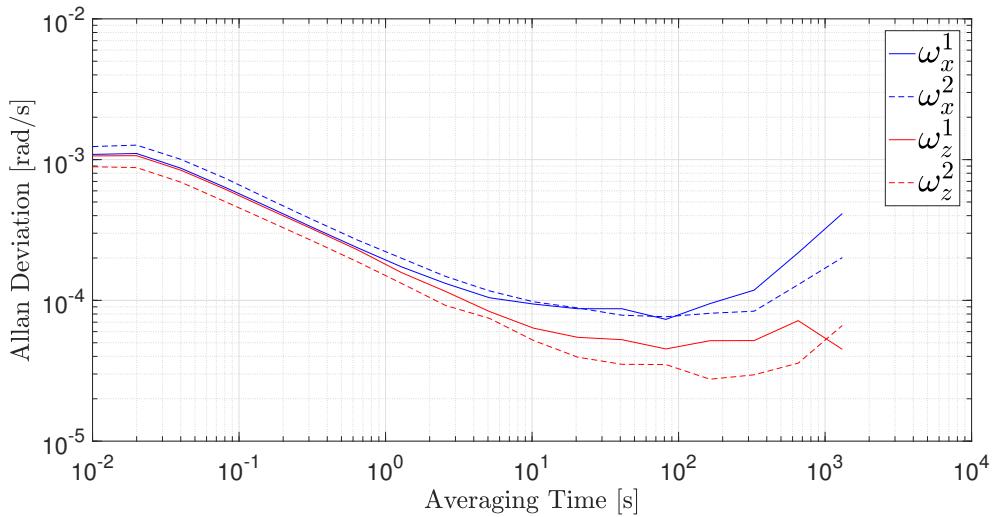


Figure 2.7: Allan deviation plot for the gyroscopes

2.1.6 Modeling of the IMU

To be able to simulate the accelerometers and gyroscopes in a realistic way they first had to be modeled. The modeling was done using data that resembles the data given from an accelerometer and a gyroscope generated from CarMaker. The sampling frequency is set to the same as for the XC90 (100 Hz). The data from CarMaker represented the actual vehicle states from the model without any noise, to make it resemble an actual sensor some dynamical noise was added. The noise added to the data consisted of a constant bias error, a zero mean Gaussian variable representing the white noise perturbation and last a random walk sequence which

represent the wandering of the bias. This yields the following representation of the modeled value

$$y_i = s_i + \epsilon + N_i + R_i, \quad (2.12)$$

where s_i is the true value, ϵ is the constant bias, N_i is a random variable, R_i is the random walk sequence and y_i is the sensor reading. After the signal was created and the noise added the signal was quantized using the same quantization level as for the real signals, see Section 2.1.5. This was done to make the signal resemble the true measurements. In Figure 2.8 the modeling of the simulated measurements are visualized.

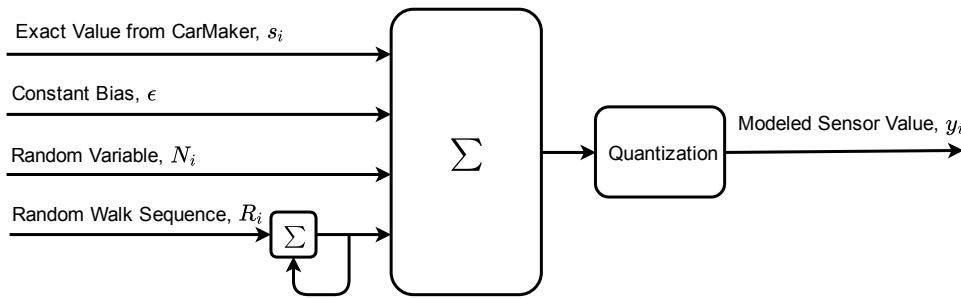


Figure 2.8: Conceptual model of the sensor modeling process

This way of modeling the sensor values was the case for both the accelerometer and the gyroscope. The variances of the white noise random variables and the random variables describing the random walk sequence was though different for the two sensors. These variances was calculated with help from the Allan deviation plot according to [9] which gives

$$\begin{aligned} \sigma_{RW} &= \frac{RW}{\sqrt{t_s}} \\ \sigma_{BI} &= BI \cdot \sqrt{\frac{t_s}{t_{BI}}} \end{aligned} \quad (2.13)$$

where RW and BI are the measurements read from the Allan deviation plot and t_{BI} is the averaging time where the bias instability measurement is made. The values for the actual IMU:s are presented in Table 2.4.

Table 2.4: RW and BI read from the Allan deviation plot in Figure 2.6, 2.7 and Appendix D

	a_x	a_y	a_z	ω_x	ω_z
RW	$0.0200 \text{ [m/s}^2/\sqrt{\text{s}}]$	$0.0291 \text{ [m/s}^2/\sqrt{\text{s}}]$	$0.0244 \text{ [m/s}^2/\sqrt{\text{s}}]$	$0.0019 \text{ }^\circ/\sqrt{\text{s}}$	$0.0018 \text{ }^\circ/\sqrt{\text{s}}$
BI	$9.4374 \cdot 10^{-6} \text{ [m/s}^2]$	$1.0318 \cdot 10^{-5} \text{ [m/s}^2]$	$2.3239 \cdot 10^{-5} \text{ [m/s}^2]$	$8.4273 \cdot 10^{-7} \text{ }^\circ/\text{s}$	$4.8415 \cdot 10^{-7} \text{ }^\circ/\text{s}$

By using these values and the calculated means for the actual sensors it was possible to model all the sensors including a gyroscope reading for pitch rate. The results

2. Sensors

from simulating the sensors during the same conditions as for the actual vehicle, (standing still without any environmental impact during a longer period of time), are visible in Figure 2.9.

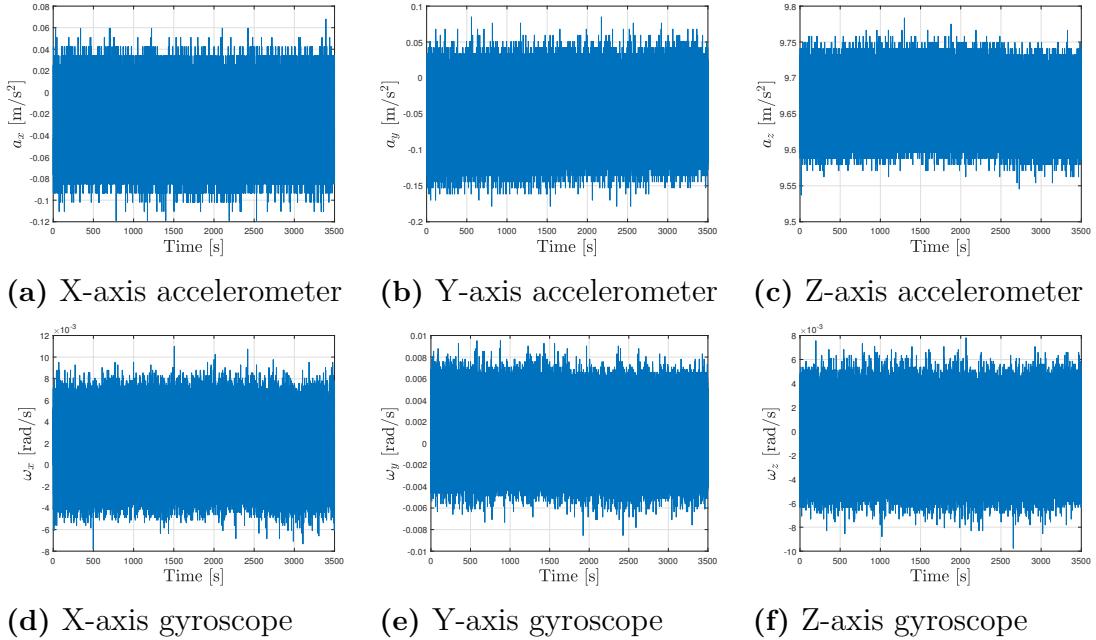


Figure 2.9: The time domain data for the simulated IMU using CarMaker

Analyzing these figures it is possible to see that the simulated data resembles the actual data and the noise levels seems to be rather similar. However it is hard to compare the data and establish the similarity by just analyzing the figures of time domain data.

2.1.6.1 Model Validation

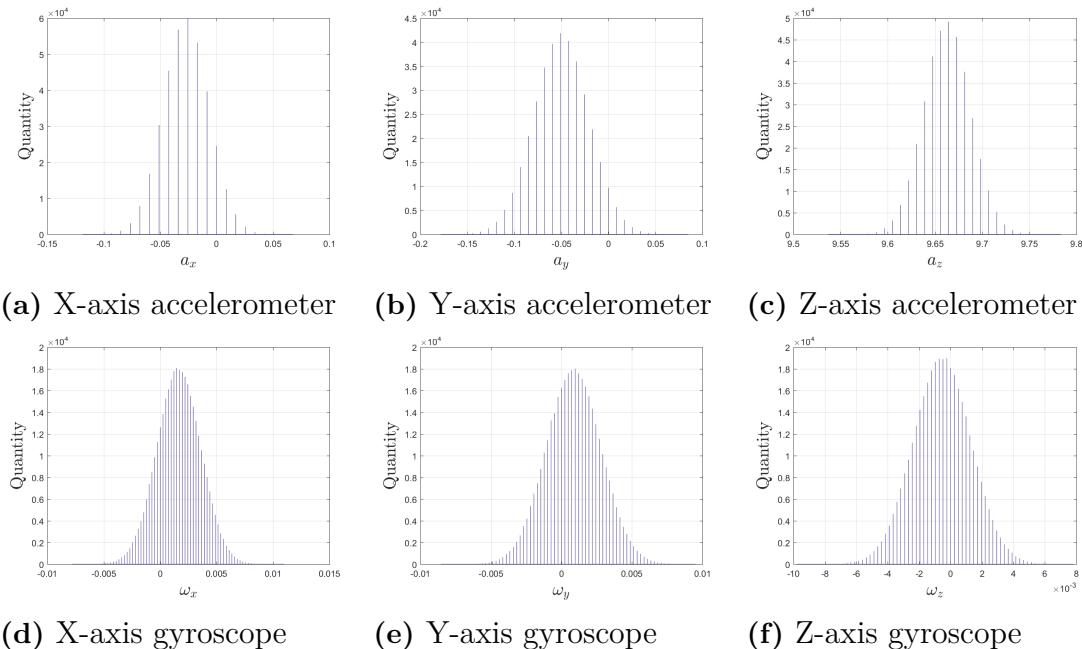
To know that the modeled IMU data could be used in simulation to act as an actual IMU it had to be verified. This was done by comparing data generated from the model and the actual sensor. The comparison was done both for the time domain data and for the Allan deviation plots. To check the repeatability multiple runs of the simulation was done during the same time interval as the actual simulation. From there it was possible to make sure that the modeled sensors behaved as the actual ones even when used multiple times and for longer periods of time.

The mean and variance for these simulated IMU signal was calculated using MATLAB and are presented in Table 2.5. As visible these values are quite similar and in the same order of magnitude as for the actual sensors which verifies that the model works.

Table 2.5: The calculated mean and variance for the simulated IMU

	a_x	a_y	a_z	ω_x	ω_z
Mean	-0.0272 [m/s ²]	-0.0502 [m/s ²]	9.6629 [m/s ²]	0.0016 [rad/s]	$-5.4966 \cdot 10^{-4}$ [rad/s]
Variance	$4.0905 \cdot 10^{-4}$ [m/s ²]	$8.5503 \cdot 10^{-4}$ [m/s ²]	$6.1115 \cdot 10^{-4}$ [m/s ²]	$3.7791 \cdot 10^{-6}$ [rad/s]	$3.3927 \cdot 10^{-6}$ [rad/s]

To verify that the simulated sensor has a similar distribution as the data from the actual sensor histograms of the simulated data was also produced in the same way as for the sensor data. In Figure 2.10 the resulting histograms are presented and as seen the signals are appearing to be Gaussian distributed.

**Figure 2.10:** Histograms for the simulated IMU

The similarity between the histograms for the actual sensor data, see Figure 2.5, and the histograms for the simulated data is not complete, as visible. This might be an affect of that the actual sensor data is not completely Gaussian distributed due to the quantization and that some of the modeling properties is not valid for a longer amount of time, however since the dead reckoning process only will be at maximum 10 seconds these approximations are assumed to hold anyway.

When the repeatability was tested the resulting Allan deviations was illustrated together with the Allan deviations from the actual IMU. The result is visible in Figure 2.11 for the accelerometer X-axis, the rest of the plots is visible in Appendix D. As seen the repeatability of the modeled sensors are high which gives a good indication that the sensors are properly modeled.

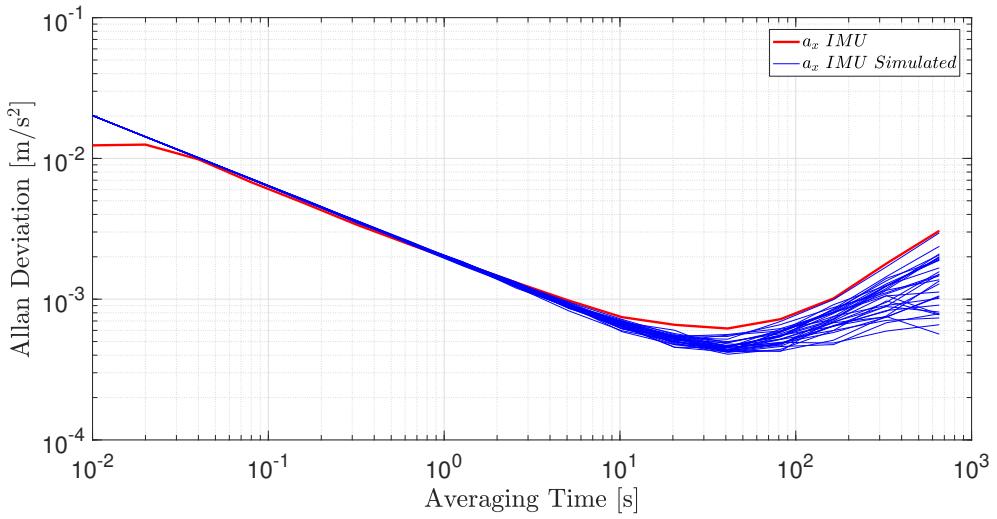


Figure 2.11: Allan deviation plot for the simulated accelerometer X-axis compared to the accelerometer in the XC90

2.2 Odometer

An odometer is a sensor that measures how much a wheel rotates during a specified time. To measure the rotational velocity the wheel is divided into different sections as in Figure 2.12. The rotational velocity is then measured by identifying how long time it takes for a section to pass. By multiplying the rotational velocity with the radius of the wheel, the forward velocity is obtained in [m/s]. The radius of the wheel will consequently estimated in the self-driving vehicles. During a safe stop the last estimated wheel radius before the severe failure will be used during the manoeuvre. This sensor is more accurate in higher velocities since an odometer shows zero if no sections has passed during the specified time and if a wheel rotates slow enough this can be the case.

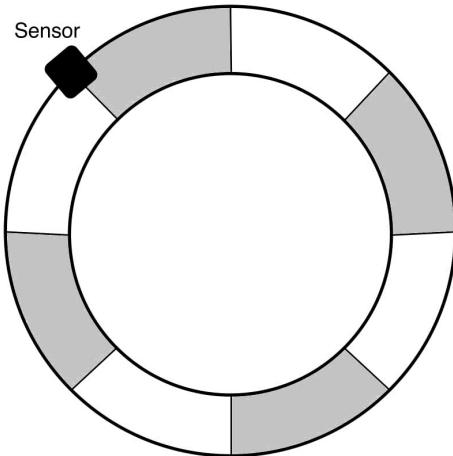


Figure 2.12: Illustration of the odometer technique

2.2.1 Characterization of the Odometers

To characterize the noise of the odometer the vehicle had to have a velocity forward otherwise the sensor will show zero as mentioned above. Sensor data was collected with the XC90 again, however this time the vehicle was traveling forward with a constant velocity. The resulting readings from the front wheel odometers, when traveling at approximately 28 [km/h] or 7.75 [m/s] are visible in Figure 2.13.

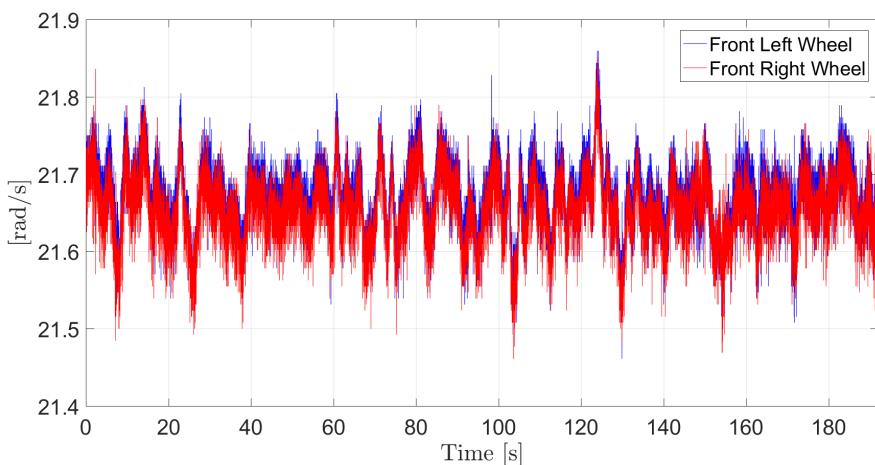


Figure 2.13: The velocity of the front wheels of the XC90 when traveling at constant velocity forward

As visible the velocities are not exactly the same at all time at both wheels which is the case with an odometer. This appears since the velocity might differ at different wheels when the vehicle for example takes a turns or accelerates. When turning for example the outer wheels will have a higher velocity than the inner ones, and when accelerating the driven wheels will have a higher velocity than the non-driven.

To more accurately decide the characteristics of the sensor the distribution was approximated by taking the front left odometer readings minus the front right odometer readings and creating a histogram of that. Doing that removed the affect of a possible change in velocity, however worth noticing is that in the plot the variance is 2σ instead of σ . The histogram is illustrated in Figure 2.14 and as visible it resembles a Gaussian distribution. This sensor also has rather obvious quantization levels (just as the IMU), in this case it was computed to be 0.007813 [rad/s].

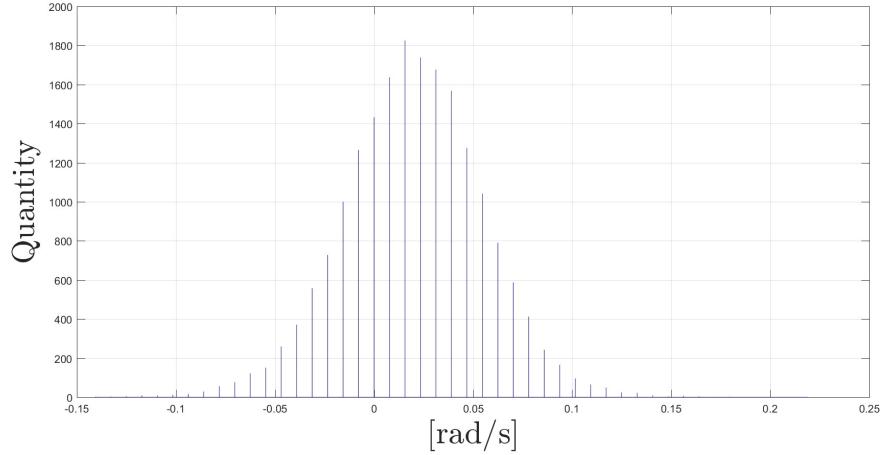


Figure 2.14: Histogram for the odometer data from the front left wheel minus the front right wheel

The variance was calculated for all the wheels and was approximated to $6.3 \cdot 10^{-4}$ [rad/s]. The mean was not possible to exactly determine since there was a human driving the vehicle when gathering the data.

2.2.2 Modeling of the Odometers

To model the odometers CarMaker was used since CarMaker could give the raw wheel speeds in [rad/s], which had the correct behavior in turns and acceleration as mentioned above. To make the generated data resemble the real odometer data some Gaussian noise was added on the generated signals from all wheels with the same variance as calculated for the sensor. There was not added any mean to the generated data since it was not possible to determine. However the estimated wheel radius that will be used is assumed to be miscalculated with 3 % which will give a constant bias at the estimated forward velocity of the wheel. The quantization level of the actual sensors was also applied to the model for more similarity.

The results from simulation are visible in Figure 2.15, both the rotational velocity in [rad/s] and a histogram for the wheel speeds (front left minus front right) in [rad/s]. As visible it is quite similar to the actual sensor data, the velocities have though a more constant appearance in the simulation case. This is probably due to the human error acting on the real sensor data.

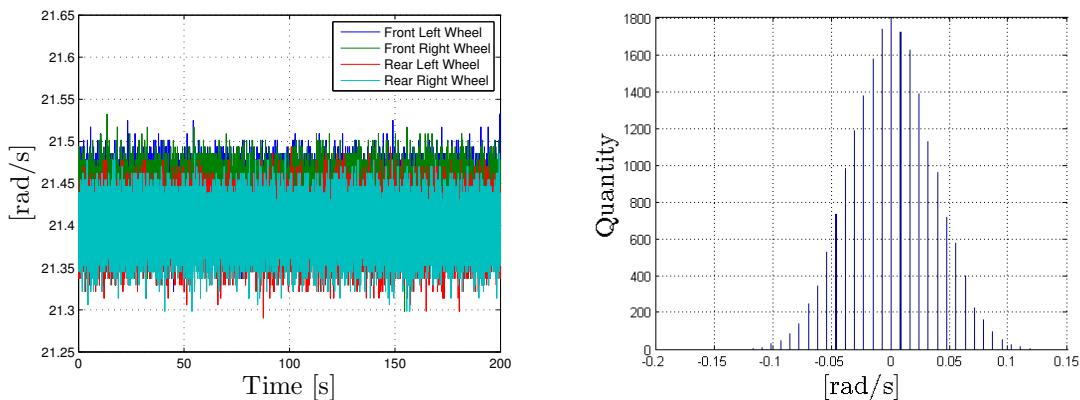


Figure 2.15: The rotational velocity and a histogram of the simulated odometers when traveling at constant velocity forward

The histogram of the simulated odometer data is also very similar to the actual which indicates that the model is valid. However the simulated data has a more even distribution, but as mentioned this is probably due to that there is a human driving when gathering the real sensor data.

2.3 Pinion Angle Sensor

The pinion angle is closely related to the steering wheel angle and the front wheel angles, for small angles it is just a factor separating them. The pinion angle is, compared to the steering wheel angle measured further down the steering column by the steering servo. The pinion angle measurement is assumed to be a more accurate measurement than the steering wheel angle if the purpose is to convert it to front wheel angle. This due to that the length of the steering column can cause a lag between the angle created by the steering wheel and the front wheels.

2.3.1 Characterization of the Pinion Angle Sensor

In the case of a small front wheel angle, approximately within the range $[-15^\circ, 15^\circ]$ there is as mentioned a linear relation between the pinion angle and the front wheel angle. Assuming that the angles are within this range the transformation between the measured pinion angle and the desired front wheel angle is just a constant gain of 16.75. The actual relation is however visible in Figure 2.16 where it is easy to see where the linear region stops.

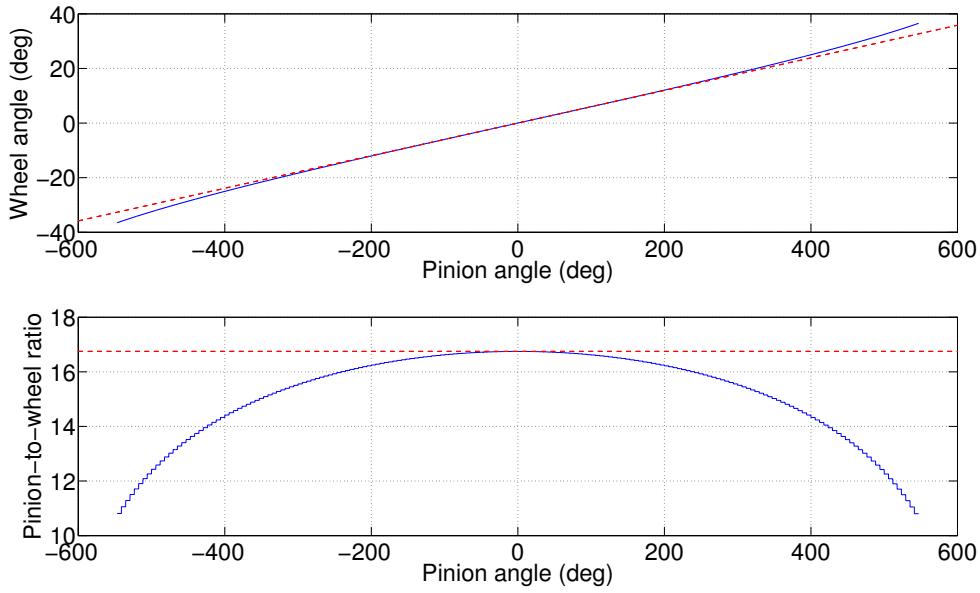


Figure 2.16: The actual ratio between the pinion angle and the front wheel angle compared to a linear approximation of 16.75

To characterize the noise of the sensor the same data gathering case as for the odometer was used, i.e. the vehicle was traveling straight forward with a constant velocity. The resulting pinion angle and front wheel angle calculated with the help from the constant gain is visible in Figure 2.17.

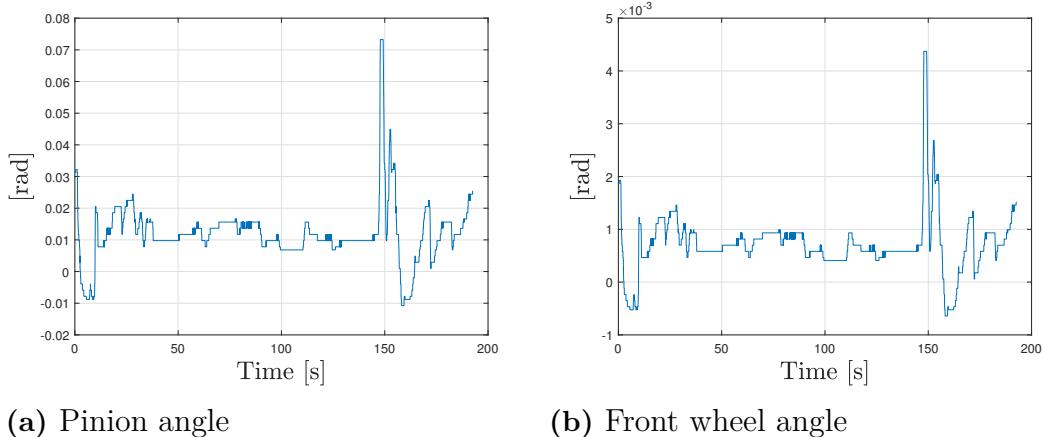


Figure 2.17: The angles created when traveling forward with constant velocity keeping the trajectory as straight as possible

As visible the angles are very close to zero but seems to vary a bit which probably is due to that a human drives. To get a reasonable assumption about the noise of the angle the variance of this signal was calculated (even though that value might be a bit to high compared to the actual noise). It was also hard to determine if the noise

was Gaussian distributed but since the noise levels seemed to be quite low it could be assumed for modeling purposes. The calculated variance for the pinion angle is $9.3576 \cdot 10^{-5}$ [rad 2] and just as the other sensor signals this one was also quantized with a quantization level of 0.0009766 [rad].

2.3.2 Modeling of the Pinion Angle Sensor

In CarMaker it was not possible to read the values of the pinion angle or the front wheel angle directly, however the steering wheel angle was possible to obtain. To make the modeling as realistic as possible the steering wheel angle was first converted to pinion angle given a constant ratio of 16.75/17.4. This ratio was used since the ratio between the steering wheel and the front wheel was established to 17.4.

When the pinion angle was obtained Gaussian noise with a variance of $9.3576 \cdot 10^{-5}$ [rad] was added. Thereafter the quantization was performed before the pinion angle was converted to front wheel angle with the given ratio of 1/16.75. This last conversion is however done within the filtering process to keep the measurements as clean as possible. Figure 2.18 summarizes how the modeled measurement is converted from steering wheel angle to front wheel angle.

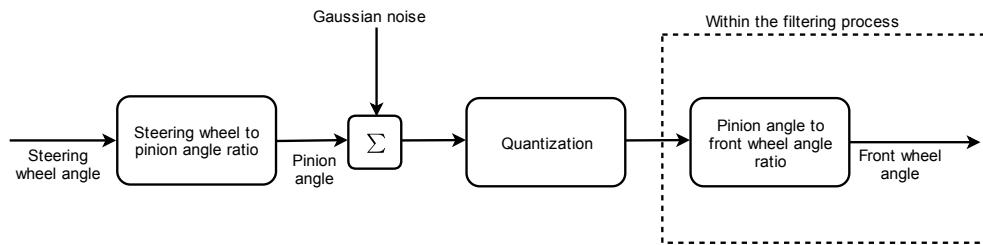


Figure 2.18: Processing of the steering wheel angle obtained in CarMaker to achieve the front wheel angle

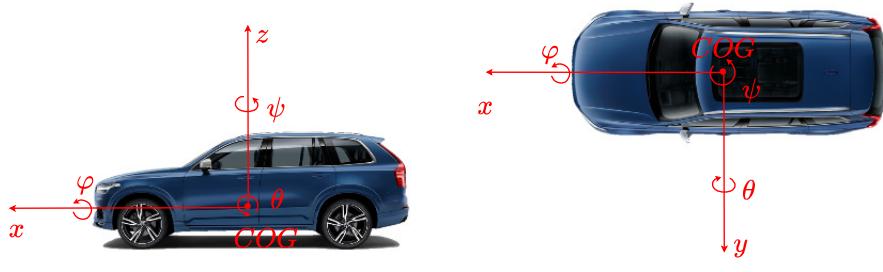
2. Sensors

3

Vehicle Kinematics and Dynamics

The physical processes of a vehicle's motion are complex and it is therefore challenging to create a proper vehicle model. A lot of the models used in vehicle dynamics are simplified, such as the commonly used bicycle model. However it is still important to know how the complexity of a vehicle works to be able to understand and analyze the result of the models.

In this chapter the kinematics and dynamics of a vehicle are designed and modeled so that they resembled an actual vehicle as much as possible. For simplicity a coordinate system is defined in the XC90 COG with the axes and rotations defined as in figure 3.1. This body coordinate system will be referred to as the vehicle coordinate system throughout the thesis and will always be defined in this way.



(a) The XC90 seen from the side (b) The XC90 seen from above

Figure 3.1: The vehicle body fixed coordinate system

3.1 Kinematic Model of a Vehicle

The kinematics of a vehicle describes how it relates to other rigid bodies in space. The basics is that between two rigid bodies with respective coordinate systems there are a kinematic relation that can be described with two components; a translation and a rotation. These relations can then describe how the body of the vehicle relate to other bodies regarding position and motion, both rotational and translational.

In the scope of this thesis, these kinematic relations are of significant importance when interpreting the sensor readings. For example if the IMU sensor is not placed in

the COG of the vehicle, this needs to be compensated for. The accelerometer is also highly affected by the gravitational acceleration influencing the vehicle, which has to be compensated for both when traveling on a flat road as well as when traveling in a slope.

3.1.1 Rotational Movement

There are different ways to describe an object's orientation in space, two examples are by using Euler angles or quaternions. Euler angles describes an objects orientation in space by defining three angles towards a fixed coordinate system, this since any objects orientation can be described by the composition of rotations around three axes, as shown in Figure 3.2.

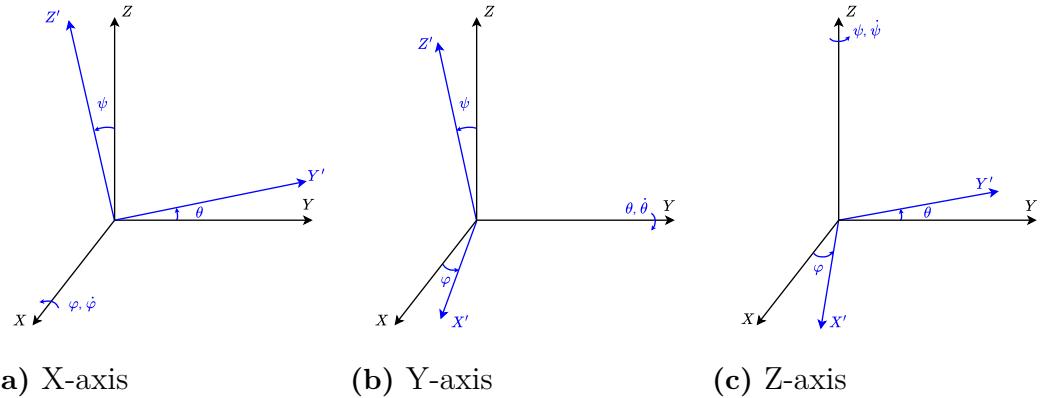


Figure 3.2: Euler angles describing rotations in space

According to [14] a rotation around the X-axis of a coordinate frame, also called roll, can be described by the use of Euler angles as

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \quad (3.1)$$

where φ is the angle that the body has rotated around the axis. Rotations around the Y-axis, also called pitch, can similarly be described as

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (3.2)$$

and rotations around the Z-axis, also called yaw, is given by

$$R_z(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.3)$$

By multiplying these rotational matrices it is possible to go from one coordinate system to another. By multiplying the inverse of the same rotational matrices in the same order the first coordinate system will be obtained again. These three rotational matrices can be multiplied in 12 different ways to rotate from one coordinate system to another. However it is important to always use the same order when multiplying the rotational matrices, since they are non commutative. This description of rotational motion using Euler angles are often used but it has one obvious disadvantage. Since cosine of $\pm 90^\circ$ is zero the loss of one degree of freedom will appear when one of the angles exceed 90° , this phenomena is called Gimbal lock. The reason that the Gimbal lock occurs is since the map from Euler angles to three rotations is not fully covering. At some points the rank of the matrices drops to 2 and that is where the Gimbal lock occurs.

One other way to describe the orientation of an object in space is to use quaternions, where four complex numbers are used to describe the objects orientation in space. Quaternions does not have the same disadvantage as Euler angles regarding Gimbal lock, however they are not as intuitive and require 4 matrices of size 4×4 to describe the orientation instead of Euler angles 3 matrices of size 3×3 . Regarding the scope of this thesis the vehicle will not rotate by $\pm 90^\circ$ around X- or Y-axis and it is assumed that during a safe stop the vehicle will not rotate $\pm 90^\circ$ around the Z-axis. Euler angles are therefore chosen to describe the vehicles orientation. In the XC90 the IMU coordinate frame is not rotated compared to the coordinate system of the vehicle so no compensation for this was done.

3.1.2 Translational Movement

It is not just the rotational movement that is necessary to take into account when describing the kinematics of the vehicle. If the IMU is not placed in COG the two positions will have a relative position, velocity and acceleration, all due to the different placement in the vehicle. A translational movement of the IMU will only affect the accelerometer readings since the angular velocity is the same regardless of position. The relative position can be described as the vector connecting the two points, as seen in [15] and [16], this vector is described in Figure 3.3 as $r_{B/A}$.

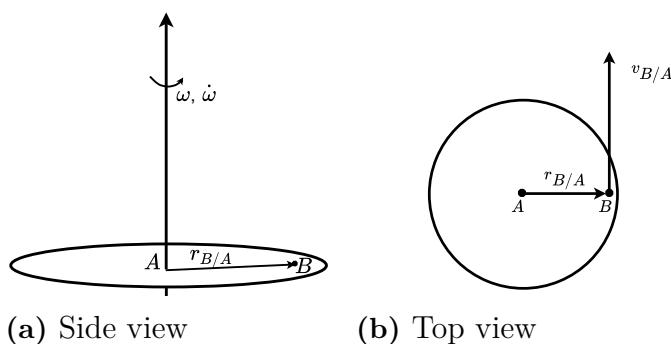


Figure 3.3: Two points placed on the same rotating rigid body

To calculate the relative velocity of the points, $v_{B/A}$ the position vector has to be differentiated as follows;

$$v_{B/A} = \dot{r}_{B/A} = \omega \times r_{B/A} \quad (3.4)$$

where ω is the rotational velocity of the vehicle obtained from the gyroscope. To get the relative acceleration ($a_{B/A}$) this velocity vector, also defined in Figure 3.3 has to be differentiated once again as

$$a_{B/A} = \dot{v}_{B/A} = \dot{\omega} \times r_{B/A} + \omega \times (\omega \times r_{B/A}). \quad (3.5)$$

To take the relative acceleration into consideration when analyzing the data given by the accelerometer is important. If the accelerometer is not placed in COG there will be an additional component to the read acceleration given by this relative acceleration $a_{B/A}$, based on the location of the IMU and the rotational velocity of the vehicle, ω .

In the XC90 the IMU is placed at

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.1586 \\ -0.0203 \\ -0.1123 \end{bmatrix} [m] \quad (3.6)$$

relative to the COG which made the translational compensation necessary. The compensation was done in Simulink, by removing the component calculated in Equation (3.5) from the given acceleration after compensating for possible rotations. The rotational velocity of the vehicle, given by the gyroscope had to be differentiated to solve Equation (3.5), this resulted in a quite noisy relative acceleration. To remove the noise from the calculated value a LP-filter was applied to the rotational velocity before differentiated.

3.1.3 Gravitational Effect on the Accelerometer

The accelerometer placed in the vehicle measures all the accelerations acting on the vehicle's body. This means that the accelerometer also measures the gravitational acceleration acting on the vehicle, which needs to be compensated for. The impact of the gravity in the different axis depends on the vehicles orientation shown for the pitch angle in Figure 3.4.

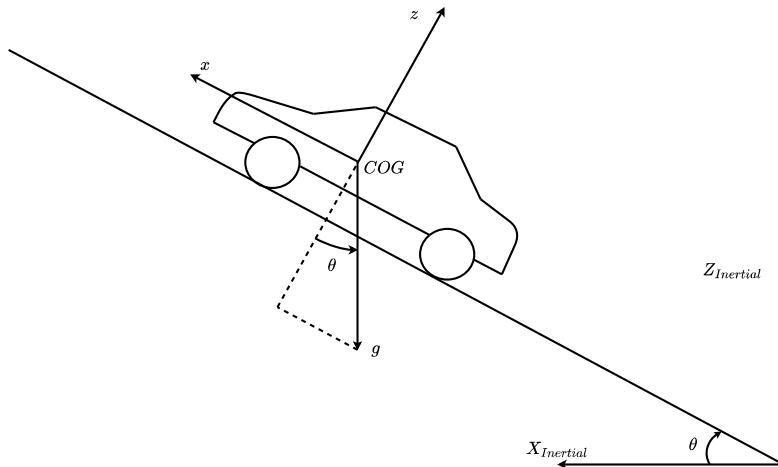


Figure 3.4: Illustration of how the gravitational acceleration affects the accelerometer readings

In the example in Figure 3.4 there will be an additional negative component in the $X_{vehicle}$ -direction which will be seen in the accelerometer readings. This will also be the case when the vehicle is traveling on a road that has a bank angle but in that case the accelerometer will show an extra component in $Y_{vehicle}$ -direction. To compensate for this and for the fact that the accelerometer readings are affected by the angular- and translational velocity of the vehicle in all directions, the equations for the accelerometer are set up as follows;

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \dot{v}_x + \omega_y v_z - \omega_z v_y \\ \dot{v}_y + \omega_z v_x - \omega_x v_z \\ \dot{v}_z + \omega_x v_y - \omega_y v_x \end{bmatrix} + g \begin{bmatrix} -\sin \theta \\ \sin \varphi \cos \theta \\ \cos \varphi \cos \theta \end{bmatrix}. \quad (3.7)$$

In these equations $a_{x,y,z}$ corresponds to the measured accelerations from the accelerometer, φ and θ corresponds to the angles between the COG of the vehicle and the inertial coordinate system. $\dot{v}_{x,y,z}$ corresponds to the acceleration in the vehicle coordinate system. Solving for $\dot{v}_{x,y,z}$ from this equations makes it possible to retain the time derivative of vehicle velocity along the x,y,-z- axis, which is desired for the filtering process.

3.1.4 Angular Velocity to Euler Angular Rate

The angular velocity ($\omega_{x,y,z}$) and the Euler angular rate between different coordinate systems ($\dot{\varphi}, \dot{\theta}, \dot{\psi}$), in this case the vehicle coordinate system and the inertial coordinate system , are usually not the same thing when dealing with rotational movement. As described in [16] the relation can be defined differently using different combinations of Euler angles, however all combinations end up in a same structured matrix

relating the two given by $\omega = \mathbf{T}(\varphi, \theta, \psi)[\dot{\varphi} \ \dot{\theta} \ \dot{\psi}]^T$ and $[\dot{\varphi} \ \dot{\theta} \ \dot{\psi}]^T = \mathbf{T}(\varphi, \theta, \psi)^{-1}\omega$. This transformation matrix is necessary to formulate since the Euler angles are needed for describing the orientation. Using the X-Y-Z convention the transformation matrix \mathbf{T} is computed as follows;

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = R_x(\varphi) \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + R_x(\varphi)R_y(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_x(\varphi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}, \quad (3.8)$$

this gives the Euler angular rates as

$$\begin{aligned} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \varphi & \cos \theta \sin \varphi \\ 0 & -\sin \varphi & \cos \theta \cos \varphi \end{bmatrix}^{-1} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \Rightarrow \\ \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \end{aligned} \quad (3.9)$$

3.1.5 Summarizing of the Kinematic Processing

All processing that was done with the raw IMU data ($a_{x,y,z}$, $\omega_{x,y,z}$) to get accelerations and Euler angular rates in the vehicle coordinate system are summarized in Figure 3.5. The accelerations and Euler angular rates ($\dot{v}_{x,y,z}$, $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$) are after processing in the same coordinate system as the COG of the vehicle, which was desired.

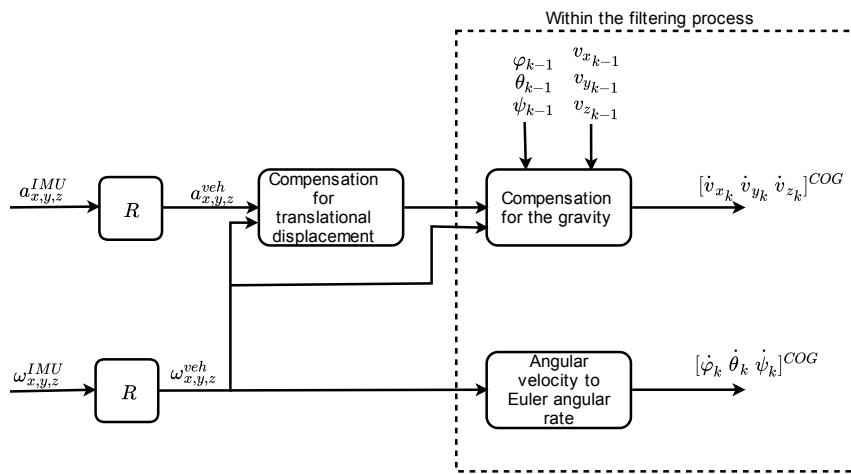


Figure 3.5: Summarizing of the signal processing performed on the data given from the IMU

In CarMaker and when defining the IMU location in the XC90 the translational movement was done before possible rotations. Therefore when compensation for the location of the IMU the rotational compensation had to be done before the translational since matrix operations are non-commutative.

3.1.6 Local to Global Coordinates

To know how the vehicle moves in space it was necessary to translate the vehicles velocity to global positions X and Y. This transform was first done using the vehicles lateral and longitudinal velocity, combined with the vehicle's heading (ψ) according to

$$\begin{aligned}\dot{X} &= v_x \cos(\psi) - v_y \sin(\psi) \\ \dot{Y} &= v_y \cos(\psi) + v_x \sin(\psi).\end{aligned}\tag{3.10}$$

Equation (3.10) gives the global velocities of the vehicle so to be able to get global positions the velocities had to be integrated according to

$$\begin{aligned}X &= \int \dot{X} dt \\ Y &= \int \dot{Y} dt.\end{aligned}\tag{3.11}$$

These global positions could then be used to track how the vehicle moved and how the position was towards a reference. In the estimation/filtering process the origin of the global coordinate system is assumed to be the position of the vehicle when the severe failure happens and the X- and Y- axis frozen as they are in that moment. However since the positions are integrated velocities the distance traveled are similar to the vehicle trip meter which means that the frozen coordinate system will be angled in case the vehicle travels for example uphill.

3.1.7 Chassi Angle

The suspension of a vehicle affects the roll and pitch angle of the chassis when forces are acting on it. For example if a vehicle is driving on a road with a lateral slope the gravitational force will give the chassis a roll angle as seen in figure 3.6. In the same way an acceleration of the vehicle will give a negative pitch angle and deceleration will give a positive pitch angle.

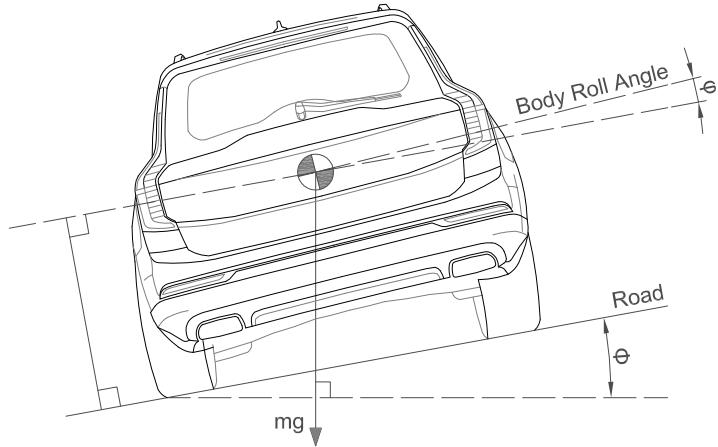


Figure 3.6: A vehicle roll angle due to the lateral forces acting on it

By only using the gyroscope the sum of the possible chassis angle and road gradient will be obtained. However to be able to rotate the coordinate system of the vehicle to the road coordinate system (since the desired position is given in the road's coordinate system) only the chassis angle is desired.

Some tests were made to see how large the chassis pitch and roll angle would be when influenced by accelerations, ($\pm 5 \text{ [m/s}^2]$ in X-direction and $\pm 2 \text{ [m/s}^2]$ in Y-direction due to limitations in the scope). In Figure 3.7 the corresponding chassis roll and pitch angle are illustrated against acceleration. As seen the chassis angle will not exceed $\pm 0.8^\circ$ when the acceleration is kept within the limitations from the scope. By analyzing the figure it seems like there is a linear relation between the acceleration of the vehicle and the chassis angle.

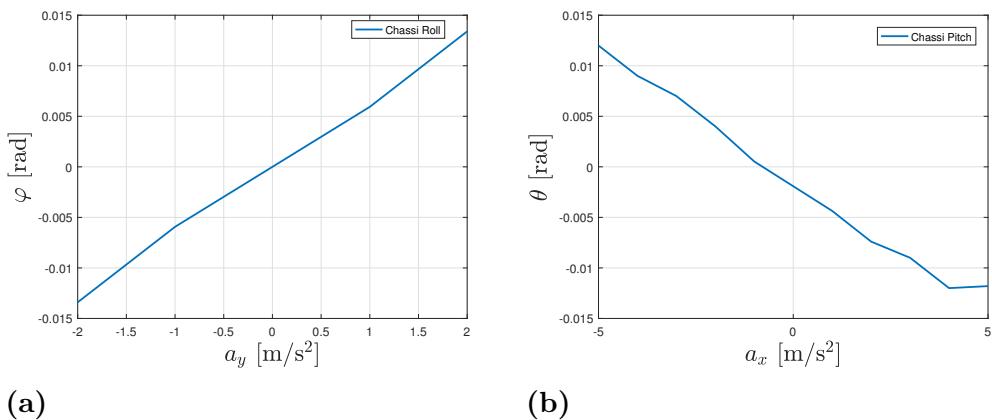


Figure 3.7: Chassis angle due to accelerations

3.2 Dynamical Model of a Vehicle

The dynamics of a vehicle can be described in many different ways with various complexity. The basics with all models is that they in some way with the help of Newton's second law describe how the vehicle changes its position and/or the heading as a function of time. In the scope for this thesis only the lateral dynamics will be considered since the longitudinal velocity will be given quite accurately from the odometers.

3.2.1 Lateral Bicycle Models

The bicycle model is a simplified model description of a vehicle which assumes that the appearance of the vehicle is like a bike, i.e. there are only one front and one rear tire as visible in Figure 3.8.

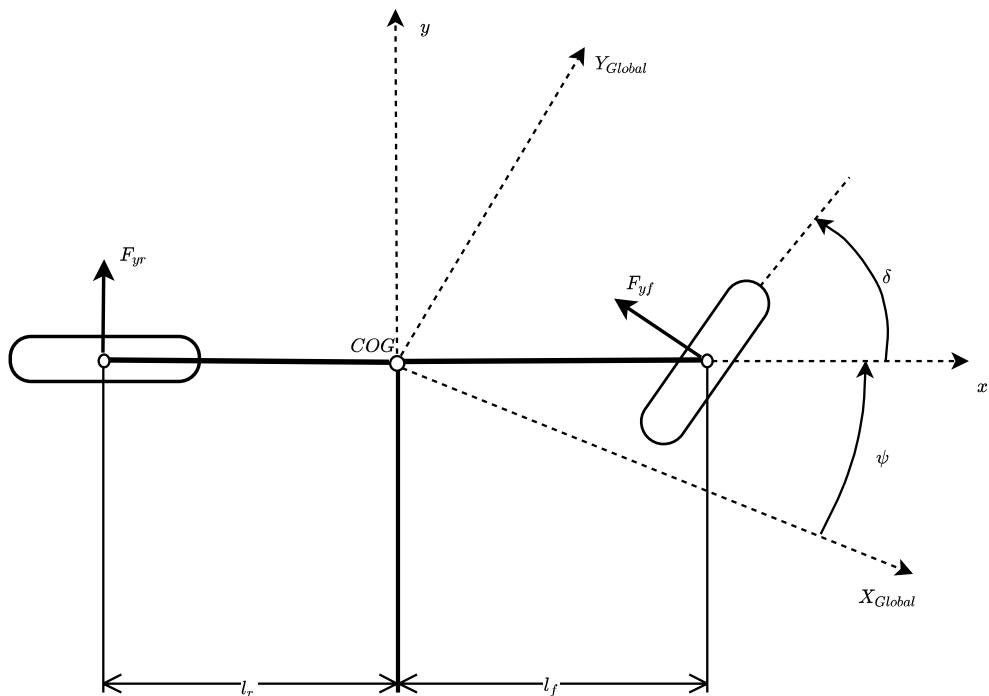


Figure 3.8: A visualization of the conceptual bicycle model

Using this bicycle model approach it is possible to derive different dynamical models which are described in this section.

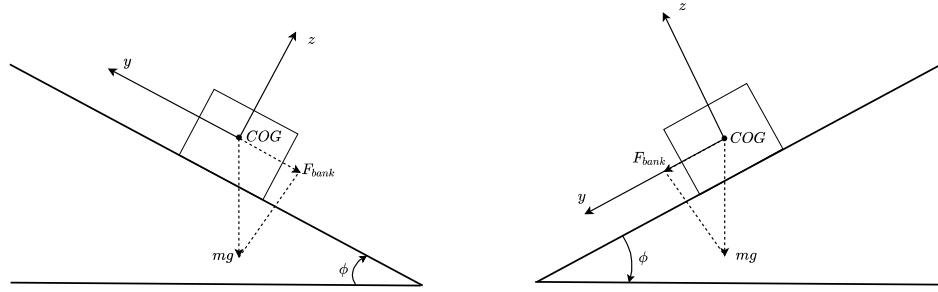
3.2.1.1 Steering-Angle Based Bicycle Model

Given the bicycle description the lateral motion can be described as

$$m(\dot{v}_y + \dot{\psi}v_x) = F_{yf} + F_{yr} + F_{bank} \quad (3.12)$$

$$F_{bank} = mg \sin(-\phi) \quad (3.13)$$

where \dot{v}_y is the lateral acceleration created by the motion and $\dot{\psi}v_x$ is the centripetal acceleration. F_{yf} and F_{yr} are the lateral tire forces acting on the front and rear tires. F_{bank} is an addition in case there are a bank angle on the road. In Equation (3.13) the minus sign in front of the bank angle, ϕ , is a result of how the vehicle coordinate system has been defined. Figure 3.9 is used to describe this; if the bank angle is positive F_{bank} will affect the force in y-direction negatively. If the bank angle instead is negative the force in y-direction will be affected positively.



(a) Vehicle on a road with positive bank angle, the vehicle is traveling into the paper
 (b) Vehicle on a road with negative bank angle, the vehicle is traveling into the paper

Figure 3.9: Illustration of how the bank angle affects the sign on F_{bank}

For the yaw dynamics of the vehicle, moment balance around the Z-axis yields the following equation

$$I_z \ddot{\psi} = l_f F_{yf} - l_r F_{yr} \quad (3.14)$$

where l_f and l_r are the distances from the vehicle COG to the front and rear tire axis respectively.

To model the tire forces F_{yf} and F_{yr} there exist multiple number of different approaches. However for small slip angles it is possible to assume that the tire force is proportional to the slip. If it also is assumed that the vehicle is front wheel steered the expressions for the front and rear slip are

$$\begin{aligned}\alpha_f &= -(\delta - \theta_{Vf}) \\ \alpha_r &= \theta_{Vr}\end{aligned}\quad (3.15)$$

where δ is the steering angle of the front wheel and the angles θ_{Vf} and θ_{Vr} are the angles created between the velocity vector \mathbf{V} and the longitudinal axis X for each tire, see Figure 3.10 for the front tire.

θ_{Vf} and θ_{Vr} can be calculated according to

$$\begin{aligned}\tan(\theta_{Vf}) &= \frac{v_y + l_f \dot{\psi}}{v_x} \\ \tan(\theta_{Vr}) &= \frac{v_y - l_r \dot{\psi}}{v_x}.\end{aligned}\quad (3.16)$$

With a given expression for the slip angle it is possible to express the lateral tire forces for the bicycle model according to:

$$\begin{aligned}F_{yf} &= -2C_f \alpha_f \\ F_{yr} &= -2C_r \alpha_r.\end{aligned}\quad (3.17)$$

C_f and C_r are the cornering stiffness for each of the tires and the factor 2 is due to that there are two front wheels and two rear wheels on an actual vehicle.

One disadvantage with this steering-angle based bicycle model is that when calculating Equation (3.16), the denominator is v_x . This causes problem when $v_x \rightarrow 0$, since that term will go towards infinity. To solve this problem in the filtering/simulation process a constant velocity model is also designed. If $|v_x|$ gets smaller than 1 [m/s], the bicycle model is not used. Instead a constant velocity model is used for the lateral velocity until $|v_x|$ is larger than 1 [m/s] again or the vehicle has come to a stop. An additional thing that might cause problem with this formulation of the steering-angle based bicycle model is the assumption of linear tire dynamics. There are as mentioned dynamics when turning the wheels due to tire relaxation and when performing quick turns the relation between the tire force and the slip gets nonlinear instead of linear. This simplified assumption then affects the phase in both the resulting values from the model, (acceleration and yaw acceleration), since the linear approximation assumes that the vehicle can turn in a way that is

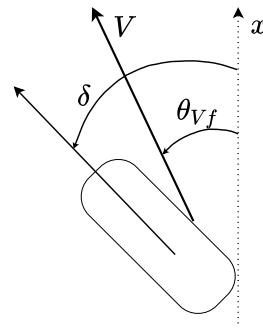


Figure 3.10: The angles created between the front tire and the longitudinal axis X

not possible without that the lateral force from the tires to be saturated by a large slip, see [17]. This behavior can be seen in quick turns (with large slip) where the steering-angle based bicycle model turns faster and therefore gets a positive phase shift compared to the true values.

3.2.1.2 Acceleration Based Bicycle Model

Another model for the lateral dynamics of a vehicle is a model that builds on a linear relationship between the lateral force on the rear tires F_{yr} and the rear tire slip α_r . This relationship for an arbitrary case is visible in Figure 3.11 and the model is valid given that the vehicle state is within the linear part of the curve.

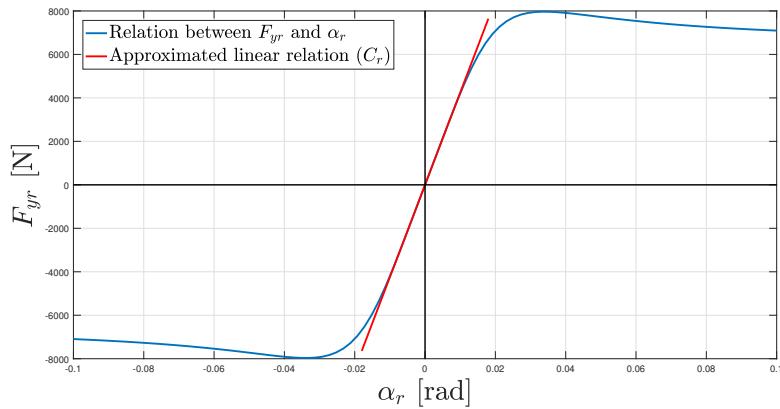


Figure 3.11: The relation between the lateral force on the rear tires and the rear tire slip angle

The slope of this curve is given by the rear tire cornering stiffness C_r which makes it possible to form the following relation

$$\alpha_r = -\frac{F_{yr}}{C_r}. \quad (3.18)$$

Another way to express the rear tire slip is by the following;

$$\alpha_r = \frac{v_{yr}}{v_x}, \quad (3.19)$$

where v_{yr} is the lateral velocity of the rear tires. Combining Equation (3.18) and (3.19) gives an expression for the lateral velocity of the rear tires which is the desired quantity

$$v_{yr} = -\frac{F_{yr}v_x}{C_r}. \quad (3.20)$$

Since this velocity is expressed at the rear tires it needs to be translated to the COG. This is done by adding the relative velocity between the rear tires and the COG ($v_y^{rear/COG}$) which is calculated by

$$v_y^{rear/COG} = l_r \omega_z. \quad (3.21)$$

Here is is neglected that there might be a difference in position in z-direction, this since that contribution would be so small in comparison. The total expression for the lateral velocity in COG is now

$$v_y = -\frac{F_{yr}v_x}{C_r} + l_r\omega_z \quad (3.22)$$

The only unknown in this equation is the rear tire force F_{yr} which can be expressed with help from Newton's second law and the accelerometer data in y-direction, a_y . Force balance for the y-axis and moment balance around the z-axis are set up for the vehicle. Assuming that there might be a steering angle of the front wheel (δ) involves the longitudinal force generated at the front wheels F_{xf} . In this model the impact of a possible bank angle can also be considered, see equation (3.13). The equation for the rear tire force F_{yr} is then given by

$$\begin{aligned} ma_y &= F_{yf} \cos(\delta) + F_{yr} + F_{xf} \sin(\delta) + F_{bank} \\ I_z \ddot{\psi} &= l_f \cos(\delta) F_{yf} - l_r F_{yr} + l_f \sin(\delta) F_{xf}. \end{aligned} \quad (3.23)$$

Solving these equations for F_{yr} and using small angle approximation the following expression is given

$$F_{yr} = \frac{-I_z \ddot{\psi} + l_f m a_y + l_f F_{bank}}{l_f + l_r} \quad (3.24)$$

where all parameters are known except from $\ddot{\psi}$ which can be obtained by differentiating yaw rate, $\dot{\psi}$. All of this gives a linear expression for the lateral velocity v_y that easily can be calculated given the longitudinal velocity and the IMU data as follows

$$v_y = -\frac{(-I_z \ddot{\psi} + l_f m a_y + l_f F_{bank}) / (l_f + l_r) \cdot v_x}{C_r} + l_r \omega_z. \quad (3.25)$$

One advantage with this model is that it will not have a phase shift compared to the true values which as mentioned is a problem with the steering-angle based bicycle model, this phenomena is visible in Figure 3.13.

3.2.2 Simulation of Dynamical Models

The two models was simulated during similar circumstances with the same input data (without bias and noise) and reference given from CarMaker. The velocity in x-direction and the steering angle of the front wheel was used as input to the steering-angle based bicycle model. The accelerometer based bicycle model had the velocity in x-direction, accelerometer data for y-axis and angular velocity in yaw as input. Since CarMaker is a more advanced and comprehensive than the designed dynamical model it is a good reference tool to use for validation of the models.

In Figure 3.12, case 1 from Section 1.4.1 is simulated for both the steering-angle based bicycle model and the acceleration based bicycle model. In case 1 the vehicle travels on a straight road with a velocity of 120 [km/h] and then decelerates with 5

3. Vehicle Kinematics and Dynamics

[m/s²]. Both models gives satisfying results for the velocity in y-direction (around zero). The steering-angle based bicycle model also estimates the yaw acceleration which follows the reference quite good, the behavior of the angular acceleration is probably due to quantization of the sensors.

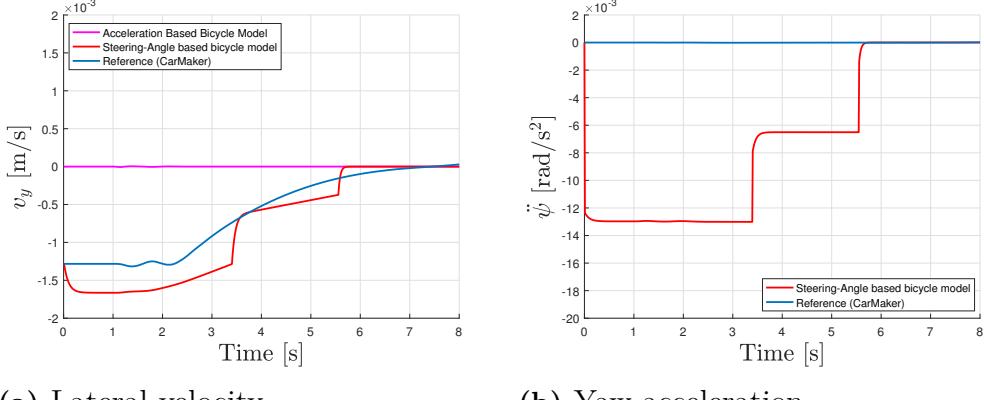


Figure 3.12: Simulated result for the different bicycle models during case 1

The simulation was also performed for case 3, see Section 1.4.1, and the results are shown in Figure 3.13. As visible both of the bicycle models have the same behaviour as the reference but the acceleration based bicycle model is though a bit closer to the reference. The yaw acceleration has a similar shape as the reference however with a bit of over and undershoot. As seen the steering-angle based bicycle model is a bit phase shifted compared to the reference and the acceleration based model which is due to the dynamics of the tires as mentioned previously.

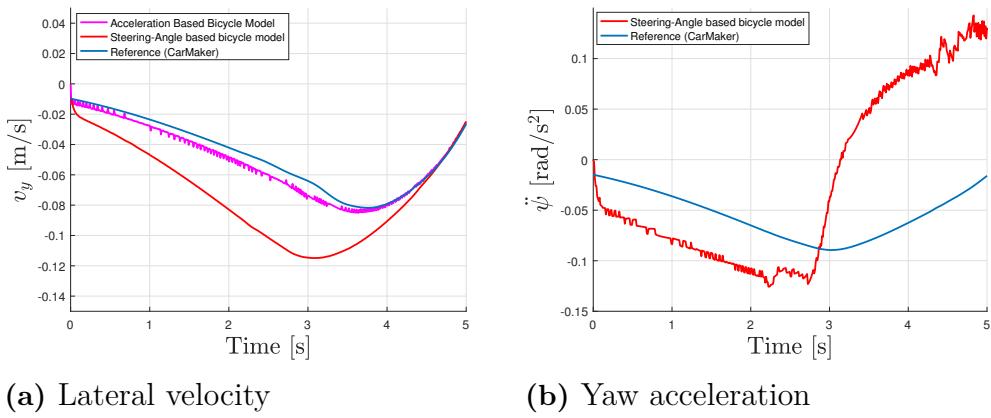


Figure 3.13: Simulated result for the different bicycle models during case 3

4

Filtering Design

This chapter treats the sensor fusion filtering part of the thesis which ties all of the previous parts together. This by using the knowledge about the sensors, vehicle kinematics and dynamics to design a filter. The chapter first considers some background theory on Bayesian filtering which is used to fuse all the data together. The filtering process and two different filtering concepts are then presented. The different concepts uses the sensor data in different ways and applies different dynamical vehicle models. This chapter also includes the treatment of the sensor data when multiple signals are received and last it is presented how the performance of the filters will be evaluated and compared.

4.1 Bayesian Filtering

Bayesian filtering is a commonly used tool to estimate a state vector \mathbf{x} , based on measurements \mathbf{y} from one or multiple sensors, as described in [18]. The state x_k (where k denotes the discrete time step) can be thought of as hidden and observed indirectly through a noisy measurement y_k . The measurements given from one or multiple sensors during a time period K can be described as

$$y_{1:K} = \{y_1, y_2, \dots, y_K\}. \quad (4.1)$$

The property that the states can be thought of as observed through the measurements can intuitively be described by a Bayesian network shown in Figure 4.1.

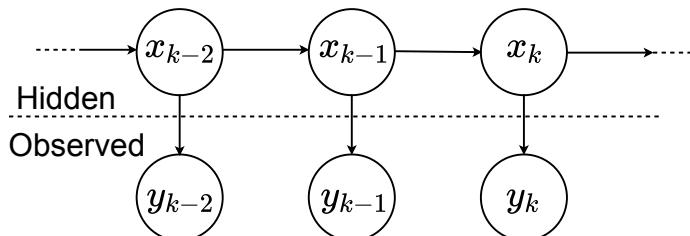


Figure 4.1: A Bayesian network describing a state space model

4. Filtering Design

From this figure two important properties that are assumed in Bayesian filtering can be identified. The first one is that the state vector \mathbf{x} satisfies the Markov property, described in [19]. This denotes that next state in the process only depends on the current state and are conditionally independent of the states before. Secondly, the measurements only depend on the state at respective time instance k . This indicates that both the states and the measurements can be modeled as stochastic Markov processes and the PDF:s, see [19], can be expressed as

$$\begin{aligned} p(x_k|x_{0:k-1}, y_{1:k-1}) &= p(x_k|x_{k-1}) \\ p(y_k|x_{0:k}, y_{1:k-1}) &= p(y_k|x_k). \end{aligned} \quad (4.2)$$

To be able to get a good estimate on the state x_k given all the measurements up until time k ($y_{1:k}$), the goal is to find the PDF for the state at time k . This is given by the conditional probability

$$p(x_k|y_{1:k}). \quad (4.3)$$

According to [20] an expression for this posterior distribution of the state x_k given the measurements $y_{1:k}$ can be set up using Bayes' rule as follows;

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_k, y_{1:k-1}) = \\ \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} &= \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \end{aligned} \quad (4.4)$$

Given Equation (4.4) it is possible to identify the necessary PDF:s to find the posterior distribution of x_k . The first term in the numerator is the likelihood, this describes the probability to get a specific measurement given a specified state. This term is also referred to as the measurement model which is set up to describe how the measurements and states are connected. The second term is the predicted density or prior, which yet is unknown. The numerator is just a normalization factor which can be neglected, this gives the following new expression;

$$\text{Posterior} \propto \text{Likelihood} \cdot \text{Prior}. \quad (4.5)$$

The predicted density needs to be calculated as mentioned, before using Bayes' rule. According to [20] this term is given by the Chapman-Kolmogorov equation in Bayesian filtering;

$$\begin{aligned} p(x_k|y_{1:k-1}) &= \int p(x_k, x_{k-1}|y_{1:k-1})dx_{k-1} = \\ \int p(x_k|x_{k-1}, y_{1:k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} &= \\ \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}. \end{aligned} \quad (4.6)$$

In the last step of this equation two terms can be recognized. The first one is the process model $p(x_k|x_{k-1})$, described in Equation (4.2). This term is a designed model which should describe how the states evolve over time. The second term is the posterior density from the last time step $k - 1$, $p(x_{k-1}|y_{1:k-1})$.

The prediction step (that calculates the prior) and the update step (that takes in the measurements using Bayes' rule) are the basic steps in all Bayesian filtering.

4.1.1 Process and Measurement Model

The process and measurement models used in Bayesian filtering are models that are set up by the user before initializing the filtering process. The process model should describe how the states relate to each other and thereby how the state vector evolves over time. This gives a prediction on how the states changes at each time step. The measurement model should describe how the measurements relate to the states, so if the states are not directly measured the transition between the measurements and the states are described here.

The discrete process and measurement models are usually set up in the following form

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_k) + q_{k-1} \\ y_k &= h_k(x_k) + r_k \end{aligned} \quad (4.7)$$

where f_{k-1} and h_k might be either linear or nonlinear functions depending on the problem. The terms q_{k-1} and r_k are the process and measurement noise respectively.

4.1.2 Kalman Filter

If both the process and measurement model are linear with additive Gaussian noise the discrete process and measurement model for the KF are given by

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + B_{k-1}u_k + q_{k-1} \\ y_k &= H_kx_k + r_k \end{aligned} \quad (4.8)$$

where F_{k-1} is the process matrix, H_k is the measurement model matrix. $x_k \in \mathbb{R}^n$ are the state vector, $q_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ is the process noise, $y_k \in \mathbb{R}^m$ are the measurements and $r_k \sim \mathcal{N}(0, R_k)$ is the measurement noise. If the prior density of the state $p(x_0)$ also is Gaussian the KF provides the optimal filter according to [18].

The posterior density in KF is given by

$$p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \quad (4.9)$$

where $\hat{x}_{k-1|k-1}$ are the estimated states and $P_{k-1|k-1}$ is covariance matrix that gives the uncertainties of the estimates at time $k - 1$. The estimated mean $\hat{x}_{k|k-1}$ and the

covariance $P_{k|k-1}$ are then given by

$$\begin{aligned}\hat{x}_{k|k-1} &= F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_k \\ P_{k|k-1} &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1}.\end{aligned}\tag{4.10}$$

The next step is the measurement update step where the predicted states and the covariance are updated using the measurements y_k . The result is a Gaussian posterior density, $\mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$ whose mean, $\hat{x}_{k|k}$, and covariance $P_{k|k}$ are given by

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - H_k\hat{x}_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} - K_kS_kK_k^T \\ S_k &= H_kP_{k|k-1}H_k^T + R_k \\ K_k &= P_{k|k-1}H_k^T S_k^{-1},\end{aligned}\tag{4.11}$$

where K_k is the so called Kalman gain which is used to correct the process covariance and state estimate. To find the posterior density at each time instant k the prediction and update steps are iterated.

4.1.3 Extended Kalman Filter

The EKF is an extension of the KF but for non-linear filtering problems. In EKF the process and measurement models are linearized and then the same equation as for the KF are used. According to [18] the linearization is done by using first order Taylor series expansion around the expected value. When using this linearization an optimal filter is no longer guaranteed as described in [20]. By assuming additive and Gaussian process and measurement noise the EKF model can be written as

$$\begin{aligned}x_k &= f_{k-1}(x_{k-1}, u_k) + q_{k-1} \\ y_k &= h_k(x_k) + r_k\end{aligned}\tag{4.12}$$

where $f_{k-1}(\cdot)$ is the process model function and $h_k(\cdot)$ is the measurement model function. $x_k \in \mathbb{R}^n$ are the state vector, $q_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ is the Gaussian process noise, $y_k \in \mathbb{R}^m$ are the measurements and $r_k \sim \mathcal{N}(0, R_k)$ is the Gaussian measurement noise.

The model prediction step for the EKF is given by

$$\begin{aligned}\hat{x}_{k|k-1} &= f_{k-1}(\hat{x}_{k-1|k-1}, u_k) \\ P_{k|k-1} &= f'_{k-1}(\hat{x}_{k-1|k-1})P_{k-1|k-1}f'_{k-1}(\hat{x}_{k-1|k-1})^T + Q_{k-1},\end{aligned}\tag{4.13}$$

where the linearization is performed around $\hat{x}_{k-1|k-1}$.

The measurement update step is given by

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - h_k(\hat{x}_{k|k-1})) \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T \\ S_k &= h'_k(\hat{x}_{k|k-1}) P_{k|k-1} h'_k(\hat{x}_{k|k-1})^T + R_k \\ K_k &= P_{k|k-1} h'_k(\hat{x}_{k|k-1})^T S_k^{-1},\end{aligned}\tag{4.14}$$

where the linerization now is performed around $\hat{x}_{k|k-1}$ instead. EKF is relatively simple to implement compared with its performance and is therefore a widely used non-linear filtering method. However since EKF uses local linearization it can cause problems when working with large non-linearities. The process and measurement functions also needs to be differentiable for the filter to be applicable. Another problem is that EKF and KF is restricted to Gaussian process and measurement noise.

4.2 The Filtering Process

The filtering process was performed using a combination of MATLAB and Simulink for both the real and the simulated sensor data from CarMaker. Two different combinations of process and measurement models, named concept 1 and concept 2, (also described in this chapter), was used in an EKF, see Section 4.1.3. An EKF was used since both of the concepts contains nonlinear process equations and concept 2 also nonlinear measurement equations. This made it necessary to use a filter that can handle nonlinearities and since an EKF requires less computational power than for example an UKF according to [20] and also is one of the most commonly used and popular nonlinear filters, see [18], it was chosen.

The parameters used for the dynamical vehicle models (in the filter equations) was for the simulated data the same as used for the vehicle in CarMaker. For the real data the vehicle parameters was measured and estimated. For the simulated data these parameters was assumed to almost exactly coincide with the parameters of the CarMaker vehicle which made it interesting to also try a set of parameters that was not that exact since that is not realistic to assume. To create this set of parameters a load of 300 [kg] is placed in the CarMaker vehicle through the GUI. By adding this load the dynamical vehicle parameters in CarMaker (m , I_z , l_f , l_r) changes and by not changing the parameters used in the filtering process the old parameter set now had realistic error margins. To also get realistic values on the cornering stiffnesses (C_f , C_r) the tires in CarMaker was changed without changing the parameters in the filter.

Another assumption that was made in the filtering process for both the concepts was that when the longitudinal vehicle velocity, (v_x) became lower than 1 [m/s] the models used for the lateral vehicle dynamics (both process and measurement equations) was switched to zero. This was assumed since when the longitudinal

velocity is small the lateral velocity is even smaller. Since it is known that the vehicle brakes the assumption that when the longitudinal velocity is smaller than 1 [m/s] the lateral velocity can be estimated as zero holds. This assumption also removed the problem created by the steering-angle based bicycle model when $v_x \rightarrow 0$.

For both the concepts described below the measurements from the accelerometer in z -direction are neglected resulting in that no state for the velocity/acceleration in z -direction exists. Before this decision was made the difference in result when modeling the z state and not was compared and the difference was minimal, sometimes benefiting the model with z state and sometimes not. Since the difference was almost non-existing and when not using the z state the benefit of a smaller state vector and less complexity was there the decision of excluding it was made.

The roll and pitch state are as mentioned in Section 3.1.7 affected of how the chassis moves. To get the accelerations in correct coordinate system it would be necessary to compensate for these angles created by the chassis. However when analyzing how much the chassis actually moves, it was at most about 1° and when implementing a compensation for that there was no difference on the result down to the millimeter. The decision of not compensate for the chassis pitch and roll was then made since that added more complexity to the filter equations without giving any improvement on the result.

The tuning of the process and measurement noise matrices Q and R was done using trial and error. This except for the parameters in the measurement noise matrices that corresponded to actual sensor readings where the calculated variances was used.

4.3 Filtering Concept 1

In the first concept the acceleration and angular velocity given from the IMU are used as input signals to the process model. The measurement model uses the odometer sensors together with a model based value for the lateral vehicle velocity (acceleration based bicycle model described in Section 3.2.1.2) as measurements. The state vector is then given as

$$\mathbf{x} = [X \ Y \ v_x \ v_y \ \varphi \ \theta \ \psi]^T. \quad (4.15)$$

The inputs to the filter are given by

$$\mathbf{u} = [a_x \ a_y \ \omega_x \ \omega_y \ \omega_z] \quad (4.16)$$

and the measurements as

$$\mathbf{y} = [\omega_{odometer} \ v_y^{AccelerationBased}]. \quad (4.17)$$

4.3.1 Process Model

The derivative of the state vector is obtained by using the kinematic equations described in Section 3.1 and 3.1.6, which yields the following equations;

$$\begin{aligned}\dot{X} &= v_x \cos(\psi) - v_y \sin(\psi) \\ \dot{Y} &= v_y \cos(\psi) + v_x \sin(\psi) \\ \dot{v}_x &= a_x + \omega_z v_y + g \sin(\theta) \\ \dot{v}_y &= a_y - \omega_z v_x - g \sin(\varphi) \cos(\theta) \\ \dot{\varphi} &= \omega_x + \sin(\varphi) \tan(\theta) \omega_y + \cos(\varphi) \tan(\theta) \omega_z \\ \dot{\theta} &= \cos(\varphi) \omega_y - \sin(\varphi) \omega_z \\ \dot{\psi} &= \frac{\sin(\varphi)}{\cos(\theta)} \omega_y + \frac{\cos(\varphi)}{\cos(\theta)} \omega_z\end{aligned}. \quad (4.18)$$

The forward Euler method is used to discretize the model which gives the following discrete process model equations;

$$\begin{aligned}X_{k+1} &= X_k + t_0 (v_{x_k} \cos(\psi_k) - v_{y_k} \sin(\psi_k)) \\ Y_{k+1} &= Y_k + t_0 (v_{y_k} \cos(\psi_k) + v_{x_k} \sin(\psi_k)) \\ v_{x_{k+1}} &= v_{x_k} + t_0 (a_x + \omega_z v_{y_k} + g \sin(\theta_k)) \\ v_{y_{k+1}} &= v_{y_k} + t_0 (a_y - \omega_z v_{x_k} - g \sin(\varphi_k) \cos(\theta_k)) \\ \varphi_{k+1} &= \varphi_k + t_0 (\omega_x + \sin(\varphi_k) \tan(\theta_k) \omega_y + \cos(\varphi_k) \tan(\theta_k) \omega_z) \\ \theta_{k+1} &= \theta_k + t_0 (\cos(\varphi_k) \omega_y - \sin(\varphi_k) \omega_z) \\ \psi_{k+1} &= \psi_k + t_0 \left(\frac{\sin(\varphi_k)}{\cos(\theta_k)} \omega_y + \frac{\cos(\varphi_k)}{\cos(\theta_k)} \omega_z \right)\end{aligned}. \quad (4.19)$$

Since these process equations are as mentioned nonlinear a Jacobian has to be calculated to perform the filtering process. The calculated Jacobian for this concept is presented in Appendix E.

4.3.2 Measurement Model

The generated measurement value for v_y is obtained by Equation (3.25) in Section 3.2.1.2, where $\dot{\omega}$ is obtained by differentiating the sensor signal ω and then applying an LP filter to remove the noise.

Since the measurements are represented by a state in the state vector the measurement model is linear and can be represented by the following matrix:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & \frac{1}{R^{whl}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4.20)$$

4.4 Filtering Concept 2

In this concept the state vector is given as

$$\mathbf{x} = [x \ Y \ v_x \ v_y \ \dot{v}_x \ \dot{v}_y \ \varphi \ \theta \ \psi \ \dot{\varphi} \ \dot{\theta} \ \dot{\psi}]^T, \quad (4.21)$$

containing states for global longitudinal- and lateral- position, velocity and acceleration as well as the roll-, pitch- yaw-angle and their rates. The input to the filtering process is the pinion angle sensor measurement (δ), which is used for the steering-angle based bicycle model, (described in Section 3.2.1.1), in the process equations.

$$\mathbf{u} = \delta. \quad (4.22)$$

The measurement update step for this concept uses measurements from the IMU and odometer which forms the measurement vector as follows

$$\mathbf{y} = [a_x \ a_y \ \omega^{odometer} \ \omega_x \ \omega_y \ \omega_z]^T. \quad (4.23)$$

4.4.1 Process Model

The process model uses as mentioned the steering-angle based lateral bicycle model equations to describe the vehicle's lateral velocity and heading. In this concept however this model is also combined with two other models to describe the longitudinal and lateral movement and the evolution of roll and pitch, the constant velocity and constant acceleration models. For the constant velocity model the velocity is assumed to be constant disregarding some noise. Assuming the state vector is formed as $\mathbf{x}(t) = [p(t) \ v(t)]^T$, where $p(t)$ is the position in one dimension and $v(t)$ is the velocity the constant velocity model is formed as:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \tilde{q}(t). \quad (4.24)$$

For the constant acceleration model the acceleration ($a(t)$) is assumed to be constant instead, forming a state vector as $\mathbf{x}(t) = [p(t) \ v(t) \ a(t)]^T$ it can be described as:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \tilde{q}(t) \quad (4.25)$$

For both models \tilde{q} is assumed to be a Gaussian noise that acts on the velocity in the constant velocity model and on the acceleration in the constant acceleration model.

Given these models the continuous state evolution process equations can be set up as follows:

$$\begin{aligned}
 \dot{X} &= v_x \cos(\psi) - v_y \sin(\psi) \\
 \dot{Y} &= v_y \cos(\psi) + v_x \sin(\psi) \\
 \dot{v}_x &= \dot{v}_x \\
 \dot{v}_y &= \frac{2C_f}{m} \left(\delta - \tan^{-1} \left(\frac{v_y + l_f \dot{\psi}}{v_x} \right) \right) + \frac{2C_r}{m} \left(-\tan^{-1} \left(\frac{v_y - l_r \dot{\psi}}{v_x} \right) \right) - v_x \dot{\psi} \\
 \ddot{v}_x &= 0 \\
 \ddot{v}_y &= 0 \\
 \dot{\varphi} &= \dot{\varphi} \\
 \dot{\theta} &= \dot{\theta} \\
 \ddot{\varphi} &= 0 \\
 \ddot{\theta} &= 0 \\
 \ddot{\psi} &= 0
 \end{aligned} \quad . \quad (4.26)$$

To use these time domain state space representation in the filtering process the equations had to be discretized. This is done using the forward Euler method, however the steering-angle based bicycle model equations are moved to act on the 6th state (the lateral acceleration state) and the 12th state (the yaw-rate state). The reason these equations are moved from the 4th state (the lateral velocity state) and the 9th state (the yaw state) are so that the measurements from the accelerometer and gyroscope can be used and affect the velocity and yaw state through the acceleration and yaw-rate states. The resulting discrete state process equations are:

$$\begin{aligned}
 X_{k+1} &= X_k + t_0 (v_{x_k} \cos(\psi_k) - v_{y_k} \sin(\psi_k)) \\
 Y_{k+1} &= Y_k + t_0 (v_y \cos(\psi_k) + v_x \sin(\psi_k)) \\
 v_{x_{k+1}} &= v_{x_k} + t_0 \dot{v}_{x_k} \\
 v_{y_{k+1}} &= v_{y_k} + t_0 \dot{v}_{y_k} \\
 \dot{v}_{x_{k+1}} &= \dot{v}_{x_k} \\
 \dot{v}_{y_{k+1}} &= \frac{2C_f}{m} \left(\delta_k - \tan^{-1} \left(\frac{v_{y_k} + l_f \dot{\psi}_k}{v_{x_k}} \right) \right) + \frac{2C_r}{m} \left(-\tan^{-1} \left(\frac{v_{y_k} - l_r \dot{\psi}_k}{v_{x_k}} \right) \right) - v_{x_k} \dot{\psi}_k \\
 \varphi_{k+1} &= \varphi_k + t_0 \dot{\varphi}_k \\
 \theta_{k+1} &= \theta_k + t_0 \dot{\theta}_k \\
 \psi_{k+1} &= \psi_k + t_0 \dot{\psi}_k \\
 \dot{\varphi}_{k+1} &= \dot{\varphi}_k \\
 \dot{\theta}_{k+1} &= \dot{\theta}_k \\
 \dot{\psi}_{k+1} &= \frac{l_f 2C_f}{I_z} \left(\delta_k - \tan^{-1} \left(\frac{v_{y_k} + l_f \dot{\psi}_k}{v_{x_k}} \right) \right) - \frac{l_r 2C_r}{I_z} \left(-\tan^{-1} \left(\frac{v_{y_k} - l_r \dot{\psi}_k}{v_{x_k}} \right) \right)
 \end{aligned} \quad . \quad (4.27)$$

Also in this case the process equations are nonlinear and the resulting Jacobian needed for the filtering process is presented in Appendix E.

4.4.2 Measurement Model

The measurement model matrix is formed using the kinematic relations described in Section 3.1. For the longitudinal velocity the state is divided by the wheel radius so that the measurement from the odometer can be used.

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \dot{v}_x - (-\sin(\varphi)\dot{\theta} + \cos(\theta)\cos(\varphi)\dot{\psi})v_y - g\sin(\theta) \\ \dot{v}_y + (-\sin(\varphi)\dot{\theta} + \cos(\theta)\cos(\varphi)\dot{\psi})v_x + g\sin(\varphi)\cos(\theta) \\ \frac{v_x}{R^{whl}} \\ \dot{\varphi} - \sin(\theta)\dot{\psi} \\ \cos(\varphi)\dot{\theta} + \cos(\theta)\sin(\varphi)\dot{\psi} \\ -\sin(\varphi)\dot{\theta} + \cos(\theta)\cos(\varphi)\dot{\psi} \end{bmatrix} \quad (4.28)$$

In this concept also the measurement model matrix contains nonlinear equation which creates the need of a Jacobian. The calculated Jacobian for this matrix is presented in Appendix E.

4.5 Treatment of Sensor Data

When using more than one set of IMU:s the mean from the sensors are used in the filter process, see the following expression;

$$IMU_{mean} = \frac{IMU_1 + IMU_2 + IMU_3}{3}. \quad (4.29)$$

By using multiple IMU:s the impact from possible sensors biases and the variances of the used signals should decrease according to [11].

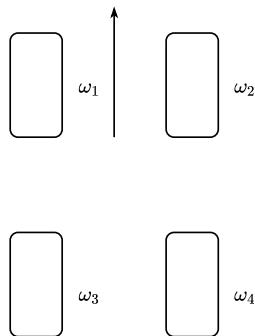


Figure 4.2: The wheels numbering seen from above

The knowledge that the vehicle will decelerate during a safe stop was used when treating the odometer data. When a vehicle brakes the tires might lock but will never rotate faster than the actual velocity. This knowledge made it favourable to use sensor data from the tires that rotated with the highest velocity. However also to take into consideration that during a turn the inner wheels rotated slower than the outer wheels the mean of the wheels with highest rotational speed on left respective right side of the vehicle was used as the odometer signal in the filter process. In Figure 4.2 the numbering of the vehicle's wheels are described and the following expression explains how the odometer value is created;

$$\omega^{odometer} = \frac{\max(\omega_1, \omega_3) + \max(\omega_2, \omega_4)}{2}. \quad (4.30)$$

4.6 Filter Evaluation Process

To compare the filter concepts and different tuning parameters the filter performance was evaluated. This was done by comparing the states and the MSE for the resulting estimation errors, defined below.

4.6.1 Defined Estimation Errors

The defined estimation errors (e_{lon} , e_{lat} , e_ψ) are used to compare the different filter concepts and tuning parameters for the different cases. They are defined according to Figure 4.3 where the coordinate system of the estimated vehicle is rotated to the same yaw angle as the coordinate system of the actual vehicle. The errors are then defined from the estimated position to the actual position by moving longitudinal and lateral as the figure shows.

4. Filtering Design

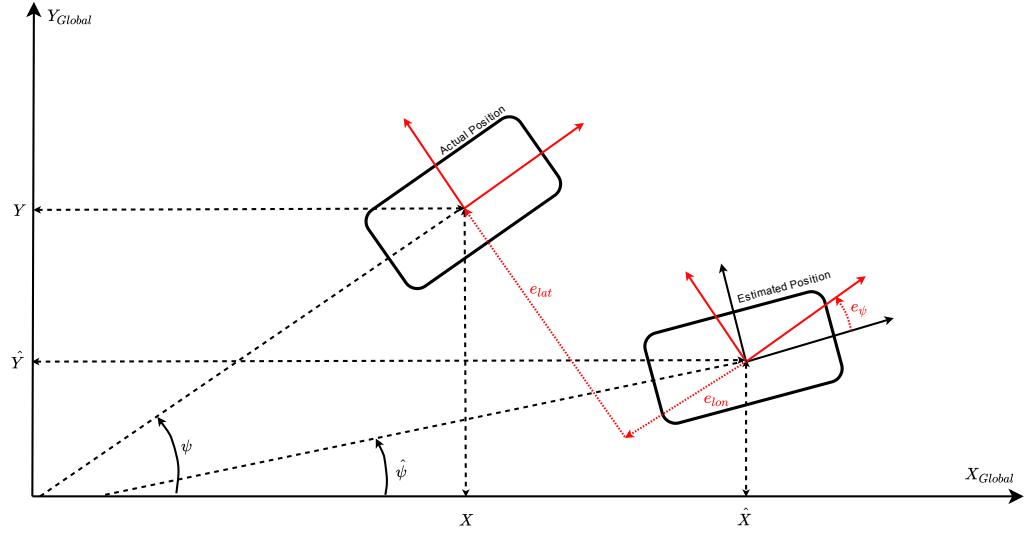


Figure 4.3: The defined estimation errors

To calculate these errors for all possible position configurations between the actual vehicle and the estimated vehicle trigonometry was together with and the fact that two rectangles with the same hypotenuse was formed, see Figure 4.4. In this figure the estimated vehicle is already compensated to have the same yaw angle as the actual vehicle, by using a rotation around the Z-axis with e_ψ . Using this configuration two angles; α and β was created which could be used together with the hypotenuse (c) and the yaw angle (ψ) to calculate e_{lon} and e_{lat} .

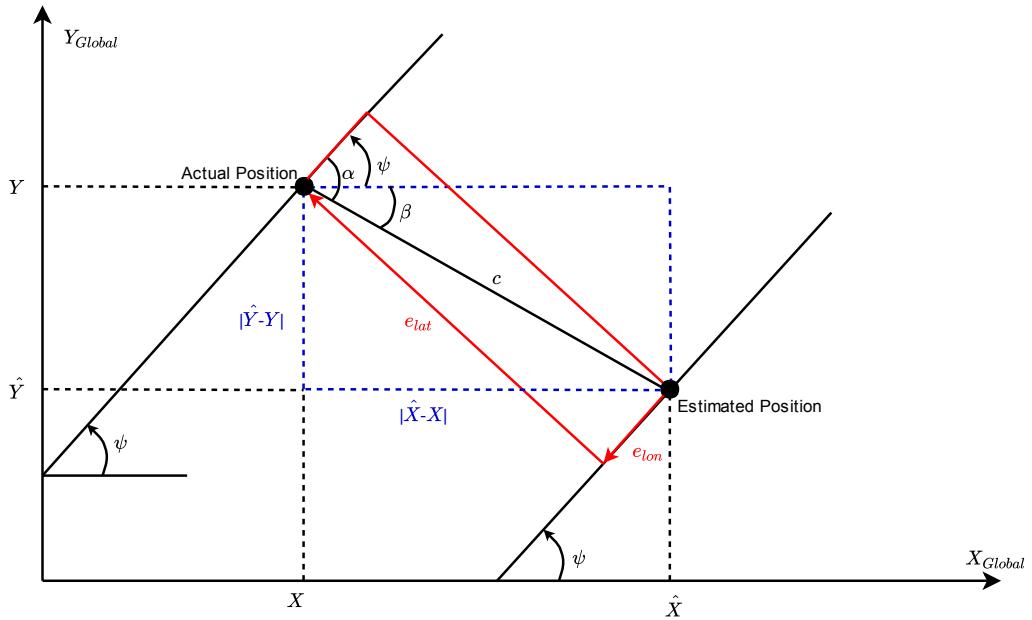


Figure 4.4: Trigonometry to calculate the errors

To get the correct expressions for e_{lon} , e_{lat} and α some conditions had to be set up for those. These are defined for each time instance k as:

$$\begin{aligned} c &= \sqrt{(\hat{X}_k - X_k)^2 + (\hat{Y}_k - Y_k)^2} \\ \beta &= \cos^{-1} \frac{|\hat{X}_k - X_k|}{c} \\ \alpha &= -\text{sign}(|\hat{Y}_k| - |Y_k|) \cdot |\beta| + \text{sign}(|\hat{X}_k| - |X_k|) \cdot |\psi| \\ e_{lon_k} &= \cos(|\alpha|) \cdot c \\ e_{lat_k} &= \sin(|\alpha|) \cdot c \\ e_{\psi_k} &= \psi_k - \hat{\psi}_k. \end{aligned} \quad (4.31)$$

For α the signs on β and ψ had to change according to Equation (4.31) since depending on where the estimate was located α was created by different combinations of β and ψ . For the sign of e_{lon} and e_{lat} all possible positions configurations was tried and the following expressions was set up for each time instance k :

$$\begin{aligned} \hat{X}^0 &= \hat{X}_k - \frac{|\hat{Y}_k|}{\tan(|\psi_k|)} \\ X^0 &= X_k - \frac{|Y_k|}{\tan(|\psi_k|)} \\ \text{sign}(e_{lon_k}) &= \text{sign}(|X_k| + |Y_k| - ||\hat{X}_k| + |\hat{Y}_k||) \\ \text{sign}(e_{lat_k}) &= \text{sign}(\hat{Y}_k) \cdot \text{sign}(\hat{X}^0 - X^0), \end{aligned} \quad (4.32)$$

where \hat{X}^0 and X^0 are the position where the yaw-line crosses the X-axis for respective position.

4.6.2 Mean Squared Error

A proper way to assess the quality of an estimator or predictor is to analyze the MSE:s produced. The MSE:s are defined as the calculated mean of all the the squared errors produced during the filtering process. Since it takes whole the process into account it can give a good indicator on the overall performance of the filter. The MSE is calculated as follows:

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (\hat{\mathbf{x}}_k - \mathbf{x}_k)^2. \quad (4.33)$$

Since the MSE only produces one single number it is also possible to only consider the squared errors $(\hat{\mathbf{x}} - \mathbf{x})^2$ to get an indication on how the performance changes over time.

4. Filtering Design

5

Simulation Results

In this chapter results from the filtering process using the simulated data are presented. The results are presented using both the concepts for filtering and the reference are the exact states given from CarMaker. For all filtering processes the initial conditions are assumed to be fully known since previous sensor data from the environmental sensors are assumed to be accessible even when a failure has occurred.

To test the robustness of the filtering estimation process different scenarios was set up in the simulation. The different scenarios tested was:

- Different numbers of IMU:s
- Increasing the constant bias on the IMU:s from zero to two times the standard bias
- Changing the initial velocity of the vehicle from 10-120 [km/h] for case 1 and from 10-50 [km/h] for case 3
- Adding an additional load of 300 [kg] (at $p=[-1.5 \ 0 \ 0.5]^T$ relative to the COG) to affect the dynamical vehicle parameters

The simulation results from the different scenarios are presented case wise with figures of all the state estimates from both the concepts in the "standard" scenario i.e. the highest velocity appropriate for the case, three IMU sensors with constant bias and no extra load. To achieve a distribution of the errors these standard cases are also simulated for 1000 runs. The results from the other scenarios are presented as the mean of e_{lon} , e_{lat} and e_ψ squared produced during 30 runs of each scenario. This to retain a statistical result and so that possible trends that appear when the velocity and bias changes could be distinguishable.

The filtering results could also be affected by the placement of the IMU:s since the placement needs to be compensated for to achieve correct readings. Therefore it was tested to place the IMU:s at different potential location, as for example under the passenger seat or under the gearbox. It could with those tests be stated that it was possible to compensate for the position relative the COG so that almost no difference on the accelerometer readings could be seen. Given this result the IMU:s was placed, one at the actual position in an XC90 and the other two in COG.

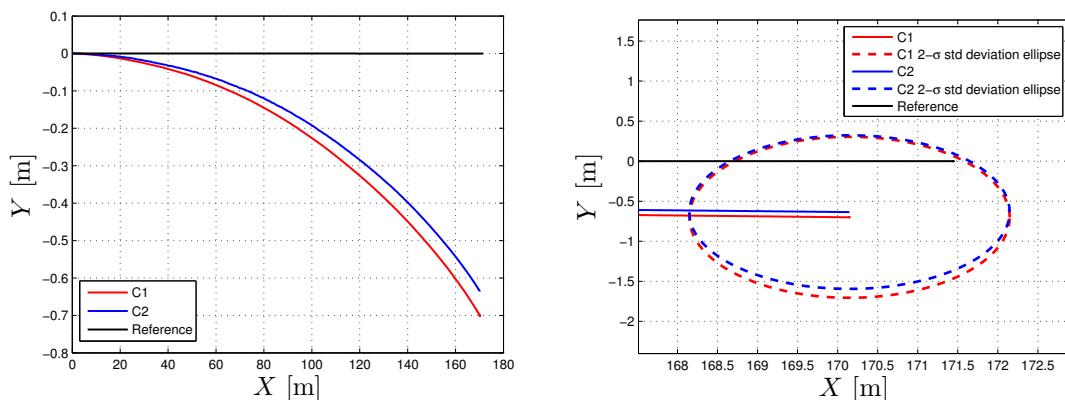
5. Simulation Results

The tuning parameters (Q and R matrices) used for the filtering process with the simulated data are the same for all cases and scenarios, however different for the two concepts. The used Q and R matrices are presented in Appendix G. The Q matrices for both the concepts was tuned so that the position errors would be as small as possible at the same time as the variance for the states would be reasonable small still keeping the reference within the 2σ region. If the reference is not kept within the 2σ region the uncertainty of the filter estimate is to small, meaning that the filter thinks it performs better than it actually does. For both the concepts the uncertainty on the X-state was set higher than the uncertainty on the Y-state in the Q matrix. This was done since it is known that the vehicle always will start traveling in positive X-direction and will therefore move further in that direction, giving room for more uncertainty. For the R matrix the knowledge about the sensor noise was used for tuning so that the correct sensor variance could be used for each value.

5.1 Case 1

The result from the filtering estimation process for case 1, (see Section 1.4.1) are presented below. Figures are visible for all the state estimates for both the concepts together with a reference generated from CarMaker. The figures are describing the results when as mentioned the initial velocity is 120 [km/h], using 3 IMU sensors and fully known dynamical vehicle parameters. When the two concepts are compared the filtering was based on the same data set.

In Figure 5.1a the resulting global position estimates (X and Y) are illustrated against each other which gives an indication on how the vehicle has traveled. In the figure the results from both the concepts are visible and since the filtering process is based on the same data set the results are quite similar. The axis size of the plot are though quite different for the two estimates which might give an indication on that the estimated vehicle has traveled further in Y -direction than it actually has (about 0.7 [m] in Y -direction and about 160 [m] in X -direction). In Figure 5.1b, Figure 5.1a is zoomed in and the covariance matrices for the final global position estimates are forming 2σ standard deviation ellipses. The meaning of these ellipses are that with 95 % probability the final global position estimates will be within these regions.



(a) The estimated X- and Y- positions illustrated against each other (b) 2- σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 5.1: The estimated X- and Y- positions for case 1

5. Simulation Results

In Figure 5.2 the estimated global positions are presented separately, and a zoomed in version of the estimate in X-direction is visible. In X-direction the estimates for both the concepts are quite similar. In Y-direction concept 2 is slightly closer to the reference than concept 1. The 2σ standard deviations for the estimated global positions are also illustrated in the same figure. The standard deviations grows with time, which is expected since dead reckoning was used in the filtering process. In X- position the deviation for the two concepts are almost exact the same, but in Y-direction the deviation is slightly larger for concept 1.

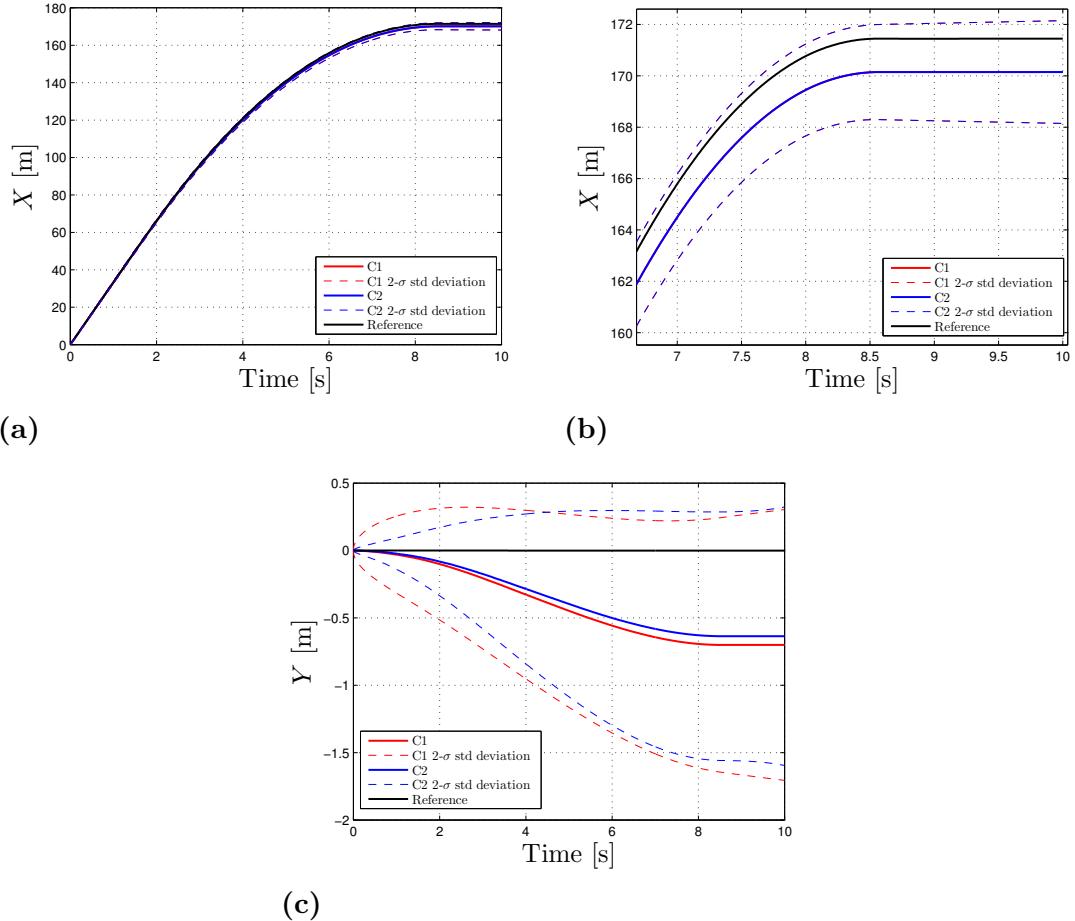


Figure 5.2: Estimated global position states with standard deviations for case 1

The state estimates for velocity and acceleration are visible in Figure 5.3, however estimates of the acceleration states are only visible for concept 2. This since the acceleration states do not exist in concept 1. The reason that the noise seems to disappear for the lateral velocity and acceleration around approximately 8 seconds is since the longitudinal velocity becomes lower than 1 [m/s]. As described when that happens the model describing the lateral motion switches, so that the lateral velocity and acceleration becomes zero. The estimated states looks quite similar for both concepts, however it is noteworthy that the lateral velocity is more noisy for concept 2 than concept 1.

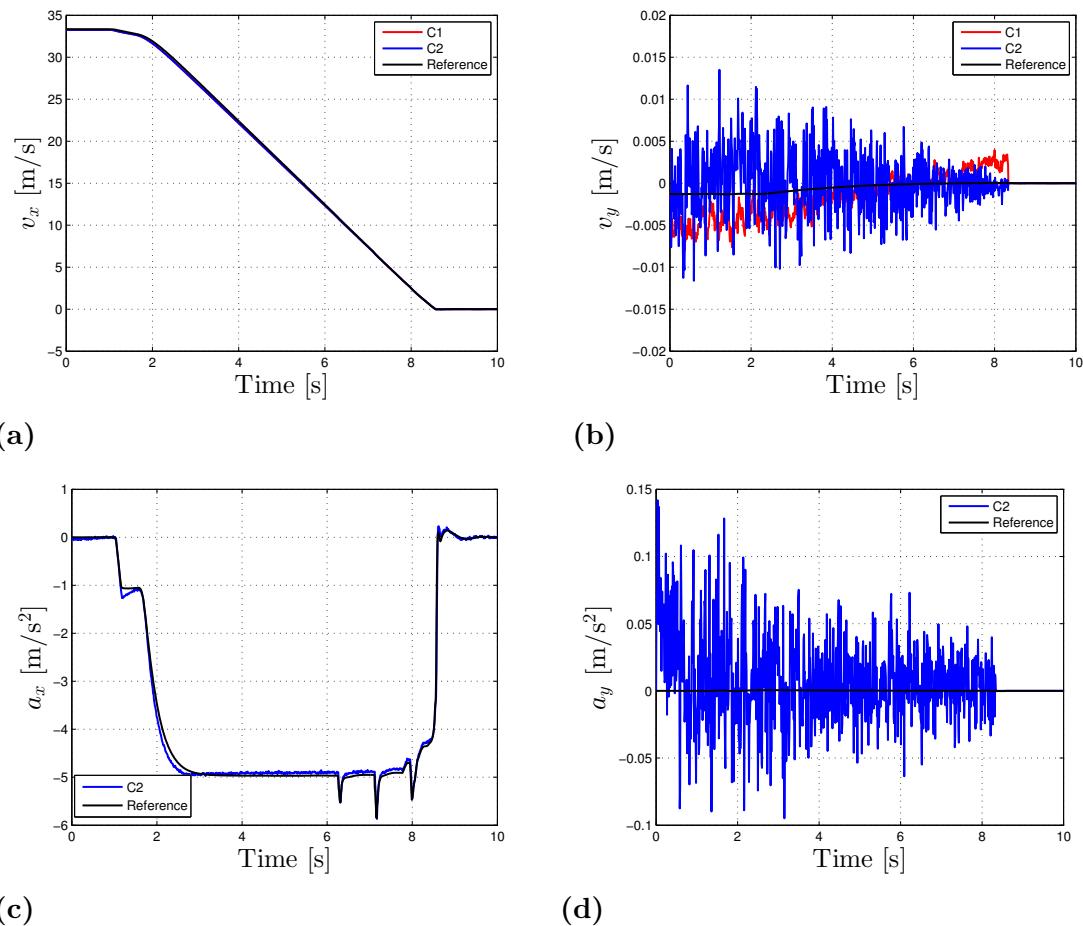


Figure 5.3: Estimated velocity and acceleration states for case 1

5. Simulation Results

The state estimates for the Euler angles (roll, pitch and yaw) and respective rates are visible in Figure 5.4. Also for these states the rates are only visible for concept 2 since concept 1 do not include those states in its state vector. All states seems to follow the reference quite well except for the roll- and yaw states. Those state seems to slowly wander with time, which probably is caused by the integration of the roll- and yaw rate.

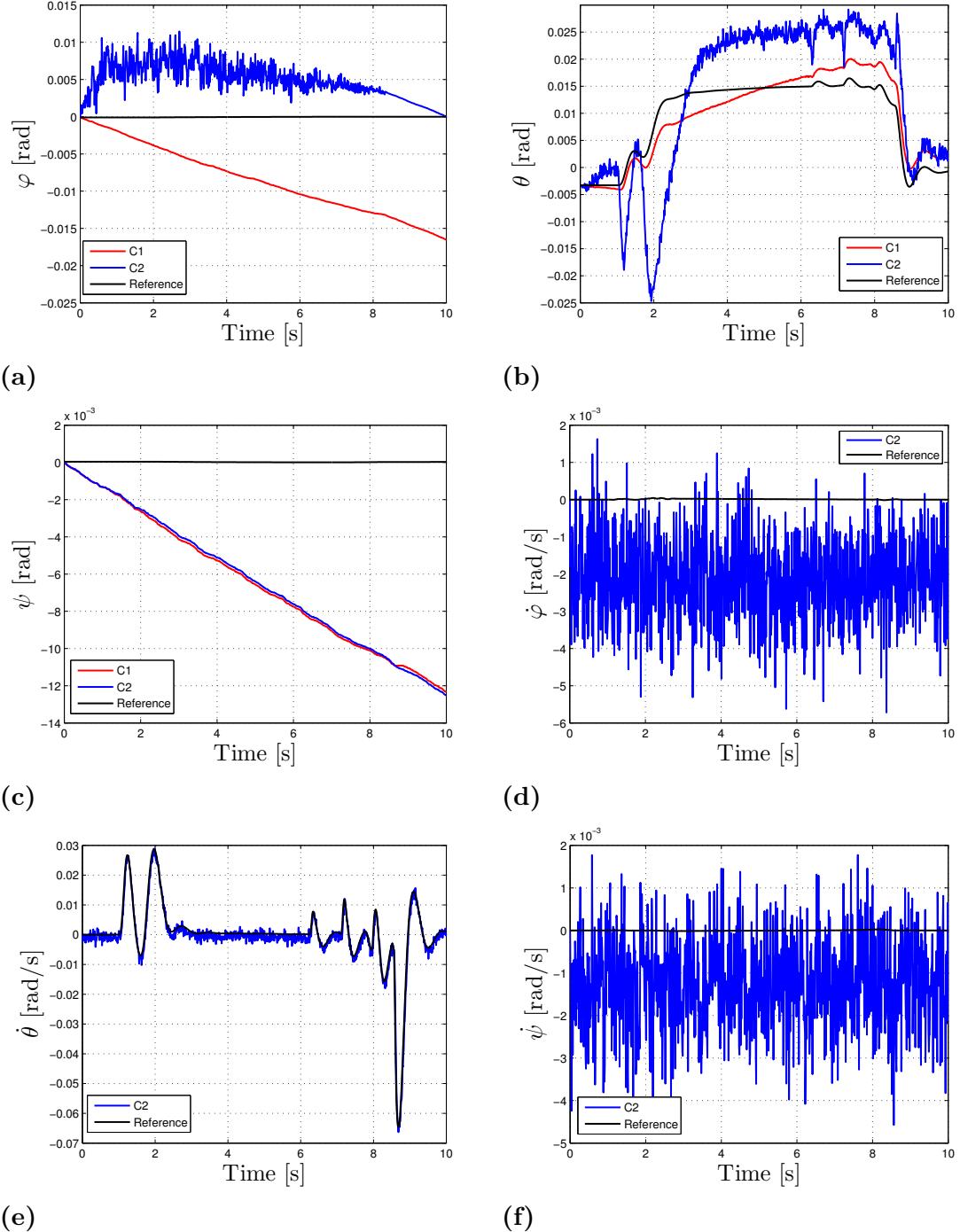


Figure 5.4: Estimated Euler angles and respective rates case 1

To get an idea on how the estimation errors are changing over time the errors ($|e_{lon}|$, $|e_{lat}|$ and $|e_\psi|$) are calculated for every time instance and the results are visible in Figure 5.5. The errors are quite similar for both concepts, however in lateral direction concept 2 has a slightly better estimation.

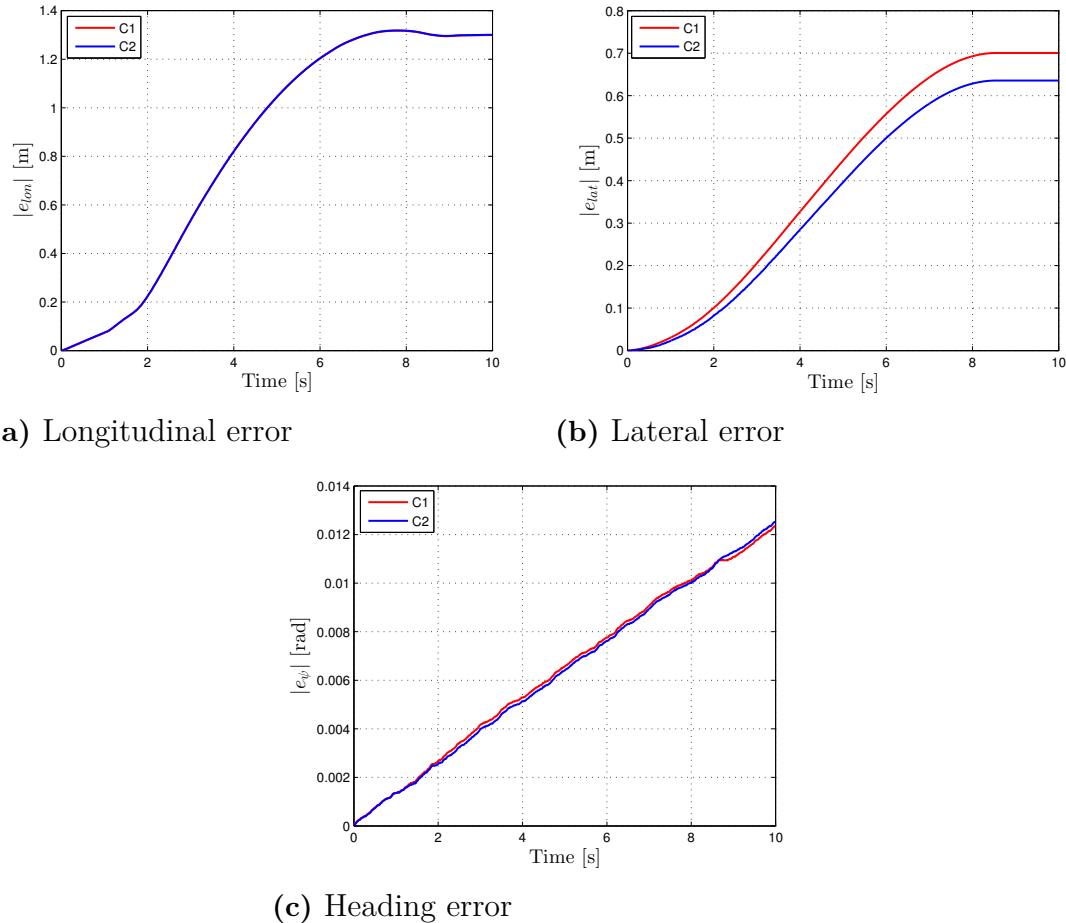


Figure 5.5: The errors of e_{lon} , e_{lat} and e_ψ for case 1

5. Simulation Results

To test the robustness of the filtering process the case was simulated for 1000 runs and filtered for both concepts. This was done during the same circumstances but with an initial velocity of 80 [km/h] as well. The mean and standard deviations for both scenarios are visible in Table 5.1. Worth noticing is though that for the longitudinal error the distribution is not Gaussian which makes the value for the standard deviation a bit misleading.

Table 5.1: Mean and standard deviations when traveling straight forward

	$\mu_{e_{lon}}$ [m]	$\sigma_{e_{lon}}$ [m]	$\mu_{e_{lat}}$ [m]	$\sigma_{e_{lat}}$ [m]	μ_{e_ψ} [rad]	σ_{e_ψ} [rad]
Case 1						
C1: 120 [km/h]	1.3023	0.0042	0.0168	0.8829	$3.3876 \cdot 10^{-4}$	0.0173
C2: 120 [km/h]	1.3034	0.0039	0.0222	0.8309	$3.4299 \cdot 10^{-4}$	0.0171
C1: 80 [km/h]	0.5656	0.0017	0.0106	0.3613	$4.1205 \cdot 10^{-4}$	0.0138
C2: 80 [km/h]	0.5665	0.0017	0.0116	0.3366	$4.1273 \cdot 10^{-4}$	0.0138

Analyzing these results for when traveling in 120 [km/h] it can be said that with 68% certainty (1 standard deviation) the error will be smaller than $e_{lat} < |0.88|$ [m] for concept 1 and smaller than $e_{lat} < |0.83|$ [m] for concept 2. Illustrations of the distribution of the errors are presented in Figure 5.6 where both e_{lat} and e_ψ seems to be almost zero mean with Gaussian distribution for both concepts, which confirm the calculated values in the table. However e_{lon} is not zero mean and does not have a Gaussian distribution as mentioned above. This is probably due to the behaviour of the odometer, since when braking the odometer might lock the tires and therefore will show a slower velocity than the reference. This results in that the estimated longitudinal position is shorter than the reference which leads to a constant positive e_{lon} as visible in the distributions.

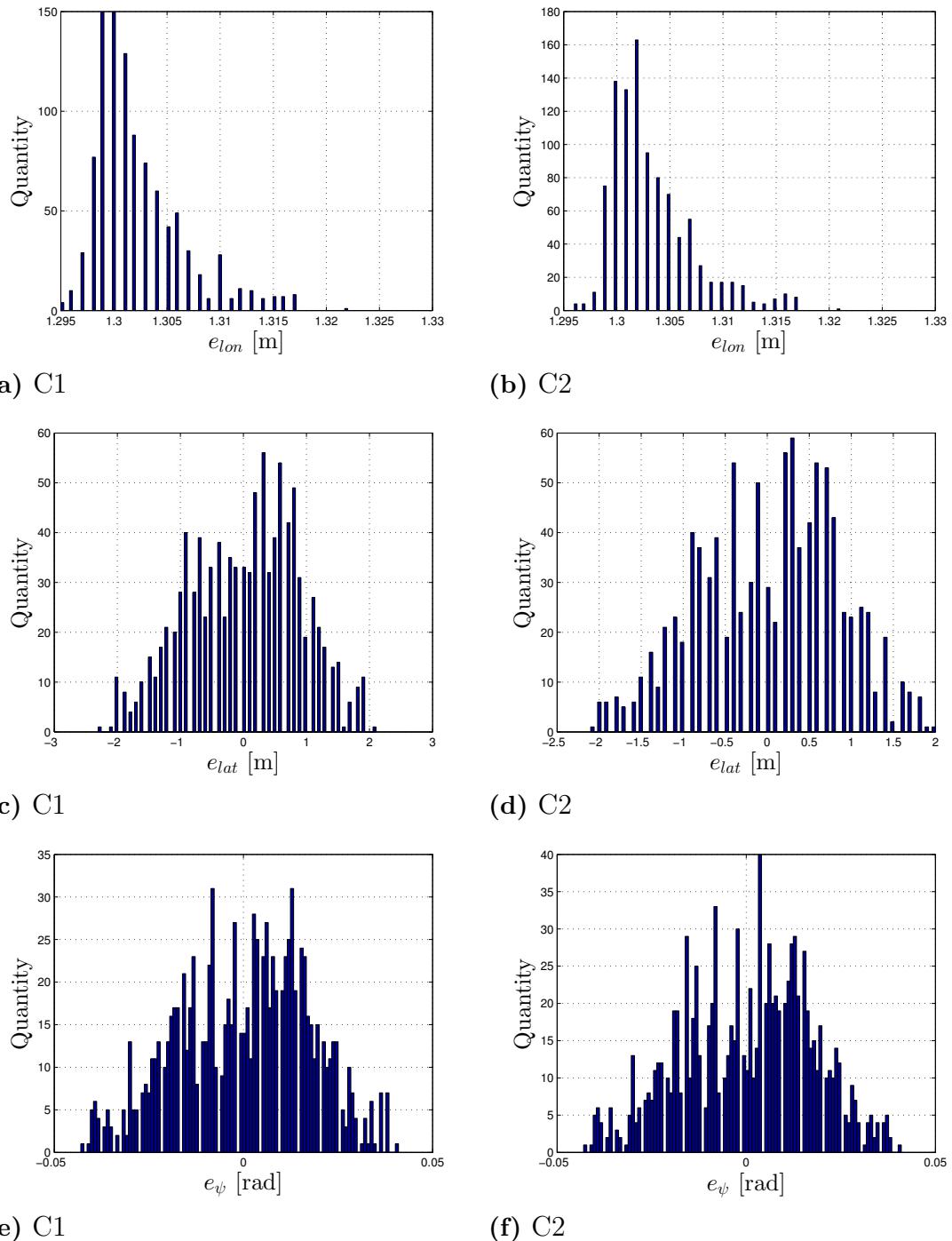


Figure 5.6: The distribution of the errors over 1000 runs for case 1

5. Simulation Results

It is also interesting to see how the number of IMU:s used in the filtering process affects the filtering estimation. Therefore the mean of the squared errors for 30 runs are compared for 1, 2 and 3 IMU:s in Figure 5.7. As expected the result improves as the number of IMU:s increases, this supports the theory about that using multiple IMU:s decreases the sensor variance and impact of biases, see Section 2.1.2. In this case it seems like the number of IMU:s has an higher impact on the estimations for the lateral and heading error than the longitudinal. This is expected since in the longitudinal direction an additional sensor (the odometer) was used.

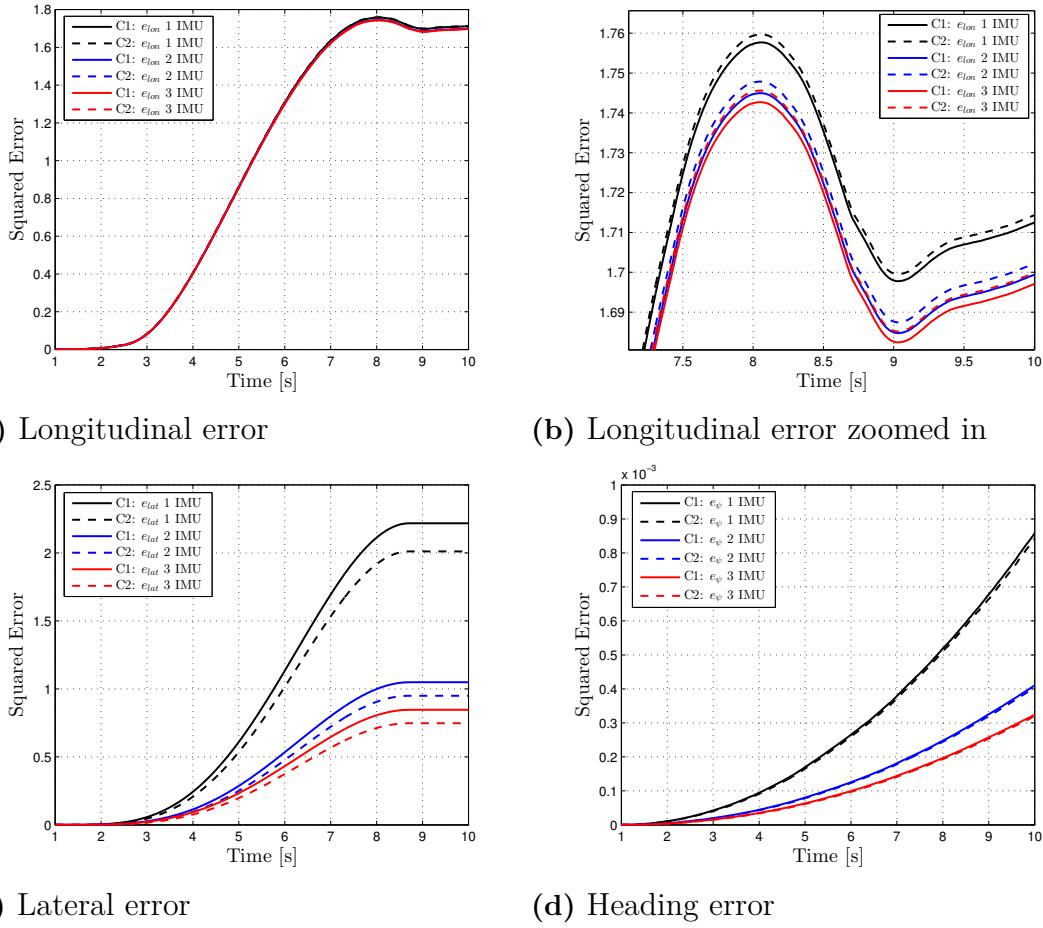


Figure 5.7: Squared error of e_{lon} , e_{lat} and e_ψ using different numbers of IMU:s for case 1

To see how the size of the constant sensor bias affects the position estimates, different constant biases for the IMU:s was tested. The squared error for the final estimates for 30 runs and the mean of the squared error for different biases was compared, see Figure 5.8. The constant bias used for the rest of the test cases are varied between [-0.1 0.1] [m/s²] for the accelerometer and [-0.005 0.005] [rad/s] for the gyroscope. As seen the squared error mean of the lateral and heading errors are reduced by more than half if the constant bias interval is reduced by half. However the constant bias does not seem to be a crucial part for the longitudinal error.

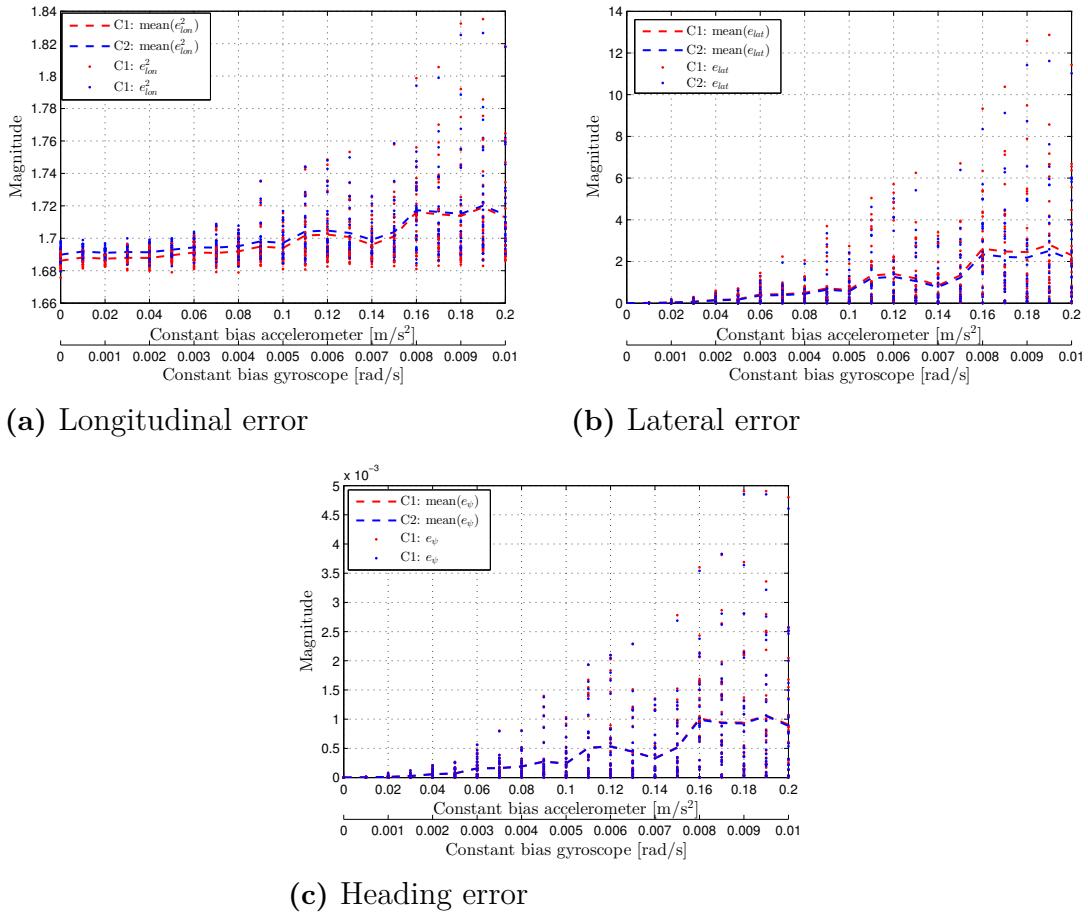


Figure 5.8: Squared error illustrated against growing constant bias of the IMU:s for case 1

5. Simulation Results

It was also tested how the initial velocity affects the result. Case 1 was simulated from 10 [km/h] up to 120 [km/h] with an interval of 10 [km/h] with 30 runs for each velocity. The mean of the absolute value of the errors is presented in table 5.2.

Table 5.2: Filtering result from simulated data with absolute values of the errors for case 1

	$ e_{lon} $ [m]	$ e_{lat} $ [m]	$ e_\psi $ [rad]
Case 1			
C1: 120 [km/h]	1.3027	0.9198	0.0180
C2: 120 [km/h]	1.3038	0.8645	0.0179
C1: 110 [km/h]	1.0918	0.7941	0.0172
C2: 110 [km/h]	1.0930	0.7540	0.0171
C1: 100 [km/h]	0.8944	0.5408	0.0126
C2: 100 [km/h]	0.8955	0.5025	0.0126
C1: 90 [km/h]	0.7212	0.4154	0.0124
C2: 90 [km/h]	0.7222	0.3868	0.0123
C1: 80 [km/h]	0.5659	0.3694	0.0140
C2: 80 [km/h]	0.5669	0.3423	0.0140
C1: 70 [km/h]	0.4185	0.2831	0.0134
C2: 70 [km/h]	0.4193	0.2574	0.0133
C1: 60 [km/h]	0.2999	0.1827	0.0110
C2: 60 [km/h]	0.3007	0.1660	0.0109
C1: 50 [km/h]	0.2010	0.1458	0.0113
C2: 50 [km/h]	0.2017	0.1302	0.0112
C1: 40 [km/h]	0.1210	0.0990	0.0104
C2: 40 [km/h]	0.1215	0.0866	0.0104
C1: 30 [km/h]	0.0610	0.0644	0.0111
C2: 30 [km/h]	0.0614	0.0553	0.0110
C1: 20 [km/h]	0.0219	0.0349	0.0087
C2: 20 [km/h]	0.0222	0.0286	0.0086

The result for varying velocities are also presented as the squared error of the final estimate and the MSE for each iteration in Figure 5.9. The mean of the squared error and MSE for each velocity is also illustrated to show the trend. As seen the initial velocity has a crucial part on the result, especially for e_{lon} .

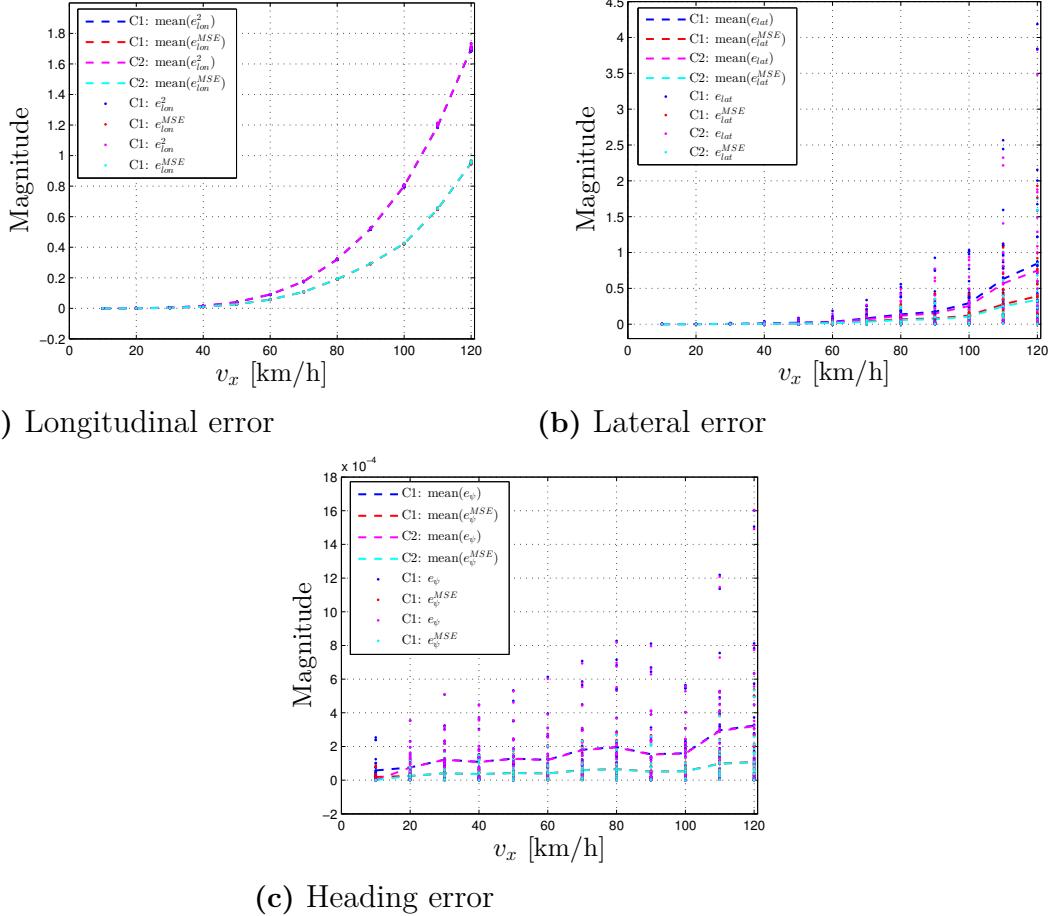
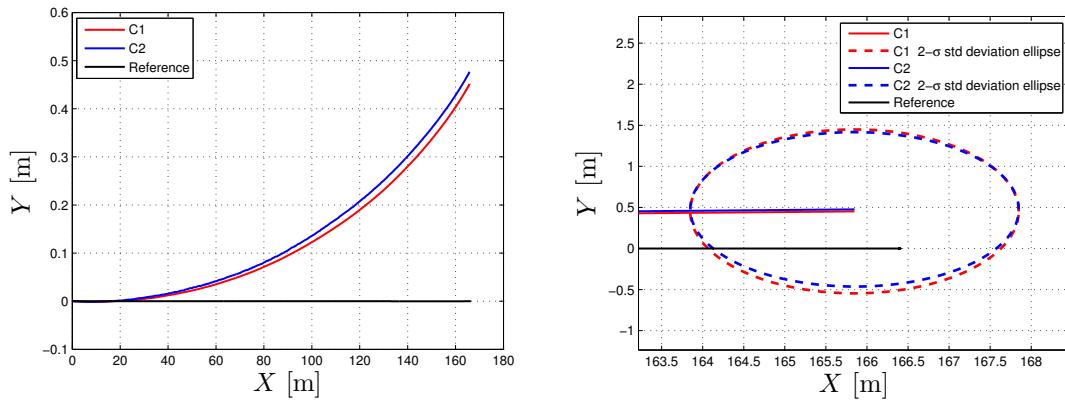


Figure 5.9: Squared error illustrated against growing initial velocity for case 1

An extra load of 300 [kg] was added in CarMaker and the tires changed without compensating for it in the filtering process to test how the result was affected. This was simulated for 30 runs and the errors were compared with errors when the model parameters were fully known. The difference of the means of these 30 runs was approximately a couple of millimeters for e_{lon} and e_{lat} and almost zero for e_ψ . This test was done for case 2 and 3 as well, which gave similar result.

5.2 Case 2

The result from the filtering process for case 2, (see Section 1.4.1), are presented below. When comparing the two concepts the filtering was based on the same data set. In Figure 5.10 the global position estimates are illustrated against each other for both concepts. The 2σ standard deviations for the final estimate of the X- and Y- position are also illustrated as ellipses for both concepts.



(a) The estimated X- and Y- positions illustrated against each other (b) 2 σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 5.10: The estimated X- and Y- positions for case 2

In Figure 5.11 the state estimates for the global position are illustrated separately. This time the estimated position in Y-direction is slightly closer to the reference than in case 1. The 2σ standard deviations for the global position estimates are also presented in the figures.

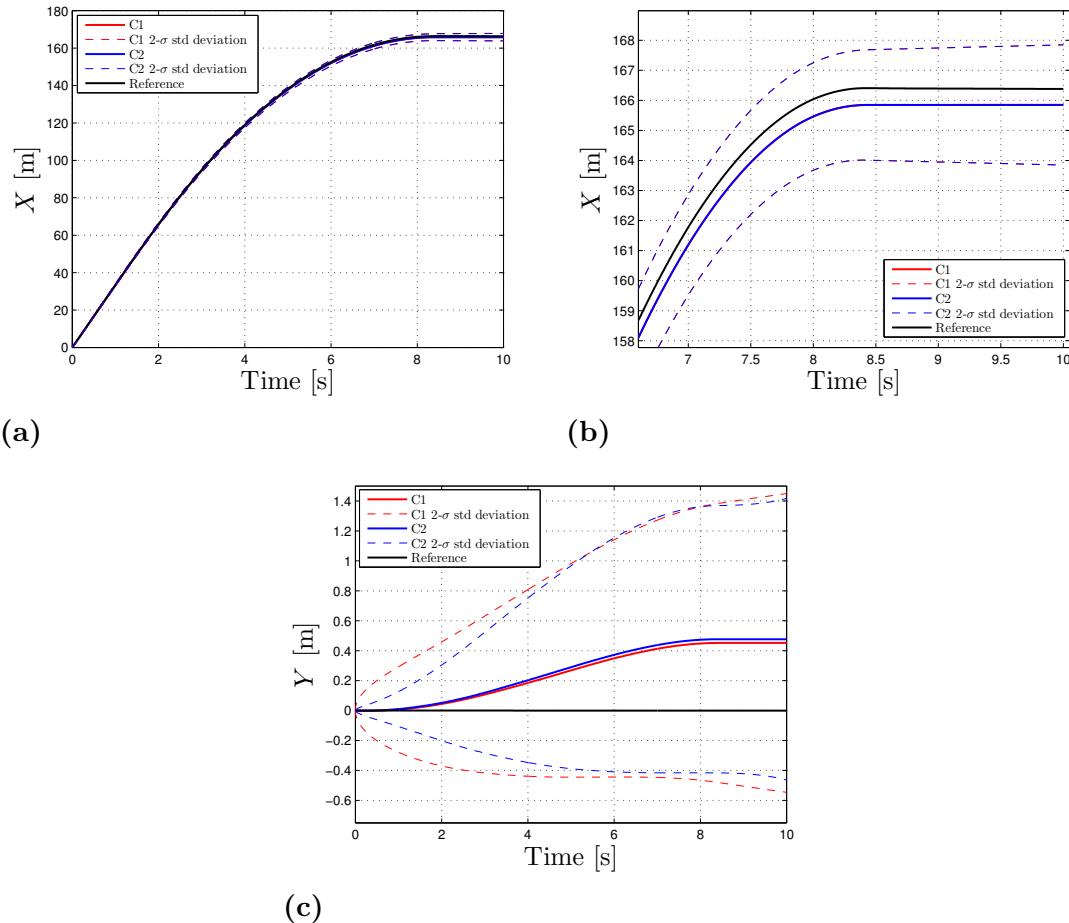


Figure 5.11: Estimated global position states with standard deviations for case 2

5. Simulation Results

In Figure 5.12 the estimates for velocity and acceleration states are illustrated. The noise of the lateral velocity is, (as in case 1), bigger for concept 2 than concept 1.

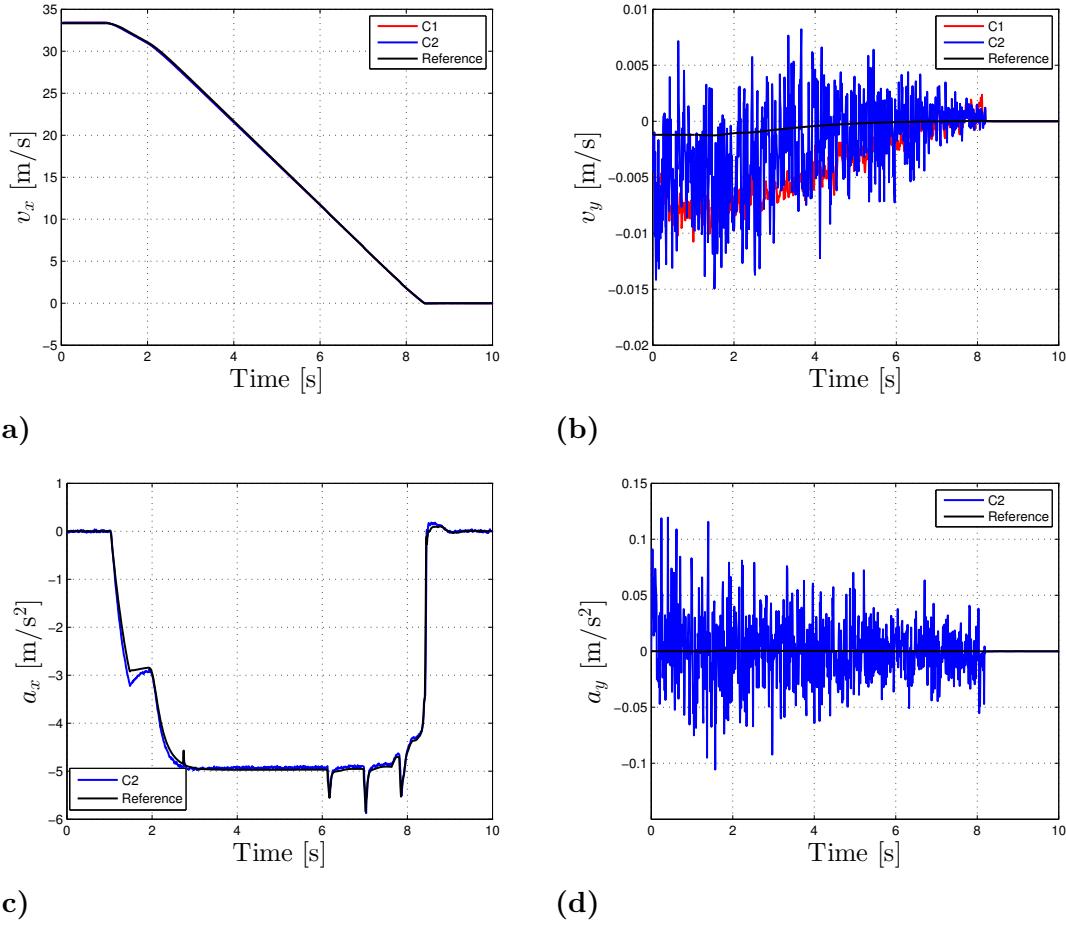


Figure 5.12: Estimated velocity and acceleration states for case 2

In Figure 5.13 the Euler angles and respective rates for both concepts are illustrated. As for case 1 the roll- and yaw states wander with time which probably also in this case is due to the integration of roll- and yaw rate since they almost have the appearance of white noise.

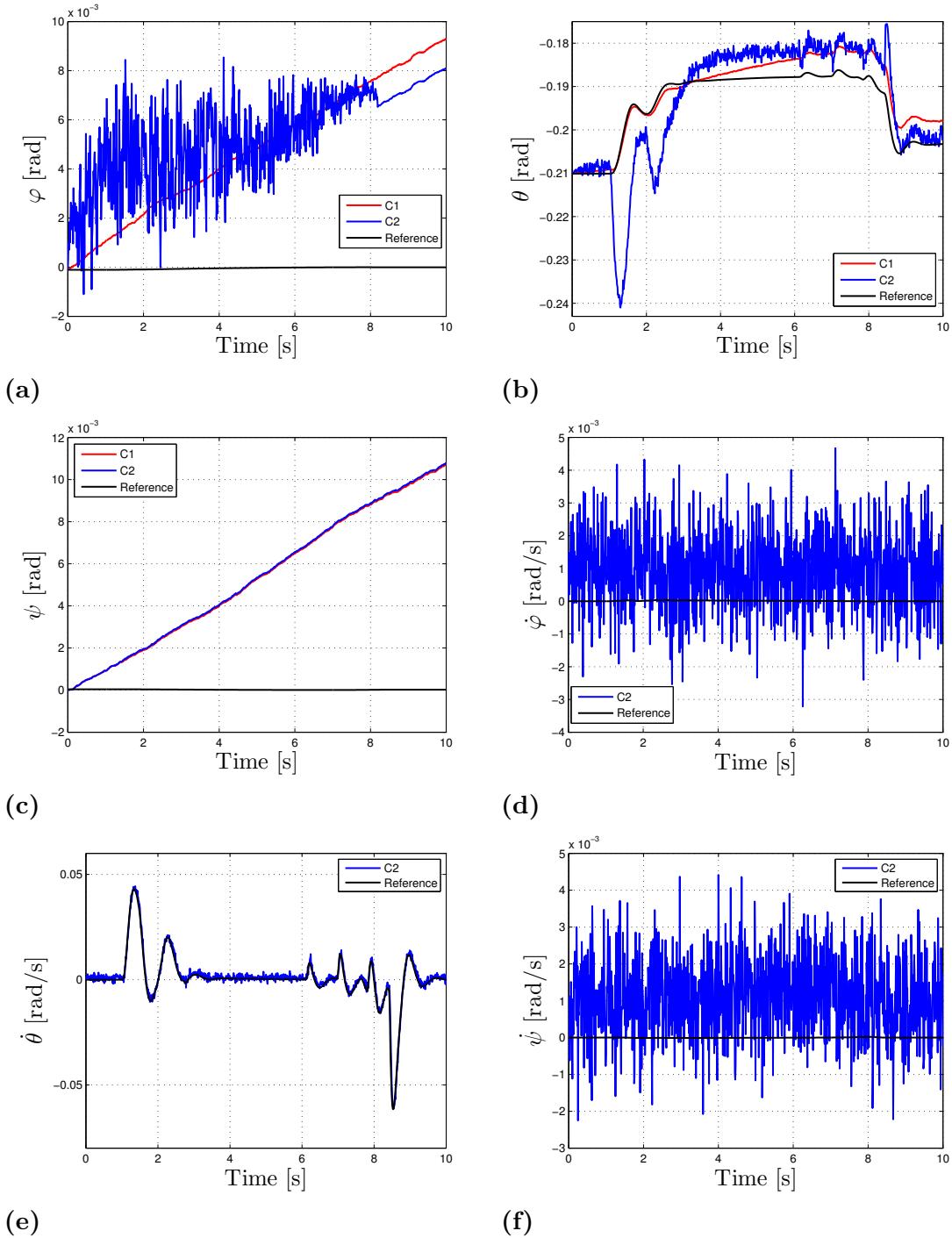


Figure 5.13: Estimated Euler angles and respective rates for case 2

5. Simulation Results

The longitudinal, lateral and heading error are illustrated in Figure 5.14. The longitudinal and heading error for both concepts are also in this case very similar. However this time the lateral error for concept 1 is smaller than the lateral error for concept 2.

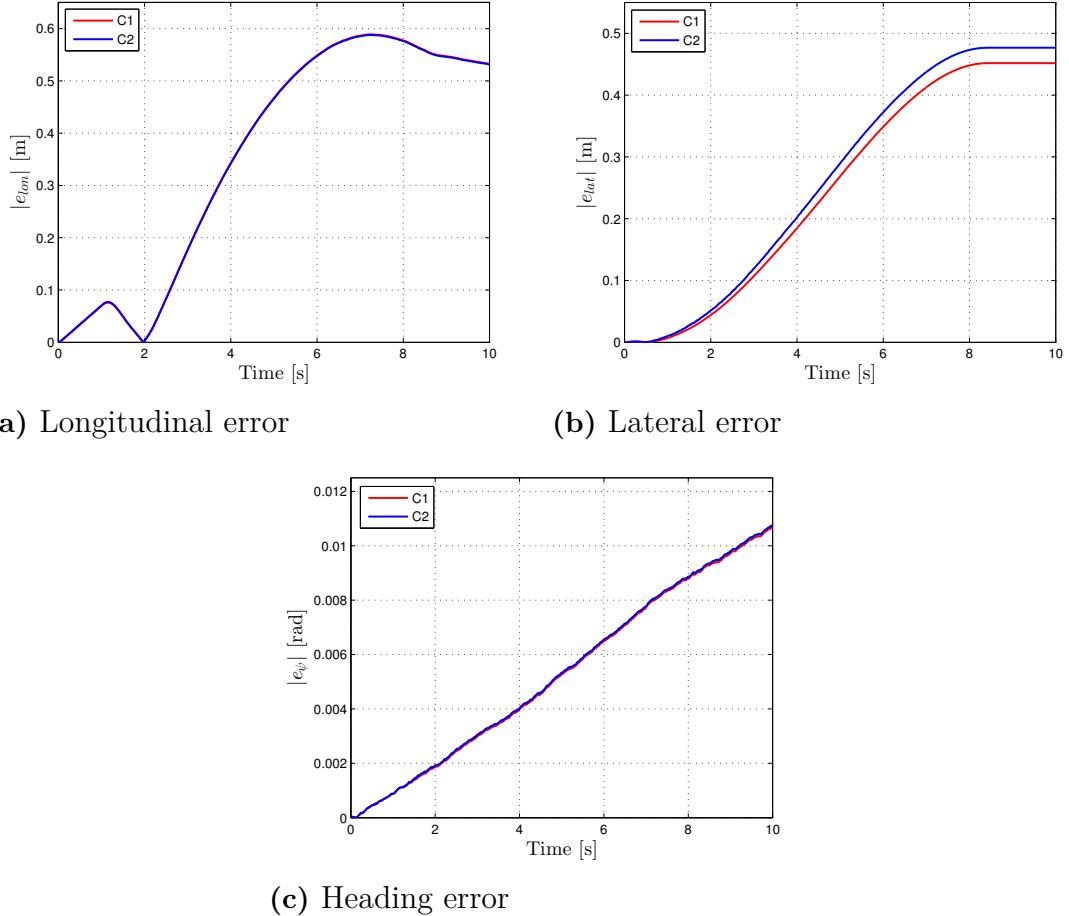


Figure 5.14: The errors of e_{lon} , e_{lat} and e_ψ for case 2

The case was simulated for 1000 runs to test the robustness of the filtering process. The distribution of the final errors for both concepts are illustrated in Appendix H and the means and standard deviations are visible in Table 5.3. The distribution of e_{lat} and e_ψ is zero mean but not e_{lon} , (the same as in case 1). However the mean for e_{lon} in case 2 is smaller than in case 1, approximately 0.53 [m] compared with approximately 1.3 [m].

Table 5.3: Mean and standard deviation for case 2

	$\mu_{e_{lon}}$ [m]	$\sigma_{e_{lon}}$ [m]	$\mu_{e_{lat}}$ [m]	$\sigma_{e_{lat}}$ [m]	μ_{e_ψ} [rad]	σ_{e_ψ} [rad]
Case 2						
C1: 120 [km/h]	0.5356	0.0041	0.0318	0.8670	$6.4692 \cdot 10^{-4}$	0.0178
C2: 120 [km/h]	0.5340	0.0037	0.0361	0.8158	$6.4737 \cdot 10^{-4}$	0.0176

To test how the number of IMU:s affects the filter estimation, once again 1, 2 and 3 IMU:s are tested and the results as an average of 30 runs are visible in Figure 5.15. In this case it is also very clear that as the number of IMU:s increases the errors of the estimates decrease.

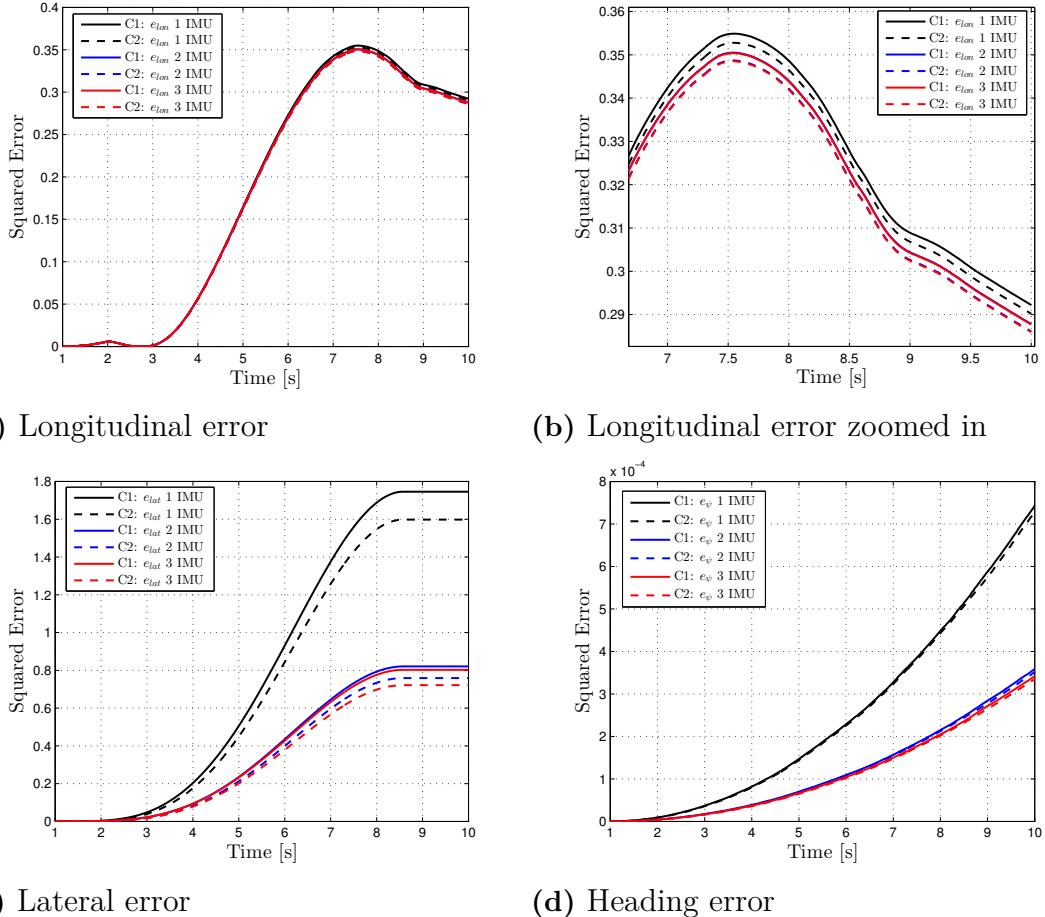
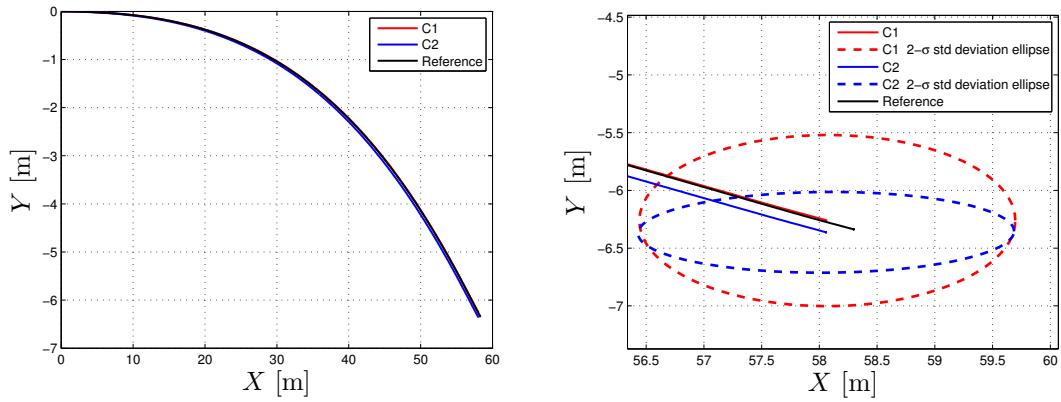


Figure 5.15: Squared error of e_{lon} , e_{lat} and e_ψ using different numbers of IMU:s for case 2

Varying constant biases was tested for this case as well and the visible trends was the same as for case 1. Filter estimates from varying initial velocity was not produced for this case since the resulting trends was almost the same as for case 1. The only difference was that all errors was a bit lower, just as visible throughout this whole case.

5.3 Case 3

The results of the filtering process for case 3, (see Section 1.4.1), are presented below. Just as for the other cases the two filtering concepts are based on the same data set. The global position estimates and the 2σ standard deviations of the final estimates are illustrated in Figure 5.16. In this case its a larger difference between the deviations in Y- direction for both concepts compared to previous cases. This is probably due to that the vehicle moves more in Y-direction and therefore differences in Y-direction are more visible.



- (a) The estimated X- and Y- positions illustrated against each other (b) 2σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 5.16: The estimated X- and Y- positions for case 3

In Figure 5.17 the estimated global positions are illustrated separately. The corresponding 2σ standard deviations for each state and concept are also visible. In X-direction the two concepts are very similar and it is therefore hard to distinguish the estimates for concept 1 in the figure. However the same trend as in Figure 5.16 is visible, that the standard deviation for concept 1 is slightly larger than for concept 2.

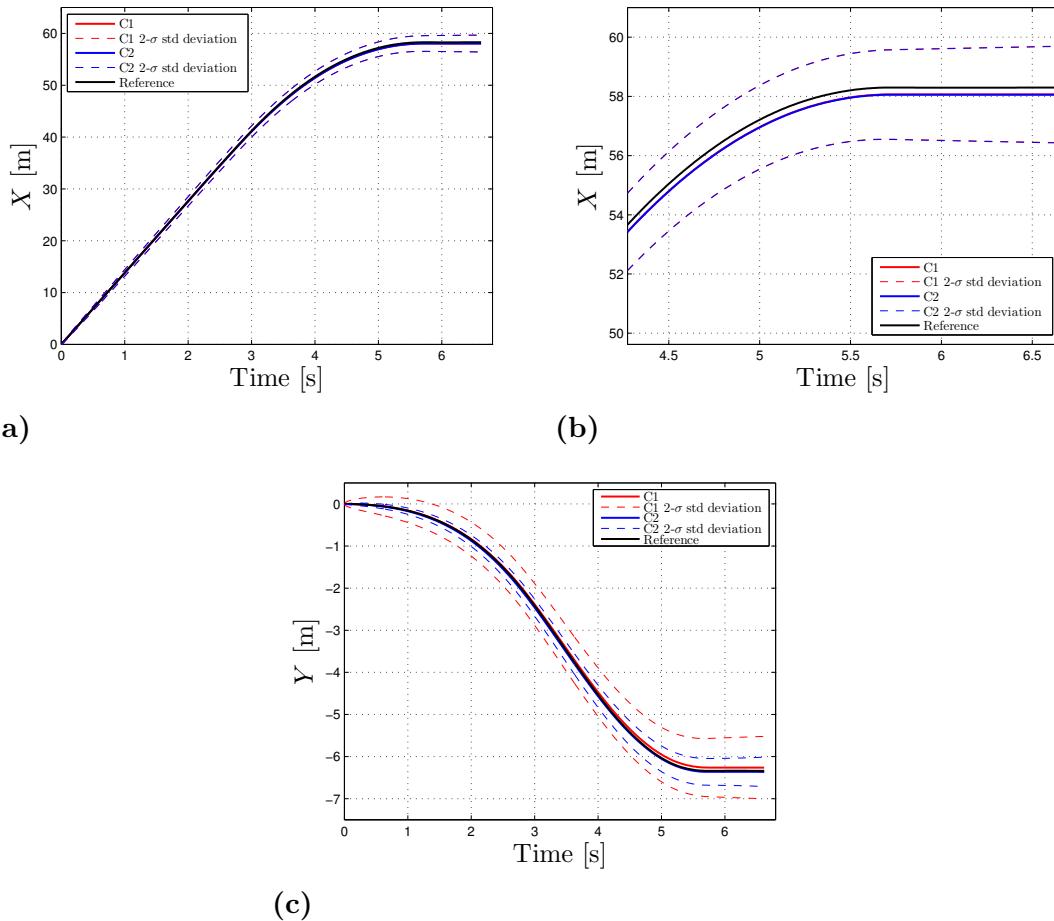


Figure 5.17: Estimated global position states with standard deviations for case 3

5. Simulation Results

In Figure 5.18 the estimated states for the vehicles velocity and acceleration are illustrated. The lateral velocity for concept 2 is phase shifted in the curve compared with the reference, this is a known problem with the steering angle based bicycle model that is used for concept 2 and this is further discussed in Section 3.2.1.1.

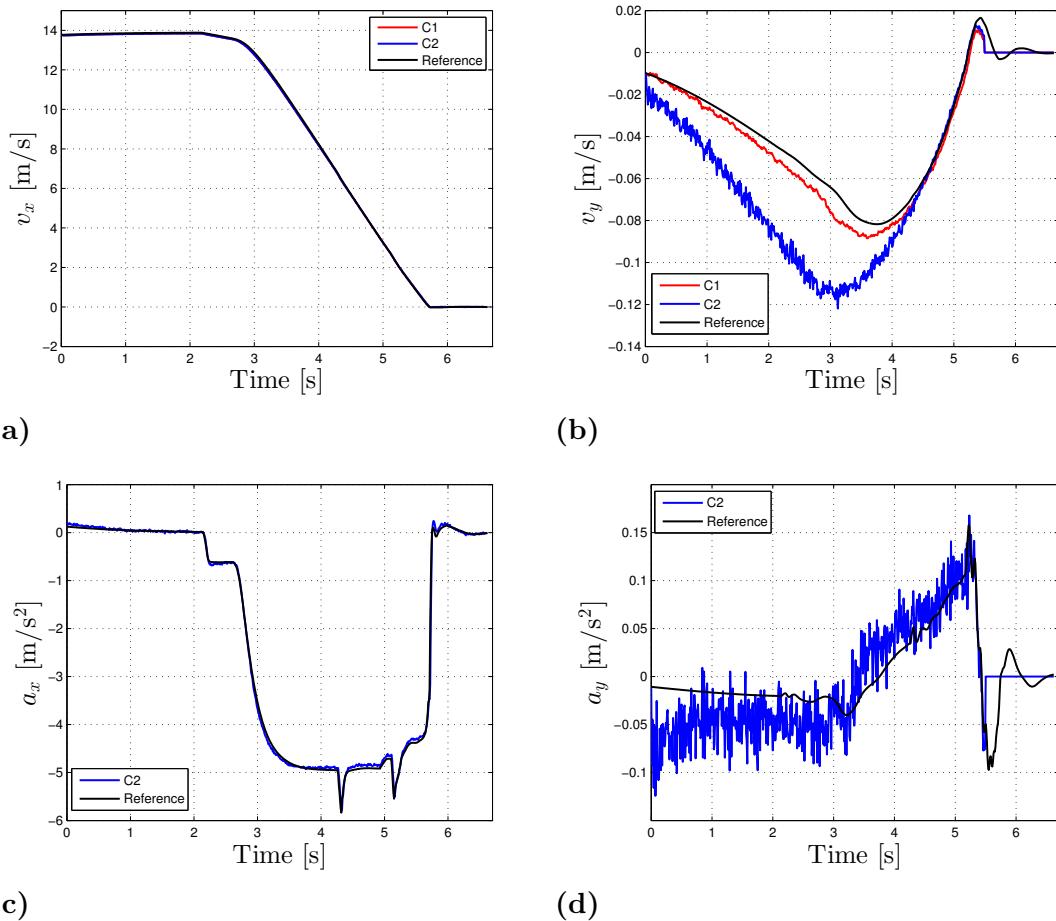


Figure 5.18: Estimated velocity and acceleration states for case 3

The Euler angles and respective rates are illustrated in Figure 5.19. In this case all the state estimates seems to follow the reference quite well except for the roll state that seems to slowly wander with time for concept 1. For both the roll- and pitch states it is visible that the noise levels are much higher for concept 2 than for concept 1, which probably as in the other cases is due to the integration of roll- and pitch rate.

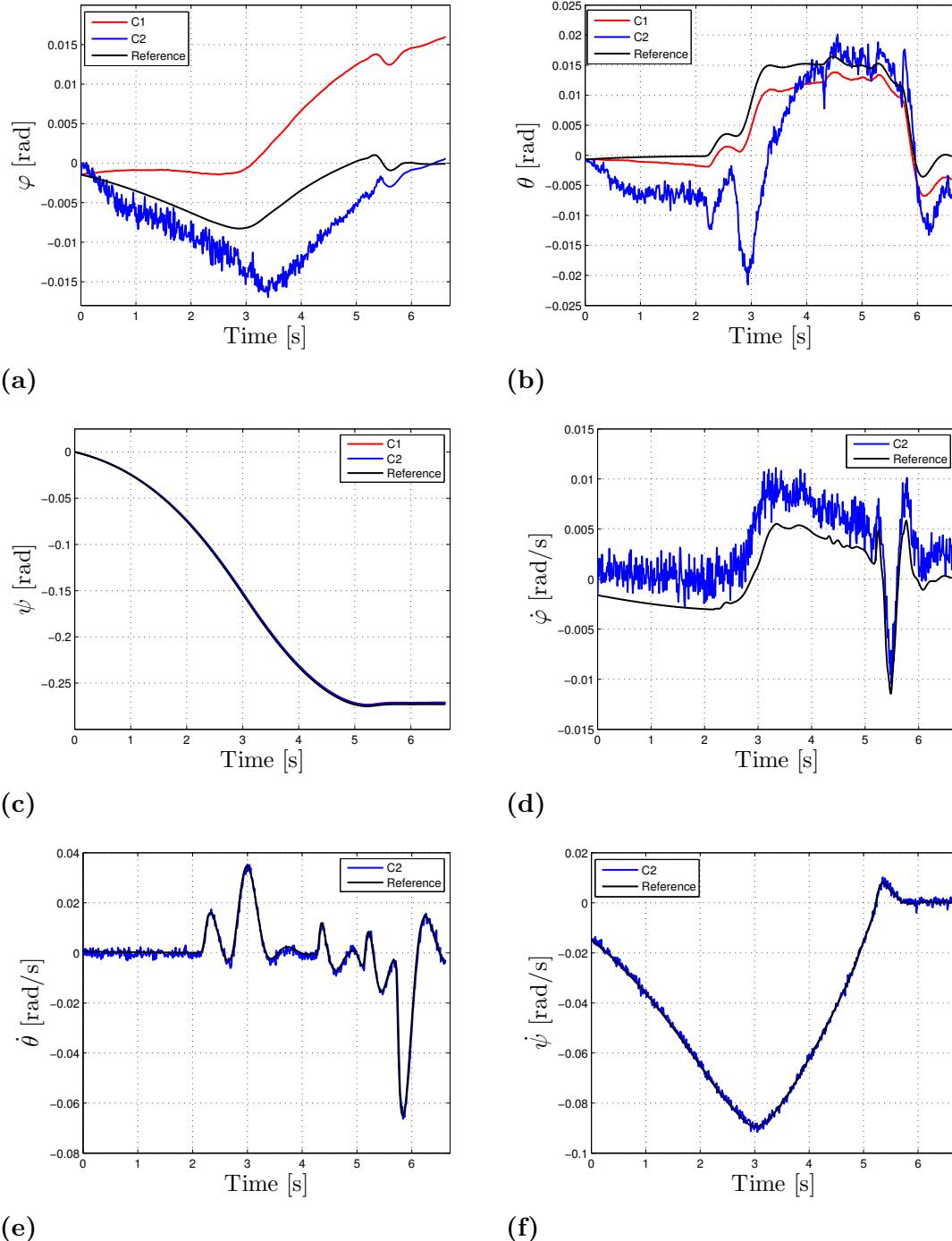


Figure 5.19: Estimated Euler angles and respective rates for case 3

5. Simulation Results

The absolute value of the estimation errors for case 3 are illustrated in Figure 5.20. In this case concept 2 is slightly better in performance for the longitudinal position but concept 1 has a better performance for the lateral position. For the yaw states the errors are very small and quite similar.

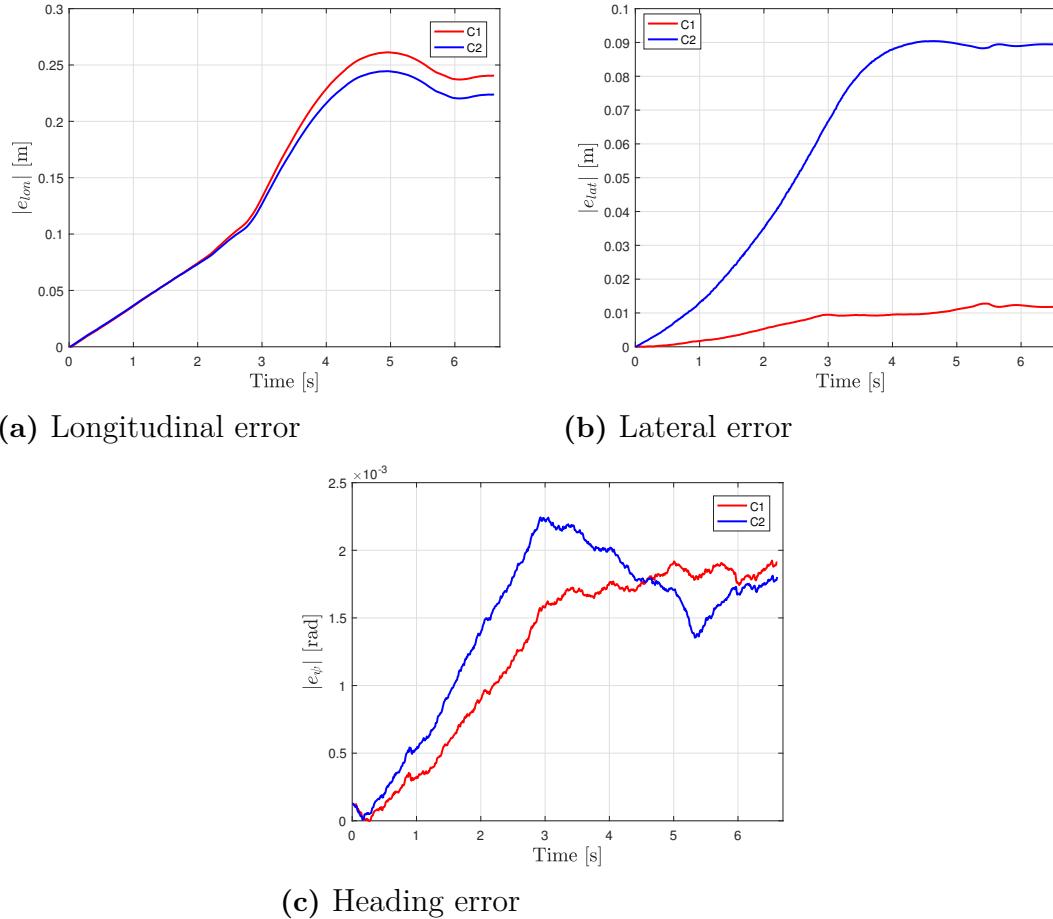


Figure 5.20: The errors of e_{lon} , e_{lat} and e_ψ for case 3

To test the robustness of the system case 3 was simulated and filtered for 1000 runs. The distribution of the errors e_{lon} , e_{lat} and e_ψ for both concepts are illustrated in Appendix H and the means and standard deviations are visible in Table 5.4. Also in this case the longitudinal errors are not zero mean, the error is however significantly smaller than for both case 1 and 2 which is probably due to the lower longitudinal velocity.

Table 5.4: Mean and standard deviations for case 3

	$\mu_{e_{lon}}$ [m]	$\sigma_{e_{lon}}$ [m]	$\mu_{e_{lat}}$ [m]	$\sigma_{e_{lat}}$ [m]	μ_{e_ψ} [rad]	σ_{e_ψ} [rad]
Case 3						
C1: 50 [km/h]	0.2336	0.0264	0.0547	0.2327	$4.0250 \cdot 10^{-4}$	0.0113
C2: 50 [km/h]	0.2198	0.0237	0.1355	0.2161	$5.9756 \cdot 10^{-4}$	0.0112

Analyzing these results it can be said that with 68% certainty the error will be smaller than $e_{lat} < |0.23|$ [m] for concept 1 and smaller than $e_{lat} < |0.21|$ [m] for concept 2.

Different numbers of IMU:s was tested also for this case, the resulting squared errors are illustrated in Figure 5.21. The largest impact for the different number of IMU:s seems to be for the lateral and heading errors. However the numbers of IMU:s seems to have a larger impact on the longitudinal squared error for this case compared to the other two cases.

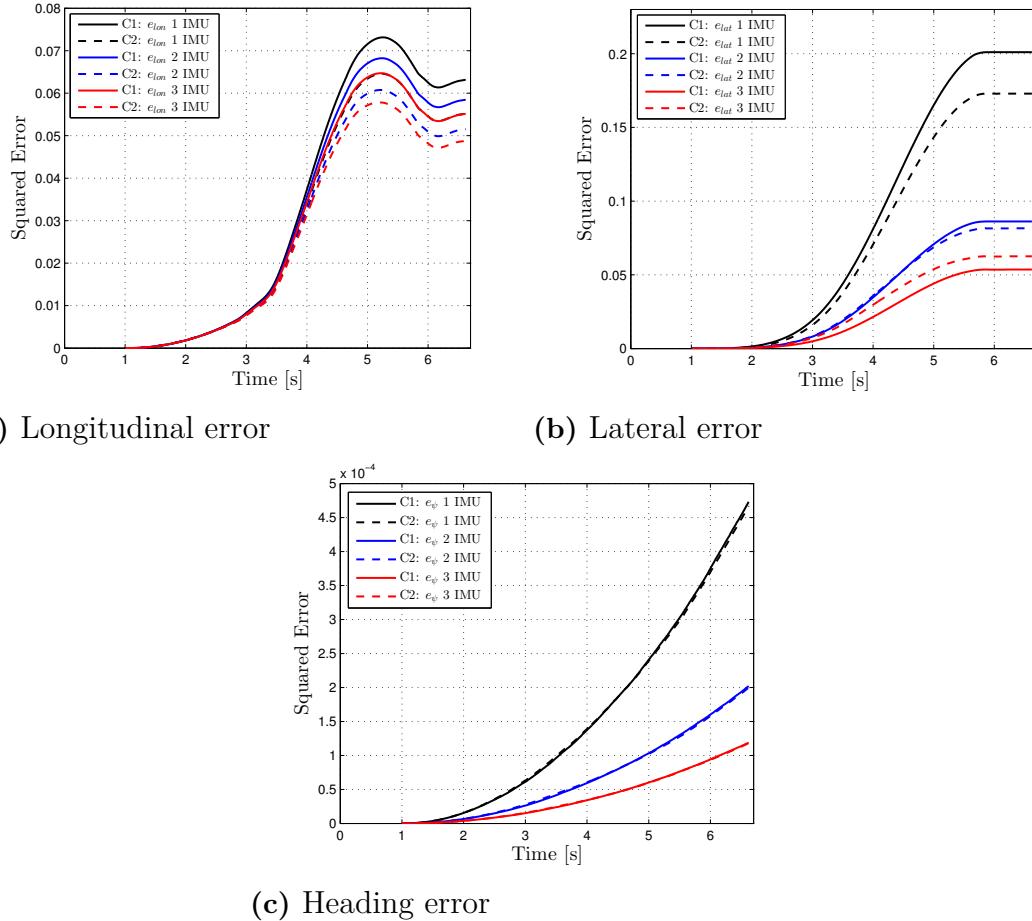


Figure 5.21: Squared error of e_{lon} , e_{lat} and e_ϕ using different numbers of IMU:s for case 3

5. Simulation Results

In Figure 5.22 the impact of constant biases for the IMU:s was tested. All three IMU:s was used and the interval that the constant bias could be was increased step wise. As seen when the biases increases the distribution of the errors increases which also increases the uncertainty of the result. The longitudinal error is not affected as much as the lateral and heading error, this is probably due to the extra sensor used for this state, the odometer.

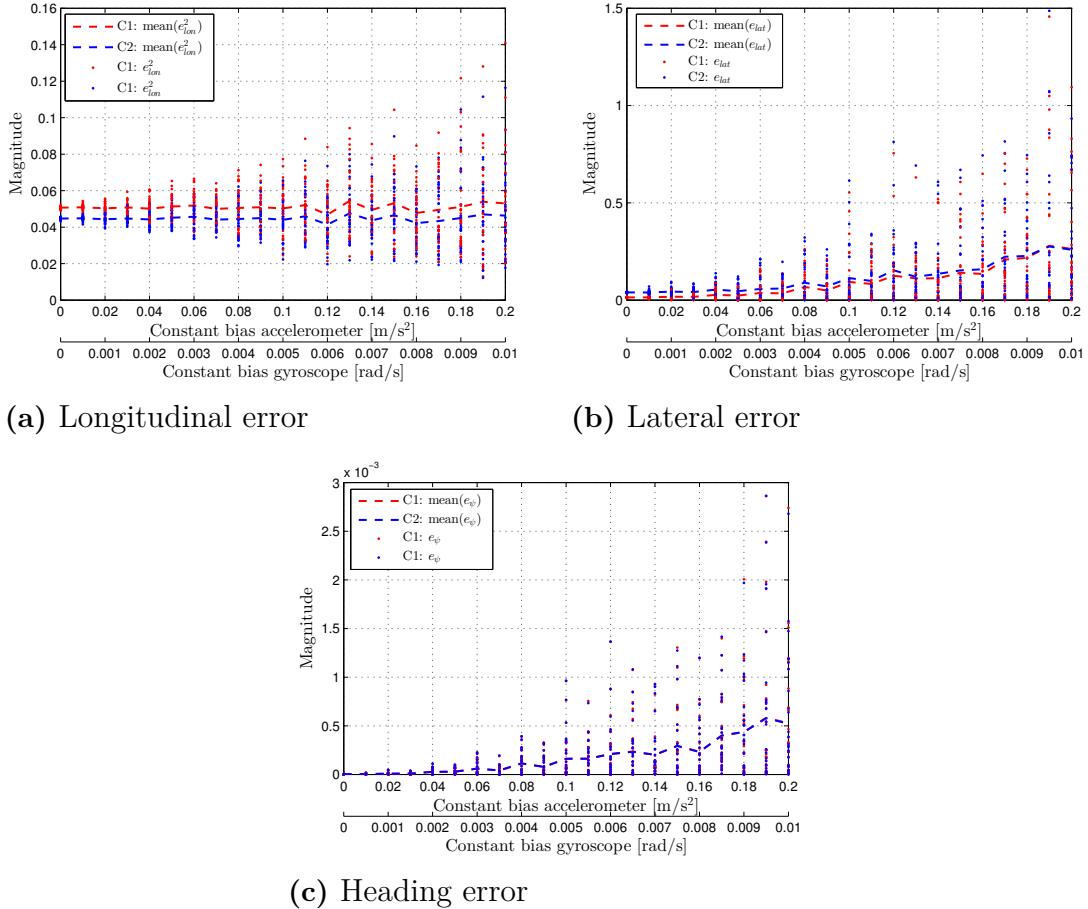


Figure 5.22: Squared error illustrated against growing constant bias of the IMU:s for case 3

It was also for case 3 tested how the initial velocity affects the filtering result. The initial velocity started at 10 [km/h] and was increased with an interval of 10 [km/h] until 50 [km/h]. Each velocity was simulated for 30 runs for both concepts. The results are presented in Table 5.5.

Table 5.5: Filtering result from simulated data with absolute value of the errors for case 3

	$ e_{lon} $ [m]	$ e_{lat} $ [m]	$ e_\psi $ [rad]
Case 3			
C1: 50 [km/h]	0.2348	0.2315	0.0109
C2: 50 [km/h]	0.2209	0.2501	0.0109
C1: 40 [km/h]	0.0893	0.0416	0.0082
C2: 40 [km/h]	0.0897	0.0355	0.0082
C1: 30 [km/h]	0.0355	0.0307	0.0077
C2: 30 [km/h]	0.0360	0.0246	0.0077
C1: 20 [km/h]	0.0051	0.0137	0.0066
C2: 20 [km/h]	0.0054	0.0098	0.0065

5. Simulation Results

The result for varying velocities are also presented as the final squared errors and MSE illustrated for each iteration in Figure 5.23. To see the trend the mean of the squared errors and MSE for each velocity are also visible in Figure 5.23. The same conclusion as in case 1 could be drawn here, that the initial velocity is crucial to the filtering estimates and the initial velocity mostly affects the longitudinal error. In this case the errors grows drastically between 40 [km/h] and 50 [km/h], so by lower the velocity by 10 [km/h] the final longitudinal error will probably be reduced by approximately 80 %.

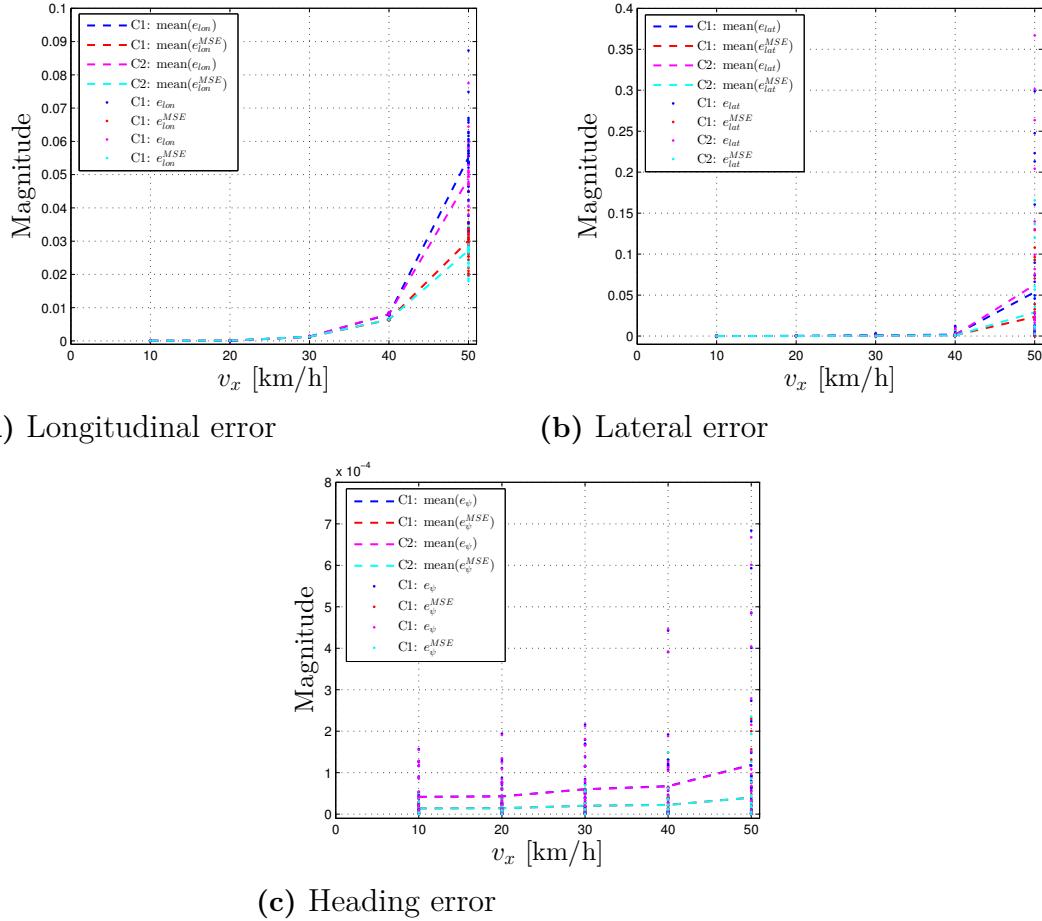


Figure 5.23: Squared error illustrated against growing initial velocity for case 3

5.4 Summary of the Simulation Results

To summarize the results from the simulated sensors using CarMaker the longitudinal and lateral errors are varying with the initial velocity of the vehicle, the number of IMU:s used and the size of the IMU noise. The curvature of the road do not seem to affect the results as long as the lateral acceleration is kept within the specified limit in Section 1.4. The absolute value of the longitudinal and lateral errors are as a mean of 30 runs when driving initially in 120 [km/h] 1.3 and 0.9 [m] respectively.

Decreasing the initial velocity to 80 [km/h] gives absolute errors of 0.6 and 0.4 [m] and if decreasing even more to an initial velocity of 40 [km/h] the errors reduced to 0.1 and 0.1 [m]. If performing infinite runs the mean of the lateral error will approach zero, however the standard deviation is quite big and is also dependent on the initial velocity.

5. Simulation Results

6

Experimental Tests and Results

To further evaluate the filter performance of the two concepts data is gathered at Volvo Cars proving ground in Hällerid using a Volvo XC90. In this chapter the post treatment of the gathered data are presented together with the result from the filtering estimation process.

6.1 Data Gathering

An additional measurement system *RT3000 Inertial and GPS Navigation System* from Oxford Technical Solutions Limited is used for reference generation, henceforth referred as RT3000. The RT3000 is equipped with a 6-degree of freedom IMU and a GPS with high precision. The RT3000 is mounted in the vehicle with a rack that holds the instrument in place. An antenna is placed at the roof of the vehicle 1 m displaced in x-direction, which is compensated for in the settings of the instrument. The RT3000 has an embedded Kalman filter and therefore filtered velocities, accelerations and Euler angles can be obtained and used as references.

From the vehicle the raw data from the IMU, odometer and pinion angle sensor are gathered. This sensor data is not treated in any way before the filtering process and are therefore not bias compensated. The estimated wheel radius and estimated mass of the vehicle are also logged during the data gathering. The vehicle used has only one IMU with 5-degrees of freedom, therefore the IMU in the RT3000 is used to generate the 6th-degree.

6.1.1 Post Treatment of Sensor Data

Some post treatment of the gathered data is necessary to retrieve the desired references. The GPS signal from the RT3000 is given in decimal degrees (describing the longitude and latitude coordinates), however the references are desired as X- Y-coordinates in the previously defined global coordinate system. The GPS signal is

6. Experimental Tests and Results

then translated to X- Y- coordinates by the following expression

$$X = 2\pi r \cos(\text{mean}(la)) \frac{\pi}{180} \frac{lo}{360} \quad (6.1)$$

$$Y = 2\pi r \frac{la}{360},$$

where lo and la are the decimal degrees given from the GPS and r is the radius of the earth. Equation 6.1 projects a tangent plane on the earth's surface, this means that this expressions are only valid for short distances. Since the safe stop will not last for longer time periods the traveled distance is narrowed and therefore the expression is assumed to hold. To place the coordinate system where the safe stop is assumed to start all positions are subtracted with their starting position. The GPS system gives the heading of the vehicle which is used to rotate the coordinate system around the Z-axis. After the rotation the global coordinate system is obtained and defined at the position where the severe failure is visualized to occur. The GPS system do however not give the traveled distance as position instead the system gives the air distance. This becomes a problem for the reference generation when the vehicle travels in a slope. The altitude given by the GPS is therefore used to recalculate the position in air distance to traveled distance.

The reference system is not placed in the COG of the vehicle, (since the rack did not allow that), therefore the sensor data needed to be compensated for the translational movement that occurred. The COG in x-direction of the vehicle is calculated by using the normal forces obtained by weighing the vehicle on a four point scale, see Figure 6.1.

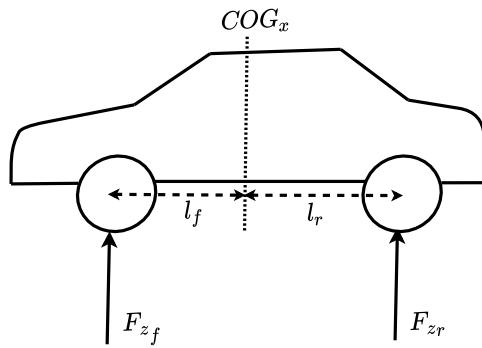


Figure 6.1: The normal forces acting on the vehicle

The COG in x-direction is then calculated by the following expressions;

$$\begin{aligned} F_{z_f} &= mg \frac{l_r}{l_f + l_r} \\ F_{z_r} &= mg \frac{l_f}{l_f + l_r}, \end{aligned} \quad (6.2)$$

where the mass for each wheel is measured by the scale and the sum of l_r and l_f is known. The COG in the y-direction of the vehicle is calculated in the same way but the normal force is calculated for the left and right side of the vehicle instead. Thereafter the RT3000 placement towards COG of the vehicle can be calculated and is obtained as $p=[0.5 \ 0 \ 0.05]^T$ [m]. The compensation for the translational movement of the sensor signals from RT3000 is done in same way as described in Section 3.1.2. The vehicles yaw inertia is previously estimated to 5589 [kgm²] with one driver by the vehicle manufacturer. Since there also are three additional passengers in the vehicle during the data gathering the inertia has to be recalculated. The passengers weights are approximated to 70 [kg] each and the ones in the back seat is located approximately 1 [m] from COG and the passenger in the front seat approximately 0.5 [m]. The inertia for each passenger are then calculated by:

$$I_z = mr^2, \quad (6.3)$$

where r is the distance from the mass to COG and m is the mass of each passenger. The inertia for each passenger is then added to the estimated inertia. To see the parameters used in the filtering process see Appendix F.

Analyzing the reference signals from the RT3000 it is noticed that the longitudinal position do not correspond to the longitudinal velocity. When integrating the longitudinal velocity the traveled distance is repeatedly longer in comparison to the RT3000 GPS. The two signals from RT300 are also compared with a GPS with lower resolution (sampling frequency of 1 Hz) that is placed within the vehicle, henceforth referred as *1Hz GPS*. The 1 Hz GPS signal is treated in the same way as the GPS from the RT3000, which is described above. The three signals do not coincide with each other, in some test cases the position difference are as big as 20 [m]. Since it is not possible to determine which signal that gives the best result all three signals are used as references for the global position estimates. In the rest of the states the RT3000 is used as reference system. This will of course affect the interpretation of the results since they are compared to references that can contain errors.

6.2 Results

It is not possible to re-create the exact same cases as for the simulation but results from similar cases for the gathered data are presented using both the filtering concepts. The deceleration is hard to keep constant since the vehicle used is not self driving, but the deceleration is tried to be kept to $5 \text{ [m/s}^2]$. This keeps the time for the safe stop minimized and at the same time it is not too uncomfortable to experience. The XC90 used to gather the data has only one IMU, this gives the sensor biases a greater impact on the filtering process which affects the results in comparison with the simulation.

The tuning is performed in a similar way as for the simulation, which results in quite similar appearance of the Q and R matrices. The tuning matrices are presented in Appendix G.

The result is presented case wise, just as for the simulation results. All cases are presented with figures for the positions estimates together with their references. Since there are some problem with the position references, (as discussed above), all three references that are available for the positions are presented. In those cases that are not presented with all states they are accessible in Appendix I.

6.2.1 Straight Forward

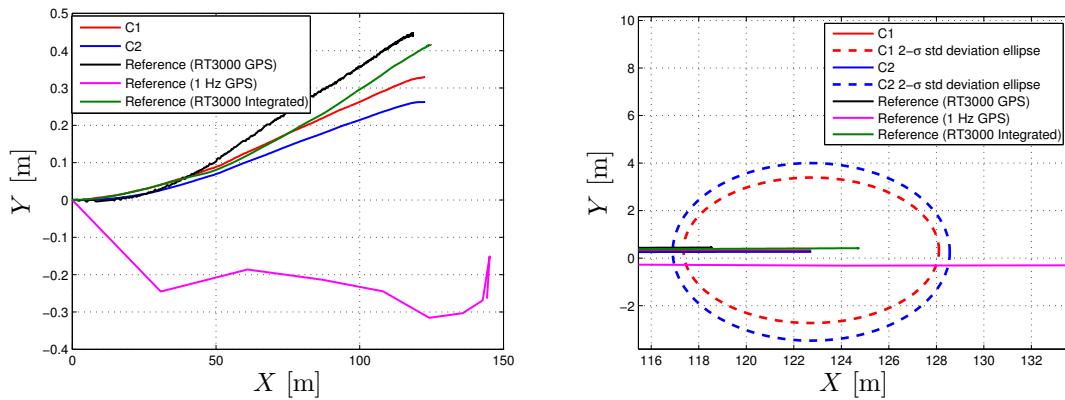
In this scenario the vehicle travels on a straight road when the severe failure is visualized to occur. Thereafter the vehicle decelerates with approximately $5 \text{ [m/s}^2]$ until the velocity of the vehicle is zero. The results for different velocities are presented as e_{lon} , e_{lat} and e_ψ calculated with the RT3000 GPS in Table 6.1. Important to state is that reference from RT3000 might be slightly off. However the trends that can be seen is still valid.

Table 6.1: Filtering result from gathered data when traveling straight forward

	e_{lon} [m]	e_{lat} [m]	e_ψ [rad]
Straight forward			
C1: 120 [km/h]	-4.2301	0.1268	0.0014
C2: 120 [km/h]	-4.2299	0.1937	0.0013
C1: 90 [km/h]	-3.5740	0.2751	-0.0060
C2: 90 [km/h]	-3.5741	-0.1863	-0.0060
C1: 70 [km/h]	-2.2237	0.1201	$6.2484 \cdot 10^{-4}$
C2: 70 [km/h]	-2.2234	0.1528	$6.7000 \cdot 10^{-4}$
C1: 50 [km/h]	-1.9248	-0.0534	$-4.3734 \cdot 10^{-4}$
C2: 50 [km/h]	-1.9249	-0.0210	$-4.6065 \cdot 10^{-4}$

The result for the X- and Y-position estimates when traveling in 120 [km/h] are vis-

ible in Figure 6.2a and in Figure 6.2b the result is zoomed in and the 2σ standard deviation ellipses are visible for the final estimates. Compared to the simulation result the ellipses are bigger which means that the uncertainty of the position estimates has increased.



- (a) The estimated X- and Y- positions illustrated against each other
 (b) 2- σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 6.2: The estimated X- and Y- positions when traveling straight forward

6. Experimental Tests and Results

The result for the position estimates and their standard deviations are presented in Figure 6.3. Here it is visible that there are some misplacement, primarily in X-position, it is though hard to say how exact the estimates are since the estimated value lies somewhere between the RT3000 GPS reference and the 1Hz GPS. For Y-position the deviation from the references seems to be at maximum about 0.4 [m] with some difference between the concepts, however also for this state the estimates lies between the different references which indicates on a reasonable estimate.

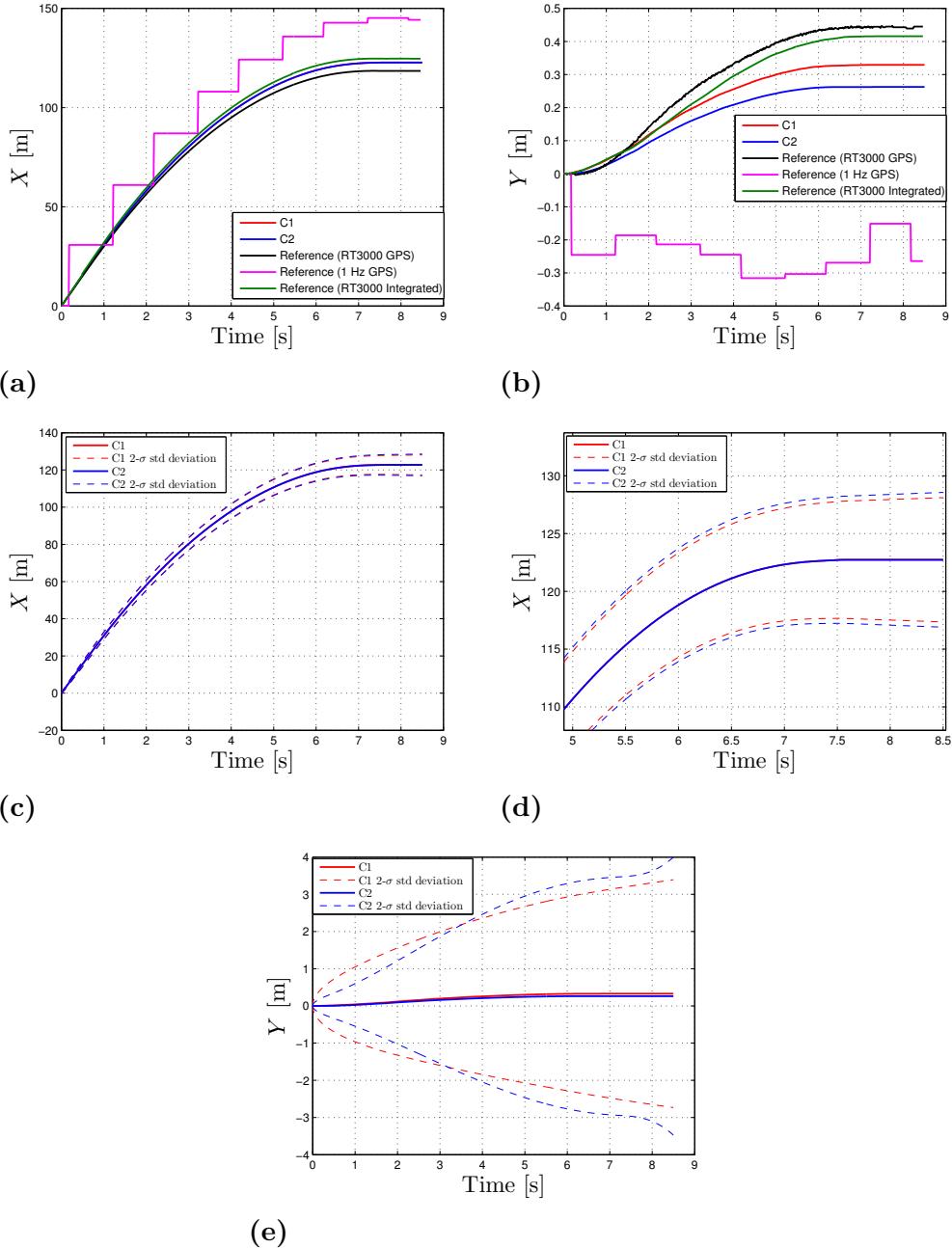


Figure 6.3: Estimated global position states with standard deviations when traveling straight forward

The estimated velocities and accelerations are presented in Figure 6.4. Here the estimated lateral accelerations are less noisy than the reference. This is probably because the reference is a raw sensor signal from the RT3000 system and the estimate is filtered.

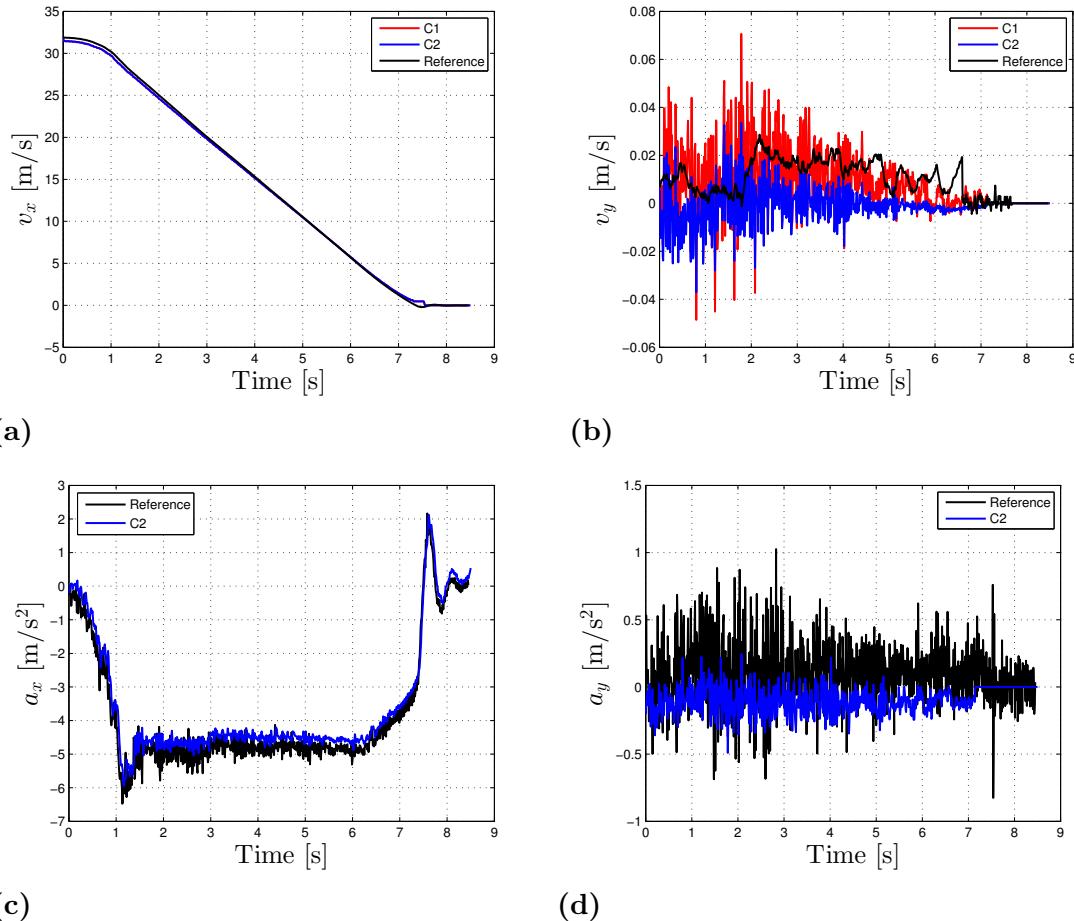


Figure 6.4: Estimated velocity and acceleration states when traveling straight forward

6. Experimental Tests and Results

In Figure 6.5 the Euler angle estimates and their rates are presented. As visible most of the states follow the reference with just some slight deviation on the roll and yaw estimates.

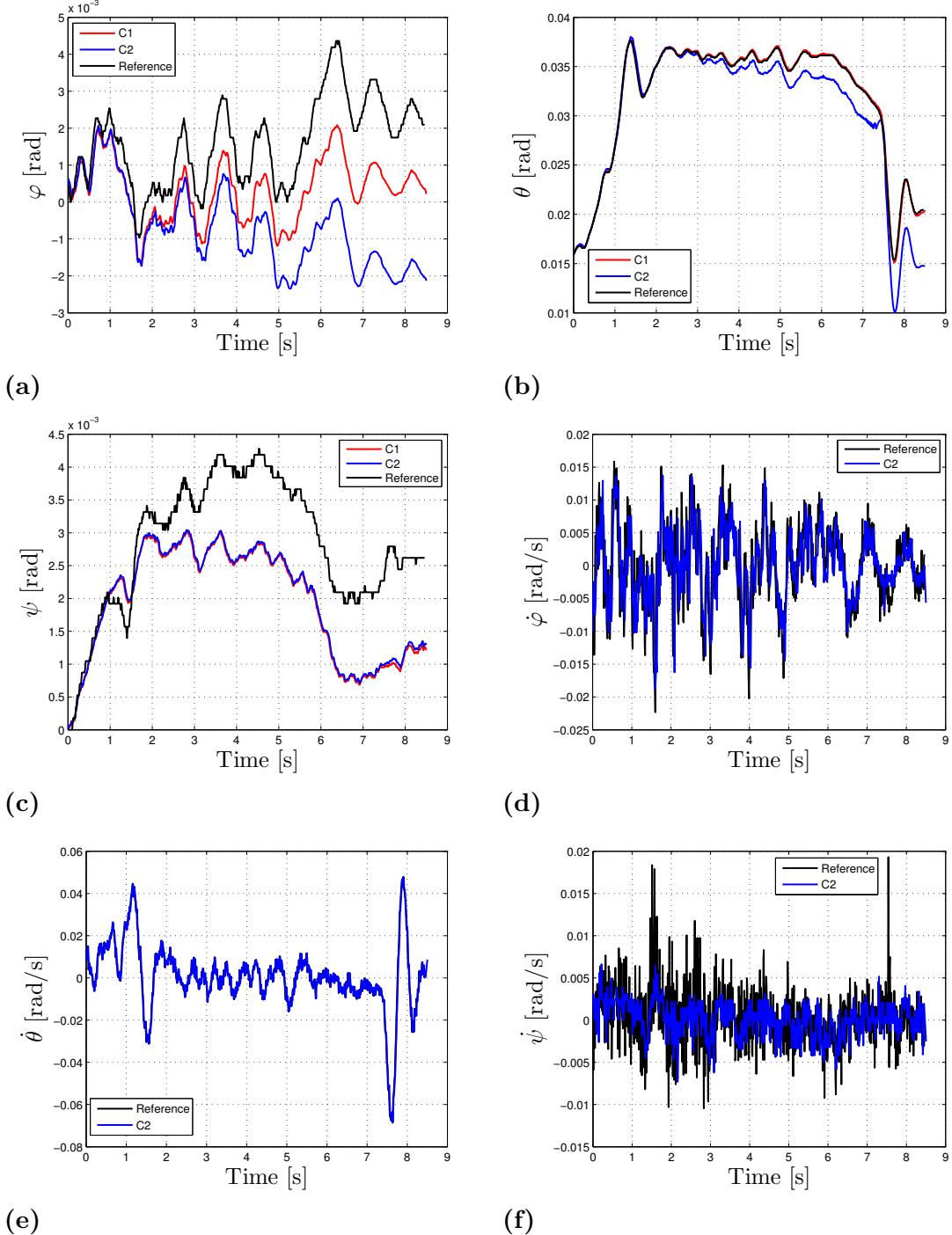


Figure 6.5: Estimated Euler angles and their rates when traveling straight forward

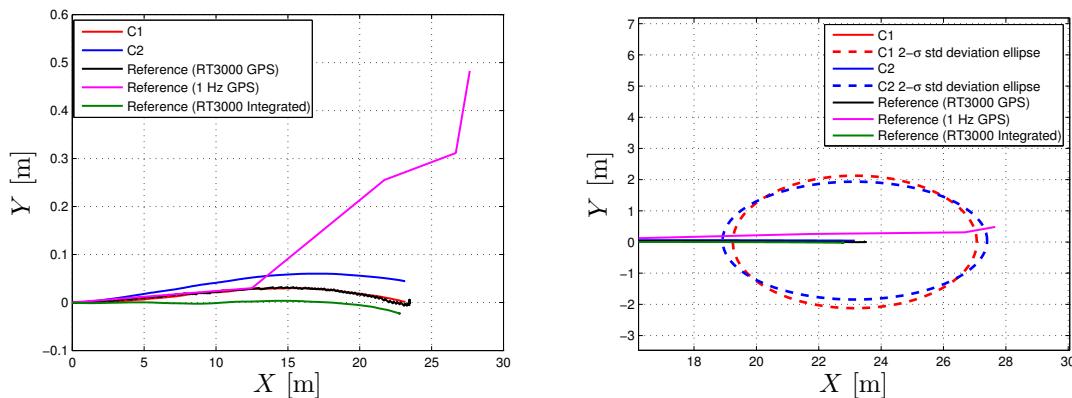
6.2.2 Straight Forward in Slope

In this scenario the vehicle travels straight forward on a road with a negative or positive longitudinal slope of 20 % and then decelerates with approximately 5 [m/s²]. The results for the longitudinal, lateral and yaw errors calculated with the reference from the RT3000 GPS are visible in Table 6.2.

Table 6.2: Filtering result from gathered data when traveling up- and downhill

	e_{lon} [m]	e_{lat} [m]	e_ψ [rad]
Straight forward downhill 20 %			
C1: 50 [km/h]	-0.1470	-0.0373	-0.0028
C2: 50 [km/h]	-0.1467	0.0058	-0.0027
C1: 30 [km/h]	-0.0594	-0.0248	0.0097
C2: 30 [km/h]	-0.0596	-0.0468	0.0095
Straight forward uphill 20 %			
C1: 30 [km/h]	-0.4972	0.0268	0.0041
C2: 30 [km/h]	-0.4971	-0.0064	0.0036

In Figure 6.6a the X- and Y- positions are presented illustrated against each other for the case when traveling initially in 50 [km/h] downhill. Figure 6.6b is zoomed in and the final estimates 2- σ standard deviation ellipses are also visible. In this case the misplacement in both X- and Y- direction is significantly smaller than for the previous case. This is probably due to the lower initial velocity.



- (a) The estimated X- and Y- positions illustrated against each other (b) 2- σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 6.6: The estimated X- and Y- positions traveling straight forward downhill

6. Experimental Tests and Results

The X- and Y-position estimates are illustrated separately in Figure 6.7 together with the standard deviations for each position estimate. The estimates seems to lie quite close to the references, and the deviations are smaller than 1 [m]. The estimates are in this case also always enclosed by the references which just as in the previous case indicates that the estimates are reasonable. The rest of the estimated states for this scenario are presented in Appendix I.

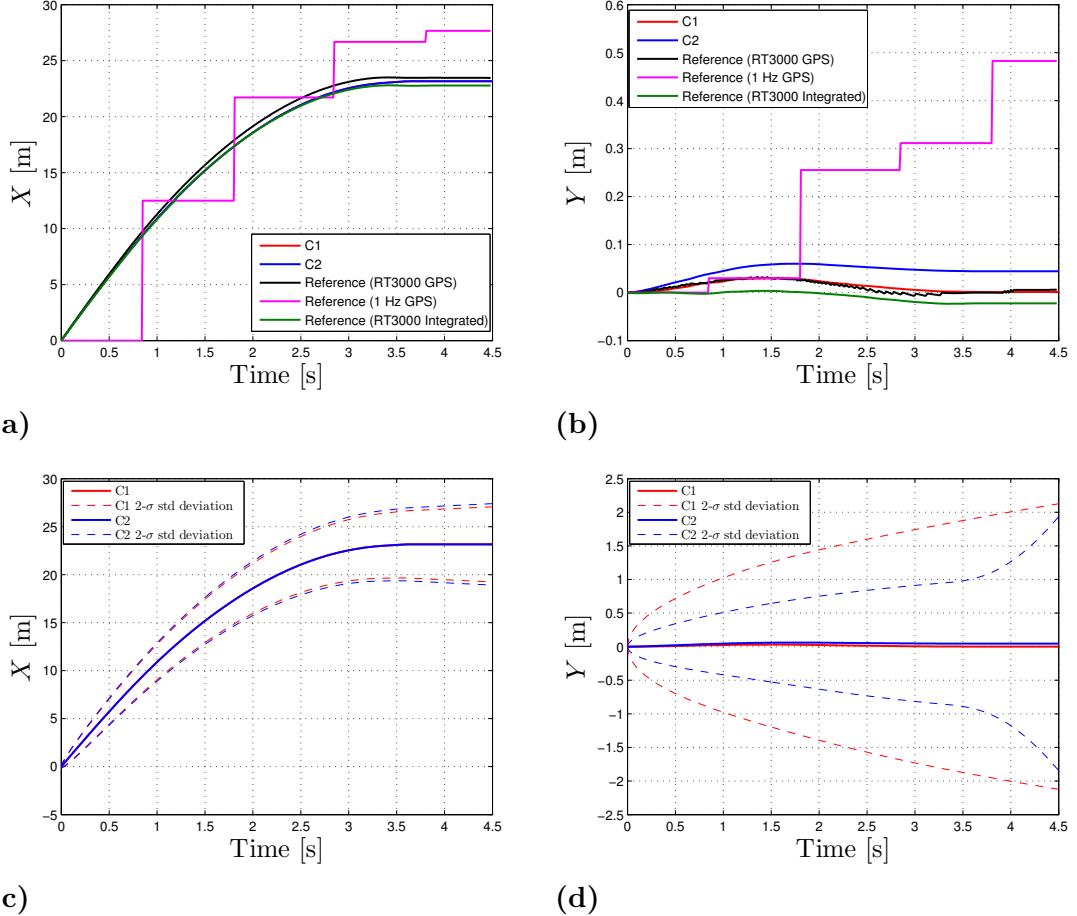


Figure 6.7: Estimated global position states with standard deviations traveling straight forward downhill

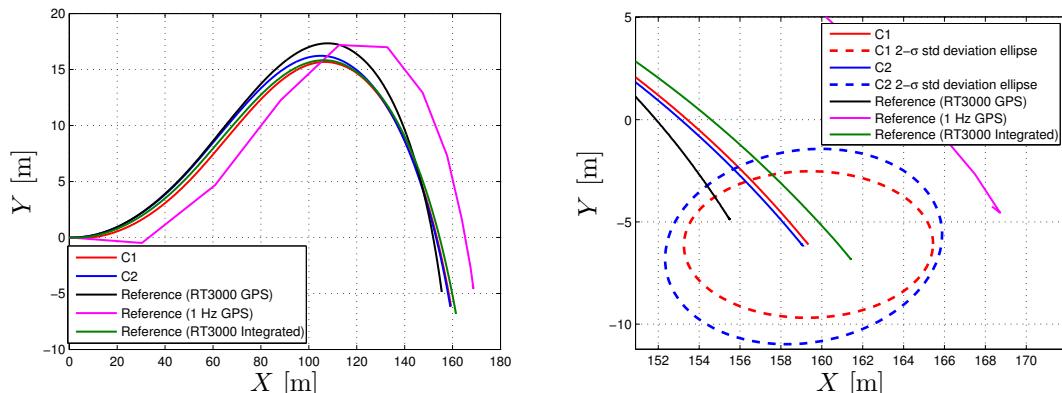
6.2.3 Right Turn

In this case the vehicle is visualized to follow a curved road initially traveling in 90 respective 120 [km/h] but brakes and stops during the curve. The calculated errors using the RT3000 GPS for both scenarios are visible in Table 6.3.

Table 6.3: Filtering result from gathered data when performing a turn

	e_{lon} [m]	e_{lat} [m]	e_ψ [rad]
Right turn			
C1: 120 [km/h]	-3.4668	-2.0992	-0.0056
C2: 120 [km/h]	-3.3862	-1.8543	-0.0075
C1: 90 [km/h]	-3.7111	-4.3550	-0.0139
C2: 90 [km/h]	-4.0545	-3.5949	-0.0149

In Figure 6.8 the X- and Y-positions estimates are visible and also the 2σ standard deviation ellipses for each concept for the case when driving initially in 120 [km/h]. As visible the performance from both concepts are quite similar and the estimates ends up a couple of meters from all the references, but still enclosed by them.



- (a) The estimated X- and Y- positions illustrated against each other (b) 2- σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 6.8: The estimated X- and Y- positions for a right turn

6. Experimental Tests and Results

The X- and Y-positions estimates are separately visible in Figure 6.9 together with respective 2σ standard deviation. As visible the standard deviation keeps growing throughout the whole estimation process which is as expected. The references are though still kept inside the 2σ space all the time except for the 1Hz GPS, which might be due to the low sampling frequency.

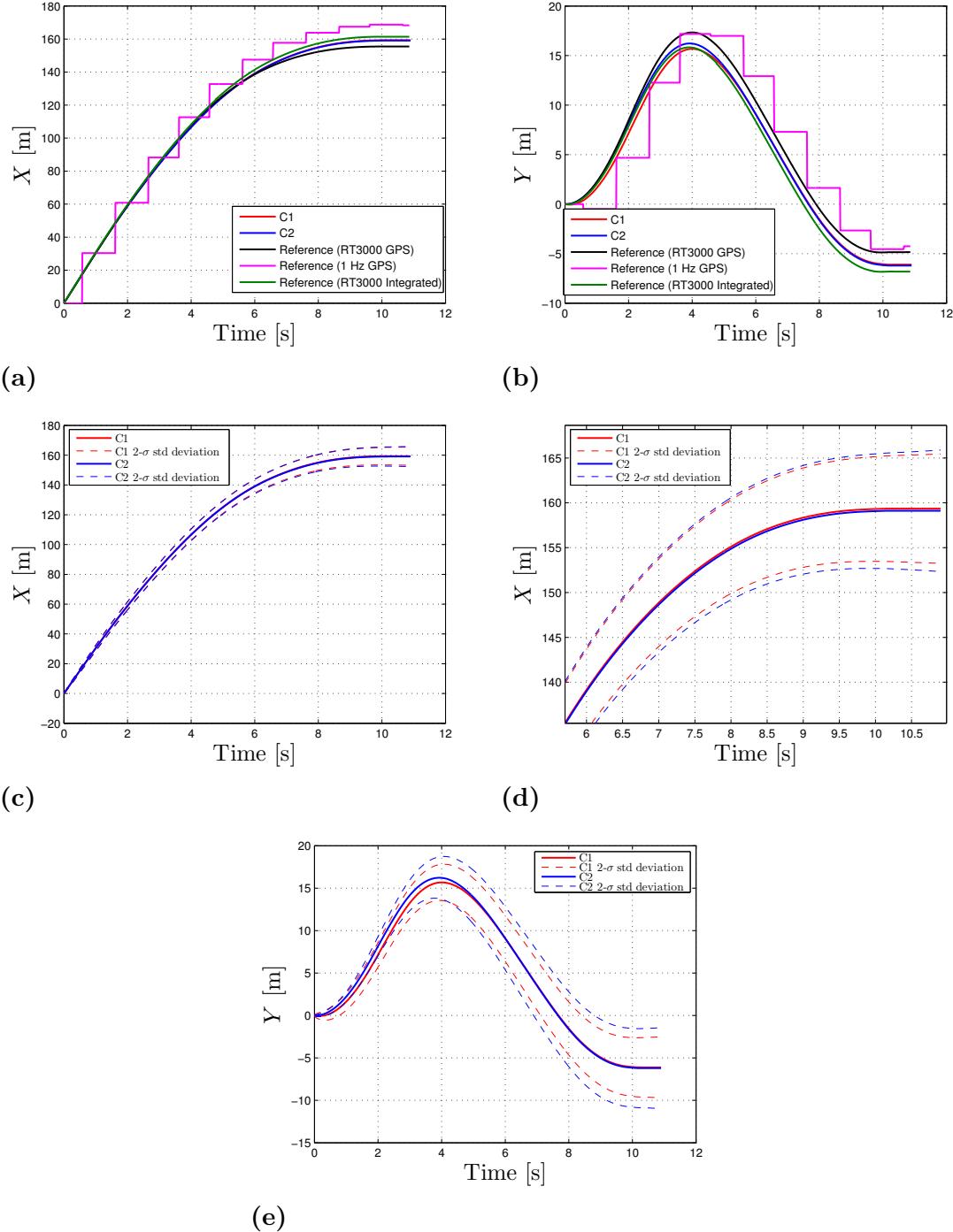


Figure 6.9: Estimated global position states with standard deviations for a right turn

The estimated states for velocities and accelerations are visible in Figure 6.10 and just as before the acceleration estimates are only visible for concept 2. For the lateral velocities there seems to be some difference between the concepts. Concept 2 seems to capture the correct velocity a bit better than concept 1, but for the longitudinal velocity they seem to perform very similar.

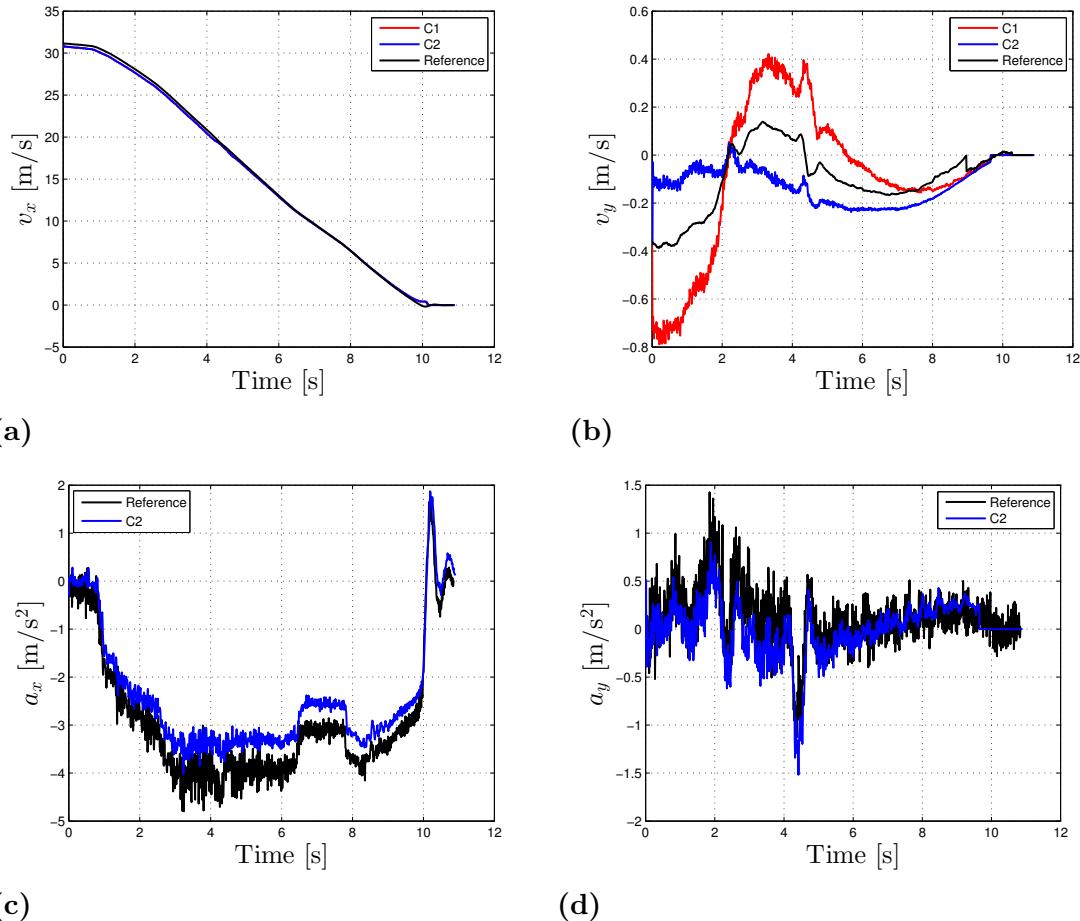


Figure 6.10: Estimated velocity and acceleration states for a right turn

6. Experimental Tests and Results

Estimated states for the Euler angles and their rates are visible in Figure 6.11. In all cases the estimates follows the reference rather well, it is just for the roll- and pitch angle that the estimates wander with time.

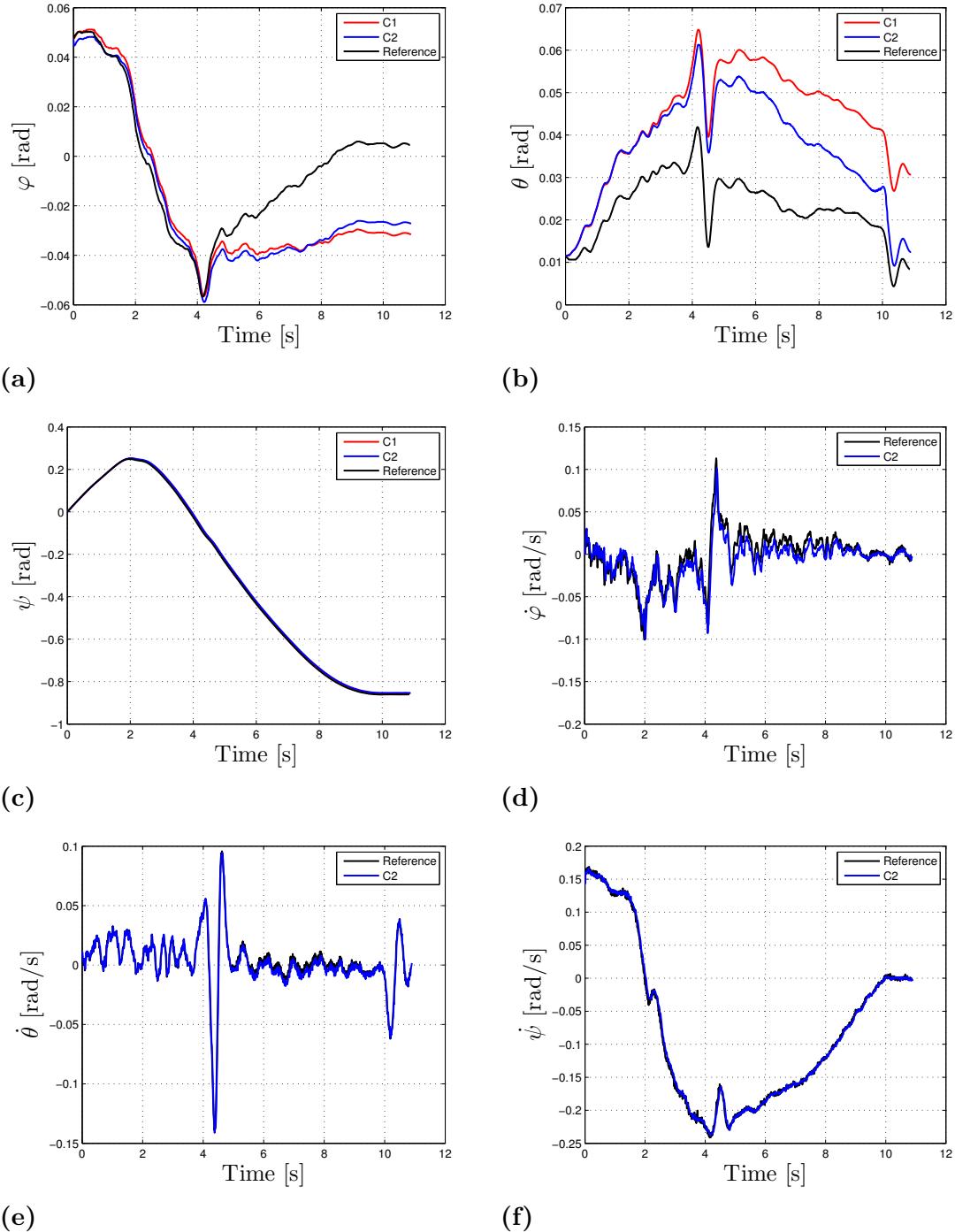


Figure 6.11: Estimated Euler angles and their rates for a right turn

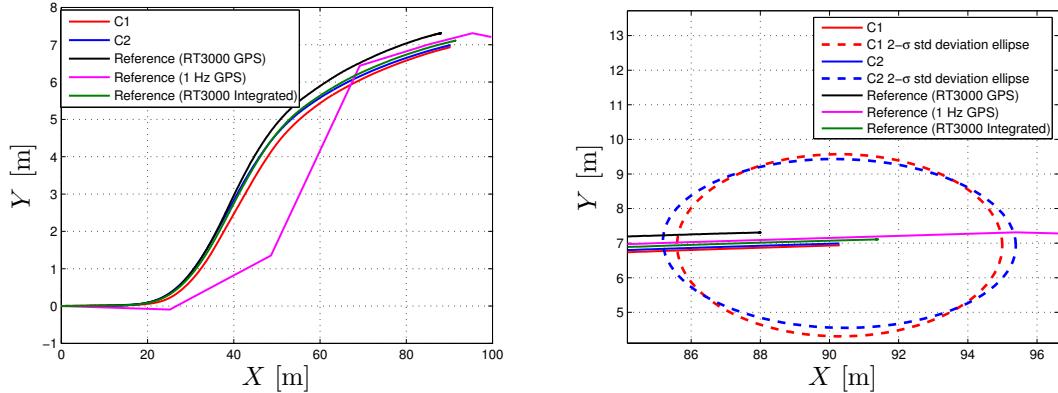
6.2.4 Lane Change

In this case a lane change is performed while braking. This is a realistic scenario in case an actual safe stop would be performed, since it then would be reasonable to stop at the roadside instead of in the middle of the road. The longitudinal, lateral and heading error calculated with RT3000 GPS for different velocities are presented in Table 6.4.

Table 6.4: Filtering result from gathered data during lane change

	e_{lon} [m]	e_{lat} [m]	e_ψ [rad]
Lane change			
C1: 90 [km/h]	-2.3183	0.4315	-0.0025
C2: 90 [km/h]	-2.3110	0.3780	-0.0025
C1: 70 [km/h]	-1.1737	0.2737	0.0036
C2: 70 [km/h]	-1.1692	0.1814	0.0035

In Figure 6.12 the X- and Y-position estimates are illustrated towards each other and also the 2σ standard deviation ellipses for the final estimate with an initial velocity of 90 [km/h].



- (a) The estimated X- and Y- positions illustrated against each other (b) 2σ standard deviation ellipses of the estimates for X- and Y- positions

Figure 6.12: The estimated X- and Y- positions for a lane change

The estimated X- and Y-position are visible separately in Figure 6.13. In this case all the references and estimates are within 0.5 [m] range for the Y-position which indicates that the actual position of the vehicle probably are within that range. For the X-position the difference are quite large between the reference systems, however the estimates are enclosed by the them and quite close to the RT3000 GPS and the integrated velocity. The remaining state estimates for this case are visible in Appendix I.

6. Experimental Tests and Results

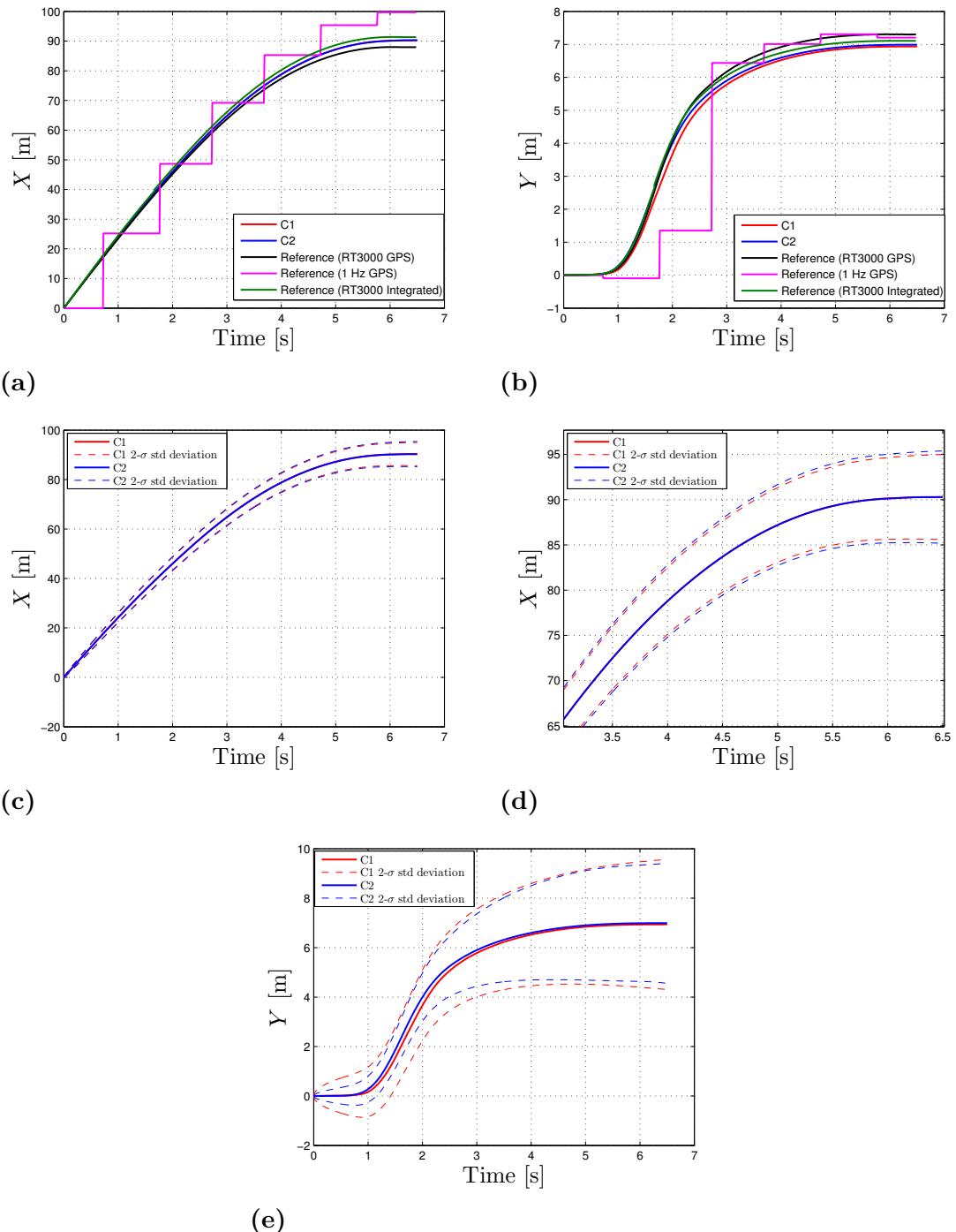


Figure 6.13: Estimated global position states with standard deviations during a lane change

6.3 Summary of Experimental Tests

To summarize the result from the experimental tests the same trends as for the simulated results can be seen. The longitudinal and lateral errors are mostly affected by the initial velocity and the curvature of the road do not have a big impact. Worth mentioning is that in the experimental tests only one IMU sensor was used which probably resulted in bigger errors, it is however hard to say the exact size of the errors since there were problem with the reference.

6. Experimental Tests and Results

7

Discussion and Future Work

The results from the filtering process for the simulated sensor data using CarMaker are the easiest to interpret since all the circumstances, (trajectory, velocity, vehicle parameters etc.), could be monitored and simply changed. This makes it possible to see trends and get a result that is of more statistical significance compared to the results from the gathered data.

7.1 Simulation Results

In the simulation results from the filtering estimation process one trend that can be seen for the longitudinal estimates are that the estimated distances are always shorter than the actual distances traveled. The distributions of the errors are also quite narrow and the difference between the maximum estimate and the minimum estimate for multiple runs are mostly less than 0.2 [m], even when traveling initially in 120 [km/h]. The reason for this behaviour are most certainly the additional sensor (odometer) used to measure the longitudinal velocity. The odometer gives rather accurate estimations on the longitudinal velocity during the conditions that the wheel radius is correctly estimated and that the wheels do not slip. The magnitude of the longitudinal position errors are though rather big when traveling fast which probably depends on that the wheels get some slip and that the wheels radius might have some error. This creates an accumulating error on the traveled distance. The initial velocity do however also have a large impact the longitudinal errors. As visible in the results the error grows almost quadratic when the velocity increases, which also supports the fact that the odometer is causing some error. This since if as mentioned the wheel radius is inaccurately estimated it will create a constant bias error on the velocity estimate which then results in a growing error on the position estimate that gets bigger for each time instance, which just is the case for our estimate. For example if the initial velocity is decreased to 90 [km/h] the errors of the final estimate decrease to approximately 0.7 [m] compared to around 1.3 [m] for 120 [km/h] when traveling straight forward.

For the lateral position estimate there are no additional sensors that corrects the estimates, instead the filtering process relies on dynamical vehicle models to estimate the lateral velocity and from there estimate the position. This fact makes

the distribution of the errors quite large, especially for high initial velocities. The reason for this high variance is since the noise of the IMU:s have quite big impact on this state and the noise of the modeled lateral velocity is also quite high in some situations creating a total variance that is even larger. The difference between the minimum and maximum estimates can be as large as 4 [m] for the same initial velocity when filtering the same case multiple times. The mean of the estimates are though always almost zero when performing multiple estimations for the same scenario, which indicates that there are no accumulating error in this case. However to lower the spread of the estimates the initial velocity can be reduced and just as for the longitudinal estimate this has a big impact, but this time on the distribution more than on the mean. If decreasing the initial velocity to 90 [km/h] the absolute value of the lateral error is reduced to approximately 1 [m] at maximum and for 80 [km/h] it is approximately 0.7 [m] at maximum, though the mean on the absolute value on multiple runs are much lower (0.4-0.5 [m]).

As mentioned above for both the longitudinal and lateral position estimates the initial velocity of the vehicle have a very big impact on how close the estimates are to the references. The reason that the initial velocity have this impact is since it directly affect the time of the safe stop. The dead reckoning process is very time sensitive since there are no measurements directly on the positions and if the time of the filtering process increases this directly affects the performance. As seen the position errors grows approximately quadratically when the time increases, this supports the previously theory that when estimating position using accelerometers the errors will grow with $\frac{\epsilon t^2}{2}$, (where ϵ is the constant sensor bias). A way to make it possible to have a higher initial velocity would be to also have a higher deceleration but due to the passenger comfort this could be challenging. Braking with a higher deceleration than 5 [m/s²] is not comfortable, especially when not prepared for it and the risk of being hit by another vehicle is also significantly increased when performing a harsh stop.

The estimation errors on the yaw angle of the vehicle are essentially almost lower than 1° for the last estimates, even when performing a turn. This gives a minimal negative impact on the estimates of the position states. The reason that these estimates are so close to the actual heading of the vehicle probably depends on that the gyroscope has relatively good performance and that there only requires one integration to achieve the angle. The sensitivity for initial velocity and increased bias on the sensor do neither seem to be that big compared to the position estimates, the errors do increase but are still almost always below 1° for the last estimates. The only drawback with the yaw estimate is that it is expressed using Euler angles which is not a durable. In case the angle exceeds 90° the estimate can go into Gimbal lock and thereby spoil the other estimates. To solve this problem quaternions can be used instead of Euler angles to represent the angles. Another solution might be to make sure that the estimate do not exceed 90°. This can be done by compensate if the angle approaches the limit by removing 90° from the estimated angle and manually track the magnitude of the estimate during the time it exceeds 90°.

Analyzing the rest of the states they mostly seems to lie within reasonable distance

from the reference, especially the states that are directly measured. The noise is though visible on some states and causes their integrates to drift, this appears as a random walk as previously discussed. Since the time of dead reckoning is quite short the random walk do not have a significant impact for the estimates.

The number of IMU:s used, their bias levels and variances do also as visible in the results have an impact on the final position estimates. When using multiple numbers of IMU:s the biases decreases, which have the same kind of affect on the bias as if a sensor with lower constant bias was used. The size of the sensor biases affects all estimates but not that much in the longitudinal estimates compared to the other estimates since they are corrected with the help from the odometer. The impact on the lateral position estimates are quite big since the constant bias on the sensors is directly integrated up to an error in velocity/angle and thereafter possibly integrated up to an even bigger error in position which keeps on growing if the biases remains. If the constant biases is reduced by using multiple IMU:s or better IMU:s with lower biases on the sensors this will directly affect the performance in a positive way. Especially by changing from one to two IMU:s could reduce the error with approximately 0.5 [m], of course depending on how good the sensors are initially. This issue could also be solved through bias compensation in case the bias is known or possible to estimate, this do however not decrease the variances of the measurements which adding another IMU does.

The significance of all the simulated results are affected by multiple factors. First of all CarMaker is a simulation environment and it can never fully represent an actual vehicle. However the model is probably as realistic as a vehicle model can be and therefore gives very trustworthy simulated sensor readings and also represents how the vehicle behaves in a realistic way. The advantages of using this simulation environment is that it was possible to run a large quantity of similar scenarios with the exact same circumstances and therefore create results with some statistical significance level. For the lateral position estimate, since the distribution can be assumed to be Gaussian it was stated that the absolute value of the error with 68% confidence would be smaller than 0.88 [m] when driving initially in 120 [km/h]. If decreasing to 80 [km/h] the absolute value of the error would be smaller than 0.37 [m] with 68% confidence, and smaller than 0.73 [m] with 95% confidence (2 standard deviations). For the longitudinal estimation errors it was challenging to determine any statistical error margin since the distribution is not normal, however the results do not have a large spread which makes it easier to reduce the error. This by either improve the wheel radius estimation or slip compensate the odometer readings.

7.2 Experimental Tests

The filtering results from the gathered data are not that easy to interpret compared to the simulation results. This is mostly due to the problems with the references but also some due to the uncertainty with the correctness of the vehicle parameters. The problems with the position references are comprehensive and can have many causes

but one of the reasons might be insufficient initiation of the RT3000. Using the 1 Hz GPS as reference is not optimal either, mostly due to the lower resolution, (1 Hz vs 100 Hz). The calculated positions given by integrating the velocities from the RT3000 may also have some problems with the accuracy, since by using integration a small bias in the velocity estimate will affect the results in position even more since dead reckoning are used. However the filtering estimates lies in most of the cases within the interval between the three references, which indicates that the filtering estimates are reasonable. The fact that the estimated states from the gathered data has a similar appearance when comparing with a similar case for simulation, (in for example the velocities and pitch angle states), supports the statement above. Another aspect that is important to emphasize is that the three references are based on different sensors and two different systems which increases the probability that the actual position is close to any of the three references.

The longitudinal distance traveled given by the estimated position states are mostly shorter than the reference given by the 1 Hz GPS. This supports the previously stated theory that the wheels locks and therefore the velocity given from the odometer needs to be slip compensated. The integrated reference and the position estimates are always quite close to each other and in most cases the estimated distance in longitudinal direction is slightly shorter than the reference. However the distance given by the position estimates are in most of the cases longer than the distance given by the reference from the GPS of the RT3000, therefore it is hard to conclude anything. Therefore further testing with a more accurate reference system is needed.

To further investigate the filter performance the robustness should also be tested with real data, this was not possible within the scope of this thesis. To test the robustness a predetermined trajectory could be followed repeatedly with a specified velocity for as many different scenarios as possible so statistical results could be obtained. These tests should of course be performed with a more reliable reference system to achieve a trustworthy result.

7.3 Concept Evaluation

The two concepts used throughout the filtering processes are as visible performing quite similar and in some cases one performs better than the other and in other cases it is the opposite. Worth noticing is though that concept 1 uses one measurement less, (since the pinion angle is not used). It also has five states less due to that it uses the IMU readings as inputs to the prediction step instead of the update step, (which requires states for the accelerations and Euler angular rates). Concept 2 do however have a more advanced dynamical vehicle model which takes more states and measurements into account. This have though not given a significant improvement in the results compared with concept 1, (just a couple of centimeters in lateral position). It is though hard to know if it would give a more significant difference when performing more advanced manoeuvres. However this is not something that could be stated when analyzing the gathered data where more challenging manoeuvres

was performed. When the different scenarios has been tested it could be stated that the vehicles trajectory is not significant for the filtering process as long as the lateral acceleration do not exceed the boundary set up in the scope, ($a_y < 2$ [m/s²]). To summarize it can be stated that concept 1 is more suitable to use in this application due to the smaller state vector and the similarity in results compared to concept 2.

7.4 Improvements and Future Work

To improve the performance of the filter in future work the most effective method is probably to improve the reliability on the sensors. An example on how the reliability of the sensors could be improved are by slip compensating the odometer, this would probably have a big impact on the longitudinal error. If this is perfectly done and the wheel radius is correctly estimated the error would be zero mean instead of shifted. The IMU sensors can also be offset compensated, this would probably affect the results of all states and as discussed previously the compensation should be performed at least with a 30 seconds interval to achieve best result. Different approaches to filter the sensor readings could also be investigated, for example start by fusing the different IMU:s separately with the odometer, receiving three state estimates that then are fused together in a main filter. Other approaches to improve the filter performance in future work could be to investigate different sensor fusion filters, such as particle filter, UKF etc. Different vehicle models describing lateral motion and more possible concepts could also be investigated further.

7. Discussion and Future Work

8

Conclusion

The main conclusion in this thesis is; it is possible to use dead reckoning for position estimation of a vehicle during a limited amount of time without accumulating large estimation errors. It can also be concluded that the circumstances of the safe stop has a crucial part of the estimation errors. The condition that has the largest impact is the amount of time elapsed from the severe failure to the complete stop, which is the time of the dead reckoning process. With high initial velocities the time of the safe stop is inevitably extended therefore by lower the initial velocity the estimation errors will be drastically reduced. The sensors performances do also have a large impact on the filtering process and by increasing the number of IMU:s or offset compensate the sensors the performance can be improved. The sensor bias of the IMU:s will though not wander significantly during the time of the safe stop which makes it unnecessary to bias compensate during that time. When driving initially in 80 [km/h], using 3 IMU:s and no bias compensation the set up objectives (a longitudinal error of 3 [m] and a lateral error of 0.75 [m]) is fulfilled with 95% confidence. However the initial velocity specified in the objective was 120 [km/h] but the lateral estimation error for that velocity could not fulfill the objective with the set up confidence.

8. Conclusion

Bibliography

- [1] Volvo Cars. Drive me. [online]. Available: <http://www.volvcars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving/drive-me>. Accessed: 2017-01-19.
- [2] Uber Newsroom. Pittsburgh, your self-driving uber is arriving now. [online]. Available: <https://newsroom.uber.com/pittsburgh-self-driving-uber/>, 2016. Accessed: 2017-01-17.
- [3] Uber Newsroom. San francisco, your self-driving uber is arriving now. [online]. Available: <https://newsroom.uber.com/san-francisco-your-self-driving-uber-is-arriving-now/>, 2016. Accessed: 2017-01-17.
- [4] Andrew J. Hawkins. Delphi and mobileye are teaming up to build a self-driving system by 2019. [online]. Available: <http://www.theverge.com/2016/8/23/12603624/delphi-mobileye-self-driving-autonomous-car-2019>, 2016. Accessed: 2017-01-17.
- [5] IEEE. Ieee news. [online]. Available: http://www.ieee.org/about/news/2012/5september_2_2012.html, 2012. Accessed: 2017-01-17.
- [6] Alexander C. Kaufman. Elon musk: We'll have driverless cars by 2023. [online]. Available: http://www.huffingtonpost.com/2014/10/15/tesla-driverless-cars_n_5990136.html, 2014. Accessed: 2017-01-17.
- [7] Chan Wei Hsu Shun-Hung Chen and Shih Chieh Huang. Recursive estimation of vehicle position by using navigation sensor fusion. *2012 12th International Conference on ITS Telecommunications*, 2012.
- [8] Salman Ali Shaukat et al. Robust vehicle localization with gps dropouts. *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2016.
- [9] O.J. Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, August 2007.
- [10] Kecman M. Sagnac effect and fiber optic gyroscopes. *Journal of physical mathematics*, 7, 2016.

- [11] J-O. Nilsson I. Skog and P. Händel. An open-source multi inertial measurement unit (MIMU) platform. *International Symposium on Inertial Sensors and Systems (ISISS)*, 2014.
- [12] S. Guerrier. Improving accuracy with multiple sensors: Study of redundant MEMS-IMU/GPS configurations. 2009.
- [13] H. Hou N. El-Sheimy and X. Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 2008.
- [14] Wolfram Alpha. Rotation matrix. [online]. Available: <http://mathworld.wolfram.com/RotationMatrix.html>. Accessed: 2017-02-15.
- [15] Wei-Chih Wang. *Lecture: ME 230 Kinematics and Dynamics*. Department of Mechanical Engineering University of Washington, Feb. 2014.
- [16] L. Villani G. Oriolo B. Siciliano, L. Sciavicco. *Robotics Modelling, Planning and Control*. Springer, 2010.
- [17] Georg Rill Wolfgang Hirschberg and Heinz Weinfurter. User-appropriate tyre-modelling for vehicle dynamics in standard and limit situations. *International Journal of Vehicle Mechanics and Mobility*, Vehicle System Dynamics 38:2:103–125, 2002.
- [18] M. Lundgren. Bayesian filtering for automotive applications. *PhD Thesis Chalmers University of Technology*, 2015.
- [19] Hwei P. Hsu. *Probability, Random Variables, and Random Processes*. McGraw-Hill, 2014.
- [20] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [21] IPG Automotive. CarMaker: Virtual testing of automobiles and light-duty vehicles. [online]. Available: <https://ipg-automotive.com/products-services/simulation-software/carmaker/>, 2017. Accessed: 2017-04-11.

A

Appendix

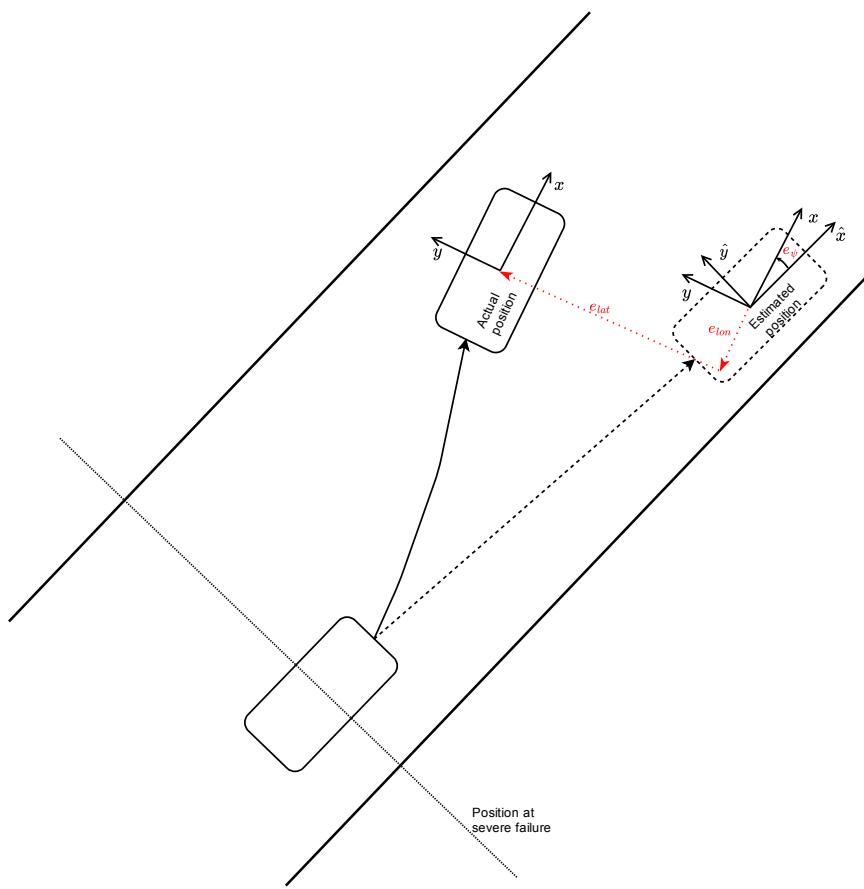


Figure A.1: Definition of the estimation errors in lateral-, longitudinal position and heading

A. Appendix

B

Appendix

B.1 IPG CarMaker

IPG CarMaker is used as a simulation tool together with Simulink in this thesis. In CarMaker it is possible to design different road properties such as gradient, slope, width of the road etc. through a GUI. It is possible to control the vehicle in different ways, for example; using the pinion angle and the desired velocity as input or assign the vehicle to follow the road with a desired velocity. It is possible to create a specific model of a vehicle, in this thesis a model of a the XC90 created by Volvo Cars is used. Different sensors, such as IMU sensors, can be placed in any position of the vehicle. By simulation in Simulink it is possible to obtain the vehicles position, velocity, IMU signals, wheel speed etc. In this thesis CarMaker is mainly used to verify the dynamical and kinematic models and to evaluate the filter performance. To read more about CarMaker see [21].

B. Appendix

C

Appendix

The data sheet describing the specifications of the XC90 IMU used for the data gathering.

Angular rate Sensor

Requirement	Applicable Signal	Performance Commitment (Production Level)	Units
Operating range	Yaw, Roll	> ± 300	deg / sec
Output resolution (Quantization)	Yaw, Roll	0,0125	deg / sec
Total Scale Factor Error	Yaw, Roll	< ± 3	%
Total Offset	Yaw, Roll	< ± 3	deg / sec
Offset Drift Due to Change in Temperature	Yaw, Roll	< 0.1	deg / sec / °C
Total Offset Drift Due to Temperature	Yaw, Roll	< ± 1,5	deg / sec
Zero Rate Output Drift Velocity	Yaw, Roll	< 0.1	deg / sec / min
Cross Axis Sensitivity	Yaw, Roll	< ± 5.2 note: on sensor level < ± 3	%
G Sensitivity sine wave +/-2g	Yaw, Roll	< 0.2 (7Hz to 100Hz) < 0.6 (100Hz to 2kHz)	deg / sec / g
Signal Noise - RMS	Yaw, Roll	< 0.25 for 26Hz BW < 0.19 (4 sigma value)	deg / sec
Signal Noise - PK - PK	Yaw, Roll	< 1.5	deg / sec

Accelerometer Sensor

Requirement	Applicable Signal	Performance Commitment (Production Level)	Units
Range	X,Y,Z	< +/-14	g
Output Resolution (Quantization)	X,Y,Z	< 4	mg
Total Scale Factor Error	X,Y,Z	< ± 3	%
Total Offset	X,Y	< ± 50 (after factory Calibration)	mg
Total Offset	Z	< ± 90 (after factory Calibration)	mg
Total Offset Drift Due to Temperature	X,Y	< ± 30	mg
Total Offset Drift Due to Temperature	Z	< ± 65	mg
Zero Rate Output Drift Velocity	X,Y,Z	< 20	mg / min
Cross Axis Sensitivity	X,Y,Z	< ± 5.2 note: on sensor level < ± 2.5	%
Signal Noise - RMS	X,Y,Z	< 5	mg
Signal Noise - PK-PK	X,Y,Z	< 30	mg

Attenuation and Delay

Requirement	Applicable Signal	Performance Commitment (Production Level)	Units
Bandwidth @ -3dB	X,Y,Z, Yaw, Roll	< 30	Hz
Pass Band Gain	X,Y,Z, Yaw, Roll	< ± 0.1	dB
Group Delay	X,Y,Z, Yaw, Roll	< 15	ms

C. Appendix

D

Appendix

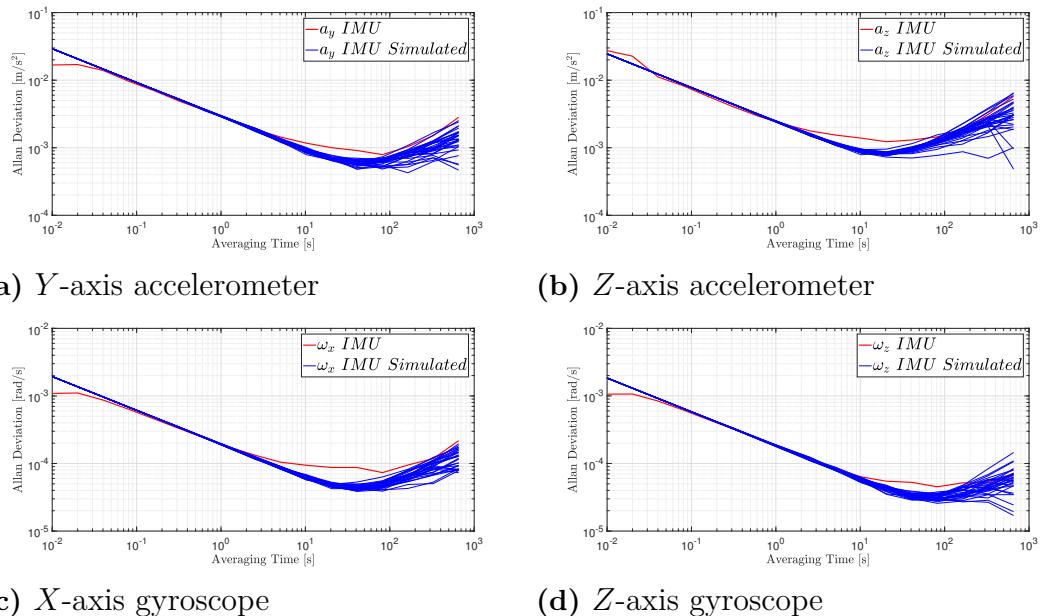


Figure D.1: Allan deviations of the simulated accelerometer and gyroscope for 30 runs compared with the Allan deviation for the vehicle IMU

D. Appendix

E

Appendix

Jacobians of the process and measurement models for the two filtering concepts.

E.1 Concept 1

$$F = \begin{bmatrix} F1 & F2 \end{bmatrix} \quad (\text{E.1})$$

$$F1 = \begin{bmatrix} 1 & 0 & t_0 \cos(\psi) & -t_0 \sin(\psi) & 0 \\ 0 & 1 & t_0 \sin(\psi) & t_0 \cos(\psi) & 0 \\ 0 & 0 & 1 & t_0 \omega_z & 0 \\ 0 & 0 & -t_0 \omega_z & 1 & -gt_0 \cos(\theta) \cos(\varphi) \\ 0 & 0 & 0 & 0 & t_0(\omega_y \cos(\varphi) \tan(\theta) - \omega_z \tan(\theta) \sin(\varphi)) + 1 \\ 0 & 0 & 0 & 0 & -t_0(\omega_z \cos(\varphi) + \omega_y \sin(\varphi)) \\ 0 & 0 & 0 & 0 & t_0((\omega_y \cos(\varphi)) / \cos(\theta) - (\omega_z \sin(\varphi)) / \cos(\theta)) \end{bmatrix} \quad (\text{E.2})$$

$$F2 = \begin{bmatrix} 0 & -t_0(v_y \cos(\psi) + v_x \sin(\psi)) \\ 0 & t_0(v_x \cos(\psi) - v_y \sin(\psi)) \\ gt_0 \cos(\theta) & 0 \\ gt_0 \sin(\theta) \sin(\varphi) & 0 \\ t_0(\omega_z \cos(\varphi)(\tan(\theta)^2 + 1) + \omega_y \sin(\varphi)(\tan(\theta)^2 + 1)) & 0 \\ 1 & 0 \\ t_0((\omega_z \cos(\varphi) \sin(\theta)) / \cos(\theta)^2 + (\omega_y \sin(\theta) \sin(\varphi)) / \cos(\theta)^2) & 1 \end{bmatrix} \quad (\text{E.3})$$

E.2 Concept 2

$$F = \begin{bmatrix} F1 & F2 \end{bmatrix} \quad (\text{E.4})$$

E. Appendix

$$F1 = \begin{bmatrix} 1 & 0 & t_0 \cos(\psi) & -t_0 \sin(\psi) & 0 & 0 \\ 0 & 1 & t_0 \sin(\psi) & t_0 \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 & t_0 & 0 \\ 0 & 0 & 0 & 1 & 0 & t_0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & ((2C_f(v_y + l_f \dot{\psi})) / (mv_x^2)) - \dot{\psi} + (2C_r(v_y - l_r \dot{\psi})) / (mv_x^2) & -((2C_f) / (mv_x)) + (2C_r) / (mv_x) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_0((2C_f l_f (v_y + l_f \dot{\psi})) / (Izv_x^2) - (2C_r l_r (v_y - l_r \dot{\psi})) / (Izv_x^2)) & -t_0((2C_f l_f) / (Izv_x) - (2C_r l_r) / (Izv_x)) & 0 & 0 \end{bmatrix} \quad (E.5)$$

$$F2 = \begin{bmatrix} 0 & 0 & -t_0(v_y \cos(\psi) + v_x \sin(\psi)) & 0 & 0 & 0 \\ 0 & 0 & t_0(v_x \cos(\psi) - v_y \sin(\psi)) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -(v_x + (2C_f l_f) / (mv_x)) - (2C_r l_r) / (mv_x) & 0 \\ 1 & 0 & 0 & t_0 & 0 & 0 \\ 0 & 1 & 0 & 0 & t_0 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 - at_0((2C_f l_f^2) / (Izv_x) + (2C_r l_r^2) / (Izv_x)) \end{bmatrix} \quad (E.6)$$

$$H = [H1 \quad H2] \quad (E.7)$$

$$H1 = \begin{bmatrix} 0 & 0 & 0 & \dot{\theta} \sin(\varphi) - \dot{\psi} \cos(\theta) \cos(\varphi) & 1 & 0 & v_y(\dot{\theta} \cos(\varphi) + \dot{\psi} \cos(\theta) \sin(\varphi)) \\ 0 & 0 & \dot{\psi} \cos(\theta) \cos(\varphi) - \dot{\theta} \sin(\varphi) & 0 & 0 & 1 & g \cos(\theta) \cos(\varphi) - v_x(\dot{\theta} \cos(\varphi) + \dot{\psi} \cos(\theta) \sin(\varphi)) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dot{\psi} \cos(\theta) \cos(\varphi) - \dot{\theta} \sin(\varphi) \\ 0 & 0 & 0 & 0 & 0 & 0 & -\dot{\theta} \cos(\varphi) - \dot{\psi} \cos(\theta) \sin(\varphi) \end{bmatrix} \quad (E.8)$$

$$H2 = \begin{bmatrix} v_y \dot{\psi} \cos(\varphi) \sin(\theta) - g \cos(\theta) & 0 & 0 & v_y \sin(\varphi) & -v_y \cos(\theta) \cos(\varphi) \\ -g \sin(\theta) \sin(\varphi) - v_x \dot{\psi} \cos(\varphi) \sin(\theta) & 0 & 0 & -v_x \sin(\varphi) & v_x \cos(\theta) \cos(\varphi) \\ 0 & 0 & 0 & 0 & 0 \\ -\dot{\psi} \cos(\theta) & 0 & 1 & 0 & -\sin(\theta) \\ -\dot{\psi} \sin(\theta) \sin(\varphi) & 0 & 0 & \cos(\varphi) & \cos(\theta) \sin(\varphi) \\ -\dot{\psi} \cos(\varphi) \sin(\theta) & 0 & 0 & -\sin(\varphi) & \cos(\theta) \cos(\varphi) \end{bmatrix} \quad (E.9)$$

F

Appendix

F.1 Simulation Parameters

Table F.1: Vehicle parameters used in filtering process for the simulated data

Cornering stiffness front [N/rad]	160602
Cornering stiffness rear [N/rad]	214454
Vehicle mass [kg]	2107
Distance from COG to front axle [m]	1.352656045
Distance from COG to rear axle [m]	1.632343955
Yaw moment of inertia [kg·m ²]	4928
Wheel radius [m]	0.3622
Ratio pinion to front wheel	1/16.75

F.2 Gathered Data Parameters

Table F.2: Vehicle parameters used in filtering process for the gathered data

Cornering stiffness front [N/rad]	159000
Cornering stiffness rear [N/rad]	253000
Vehicle mass [kg]	2631
Distance from COG to front axle [m]	1.4732
Distance from COG to rear axle [m]	1.5118
Yaw moment of inertia [kg·m ²]	5746.5
Wheel radius [m]	0.362
Ratio pinion to front wheel	1/16.75

G

Appendix

Here the tuning matrices used in the filtering process for both the simulated and real data are presented for both concepts.

G.1 Simulation

G.1.1 Concept 1

$$Q = \begin{bmatrix} 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 \cdot 10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-8} \end{bmatrix} \quad (\text{G.1})$$

$$R = \begin{bmatrix} 6.3 \cdot 10^{-4} & 0 \\ 0 & 0.001 \end{bmatrix}$$

G.1.2 Concept 2

$$Q = \begin{bmatrix} 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.3 \cdot 10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \quad (\text{G.2})$$

$$R = \begin{bmatrix} 0.0005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.3 \cdot 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \cdot 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \cdot 10^{-6} \end{bmatrix}$$

G.2 Real data

G.2.1 Concept 1

$$Q = \begin{bmatrix} 8.5 \cdot 10^{-3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.5 \cdot 10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-8} \end{bmatrix} \quad (\text{G.3})$$

$$R = \begin{bmatrix} 6.3 \cdot 10^{-4} & 0 \\ 0 & 0.0001 \end{bmatrix}$$

G.2.2 Concept 2

$$Q = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \end{bmatrix} \quad (\text{G.4})$$

$$R = \begin{bmatrix} 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.3 \cdot 10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \cdot 10^{-6} & 0 \end{bmatrix}$$

H

Appendix

The distributions of the longitudinal, lateral and heading errors for the simulated data. The errors are presented for case 2 and 3 using both concepts.

H.1 Case 2

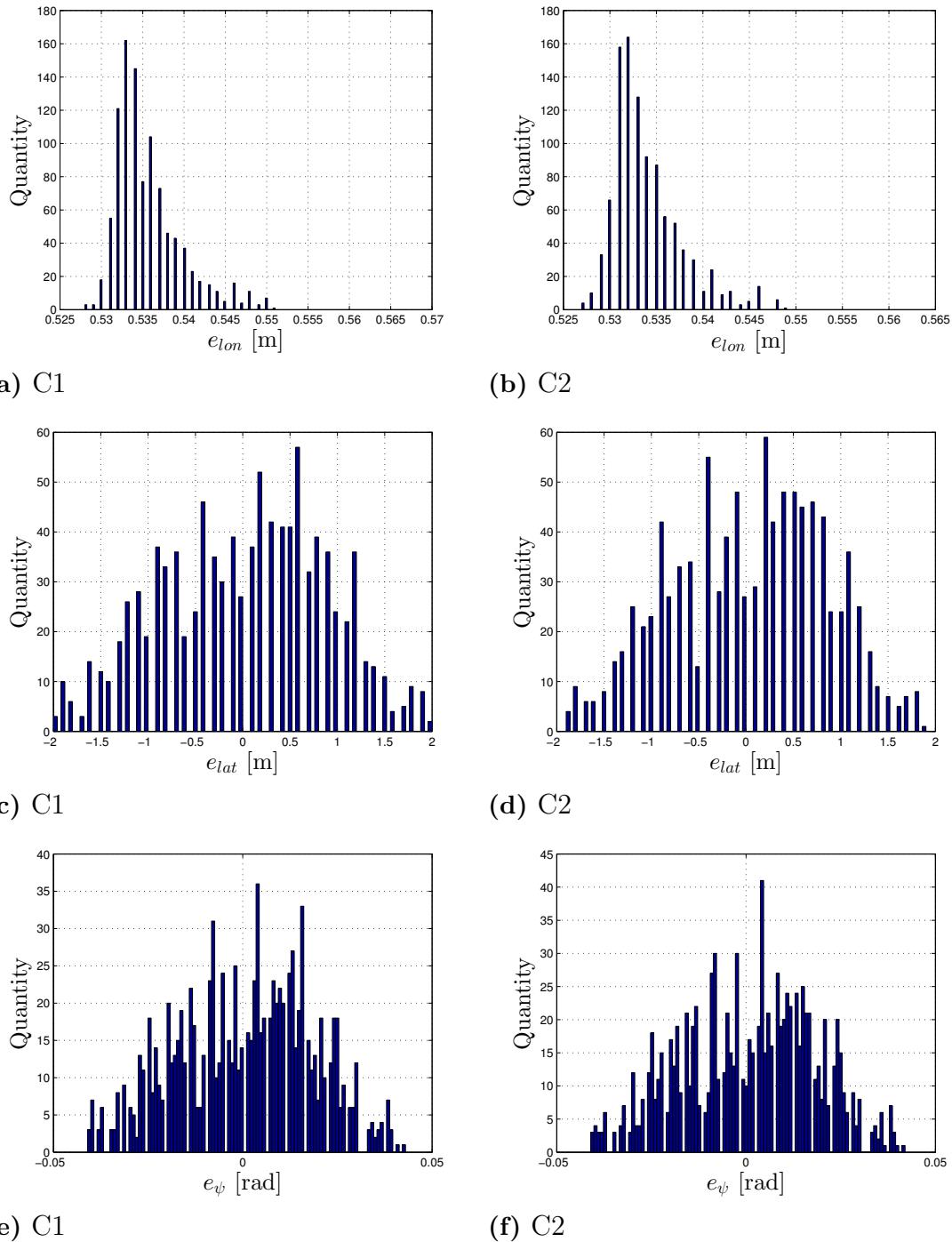


Figure H.1: The distribution of the errors over 1000 runs for case 2

H.2 Case 3

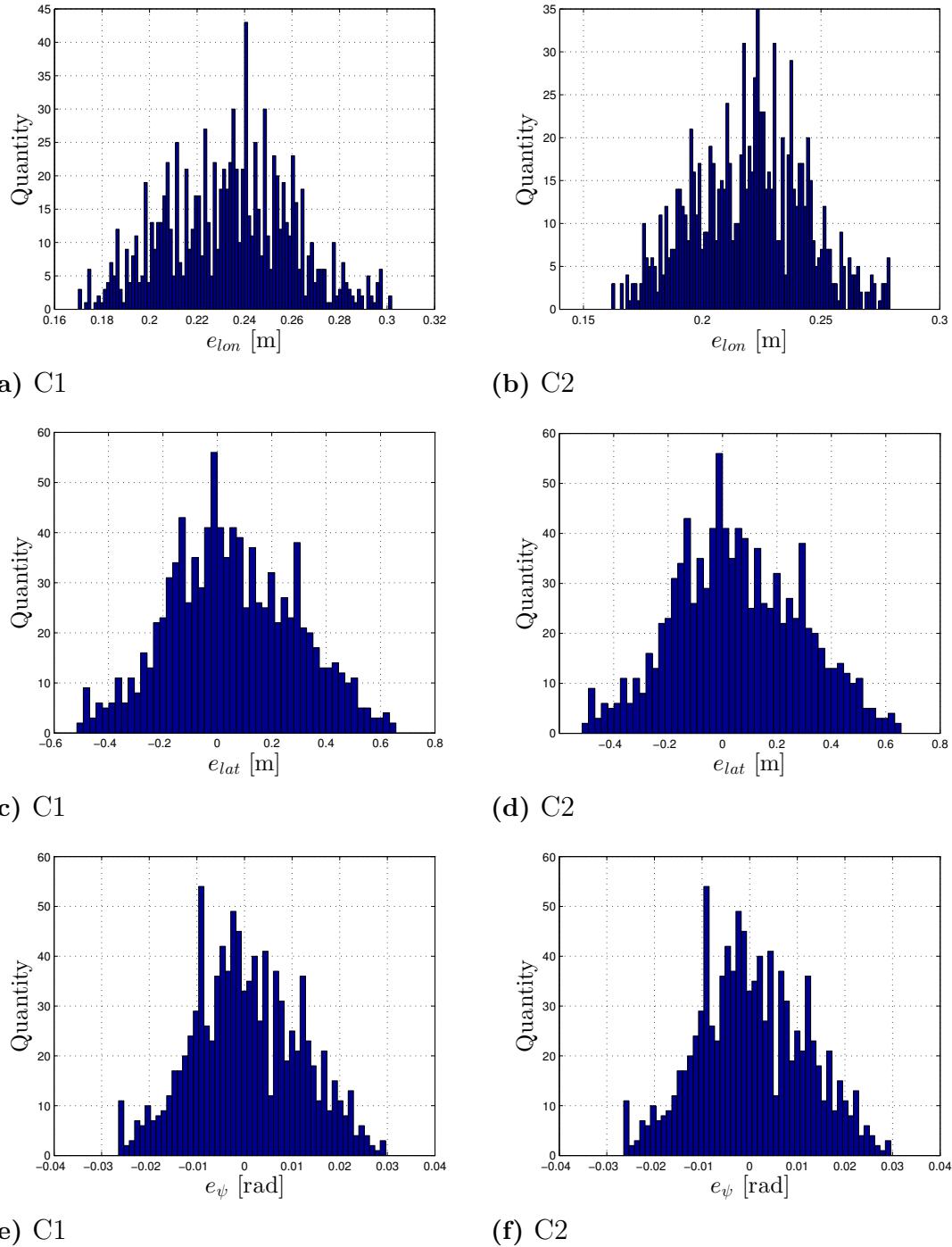


Figure H.2: The distribution of the errors over 1000 runs for case 3

H. Appendix

I

Appendix

Here the state estimates are presented for the filtering process using gathered data when traveling straight downhill and performing a lane change.

I.1 Straight Forward Downhill 20 % 50 [km/h]

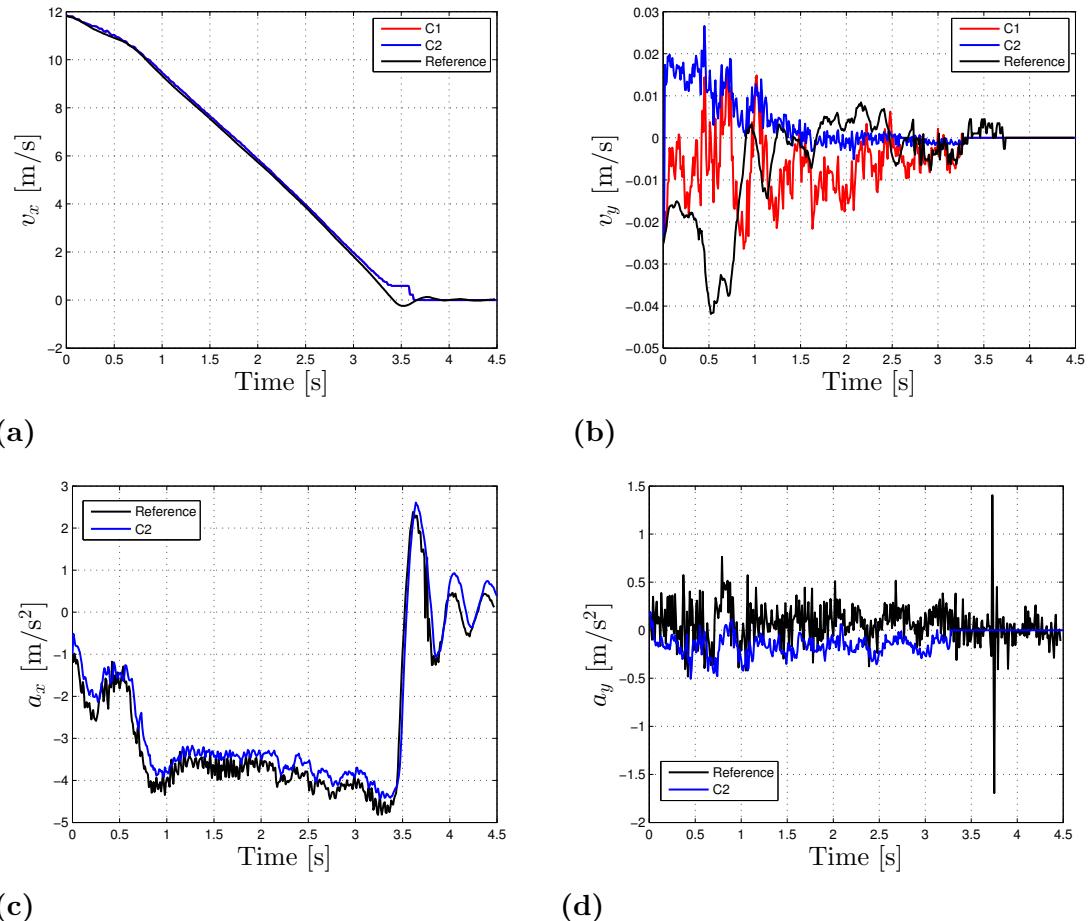


Figure I.1: Estimated velocity and acceleration states traveling straight forward downhill

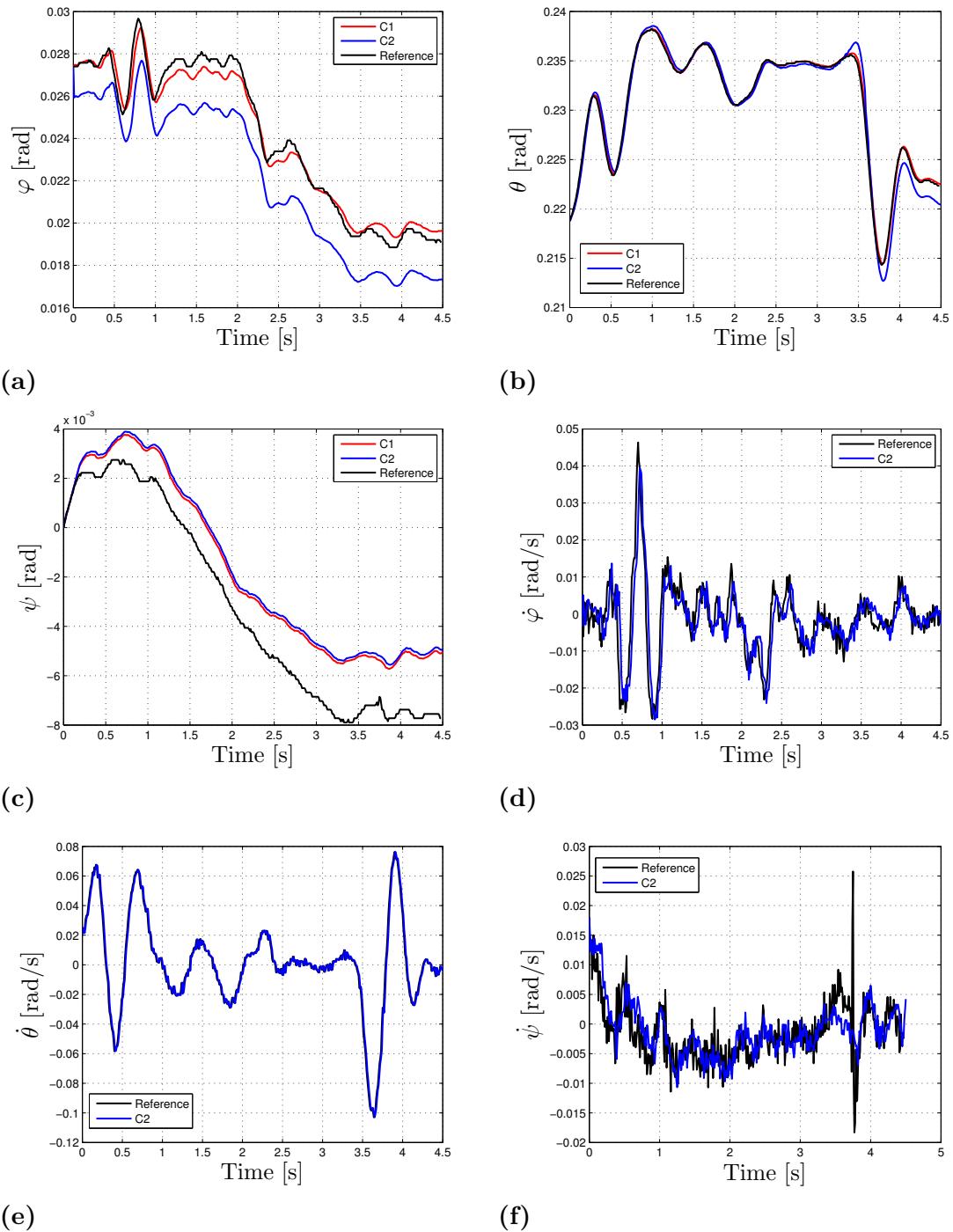


Figure I.2: Estimated Euler angles and their rates traveling straight forward down-hills

I.2 Lane Change 90 [km/h]

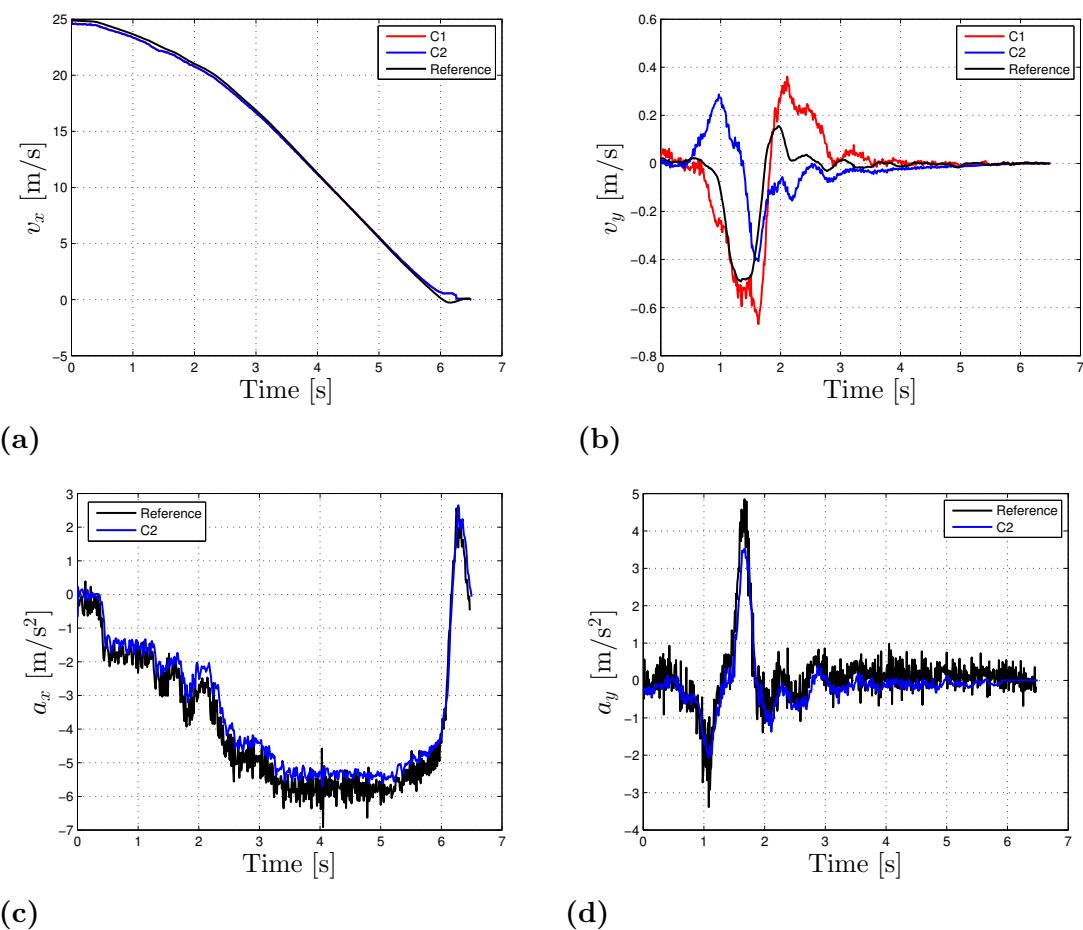


Figure I.3: Estimated velocity and acceleration states during a lane change

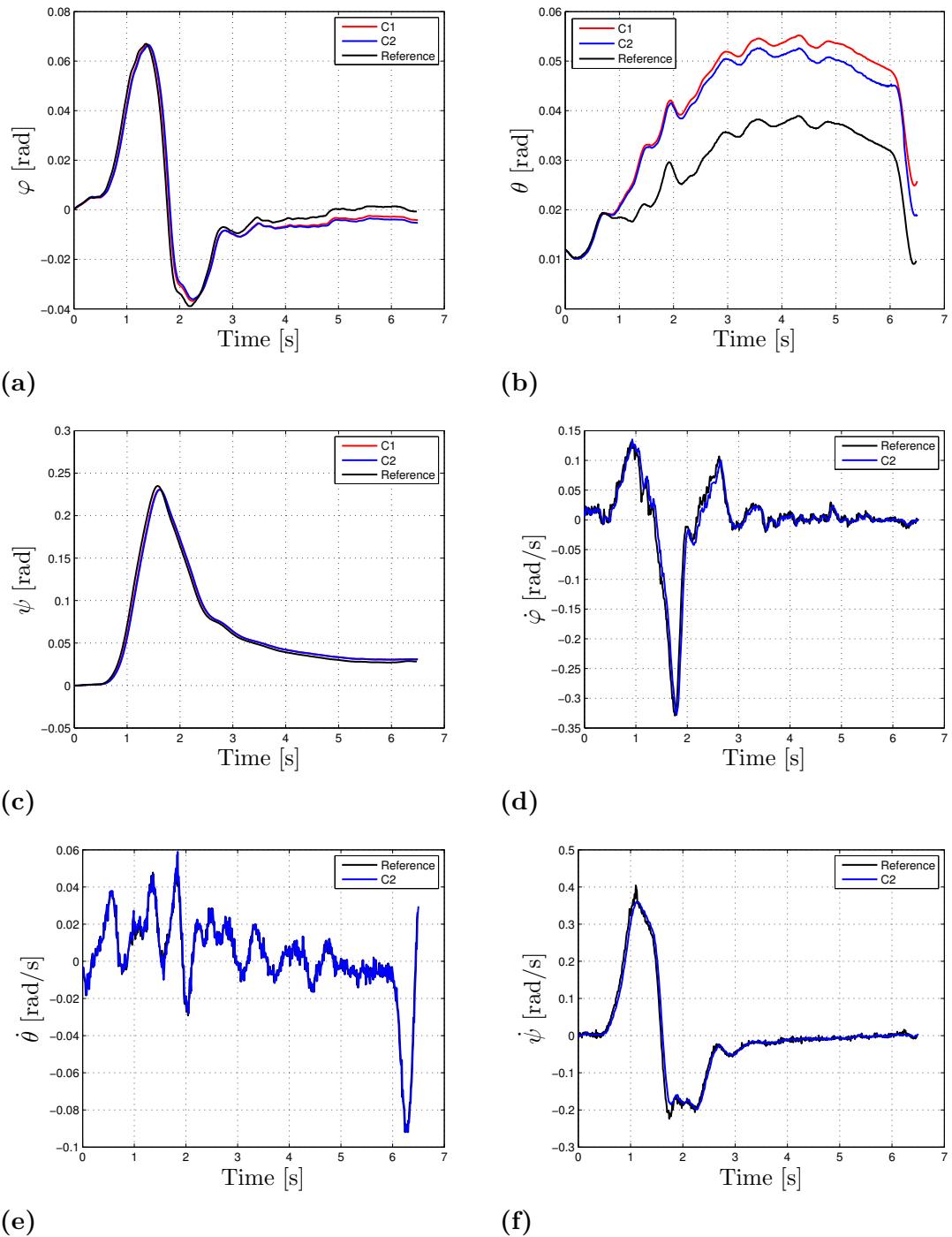


Figure I.4: Estimated Euler angles and their rates during a lane change