CHALMERS
UNIVERSITY OF TECHNOLOGY

Estimation of position and orientation of truck kinematic frames

Master's Thesis in Systems, Control and Mechatronics

EDVIN AGNAS
MARCUS JERENVIK

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis 2016:EX027

Estimation of position and orientation of truck kinematic frames

An investigation of the possibility to estimate the position and orientation of the cabin relative the chassis as well as the position, orientation and trajectory of the chassis in a world frame

EDVIN AGNAS

MARCUS JERENVIK

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and System
Division of Signals Processing and Biomedical Engineering
Signal Processing Group
Chalmers University of Technology
Gothenburg, Sweden 2016

Estimation of position and orientation of truck kinematic frames

EDVIN AGNAS
MARCUS JERENVIK

Supervisor: Anders Karlsson, Volvo Trucks Technology and Signals and System
Examiner: Lennart Svensson, Signals and System

Master's Thesis 2016:EX027
Department of Signals and System
Division of Signals Processing and Biomedical Engineering
Signal Processing Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2016

Estimation of position and orientation of truck kinematic frames
EDVIN AGNAS
MARCUS JERENVIK
Department of Signals and System
Chalmers University of Technology

Abstract

This thesis investigates the possibility to estimate the cabin's position and orientation relative to the chassis as well as the position, orientation and trajectory of the chassis in a world frame. The intention is to enable the usage of cabin mounted sensors such as LIDAR and cameras for localization even if the motion of the cabin differs from the rest of the truck. The thesis clarifies the importance of different sensor combinations and the influence of using constraints in the motion model to connect two vehicle bodies. The estimation process consists of measurements from three GNSS sensors, an embedded velocity sensor and two inertial measurement units (IMU) consisting of gyroscope and accelerometer. These measurements are filtered with a Cubature Kalman filter that is able to handle nonlinear motion and measurement models. These models are based on rigid body dynamics and the filter is used offline. The filter is analyzed with simulated data generated from an advanced truck model provided by Volvo Trucks and with real data gathered from real test scenarios. Results with the accurate simulation model show that it is possible to estimate the position and orientation of both bodies with the mentioned filter. The filter seems to perform well with real data but cannot be verified due to insufficient orientation information from the reference system. It is noticeable that the GNSS sensors struggle with receiving signals when the truck is moving and that it is difficult to select a good prior using real data.

Keywords: Truck, Chassis, Cab, Estimation, Position, Orientation, GNSS, IMU, Gyroscope, Accelerometer, Sensor fusion, Cubature Kalman Filter, Rigid body dynamics.

Acknowledgements

We would first like to thank our great supervisor Anders Karlsson for his commitment and for being supportive in both theoretical and bureaucratic situations. We want to direct a special thanks to Erik Stenborg who has dedicated his own lunch break for helping us with GNSS theory and of course our examiner Lennart Svensson for interesting discussions and theoretical advice.

We would also like to thank the Volvo Group Advanced Technology & Research department for all support and especially the employees Robert, Mike and Thomas in the truck concept lab for our time spent in the garage as well as in the truck.

Last but not least we want to thank our families and girlfriends who have been supporting us and encouraged us through not only this thesis but also our time here at Chalmers, without you it would not have been possible.

Thanks again!

Yours truly, Edvin Agnas & Marcus Jerenvik, Gothenburg, June 2016

Contents

List of Figures

List of Tables

List of Symbols

$\boldsymbol{\chi}$	Unit sigma points
$\hat{\boldsymbol{x}}$	Estimated states
$\boldsymbol{\mathcal{Y}}$	Measurement sigma point
$\boldsymbol{\omega}$	Derivative of roll, pitch and yaw angles
$\boldsymbol{\varepsilon}$	Vector of roll, pitch and yaw angles
$\boldsymbol{\xi}$	State sigma point
$\boldsymbol{A}$	Linear motion matrix
$\boldsymbol{e}_x$	Unit vector in x-direction
$\boldsymbol{e}_y$	Unit vector in y-direction
$\boldsymbol{e}_z$	Unit vector in z-direction
$\boldsymbol{F}$	Force vector
$\boldsymbol{H}$	Linear measurement matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{I}_O$	Inertia matrix
$\boldsymbol{K}_k$	Kalman gain
$\boldsymbol{L}_O$	Angular momentum
$\boldsymbol{M}$	Torque vector
$\boldsymbol{P}$	Covariance matrix
$\boldsymbol{Q}$	Covariance matrix for motion noise
$\boldsymbol{q}$	Motion noise
$\boldsymbol{R}$	Covariance matrix for measurement noise
$\boldsymbol{R}$	Rotation matrix
$\boldsymbol{r}_i$	Vector to point i
$\boldsymbol{r}_{i/j}$	Vector from point j to point i
$\boldsymbol{R}_{zyx}$	Complete rotation matrix in z-y-x system
$\boldsymbol{S}_k$	Innovation covariance matrix
$\boldsymbol{v}_k$	Innovation vector
$\boldsymbol{X}$	State vector
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Measurement vector
δ	Clock bias
Γ	Motion noise matrix
λ	Wave-length
$\mathfrak{q}$	Quaternion
Φ	Carrier phase measurement
ψ	Yaw angle
ρ	Geometric distance

θ	Roll angle
φ	Pitch angle
C	Measurement noise matrix
c	Speed of light in a vacuum inertial frame
d	Distance
f	Frequency
N	Ambiguity term
n	Amount of states
T	Sample time

1
Introduction

This thesis investigates the possibility to estimate the position and orientation of a truck cabin relative to the chassis and the position, orientation and trajectory of the chassis in a world frame to enhance sensor information for autonomous trucks. This chapter contains a problem formulation explaining the importance of this estimation process followed by the objectives, related works, proposed solution and a brief outline of the thesis.

1.1 Problem formulation

It is safe to say that mankind has always striven for a better and more comfortable lifestyle with inventing products that simplifies everyday life. Just look at products such as autonomous lawn mowers or vacuum cleaners that perform household chores without complaining. Nowadays people do not even want to drive their own vehicles anymore. This together with other aspects puts pressure on research teams in the vehicle industries and high demands on sensor performance.

The fundamentals of an autonomous navigation system is to know where the system is positioned, how it is oriented and how it is located to the surroundings. Autonomous vehicles often use cameras, LIDAR and RADAR sensors to create maps over their surroundings and use IMU, GNSS or other inertial sensors to know their orientation and position.

An autonomous truck requires at least the same set of sensors to be able to navigate autonomously, but there is a problem. The truck cabin is suspended from the chassis and differs in motion compared to the actual motion of the truck. This means that it is not preferable to mount sensors on the cabin due to the fact that their information may be misleading. However, there are several advantages of mounting sensors on the cabin. GNSS sensors that are mounted on the cabin roof are able to get a clear sight to surrounding satellites and it is possible to increase the field of view for cameras and LIDAR sensors.

To be able to use information from cabin mounted sensors in precise autonomous navigation, this thesis will investigate the possibility to estimate the position and orientation of the cabin relative to the chassis and the chassis' position, orientation and trajectory in a world frame using GNSS and IMUs sensors together with a velocity sensor. These states will enable fusion of multiple sensors such as cameras,

LIDAR and RADAR sensors mounted on different bodies to enhance the quality in creating maps of the surroundings.

1.2 Objectives

The main objective is to investigate the possibility to determine the position and orientation of a truck cabin relative to the chassis as well as the chassis' position, orientation and trajectory in a world frame. The outcome is important for Volvo Trucks to know if the GNSS sensors are able to provide precise information about the position and orientation of the truck and to enable the usage of cabin mounted sensors for accurate measurements. Following list explains the objectives of the thesis.

1. Determine the orientation and position of the truck cabin relative to the chassis using GNSS-sensors, a velocity sensor and IMU sensors consisting of gyroscopes and accelerometers.
2. Determine the orientation, position and trajectory in the world frame of the chassis using above stated sensors.
3. Evaluate the result with both real and simulated data.
4. Analyze the influence of different sensor combinations.
5. Analyze the outcome of using motion constraints in filtering to connect two bodies.

1.3 Related work

Position and orientation estimation is a common subject in the research for autonomous vehicles. Although there has been a variety of research in this area, no specific theory on how to estimate cabin motions relative to the chassis has been found. Westerlund and Jakobsson Larsson, (2015) [12] have made similar approaches with both GNSS and IMUs to successfully estimate the position of the truck but have ignored the movement of the cabin. Another report that have made similar approaches with both GNSS and IMU is the one from Elisson and Gässler, (2015) [23], though their research was focused to estimate the relative position between two cars. An article written by Stenborg and Hammastrand, (2016) [14] describes the use of GNSS and IMU setup with success and is related to this thesis in the fact that they use carrier phase measurements without base stations to estimate the relative position and orientation, though on a car. Even if these reports are not directly applicable, their results are very important and have inspired the work in this thesis.

Another important aspect in this thesis is to create a motion model for the system. Truck dynamics is fortunately a well researched area and there are a lot of information about these kinds of vehicle models. A report whose findings is not related to this thesis in the sense of positioning, is written by Ibrahim, (2004) [13], but provides useful information about how to set up an accurate dynamical truck model.

1.4 Proposed solution

Three GNSS sensors, two gyroscopes, two accelerometers and one velocity sensor is the proposed sensor setup for this thesis. The GNSS sensors together with one set of gyroscope and accelerometer will be mounted on the cabin and the others on the chassis. These sensors will be fused together by filtering the measured data with a Kalman filter in ambition to provide information about position and orientation of both chassis and cabin. The filter will be tested with different motion models and different combinations of sensors. The filter will then be evaluated with both simulated and real data.

Simulated vehicle measurement data such as position, orientation, speed and acceleration of the bodies will be generated with the help of an advanced vehicle model from Volvo Trucks' own Simulink toolbox called *Virtual Transport Model*, VTM.

The simulation results will be used to investigate if the proposed solution is feasible while the real result will show the performance of the sensors and if this is achievable with real data. The results from the real data will then be validated with a reference system called VBOX.

1.5 Thesis outline

The thesis starts with a system overview that describes the sensor setup, the simulation environment and how data are simulated. This is followed by a theory chapter that describes the theoretical background required to understand all details of the thesis. The next chapter, filter design, describes how the theories are applied to create filters where the result chapter evaluates the filters that are created with data generated from the simulation environment and from scenarios using real hardware. The thesis ends with a discussion of the results and a short conclusion of the stated objectives.

2
System overview

The system overview chapter declares how the whole system intends to work, in terms of sensors, how they are used together, how to gather data from them and how they are simulated. The chapter starts with the system setup and follows with describing the hardware components and concludes with a description over the simulation environment.

The sensor setup (see Figure 2.1) consists of three GNSS sensors, two inertial measurement units (IMU), equipped with gyroscopes and accelerometers and the speed sensor of the Truck. Other sensors that are used for this thesis but not in the actual filter is a reference system called VBOX that is used to validate the estimated states.

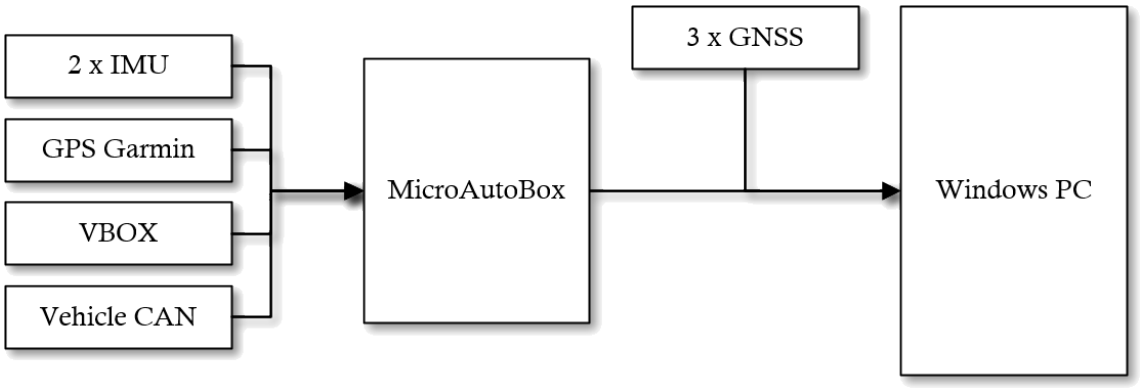
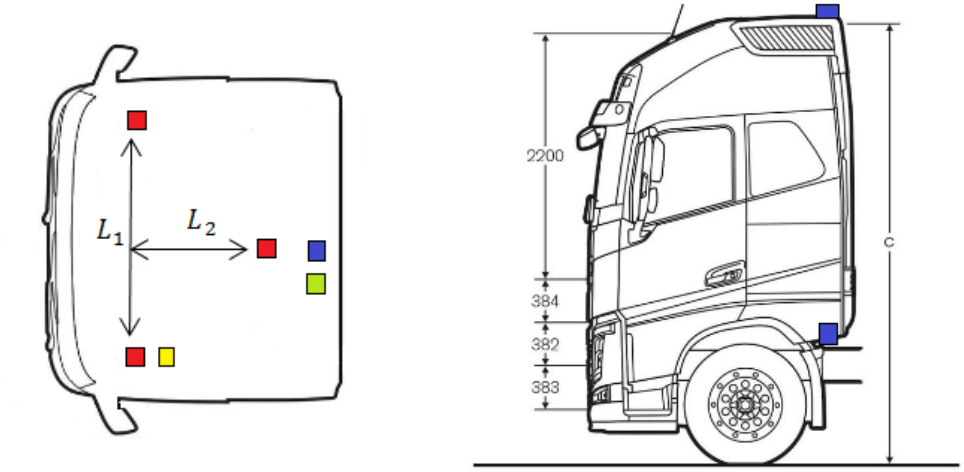


Figure 2.1: A system overview illustrating the setup of the sensors.

The GNSS antennas are placed on the roof of the cabin in a triangle, two antennas in the front and one in the back (see Figure 2.2), with the intention to measure both position and orientation. The data is transferred from the antennas through an evaluation unit that outputs raw GNSS data with serial communication to a computer. This data is processed with bit conversion in the computer to provide correct information.

The IMUs are placed at two different positions on the truck. One on the cabin roof, centered in the back and one on the chassis, centered as close to the front axle as possible. The IMU data does not provide any information about when the signal was transmitted and requires being timestamped with the same time base as the GNSS sensors. This is done by using a system called MicroAutobox together with a Garmin GPS sensor where the IMU signals are transferred with CAN communication to the system and then timestamped with the GPS sensor.



(a) Truck displayed from above. (■ CPAC IMU, ■ GNSS sensors, ■ Garmin GPS, ■ VBOX IMU) $L_1 = 1.78\ [m],\ L_2 = 1.00\ [m]$

(b) Truck displayed from the side. (■ CPAC IMU)

Figure 2.2: A description over sensor positions.

2.1 Hardware components

This section intends to describe each component briefly to provide an understanding in which kind of components that are used and why they are used.

2.1.1 U-Blox GNSS antennas

The U-Blox Global Navigation Satellite System (GNSS) sensors are affordable sensors that are able to gather satellite information from most satellite systems orbiting the earth. U-Blox has a lot of different GNSS sensors but the ones used in this thesis are called *EVK-M8T*. Each sensor contains one active antenna with 3 m cable, and an evaluation kit including a GNSS time module that are used to provide GNSS data [21]. These sensors are used to provide information about the position of the truck and hopefully the orientation with the proposed mounting plan.

Figure 2.3: Image of a U-blox GNSS sensor [21].

2.1.2 Inertial measurement unit - IMU

Inertial measurement units (IMU) are able to provide information about inertial motion of the bodies they are attached to, where the information depends on which sensors the IMUs are equipped with. A common setup is gyroscopes, accelerometers and magnetometers that are able to measure angular velocity, translational acceleration and surrounding magnetic fields. The IMUs used in this thesis are prototypes delivered by *CPAC Systems* and are able to provide angular velocity and translational acceleration every 4 ms of the body they are mounted to.

The integrated gyroscope is a three-axis vibratory MEMS-gyroscope, which means that it has three built in MEMS-gyroscopes to provide rotation rate about each axis in three dimensions. The integrated accelerometer is a three axis MEMS-accelerometer containing three separated proof masses used to provide acceleration information in each axis [19].

2.1.3 MicroAutobox

The microAutobox II from *dSPACE* is what the supplier calls a prototyping system or a "Real-time system for performing fast function prototyping" and can be used for several applications for example testing control algorithms [28]. The system is used as a central unit in this thesis, meaning that except for timestamping the IMU data it is used to log data from the reference system and from integrated truck sensors.

Figure 2.4: Image of a MicroAutobox II [28].

2.1.4 Garmin GPS

Garmin is a known supplier of navigation solutions and the GPS receiver that is used in this thesis is of type *GPS 18x LVC*. This product consists of an antenna with an embedded receiver that is able to transmit information needed for navigation and timestamping. The unit provides a highly accurate pulse-per-second (PPS) aligned to the start of each GPS second with a following message containing the time the pulse occurred [20]. This PPS signal together with the mentioned message are used for timestamping the IMU signals.

Figure 2.5: Image of a Garmin GPS [20].

2.1.5 Reference system - VBOX

VBOX systems from *Racelogics* are actually a variety of different products used for automotive testing. The reference system that is used in this thesis is a VBOX 3i Dual Antenna system together with a RTK DGPS base station and a VBOX IMU. The dual antennas are able to provide information about for example yaw rate, true heading and pitch or roll angle, depending on how the antennas are mounted. If this system is added with a VBOX IMU it could provide additional information of both pitch and roll angle and improve the yaw rate measurements. The accuracy of the VBOX system with GNSS dual antenna, base station and VBOX IMU is shown in Table 2.1 [25].

Figure 2.6: Image of a VBOX reference system [25].

Table 2.1: VBOX accuracy with GNSS dual antenna, base station and VBOX IMU.

	Accuracy
Absolute position	2 cm
Heading	0.06° RMS
Roll angle	$< 0.047°$ RMS
Pitch angle	0.06° RMS

2.2 Simulation environment

A simulation environment is often used to evaluate the performance of a system before it is tested with real data and preferable when testing the abilities of a system to determine if a proposed solution is feasible. This thesis makes use of an advanced truck model provided by Volvo Trucks from their own *Simulink* library called *Virtual Transport Model* (VTM). The vehicle model that is used is similar to the real truck. This section describes how this vehicle model is used to generate simulated reference states and simulated measurement values for the GNSS, IMU and velocity sensors.

This truck model is used to provide information of the truck motion in terms of position, velocity, acceleration, angle, angular velocity and angular acceleration for both the cabin and the chassis. The position of the truck is also simulated to be on the surface of the earth and in the global coordinate frame ECEF by using a real GNSS position measurement. This information is used as references.

The simulated measurements for the IMUs are generated from the local angular velocity and the local acceleration of the points where the IMUs are attached. These data are then added with normal distributed random values with the same variance as the real sensors to match the real measurements as much as possible. The velocity measurements are generated with the same approach but with data provided by the norm of the chassis' velocity.

The GNSS measurements are generated a bit differently because these measurements represent the distance between receiver and satellites. To be as accurate as possible, real ephemeris data from satellites are being used to simulate the position and orbits of the satellites. The position of these satellites is then used to simulate the distance between satellites to three fixed points on the cabin roof, one for each sensor position. This data are then added with simulated receiver drifts, explanation about what this is can be found in the theory chapter, gathered from a real measurement and normal distributed random values with the same variance as a real sensor.

3
Theory

The theory chapter includes useful information needed to understand certain theoretical parts of the thesis. It starts with theory about coordinate frames followed by rotations and dynamical systems and continues with theory about sensors and concludes with sensor fusion theory.

3.1 Coordinate frames

An object's position can be described in many different ways, with various approaches and precisions. A coordinate frame that specifies the position with a value along a specific axis is the most common method for positioning in science. Such coordinate frame is often used in GNSS-theory and this section will clarify two types that are being used in this thesis.

3.1.1 Earth-centered, Earth-fixed - ECEF

The Earth-Centered, Earth-Fixed (ECEF) coordinate frame (see Figure 3.1) is as the name declare, an Earth fixed frame with the origin at the center of the ellipsoid modeling the Earth surface, approximately Earth's center of mass. The z-axis is direct along Earth's rotation axis, the x-axis to the IERS Reference Meridian (IRM) and the y-axis according to the right-handed orthogonal set [1].

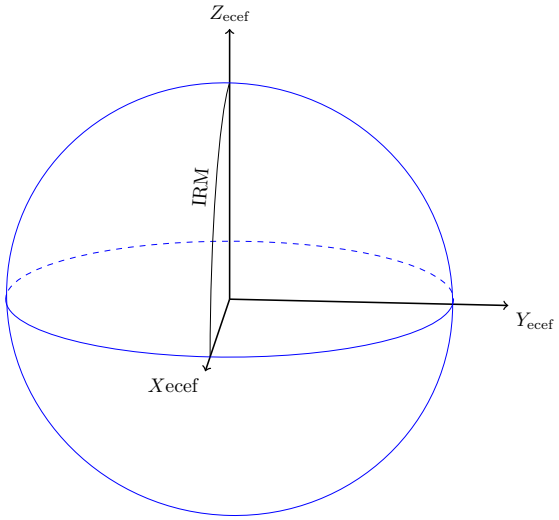


Figure 3.1: Earth-Centered, Earth-Fixed (ECEF) coordinate frame.

3.1.2 East, north, up - ENU

The East, North, Up (ENU) coordinate frame is a local navigation frame that depends on a predefined reference position (see Figure 3.2). The system is based on an orthogonal plane at the reference position with the y-axis pointing towards north. This coordinate frame is often used at small limited areas, where the surface of Earth can be approximated to a tangent plane [1].

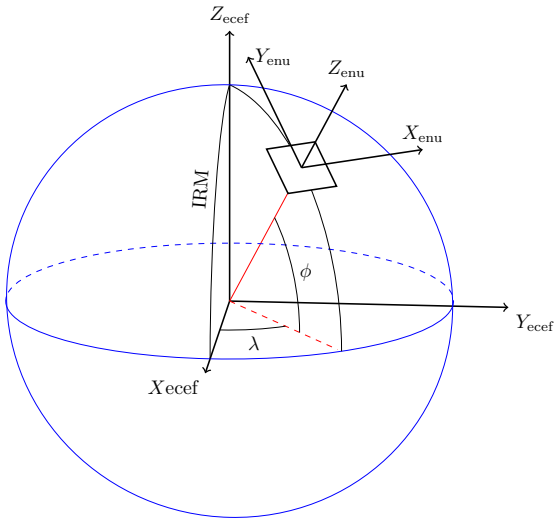


Figure 3.2: East, North, Up (ENU) coordinate frame.

3.1.3 Transformation between ECEF and ENU

The local ENU coordinate frame is preferable to use when dealing with objects on the ground. The transformation between the ECEF to the ENU coordinate frame is sometimes useful since some objects are described in the ECEF coordinate frame.

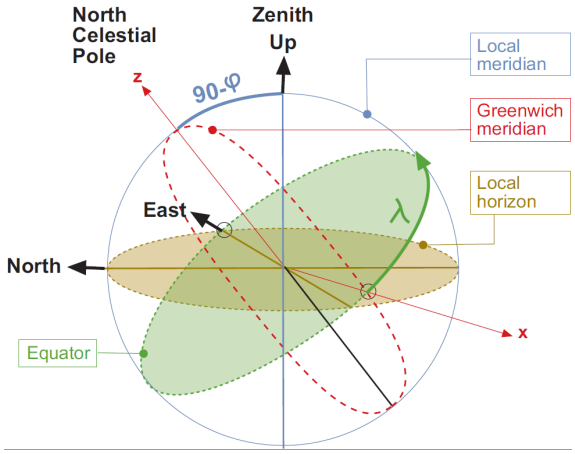


Figure 3.3: Transforming between ECEF to ENU [24].

Figure 3.3 illustrate that two rotations are required to switch between the frames, namely the x- and z- axis. Rotation of $90 - \varphi$ around x-axis and $90 + \lambda$ around

z-axis, where longitudinal coordinate φ and latitudinal coordinate λ are ellipsoidal coordinates.

The transformation equation between ENU and ECEF are

$$\begin{bmatrix} X_{ecef} \\ Y_{ecef} \\ Z_{ecef} \end{bmatrix} = \boldsymbol{R}_z(-(\frac{\pi}{2} + \lambda))\boldsymbol{R}_x(-(\frac{\pi}{2} - \varphi)) \begin{bmatrix} X_{enu} \\ Y_{enu} \\ Z_{enu} \end{bmatrix}, \tag{3.1}$$

$$\begin{bmatrix} X_{enu} \\ Y_{enu} \\ Z_{enu} \end{bmatrix} = \boldsymbol{R}_x(\frac{\pi}{2} - \varphi))\boldsymbol{R}_z(\frac{\pi}{2} + \lambda) \begin{bmatrix} X_{ecef} \\ Y_{ecef} \\ Z_{ecef} \end{bmatrix}, \tag{3.2}$$

where the rotation matrices $\boldsymbol{R}_x$ and $\boldsymbol{R}_z$ are described in the next section.

3.2 Rotations

To be able to determine the position and orientation of an object in three dimensions it is important to know how rotations are defined. The two most common ways is to either express rotation with Euler angles, that represents rotation around three consecutive axes or with Quaternions, rotation with complex numbers. The Euler angle approach is more intuitive than Quaternions but the Euler angles are always at risk of losing one degree of freedom that occur when the angels are at values that makes two Euler axes coincide, a singularity event often stressed as Gimbal lock [15]. It is therefore important to understand the motions of the object before determine which approach to represent rotations. This thesis will make use of both Euler angles and Quaternions and the following sections describe common basics of these representations, how to rotate between coordinate frames and how to differentiate rotations.

3.2.1 Euler angles

As mentioned earlier, Euler angles are angles that describe rotations around three consecutive axes

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{3.3}$$

$$\boldsymbol{R}_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix}, \tag{3.4}$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

The order of these axes is important because two different sequences results in two different orientations, a rule is that two consecutive rotations should never rotate

around the same axis, hence it only counts as one rotation. The most common sequence in vehicles and aircrafts are the z-y-x rotation or often referred as the Roll-Pitch-Yaw rotation

$$\boldsymbol{R}_{zyx} = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\varphi)\boldsymbol{R}_x(\theta), \tag{3.6}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{R}_{zyx} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \tag{3.7}$$

where the inverse and the transpose ($\boldsymbol{R}^{-1} = \boldsymbol{R}^T$) reverse the transformation

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \boldsymbol{R}_{zyx}^T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{3.8}$$

This is often described in a body fixed coordinate frames where the roll angle is expressed as θ, pitch as φ and yaw as ψ, collected to $\boldsymbol{\varepsilon} = [\theta \ \varphi \ \psi]^T$. These three angles, θ, φ and ψ are preferable to use not only in rotation matrices to transform between different coordinate frames but also to describe the orientation of a vehicle in three dimensions.

3.2.2 Quaternions

Quaternions are another way of describing rotations, a rather complex representation but commonly used in situations where *Gimbal lock* is a possibility for Euler angles. This report will not explain quaternions in detail but rather how to use them for orientation purpose. A quaternion can be described as following

$$\mathfrak{q} = \begin{bmatrix} \mathfrak{q}_0 & \mathfrak{q}_1 & \mathfrak{q}_2 & \mathfrak{q}_3 \end{bmatrix}^T, \tag{3.9}$$

where the first component $\mathfrak{q}_0$ is the scalar real part and $\mathfrak{q}_1, \mathfrak{q}_2$ and $\mathfrak{q}_3$ is the imaginary vector part of the quaternion

$$\mathfrak{q} = \mathfrak{q}_0 + i\mathfrak{q}_1 + j\mathfrak{q}_2 + k\mathfrak{q}_3 \tag{3.10}$$

and the norm of the quaternion is equal to one

$$\|\mathfrak{q}\| = \sqrt{\mathfrak{q}_0^2 + \mathfrak{q}_1^2 + \mathfrak{q}_2^2 + \mathfrak{q}_3^2} = 1. \tag{3.11}$$

There are some theory that are useful when estimating the orientation of a body, for example how to rotate between different coordinate frames and how to express the quaternion derivative.

A rotation matrix can be derived by converting the unit quaternion[8][11] as

$$\boldsymbol{R}_{zyx} = \mathfrak{Q}(\mathfrak{q}) = \begin{bmatrix} 2\mathfrak{q}_0^2 - 1 + 2\mathfrak{q}_1^2 & 2\mathfrak{q}_1\mathfrak{q}_2 - 2\mathfrak{q}_0\mathfrak{q}_3 & 2\mathfrak{q}_1\mathfrak{q}_3 + 2\mathfrak{q}_0\mathfrak{q}_2 \\ 2\mathfrak{q}_1\mathfrak{q}_2 + 2\mathfrak{q}_0\mathfrak{q}_3 & 2\mathfrak{q}_0^2 - 1 + 2\mathfrak{q}_2^2 & 2\mathfrak{q}_2\mathfrak{q}_3 - 2\mathfrak{q}_0\mathfrak{q}_1 \\ 2\mathfrak{q}_1\mathfrak{q}_3 - 2\mathfrak{q}_0\mathfrak{q}_2 & 2\mathfrak{q}_2\mathfrak{q}_3 + 2\mathfrak{q}_0\mathfrak{q}_1 & 2\mathfrak{q}_0^2 - 1 + 2\mathfrak{q}_3^2 \end{bmatrix}, \tag{3.12}$$

and it is also possible to reverse the operation by using a rotation matrix to derive the quaternions[18]

$$\mathfrak{q} = \mathfrak{Q}^{-1}(\boldsymbol{R}_{zyx}) = \begin{bmatrix} \frac{1}{2}\sqrt{R_{11}+R_{22}+R_{33}+1} \\ (R_{23}-R_{32})/4(\frac{1}{2}\sqrt{R_{11}+R_{22}+R_{33}+1}) \\ (R_{31}-R_{13})/4(\frac{1}{2}\sqrt{R_{11}+R_{22}+R_{33}+1}) \\ (R_{12}-R_{21})/4(\frac{1}{2}\sqrt{R_{11}+R_{22}+R_{33}+1}) \end{bmatrix}, \tag{3.13}$$

where R_{ij} represent the value at row i, column j in the rotation matrix $\boldsymbol{R}_{zyx}$. It is often useful to derive the time derivative of a quaternion that is the quaternion rates which are related to the angular rate

$$\dot{\mathfrak{q}} = \frac{1}{2}\boldsymbol{S}(\boldsymbol{\omega})\mathfrak{q} = \frac{1}{2}\bar{\boldsymbol{S}}(\mathfrak{q})\boldsymbol{\omega}, \tag{3.14}$$

where

$$\boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}, \ \bar{\boldsymbol{S}}(\mathfrak{q}) = \begin{bmatrix} -\mathfrak{q}_1 & -\mathfrak{q}_2 & -\mathfrak{q}_3 \\ \mathfrak{q}_0 & -\mathfrak{q}_3 & \mathfrak{q}_2 \\ \mathfrak{q}_3 & \mathfrak{q}_0 & -\mathfrak{q}_1 \\ -\mathfrak{q}_2 & \mathfrak{q}_1 & \mathfrak{q}_0 \end{bmatrix}, \tag{3.15}$$

and $\boldsymbol{\omega}$ is the angular rate along each axis. The relation between the quaternion rate and the angular rate allows translations between these descriptions.

3.2.3 Derivative of vector in rotating frames

The derivative of rotating coordinate frames is not as simple as for translational derivatives, since the change of the direction of unit vectors has to be taken into account. Let the transformation of a constant vector to another coordinate frame be investigated [16]

$$\boldsymbol{r}_P = \boldsymbol{R}_{zyx}\boldsymbol{r}_{P/O}. \tag{3.16}$$

The angles θ_{x_1}, θ_{y_1} and θ_{z_1} denotes the rotations to respective axis in the rotation matrix $\boldsymbol{R}_{zyx}$ and the derivative of equation (3.16) becomes

$$\dot{\boldsymbol{r}}_{P/O} = \boldsymbol{\omega} \times \boldsymbol{r}_{P/O}, \tag{3.17}$$

where $\boldsymbol{\omega} = [\dot{\theta}_{x_1}\dot{\theta}_{y_1}\dot{\theta}_{z_1}]^T$. This is true for any constant vector in a rotating coordinate frame, that is the vector does not need to be a position

$$\dot{\boldsymbol{b}} = \boldsymbol{\omega} \times \boldsymbol{b}. \tag{3.18}$$

This can also be used to find the derivative of a time dependent vector

$$\dot{\boldsymbol{b}} = \frac{d}{dt}(b_x\boldsymbol{e}_x + b_y\boldsymbol{e}_y + b_z\boldsymbol{e}_z), \tag{3.19}$$

since the unit vectors are constant equation (3.18) can be applied

$$\dot{\boldsymbol{b}} = \dot{b}_x\boldsymbol{e}_x + \dot{b}_y\boldsymbol{e}_y + \dot{b}_z\boldsymbol{e}_z + b_x\boldsymbol{\omega} \times \boldsymbol{e}_x + b_y\boldsymbol{\omega} \times \boldsymbol{e}_y + b_z\boldsymbol{\omega} \times \boldsymbol{e}_z \tag{3.20}$$

$$= \frac{d\boldsymbol{b}}{dt} + \boldsymbol{\omega} \times \boldsymbol{b} \tag{3.21}$$

and the derivative for the time dependent vector becomes

$$\dot{\boldsymbol{b}} = \frac{d\boldsymbol{b}}{dt} + \boldsymbol{\omega} \times \boldsymbol{b}. \tag{3.22}$$

This can be used to derive the relative motion of a rigid body.

3.3 Rigid body dynamics - Relative motion

The dynamics of a body is often used to improve the performance of filtering. The motion of a body can be derived from the body dynamics, due to forces and torques, and a prediction of the next state can be done in the filter. The dynamics is divided into two main parts, kinematics that deals with motion constraints and kinetics that concern the influence of forces and torques.

This section is mainly inspired by Boström (2015)[16] and treat equations to determine the motion of a rigid body. The section starts with a description of general equations of motion and follows with explaining relative motion of rotating and translating bodies.

3.3.1 General equations of motion

The general equation of motions for a rigid body can simply be derived from Newton's second law and the law of angular momentum. A rigid body can be seen as a quantity of particles with the total mass m. The translation motion is equal to the sum of the forces

$$\sum \boldsymbol{F} = m\boldsymbol{a} \tag{3.23}$$

and the rotation is described by a sum of torques,

$$\sum \boldsymbol{M}_O = \dot{\boldsymbol{L}}_O, \tag{3.24}$$

where $\boldsymbol{L}_O$ is the angular momentum of the particles about a the fixed point O,

$$\boldsymbol{L}_O = \sum_{i=0}^{N} \boldsymbol{r}_{i/O} \times m_i \boldsymbol{v}_i. \tag{3.25}$$

If the sum is modified to an integral with infinitely many particles with small mass dm

$$\boldsymbol{L}_O = \int \boldsymbol{r} \times \boldsymbol{v} dm, \tag{3.26}$$

together with the property that $\boldsymbol{v}$ can be reduced to $\boldsymbol{\omega} \times \boldsymbol{r}$, $\boldsymbol{L}_O$ can be formulated as

$$\boldsymbol{L}_O = \int \boldsymbol{r} \times \boldsymbol{\omega} \times \boldsymbol{r} dm. \tag{3.27}$$

The angular momentum is rewritten with inertia matrix $\boldsymbol{I}_O$ as

$$\boldsymbol{L}_O = \boldsymbol{I}_O \boldsymbol{\omega}, \tag{3.28}$$

$$\boldsymbol{I}_O = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}. \tag{3.29}$$

It is always good to keep in mind that a rotation coordinate frame affects the derivative according to equation (3.22)

$$\dot{\boldsymbol{L}}_O = \frac{d\boldsymbol{L}_O}{dt} + \boldsymbol{\Omega} \times \boldsymbol{L}_O. \tag{3.30}$$

Other derived laws that may come in handy for dynamical systems are springs and dampers

Linear springs

$$F = k_p \bar{x}, \tag{3.31}$$

Linear dampers

$$F = c_p \dot{\bar{x}}, \tag{3.32}$$

Torsion springs

$$M = k_\varepsilon \bar{\varepsilon}, \tag{3.33}$$

Torsion dampers

$$M = c_\varepsilon \dot{\bar{\varepsilon}}, \tag{3.34}$$

where k_p, k_ε denotes the spring constants, c_p, c_ε denotes the damping constants, $\bar{x}$ is the spring elongation and $\bar{\varepsilon}$ is the rotation angle.

3.3.2 Relative angular velocity and angular acceleration

The relative angular rate and angular acceleration of a rigid body are truly important for the description of the motion. The ability to describe angular motion in different correlated coordinate frames, where a rotation in one frame can affect another, are useful for both motion models and measurement models when designing the filter.

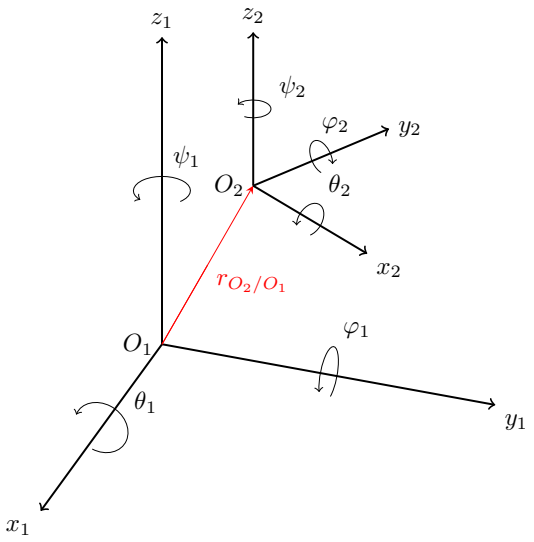


Figure 3.4: Rotating coordinate frames [29].

The relative angular rate of a rotating coordinate frame defined in another rotating coordinate frame, is simply the sum of the rates

$$\begin{aligned}\boldsymbol{\omega} &= \dot{\theta}_1 \boldsymbol{e}_{x_1} + \dot{\varphi}_1 \boldsymbol{e}_{y_1} + \dot{\psi}_1 \boldsymbol{e}_{z_1} + \dot{\theta}_2 \boldsymbol{e}_{x_2} + \dot{\varphi}_2 \boldsymbol{e}_{y_2} + \dot{\psi}_2 \boldsymbol{e}_{z_2} \\ &= \boldsymbol{\omega}_1 + \boldsymbol{R}_{zyx} \boldsymbol{\omega}_2 \\ &= \boldsymbol{R}_{zyx}^T \boldsymbol{\omega}_1 + \boldsymbol{\omega}_2.\end{aligned} \tag{3.35}$$

If rotations are defined in the same coordinate frame the angular rate simply becomes

$$\boldsymbol{\omega} = \boldsymbol{\omega}_1 + \boldsymbol{\omega}_2. \tag{3.36}$$

The angular acceleration gets slightly more complicated if the angles are defined in a rotation coordinate frame (with angular rate $\boldsymbol{\Omega}$), as discussed in Section 3.2.3. The angular acceleration is defined as the time derivative of the angular rate

$$\boldsymbol{\alpha} = \dot{\boldsymbol{\omega}} = \frac{d}{dt}(\omega_x \boldsymbol{e}_x + \omega_y \boldsymbol{e}_y + \omega_z \boldsymbol{e}_z). \tag{3.37}$$

By using equation (3.22) the relative angular acceleration becomes

$$\boldsymbol{\alpha} = \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\Omega} \times \boldsymbol{\omega} \tag{3.38}$$

and if the angles are defined in the same coordinate frame ($\boldsymbol{\Omega} = \boldsymbol{\omega}$)

$$\boldsymbol{\alpha} = \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \boldsymbol{\omega} = \frac{d\boldsymbol{\omega}}{dt}. \tag{3.39}$$

Both the angular rate and angular acceleration are necessary components to describe the relative translational velocities and accelerations of a body, which is further described in next section.

3.3.3 Relative velocity and acceleration

The relative velocity and acceleration of a body can be used in filtering. Let the coordinate frame O' be a translating and rotating frame in the fixed frame O and point P be a moving point in frame O' (see Figure 3.5). The vector can then be described as

$$\boldsymbol{r}_{P/O} = \boldsymbol{r}_{O'/O} + \boldsymbol{r}_{P/O'}. \tag{3.40}$$

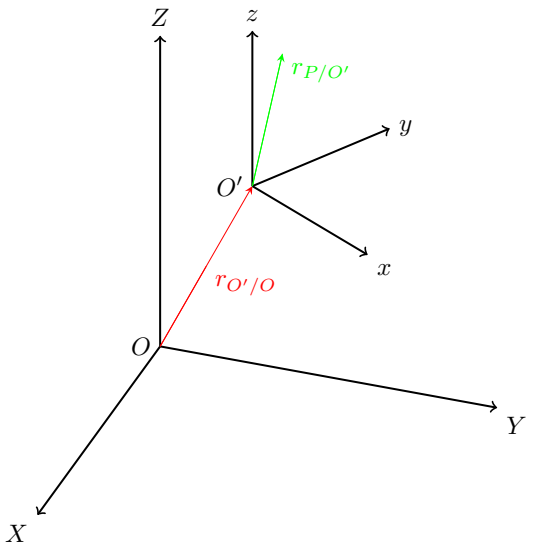


Figure 3.5: Moving point P in translating and rotating coordinate frame.

With the property of differentiating vectors in rotating frames, discussed in 3.2.3, the derivative of $\boldsymbol{r_{P/O}}$ becomes

$$\begin{aligned}\boldsymbol{v}_P = \dot{\boldsymbol{r}}_{P/O} = \boldsymbol{v}_{O'/O} + (\boldsymbol{v}_{O'})_O + \boldsymbol{\omega}_{O'} \times \boldsymbol{r}_{O'/O} \\ + (\boldsymbol{v}_P)_{O'} + \boldsymbol{\omega}_P \times \boldsymbol{r}_{P/O'},\end{aligned} \tag{3.41}$$

and the acceleration is

$$\begin{aligned}\boldsymbol{a}_P = \dot{\boldsymbol{v}}_P = \boldsymbol{a}_{O'/O} + (\boldsymbol{a}_{O'})_O + \boldsymbol{\alpha}_{O'} \times \boldsymbol{r}_{O'/O} \\ + 2\boldsymbol{\omega}_{O'} \times (\boldsymbol{v}_{O'})_O + \boldsymbol{\omega}_{O'} \times (\boldsymbol{\omega}_{O'} \times \boldsymbol{r}_{O'/O})\end{aligned} \tag{3.42}$$

where $(\boldsymbol{v}_{O'})_O$ and $(\boldsymbol{a}_{O'})_O$ are defined as the velocity and acceleration of the O' coordinate frame in the O coordinate frame.

3.4 Global navigation satellite systems - GNSS

Global Navigation Satellite Systems (GNSS) is the generic name for navigation systems that use orbiting satellites to determine the position of an object placed on

earth. Well known systems, as the American *NAVSTAR GPS*, more commonly known as *GPS*, the Russian *GLONASS* as well as the Chinese *Beidou* and the European *Galileo* are all examples of Global Navigation Satellite Systems.

There are several advantages using satellite information from all kinds of system. More areas of the earth can be covered due to different orbits of satellites and increasing amount of available satellites leads to higher precision in positioning. The downside is that the protocols differ depending on which system that is used. For example some systems provide satellite position directly when some only provides variables to calculate the position and predict the orbit of the satellites. However, the basics of positioning are the same for all kinds of system and will be described in Section 3.4.1.

3.4.1 Positioning

The principles of positioning or more accurately, signal-based positioning is the same for all kinds of satellite systems. The basics is to measure the distance between a satellite with a known position and a receiver whose position is to be determined. This distance is calculated by taking the difference in time between a transmitted signal and a received signal and multiplying it with the speed of the signal, namely the speed of light, c [10],

$$d = \Delta t \; c. \tag{3.43}$$

If an object's position should be determined in one dimension, equation (3.43) would be enough because one unknown variable is solvable with one equation. However, this is only true if the satellite and the receiver clock are synchronized. This would require a very accurate clock, for example an atomic clock, in the receiver and it is not economically viable. Another solution is to add measurements from another satellite with known position to calculate the difference between the satellite clock and the receiver clock [10]. A simple example to explain this can be seen in Figure 3.6.

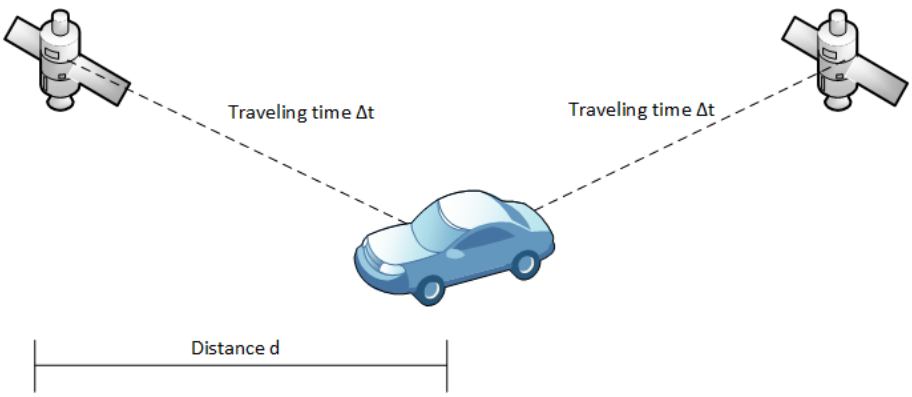


Figure 3.6: Concept of positioning in one dimension with two satellites and one receiver.

The previous example is only for positioning in one dimension, this means that $n+1$ satellites are required for n dimensions, meaning that four satellites are required for positioning in three dimension, one for each dimension and one for time [10].

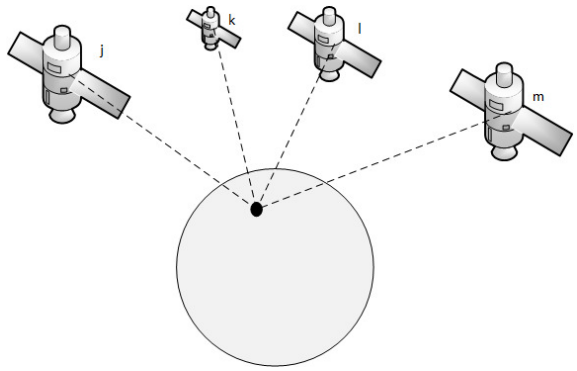


Figure 3.7: An illustration of the need of 4 satellites.

It is important to keep in mind that this is just the basic of positioning. The distance calculation is a bit more complicated than just multiplying the time difference with the speed of light. There are several ways to determine this distance but this thesis will only provide theory of two common ones. The first one is by using the message from pseudorange code measurements and the second is to calculate the length of the carrier wave phases, more on this in Section 3.4.3 and 3.4.4.

3.4.2 Satellite positioning - Ephemeris data

To be able to determine the position of a receiving object using radio wave signals it is important to know from where and when the signal was transmitted, that is, to know where the satellites are located. The determination of satellite position may differ depending on the system being used. This thesis will only provide theory for determine the position of GPS satellites.

GPS satellites transmits a message every hour containing several variables called ephemeris parameters, see Table 3.1. These variables are then used to calculate the position and the proposed orbit of the satellites to get information about future positions until the next ephemeris message. The proposed orbits are predictions and get worse over time, but as long as the messages are being received this will not be a problem.

Table 3.1: Ephemeris parameters [22].

Ephemeris Data

Name	Description	Units
M_0	Mean anomaly at reference time	semi-circles
Δn	Mean motion difference from computed value	semi-circles/s
e	Eccentricity	-
$\sqrt{A}$	Square root of the semi-major axis	$\sqrt{m}$
Ω_0	Longitude of ascending node of orbit plane at weekly epoch	semi-circles
i_0	Inclination angle at reference time	semi-circles
ω	Argument of perigee	semi-circles
$\dot{\Omega}$	Rate of right ascension	semi-circles/s
$IDOT$	Rate of inclination angle	semi-circles/s
C_{uc}	Amplitude of the cosine harmonic correction term to the argument of latitude	radians
C_{us}	Amplitude of the sine harmonic correction term to the argument of latitude	radians
C_{rc}	Amplitude of the cosine harmonic correction term to the orbit radius	m
C_{rs}	Amplitude of the sine harmonic correction term to the orbit radius	m
C_{ic}	Amplitude of the cosine harmonic correction term to the angle of inclination	radians
C_{is}	Amplitude of the sine harmonic correction term to the angle of inclination	radians
t_{oe}	Reference time ephemeris	s

Table 3.2: Required *WGS 84* parameters [22].

WGS 84 constants

constants	Description
$\mu = 3.986005 * 10^{14}\ m^3/s^2$	Earth's gravitational constant
$\dot{\Omega}_e = 7.2921151467 * 10^{-5}\ radians/s$	Earth's rotation rate

With the information in Table 3.1 together with the constants in Table 3.2 the position of the satellite in the ECEF coordinate frame can be derived by using the equations in Table 3.3.

Table 3.3: Equations regarding position of satellite [22].

Equations to provide XYZ position of satellite

Equations	Description
$A = (\sqrt{A}\,)^2$	Semi-major axis
$n_0 = \frac{\mu}{A^3}$	Computed mean motion
$t_k = t - t_{oe}$	Time from ephemeris reference epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e\sin E_k$	Kepler's equation of eccentric anomaly
$v_k = \tan^{-1}\left\{\left(\frac{\sqrt{1-e^2}\sin E_k)/(1-e\cos E_k)}{(\cos E_k - e)/(1-e\cos E_k)}\right)\right\}$	True anomaly
$E_k = \cos^{-1}\left\{\frac{e+\cos v_k}{1+e\cos v_k}\right\}$	Eccentric anomaly
$\Phi_k = v_k + \omega$	Argument of latitude
$\delta u_k = C_{us}\sin 2\Phi_k + C_{uc}\cos 2\Phi_k$	Argument of latitude Correction
$\delta r_k = C_{rs}\sin 2\Phi_k + C_{rc}\cos 2\Phi_k$	Radius correction
$\delta i_k = C_{is}\sin 2\Phi_k + C_{ic}\cos 2\Phi_k$	Inclination correction
$u_k = \Phi_k + \delta u_k$	Corrected argument of latitude
$r_k = A(1 - e\cos E_k) + \delta r_k$	Corrected radius
$i_k = i_0 + \delta i_k + (IDOT)t_k$	Corrected inclination
$x'_k = r_k\cos u_k$	positions in orbital plane
$y'_k = r_k\sin u_k$	positions in orbital plane
$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node
$x_k = x'_k\cos\Omega_k - y'_k\cos i_k\sin\Omega_k$	X-coordinate in ECEF coordinate frame
$y_k = x'_k\sin\Omega_k + y'_k\cos i_k\cos\Omega_k$	Y-coordinate in ECEF coordinate frame
$z_k = y'_k\sin i_k$	Z-coordinate in ECEF coordinate frame

The time from ephemeris reference epoch t_k must take week crossovers into account when calculating the position, this means that t_k need to be subtracted with 604 800 seconds if t_k is greater than 302 400 seconds and added with 604 800 seconds if less than $-302\,400$ seconds. Another note is that Kepler's equation of eccentric anomaly is solved numerically [22].

3.4.3 Pseudorange code positioning

Pseudorange positioning is a common method to determine an object's position on the surface of the earth using satellite information. The approach is the same as de-

scribed in Section 3.4.1, namely calculating the signal travel time between satellite and receiver. For pseudorange positioning the time is derived using the information regarding the satellite clock readings provided by a navigation message that is transmitted together with a pseudo random noise code, hereinafter PRN-code, over a carrier wave. There are some different PRN codes depending on which satellites that are used but they all have at least one in common, namely the Coarse/acquisition-code hereinafter C/A-code, which is the one that will be concerned in this thesis [3]. The C/A-code is a 1023 bit long message and is used by the receiver to determine which satellite the signal originates from, meaning that every satellite has its own C/A-code [9].

The GPS satellites have in turn two different carrier waves, L1 and L2 that are running at different frequencies. L1 is running at 1575.42 MHz and L2 at 1227.60 MHz. L1 is the one used in positioning for civilians due to the fact that L2 is reserved for the US military and other authorized users. The C/A-code is amplitude modulated on the L1 carrier wave at a frequency of 1.023 MHz. The wavelength of this signal can be calculated accordingly, where c is the speed of light and $f_{c/a}$ is the frequency

$$\lambda_{c/a} = \frac{c}{f_{c/a}} = \frac{299792458}{1.023 * 10^6} \approx 293 \ [m] \tag{3.44}$$

where the maximum precision of this positioning method can be approximated to 1 % of the wavelength, namely $\sim 3 \ m$ [2].

3.4.4 Carrier phase positioning

Carrier phase positioning is similar to the Pseudorange code method in a sense that both uses the same carrier wave L1 together with the PRN code to determine transmitting satellites. The difference is that instead of using the satellite clock readings from the navigation message it calculates the distance between satellite and receiver by counting the number of carrier waves between them [3]. By applying the same formula as in equation (3.44) but with the frequency of the L1 carrier wave, namely 1575.42 MHz, the wavelength gets approximately 0.19 m,

$$\lambda_{L1} = \frac{c}{f_{L1}} = \frac{299792458}{1575.42 * 10^6} \approx 0.19 \ [m]. \tag{3.45}$$

It is though important to use the exact answer of this formula and not the rounded number 19 cm because small numbers multiplied with the speed of light gets quite large,

$$\begin{aligned} 0.1902 \ c &= 5.7021 * 10^7, \\ 0.19 \ c &= 5.6961 * 10^7, \\ \Delta d &= 5.9958 * 10^4. \end{aligned}$$

Maximum precision of this method can be approximated in the same way as with pseudorange. 1 % of the wavelength provides a precision of $\sim 0.0019 \ m$ meaning accuracy at millimeter level [2].

This method has unfortunately one drawback. Although the length of one carrier wave is known there is no information about the number of waves in total. A term often referred as the ambiguity term, N, displayed in Figure 3.8.

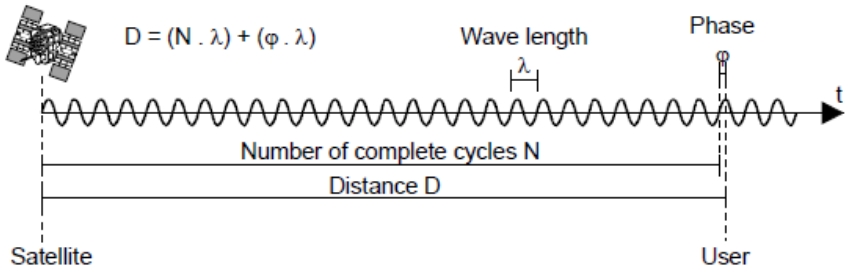


Figure 3.8: Carrier phase positioning with Ambiguity term [9].

Apart from this there are other terms that can cause errors, for example the receiver and satellite clock biases. These terms are important when it comes to using the measurements in a filter for estimation since a model is used to describe the measurements, see Section 3.6.2 for further information.

A simplified model of the carrier phase measurements is as following, where tropospheric, ionospheric and multipath errors are left out [3]

$$\lambda_{L1}\Phi_r^s(t) = \rho_r^s(t) + N + \delta_r(t) + \delta^s(t), \tag{3.46}$$

$$\rho_r^s(t) = \sqrt{(X^s(t) - X_r)^2 + (Y^s(t) - Y_r)^2 + (Z^s(t) - Z_r)^2}, \tag{3.47}$$

where Φ is the carrier phase measurement, r and s stands for receiver and satellite, δ is clock bias and ρ is the geometrical distance between the receiver and the satellite.

For situations where the exact position is not necessary but millimeter precision is desired it is preferable to use relative positioning. The advantage is then that several error terms can be cancelled or neglected by phase difference techniques [3]. These techniques are further described in Sections 3.4.4-3.4.4. However, measurements from four satellites are still required to provide information about the position.

Single differencing

Single difference is a technique to cancel the satellite clock bias from the carrier phase measurements. The method requires two receivers and one satellite per measurement since the difference is derived between two carrier phase measurements from the same satellite at the same time epoch [3]. To keep the receivers apart they are denoted by A and B and the satellite is denoted by j

$$\lambda_{L1}\Phi_A^j(t) = \rho_A^j(t) + \lambda_{L1}N_A^j + c\delta_A(t) + c\delta^j(t), \tag{3.48}$$

$$\lambda_{L1}\Phi_B^j(t) = \rho_B^j(t) + \lambda_{L1}N_B^j + c\delta_B(t) + c\delta^j(t). \tag{3.49}$$

The difference of equations (3.48)-(3.49) are then

$$\lambda_{L1}\Phi_{BA}^j(t) = \rho_{BA}^j(t) + \lambda_{L1}N_{AB}^j + c\delta_{BA}, \tag{3.50}$$

where

$$\begin{aligned}\Phi^t_{BA}(t) &= \Phi^j_B(t) - \Phi^j_A(t),\\ \rho^t_{BA}(t) &= \rho^j_B(t) - \rho^j_A(t),\\ N^j_{BA} &= N^j_B - N^j_A,\\ \delta^t_{BA}(t) &= \delta_B - \delta_A.\end{aligned}$$

The satellite clock bias is eliminated due to the difference between the phase measurements for each receiver.

Double differencing

Double differencing uses the same approach as the single differencing but is adding another satellite to the equations. The receivers are still denoted by A and B and the satellites by j and k. This technique is actually the difference between two single differences [3]

$$\lambda_{L1}\Phi^j_{BA}(t) = \rho^j_{BA}(t) + \lambda_{L1}N^j_{AB} + c\delta_{BA}, \tag{3.51}$$
$$\lambda_{L1}\Phi^k_{BA}(t) = \rho^k_{BA}(t) + \lambda_{L1}N^k_{AB} + c\delta_{BA}. \tag{3.52}$$

The difference of equations (3.51)-(3.52) is then

$$\lambda_{L1}\Phi^{kj}_{BA}(t) = \rho^{kj}_{BA}(t) + \lambda_{L1}N^{kj}_{AB}, \tag{3.53}$$

where

$$\begin{aligned}\Phi^{kj}_{BA}(t) &= \Phi^k_{BA}(t) - \Phi^j_{BA}(t),\\ \rho^{kj}_{BA}(t) &= \rho^k_{BA}(t) - \rho^j_{BA}(t),\\ N^{kj}_{BA} &= N^k_{BA} - N^j_{BA}.\end{aligned}$$

The receiver clock bias is then eliminated which is the main reason why this method is used due to the fact that the receiver bias is much greater than the satellite clock bias and even greater because multiplied with the speed of light.

Triple differencing

Single and double differencing have cancelled nearly all the extra terms but one, the ambiguity term, N. As long as the receiver has phase lock between two epochs t the ambiguity term does not change, meaning a difference between two epochs will eliminate this term. Triple differencing is actually the difference between two double differences in two consecutive time epochs [3]

$$\lambda_{L1}\Phi^{kj}_{BA}(t_1) = \rho^{kj}_{BA}(t_1) + \lambda_{L1}N^{kj}_{AB}, \tag{3.54}$$
$$\lambda_{L1}\Phi^{kj}_{BA}(t_2) = \rho^{kj}_{BA}(t_2) + \lambda_{L1}N^{kj}_{AB}. \tag{3.55}$$

The difference of equations (3.54)-(3.55) is then

$$\lambda_{L1}\Phi_{BA}^{kj}(t_2 - t_1) = \rho_{BA}^{kj}(t_2 - t_1), \tag{3.56}$$

where

$$\Phi_{BA}^{kj}(t_2t_1) = \Phi_{BA}^{k}(t_2) - \Phi_{BA}^{j}(t_1),$$
$$\rho_{BA}^{kj}(t_2t_1) = \rho_{BA}^{k}(t_2) - \rho_{BA}^{j}(t_1),$$

and the ambiguity term is eliminated as long as it remains constant between the epochs.

Time differencing

Another approach to cancel out the ambiguity term is to skip both the single and double difference steps and take the difference between two epochs t directly using only one receiver, denoted by A, and one satellite, denoted by j, per measurement

$$\lambda_{L1}\Phi_{A}^{j}(t_1) = \rho_{A}^{j}(t_1) + \lambda_{L1}N_{A}^{j} + c\delta_{A}(t_1) + c\delta^{j}(t_1), \tag{3.57}$$
$$\lambda_{L1}\Phi_{A}^{j}(t_2) = \rho_{A}^{j}(t_2) + \lambda_{L1}N_{A}^{j} + c\delta_{A}(t_2) + c\delta^{j}(t_2). \tag{3.58}$$

The difference of equations (3.57)-(3.58) is then

$$\lambda_{L1}(\Phi_{A}^{j}(t_2) - \Phi_{A}^{j}(t_1)) = (\rho_{A}^{j}(t_2) - \rho_{A}^{j}(t_1)) + c(\delta_{A}(t_2) - \delta_{A}(t_1)) + c(\delta^{j}(t_2) - \delta^{j}(t_1)). \tag{3.59}$$

The ambiguity term is then cancelled but the clock biases are remaining. The satellite clock bias can be neglected due to accurate clock in the satellites but the receiver clock bias need to be estimated, for example by a Kalman filter [14].

3.5 Inertial measurement unit - IMU

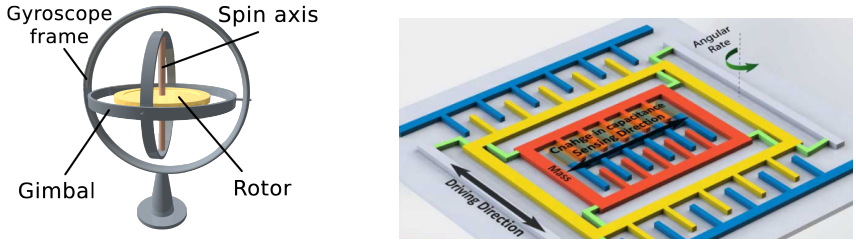
Inertial measurement unit (IMU) is as the name reveals a device that is able to measure an object's inertial motion where the measurement depends on the sensors that the unit is equipped with. The usual setup is accelerometers to measure the accelerations of the object, gyroscopes to measure the angular rate and magnetometers to measure surrounding magnetic fields. The last sensor is not always included in the units but can be at help in finding directions, for example where north is. The IMUs are getting cheaper and are frequently used in products that need to know its orientation, from autonomous vehicles to ordinary products as mobile phones. The IMUs in this thesis are not equipped with magnetometers so there will be no theory of this sensor. The following two sections will instead describe gyroscopes and accelerometers.

3.5.1 Gyroscope

Gyroscopes are sensors that measure the rate of rotation of an object. There are several different types of gyroscopes, for example the one with the spinning disc seen

in Figure 3.9a that may be the best known but not suitable for embedded systems. Then there are high precision gyroscopes such as ring-laser or fiber-optic sensors that are too expensive to use in mass produced products such as vehicles and mobile phones. Another gyroscope is the small MEMS-sensor, which by its light weight and fair price becomes a great substitution to the expensive ones. The MEMS sensor is a combination of both electrical and mechanical systems at a microscopically level, hence the name Micro ElectroMechanical System[26] (see Figure 3.9b).

There are different kinds of MEMS sensors and as described in Section 2.1.2 the sensor that will be used in this thesis is a vibratory MEMS-sensor. This type can be briefly explained as a sensor without rotating or bearing parts as the ordinary gyroscope. This sensor is instead equipped with a vibrating proof-mass suspended by flexible beams that induces Coriolis force when it gets an orthogonal angular rate input. The Coriolis force then give rise to energy that gets translated to a rotation rate and when there are no rotation rate inputs the sensor should output zeros [26].



(a) Original Gyroscope[26]. **(b)** MEMS gyroscope[27].

Figure 3.9: Two different kinds of gyroscopes.

3.5.2 Accelerometer

Accelerometers, as the name points out, are sensors that measure translational acceleration of an object. There are several different accelerometers, some more applicable to problems like this and some not. The basic theory is based on Newton's second law, that is, a force acting on a body with mass m causes the body to accelerate, $F = ma$.

The accelerometer sensor used in this thesis is a MEMS sensor described in Section 2.1.2. This sensor uses a proof mass connected to a spring where the force produces energy that is translated to acceleration (see Figure 3.10). When the object is stationary it will measure the force acting to stop it from falling, the gravitational force g.

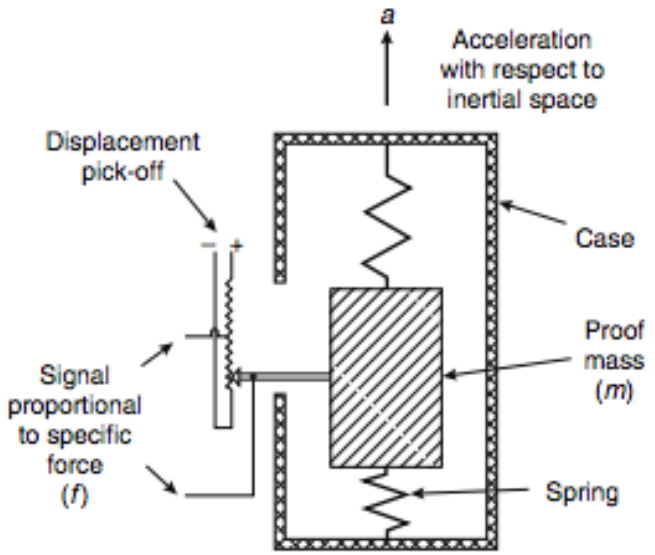


Figure 3.10: Simplified Accelerometer description [5].

3.6 Sensor fusion and filtering

The basics of sensor fusion are to combine data from multiple sensors to enhance the quality of gathered information using statistical theory. A sensor used for position determination merged with a sensor for angular rate may provide more information of the measured object than if used separately. Sensor fusion is not only about combining different kinds of sensors. Equal kinds of sensors are also able to improve the measurements by redundancy [7].

The name sensor fusion is only the concept of merging sensors where filtering performs the actual merging and this is where the statistical theory comes in place. The filtering in this thesis is restricted to Bayesian filtering which refers to methods for estimating time-varying systems.

Bayesian filtering refers to computation of the "*marginal posterior distribution or filtering distribution of the state* $\boldsymbol{x}_k$ *at each time step k given the history of the measurements up to the time step k*"[4]. This recursive algorithm consists of two steps, a prediction step and an update step where the algorithm requires initialization by a prior distribution $p(\boldsymbol{x}_0)$.

The prior distribution serves to help the estimation process to faster converge to desired values by providing it with known information before the actual filtering starts. For example, a starting position of an object, which position is being estimated. This is especially important for systems that are nonlinear since there is more than one possible outcome.

The prediction step makes use of a process model, often referred to as the motion

model of the system, to predict the next state at time step k by computing the *Chapman-Kolmogorov equation*

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})\,\mathrm{d}\boldsymbol{x}_{k-1}, \tag{3.60}$$

where $\boldsymbol{y}_{1:k-1}$ are measurements up to time $k-1$ and $\boldsymbol{x}_k$ describes the state at time k.

The update step makes use of the all sensors available, often referred to as the measurement model of the system, at time step k to update the predicted states by using *Bayes' rule*

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{\int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})\,\mathrm{d}\boldsymbol{x}_k}. \tag{3.61}$$

The selection of motion models and measurement models depends on the system being estimated and the sensors that are applied, see Sections 3.6.1-3.6.2 for further description.

The choice of filter depends on the properties of these models. The Kalman filter is a well known filter that applies Bayesian equations for state estimation. Although the Kalman filter is great for estimation using linear models it is not able to handle non-linear ones. However, there are extensions for this filter to make this possible. This thesis will concern three kinds of Kalman filters, namely the ordinary *Kalman filter*, the *Extended Kalman filter* and the *Cubature Kalman filter* [4], see Section 3.6.3.

3.6.1 Motion model

The motion model describes the behaviour of the estimated system and its uncertainties as a *Markov sequence* [4]. This means that the state x_k given x_{k-1} depends only on the previous state and not on states occurred before x_{k-1} [6], thus the density function becomes

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{x}_{k-2}, \ldots, \boldsymbol{x}_0) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}). \tag{3.62}$$

The motion model is displayed in continuous time as

$$\dot{\boldsymbol{x}}(t) = \tilde{f}(\boldsymbol{x}(t)) + \tilde{\boldsymbol{q}}(t), \tag{3.63}$$

and in discrete time as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{q}_{k-1}, \tag{3.64}$$

where the motion noise is described as

$$\begin{aligned} \boldsymbol{q}_{k-1} &\sim \mathcal{N}(0, \boldsymbol{Q}_{k-1}), \\ \boldsymbol{Q}_{k-1} = cov(\boldsymbol{q}_{k-1}) &= cov(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = cov(\boldsymbol{x}(t+T)|\boldsymbol{x}(t)). \end{aligned} \tag{3.65}$$

The complexity of these models varies depending on the system being estimated. Two models that are commonly used to describe vehicle motions are the constant velocity (CV) model and constant acceleration (CA) model. A vehicle in one dimension with the state vector $\boldsymbol{x}(t) = [p(t)\ v(t)]$, where $p(t)$ is the position and $v(t)$ is the velocity, can be modeled as a CV model

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{q}(t) \tag{3.66}$$

which gives a good representation of a vehicle with constant velocity. The vehicle can be modeled as a CA model by adding an acceleration $a(t)$ to the state vector, $\boldsymbol{x}(t) = [p(t)\ v(t)\ a(t)]$,

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \boldsymbol{q}(t) \tag{3.67}$$

which gives a good representation of a vehicle with constant acceleration. These are two examples of how continuous motion models can look like.

Computer based filters operates in discrete time which means that it is necessary to discretize the models before using them in the filter. There are different ways to discretize a time continuous model and this thesis will explain two different approaches, The *Analytic solution* and the *Euler discretization method*.

Both methods works well if the motion models are linear and the filter is sampled at high frequency, meaning a small sample time T, hence the models becomes very similar (see equation (3.69)-(3.70)). However, the Euler method gets inaccurate if the frequency is low and the analytic solution is then preferable. The downside for the analytic solution is though that this method is only adapted to linear models when the Euler method is not.

The linear motion function is often denoted

$$f(\boldsymbol{x}_{k-1}) = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} \tag{3.68}$$

and the continuous transition matrix $\tilde{\boldsymbol{A}}$ is discretized using the analytic solution where the exact discretization becomes

$$\boldsymbol{A}_{k-1} = e^{(T\tilde{\boldsymbol{A}}_{k-1})} = \boldsymbol{I} + T\tilde{\boldsymbol{A}} + \frac{T^2}{2}\tilde{\boldsymbol{A}}^2 + \cdots, \tag{3.69}$$

or by using the Euler method where the approximated discretization is

$$\boldsymbol{A}_{k-1} = \boldsymbol{I} + T\tilde{\boldsymbol{A}}. \tag{3.70}$$

The nonlinear motion function is discretized with the *Euler method* as

$$f(x_{k-1}) = x_{k-1} + T\tilde{f}(x_{k-1}) \tag{3.71}$$

where $\tilde{f}$ denotes the continuous function.

The discretization of the motion noise covariance matrix can be derived with the analytic solution, where the exact discretization is

$$\boldsymbol{Q}_{k-1} = \int_0^T e^{\tilde{\boldsymbol{A}}\tau} \tilde{\boldsymbol{Q}} e^{\tilde{\boldsymbol{A}}^T\tau} \mathrm{d}\tau \tag{3.72}$$

or with the Euler method, where the approximated discretization is

$$\boldsymbol{Q}_{k-1} = T\tilde{\boldsymbol{Q}}, \tag{3.73}$$

where $\tilde{\boldsymbol{Q}} = \Gamma \boldsymbol{Q} \Gamma^T$.

The stated CV model in equation (3.66) can be discretized with the Euler method as

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\Gamma} \boldsymbol{q}_{k-1}, \tag{3.74}$$

$$\boldsymbol{Q}_{k-1} = T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma^2 \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & T\sigma^2 \end{bmatrix} \tag{3.75}$$

and the stated CA model in equation (3.67) as

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ \ddot{x}_{k-1} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\Gamma} \boldsymbol{q}_{k-1}, \tag{3.76}$$

$$\boldsymbol{Q}_{k-1} = T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \sigma^2 \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & T\sigma^2 \end{bmatrix}. \tag{3.77}$$

3.6.2 Measurement model

The measurement model relates the current states to the measurements being made. If there are two kinds of sensors then the measurement model requires two functions, one for each type of sensor. For example, a gyroscope that measures angular rate should be related to a current angular rate state. The measurement model is $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and could be displayed in continuous time as

$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t)) + \boldsymbol{r}(t), \tag{3.78}$$

and in discrete time as

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{r}_k, \tag{3.79}$$

where h is the measurement function and $\boldsymbol{r}$ is the measurement noise with the distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$.

By using the same approach as for the motion noise, the measurement noise covariance matrix is derived as

$$\boldsymbol{R} = \boldsymbol{C}\boldsymbol{R}_{mat}\boldsymbol{C}^T, \boldsymbol{R}_{mat} = \boldsymbol{I}_n\boldsymbol{r}_{vec}, \tag{3.80}$$

where C is a matrix that relates the noise to correct measurements, similar to the motion vector Γ.

3.6.3 Filter

This section describes three mentioned filters, namely the ordinary *Kalman filter*, the *Extended Kalman filter* and the *Cubature Kalman filter*.

Kalman filter - KF

The Kalman filter is a Bayesian filter for systems with linear motion- and measurement models. The Kalman filter is the linear minimum mean square error (LMMSE) estimator and is a commonly used filter for linear problems where the motion- and measurement model can be described as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{q}_{k-1} = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{q}_{k-1}, \tag{3.81}$$
$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{r}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{r}_k. \tag{3.82}$$

$\boldsymbol{A}$ is an $n \times n$ matrix and $\boldsymbol{H}$ is an $m \times n$ matrix, where N is the amount of states and M the amount of measurements[4].

The *prediction step* provides the predicted estimated state $\hat{\boldsymbol{x}}_{k|k-1}$ together with the covariance matrix $\boldsymbol{P}_{k|k-1}$ based on the current estimated state by using the linear motion model and covariance matrix $\boldsymbol{Q}$ as

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1}, \tag{3.83}$$
$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1}. \tag{3.84}$$

The *update step* corrects the prediction based on the Kalman gain $\boldsymbol{K}_k$, the innovation $\boldsymbol{v}_k$ and the innovation covariance $\boldsymbol{S}_k$ as

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\boldsymbol{v}_k, \tag{3.85}$$
$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^T, \tag{3.86}$$

where

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1}, \tag{3.87}$$
$$\boldsymbol{v}_k = \boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}, \tag{3.88}$$
$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k. \tag{3.89}$$

The innovation $\boldsymbol{v}_k$ describes the difference between the measurement and the predicted state, the Kalman gain determines the creditability of the new information and the $\boldsymbol{S}_k$ is the innovation covariance [17].

Extended Kalman filter - EKF

The *Extended Kalman Filter* (EKF) is an extension of the Kalman Filter that is capable of filtering nonlinear problems

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{q}_{k-1}, \tag{3.90}$$
$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{r}_k. \tag{3.91}$$

The filter utilizes the linearization of the system by the partial derivative of the motion- and measurement model in the region of the estimated state

$$f'_x(\boldsymbol{x}) = \frac{\delta f(\boldsymbol{x})}{\delta \boldsymbol{x}}, \tag{3.92}$$
$$h'_x(\boldsymbol{x}) = \frac{\delta h(\boldsymbol{x})}{\delta \boldsymbol{x}}. \tag{3.93}$$

Matrices $\boldsymbol{A}$ and $\boldsymbol{H}$ from the KF, in Section 3.6.3, are approximated to $f'_x(\boldsymbol{x})$ respectively $h'_x(\boldsymbol{x})$.

The EFK tends to perform well when the models are not too nonlinear and the partial derivative linearization is a good approximation in the region of interest [17].

Cubature Kalman filter - CKF

Highly nonlinear filter problems cannot be efficiently solved by the Extended Kalman Filter, since the partial derivative linearization at the area of interest can be a bad approximation of the distribution.

Another way to filter highly non-linear problems is with Gaussian moment matching [4], by approximating measurements $h(\boldsymbol{x})$ to fit the Gaussian integrals of the form

$$\int h(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{x}}, \boldsymbol{P}). \tag{3.94}$$

There are many different methods to approximating the integral, such as *Gauss –Hermite cubature*, *Gauss–Hermite quadrature* and *Spherical cubature integration*, but this section will focus on the *Cubature Kalman filter* (CKF) that utilizes the *Spherical cubature integration*.

The moment matching in the *prediction step* is based on a distribution of unit sigma points $\boldsymbol{\xi}$ that in turn are based on the amount of states n as

$$\boldsymbol{\xi}^{(i)} = \begin{cases} +\sqrt{n}\boldsymbol{e}_i, & i = 1, ..., n \\ -\sqrt{n}\boldsymbol{e}_{i-1}, & i = n+1, ..., 2n \end{cases}. \tag{3.95}$$

The unit sigma points $\boldsymbol{\xi}$ are used to produce sigma points $\boldsymbol{\chi}_{k-1}$ to propagate through the model

$$\boldsymbol{\chi}^{(i)}_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1} + \sqrt{\boldsymbol{P}_{k-1|k-1}}\boldsymbol{\xi}^{(i)} \quad i = 1, ..., 2n, \tag{3.96}$$

where $\sqrt{\boldsymbol{P}}$ is a matrix square root defined by $\boldsymbol{P} = \sqrt{\boldsymbol{P}}\sqrt{\boldsymbol{P}}^T$.

The mean of the propagated sigma points is calculated to estimate the predicted state $\hat{\boldsymbol{x}}_{k|k-1}$ and the covariance matrix can be obtained as

$$\hat{\boldsymbol{x}}_{k|k-1} \approx \frac{1}{2N}\sum_{i=1}^{2N} f(\boldsymbol{\chi}_{k-1}^{(i)}), \tag{3.97}$$

$$\boldsymbol{P}_{k|k-1} \approx \frac{1}{2N}\sum_{i=1}^{2N}(f(\boldsymbol{\chi}_{k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1})(f(\boldsymbol{\chi}_{k-1}^{(i)}) - \hat{\boldsymbol{x}}_{k|k-1})^T + \boldsymbol{Q}_{k-1}. \tag{3.98}$$

A Cubature *update step* is quite similar to the prediction step. Firstly the sigma points is created

$$\boldsymbol{\chi}_k^{(i)} \quad = \hat{\boldsymbol{x}}_{k|k-1} + \sqrt{\boldsymbol{P}_{k|k-1}}\boldsymbol{\xi}^{(i)} \quad i = 1, ..., 2n, \tag{3.99}$$

where the unit sigma points $\boldsymbol{\xi}^{(i)}$ are the same as in the prediction. The sigma points are used to create corresponding measurements

$$\boldsymbol{\mathcal{Y}}_k^{(i)} = h(\hat{\boldsymbol{\chi}}_k^{(i)}). \tag{3.100}$$

The desired moments can be computed as

$$\hat{\boldsymbol{y}}_{k|k-1} \approx \frac{1}{2N}\sum_{i=1}^{2N}\boldsymbol{\mathcal{Y}}_k^{(i)}, \tag{3.101}$$

where

$$\boldsymbol{v}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}, \tag{3.102}$$

$$\boldsymbol{S}_k = \frac{1}{2N}\sum_{i=1}^{2N}(\boldsymbol{\mathcal{Y}}_k^{(i)} - \hat{\boldsymbol{y}}_{k|k-1})(\boldsymbol{\mathcal{Y}}_k^{(i)} - \hat{\boldsymbol{y}}_{k|k-1})^T + \boldsymbol{R}_k, \tag{3.103}$$

$$\boldsymbol{C}_k = \frac{1}{2N}\sum_{i=1}^{2N}(\boldsymbol{\chi}_k^{(i)} - \hat{\boldsymbol{x}}_{k|k-1})(\boldsymbol{\mathcal{Y}}_k^{(i)} - \hat{\boldsymbol{y}}_{k|k-1})^T, \tag{3.104}$$

$$\boldsymbol{K}_k = \boldsymbol{C}_k\boldsymbol{S}_k^{-1}. \tag{3.105}$$

The innovation $\boldsymbol{v}_k$, the innovation covariance $\boldsymbol{S}_k$ and the Kalman gain $\boldsymbol{K}_k$ can be calculated and used to estimate the state, as in Section 3.6.3.

4
Filter design

This chapter describes how the filter was designed in detail. It will start with a brief motivation that explains the selection of the filter. Then it continues with describing two different motion models, denoted as separated and merged model. The chapter then ends with describing the measurement models used in this thesis.

There are of course a variety of filters that can be used for this type of estimation but some filters are more suitable than others depending on the motion and measurement models. The models in this system are all nonlinear as seen in Sections 4.1-4.2. From the filters described in Section 3.6 there are two that handles nonlinearities, namely the *Extended Kalman Filter*, *EKF* and the *Cubature Kalman Filter*, *CKF*. There are of course a variety of other filters that handles this but these are the one mentioned in this thesis. To avoid linearizing the models and to use an *EKF* on a system that may be too nonlinear it is preferable to use a Cubature Kalman filter even though it is a high dimensional system that results in a lot of sigma points.

The theory chapter describes two different measurement models for GNSS sensors, one using the triple differencing method and another with time differencing. However, the filter will only use one of the methods in this thesis, namely the time differencing (see Section 3.4.4). This means that six additional states, drift and drift velocity for each GNSS sensor, are required in the state vector.

4.1 Motion model

The Cubature Kalman filter are constructed with two kinds of motion models. The first is with no connection between the bodies and is therefore denoted as the separated body motion model (SB). The second is with constraints and is therefore denoted as the merged body motion model (MB). It is quite obvious that the motion model without any connection between the bodies may drift away from each other. However this model is used to evaluate the influence of connecting bodies with motion constraints in filtering.

4.1.1 Separated bodies

The *separated body model* (SB) consists of 44 states and is mostly based on *Constant Accelerating* (CA) and *Constant Velocity* (CV) motion models, discretized with Euler method. The cabin and chassis has it is own positions $\boldsymbol{p} = [x\ y\ z]^T$, velocity

$\dot{\boldsymbol{p}}$, and acceleration $\ddot{\boldsymbol{p}}$ in the global coordinate frame (ECEF) and orientations are explained through quaternions $\mathfrak{q} = [\mathfrak{q}_0\ \mathfrak{q}_1\ \mathfrak{q}_2\ \mathfrak{q}_3]^T$ to prevent a possible Gimbal lock. However, the motion of the quaternions is made with angular velocities $\boldsymbol{\omega}$ and angular accelerations $\dot{\boldsymbol{\omega}}$ as rotations about each axis. The receiver clock biases δ_i are also included in the state vector but as the difference in bias between two epochs, often referred as the drift

$$\boldsymbol{\Delta\delta} = \frac{1}{t_1 - t_2}\begin{bmatrix}\delta_1(t_2) - \delta_1(t_1)\\ \delta_2(t_2) - \delta_2(t_1)\\ \delta_3(t_2) - \delta_3(t_1)\end{bmatrix}. \tag{4.1}$$

The quaternion rates are depended by angular rates as described in Section 3.2.2,

$$\dot{\mathfrak{q}} = f_{\mathfrak{q}}(\mathfrak{q}, \boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{S}(\Omega(\mathfrak{q})^T\boldsymbol{\omega})\mathfrak{q}. \tag{4.2}$$

The states is combined to the state vector $\boldsymbol{X}$ where the function $f(\boldsymbol{X})$ describes the motion

$$\boldsymbol{X}_k = f(\boldsymbol{X}_{k-1}) + \boldsymbol{\Gamma}\boldsymbol{q}_{k-1}, \tag{4.3}$$

$$\underbrace{\begin{bmatrix}\boldsymbol{p}_k^{cabin}\\ \dot{\boldsymbol{p}}_k^{cabin}\\ \ddot{\boldsymbol{p}}_k^{cabin}\\ \mathfrak{q}_k^{cabin}\\ \boldsymbol{\omega}_k^{cabin}\\ \dot{\boldsymbol{\omega}}_k^{cabin}\\ \boldsymbol{p}_k^{chassis}\\ \dot{\boldsymbol{p}}_k^{chassis}\\ \ddot{\boldsymbol{p}}_k^{chassis}\\ \mathfrak{q}_k^{chassis}\\ \boldsymbol{\omega}_k^{chassis}\\ \dot{\boldsymbol{\omega}}_k^{chassis}\\ \boldsymbol{\Delta\delta}_k\\ \boldsymbol{\Delta\dot{\delta}}_k\end{bmatrix}}_{\boldsymbol{X}_k} = \underbrace{\begin{bmatrix}\boldsymbol{A}_{CA}\begin{bmatrix}\boldsymbol{p}_{k-1}^{cabin}\\ \dot{\boldsymbol{p}}_{k-1}^{cabin}\\ \ddot{\boldsymbol{p}}_{k-1}^{cabin}\end{bmatrix}\\ \mathfrak{q}_{k-1}^{cabin} + f_{\mathfrak{q}}(\mathfrak{q}_{k-1}^{cabin}, \boldsymbol{\omega}_{k-1}^{cabin})T\\ \boldsymbol{A}_{CV}\begin{bmatrix}\boldsymbol{\omega}_{k-1}^{cabin}\\ \dot{\boldsymbol{\omega}}_{k-1}^{cabin}\end{bmatrix}\\ \boldsymbol{A}_{CA}\begin{bmatrix}\boldsymbol{p}_{k-1}^{chassis}\\ \dot{\boldsymbol{p}}_{k-1}^{chassis}\\ \ddot{\boldsymbol{p}}_{k-1}^{chassis}\end{bmatrix}\\ \mathfrak{q}_{k-1}^{chassis} + f_{\mathfrak{q}}(\mathfrak{q}_{k-1}^{chassis}, \boldsymbol{\omega}_{k-1}^{chassis})T\\ \boldsymbol{A}_{CV}\begin{bmatrix}\boldsymbol{\omega}_{k-1}^{chassis}\\ \dot{\boldsymbol{\omega}}_{k-1}^{chassis}\end{bmatrix}\\ \boldsymbol{A}_{CV}\begin{bmatrix}\boldsymbol{\Delta\delta}_{k-1}\\ \boldsymbol{\Delta\dot{\delta}}_{k-1}\end{bmatrix}\end{bmatrix}}_{f(\boldsymbol{X}_{k-1})} + \underbrace{\begin{bmatrix}\boldsymbol{0}_{6\times3} & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \cdots & \cdots & \cdots\\ \boldsymbol{0}_{7\times3} & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \cdots & \cdots\\ \boldsymbol{0}_{6\times3} & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{0}_{3\times3} & \cdots & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \cdots\\ \boldsymbol{0}_{7\times3} & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{0}_{3\times3} & \cdots & \cdots & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{0}_{3\times3} & \cdots & \cdots & \cdots & \boldsymbol{I}_3\end{bmatrix}}_{\boldsymbol{\Gamma}} \underbrace{\begin{bmatrix}\boldsymbol{q}_{k-1}^{cabin^p}\\ \boldsymbol{q}_{k-1}^{cabin^\omega}\\ \boldsymbol{q}_{k-1}^{chassis^p}\\ \boldsymbol{q}_{k-1}^{chassis^\omega}\\ \boldsymbol{q}_{k-1}^{\Delta}\end{bmatrix}}_{\boldsymbol{q}_{k-1}},$$

where $\boldsymbol{I}_n$ is an identity matrix of size n and $\boldsymbol{0}_{n\times m}$ is a matrix of size $n \times m$ consisting of zeros. The quaternions are iteratively normalized to size 1, $\mathfrak{q} := \frac{\mathfrak{q}}{\|\mathfrak{q}\|}$ and the matrices $\boldsymbol{A}_{CV}$ and $\boldsymbol{A}_{CA}$ are

$$\boldsymbol{A}_{CV} = \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix} \qquad \boldsymbol{A}_{CA} = \begin{bmatrix}1 & T & 0\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix}. \tag{4.4}$$

The motion noise $\boldsymbol{q}_k$ is modeled as Gaussian distributed with covariance matrix $\boldsymbol{Q}$ as

$$\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}) \tag{4.5}$$

and the discrete covariance matrix $\boldsymbol{Q}_k$ can be calculated as

$$\boldsymbol{Q}_k = T\boldsymbol{\Gamma}\boldsymbol{Q}\boldsymbol{\Gamma}, \tag{4.6}$$

where T is the sample time, shown in equation (3.70).

4.1.2 Merged bodies

The *Merged body model* (MB) is a more advanced model than the SB model but with one less state, that is 43 states. The chassis position $\boldsymbol{p} = [x\ y\ z]^T$, velocity $\dot{\boldsymbol{p}}$, and acceleration $\ddot{\boldsymbol{p}}$ are defined in the global coordinate frame (ECEF) and the orientation is described with quaternions $\mathfrak{q} = [\mathfrak{q}_0\ \mathfrak{q}_1\ \mathfrak{q}_2\ \mathfrak{q}_3]^T$ to prevent possible Gimbal lock as in the SB model. The cabin's position and orientation are defined as a displacement between the chassis. The cabin's position, velocity, and acceleration are in the chassis local coordinate frame and since the relative rotations will remain around zero the Euler angles $\boldsymbol{\varepsilon} = [\theta\ \varphi\ \psi]^T$ will be at no risk for ending up in Gimbal lock and are therefore used to describe the orientation. The motion of the orientation for both the chassis and the cabin is made with angular velocities $\boldsymbol{\omega}$ and angular accelerations $\dot{\boldsymbol{\omega}}$ as rotations about each axis, but the chassis states are defined in the global coordinate frame (ECEF) and the cabin's in the chassis local coordinate frame. The differences of the three GNSS receiver clock biases δ_i, are also included as states in the same manner as for the SB model

$$\boldsymbol{\Delta\delta} = \frac{1}{t_1 - t_2}\begin{bmatrix}\delta_1(t_2) - \delta_1(t_1)\\ \delta_2(t_2) - \delta_2(t_1)\\ \delta_3(t_2) - \delta_3(t_1)\end{bmatrix}. \tag{4.7}$$

The chassis' position is modeled as a CA model and the quaternion rates are depended by angular rates as described in Section 3.2.2,

$$\dot{\mathfrak{q}} = f_{\mathfrak{q}}(\mathfrak{q}, \boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{S}(\mathfrak{Q}(\mathfrak{q})^T\boldsymbol{\omega})\mathfrak{q}. \tag{4.8}$$

The cabin's motion is based on rigid body dynamics, Section 3.3. Each translation and rotation axis has been added with springs and dampers, forcing the cabin to an

equilibrium point. The translation equation of motion becomes

$$m\boldsymbol{a} = \sum \boldsymbol{F}, \tag{4.9}$$

$$\begin{aligned} m(\ddot{\boldsymbol{p}}^{cabin} + \boldsymbol{\mathfrak{Q}}(\mathbf{q}^{chassis})^T \ddot{\boldsymbol{p}}^{chassis}) = &- \underbrace{\begin{bmatrix} k_x & & \\ & k_y & \\ & & k_z \end{bmatrix}}_{\boldsymbol{k}_p} (\boldsymbol{p}^{cabin} - \underbrace{\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}}_{\boldsymbol{p}^r}) \\ &- \underbrace{\begin{bmatrix} c_x & & \\ & c_y & \\ & & c_z \end{bmatrix}}_{\boldsymbol{c}_p} \dot{\boldsymbol{p}}^{cabin}, \end{aligned} \tag{4.10}$$

and can be simplified to

$$\begin{aligned} \ddot{\boldsymbol{p}}^{cabin} &= f_p(\boldsymbol{X}) \\ &= -\frac{\boldsymbol{k}_p}{m}(\boldsymbol{p}^{cabin} - \boldsymbol{p}^r) - \frac{\boldsymbol{c}_p}{m}\dot{\boldsymbol{p}}^{cabin} - \boldsymbol{\mathfrak{Q}}(\mathbf{q}^{chassis})^T \ddot{\boldsymbol{p}}^{chassis}. \end{aligned} \tag{4.11}$$

The chassis' angular acceleration is rotated to the body frame of the cabin and every inertia in the inertia matrix $\boldsymbol{I}_O$ are for simplicity selected to the same value I_{xyz}

$$\boldsymbol{I}_O = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} I_{xyz}, \tag{4.12}$$

which eliminates terms in equation (4.16) and reduces the number of coefficients to determine.

The angular momentum becomes

$$\begin{aligned} \dot{\boldsymbol{L}}_O &= \frac{\delta \boldsymbol{L}_O}{\delta t} + (\boldsymbol{\omega}^{cabin} + \boldsymbol{\mathfrak{Q}}(\mathbf{q}^{chassis})^T \boldsymbol{\omega}^{chassis}) \times \boldsymbol{L}_O \\ &= \frac{\delta \boldsymbol{L}_O}{\delta t} + 0 \\ &= \dot{\boldsymbol{\omega}}^{cabin} + \boldsymbol{\mathfrak{Q}}(\mathbf{q}^{chassis})^T \dot{\boldsymbol{\omega}}^{chassis}, \end{aligned} \tag{4.13}$$

and the rotational equation of motion becomes

$$\dot{\boldsymbol{L}}_O = \sum \boldsymbol{M}_O, \tag{4.14}$$

$$\begin{aligned} \boldsymbol{I}_{xyz}(\dot{\boldsymbol{\omega}}^{cabin} + \boldsymbol{\mathfrak{Q}}(\mathbf{q}^{chassis})^T \dot{\boldsymbol{\omega}}^{chassis}) = &- \underbrace{\begin{bmatrix} k_\theta & & \\ & k_\varphi & \\ & & k_\psi \end{bmatrix}}_{\boldsymbol{k}_\varepsilon} (\boldsymbol{\varepsilon}^{cabin} - \boldsymbol{\varepsilon}^r) \\ &- \underbrace{\begin{bmatrix} c_\theta & & \\ & c_\varphi & \\ & & c_\psi \end{bmatrix}}_{\boldsymbol{c}_\varepsilon} (\boldsymbol{\omega}^{cabin} - \dot{\boldsymbol{\varepsilon}}^r). \end{aligned} \tag{4.15}$$

and can be simplified to

$$\begin{aligned}\dot{\boldsymbol{\omega}}^{cabin} &= f_{\varepsilon}(\boldsymbol{X}) \\ &= -\frac{\boldsymbol{k}_{\varepsilon}}{I_{xyz}}(\boldsymbol{\varepsilon}^{cabin} - \boldsymbol{\varepsilon}^{r}) - \frac{\boldsymbol{c}_{\varepsilon}}{I_{xyz}}(\boldsymbol{\omega}^{cabin} - \dot{\boldsymbol{\varepsilon}}^{r})) - \boldsymbol{\Omega}(\mathbf{q}^{chassis})^{T}\dot{\boldsymbol{\omega}}^{chassis}.\end{aligned} \tag{4.16}$$

The $\boldsymbol{\varepsilon}^{r}$ describes how the relative angle should be if the cabin's position differs from its equilibrium point (see Figure 4.1). It should be the same as the angle between the position vector and the chassis local z-axis and is calculated as

$$\boldsymbol{\varepsilon}^{r} = \sin^{-1}\left(\boldsymbol{e}_{z} \times \frac{\boldsymbol{p}^{cabin}}{\|\boldsymbol{p}^{cabin}\|}\right). \tag{4.17}$$

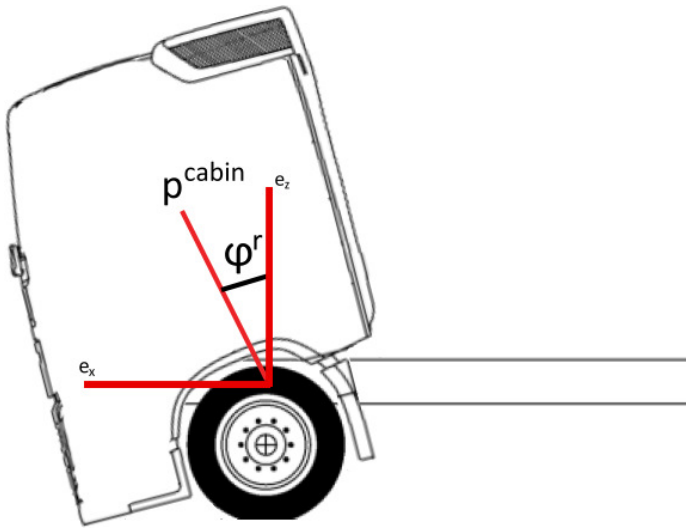


Figure 4.1: An illustration of the concept of the reference angle $\boldsymbol{\varepsilon}^{r}$ by showing the pitch angle φ^{r}.

The states are combined to the state vector $\boldsymbol{X}$ where the function $f(\boldsymbol{X})$ describes

the motion

$$\boldsymbol{X}_k = f(\boldsymbol{X}_{k-1}) + \boldsymbol{\Gamma}\boldsymbol{q}_{k-1}, \tag{4.18}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{p}_k^{cabin} \\ \dot{\boldsymbol{p}}_k^{cabin} \\ \ddot{\boldsymbol{p}}_k^{cabin} \\ \boldsymbol{\varepsilon}_k^{cabin} \\ \boldsymbol{\omega}_k^{cabin} \\ \dot{\boldsymbol{\omega}}_k^{cabin} \\ \boldsymbol{p}_k^{chassis} \\ \dot{\boldsymbol{p}}_k^{chassis} \\ \ddot{\boldsymbol{p}}_k^{chassis} \\ \mathbf{q}_k^{chassis} \\ \boldsymbol{\omega}_k^{chassis} \\ \dot{\boldsymbol{\omega}}_k^{chassis} \\ \boldsymbol{\Delta\delta}_k \\ \boldsymbol{\Delta\dot{\delta}}_k \end{bmatrix}}_{\boldsymbol{X}_k} = \underbrace{\begin{bmatrix} \boldsymbol{p}_{k-1}^{cabin} + \dot{\boldsymbol{p}}_{k-1}^{cabin} T \\ \dot{\boldsymbol{p}}_{k-1}^{cabin} + \ddot{\boldsymbol{p}}_{k-1}^{cabin} T \\ f_p(\boldsymbol{X}_{k-1}) \\ \boldsymbol{\varepsilon}_{k-1}^{cabin} + \dot{\boldsymbol{\varepsilon}}_{k-1}^{cabin} T \\ \dot{\boldsymbol{\varepsilon}}_{k-1}^{cabin} + \ddot{\boldsymbol{\varepsilon}}_{k-1}^{cabin} T \\ f_\varepsilon(\boldsymbol{X}_{k-1}) \\ \boldsymbol{A}_{CA} \begin{bmatrix} \boldsymbol{p}_{k-1}^{chassis} \\ \dot{\boldsymbol{p}}_{k-1}^{chassis} \\ \ddot{\boldsymbol{p}}_{k-1}^{chassis} \end{bmatrix} \\ \mathbf{q}_{k-1}^{chassis} + f_{\mathbf{q}}(\mathbf{q}_{k-1}^{chassis}, \boldsymbol{\omega}_{k-1}^{chassis}) T \\ \boldsymbol{A}_{CV} \begin{bmatrix} \boldsymbol{\omega}_{k-1}^{chassis} \\ \dot{\boldsymbol{\omega}}_{k-1}^{chassis} \end{bmatrix} \\ \boldsymbol{A}_{CV} \begin{bmatrix} \boldsymbol{\Delta\delta}_{k-1} \\ \boldsymbol{\Delta\dot{\delta}}_{k-1} \end{bmatrix} \end{bmatrix}}_{f(\boldsymbol{X}_{k-1})} + \underbrace{\begin{bmatrix} \mathbf{0}_{6\times3} & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \cdots & \cdots & \cdots \\ \mathbf{0}_{6\times3} & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \cdots & \cdots \\ \mathbf{0}_{6\times3} & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{3\times3} & \cdots & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \cdots \\ \mathbf{0}_{7\times3} & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{3\times3} & \cdots & \cdots & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{3\times3} & \cdots & \cdots & \cdots & \boldsymbol{I}_3 \end{bmatrix}}_{\boldsymbol{\Gamma}} \underbrace{\begin{bmatrix} \boldsymbol{q}_{k-1}^{cabin^p} \\ \boldsymbol{q}_{k-1}^{cabin^\omega} \\ \boldsymbol{q}_{k-1}^{chassis^p} \\ \boldsymbol{q}_{k-1}^{chassis^\omega} \\ \boldsymbol{q}_{k-1}^{\Delta\delta} \end{bmatrix}}_{\boldsymbol{q}_{k-1}},$$

where $\boldsymbol{I}_n$ is an identity matrix of size n and $\mathbf{0}_{n\times m}$ is a matrix of size $n \times m$ consisting of zeros. The quaternions are iteratively normalized to size 1, $\mathbf{q} := \frac{\mathbf{q}}{\|\mathbf{q}\|}$ and the matrices $\boldsymbol{A}_{CV}$ and $\boldsymbol{A}_{CA}$ are

$$\boldsymbol{A}_{CV} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{A}_{CA} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.19}$$

The motion noise $\boldsymbol{q}_k$ is modeled as Gaussian distributed with covariance matrix $\boldsymbol{Q}$ as

$$\boldsymbol{q} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}) \tag{4.20}$$

and the discrete covariance matrix $\boldsymbol{Q}_k$ can be calculated as

$$\boldsymbol{Q}_k = T\boldsymbol{\Gamma}\boldsymbol{Q}\boldsymbol{\Gamma}, \tag{4.21}$$

where T is the sample time, shown in equation (3.70).

4.2 Measurement model

There are three different kinds of measurements that need to be related to the states with three different measurement models, GNSS, IMU and velocity measurements.

For each measurement there will be two different models depending on which motion model that are used, one for the separated body (SB) and one for the merged body (MB).

The complete measurement model can be derived as

$$\begin{bmatrix} \boldsymbol{y}_k^{GNSS} \\ \boldsymbol{y}_k^{IMU} \\ \boldsymbol{y}_k^{v} \end{bmatrix} = \begin{bmatrix} h_{GNSS}(\boldsymbol{x}_k) \\ h_{IMU}(\boldsymbol{x}_k) \\ h_v(\boldsymbol{x}_k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{r}_k^{GNSS} \\ \boldsymbol{r}_k^{IMU} \\ \boldsymbol{r}_k^{v} \end{bmatrix}, \ \boldsymbol{r}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}). \tag{4.22}$$

The measurement noise covariance matrix R is described in Section 4.2.4.

4.2.1 GNSS measurements

These measurement models are based on the time difference theory found in Section 3.4.4. The basics are to relate the difference between two carrier phase measurements at two consecutive time epochs with the difference in receiver position over two epochs together with a difference in receiver clock bias. The measurement model is

$$\begin{aligned} y_k^{GNSS} &= \lambda(\Phi_k^{rec_i} - \Phi_{k-k_g}^{rec_i}) = h_{GNSS}(\boldsymbol{x}_k) + r_k^{GNSS} \\ &= (\rho_k - \rho_{k-k_g}) + c(\delta_k - \delta_{k-k_g}) + r_k^{GNSS}, \end{aligned} \tag{4.23}$$

where

$$\rho_k = \sqrt{(x_k^r - x_k^s)^2 + (y_k^r - y_k^s)^2 + (z_k^r - z_k^s)^2}, \tag{4.24}$$

$$\rho_{k-k_g} = \sqrt{(x_{k-k_g}^r - x_{k-k_g}^s)^2 + (y_{k-k_g}^r - y_{k-k_g}^s)^2 + (z_{k-k_g}^r - z_{k-k_g}^s)^2}, \tag{4.25}$$

$$k_g = T_{GNSS}/T_{sampleTime}. \tag{4.26}$$

The term k_g is used because the filter is sampled faster than the GNSS measurements, which are sampled with T_{GNSS} and the variables denoted with an s are known satellite positions.

These measurements are then related to the states but the receiver positions are not in the state vector. The position of each receiver is instead related to the cabin's position state. There will be two different models depending on how the cabin states are described one in global and one in the chassis coordinate frame.

The relations for the separated body model, where the cabin's position is global are

$$(\delta_k - \delta_{k-k_g}) = \Delta\hat{\delta}_k \ T_{GPS}, \tag{4.27}$$

$$\begin{bmatrix} x_k^r \\ y_k^r \\ z_k^r \end{bmatrix} = \hat{\boldsymbol{p}}_k^{cabin} + \mathfrak{Q}(\hat{\mathbf{q}}_k^{cabin}) \begin{bmatrix} x_{rec_i} \\ y_{rec_i} \\ z_{rec_i} \end{bmatrix}, \ i = 1, 2, 3, \tag{4.28}$$

$$\begin{bmatrix} x_{k-k_g}^r \\ y_{k-k_g}^r \\ z_{k-k_g}^r \end{bmatrix} = \hat{\boldsymbol{p}}_{k-k_g}^{cabin} + \mathfrak{Q}(\hat{\mathbf{q}}_{k-k_g}^{cabin}) \begin{bmatrix} x_{rec_i} \\ y_{rec_i} \\ z_{rec_i} \end{bmatrix}, \ i = 1, 2, 3. \tag{4.29}$$

The relations for the merged body model, where the cabin's position is described in the chassis coordinate frame are

$$(\delta_k - \delta_{k-k_g}) = \Delta\hat{\delta}_k \; T_{GPS}, \tag{4.30}$$

$$\begin{bmatrix} x_k^r \\ y_k^r \\ z_k^r \end{bmatrix} = \hat{\boldsymbol{p}}_k^{chassis} + \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})(\hat{\boldsymbol{p}}_k^{cabin} + \boldsymbol{R}_{zyx}(\hat{\boldsymbol{\varepsilon}}_k) \begin{bmatrix} x_{rec_i} \\ y_{rec_i} \\ z_{rec_i} \end{bmatrix}) \ , \ i = 1, 2, 3, \tag{4.31}$$

$$\begin{bmatrix} x_{k-k_g}^r \\ y_{k-k_g}^r \\ z_{k-k_g}^r \end{bmatrix} = \hat{\boldsymbol{p}}_{k-k_g}^{chassis} + \mathfrak{Q}(\hat{\mathbf{q}}_{k-k_g}^{chassis})(\hat{\boldsymbol{p}}_{k-k_g}^{cabin} + \boldsymbol{R}_z yx(\hat{\boldsymbol{\varepsilon}}_{k-k_g}) \begin{bmatrix} x_{rec_i} \\ y_{rec_i} \\ z_{rec_i} \end{bmatrix}) \ , \ i = 1, 2, 3. \tag{4.32}$$

Variables denoted with *rec* are constants describing the position of different receivers and states denoted with $k - k_g$ are old estimated states. It is not correct by the Markov chain assumption to use old states in Kalman filters but it is a pragmatic solution. How to use previous states in measurement models are left for discussion and future work.

4.2.2 IMU measurements

The IMU sensors measure the angular velocity and the acceleration of the body from the point they are installed. This means that the measurement models for the two IMUs may differ in appearance but they will follow the same basic principles. A gyroscope measurement may be modeled as

$$\boldsymbol{y} = \boldsymbol{\omega} + \boldsymbol{\mu} + \boldsymbol{r}_k^{Gyro}, \tag{4.33}$$

where ω is the local angular velocity and μ is a bias that can be estimated or calculated and then used as a constant depending on how much it will drift.

Accelerometer measurement are a bit different and may be modeled as

$$\boldsymbol{y} = \boldsymbol{R}_{zyx}(\ddot{\boldsymbol{x}} + g_0) + \boldsymbol{\mu} + \boldsymbol{r}_k^{Acc}, \tag{4.34}$$

where $\ddot{\boldsymbol{x}}$ is the acceleration of the point, g_0 is the gravity vector and $\boldsymbol{R}_{zyx}$ a rotation matrix. The IMU measurement consists of four measurements, two for each IMU.

$$\boldsymbol{y}_k^{IMU} = \begin{bmatrix} \boldsymbol{y}_k^{CabinGyro} \\ \boldsymbol{y}_k^{ChassisGyro} \\ \boldsymbol{y}_k^{CabinAcc} \\ \boldsymbol{y}_k^{ChassisAcc} \end{bmatrix} \tag{4.35}$$

Depending on which motion model that are used there will be two different sets of IMU measurement models with the same motivation as with GNSS.

The first one is with the separated body motion model where the global angular velocities in the state vector can be related to the local gyroscope measurements with a rotation matrix seen in equations (4.36)-(4.37), together with biases, μ, that are calculated and used as constants

$$\boldsymbol{y}_k^{CabinGyro} = \mathfrak{Q}(\hat{\mathbf{q}}_k^{cabin})^T \hat{\boldsymbol{\omega}}_k^{cabin} + \boldsymbol{\mu}^{CabinGyro} + \boldsymbol{r}_k^{CabinGyro}, \tag{4.36}$$

$$\boldsymbol{y}_k^{ChassisGyro} = \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} + \boldsymbol{\mu}^{ChassisGyro} + \boldsymbol{r}_k^{ChassisGyro}. \tag{4.37}$$

The global acceleration states of the bodies need to be transformed to the local acceleration of the point where the IMUs are placed. This is done by using equation (3.42) from the theory chapter where the acceleration and the velocity terms of the IMU in the global coordinate frame is zero. The acceleration measurement models are then

$$\begin{aligned}\boldsymbol{y}_k^{CabinAcc} = \mathfrak{Q}(\hat{\mathbf{q}}_k^{cabin})^T \Big[\ddot{\hat{\boldsymbol{p}}}_k^{cabin} + [\dot{\hat{\boldsymbol{\omega}}}_k^{cabin} \times \mathfrak{Q}(\hat{\mathbf{q}}_k^{cabin}) \boldsymbol{p}^{IMU_1}] \ \cdots \\ + \Big[\hat{\boldsymbol{\omega}}_k^{cabin} \times [\hat{\boldsymbol{\omega}}_k^{cabin} \times \mathfrak{Q}(\hat{\mathbf{q}}_k^{cabin}) \boldsymbol{p}^{IMU_1}] \Big] + g \frac{\hat{\boldsymbol{p}}_k^{cabin}}{\| \hat{\boldsymbol{p}}_k^{cabin} \|} \Big] + \boldsymbol{\mu}^{CabinAcc} + \boldsymbol{r}_k^{CabinAcc}\end{aligned} \tag{4.38}$$

and

$$\begin{aligned}\boldsymbol{y}_k^{ChassisAcc} = \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \Big[\ddot{\hat{\boldsymbol{p}}}_k^{chassis} + [\dot{\hat{\boldsymbol{\omega}}}_k^{chassis} \times \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis}) \boldsymbol{p}^{IMU_2}] \ \cdots \\ + \Big[\hat{\boldsymbol{\omega}}_k^{chassis} \times [\hat{\boldsymbol{\omega}}_k^{chassis} \times \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis}) \boldsymbol{p}^{IMU_2}] \Big] + g \frac{\hat{\boldsymbol{p}}_k^{chassis}}{\| \hat{\boldsymbol{p}}_k^{chassis} \|} \Big] + \boldsymbol{\mu}^{ChassisAcc} + \boldsymbol{r}_k^{ChassisAcc}.\end{aligned} \tag{4.39}$$

The second one is with the merged body motion model where the measurement models used for the IMU on the chassis is the same as for the separated body model seen in equations (4.37) and (4.39). However, the other models are changed since the cabin states depend on the chassis states, hence the cabin states are no longer global. The gyroscope model will be derived using equation (3.35). The global angular velocities of the chassis are first rotated to the local coordinate frame of the chassis, the same as the cabin's angular velocity is in. The sum of these two is then rotated to the local coordinate frame of the cabin to match the angular measured by the gyroscope,

$$\boldsymbol{y}_k^{CabinGyro} = \boldsymbol{R}_{zyx}(\hat{\boldsymbol{\varepsilon}}_k)^T \Big[\hat{\boldsymbol{\omega}}_k^{cabin} + \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} \Big] + \hat{\boldsymbol{\mu}}_k^{cabin} + \boldsymbol{r}_k^{CabinGyro}. \tag{4.40}$$

The accelerometer model becomes a bit more complicated. Instead of having two position vectors, $\boldsymbol{r}$ as in equation (3.40) there will be a differentiation of three vectors. The chassis states will be rotated from global to the local coordinate frame of the chassis, where the sum of all terms in the chassis coordinate frame will be rotated to the local coordinate frame of the cabin to match the acceleration of the

cabin IMU

$$\begin{aligned}\boldsymbol{y}_k^{CabinAcc} = \boldsymbol{R}_{zyx}(\hat{\boldsymbol{\varepsilon}}_k)^T \Big[& \mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \ddot{\hat{\boldsymbol{p}}}_k^{chassis} + \ddot{\hat{\boldsymbol{p}}}_k^{cabin} \quad \cdots \\ & + [\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \dot{\hat{\boldsymbol{\omega}}}_k^{chassis} \times \hat{\boldsymbol{p}}_k^{cabin}] + [2\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} \times \dot{\hat{\boldsymbol{p}}}_k^{cabin}] \quad \cdots \\ & + [\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} \times [\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} \times \dot{\hat{\boldsymbol{p}}}_k^{cabin}]] \cdots \\ & + [[\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \dot{\hat{\boldsymbol{\omega}}}_k^{chassis} + \dot{\hat{\boldsymbol{\omega}}}_k^{cabin}] \times \mathcal{R}_{zyx}(\hat{\boldsymbol{\varepsilon}}_k) \boldsymbol{p}^{IMU_1}] \cdots \\ & + \Big[[\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} + \hat{\boldsymbol{\omega}}_k^{cabin}] \times [[\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \hat{\boldsymbol{\omega}}_k^{chassis} + \hat{\boldsymbol{\omega}}_k^{cabin}] \times \boldsymbol{R}_{zyx}(\hat{\boldsymbol{\varepsilon}}_k) \boldsymbol{p}^{IMU_1}] \Big] \cdots \\ & + g\mathfrak{Q}(\hat{\mathbf{q}}_k^{chassis})^T \frac{\hat{\boldsymbol{p}}_k^{chassis}}{\|\hat{\boldsymbol{p}}_k^{chassis}\|} \Big] + \boldsymbol{\mu}^{CabinAcc} + \boldsymbol{r}_k^{CabinAcc}. \end{aligned} \tag{4.41}$$

The biases are calculated and used as constants as in the previous case.

4.2.3 Velocity measurements

The velocity measurements are related to the velocity of the truck which is assumed to be the norm of the chassis velocity states

$$\boldsymbol{y}_k^v = \|\dot{\hat{\boldsymbol{p}}}_k^{chassis}\| + \boldsymbol{r}_k^v. \tag{4.42}$$

This means that it will be the same for both motion models.

4.2.4 Measurement noise covariance matrix

All measurements have a noise scalar/vector denoted as r_k^i where the complete noise vector can be seen in equation (4.22). The measurement noise covariance matrix R can be calculated as

$$\boldsymbol{R} = \boldsymbol{C}\boldsymbol{R}_{mat}\boldsymbol{C}^T, \tag{4.43}$$

$$\boldsymbol{R}_{mat} = \boldsymbol{I}_n \begin{bmatrix} \boldsymbol{r}_k^{GNSS} \\ \boldsymbol{r}_k^{IMU} \\ \boldsymbol{r}_k^v \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_k^{GNSS} & 0 & 0 \\ 0 & \boldsymbol{r}_k^{IMU} & 0 \\ 0 & 0 & \boldsymbol{r}_k^v \end{bmatrix}, \tag{4.44}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{GNSS} & 0 & 0 \\ 0 & \boldsymbol{C}_{IMU} & 0 \\ 0 & 0 & \boldsymbol{C}_v \end{bmatrix}. \tag{4.45}$$

The $\boldsymbol{C}$ matrix connects the measurements with correct noise and $\boldsymbol{I}_n$ is an identity matrix with the same size as the noise vector.

5
Results

The result chapter describes the performance of the filter and the characteristics of the sensors that are used in both simulation and with real data measurements. The chapter is divided in four sections, reference system - VBOX, sensor characteristics, filter with simulated data and filter with real data. The results are mainly focused on simulated data because the reference system was not able to provide enough information as first intended. The reason to this is explained in Section 5.1.

The sensor characteristics section provides results regarding the performance of the sensors such as variances and noise characteristics. The variances are in turn used in the measurement models for every filter in both real and simulated environments. The sample time for both filters are chosen to be 1 ms to handle asynchronous IMU measurements that have sample time 4 ms. The GNSS measurements have sample time 200 ms and the velocity measurements have 40ms.

5.1 Reference system - VBOX

The VBOX reference system with the GNSS dual antenna together with the base station and the IMU, described in Section 2.1.5, could not be used as intended. The system encountered problems with initiating the IMU and could therefore not provide reference angles about all axes as desired (see Table 5.1). The system is still able to provide reference in position, heading and roll angel but not in the pitch angle. Another remark is that this reference system is only able to provide information about the cabin's motion in a global coordinate frame and not the cabin's motion relative the chassis nor the chassis' motion.

Table 5.1: VBOX accuracy with GNSS dual antenna and base station.

	Accuracy	**Desired**
Absolute position	2 cm	2 cm
Heading	0.1°	0.06° RMS
Roll angle	$< 0.047°$ RMS	$< 0.047°$ RMS
Pitch angle	none	0.06° RMS

5.2 Sensor characteristics

This section describes the characteristics of the sensors such as accuracy, noise and performance. The section is divided into three parts, one for each sensor. The data used for characterizing the GNSS and IMU sensors are gathered during 2 minutes in a stationary position with the engine running. The velocity data is gathered when driving at approximately constant acceleration because no constant velocity was found in the real data and that the sensors outputs zero when the truck stands still.

5.2.1 GNSS antennas

The noise variances of the carrier phases are supplied by the U-Blox sensor in each epoch and for each satellite and GNSS receiver. The mean of each receivers variances are shown in Table 5.2, where receivers are denoted as REC.

Table 5.2: Mean variance of the GNSS noise characteristics in meters $[m]$.

	REC1	**REC2**	**REC3**
Mean variance	0.00085053	0.00076972	0.00086828

A discovery made from using these GNSS sensors is that the sample frequency tends to diverge from the assigned 5 Hz when exposed to movement (see Figure 5.1).

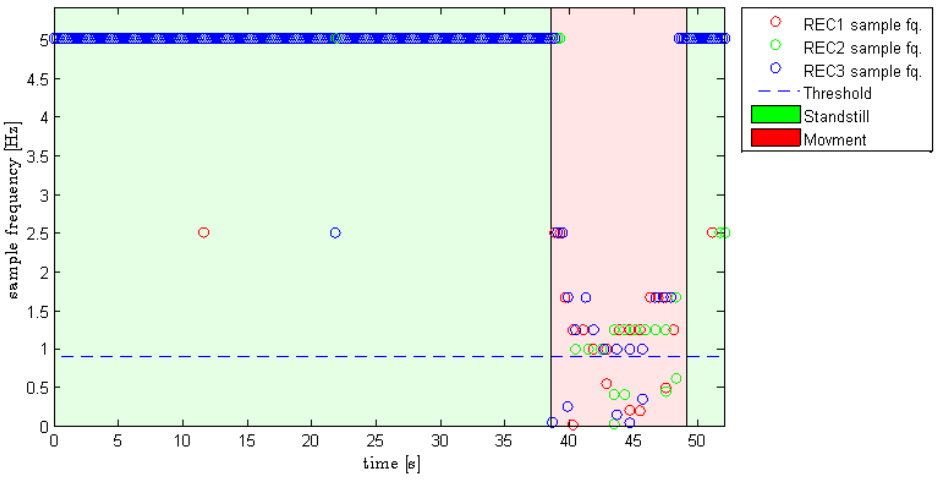


Figure 5.1: A description over varying sample frequency of GNSS signals.

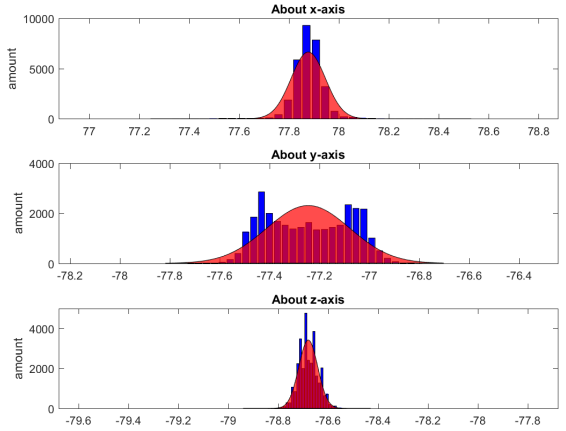
However, the sampling frequency in simulation is fixed to 5 Hz.

5.2.2 Inertial measurement unit - IMU

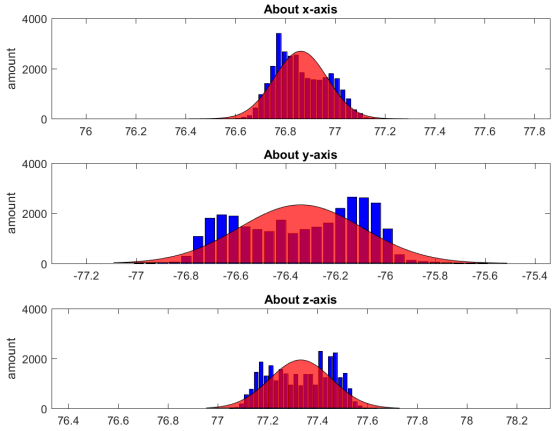
The IMU sensor characteristics starts with describing the gyroscopes and follows with an explanation of the accelerometers where the cabin mounted IMU sensor is denoted as 1 and the chassis as 2.

Gyroscope

The noise of the gyroscopes is shown in Figure 5.2 as histograms together with Gaussian approximations of the distributions. The variances and mean values of the noise are stated in Table 5.3.



(a) Noise characteristics of gyroscopes for IMU 1, (■ noise histograms, ■ approximated distribution)



(b) Noise characteristics of gyroscopes for IMU 2, (■ noise histograms, ■ approximated distribution)

Figure 5.2: Noise characteristics for integrated gyroscopes.

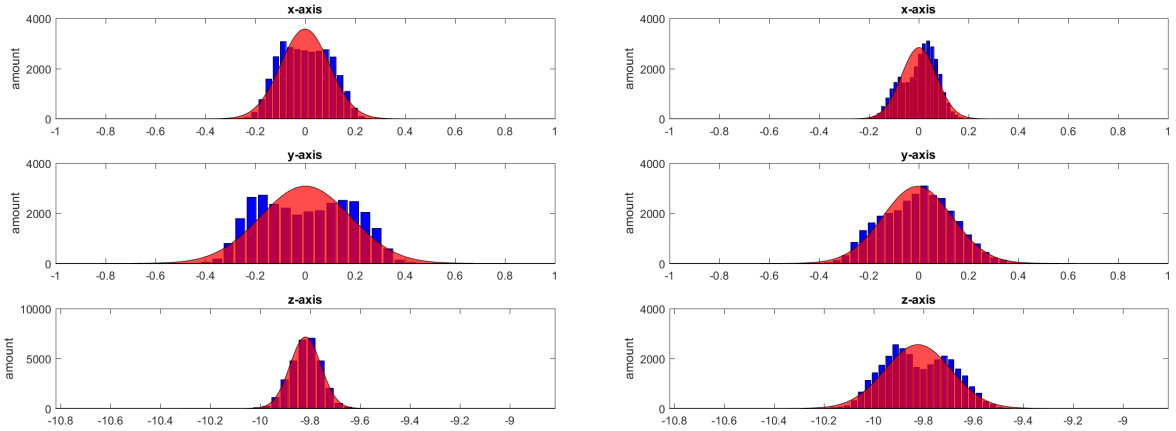
It can be seen that the approximations of the distributions are similar to the histograms about the x- and z-axis for IMU 1 but not for the others, (see Figure 5.2).

Table 5.3: The variance and mean values of gyroscopes noise characteristics in $[deg/s]$.

	About	IMU1	IMU2
Variances	x	0.0047282	0.01127
	y	0.027785	0.061623
	z	0.0014584	0.015652
Mean values	x	77.8773	76.8622
	y	-77.2444	-76.3403
	z	-78.6804	77.3339

Accelerometer

The noise of the accelerometer is shown in Figure 5.3 as histograms together with Gaussian approximations of the distributions. The variances and mean values of the noise are stated in Table 5.4.



(a) Noise characteristics of accelerometers for IMU 1, (■ noise histograms, ■ approximated distribution)

(b) Noise characteristics of accelerometers for IMU 2, (■ noise histograms, ■ approximated distribution)

Figure 5.3: Noise characteristics for integrated accelerometers.

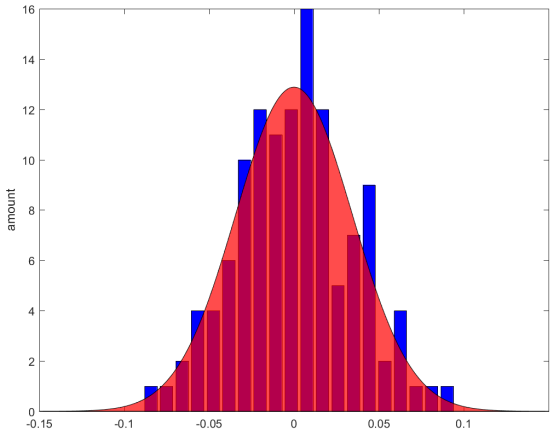
It can be seen that the approximations of the distributions are similar to the histograms about the z-axis for IMU 1 and the y-axis for IMU 2 but not for the others, (see Figure 5.3).

Table 5.4: The variance and mean values of the accelerometers noise characteristics in $[m/s^2]$.

		IMU1	**IMU2**
Variances	x	0.0095881	0.0049537
	y	0.032604	0.019439
	z	0.0038512	0.01782
Mean values	x	-0.0010449	0.0010867
	y	0.00084368	-0.0054679
	z	-9.8192	-9.8229

5.2.3 Velocity sensor

The noise of the velocity sensor is shown in Figure 5.4 as histograms together with Gaussian approximations of the distributions. The variance of the noise is calculated with a set of data collected during an approximately constant acceleration where the trends are removed. The variance is 0.0012 $[m/s]$.



(■ noise histograms, ■ approximated distribution)

Figure 5.4: Noise characteristics of velocity sensor.

5.3 Filter with simulated data

This section contains estimation result in simulation from both the filter with the separated body model (SB) and the filter with the merged body model (MB). As described in Section 4.1, the separated body model may not be a great model for estimating the relative position of the cabin and the chassis due to no connection between the two parts. However, it may provide information about the influence of adding motion constraints or not. To not lose focus from the merged body results that are more relevant to the main subject, these results are mainly found in appendix A.
Both filters are tested with a scenario where the truck is stationary for 10 seconds, drives forward with constant acceleration in 20 seconds where it starts to perform S-turns and then ends with as hard brake.

To make the illustrations in Figure 5.5 and Figure 5.12 to 5.18 more intuitive, the states have been rotated to a local coordinate frame by using the rotation between ECEF to ENU, stated in Section 3.1.3. The orientation states are shown in Euler angles, that is, θ- roll, φ- pitch and ψ- yaw, described in Section 3.2.1.

5.3.1 Separated bodies

The results in this section are with the motion model where there is no connection between the two bodies (SB), described in Section 4.1.1. The intention with the results in this section are to show what happens if there are no motion constraints available between two bodies. This means that both the cabin and the chassis states are in the global coordinate frame and the orientation are defined with quaternions.

Figure 5.5 and 5.6 displays how the filter estimates the position and the orientation with present measurements from the GNSS, IMU and velocity sensors. It is noticeable that these measurements are able to estimate the position and the motions of the angles for the cabin, though with an offset when the truck starts to move. The chassis position has a drift and a step in the z-axis and offsets are visible in z- and y-axis. The orientation estimation for the chassis is similar to the cabin.

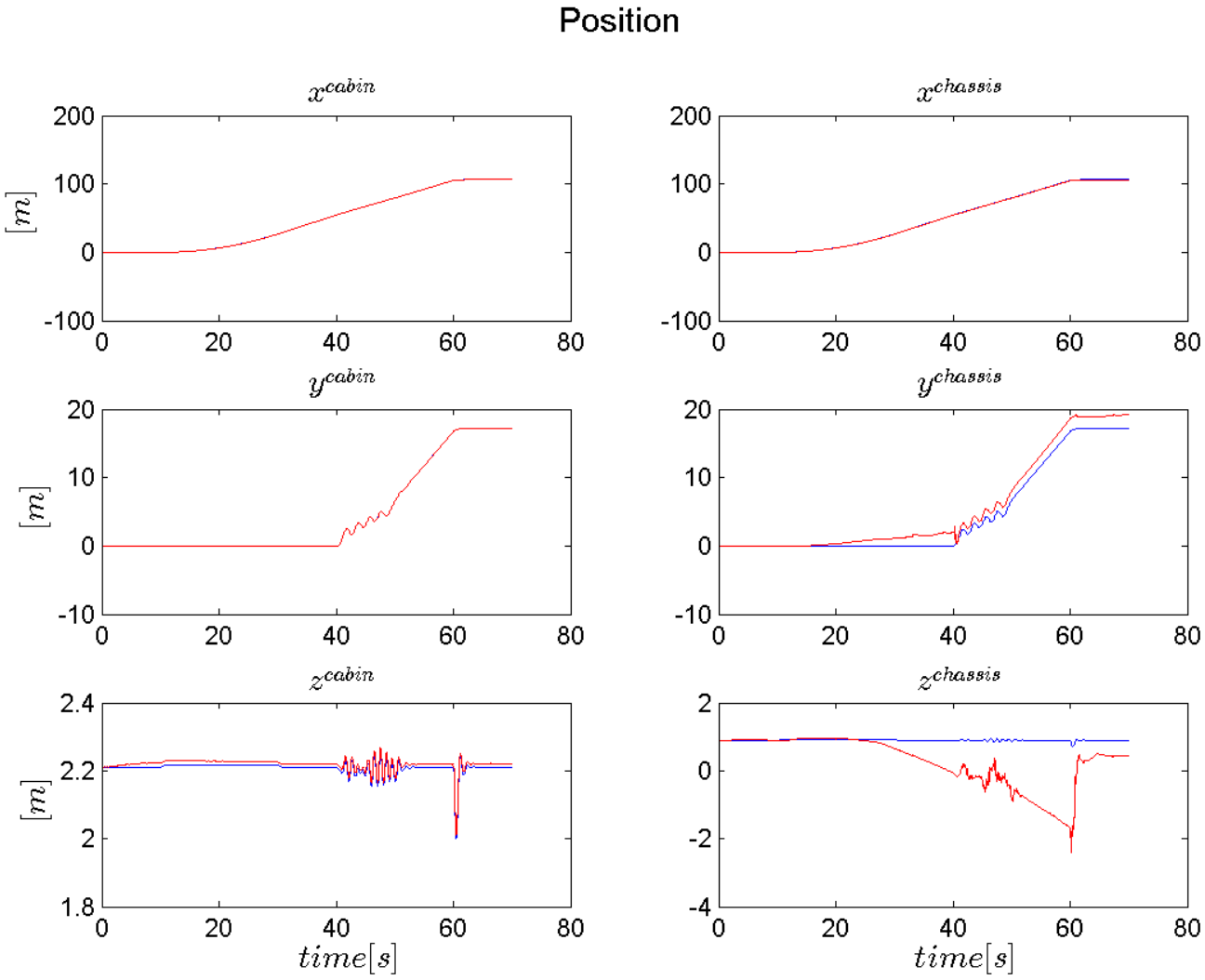


Figure 5.5: Estimated position states for the SB model using GNSS, IMU and velocity sensors.

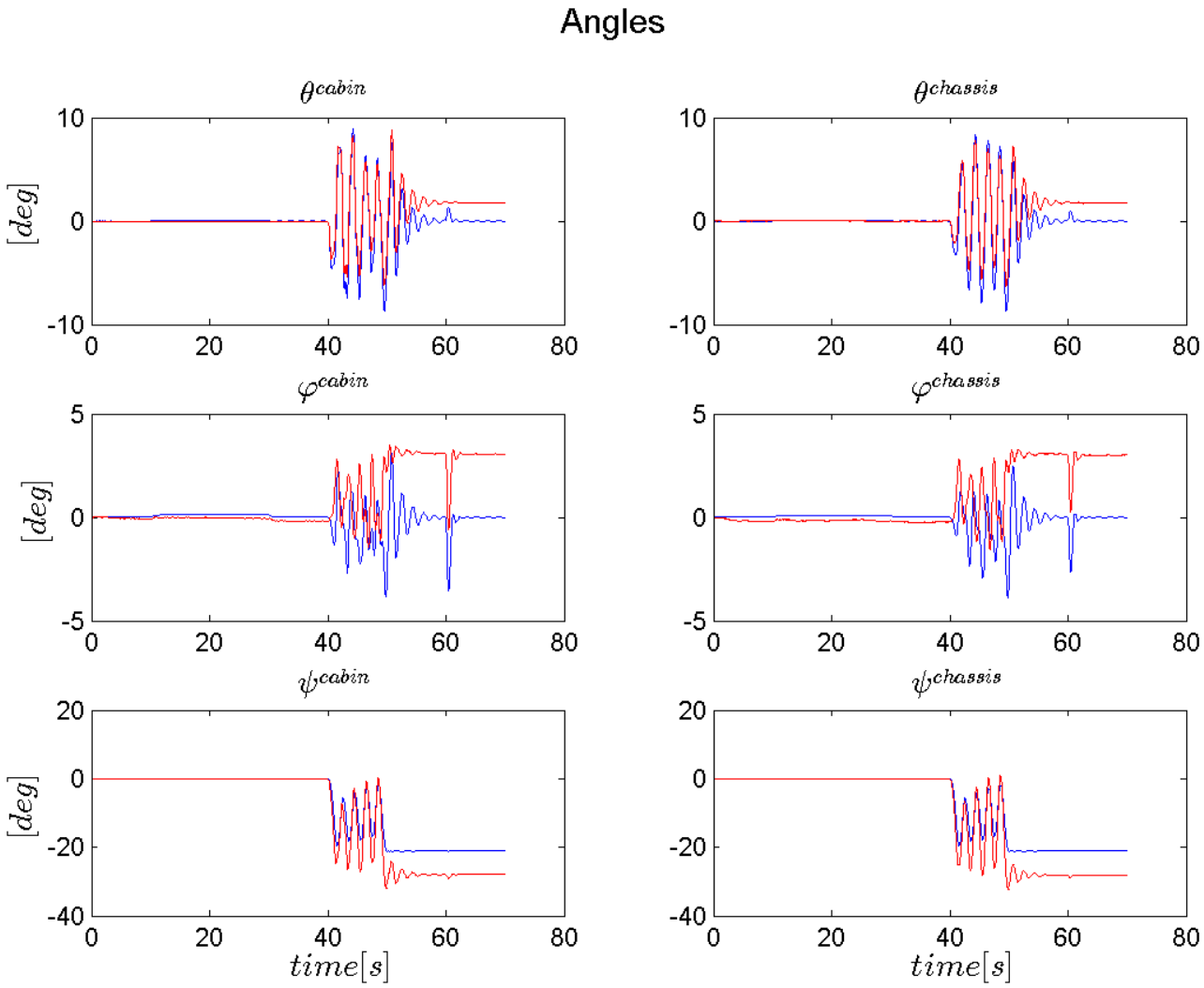
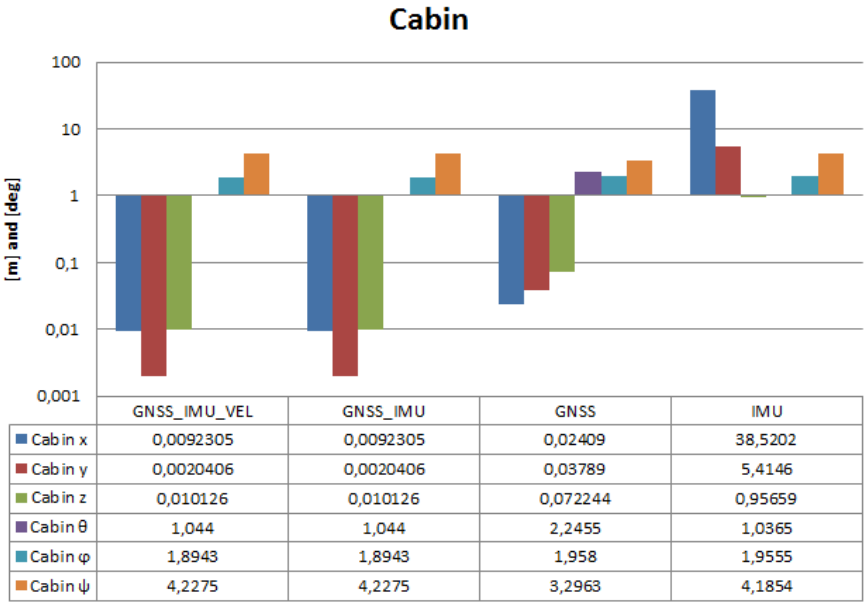


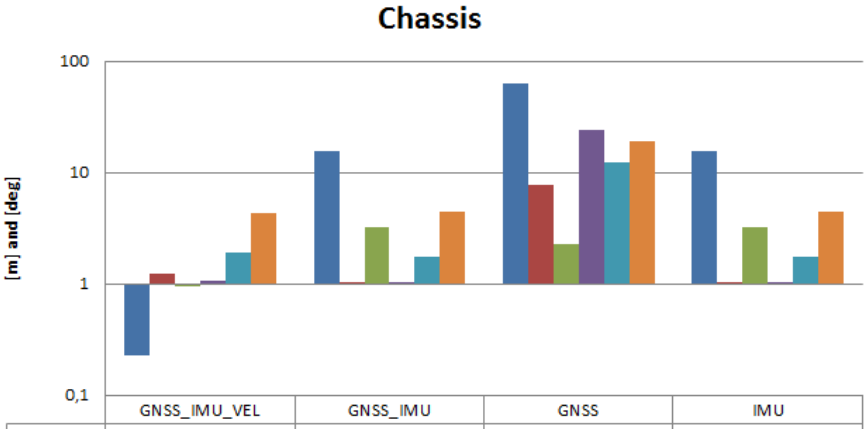
Figure 5.6: Estimated orientations states shown with Euler Angles for the SB model using GNSS, IMU and velocity sensors.

The evaluation of this filter can be seen in Figures 5.7 - 5.8 where the first one is for the cabin states and the second for the chassis states. These figures illustrate the root mean square (RMS) error between the references and the estimated states.



	GNSS_IMU_VEL	GNSS_IMU	GNSS	IMU
Cabin x	0,0092305	0,0092305	0,02409	38,5202
Cabin y	0,0020406	0,0020406	0,03789	5,4146
Cabin z	0,010126	0,010126	0,072244	0,95659
Cabin θ	1,044	1,044	2,2455	1,0365
Cabin φ	1,8943	1,8943	1,958	1,9555
Cabin ψ	4,2275	4,2275	3,2963	4,1854

Figure 5.7: RMS value for different sensor combinations for the cabin with the SB model.



	GNSS_IMU_VEL	GNSS_IMU	GNSS	IMU
Chassis x	0,22625	15,6849	62,9727	15,6841
Chassis y	1,2182	1,0257	7,8014	1,0246
Chassis z	0,95801	3,2615	2,2706	3,2613
Chassis θ	1,0539	1,0224	24,1572	1,0224
Chassis φ	1,8882	1,7463	12,4739	1,7463
Chassis ψ	4,356	4,4147	18,9159	4,4147

Figure 5.8: RMS value for different sensor combinations for the chassis with the SB model.

It can be seen that there is no difference in estimation performance for the cabin states when adding a velocity sensor but it is a clear improvement for the chassis. Likewise, the GNSS has no impact on the chassis but improves both the position and orientation for the cabin. Additional results using this filter with different sensor combinations can be seen in Appendix A.

5.3.2 Merged bodies

Following results in this section are with the motion model where two bodies are merged (MB) with springs and dampers, described in Section 4.1.2. The cabin states are in the coordinate frame of the chassis and the orientation is defined with Euler angels, where it is important to notice that these states are relative to the chassis. The chassis states are in the global coordinate frame and the orientation is defined with quaternions. The values of the coefficients used for the springs and dampers (see Table 5.5) in this model are selected by trial and error.

The first part of this section describes the motion model's influence on the cabin states without any measurements to provide result over the accuracy of the motion model itself, where the chassis states are considered known (see Figures 5.9 - 5.10). The second part describes the estimation results using different combination of sensor information, each denoted by a bold text describing which measurements that are used. The chapter then concludes with an evaluation of these sensor combinations described with a bar chart.

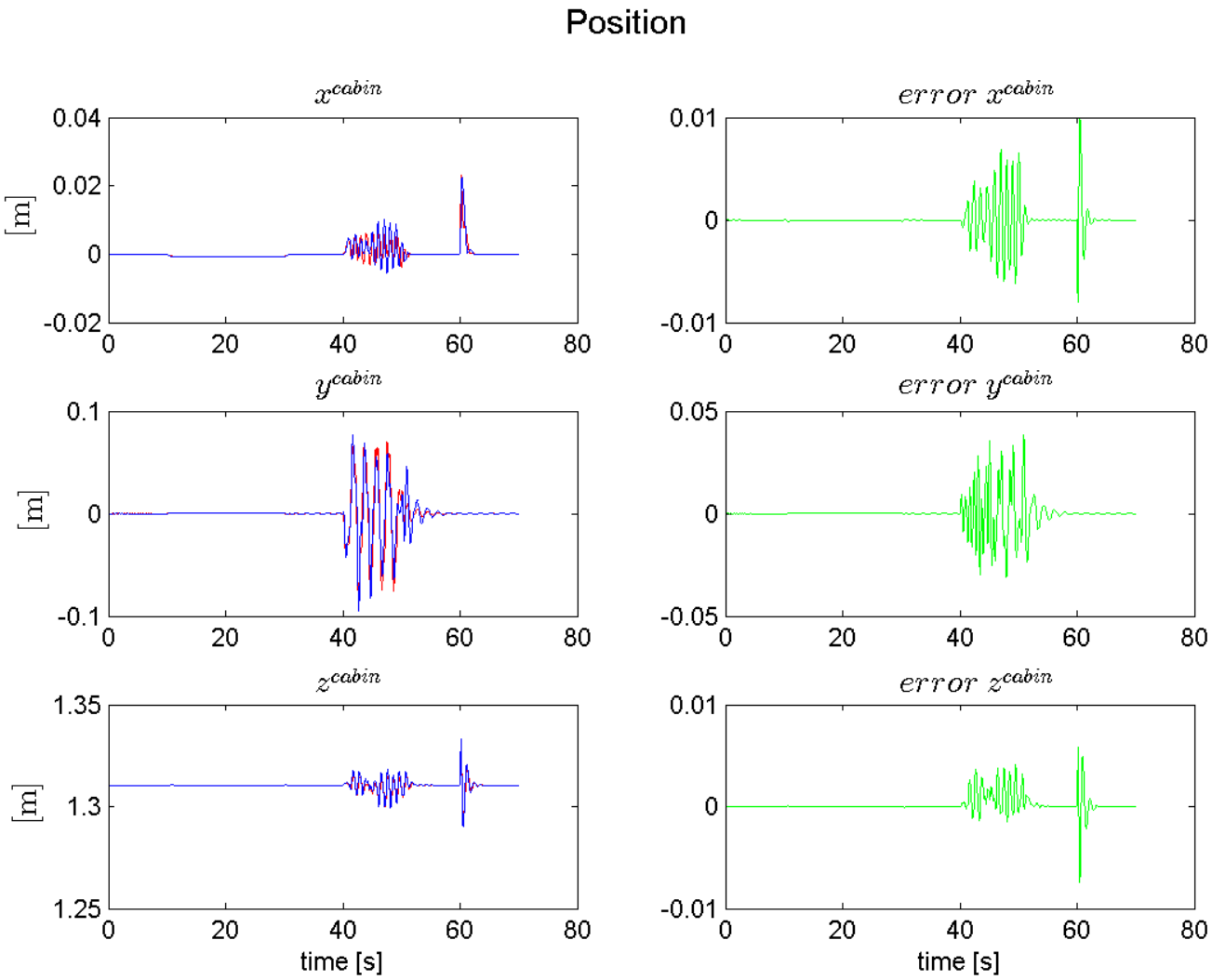
Table 5.5: Used coefficients for the MB model.

	Value	**Unit**
k_x	140	N/m
k_y	90	N/m
k_z	140	N/m
c_x	25	Ns/m
c_x	10	Ns/m
c_x	25	Ns/m
m	1	kg
k_θ	110	N/deg
k_φ	250	N/deg
k_psi	1500	N/deg
c_θ	10	Ns/deg
c_φ	50	Ns/deg
c_ψ	70	Ns/deg
I_{xyz}	1	kgm^2
x_0	0	m
y_0	0	m
z_0	1.3015	m

Motion model evaluation

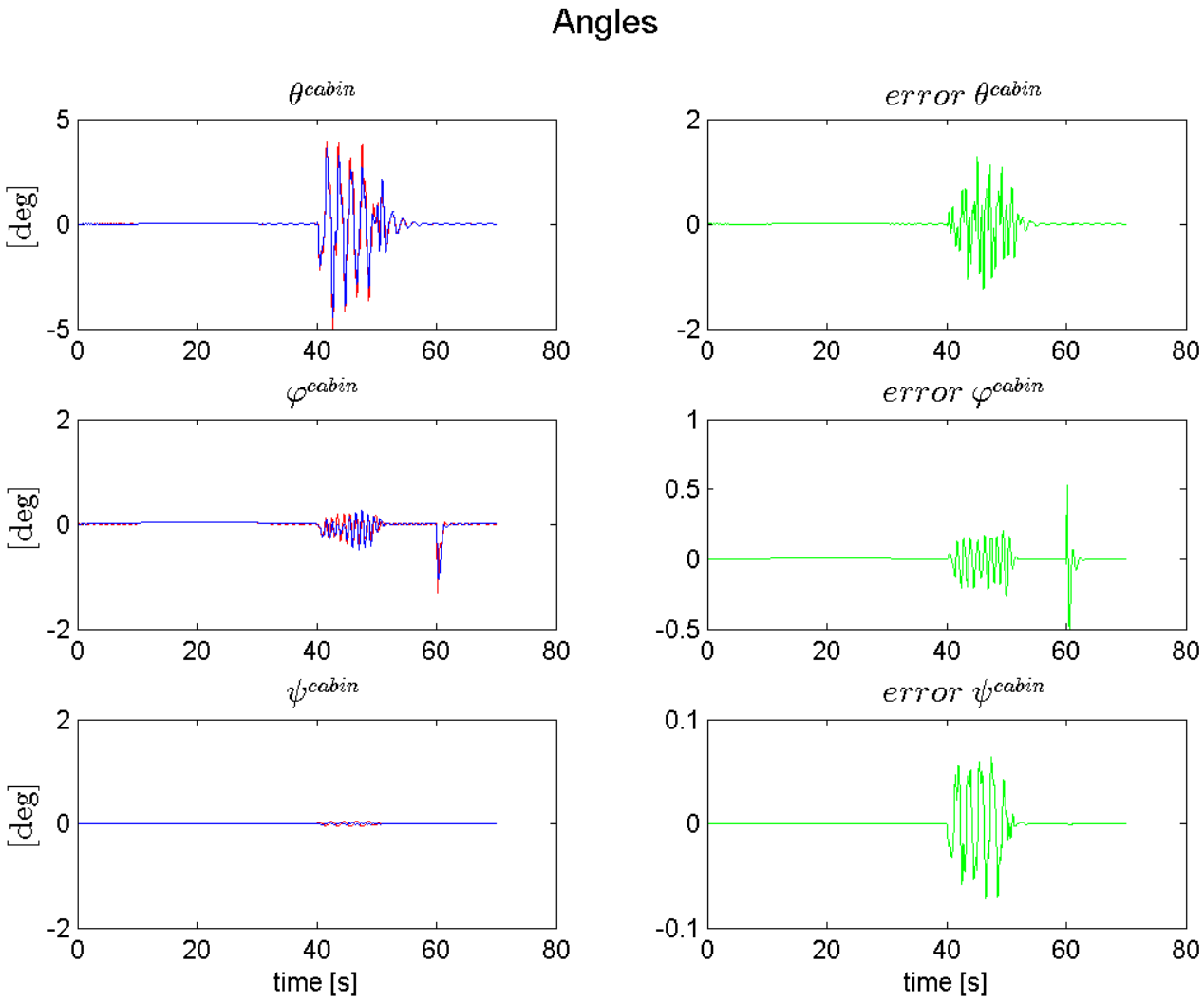
The performance of the motion model for the cabin is evaluated using the values in Table 5.5 together with simulated translational- and angular accelerations of the chassis. These accelerations are considered known and used as inputs to the motion model to predict the relative motion between the chassis and the cabin using no measurements at all. The cabin's position and angles are then compared with simulated references (see Figure 5.9 - 5.10). The root mean square (RMS) errors

between output and the simulated reference are seen in Table 5.6.



- estimated states, - references, - error.

Figure 5.9: Predicted position states for the cabin together with the MB model.



- estimated states, - references, - error

Figure 5.10: Predicted orientation states shown with Euler Angles for the cabin together with the MB model.

Table 5.6: Position and orientation RMS value for the prediction.

	RMS	**Unit**
x	0,001445	m
y	0,006618	m
z	0,000898	m
θ	0,20263	deg
φ	0,0592	deg
ψ	0,014138	deg

GNSS

Figure 5.11 and 5.12 displays how the filter estimates the position and the orientation with measurements from GNSS sensors. It is noticeable that these measurements are able to estimate both the cabin and the chassis states, though there is an offset after the truck starts to move in the chassis angle states and in the z-axis.

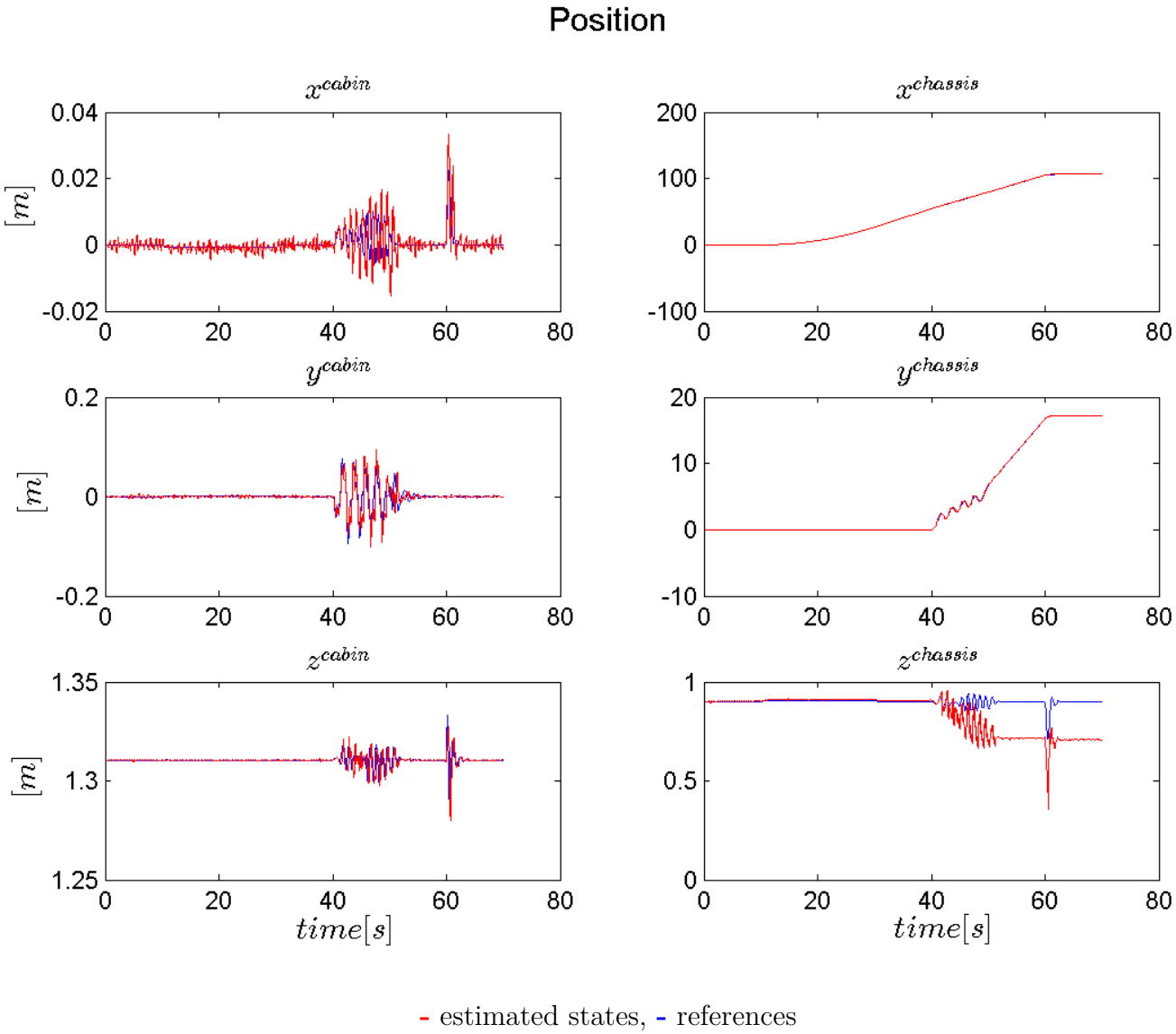
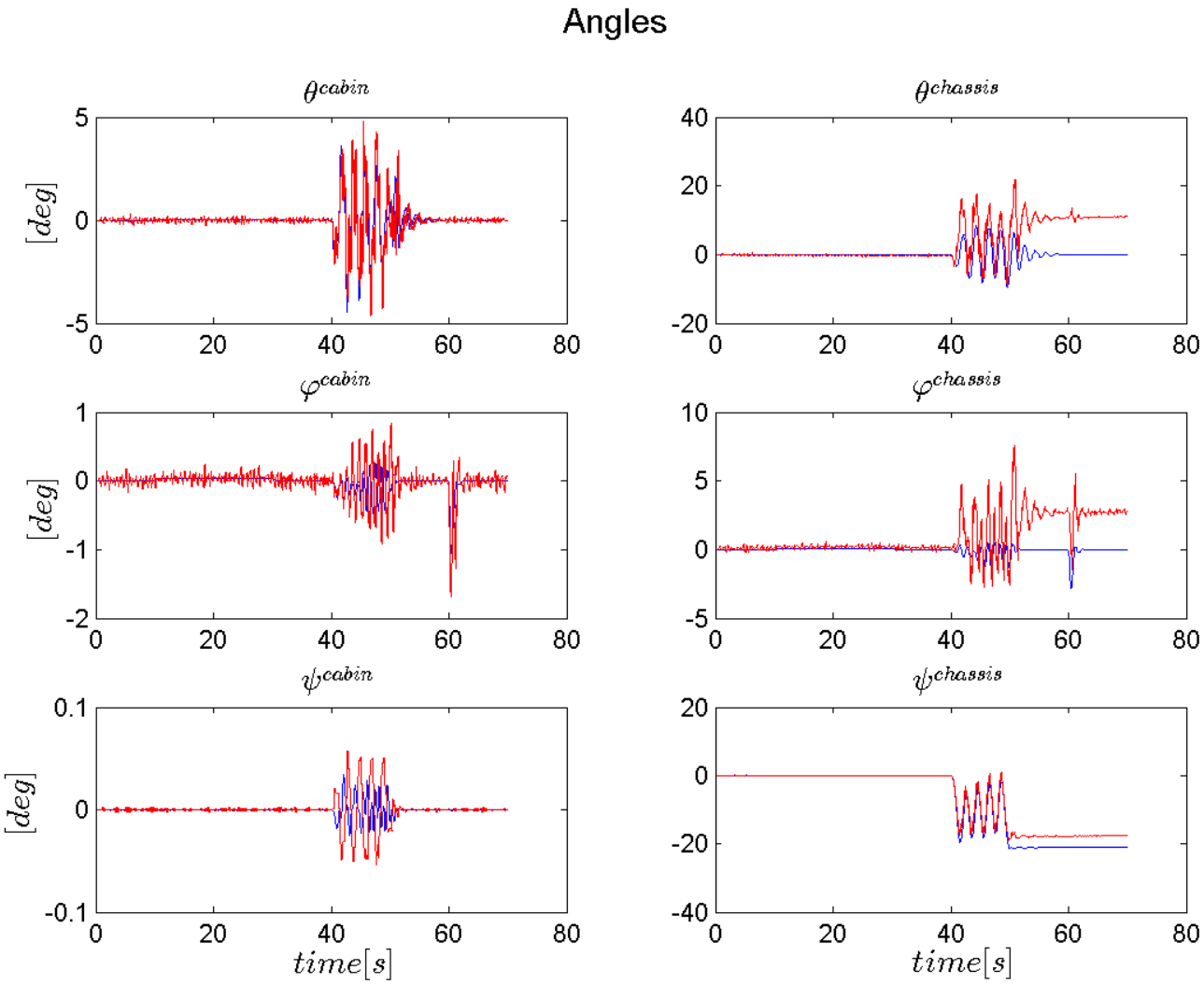


Figure 5.11: Estimated position states for the MB model using GNSS sensors.



- estimated states, - references

Figure 5.12: Estimated orientation states shown with *Euler Angles* for the MB model using GNSS sensors.

IMU

Figure 5.13 and 5.14 displays how the filter estimates the position and the orientation with only measurements from the IMU sensors. It is noticeable that these measurements are able to estimate the cabin states but fails for the chassis position. The chassis orientation states gets an offset after the truck starts to move.

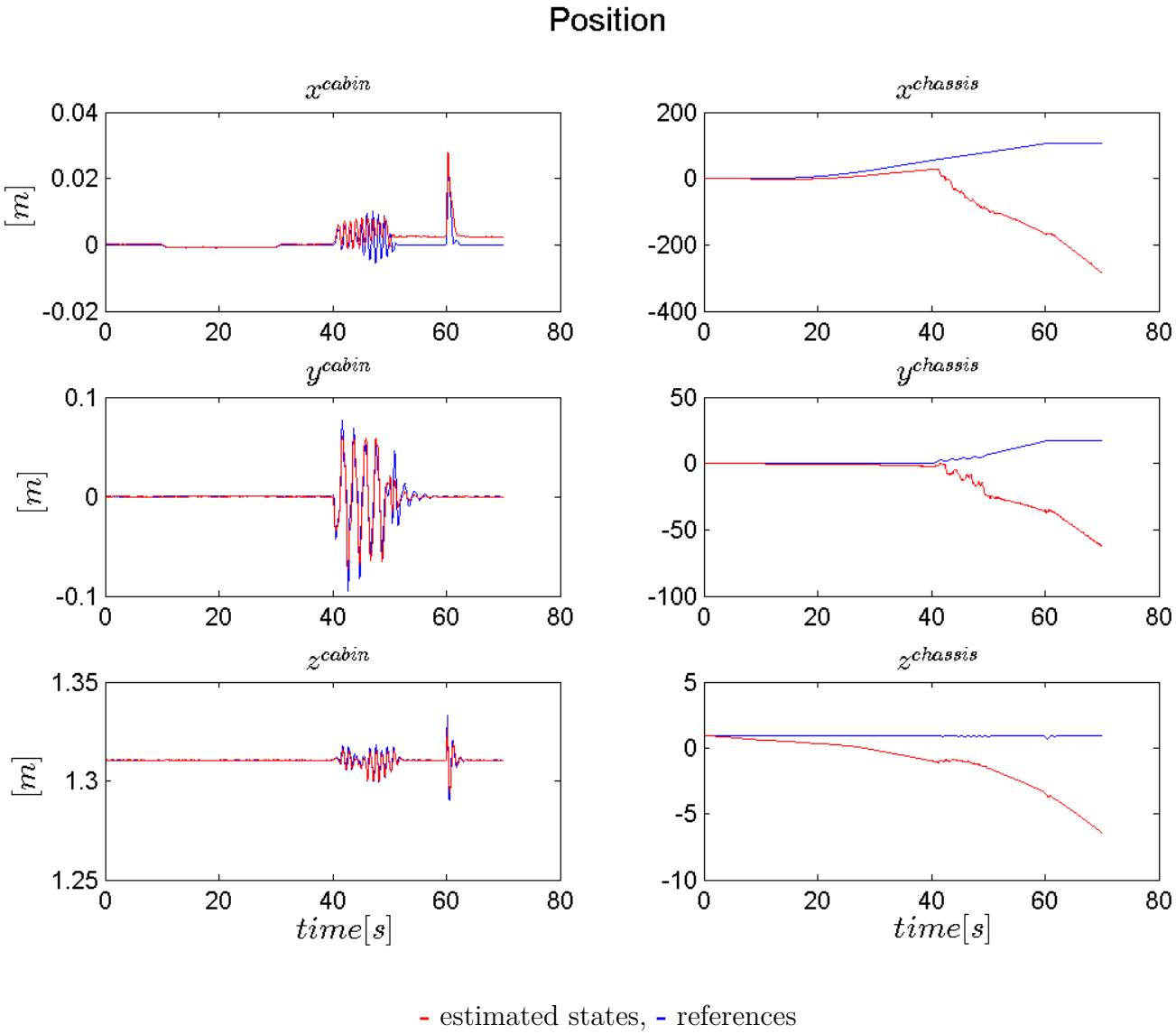


Figure 5.13: Estimated position states for the MB model using IMU sensors.

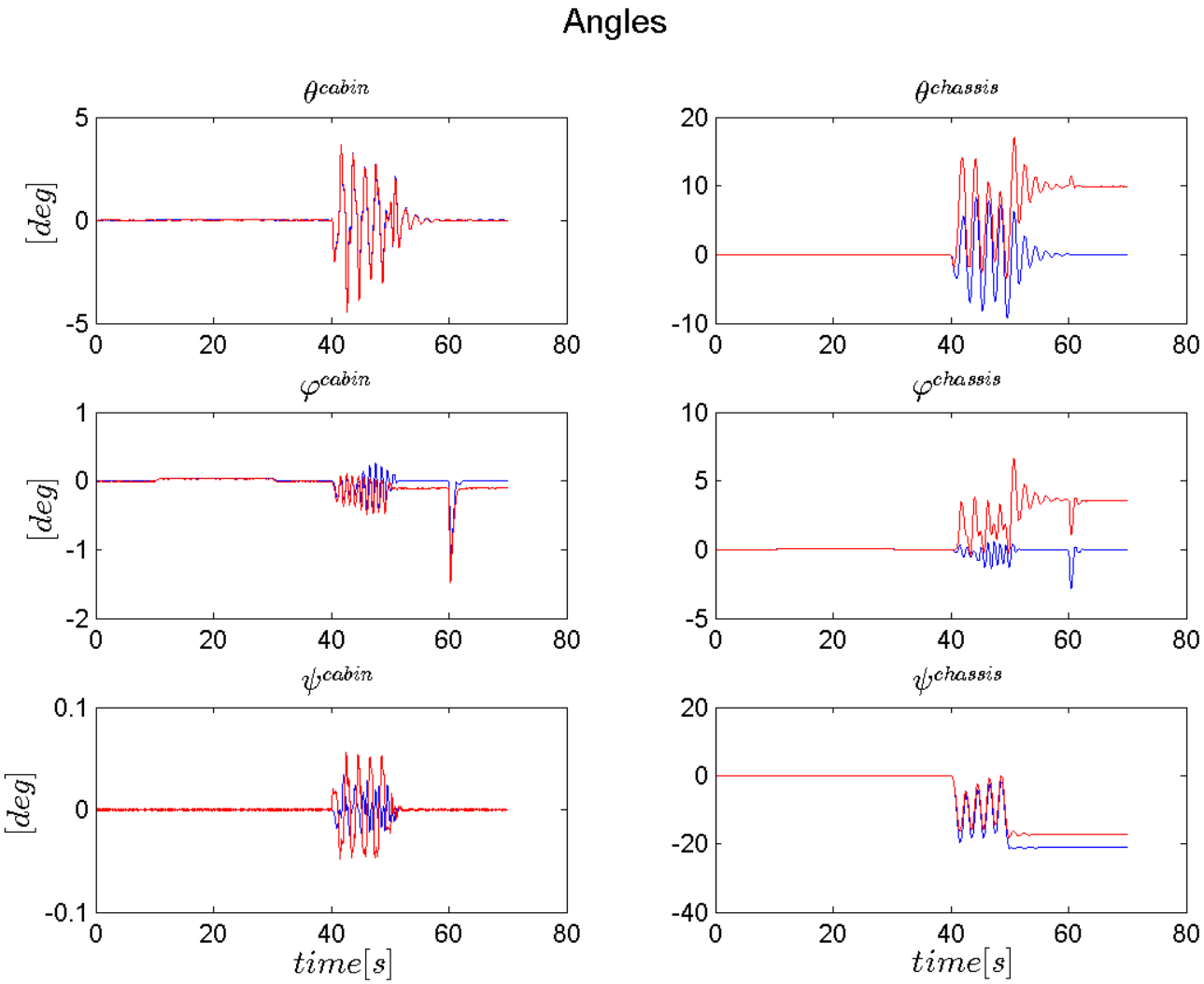


Figure 5.14: Estimated orientation states shown with *Euler Angles* for the MB model using IMU sensors.

GNSS and IMU

Figure 5.15 and 5.16 displays how the filter estimates the position and the orientation with measurements from GNSS and IMU sensors. It is noticeable that these measurements are able to estimate the cabin states and the chassis position, though there is an offset in the chassis orientation after the truck starts to move.

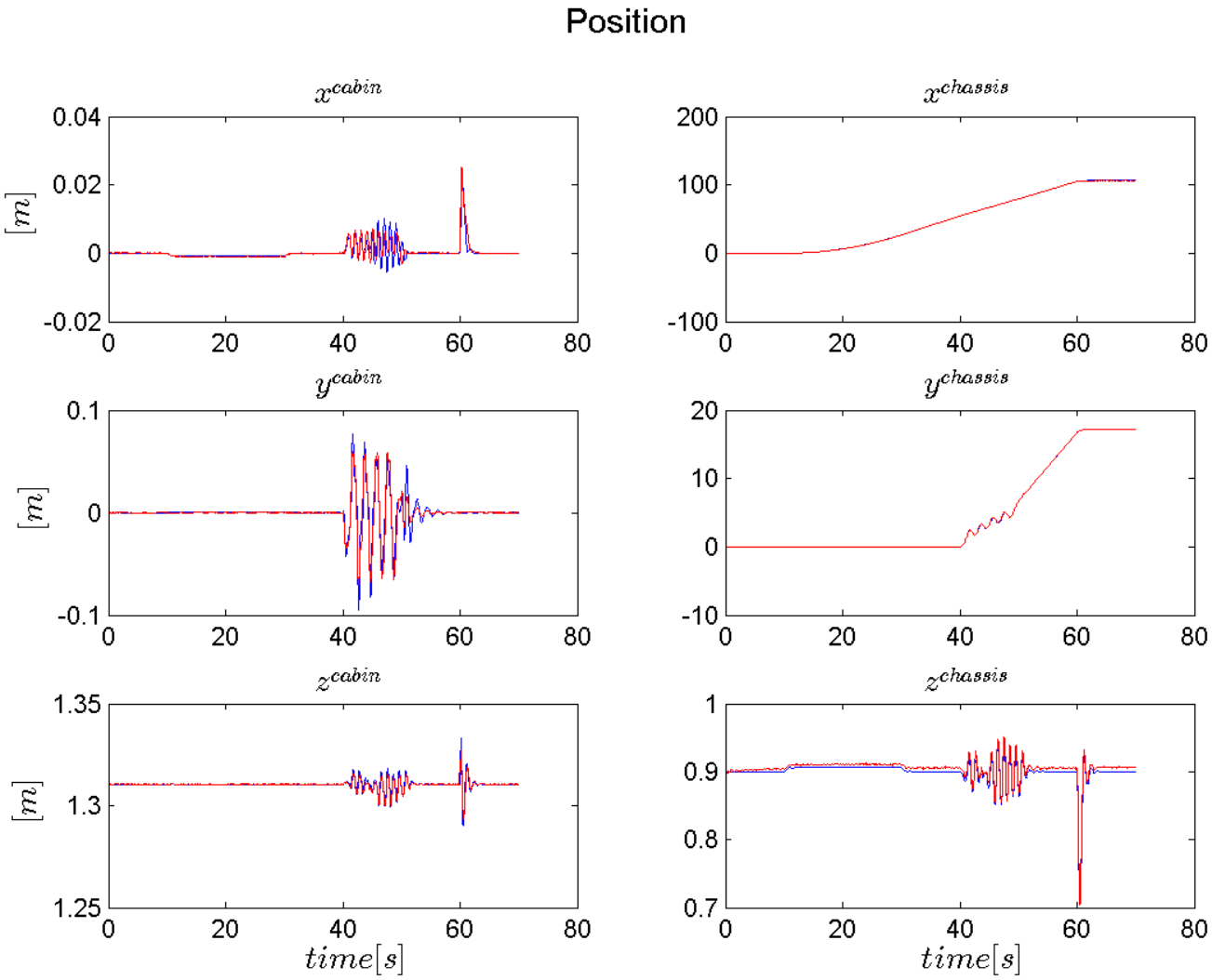
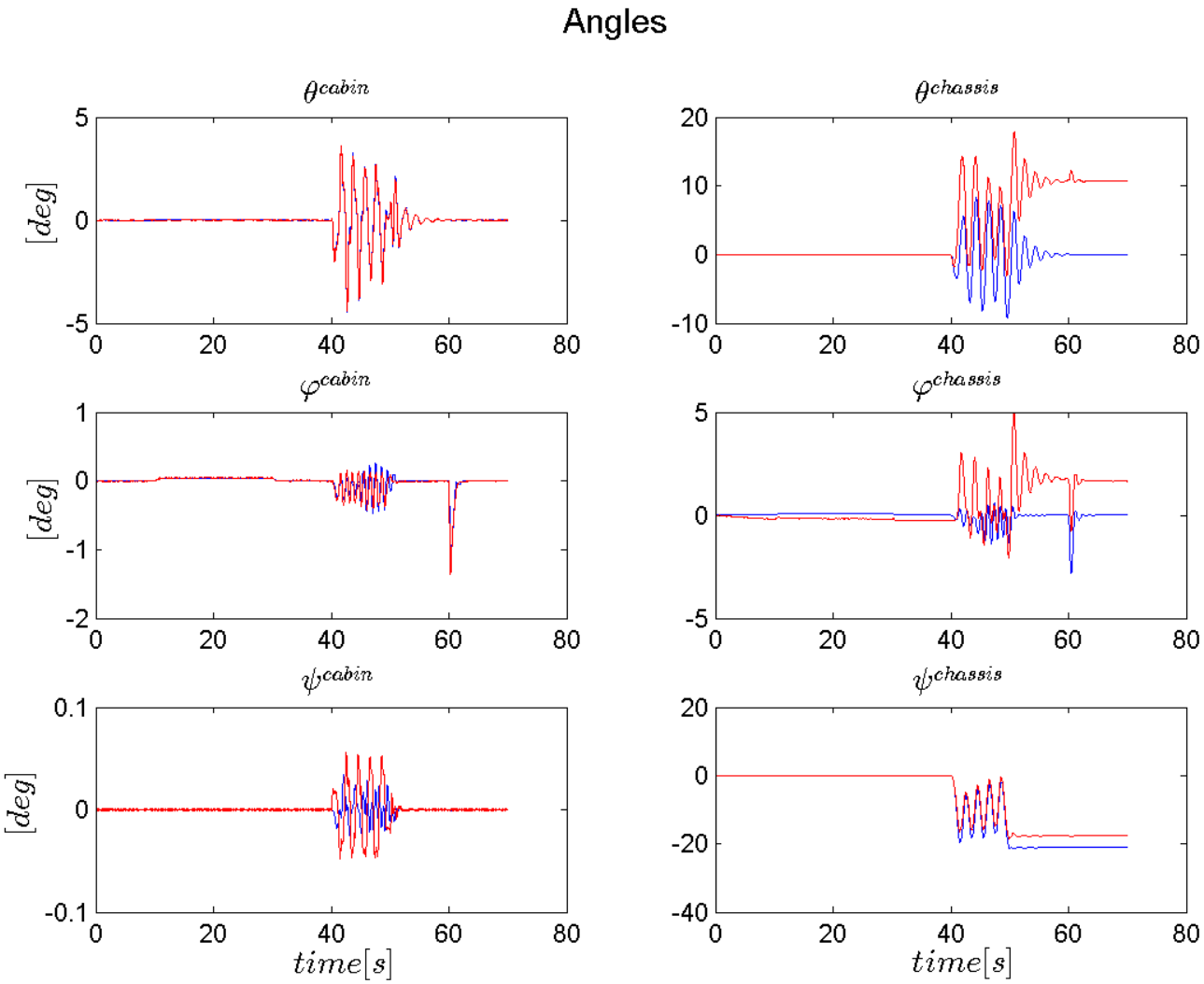


Figure 5.15: Estimated position states for the MB model using GNSS and IMU sensors.



- estimated states, - references

Figure 5.16: Estimated orientation states shown with *Euler Angles* for the MB model using GNSS and IMU sensors.

GNSS, IMU and truck velocity

Figure 5.17 and 5.18 displays that there are no larger differences if the velocity measurements are included with the GNSS and IMU sensors or not.

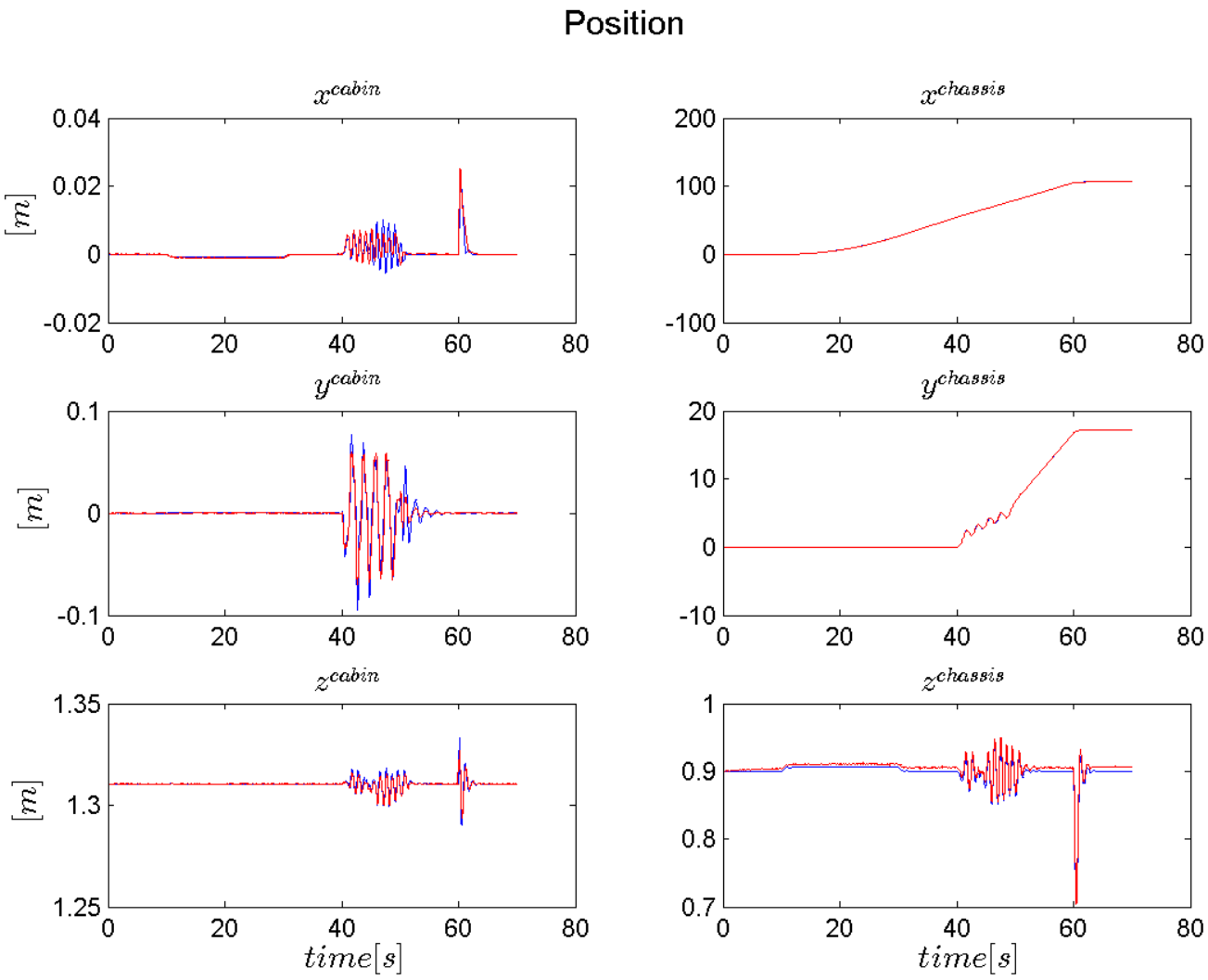
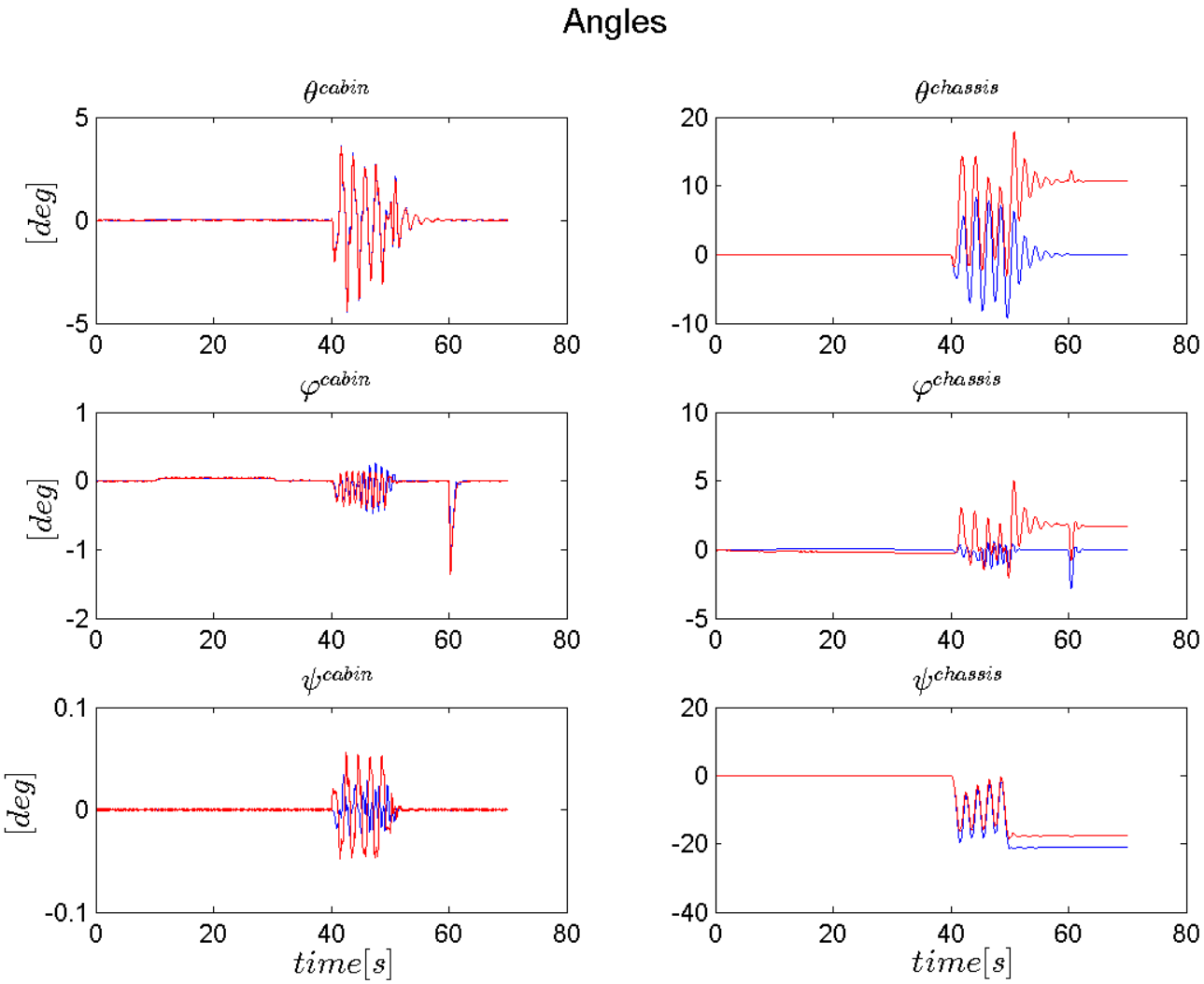


Figure 5.17: Estimated position states for the MB model using GNSS, IMU and velocity sensors.

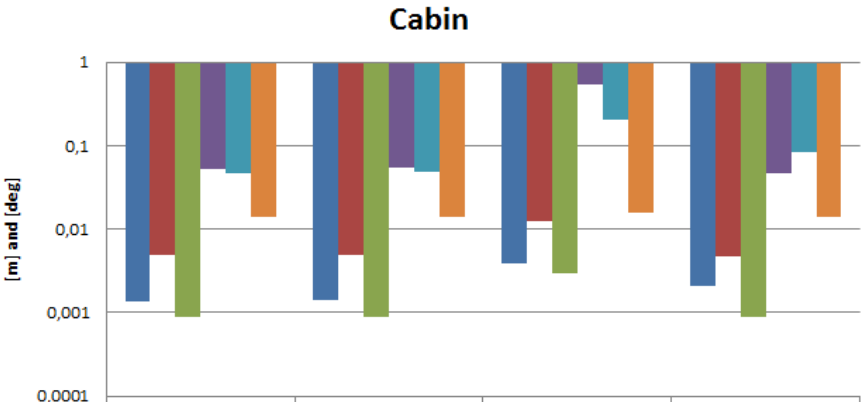


- estimated states, - references

Figure 5.18: Estimated orientation states shown with *Euler Angles* for the MB model using GNSS, IMU and velocity sensors.

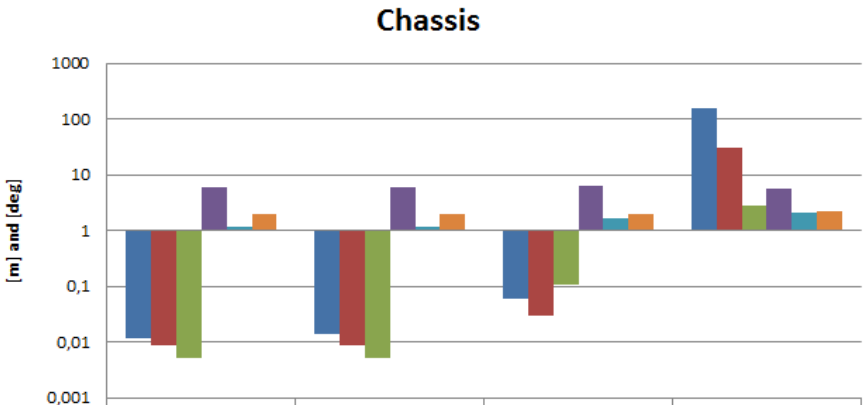
Filter evaluation

The evaluation of this filter can be seen in Figures 5.19 - 5.20 where the first one is for the cabin states and the second for the chassis states. These figures illustrate the root mean square (RMS) error between the reference and the estimated states.



	GNSS_IMU_VEL	GNSS_IMU	GNSS	IMU
Cabin x	0,0013797	0,0014036	0,0038206	0,0021178
Cabin y	0,0049371	0,0049558	0,012272	0,0047262
Cabin z	0,00087908	0,00088718	0,0029674	0,00088486
Cabin θ	0,053227	0,053518	0,53409	0,046469
Cabin φ	0,047455	0,048257	0,20797	0,082682
Cabin ψ	0,01408	0,014081	0,015481	0,014083

Figure 5.19: RMS value for different sensor combinations for the cabin with the MB model.



	GNSS_IMU_VEL	GNSS_IMU	GNSS	IMU
Chassis x	0,01192	0,013599	0,060706	154,8812
Chassis y	0,0085549	0,0085697	0,029653	29,8062
Chassis z	0,005143	0,0051481	0,10668	2,7165
Chassis θ	6,1376	6,1395	6,1661	5,7212
Chassis φ	1,1477	1,135	1,6852	2,1221
Chassis ψ	1,9803	1,9783	1,946	2,2511

Figure 5.20: RMS value for different sensor combinations for the chassis with the MB model.

It can be seen that there is small improvements by adding a velocity sensor for the chassis but nearly unchanged for the cabin. The GNSS sensor creates lower RMS values for the chassis than the IMU but the other way around for the cabin.

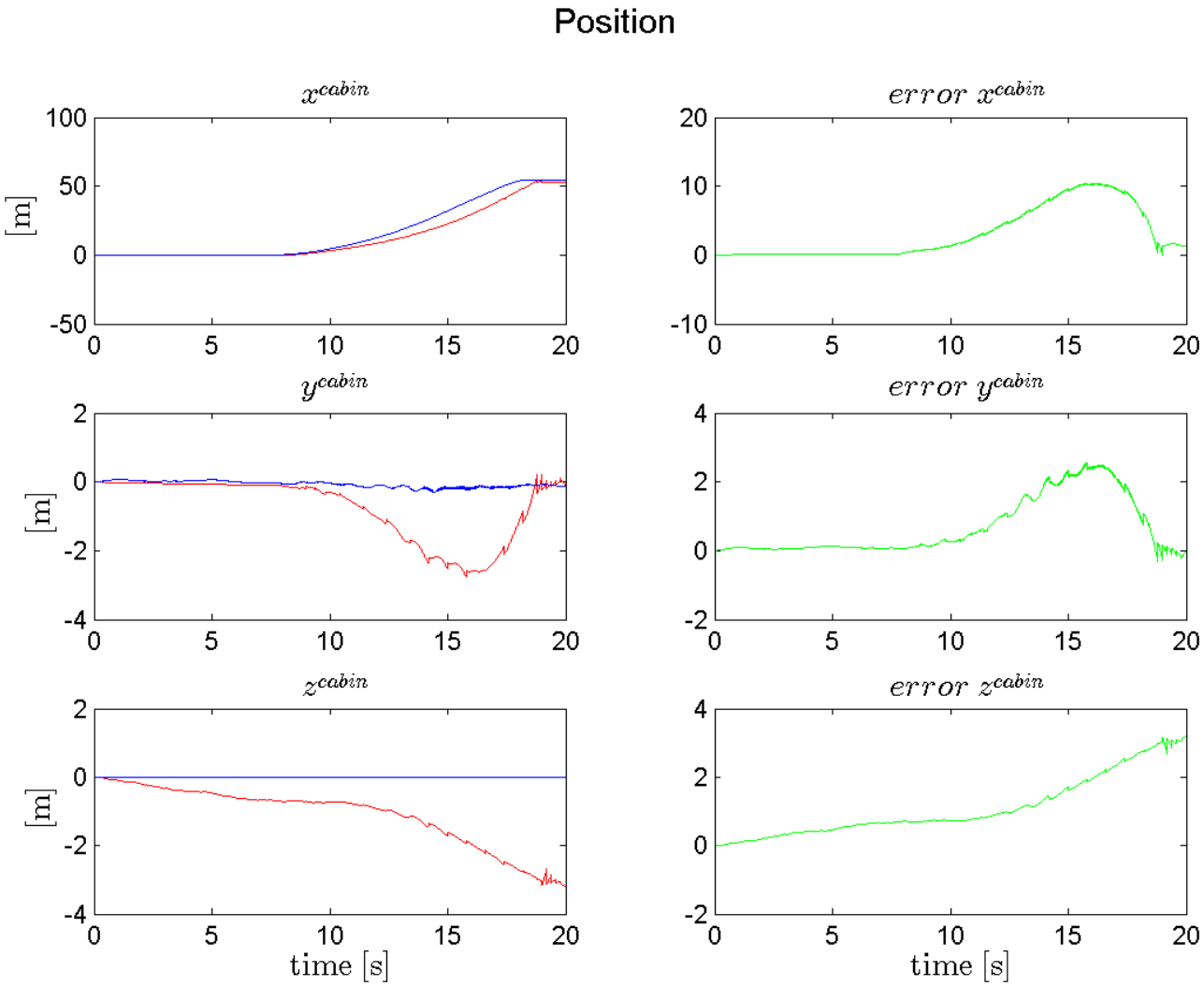
5.4 Filter with real data

This section contains estimation results with real data using the separated body model (SB) since the references is complete for the cabin's absolute position. However, the references are not sufficient for the orientation and to be able to validate the estimation with real data is it compared to a simulated version of the real scenario.

The scenario is when the truck is first stationary for 5 seconds, followed by accelerating in 5 seconds and ends with a hard brake. The simulated scenario is tested with different sources of errors to explain certain behaviors of the estimation with real data.

To make the illustrations in Figure 5.21 to 5.29 more intuitive, the states have been rotated to a local coordinate frame by using the rotation between ECEF to ENU, stated in Section 3.1.3. The orientation states are shown in Euler angles, that is, θ-roll, φ- pitch and ψ- yaw, described in Section 3.2.1.

Figure 5.21 and 5.22 shows estimation with real data. The estimated position follows the reference quite well, but the orientation is hard to evaluate. This is instead valuated in the following section together with common sources of errors to explain the behavior of the estimations.



- estimated states, - references, - error

Figure 5.21: Estimated position states for the SB model using GNSS and IMU sensors with real data.

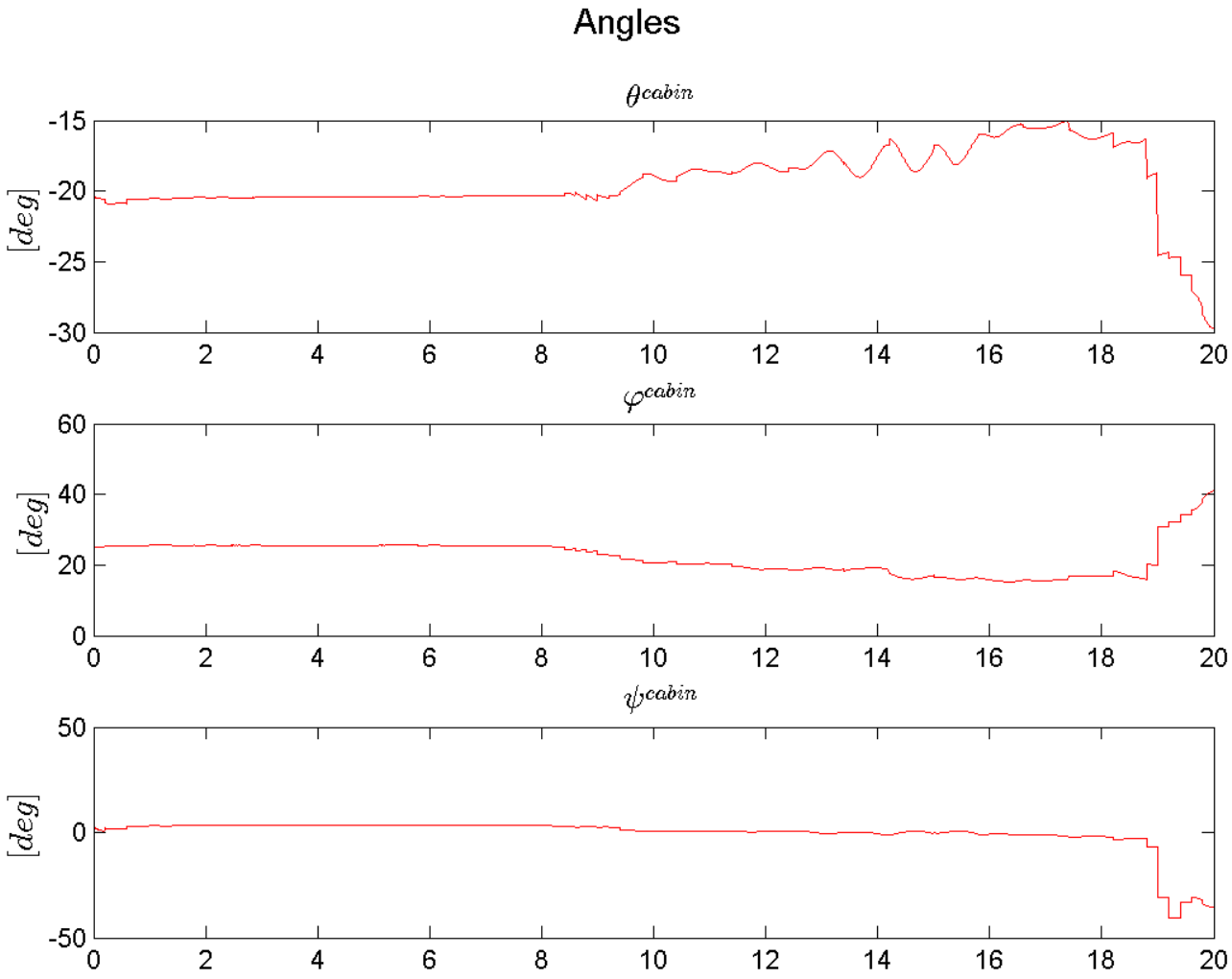
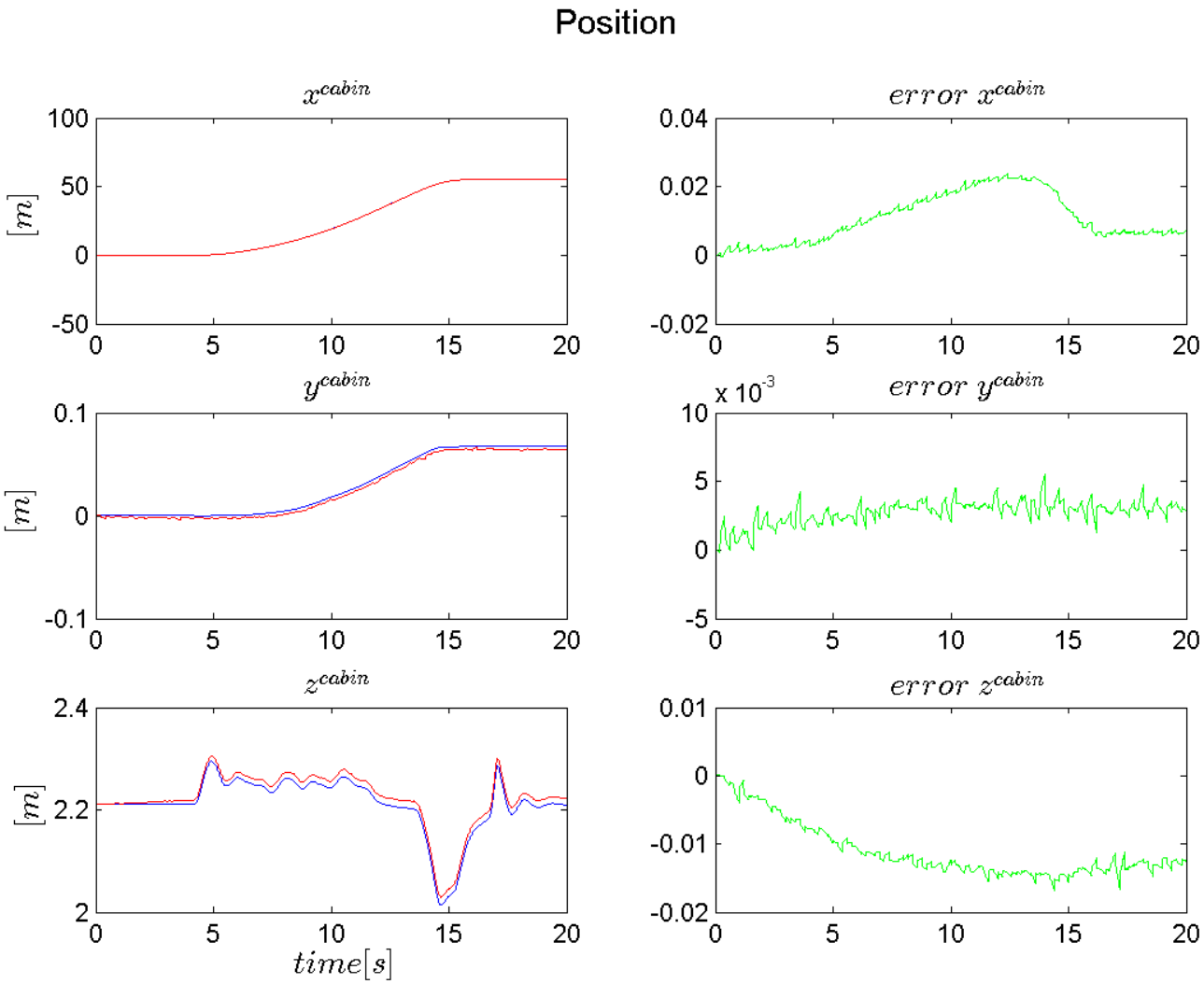


Figure 5.22: Estimated orientation states shown with *Euler Angles* for the SB model using GNSS and IMU sensors with real data.

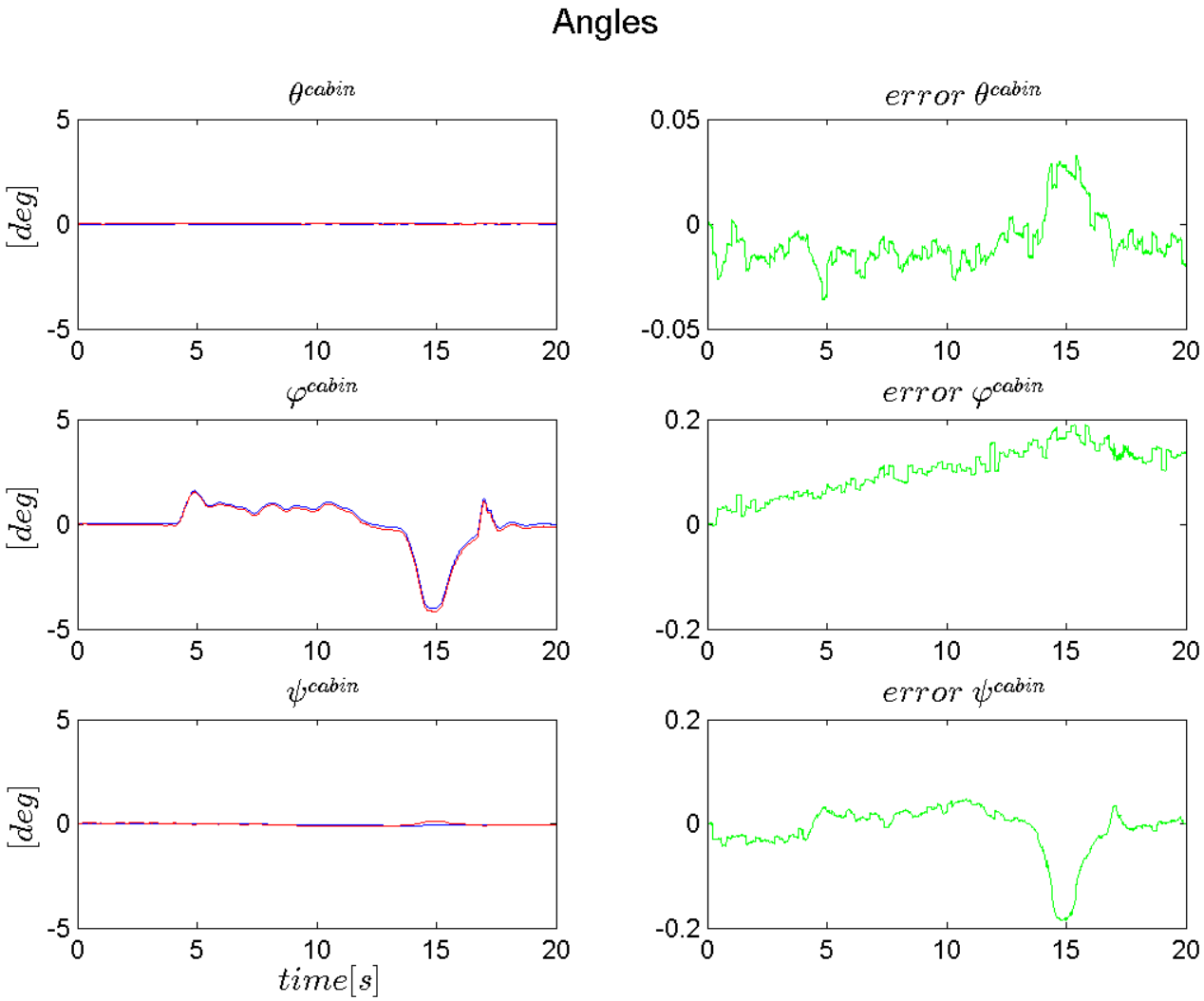
5.4.1 Sources of error

Common errors to explain the behavior of the estimates are inaccurate prior, varying GNSS frequency, incorrect distances between the sensors and incorrectly rotated accelerometer measurements. When the filter is not affected by any of the above mentioned errors it behaves as in Figure 5.23 and 5.24.



- estimated states, - references, - error

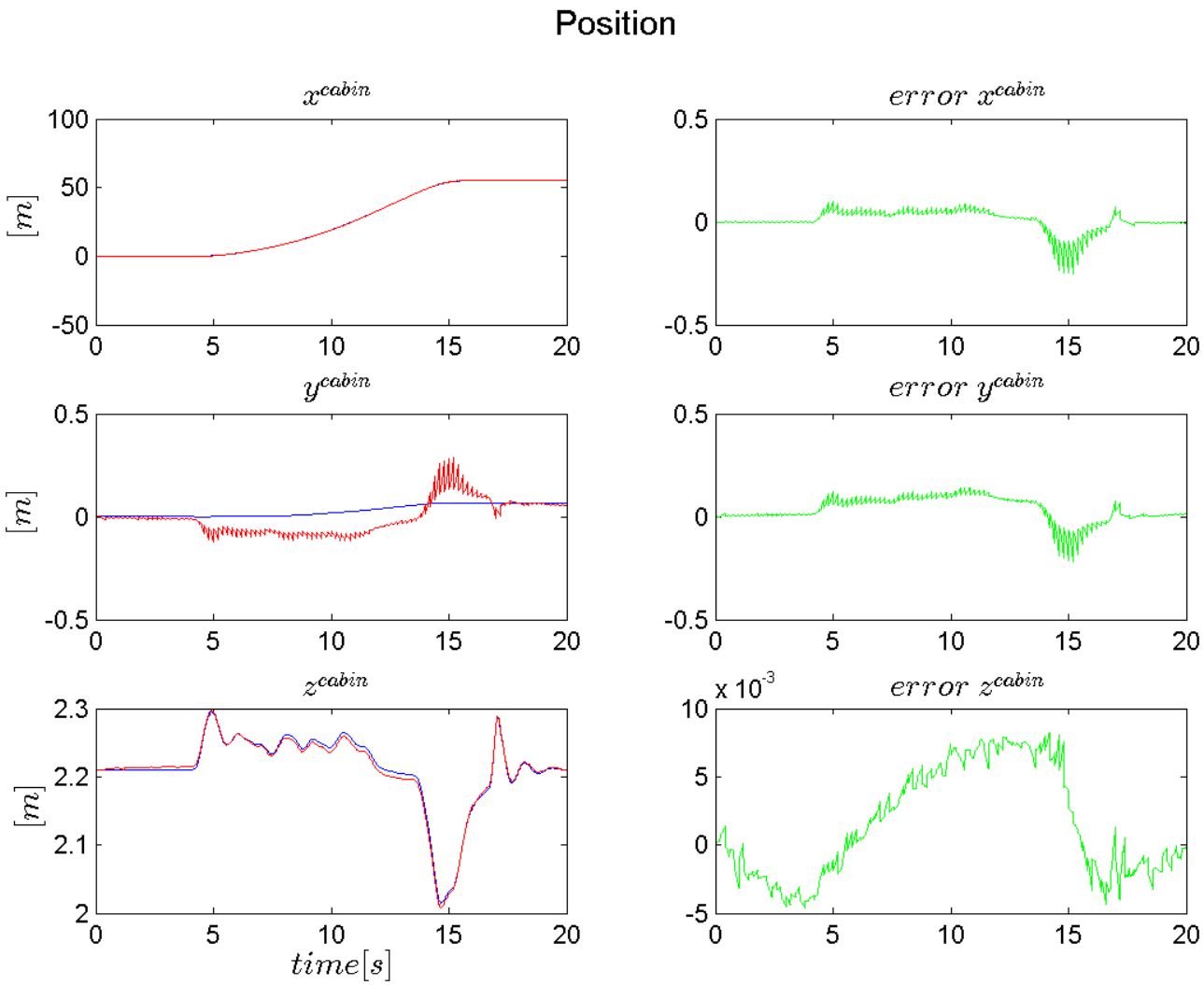
Figure 5.23: Estimated position states for the SB model using GNSS and IMU sensors with simulated data without any source of error.



- estimated states, - references, - error

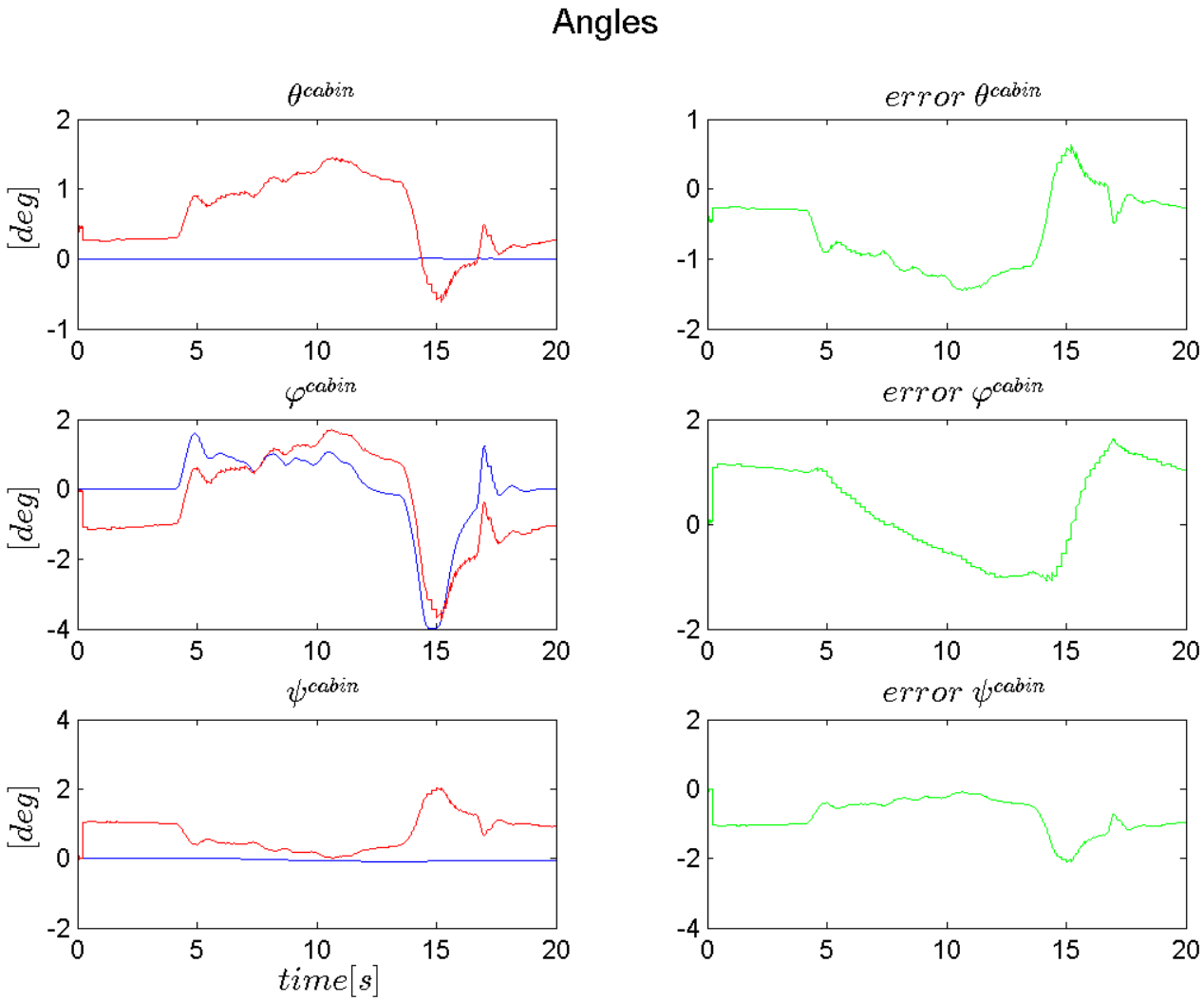
Figure 5.24: Estimated orientation states shown with *Euler Angles* for the SB model using GNSS and IMU sensors with simulated data without any source of error.

When an inaccurate prior of the orientation is introduced (see Figure 5.25 - 5.26), the filter is still able to estimate the trajectory of the position, but the orientation estimated states are incorrect. A high frequency error is also observed in the position states.



- estimated states, - references, - error

Figure 5.25: Estimated position states for the SB model using GNSS and IMU sensors with simulated data with a wrong prior.



- estimated states, - references, - error

Figure 5.26: Estimated orientation states shown with *Euler Angles* for the SB model using GNSS and IMU sensors with simulated data with a wrong prior.

If the GNSS receivers lose reception and the frequency varies (see Figure 5.1), the estimation gets worse. It can be seen that an offset is introduced in the position as well as steps in the orientation states after 15 s if receiver 1 and 3 gets sampled at $1.25Hz$ and receiver 2 at $1.67Hz$ instead of $5Hz$ (see Figure 5.27 and 5.28).

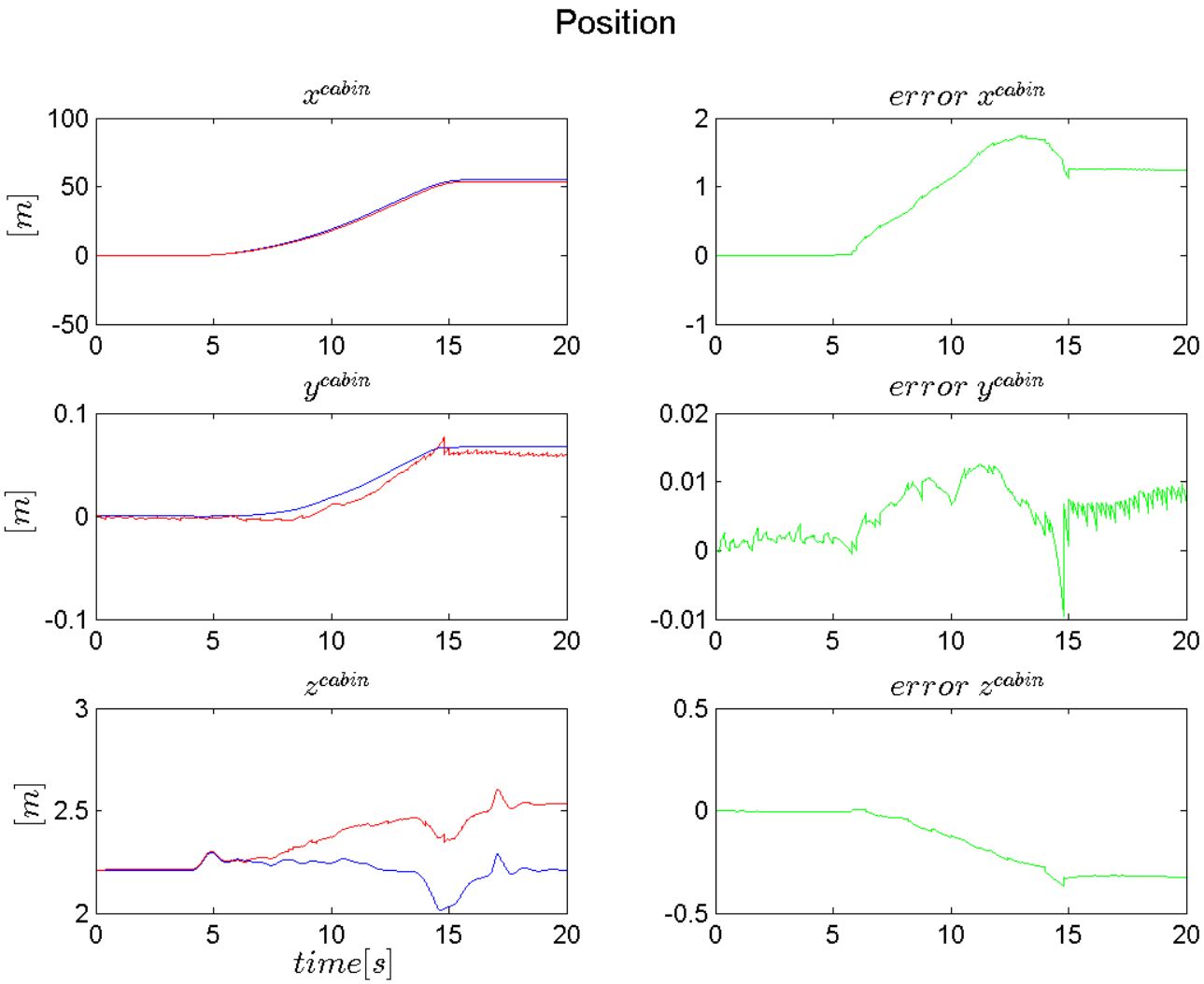
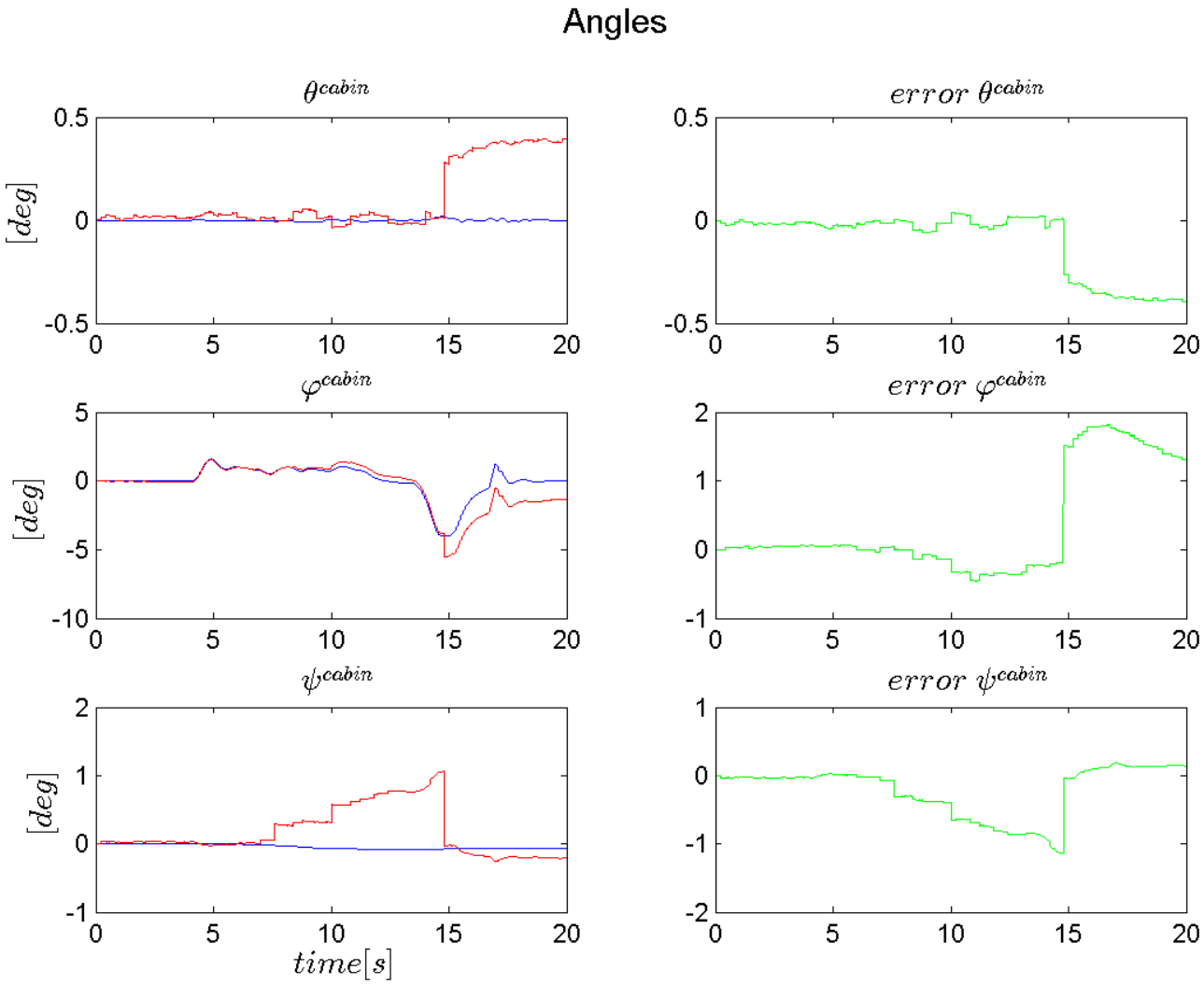


Figure 5.27: Estimated position states for the SB model using GNSS and IMU sensors with simulated data with varying GNSS frequency.



- estimated states, - references, - error

Figure 5.28: Estimated orientation states shown with *Euler Angles* for the SB model using GNSS and IMU sensors with simulated data with varying GNSS frequency.

Both the errors mentioned above are introduced in the filter (see Figure 5.29 and 5.30). The behavior of the estimated position is clearly similar to the estimated states with real data (see Figure 5.21). However, the estimated orientation is still hard to evaluate.

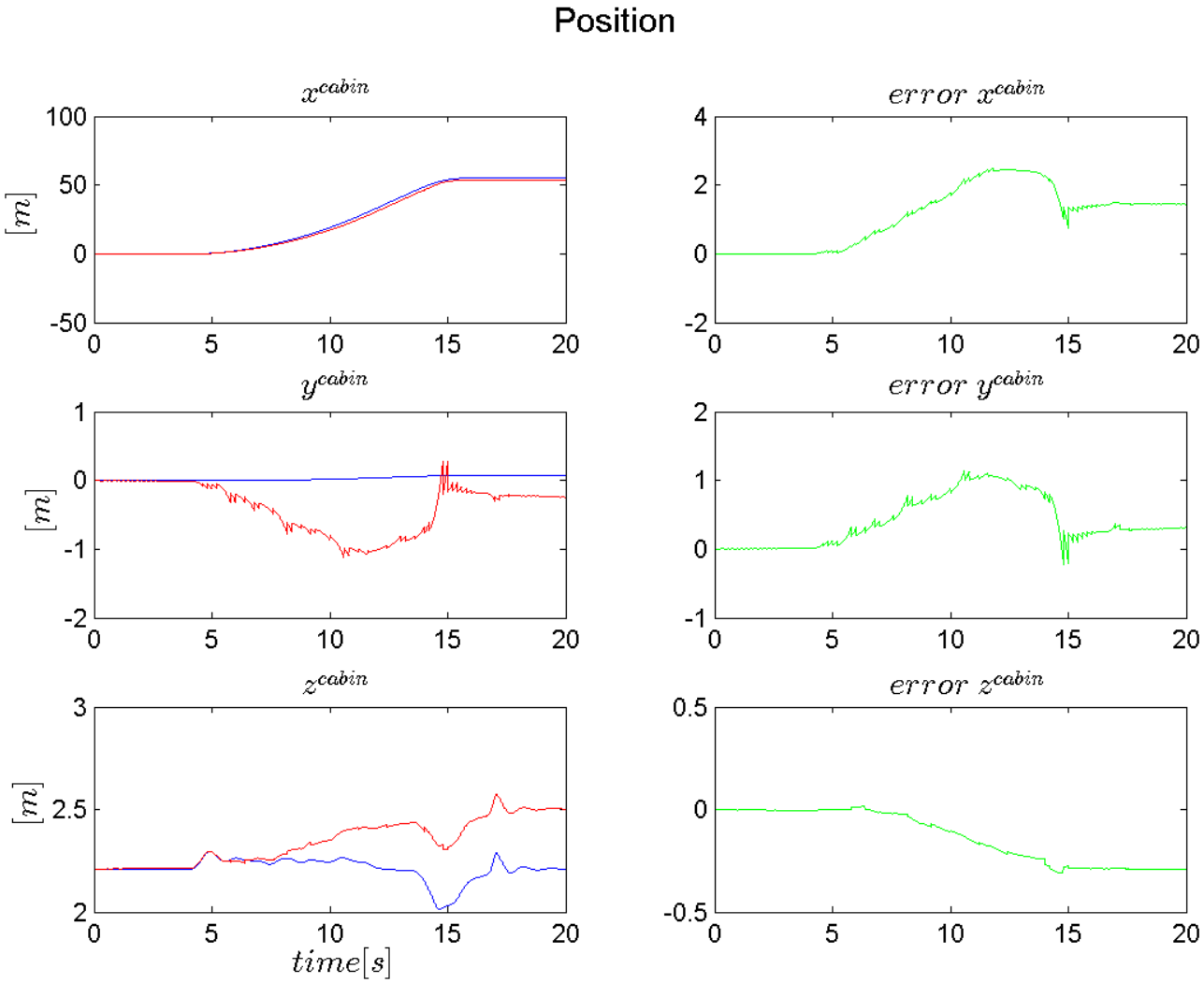
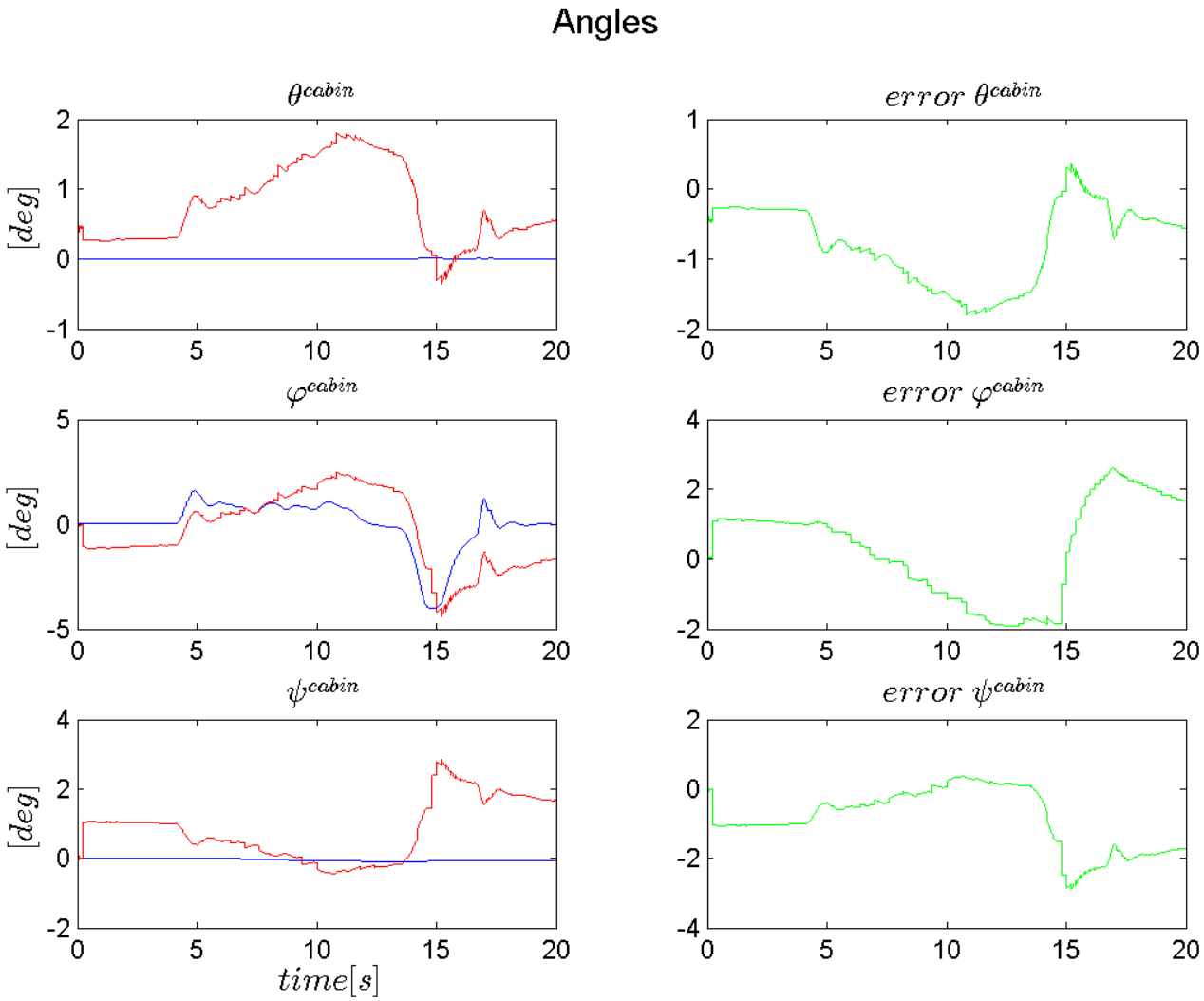


Figure 5.29: Estimated position states for the SB model using GNSS and IMU sensors with simulated data with wrong prior and varying GNSS frequency.



- estimated states, - references, - error

Figure 5.30: Estimated orientation states shown with *Euler Angles* for the SB model using GNSS and IMU sensors with wrong prior and varying GNSS frequency.

The filter is not sensitive to errors regarding inaccurate measured distances between sensors or to slightly incorrect rotations of the accelerometers when tested in simulation. There are no figures to illustrate this because all look very similar to Figure 5.23 and 5.24.

6
Discussion

The discussion chapter is divided into four topics. The first concerns a discussion about the sensor characteristics, the second is a general discussion about the filter results, the third a general discussion of the whole thesis and the fourth concerns future work.

Sensor characteristics

The GNSS receivers have a tendency of losing reception when the truck is moving, that is a decreased sampling frequency. It is known that GNSS signals are sensitive for disturbances, but not at open areas. Why the sensors are behaving like this is unknown but the problem seems to be in the actual sensors with loss of phase lock. A solution to this may be to use a slower sample time or to further investigate what the problem is before changing the sensors. Another approach is to investigate the possibility to manage this problem in simulation by imitating this behavior. The GNSS measurement variance is directly received from the U-Blox sensors and assumed to be correct but not verified since the thesis have focused on simulated data. A further investigation could be necessary when dealing with real data.

The results from the sensor characteristics indicates that the approximation of the Gaussian distributions of the noise for both the gyroscope and the accelerometer are inaccurate, especially for the IMU mounted on the chassis denoted as IMU 2. The distributions show more similarity with two mixed Gaussian distributions with individual mean values. Since the phenomenon appears in almost all sensors for IMU 2, namely the one closest to the engine, a hypothesis can be that these appearances are caused by vibrations from the motor.

The velocity sensor noise characteristic is investigated when the truck has a constant acceleration, which may not be the best approach. A constant velocity would be preferable, but could not be performed due to missing data. Another interesting aspect is that the velocity measurement is assumed related to the norm of the chassis velocity states. Another approach could be to further investigate the velocity measurement for example by including the steering angle together with the velocity of the wheels to provide a better relation for the integrated velocity sensor.

Estimation results

The results from the separated body model are rather expected. The GNSS sensors are not able to position the chassis because there is not a connection between the two bodies and the velocity sensor are not improving the cabin estimates due to the same motivation. The results indicate that the combination using only IMU sensors is better in estimating the chassis' position than the cabin's position, which may depend on larger motions for the cabin.

Another interesting aspect is that the shape of the estimated cabin angles is very similar in the estimations using either only GNSS or only IMU. This is an indication that the triangular setup may be better at estimating the angular velocity rather than the angles of the cabin. This is not that surprising since the GNSS sensors accurately measures the difference of the position in two time instances. Various test scenarios in simulation, like driving in circles, could have been used to further investigate this hypothesis.

The results from the merged body model shows that it is possible to use measurements from sensors that are mounted on one body to improve the estimates of another when they are connected, for example the cabin mounted GNSS sensors are now able to estimate the chassis states. The cabin's position and orientation relative to the chassis are accurate but the difference in using GNSS or IMUs is not large. This means that if only the relative position and orientation are required, then it is enough to use only the IMUs, which has the positive side effect of a faster computation time for the filter. This type of motion model has therefore been shown to be very helpful and important in this thesis. However, the chassis model could be improved with a more suitable motion model for a truck, for example a bicycle model that is able to capture the real movement of a vehicle. This may help solving the offset for the chassis angle estimates, but has not been further investigated since focus has been on the relative motion between the chassis and the cabin.

The coefficient values of the merged body model (see Table 5.5) are probably not optimal, since the selection was made by trial and error. It would be preferable to select these values with some kind of minimum error estimator, like a minimum mean square error method. However, this is hard to accomplish with the real system, since there is no reference of the relative position and orientation. It would therefore be interesting to evaluate the need of accurate coefficients.

The reference orientation data, roll and yaw, that are provided by the VBOX system is not showed in the evaluation of the real data. The reason for this is that they require a transformation that is hard to validate compared to dealing with the position reference. This validation has not been a priority since the reference pitch angle could not be provided.

By comparing Figures 5.21 to and 5.23 it is clear that the simulated estimate is better than the real one. The interesting aspect is however that when the simulated data are exposed to errors such as inaccurate prior and varying GNSS frequency

the errors in position are very similar to the position error of the real data (see Figure 5.21 and 5.29). This means that the prior in the real data filtering is probably incorrect. The angles may not have the same appearance but the used prior error is maybe not the exact same prior error that is used in the real data. Finding this accurate error prior should require the same effort as finding the real prior. A complete reference, fixed frequency GNSS sensors, accurate prior and some tuning may result in good estimates using real data.

General discussion

There are different approaches to solve the stated problem that is usually the case when the subject is rather new and unexplored. A Cubature Kalman Filter may not be the best suitable filter in this multidimensional case due to the fact that the number of sigma points is doubled to the number of states. This results in slow filtering but not to forget, the filter is able to cope with the nonlinearities. An Extended Kalman Filter may handle the nonlinearities and it would have been interesting to compare the performance to the Cubature Kalman Filter. It would also be interesting to further investigate different kind of filtering methods than the ones mentioned in this thesis, for example a particle filter. If these filters are going to be used for online control algorithms or similar there must be either a decrease in number of states or a faster algorithm.

Another interesting question is if the time differencing method for the GNSS measurement model is enough or if there are other errors that are not handled when taking two measurements in two different time epochs. It would have been interesting to evaluate the triple difference method in these filters, which would have decreased the number of states due to no drift estimation, but increased the number of calculations. In an early stage of this thesis it was some tests to compare these two measurement models using a very simple motion model. The time differencing method was actually better to estimate the angles than the triple method and the hypothesis is that the triple difference method is more suitable to determine the distance between two receivers. This means that it measures a distance between two receivers that is considered fixed and therefore independent of whether the truck is stationary or not. This type of measurement model would instead be preferable if base stations are used and with that measure the distances between a base station and every receiver. Another possible reason to use the triple differencing method is that the drift estimation in the time differencing method is very sensitive and requires an accurate prior with small motion noise to work properly.

Future work

This thesis has used an old estimated state to describe an earlier state in the Kalman filter, which goes against the Markov chain assumption. The solution is therefore pragmatic and the filter could be improved by not using this approach. A future work could be to find a way to accurately describe old states in the state vector.

An accurate prior seems, from the filter evaluation of real data, to be required for the filters to work properly but it is rather expected when the model is nonlinear, described in Section 3.6. It is suggested to develop some kind of algorithm to determine the prior. A solution can be for example to use a particle filter to sift out good priors.

It has been found that an accurate motion model provides a better result for the cabin estimation. An accurate model over the chassis motion could be an approach to get an even better estimation in the chassis angles and the whole system.

7
Conclusion

This thesis have concluded that it is possible in simulation to estimate the position and orientation of a truck cabin relative to the chassis and the chassis' position and orientation in a world frame using GNSS-sensors, gyroscopes, accelerometers and a velocity sensor. An accurate prior is found to be very important to make the estimation feasible, and the motion model has been shown to be of great significance. Springs and dampers are great in simulating the real connection between the chassis and the cabin, and they provide satisfying results when used in the motion model. The merged model works well in estimating the relative position and orientation between the chassis and the cabin, and it allows GNSS measurements to improve the states of the chassis.

Bibliography

[1] Groves, P.D., Books24x7 (e-book collection) & Ebrary (e-book collection) 2013, *Principles of GNSS, inertial, and multisensor integrated navigation systems*, Second;2nd; edn, Artech House, Boston.

[2] Van Sickle, J. 2008, *GPS for land surveyors*, 3rd edn, CRC Press, Boca Raton, FL.

[3] Hofmann-Wellenhof, B., Lichtenegger, H., Wasle, E. & SpringerLink (e-book collection) 2008, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*, Springer, Wien;New York;.

[4] Simo, S. 2013, *Bayesian filtering and smoothing*, Cambridge University Press, Cambridge.

[5] Titterton, D.H. & Weston, J.L. 2011, *Strapdown inertial navigation technology*, 2.th edn, IEE, Stevenage.

[6] Freedman, D., SpringerLink (Online service) & SpringerLink Archive (e-book collection) 1983, *Markov Chains*, Springer New York, New York, NY.

[7] Mitchell, H.B. & SpringerLink (e-book collection) 2007, *Multi-sensor data fusion: an introduction*, Springer, New York;Berlin;

[8] Senan, N.A.F. & O'Reilly, O.M. 2009, "On the use of quaternions and Euler–Rodrigues symmetric parameters with moments and moment potentials", International Journal of Engineering Science, vol. 47, no. 4, pp. 595-609.

[9] U-Blox. 2015, *GPS locate, communicate, accelerate Essentials of Satellite Navigation Compendium*, u-blox AG.

[10] U-Blox. 2015, *Essentials of Satellite Navigation*, u-blox AG, Switzerland.

[11] Diebel J. 2006, *Euler Angles, Unit Quaternions, and Rotation Vectors*, Stanford University, California.

[12] Westerlund, A. & Larsson, H.J. 2015, *Recursive Bayesian Estimation Applied to Autonomous Vehicles*, Chalmers University of Technology, Göteborg.

[13] Ibrahim, I.M. 2004, "A generally applicable 3D truck ride simulation with coupled rigid bodies and finite element models", *International Journal of Heavy Vehicle Systems*, vol. 11, no. 1, pp. 67.

[14] Stenborg, E. Hammarstrand, L. 2016, *Using a single band GNSS receiver to improve relative positioning in autonomous cars*, Chalmers University of Technology, Göteborg.

[15] Gao, Y., Balin, A.K., Dullens, R.P.A., Yeomans, J.M. & Aarts, Dirk G A L 2015, "Thermal Analog of Gimbal Lock in a Colloidal Ferromagnetic Janus Rod", *Physical review letters*, vol. 115, no. 24, pp. 248-301.

[16] Boström A. 2015, *Rigid body dynamics*, Chalmers University of Technology, Göteborg.

[17] Svensson L. 2015, "The Kalman filter", *SSY320 – Sensor fusion and nonlinear filtering* [Lecture slides], Chalmers University of Technology, Göteborg.
[18] Svensson L. 2015, "Orientation estimation using smartphone sensors", *SSY320 – Sensor fusion and nonlinear filtering* [Lecture slides], Chalmers University of Technology, Göteborg.
[19] InvenSense. 2013, *MPU-6000 and MPU-6050 Product Specification* [Datasheet]. rev 3.4.
[20] GARMNIN. 2011, *GPS 18x TECHNICAL SPECIFICATION* [Datasheet].
[21] U-Blox. 2016, *EVK-M8T Evaluation Kit* [Datasheet].
[22] Global positioning systems directorate, systems engineering & integration. 2014. Interface specification, IS-GPS-200. [ONLINE] Available at: http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf. [Accessed 29 February 2016].
[23] Elisson, V. & Gässler, G. 2014, *Low cost relative GNSS positioning with IMU integration*, Chalmers University of Technology, Göteborg.
[24] Navipedia. 2011. *Transformations between ECEF and ENU coordinates.* [ONLINE] Available at: http://www.navipedia.net/index.php/Transformations_between_ECEF_and_ENU_coordinates. [Accessed 23 March 2016].
[25] VBOX Automotive. 2016. *VBOX 3i Dual Antenna / 100Hz Vehicle Dynamics Measurement.* [ONLINE] Available at: https://www.vboxautomotive.co.uk/index.php/en/products/data-loggers/vbox-3i-dual-antenna. [Accessed 5 April 2016].
[26] howtomechatronics. 2016. *MEMS Accelerometer Gyroscope Magnetometer & Arduino.* [ONLINE] Available at: http://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/. [Accessed 5 April 2016].
[27] Wikipedia. 2016. *Gyroscope.* [ONLINE] Available at: https://en.wikipedia.org/wiki/Gyroscope. [Accessed 9 May 2016].
[28] dSPACE. 2016. *MicroAutoBox II.* [ONLINE] Available at: https://www.dspace.com/en/ltd/home/products/hw/micautob.cfm. [Accessed 5 April 2016].
[29] TEXample. 2009. Example: The 3dplot package. [ONLINE] Available at: http://www.texample.net/tikz/examples/the-3dplot-package/. [Accessed 12 April 2016].

A

Results regarding filter with separated motion model

The results in this appendix are from the filter with the separated body model, where there are no connections between the two bodies. Important to notice is that the cabin and the chassis states are both in the global coordinate frame and the orientation are defined with quaternions. The filter is tested with a specific scenario where the truck is stationary for 10 seconds, drives forward with constant acceleration in 20 seconds, then performs several S-turns, and ends with a hard brake.

To make illustrations in Figure A.1 to A.6 more intuitive, the states have been rotated to a local coordinate frame by using the rotation between ECEF to ENU, stated in Section 3.1.3. The orientation states are shown in Euler angles, θ- roll, φ- pitch and ψ- yaw , further described in Section 3.2.1.

GNSS

Figure A.1 and A.2 display how the filter estimates the position and the angles with only measurements from the GNSS sensors. It is noticeable that these measurements enable the filter to estimate the trajectory of the cabin's position and that it captures the motion of the orientation but not the states for the chassis. It is clearly displayed that the estimated z, θ and φ capture the motion but there is an offset when the truck moves.

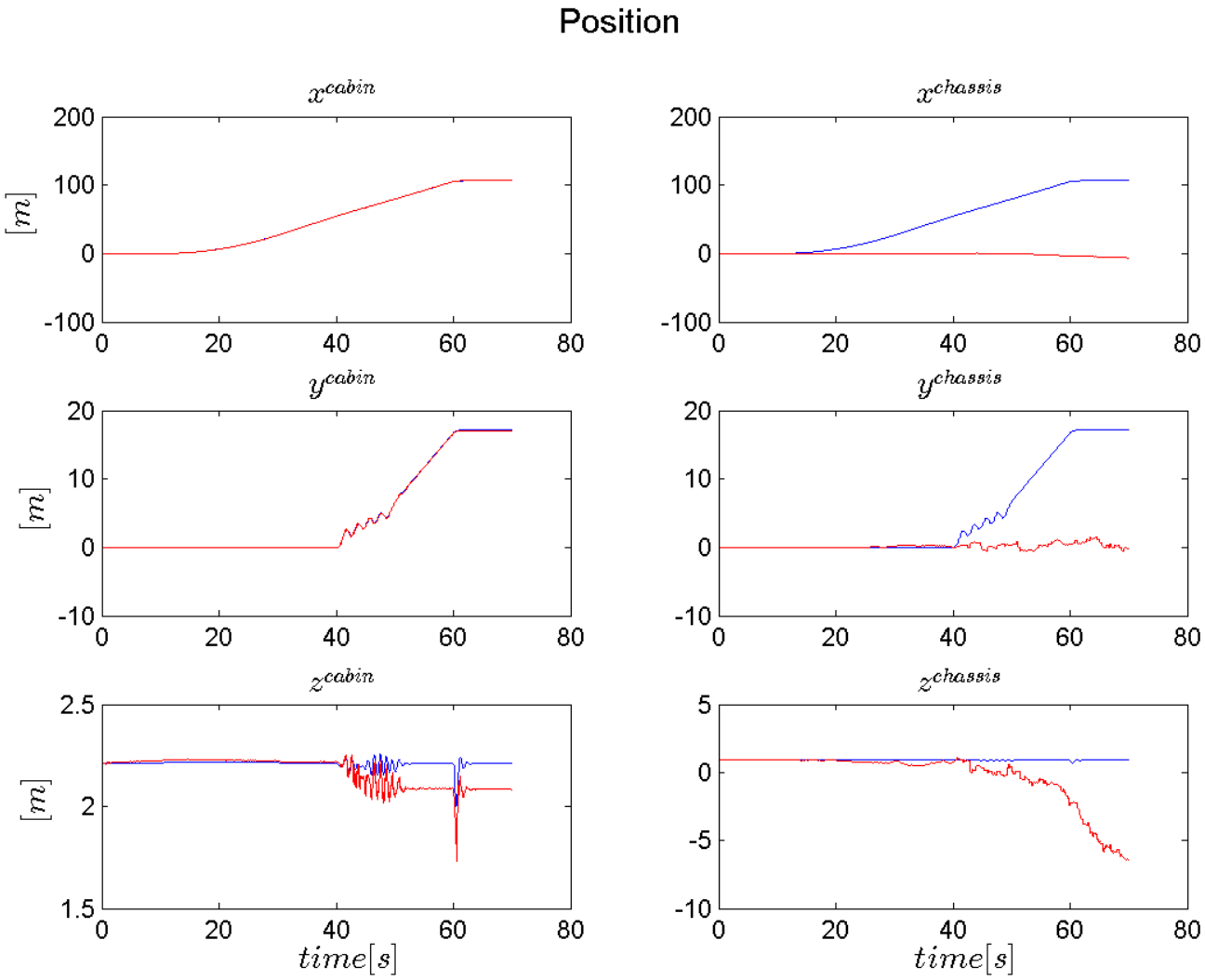


Figure A.1: Estimated position states for the SB model using GNSS sensors.

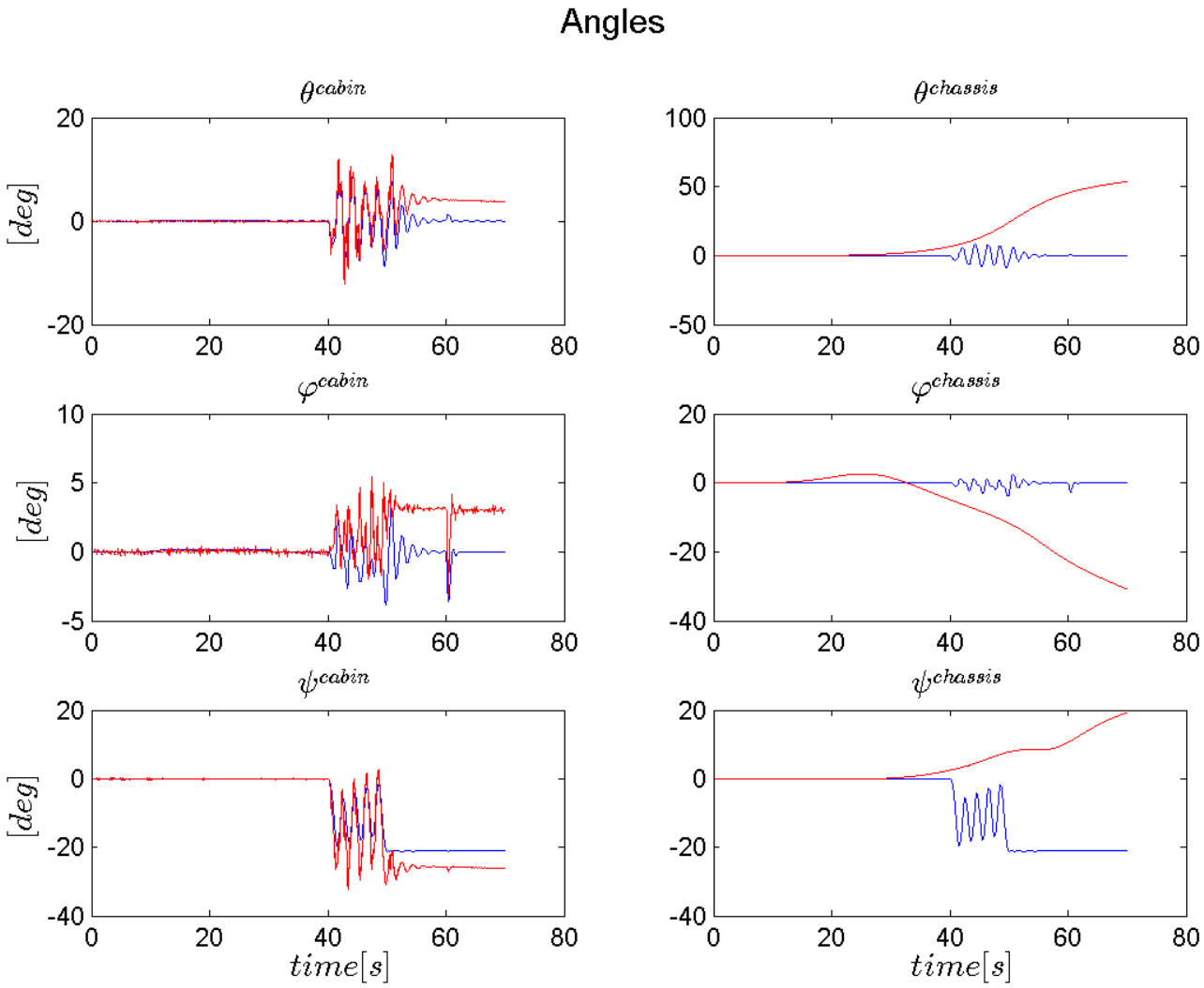


Figure A.2: Estimated orientations states shown with Euler Angles for the SB model using GNSS sensors.

IMU

Figure A.3 and A.4 display how the filter estimates the position and the orientation with only measurements from the IMU sensors. It is noticeable that these measurements are able to capture the motion of the orientation of both the chassis and the cabin, though with an offset when the truck starts to move. However, the filter is not able to estimate the position states for neither of the bodies except for the chassis' y position that follows a similar trajectory as the reference.

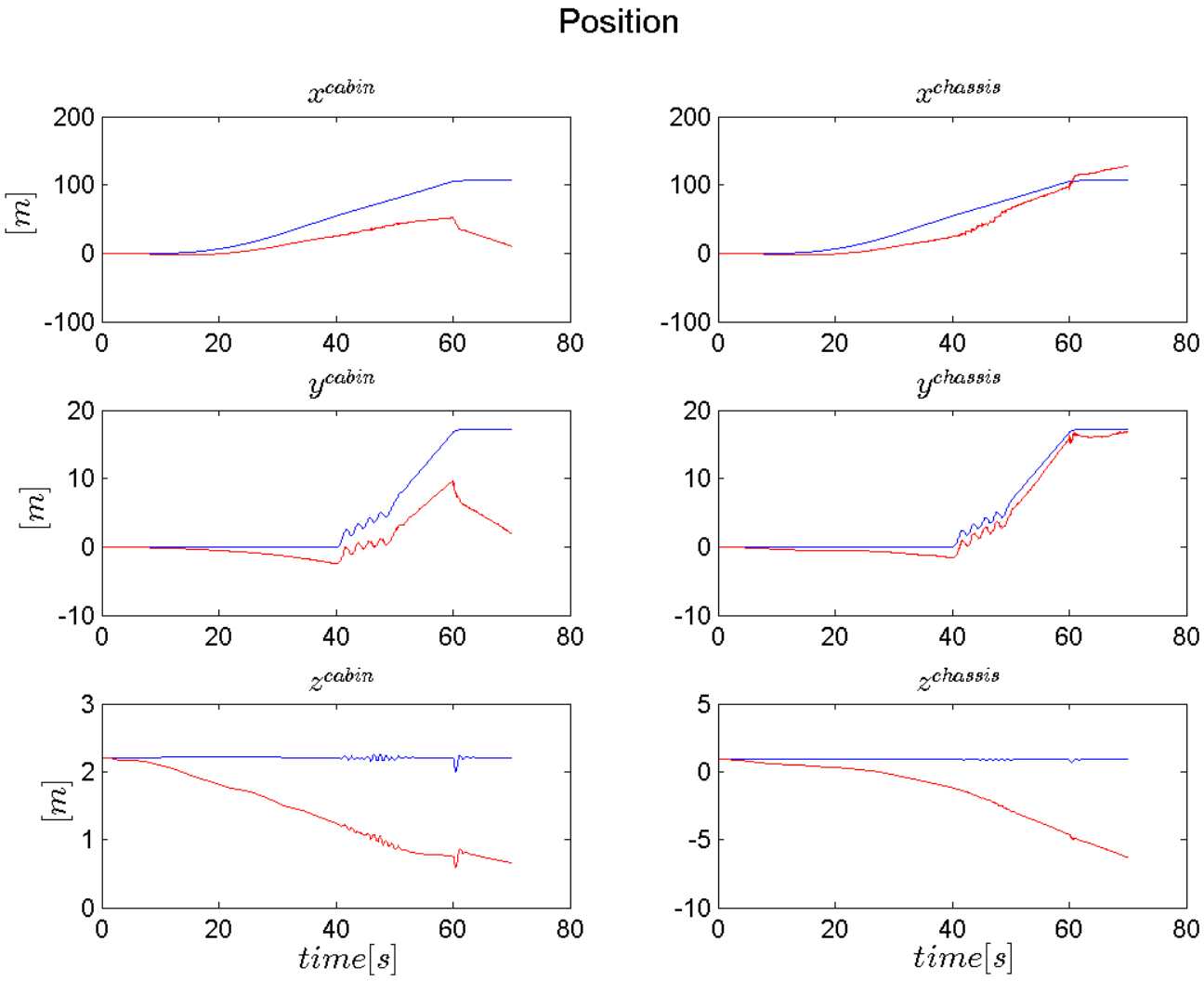


Figure A.3: Estimated position states for the SB model using IMU sensors.

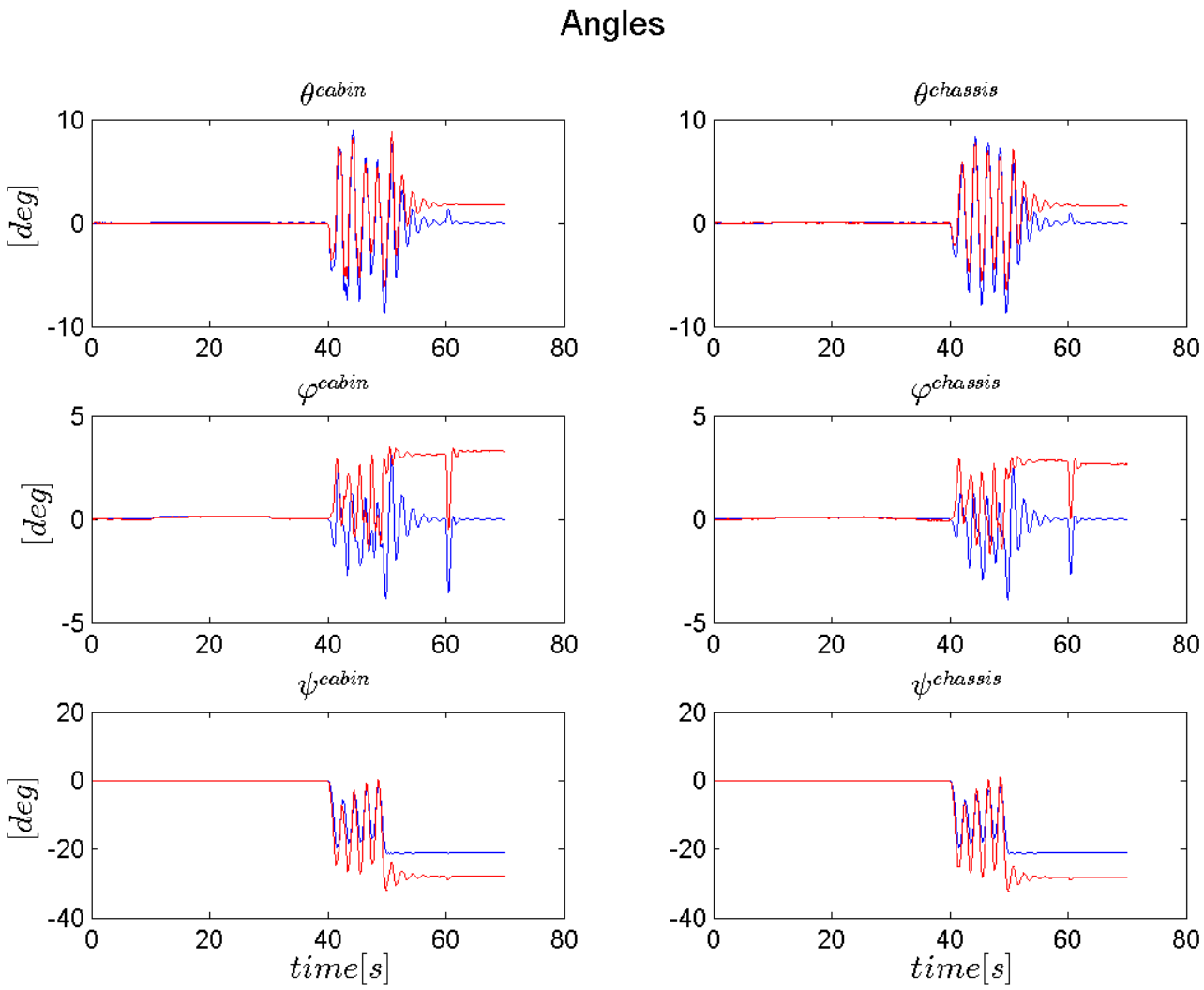


Figure A.4: Estimated orientations states shown with Euler Angles for the SB model using IMU sensors.

GNSS and IMU

Figure A.5 and A.6 display how the filter estimates the position and the orientation with measurements from both GNSS and the IMU sensors. It is noticeable that these measurements are able to estimate the position and the motions of the angles for the cabin, though with an offset when the truck starts to move. The states of the chassis remain unchanged from the measurements with only IMU's.

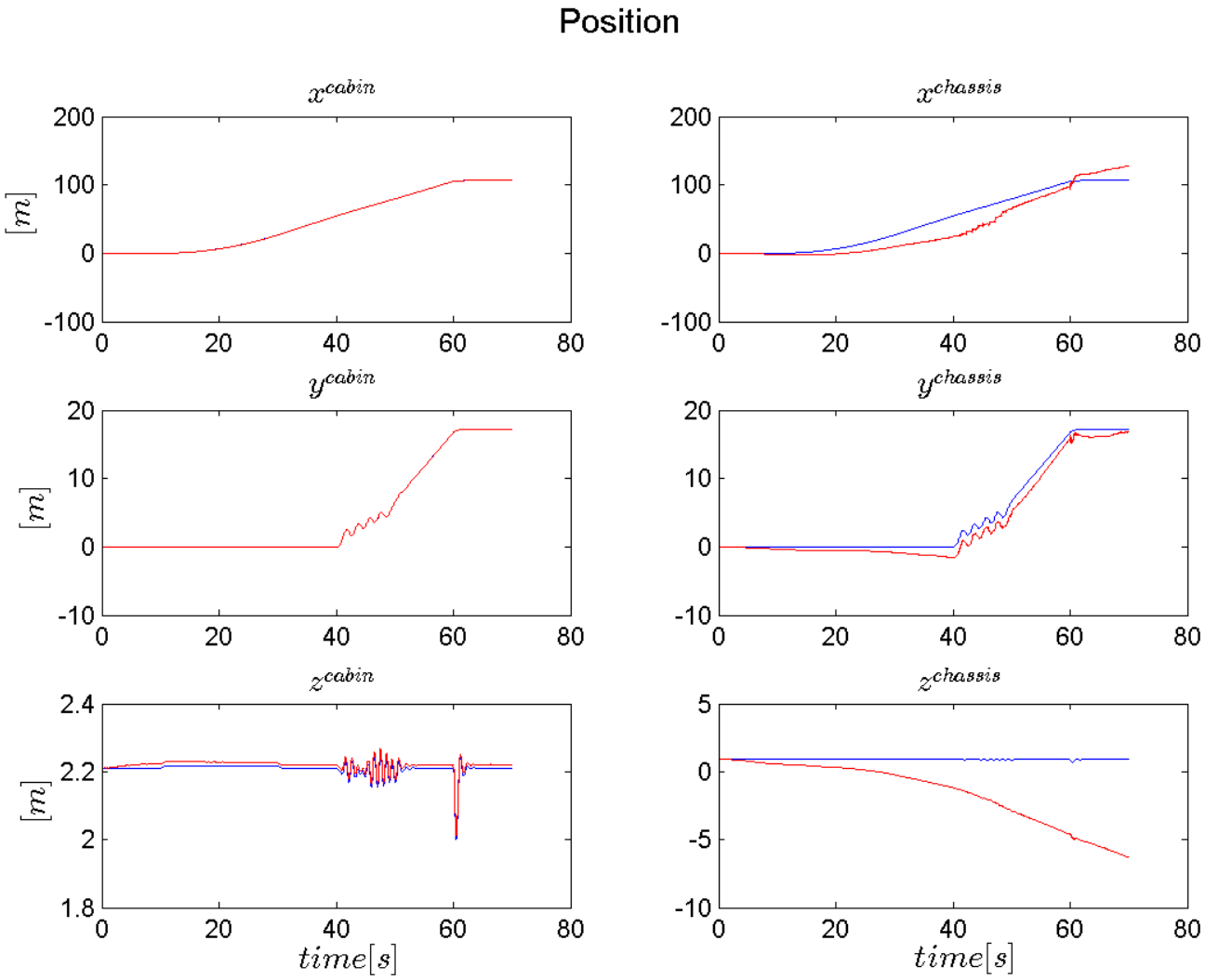


Figure A.5: Estimated position states for the SB model using GNSS and IMU sensors

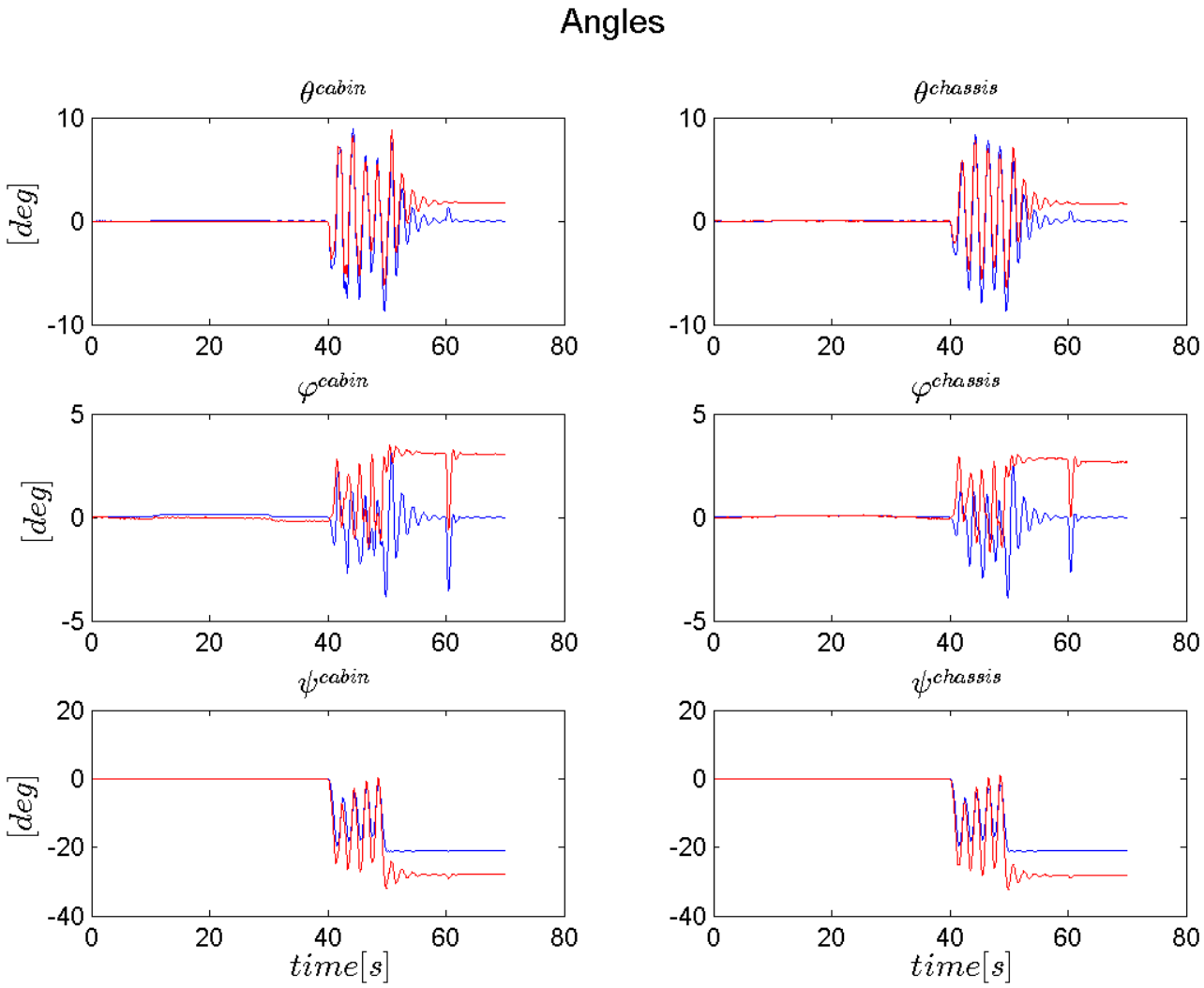


Figure A.6: Estimated orientations states shown with Euler Angles for the SB model using GNSS and IMU sensors.