

Master of Science Thesis in Electrical Engineering  
Department of Electrical Engineering, Linköping University, 2017

# Vehicle speed estimation for articulated heavy-duty vehicles

**Markus Rombach**



Master of Science Thesis in Electrical Engineering

**Vehicle speed estimation for articulated  
heavy-duty vehicles**

Markus Rombach

LiTH-ISY-EX--18/5120--SE

Supervisor:   **Clas Veibäck**  
                  ISY, Linköpings universitet  
                  **Selvam Kausihan**  
                  Volvo Construction Equipment

Examiner:     **Svante Gunnarsson**  
                  ISY, Linköpings universitet

*Division of Automatic Control  
Department of Electrical Engineering  
Linköping University  
SE-581 83 Linköping, Sweden*

Copyright © 2017 Markus Rombach

## **Sammanfattning**

En tydlig trend inom fordonsindustrin idag är semiautonoma funktioner och helautonoma lösningar. Denna funktionalitet kräver ofta god kännedom om fordonets tillstånd. En precis kännedom om hastigheten är viktig för många funktioner, bland annat kan en osäker hastighetsskattning leda till fel i positionsstämningen.

I detta examensarbete undersöks hur en IMU (Inertial Measurement Unit), innehållande ett gyroskop och en accelerometer, kan stödja hastighetsskattningen från odometri. IMU:n monterades för detta syfte på en hjullastare hos Volvo Construction Equipment. För att undersöka hastighetsskattningen utvecklades EKF:er (Extended Kalman Filter) med olika fordons- och sensormodeller. Till samtliga EKF:er togs även en Kalmansmoothen fram.

Undersökningen inleddes med en analys av sensorerna. EKF:en utvecklades och verifierades sedan med hjälp av en simuleringsmodel utvecklad av Volvo. Slutligen implementerades EKF:en på maskinen och testades på verklig data.



## **Abstract**

Common trends in the vehicle industry are semiautonomous functions and autonomous solutions. This new type of functionality sets high requirements on the knowledge about the state of the vehicle. A precise vehicle speed is important for many functions, and one example is the positioning system which often is reliant on an accurate speed estimation.

This thesis investigates how an IMU (Inertial Measurement Unit), consisting of a gyroscope and an accelerometer, can support the vehicle speed estimation from wheel speed sensors. The IMU was for this purpose mounted on a wheel loader. To investigate the speed estimation EKFs (Extended Kalman Filters) with different vehicle and sensor models are implemented. Furthermore all filters are extended to Kalman smoothers.

First an analysis of the sensors was performed. The EKFs were then developed and verified using a simulation model developed by Volvo Construction Equipment. The filters were also implemented on the wheel loader and tested on data collected from real world scenarios.



## Acknowledgments

I started and did most of the work for this thesis some time back, therefore I am certain I will forget to thank some people in this section. So thanks to all of you!

First and foremost I would like to thank everyone at Volvo Construction Equipment for giving me the opportunity to do this thesis. Especially I would like to thank my supervisor Kausihan Selvam and also Johan Sjöberg for all the support throughout the whole duration of the thesis work. Thank you for being so enthusiastic and pushing until the very end!

I would also like to thank Ted Samuelsson and Albin Nilsson at Volvo for all the coffee break discussions (including those about the actual thesis).

I would like to thank my supervisor Clas Veibäck and examiner Svante Gunnarsson at Linköping University for all the support along the way. Apart from the interesting discussions I am very impressed by the quick response time on e-mails which really helped since I was located in Eskilstuna. I also want to thank my former examiner André Carvalho Bittencourt, who left the university before I finished the thesis.

*Linköping, Juni 2018  
Markus Rombach*



---

# Contents

<b>Notation</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem formulation . . . . .	1
1.3 Related work . . . . .	1
1.4 The wheel loader . . . . .	2
1.5 Purpose . . . . .	2
1.6 Delimitations . . . . .	3
1.7 Thesis outline . . . . .	3
<b>2 Sensors</b>	<b>5</b>
2.1 Communication . . . . .	5
2.2 Sensor evaluation . . . . .	6
2.2.1 Auto correlation . . . . .	7
2.2.2 Histograms . . . . .	8
<b>3 Reference frames</b>	<b>13</b>
3.1 Common reference frames . . . . .	13
3.2 Used reference frames . . . . .	14
3.3 Rotation between different reference frames . . . . .	15
3.4 Rotational kinematics . . . . .	16
<b>4 Motion and sensor models</b>	<b>17</b>
4.1 Discretization . . . . .	17
4.2 Measurement models . . . . .	17
4.3 Wheel speed sensors . . . . .	18
4.3.1 Wheel angular velocity to wheel speed . . . . .	18
4.3.2 Relating velocities in a rotating frame . . . . .	19
4.4 Inertial Measurement Unit . . . . .	21
4.4.1 Accelerometer . . . . .	22
4.4.2 Gyroscope . . . . .	23
4.5 Standard motion model . . . . .	23

4.5.1 Motion model . . . . .	23
4.5.2 Odometry . . . . .	24
4.6 Extended motion model . . . . .	26
4.7 Zero velocity updates . . . . .	27
4.8 Observability . . . . .	27
4.8.1 Standard model . . . . .	28
4.8.2 Extended model . . . . .	31
<b>5 State estimation</b>	<b>33</b>
5.1 Pre-processing . . . . .	33
5.2 Kalman filtering . . . . .	34
5.3 Square root implementation . . . . .	36
5.4 Smoothing . . . . .	37
<b>6 Simulations</b>	<b>39</b>
6.1 Test scenarios . . . . .	39
6.1.1 Transportation . . . . .	40
6.1.2 Short cycle loading . . . . .	40
6.2 Simulation tests . . . . .	41
6.2.1 Transportation . . . . .	42
6.2.2 Short cycle loading . . . . .	42
6.3 Standard model without slip states . . . . .	42
6.3.1 With slip handling . . . . .	45
6.4 Standard model . . . . .	49
6.4.1 Adding an aiding sensor . . . . .	51
6.5 Extended model . . . . .	54
6.6 Smoother . . . . .	54
6.7 Conclusions . . . . .	57
<b>7 Experiments</b>	<b>59</b>
7.1 Test set up . . . . .	59
7.2 Standard model without slip states . . . . .	60
7.2.1 With slip handling . . . . .	60
7.3 Standard model . . . . .	61
7.4 Extended model . . . . .	63
7.5 Conclusions . . . . .	64
<b>8 Concluding remarks</b>	<b>67</b>
8.1 Future work . . . . .	68
<b>A Smoother</b>	<b>71</b>
<b>Bibliography</b>	<b>75</b>

---

# Notation

## ABBREVIATIONS

Abbreviation	Definition
ACF	Autocorrelation function
ABS	Anti-lock braking system
VCE	Volvo Construction Equipment
IMU	Inertial measurement unit
KF	Kalman filter
EKF	Extended Kalman filter
GPS	Global Positioning System
NED	North east down
CAN	Controller area network
ECU	Electronic control unit
RPM	Rotations per minute
RTS	Rauch-Tung-Striebel
SCL	Short cycle loading

---

## ROTATIONS

Notation	Definition
$\alpha$	Articulation angle
$\phi$	Roll angle
$\theta$	Pitch angle
$\psi$	Yaw angle
$\omega$	Wheel angular velocity
$\omega_i^j$	Angular rate about axis $i$ in reference frame $j$
$R^{i \rightarrow j}$	Rotation from frame $i$ to $j$

---

**REFERENCE FRAMES**

<b>Notation</b>	<b>Definition</b>
$f$	Front part
$r$	Rear part
$n$	Inertial frame

**ESTIMATION**

<b>Notation</b>	<b>Definition</b>
$x$	State vector
$\dot{x}$	Time derivative of state vector
$\hat{x}$	Estimate of $x$
$\hat{x}_{k k-1}$	State estimate at time $k$ given measurements up to $k-1$
$\delta$	Sensor errors
$e$	White gaussian noise





# 1

---

## Introduction

### 1.1 Background

Accurate vehicle speed estimates are important for advanced control algorithms. For example in anti-lock braking systems (ABS) the vehicle speed estimate needs to be accurate in order to compensate for wheel slip. Volvo Construction Equipment's (VCE's) heavy-duty vehicles most commonly use four-wheel drive making all wheels subject to wheel slip. In addition, since the VCE machines normally are heavy vehicles used on working sites without paved roads they will often be subject to a lot of wheel slip.

### 1.2 Problem formulation

The current solution to estimate the speed does not use sensor fusion. By not using sensor fusion, the wheel slips are not observable and the estimates are sensitive to noise. A new solution for estimating the vehicle speed and the slip using IMU and odometer needs to be investigated. The IMU might improve the speed estimate without increasing the cost too much.

### 1.3 Related work

A lot of work has been done on vehicle speed estimation and many different approaches have been evaluated. In Daiß and Kiencke [1995] a fuzzy logic estimator is presented where all measurements are weighted together to get a speed estimate. This method does not estimate the slip directly but uses pre- and post-processing to handle slip. In Gustafsson et al. [2001], a Kalman filter is used on wheel speed sensor data for the non-driven wheels together with an IMU. A

Kalman filter allows for estimation of other disturbances in the sensors such as drift in the IMU.

In Hol [2011] it is noted that the drifting of IMUs is a well known problem. Further it is concluded that there is no generic solution to 3D target tracking but rather that the solutions are situational and require different sets of aiding sensors. To handle drifting IMUs the fusion of an IMU with vision, Ultra Wide-Band and GPS are discussed. In Scheding et al. [1997] a laser sensor is used for the same purpose on an articulated vehicle in mines. The set up is tested in a real scenario where it shows good results on estimating slip angles. The downside of using a laser sensor is that they cost a lot and that dust can cover the receiver.

Modelling of articulated vehicles is discussed in Corke and Ridle [2001] where a model similar to a bicycle model is used and an extra term with the articulation angular rate is added. The extra term decreases the modelling error significantly.

Research on IMUs and wheel speed sensors is vast. A common reference on IMUs is Titterton and Weston [2004] which has a thorough discussion on inertial navigation systems. In Gustafsson [2010] the possibilities and disturbances in wheel speed sensors are discussed and it is concluded that wheel speed sensors are information rich sensors.

## 1.4 The wheel loader

The wheel loader used in this thesis is comparable to an L120G series wheel loader, see Figure 1.1. It has an operating weight of approximately 15 tons and a bucket volume of  $3\ m^3$ . The maximum speed is around 40 km/h depending on bucket load.

The wheel loader consists of a front and a rear body part connected by a rigid free joint. The steering is performed by changing the angle between the two body parts. The angle is called articulation angle and the steering articulated steering.

## 1.5 Purpose

The purpose of this thesis is to investigate methods in speed estimation using an IMU and wheel speed sensors. This will be done using an articulated wheel loader with four wheel drive. The goal of this thesis is to investigate methods to

- Estimate vehicle velocity
- Estimate wheel slip

However, the main focus will be estimating vehicle velocity.



*Figure 1.1: Volvo wheel loader 120g.*

## 1.6 Delimitations

The wheel radius is considered constant. This is not true, especially not for a wheel loader. The wheel radius is affected by wear, vehicle speed and vehicle mass. A wheel loader in operation will generally differ in weight. In this thesis the mass and weight distribution of the wheel loader is assumed constant. Additionally zero lateral slip is assumed.

## 1.7 Thesis outline

**Chapter 1** gives a brief introduction to the thesis.

**Chapter 2** gives an overview of the sensors.

**Chapter 3** discusses reference frames.

**Chapter 4** presents the motion and measurement models.

**Chapter 5** gives an introduction to state estimation theory and the implementations used in this thesis.

**Chapter 6** presents the results from simulations.

**Chapter 7** presents the results from real world data.

**Chapter 8** gives concluding remarks about the work and suggestions for further work.



# 2

---

## Sensors

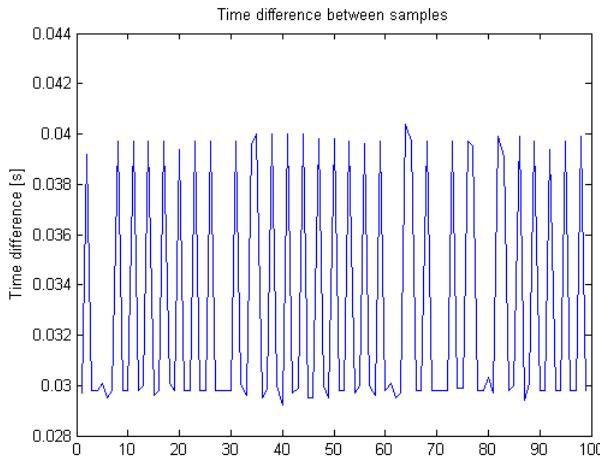
This chapter gives a brief overview of the communication between sensors and control units on the wheel loader and an evaluation of the sensors.

### 2.1 Communication

The communication between sensors and control units on the wheel loader is handled by the controller area network (CAN). CAN is a standard bus for communication between hardware in the vehicle industry, see for example Nitale et al. [2012] for a more detailed description. On the CAN bus all messages are given a certain priority. The message with the highest priority is processed first and the other messages wait. This can cause problems when reading data using an external program. The program used in this thesis to read CAN data was ATI Vision. ATI Vision is a program for data acquisition and calibration developed by Accurate Technologies, see [ATI]. The data reading rate was set to a fixed rate but the software has low priority which means that a varying data rate may occur anyway. An example of the time difference can be seen in Figure 2.1. Preferably Figure 2.1 should be a flat line meaning a constant rate. This time difference only affects data being read with ATI Vision because the hardware on the wheel loader has a higher priority on the CAN and will therefore not be influenced as much.

The signals are sent from the sensors to a data buffer before being collected by an electronic control unit (ECU). A vehicle typically includes many ECUs. The algorithms developed in this thesis are implemented and run on one of the ECUs. The buffer, where sensor data is collected before entering the ECU, is updated every time a new signal value arrives from a sensor. The ECU always uses the values on the data buffer and updates at a fixed rate. For unsynchronized sensors this can cause problems as the ECU can update with samples from different sam-

pling times. For example it could happen that one of the wheel speed sensors has managed to send its measurement value while another wheel speed sensor has not. The estimated velocity will then be a mix of two time instants.



**Figure 2.1:** Time difference between adjacent data samples. The x-axis shows the sample number  $i$  and  $f(i) = t_i - t_{i-1}$ , where  $t_i$  is the time stamp of sample  $i$ .

## 2.2 Sensor evaluation

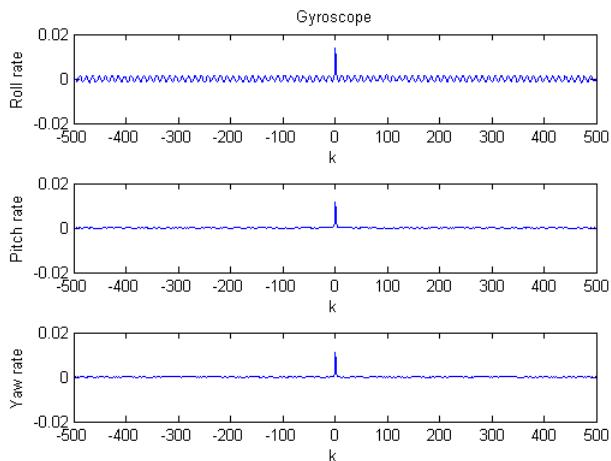
Analysing the noise in the sensors is an important part of signal processing. Data sheets of the sensors used in this thesis provide estimates of bias errors and noise. When attaching the sensors to machines the noise might increase. The noise is for example affected by the engine running, which creates vibrations.

Therefore an analysis was performed of the sensors. The analysis was done by gathering sensor data from the machine at stand still. At stand still the data gathered from the type of sensors used in this thesis should give zero output. Hence all non zero data gathered from stand still are disturbances in the sensors. The disturbance in a sensor includes a bias part, which can be estimated by taking the mean value of the data, and one stochastic noise part. The bias is first removed so that the purely stochastic part can be analysed. In the state estimation algorithms explained in Chapter 5, the sensor noise is assumed to be zero mean white Gaussian noise. The collected data was used to compute the autocorrelation function (ACF) in order to draw conclusions on the whiteness of the noise and plotting histograms to see if the noise is Gaussian.

The analysis is performed on the IMU. The odometers and articulation angle sensors show no noise when standing still. They might contain noise as well but due to quantization effects this noise is negligible.

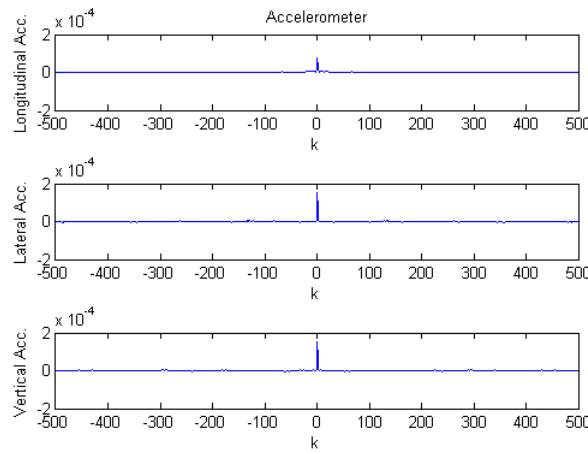
### 2.2.1 Auto correlation

The ACF correlates a signal with itself for different time shifts  $k$ . Ideal white noise consists of completely uncorrelated measurements which is equal to the ACF having a peak at  $k = 0$ , i.e. a dirac pulse. Figures 2.2, 2.3 and 2.4 show the ACFs for a maximum time shift  $k$  of 500. The ACF was calculated using Bartlett's method as described in Olofsson [2011]. The first 15000 samples from stand still were taken into account. At a sampling rate of 50Hz this equals 75 seconds of measurements.

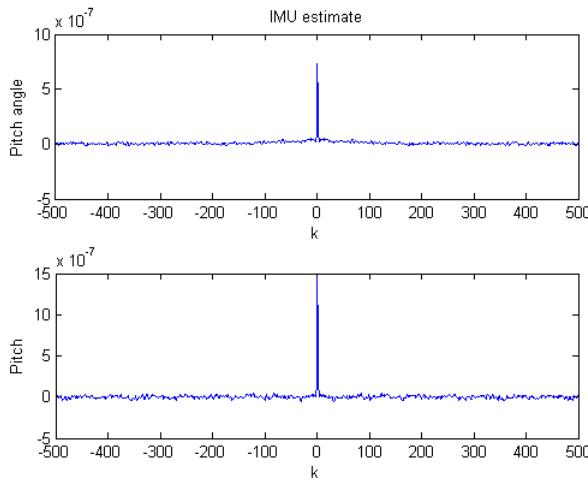


**Figure 2.2:** Autocorrelation function for the gyroscope in all directions.

Why the roll rate ACF resembles white noise less than pitch- and yaw rate is unknown. The machine might have been under the influence of a small roll angle. The roll rate sensor might also be slightly worse than pitch rate and yaw rate sensors. The ACFs are considered to resemble a dirac pulse and the assumption of white noise is therefore considered valid.



**Figure 2.3:** Autocorrelation function for the accelerometer in all directions.



**Figure 2.4:** Autocorrelation function of pitch and roll angles.

## 2.2.2 Histograms

The validation of Gaussian noise is examined through histograms. Perfect Gaussian noise would have a distribution according to

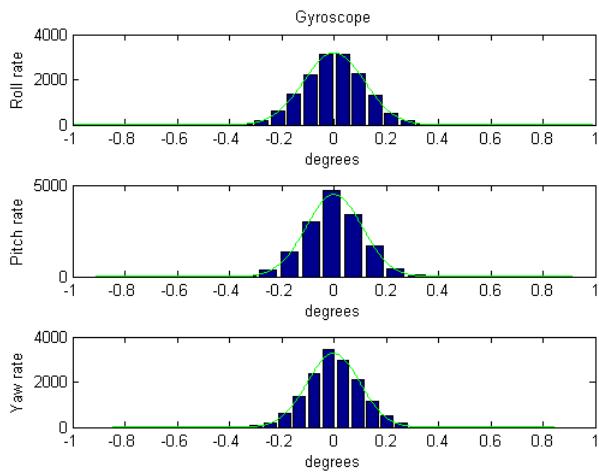
$$g(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

**Table 2.1:** Sensor noise variance. Accelerations are denoted  $a^i$  and angular velocities from the gyroscope are denoted  $\omega^i$ .

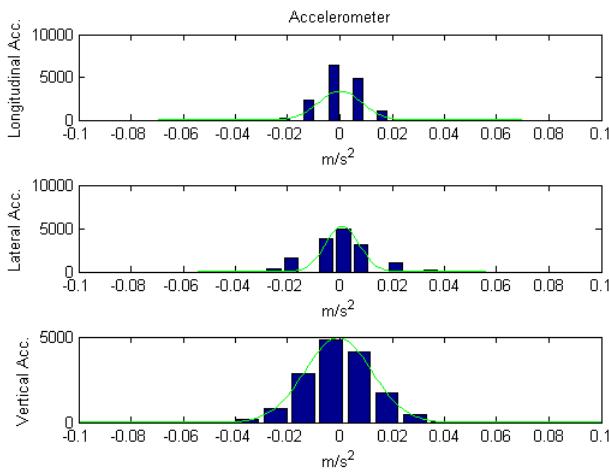
Sensor	Variance
$a^x$	$0.0083^2 [m/s^2]$
$a^y$	$0.0065^2 [m/s^2]$
$a^z$	$0.0128^2 [m/s^2]$
$\omega^x$	$0.002^2 [rad/s]$
$\omega^y$	$0.0019^2 [rad/s]$
$\omega^z$	$0.0017^2 [rad/s]$
$\phi$	$0.0019^2 [rad]$
$\theta$	$0.001243^2 [rad]$

where  $\sigma$  is the standard deviation and  $\mu$  is a bias. The histograms with fitted Gaussian curves are shown in Figures 2.5, 2.6 and 2.7. They are considered to resemble Gaussian curves closely enough to motivate the Gaussian noise assumption. As can be seen in the histogram for the longitudinal accelerometer the data is mapped on to separated bars. This could indicate that the data is quantized i.e. that the resolution of the accelerometer signal is lower than that of the gyroscope. This might affect the estimation results.

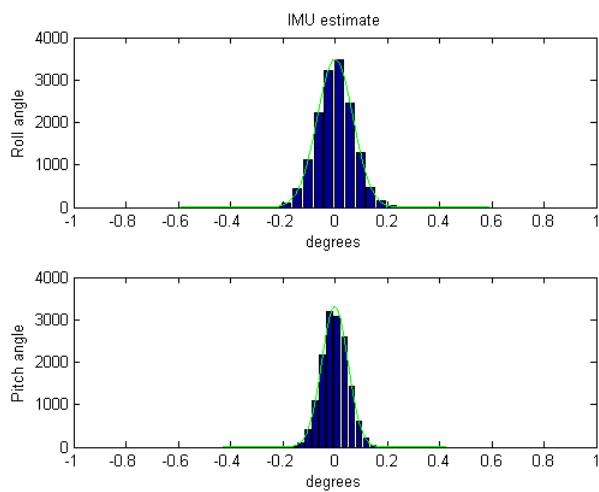
From the fitted Gaussian curves an estimation of the variance can be obtained. This variance is the noise variance in the sensors. The variances are shown in Table 2.1.



**Figure 2.5:** Histogram with fitted Gaussian curves in green for all directions in the gyroscope.



**Figure 2.6:** Histogram with fitted Gaussian curves in green for all directions in the accelerometer.



**Figure 2.7:** Histogram with fitted Gaussian curves in green for pitch and roll angle measurements.



# 3

---

## Reference frames

In order to fuse the information from the different sensors it is convenient to define reference frames. The reference frames will relate different measurements with each other. This chapter gives an introduction to reference frames and rotations between them.

### 3.1 Common reference frames

There are numerous choices of reference frames. Some of the more common ones are described in Titterton and Weston [2004]. Here they are briefly presented.

The inertial frame: A frame with its origin fixed at the center of earth and axis fixed at distant stars. With this frame the rotation of earth is taken in to account.

The earth frame: A frame with its origin fix at the center of earth but rotating with earth. Thus rotating with respect to the inertial frame.

The navigation frame, n-frame: The local NED-frame (north ,east ,down) has its origin at the location of the navigation system. With its axes pointing north, east and down towards the center of gravity.

The body frame, b-frame: A set of axes fixed to the body and aligned with its roll, pitch and yaw.

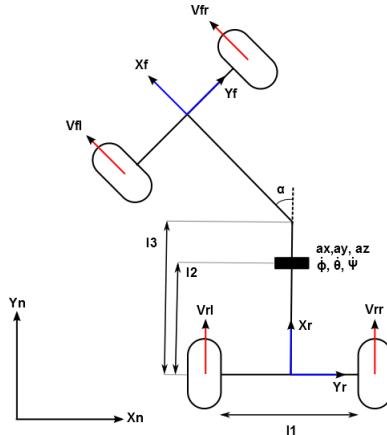
For this thesis the inertial and earth frames above are omitted. Since there is no GPS there is no knowledge of the vehicle position on earth. Thus the n-frame is approximated as an inertial frame. This means that fictitious forces from earth are omitted as well. The fictitious forces include the Coriolis effect due to the

rotation of earth and the centrifugal force.

## 3.2 Used reference frames

Articulated vehicles are vehicles consisting of two or more parts linked together. Most common is to use two parts, for example wheel loaders and haulers, linked together with a rigid free joint.

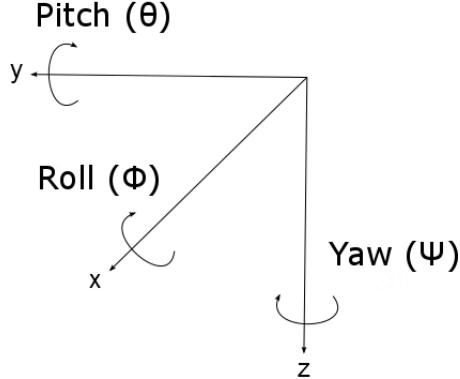
Since the wheel loader consists of two body parts, two body frames will be used. One for the front part, denoted  $f$ , and one for the rear part, denoted  $r$  with their respective origins as shown in Figure 3.1. The two body reference frames are related to each other via the articulation angle  $\alpha$ . The third frame used is the local inertial frame, denoted  $n$ , see Figure 3.1. The use of a NED frame is common due to convention. Using any orthogonal right basis will yield the same results.



**Figure 3.1:** Wheel loader with needed coordinate systems and measured signals. The measured signals are articulation angle ( $\alpha$ ), IMU measurements ( $a_x, a_y, a_z, \dot{\phi}, \dot{\theta}, \dot{\psi}$ ) and separate wheel speeds ( $v_i$ ). The width of the rear axis is  $l_1$ ,  $l_2$  the distance from the rear axis to the IMU and  $l_3$  the distance from the rear axis to the articulation joint.

### 3.3 Rotation between different reference frames

The most intuitive approach for representing rotation between reference frames is using Euler angles, roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ), see Figure 3.2. The rotation



**Figure 3.2:** Definition of Euler angles in a NED framework.

matrices are then defined as

$$\mathcal{R}_x^\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix}, \quad (3.1)$$

$$\mathcal{R}_y^\theta = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (3.2)$$

and

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

Multiplying all rotation matrices gives

$$\mathcal{R}_x^\phi \mathcal{R}_y^\theta \mathcal{R}_z^\psi = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{pmatrix} \quad (3.4)$$

where  $c_i$  and  $s_i$  stands for  $\cos(i)$  and  $\sin(i)$  respectively. In (3.4) the order in which the rotations are performed is important. A different order of rotations

yields different results. By using the rotation matrices the same vector  $\mathbf{v}$  can be expressed in different reference frames, related by the Euler angles  $\phi$ ,  $\theta$  and  $\psi$ , as

$$\mathbf{v}_i = \mathcal{R}_x^\phi \mathcal{R}_y^\theta \mathcal{R}_z^\psi \mathbf{v}_j = \mathcal{R}^{j \rightarrow i} \mathbf{v}_j \quad (3.5)$$

where  $\mathbf{v}_i$  and  $\mathbf{v}_j$  is the same vector expressed in the different reference frames  $i$  and  $j$  and  $\mathcal{R}^{j \rightarrow i}$  is defined as the rotation matrix (3.4) from  $j$  to  $i$ . Since the rotation matrices are orthogonal the inverse relation of (3.5) becomes

$$\mathcal{R}^{i \rightarrow j} \mathbf{v}_i = (\mathcal{R}_z^\psi)^T (\mathcal{R}_y^\theta)^T (\mathcal{R}_x^\phi)^T \mathbf{v}_i = \mathbf{v}_j. \quad (3.6)$$

### 3.4 Rotational kinematics

Since the measured angular velocities are in the rear body frame they need to be rotated to the inertial frame. According to Gustafsson [2012] the derivative of the Euler angles are given by

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & -\sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi / \cos \theta & \cos \phi / \cos \theta \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}. \quad (3.7)$$

As can be seen the Euler angles suffer from singular points at  $\theta = \pm\pi/2 + \pi n$ . One way to deal with the singularities is to use quaternion representation instead, which is common for airborne vehicles. However, since a pitch angle of  $\pm\pi/2$  would mean that the vehicle moves vertically, this is not a problem for land moving vehicles. Therefore, the Euler angle representation is used in this project.

# 4

---

## Motion and sensor models

In this chapter general sensor models with standard and extended motion models are presented.

### 4.1 Discretization

Systems are often modelled as time continuous. The data from sensors are sampled and therefore the time continuous motion models need to be discretized. Different sampling methods can be used. A differential equation  $\dot{x} = a(x)$  being sampled with a sampling time  $T$  can be exactly discretized by solving

$$x(t+T) = x(t) + \int_t^{t+T} a(x(\tau))d\tau. \quad (4.1)$$

Another sampling approach is Euler sampling then instead of solving (4.1), the update formulation becomes

$$x(t+T) = x(t) + T a(x). \quad (4.2)$$

In this thesis Euler sampling has been used. For more information on discretization see for example Gustafsson et al..

### 4.2 Measurement models

In an ideal world sensors give disturbance free and exact measurements. In reality that is normally not the case. Instead an important part in sensor fusion is finding out what kind of disturbances affect the sensors and model them. Let  $y$

be the value given from a sensor and  $x$  the actual physical quantity being measured. Instead of modelling the sensors simply as  $y = x$  a common sensor model for the sensors used in this thesis is

$$y = (1 + \delta_x)x + b_x + e \quad (4.3)$$

where  $\delta_x$  is a scaling error,  $b_x$  is a strongly correlated bias error and  $e$  is white Gaussian noise. Sensor models can be made more complex but this sets high requirements on the filters.

The scaling and bias errors are often hard to estimate beforehand because in many cases they depend on time-varying quantities such as temperature and wear. The random error,  $e$ , is often specified in the data sheets of sensors. Otherwise a variety of methods to evaluate the sensors exist, see Chapter 2.

## 4.3 Wheel speed sensors

There exist two different types of wheel speed sensors, rotary encoders that measure the angle and incremental rotary encoders that measure angular velocity. The wheel speed sensors used here are incremental rotary encoders.

The basic functionality of a rotary encoder is that a cogwheel is attached to a fixed axis. The cogs are then detected and a velocity is calculated. The sensors used in this thesis detect cogs at 6 kHz and give a wheel rotational speed with measurement frequency 100 Hz.

### 4.3.1 Wheel angular velocity to wheel speed

Let  $\omega$  be the angular velocity of a wheel. Then a fairly general wheel speed sensor model can be expressed as

$$y_\omega = (1 + \delta_\omega)\omega + b_\omega + e \quad (4.4)$$

where the errors,  $\delta_\omega$  and  $b_\omega$  can be interpreted as quantization effects and imperfect cog wheel size and  $e$  is white Gaussian noise. Since wheel slip is of high importance in this thesis an alternative measurement model will be used. If the errors in (4.4) are neglected, the angular rate can be multiplied with the tire radius  $R$  to give a speed measurement as

$$y_v = \omega R = v. \quad (4.5)$$

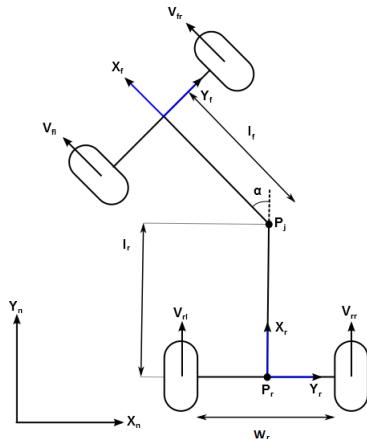
Errors affecting this measurement are slip and imperfect wheel radius according to

$$\begin{aligned} y_v &= (1 + s)\omega(\bar{R} + \delta_R) + e \\ &= \omega\bar{R} + \omega\delta_R + s\omega\bar{R} + s\omega\delta_R + e, \end{aligned} \quad (4.6)$$

where  $s$  denotes slip,  $\delta_R$  denotes a bias in the wheel radius and  $\bar{R}$  is the nominal wheel radius. The wheel radius will change with tyre wear and different vehicle loads. This difference can be up to 10% according to Wong [2008] for off road vehicles. For a wheel loader the difference can be larger due to changing center of gravity when filling the bucket. Wheel slip affects all driven wheels, and this machine uses four-wheel drive all wheels will be affected. The cog wheel and quantization errors are assumed to be small in comparison to the errors in the speed measurement. In this thesis the wheel radius will be assumed to be known and constant which gives a sensor model as

$$y = \omega \bar{R} + s\omega \bar{R} + e = (1 + s)v + e. \quad (4.7)$$

where  $v$  is the velocity for the different wheels. Since all sensors need to measure the same quantity in order to be fused the relation between the different wheels and a fixed point in the rear axis needs to be calculated.



**Figure 4.1:** A picture describing the relationship between front and rear axis.

### 4.3.2 Relating velocities in a rotating frame

The wheel speed sensors give the longitudinal speed of the vehicle in four different positions in body coordinates. This speed can be translated to a scaled speed at any position of the vehicle using the Coriolis equation as

$$\mathbf{v}_{p_2} = \mathbf{v}_{p_1} + \boldsymbol{\omega} \times \mathbf{r}_{p_1 p_2} \quad (4.8)$$

where  $\mathbf{r}_{p_1 p_2} = p_2 - p_1$  is the distance between the two points  $p_2$  and  $p_1$  in the body frame and  $\boldsymbol{\omega}$  is the angular rate of the body frame with respect to an inertial frame.

### Rear wheels

The point chosen to relate the measurements is the middle point, denoted  $P_r$  in Figure 4.1. Let  $v_x^r$  be the velocity at the center of the rear axis,  $w_r$  the length of the rear axis,  $v_{rl}$  and  $v_{rr}$  the measured speed for left and right speed respectively and  $\omega_z^r$  the rotation rate about the z-axis in the rear body frame. Using (4.8) and solving for the left wheel gives

$$\mathbf{v}_{rl}^r = \begin{pmatrix} v_x^r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \omega_z^r \end{pmatrix} \times \begin{pmatrix} 0 \\ -w_r/2 \\ 0 \end{pmatrix} = \begin{pmatrix} v_x^r - \omega_z^r \frac{w_r}{2} \\ 0 \\ 0 \end{pmatrix} \quad (4.9)$$

and solving for the rear right wheel gives

$$\mathbf{v}_{rr}^r = \begin{pmatrix} v_x^r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \omega_z^r \end{pmatrix} \times \begin{pmatrix} 0 \\ w_r/2 \\ 0 \end{pmatrix} = \begin{pmatrix} v_x^r + \omega_z^r \frac{w_r}{2} \\ 0 \\ 0 \end{pmatrix}. \quad (4.10)$$

This is the velocity in the rear body frame but can be expressed in any reference frame by using the rotations matrices from Chapter 2. With slip as a scale factor the measurement equations for the rear wheels become

$$\begin{aligned} y_{rl} &= (1 + s_{rl})(v_x^r - \omega_z^r \frac{w_r}{2}) + e_{rl} \\ y_{rr} &= (1 + s_{rr})(v_x^r + \omega_z^r \frac{w_r}{2}) + e_{rr} \end{aligned} \quad (4.11)$$

where,  $v_x^r$  is the speed at point  $P_r$ ,  $\omega_z^r$  is the angular rotation about the z-axis in the rear vehicle frame,  $w_r$  is the length of the rear axis,  $e_{rl}$  and  $e_{rr}$  are white Gaussian noise,  $s_{rl}$  and  $s_{rr}$  are the slips for rear and right wheels respectively.

### Front wheels

Let  $(l_r, 0, 0)$  be the joint position in the rear frame, denoted  $P_j$  in Figure 4.1, and  $v_x^r$  is still the velocity at point  $P_r$ . The velocity of the front wheels can then be calculated as

$$\mathbf{v}_j^r = \begin{pmatrix} v_x^r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \omega_z^r \end{pmatrix} \times \begin{pmatrix} l_r \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v_x^r \\ \omega_z^r l_r \\ 0 \end{pmatrix} \quad (4.12)$$

where  $\mathbf{v}_j^r$  is the velocity of the joint position in the rear frame. The velocity can be expressed in the front frame by rotating with  $-\alpha$  according to the definition of the articulation angle. The resulting joint velocity in the front frame is

$$\mathbf{v}_j^f = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} v_x^r \\ \omega_z^r l_r \end{pmatrix} = \begin{pmatrix} \cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r \\ -\sin(\alpha)v_x^r + \cos(\alpha)\omega_z^r l_r \end{pmatrix}. \quad (4.13)$$

This gives the relation between the mid-axis speed in the rear frame and the measured front wheel speeds in the front frame. Let  $l_f$  be the distance between the joint and the front wheel axis in the front frame as in Figure 4.1. Both rear and front axis are the same lengths i.e.  $w_f = w_r$ . The models for the front wheels can then be calculated as

$$\begin{aligned}\mathbf{v}_{fl}^f &= \mathbf{v}_j^f + \begin{pmatrix} 0 \\ 0 \\ \omega_z^f \end{pmatrix} \times \begin{pmatrix} l_f \\ \frac{w_f}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r - \omega_z^f \frac{w_f}{2} \\ -\sin(\alpha)v_x^r + \cos(\alpha)\omega_z^r l_r + \omega_z^f l_f \\ 0 \end{pmatrix} \\ \mathbf{v}_{fr}^f &= \mathbf{v}_j^f + \begin{pmatrix} 0 \\ 0 \\ \omega_z^f \end{pmatrix} \times \begin{pmatrix} l_f \\ -\frac{w_f}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r + \omega_z^f \frac{w_f}{2} \\ -\sin(\alpha)v_x^r + \cos(\alpha)\omega_z^r l_r + \omega_z^f l_f \\ 0 \end{pmatrix}\end{aligned}\quad (4.14)$$

and

$$\omega_z^f = \omega_z^r - \dot{\alpha} \quad (4.15)$$

where  $\omega_z^f$  is the angular velocity of the front frame relative to the inertial frame and  $\omega_z^r$  the angular velocity of the rear frame and  $\alpha$  the articulation angle.

Thus we have a relation between the rear axis velocity and the velocity of the front and rear wheels respectively. Hence the sensor models for the wheel speed sensors, measuring the longitudinal speed at the front wheels are

$$\begin{aligned}y_{fl} &= (1 + s_{fr})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r - \omega_z^f \frac{w_f}{2}) + e_{fl} \\ y_{fr} &= (1 + s_{fr})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r + \omega_z^f \frac{w_f}{2}) + e_{fr}\end{aligned}\quad (4.16)$$

where,  $v_x^r$  is the speed at point  $P_r$ ,  $\omega_z^r$  is the angular rotation about the z-axis in the rear vehicle frame,  $\omega_z^f$  is the angular rotation about the z-axis in the front vehicle frame,  $w_r$  is the length of the rear axis,  $e_{fl}$  and  $e_{fr}$  are white Gaussian noise,  $s_{fl}$  and  $s_{fr}$  are the slips for rear and right wheels respectively. The length from the middle of the rear axis to the articulation joint is denoted  $l_r$ , from articulation joint to the middle of the front axis is denoted  $l_f$ . The articulation angle is denoted  $\alpha$ .

## 4.4 Inertial Measurement Unit

An IMU measures the acceleration and turn rates around fixed axes. It typically includes a gyroscope and an accelerometer. The IMU used in this thesis is a three dimensional IMU, meaning that it measures the acceleration and turn rates about three axes perpendicular to each other. The IMU outputs angular velocities and accelerations in all direction and estimates the pitch and roll angles.

The measurement models described in this section are for full 3D motion, including error states which later will be shown not to be observable with the given sensors. The implemented sensor models, with the delimitations applied, are described below.

#### 4.4.1 Accelerometer

The IMU measures the motion of the rear body frame. It gives measurements relative to the inertial frame. The sensor model for acceleration thus becomes

$$\mathbf{y}_{acc} = (\mathcal{I} + \delta_{acc})\mathbf{a}_{imu} + \mathbf{b}_a - \mathbf{R}^{n \rightarrow r}\mathbf{g} + \mathbf{e}, \quad (4.17)$$

where the gravity needs to be rotated to the rear body frame and compensated for. Here  $\delta_{acc}$  is a scale error factor  $\mathbf{b}_a$  a bias error and  $\mathbf{R}^{n \rightarrow r}$  is the rotation matrix as described in Chapter 3,  $\mathcal{I}$  is the identity matrix and  $\mathbf{g}$  is the gravity vector in the inertial frame and  $\mathbf{e}$  is zero mean white Gaussian noise.

The gravity needs to be compensated for because the accelerometer measures specific force. It means that it measures acceleration and other forces acting on the sensors, such as gravity and the Coriolis force from earth's rotation. The Coriolis force is considered small and is therefore neglected. Typically the measured movement of the sensors is less than gravity.

#### Relating IMU measurements to the rear frame

As for the odometer measurements the measurements from the IMU needs to be translated to the middle of the rear axis, point  $P_r$  in Figure 4.1. Since the sensors are fixed in the vehicle they will have zero velocity relative to the rear axis, leading to the relationship

$$\mathbf{a}_{p_2} = \mathbf{a}_{p_1} + \dot{\omega} \times \mathbf{r}_{p_1 p_2} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{p_1 p_2}) \quad (4.18)$$

between the acceleration in the two points. See for example Essén [2002] for relating motion in different reference frames. Let  $\mathbf{l} = (l_x, l_y, l_z)$  be the distance vector from the mid rear axis to the IMU in the rear body frame. Using (4.18) then gives

$$\begin{pmatrix} a_{imu}^x \\ a_{imu}^y \\ a_{imu}^z \end{pmatrix} = \begin{pmatrix} a_r^x \\ a_r^y \\ a_r^z \end{pmatrix} + \begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} \times \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} + \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \times \left( \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \times \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \right) \Rightarrow \quad (4.19)$$

$$\begin{pmatrix} a_{imu}^x \\ a_{imu}^y \\ a_{imu}^z \end{pmatrix} = \begin{pmatrix} a_r^x + \dot{\omega}_y l_z - \dot{\omega}_z l_y + \omega_y(\omega_x l_y - \omega_y l_x) - \omega_z(\omega_z l_x - \omega_x l_z) \\ a_r^y + \dot{\omega}_z l_x - \dot{\omega}_x l_z + \omega_z(\omega_y l_z - \omega_z l_y) - \omega_x(\omega_x l_y - \omega_y l_x) \\ a_r^z + \dot{\omega}_x l_y - \dot{\omega}_y l_x + \omega_x(\omega_z l_x - \omega_x l_z) - \omega_y(\omega_y l_z - \omega_z l_y) \end{pmatrix} \quad (4.20)$$

where  $\omega_i$  and  $\dot{\omega}_i$  are the angular rate and angular acceleration about axis  $i$  respectively. Equation (4.20) describes the relationship between accelerations expressed at the IMU location and the mid point. It is clear that having the IMU located on the rear axis mid point will yield linear equations. Equation 4.20 shows that if the IMU is placed close to the rear axis an assumption of a linear relation can be used to have less complicated measurement equation. The further away the IMU is placed the less valid this assumption becomes.

#### 4.4.2 Gyroscope

The gyroscope measures turn rate in the rear vehicle frame relative to the inertial frame. This gives the turn rate for the whole rear body part since the turn rate of a rigid body is the same at every point. A generic measurement model for the gyroscope is

$$\mathbf{y}_{gyro} = (\mathbf{I} + \delta_{gyro})\boldsymbol{\omega}_i + \mathbf{b}_{gyro} + \mathbf{e} \quad (4.21)$$

where bold symbols indicates vectors and matrices and  $\boldsymbol{\omega}_i$  denotes the angular rate about axis  $i$ ,  $\delta_{gyro}$  is a scale error factor,  $\mathbf{b}_a$  a bias error and  $\mathbf{e}$  is white Gaussian noise. The vehicle can be seen as a rigid body in two degrees. The angular velocity of the front frame will differ from the rear frame due to the articulation joint.

### 4.5 Standard motion model

The motion model called the standard model is a constant acceleration model with where dynamics in angular velocity can be added. This model has been used by for example Corke and Ridle [2001] but it is often used without a derivation. In section 4.5.2 the derivation is briefly discussed.

#### 4.5.1 Motion model

Adding states for velocity and acceleration and the disturbance states as discussed above the motion model for the wheel loader in continuous time is

$$\frac{d}{dt} \begin{pmatrix} v_x^r \\ a_x^r \\ \omega_z^r \\ \delta_{o,acc} \\ \delta_{o,gyro} \\ \delta_{rl} \\ \delta_{rr} \\ \delta_{fl} \\ \delta_{fr} \end{pmatrix} = \begin{pmatrix} a_x^r \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \nu_{v_x^r} \\ \nu_{a_x^r} \\ \nu_{\omega_z^r} \\ \nu_{\delta_{o,acc}} \\ \nu_{\delta_{o,gyro}} \\ \nu_{rl} \\ \nu_{rr} \\ \nu_{fl} \\ \nu_{rr} \end{pmatrix} \quad (4.22)$$

where the bias in the sensor measurements are denoted  $\delta_i$ . The states  $v_x^r$ ,  $a_r$  and  $\omega_z^r$  are velocity, acceleration and yaw rate for the rear frame respectively. The

model noise  $v_i$  is modelled as additive white Gaussian noise. The model assumes constant acceleration which will affect estimates at changing accelerations. The measurements needed for this model are the four wheel speeds, yaw rate and longitudinal acceleration as

$$\begin{pmatrix} y_{rl} \\ y_{rr} \\ y_{fl} \\ y_{fr} \\ y_{acc} \\ y_{gyro} \end{pmatrix} = \begin{pmatrix} (1 + \delta_{rl})(v_x^r - \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{rr})(v_x^r + \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{fl})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r - \omega_z^f \frac{w_r}{2}) \\ (1 + \delta_{fr})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r + \omega_z^f \frac{w_r}{2}) \\ a_r^x + \omega_z^{r2} l_x + \delta_{o,acc} \\ \omega_z + \delta_{o,gyro} \end{pmatrix} + \begin{pmatrix} e_{rl} \\ e_{rr} \\ e_{fl} \\ e_{fr} \\ e_{acc} \\ e_{gyro} \end{pmatrix} \quad (4.23)$$

where the articulation angle  $\alpha$  is seen as input to the filter. The measurement of  $\alpha$  contains noise, however this is neglected in these models. The steering angle,  $\alpha$ , is accurately measured, leading to low noise levels. Furthermore  $l_x$  is the longitudinal distance from the mid rear axis to the IMU. In this model the lateral and vertical displacement of the IMU in (4.20) are assumed zero leading to the measurement equation for the accelerometer as above. The measurement noises  $e_i$  are modelled as additive Gaussian white noise.

### 4.5.2 Odometry

Additional dynamics for the standard model can be included by using odometry data. This could improve yaw and yaw rate estimates. The goal is to express the yawrate  $\dot{\psi}_z = \omega_z$  as

$$\omega_z = f(v, \dot{\alpha}). \quad (4.24)$$

The relationship between the two points according to Figure 4.2 is

$$x_f = x_r + l_r \cos(\psi_r) + l_f \cos(\psi_f), \quad (4.25a)$$

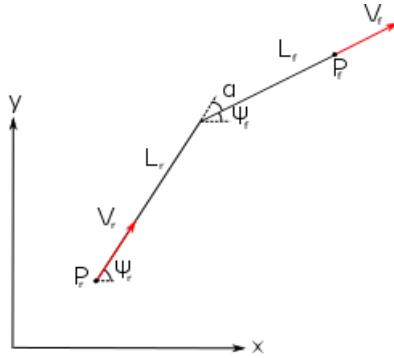
$$y_f = y_r + l_r \sin(\psi_r) + l_f \sin(\psi_f). \quad (4.25b)$$

For the velocity in  $P_1$  the relationship

$$\begin{aligned} \dot{x}_r &= v_r \cos(\psi_r) \\ \dot{y}_r &= v_r \sin(\psi_r) \end{aligned} \quad (4.26)$$

holds and the same relationship holds for  $P_2$ . The assumption of zero lateral slip can then derived from (4.26) as

$$\begin{aligned} \dot{x}_r \sin(\psi_r) &= \dot{y}_r \cos(\psi_r), \\ \dot{x}_f \sin(\psi_f) &= \dot{y}_f \cos(\psi_f). \end{aligned} \quad (4.27)$$



**Figure 4.2:** A picture describing the relationship between front and rear axis.

Differentiating (4.25) with respect to time gives

$$\dot{x}_f = \dot{x}_r - \dot{\psi}_r l_r \sin(\psi_r) - \dot{\psi}_f l_f \sin(\psi_f), \quad (4.28a)$$

$$\dot{y}_f = \dot{y}_r + \dot{\psi}_r l_r \cos(\psi_r) + \dot{\psi}_f l_f \cos(\psi_f). \quad (4.28b)$$

By inserting (4.27) and (4.26) in (4.28a) and multiplying this equation with  $\sin(\psi_f)$  and replacing  $\dot{y}_r$  according to (4.27) in (4.28b) and multiplying this equation with  $\cos(\psi_f)$  we obtain

$$\dot{y}_f \cos(\psi_f) = v_r \cos(\psi_r) \sin(\psi_f) - \dot{\psi}_r l_r \sin(\psi_r) \sin(\psi_f) - \dot{\psi}_f l_f \sin(\psi_f) \sin(\psi_f) \quad (4.29a)$$

$$\dot{y}_f \cos(\psi_f) = v_r \sin(\psi_r) \cos(\psi_f) + \dot{\psi}_r l_r \cos(\psi_r) \cos(\psi_f) + \dot{\psi}_f l_f \cos(\psi_f) \cos(\psi_f). \quad (4.29b)$$

Now letting (4.29a) equal (4.29b) and using trigonometric identities the resulting equation becomes

$$v_r \sin(\psi_f - \psi_r) = \dot{\psi}_r l_r \cos(\psi_r - \psi_f) + \dot{\psi}_f l_f. \quad (4.30)$$

Finally using the relationship  $\alpha = \psi_f - \psi_r$  to replace  $\dot{\psi}_r$  and  $\dot{\psi}_f$  respectively the front and rear yaw rates can be calculated as

$$\dot{\psi}_r = v_r \frac{\sin(\alpha)}{l_r \cos(\alpha) + l_f} - \dot{\alpha} \frac{l_f}{l_r \cos(\alpha) + l_f} \quad (4.31a)$$

$$\dot{\psi}_f = v_r \frac{\sin(\alpha)}{l_r \cos(\alpha) + l_f} + \dot{\alpha} \frac{l_r \cos(\alpha)}{l_r \cos(\alpha) + l_f}. \quad (4.31b)$$

## 4.6 Extended motion model

In this chapter the model is extended for full 3D motion thus trying to estimate the roll, pitch and yaw angles in the filter. To estimate all angles the rotational dynamics from Chapter 3 need to be included. This also means that all the IMU outputs are included as measurements. The extended motion model used in this thesis is

$$\begin{pmatrix} \phi^r \\ \theta^r \\ \psi^r \\ \omega_x^r \\ \omega_y^r \\ \omega_z^r \\ v_x^r \\ a_x^r \\ a_y^r \\ a_z^r \\ \frac{d}{dt} \delta_{o,acc}^x \\ \delta_{o,acc}^y \\ \delta_{o,acc}^z \\ \delta_{o,gyro}^x \\ \delta_{o,gyro}^y \\ \delta_{o,gyro}^z \\ \delta_{rl} \\ \delta_{rr} \\ \delta_{fl} \\ \delta_{fr} \end{pmatrix} = \begin{pmatrix} \omega_x^r - \sin \phi^r \tan \theta^r \omega_y^r + \cos \phi^r \tan \theta^r \omega_z^r \\ \cos \phi^r \omega_y^r + \sin \phi^r \omega_z^r \\ \frac{-\sin \phi^r}{\cos \theta^r} \omega_y^r + \frac{\cos \phi^r}{\cos \theta^r} \omega_z^r \\ 0 \\ 0 \\ 0 \\ a_x^r \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \nu_{\phi^r} \\ \nu_{\theta^r} \\ \nu_{\psi^r} \\ \nu_{\omega_x^r} \\ \nu_{\omega_y^r} \\ \nu_{\omega_z^r} \\ \nu_{v_x^r} \\ \nu_{a_x^r} \\ \nu_{a_y^r} \\ \nu_{a_z^r} \\ \nu_{\delta_{o,acc}^x} \\ \nu_{\delta_{o,acc}^y} \\ \nu_{\delta_{o,acc}^z} \\ \nu_{\delta_{o,gyro}^x} \\ \nu_{\delta_{o,gyro}^y} \\ \nu_{\delta_{o,gyro}^z} \\ \nu_{\delta_{rl}} \\ \nu_{\delta_{rr}} \\ \nu_{\delta_{fl}} \\ \nu_{\delta_{fr}} \end{pmatrix} \quad (4.32)$$

with measurement models as

$$\begin{pmatrix} y_{rl} \\ y_{rr} \\ y_{fl} \\ y_{fr} \\ y_{acc} \\ y_{gyro} \\ y_\phi \\ y_\theta \end{pmatrix} = \begin{pmatrix} (1 + \delta_{rl})(v_x^r - \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{rr})(v_x^r + \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{fl})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r - \omega_z^f \frac{w_r}{2}) \\ (1 + \delta_{fr})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r + \omega_z^f \frac{w_r}{2}) \\ \mathbf{a}_{imu} + \delta_{acc} + \mathbf{R}^{n \rightarrow b} \mathbf{g} \\ \boldsymbol{\omega} + \delta_{gyro} \\ \phi + \delta_\phi \\ \theta + \delta_\theta \end{pmatrix} + \begin{pmatrix} e_{rl} \\ e_{rr} \\ e_{fl} \\ e_{fr} \\ e_{acc} \\ e_{gyro} \\ e_\phi \\ e_\theta \end{pmatrix} \quad (4.33)$$

where  $R^{n \rightarrow b}$  is the rotation matrix as described in Chapter 3,  $\phi^r$ ,  $\theta^r$ ,  $\psi^r$  are roll, pitch, yaw respectively. The angular velocity and acceleration for each individual axle in the rear vehicle frame are denoted  $\omega_i^r$  and  $a_i^r$  respectively. The derivatives of angular accelerations in (4.20) are assumed to be small and are therefore omitted in the accelerometer measurement equations. This model sets higher requirements on state estimation than that of the standard model, especially due to the gravity in the accelerometer sensor models.

## 4.7 Zero velocity updates

In Chapter 2 it was mentioned that the odometers contain very little noise. This means that they can be a good indicator of whether the wheel loader is moving or not. This approach has been discussed in Woodman [2010] for indoor pedestrian localisation. A threshold is introduced for the average measured angular velocities as

$$|(\omega_{rl} + \omega_{rr} + \omega_{fl} + \omega_{fr})/4| < C \quad (4.34)$$

where  $\omega_i$  are the measured velocities from the odometers and  $C$  is a tuning constant. This constant can be empirically tested but should not be too large. In the zero velocity phase pseudo measurements can be implemented. At standstill the gyro only measure bias states and noise and the measurement equations can thus be written as

$$y_{gyro} = \delta_{o,gyro} + e \quad (4.35)$$

In order to stabilize the filter pseudo measurements of different states can be implemented with measurement values as 0. For example at standstill the velocity is zero, introducing a measurement equation as

$$y_{pseudo,v} = v_x^r + e = 0. \quad (4.36)$$

This is not necessary for the standard case but in the extended model observability issues leads to this giving better results. For example to stabilize the roll and pitch angles in the extended model the accelerometers can be used during the zero velocity phase since the accelerometer will give measurements on roll and pitch and bias errors during stand still.

## 4.8 Observability

Observability of the states is mandatory in order to estimate them. For non-linear systems local weak observability can easily be tested. Let

$$h_x = \left[ \frac{\partial h}{\partial x_1} \cdots \frac{\partial h}{\partial x_n} \right] \quad (4.37)$$

and

$$\begin{aligned} h_x^{(1)} &= \left[ \frac{\partial h}{\partial x_1} \dots \frac{\partial h}{\partial x_n} \right] f \\ h_x^{(2)} &= \left[ \frac{\partial h^{(1)}}{\partial x_1} \dots \frac{\partial h^{(1)}}{\partial x_n} \right] f \end{aligned} \quad (4.38)$$

The system is said to be weakly observable if the matrix

$$\begin{bmatrix} h_x(x) \\ h_x^{(1)}(x) \\ h_x^{(2)}(x) \\ \vdots \\ \vdots \\ h_x^{(n)}(x) \end{bmatrix} \quad (4.39)$$

fulfils an observability rank condition. The rank condition does not give any information on the quality of the estimates but only if it is possible to estimate a certain state or not. The rank test does not show how much noise a system can handle, and this needs to be tested in simulations instead. Local observability is a strict requirement. It means that at every time instance all the states can be uniquely calculated.

As in the linear case integrated states are not observable unless they are also directly measured. For example if velocity is measured and the position is wanted one must know the prior position in order to know the new position.

The dynamics of the states are also of importance. Slip is a fast changing state, whereas bias states in the sensors, which are changing with temperature, have slow dynamics. Different dynamics can be handled by the EKF by tuning the process and measurement noise appropriately. For further discussion on observability of nonlinear systems see for example Glad [2012].

### 4.8.1 Standard model

Using the motion and measurement models for the standard model discussed in section 4.5.1. Where  $\mathbf{x} = [v_r, a_r, \omega_z^r, o_{gyro}, o_{acc}, \delta_{rl}, \delta_{rr}, \delta_{fl}, \delta_{fr}]$ ,  $\omega_z^r = \omega_z^f - \dot{\alpha}$ . The inputs to the filer  $\alpha$  and  $\dot{\alpha}$  are considered constant in these calculations. The entire observability matrix for the models below,  $J = [h_x(\mathbf{x}), h_x^{(1)}(\mathbf{x}), h_x^{(2)}(\mathbf{x})]$ , can be seen in equations 4.42-4.46. This case assumes that the machine is moving and steering, i.e that  $\alpha \neq 0$  and  $\dot{\alpha} \neq 0$ . The matrix  $J$  has rank 9, which indicates that the system is observable according to the observability test in section 4.8. It is not obvious that the matrix has rank 9, this was calculated using Matlab®, see [Mat].

$$h(\mathbf{x}) = \begin{pmatrix} (1 + \delta_{rl})(v_x^r - \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{rr})(v_x^r + \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{fl})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r - \omega_z^f \frac{w_r}{2}) \\ (1 + \delta_{fr})(\cos(\alpha)v_x^r + \sin(\alpha)\omega_z^r l_r + \omega_z^f \frac{w_r}{2}) \\ a_x^r + \omega_z^{r2} l_x + \delta_{o,acc} \\ \omega_z + \delta_{o,gyro} \end{pmatrix} \quad (4.40)$$

and the motion model as

$$f(\mathbf{x}) = \begin{pmatrix} a_x^r \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.41)$$

$$h_{\mathbf{x}}(\mathbf{x})(:, 1 : 3) = \begin{pmatrix} \delta_{rl} + 1 & 0 & -\frac{w_r(\delta_{rl}+1)}{2} \\ \delta_{rr} + 1 & 0 & \frac{w_r(\delta_{rr}+1)}{2} \\ c(\alpha)(\delta_{fl} + 1) & 0 & -(\delta_{fl} + 1) \left( \frac{w_r}{2} - l_r s(\alpha) \right) \\ c(\alpha)(\delta_{fr} + 1) & 0 & (\delta_{fr} + 1) \left( \frac{w_r}{2} + l_r s(\alpha) \right) \\ 0 & 1 & 2 l_x \omega_z^r \\ 0 & 0 & 1 \end{pmatrix} \quad (4.42)$$

$$h_{\mathbf{x}}(\mathbf{x})(:, 4 : 7) = \begin{pmatrix} 0 & 0 & v - \frac{\omega_z^r w_r}{2} & 0 \\ 0 & 0 & 0 & v + \frac{\omega_z^r w_r}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (4.43)$$

$$h_x(\mathbf{x})(:, 8 : 9) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ v c(\alpha) - \frac{w_r(\omega_z^r + \dot{\alpha})}{2} + l_r \omega_z^r s(\alpha) & 0 \\ 0 & \frac{w_r(\omega_z^r + \dot{\alpha})}{2} + v c(\alpha) + l_r \omega_z^r s(\alpha) \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.44)$$

$$h_x^1(\mathbf{x}) = \begin{pmatrix} 0 & \delta_{rl} + 1 & 0 & 0 & 0 & a & 0 & 0 & 0 \\ 0 & \delta_{rr} + 1 & 0 & 0 & 0 & 0 & a & 0 & 0 \\ 0 & c(\alpha) (\delta_{fl} + 1) & 0 & 0 & 0 & 0 & 0 & a c(\alpha) & 0 \\ 0 & c(\alpha) (\delta_{fr} + 1) & 0 & 0 & 0 & 0 & 0 & 0 & a c(\alpha) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.45)$$

$$h_x^2(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.46)$$

## Going straight forward

A second interesting case is when the machine is moving straight forward on a line. Setting  $\alpha = \dot{\alpha} = 0$  in equation 4.40 yields equation 4.47. The observability matrix,  $J = [h_x(\mathbf{x}), h_x^{(1)}(\mathbf{x})]$ , for the measurement equation 4.47 and the motion model in 4.41 can be seen in equations 4.48-4.49. In contrast to the case while turning  $J$  now has rank 8, this was calculated the same way as in section 4.8.1. This indicates that not all states are observable while the machine is going

straight forward.

$$h(\mathbf{x}) = \begin{pmatrix} (1 + \delta_{rl})(v_x^r - \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{rr})(v_x^r + \omega_z^r \frac{w_r}{2}) \\ (1 + \delta_{fl})(v_x^r - \omega_z^f \frac{w_r}{2}) \\ (1 + \delta_{fr})(v_x^r + \omega_z^f \frac{w_r}{2}) \\ a_r^x + \omega_z^{r2} l_x + \delta_{o,acc} \\ \omega_z + \delta_{o,gyro} \end{pmatrix} \quad (4.47)$$

$$h_{\mathbf{x}}(\mathbf{x}) = \begin{pmatrix} \delta_{rl} + 1 & 0 & -\frac{w_r(\delta_{rl}+1)}{2} & 0 & 0 & v - \frac{\omega_z^r w_r}{2} & 0 & 0 & 0 \\ \delta_{rr} + 1 & 0 & \frac{w_r(\delta_{rr}+1)}{2} & 0 & 0 & 0 & v + \frac{\omega_z^r w_r}{2} & 0 & 0 \\ \delta_{fl} + 1 & 0 & -\frac{w_r(\delta_{fl}+1)}{2} & 0 & 0 & 0 & 0 & v - \frac{\omega_z^r w_r}{2} & 0 \\ \delta_{fr} + 1 & 0 & \frac{w_r(\delta_{fr}+1)}{2} & 0 & 0 & 0 & 0 & 0 & v + \frac{\omega_z^r w_r}{2} \\ 0 & 1 & 2 l_x \omega_z^r & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.48)$$

$$h_{\mathbf{x}}^1(\mathbf{x}) = \begin{pmatrix} 0 & \delta_{rl} + 1 & 0 & 0 & 0 & a & 0 & 0 & 0 \\ 0 & \delta_{rr} + 1 & 0 & 0 & 0 & 0 & a & 0 & 0 \\ 0 & \delta_{fl} + 1 & 0 & 0 & 0 & 0 & 0 & a & 0 \\ 0 & \delta_{fr} + 1 & 0 & 0 & 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.49)$$

## 4.8.2 Extended model

The most important difference between the standard and extended models are the requirements of highly accurate pitch and roll estimates for the extended model. The gravity vector needs to be rotated from the inertial frame to the body frame. With less accurate roll and pitch estimates the gravity will affect the accelerometer measurements strongly and the filter might diverge. If roll, pitch, roll rate and pitch rate are measured the biases in the gyroscope are observable. The system is not fully observable as the vertical and lateral accelerometers needs a reference in order for the bias states to be observable.



# 5

---

## State estimation

In this chapter different methods for state estimation are discussed. Pre-processing and the Kalman filter with possible improvements are described.

### 5.1 Pre-processing

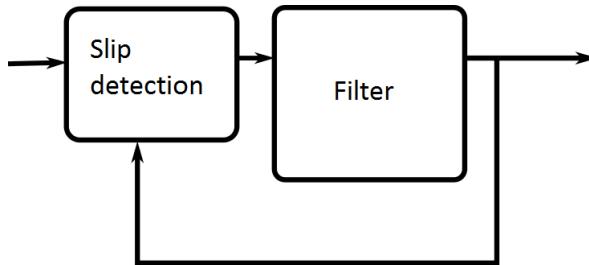
Pre-processing the data is an important step in state estimation. Data can be pre-processed, making it possible to estimate and compensate for disturbances such as scaling factors and biases in the sensors. If the bias states are estimated offline they can be removed from the state space models in Chapter 4. If a system is not fully observable, as discussed in Chapter 4, estimating the biases offline can improve results. However this method does not compensate for time varying biases. Pre-processing to identify bias states is therefore not robust in the long term. The zero velocity phase introduced in Chapter 4 is one way of implementing this in a real-time application.

Pre-processing the sensor values from the wheel speed sensors before using them as input into a filter could be one way of improving the results. The slip states in the motion models in Chapter 4 could then be removed.

In Figure 5.1 a simple sketch can be seen. One way to implement this slip detection is to feed back the estimated velocity from the filter and create residuals as

$$\epsilon_i = y_i - \hat{v}_i \quad (5.1)$$

where  $y_i$  is a measurement for wheel  $i$  and  $\hat{v}_i$  is the estimated velocity for that wheel. Then if  $\epsilon_i$  is larger than a certain threshold the specific sensor can be neglected. That is, to neglect one or several of the wheel speed sensors while the



**Figure 5.1:** Introducing a sensor validation before filtering the signals.

slip is too large in order to keep the estimate correct. This involves the assumption that slip is fast changing and will be detected without needing a big delay in the estimate. In the section on slip in Chapter 4 it was noted that in a four wheel drive all wheels can be subject to slip. During all wheel slip dead reckoning of the IMU data could be used.

## 5.2 Kalman filtering

Filtering is an extension of estimation problems to non stationary parameters  $x$ , Gustafsson [2012]. When the estimated parameters  $x$  are changing according to a dynamical model the estimation problem becomes a filtering problem.

The Kalman filter (KF), was first introduced in Kalman [1960]. For linear models this is the best unbiased filter, with respect to estimation variance and estimation error. The Kalman filter iterates between a time update and a measurement update. The time update involves predicting the next states using a motion model, describing the dynamics of the states. The measurement update then corrects the predictions using measurement data. Motion models often describe time continuous systems. The measurement models gives measurement in discrete time with an interval of  $T$ , the sampling time. The motion model thus needs to be discretized. Different approaches to this exist and are earlier briefly discussed in Chapter 4.

Real world systems are often nonlinear which the KF cannot handle. This problem has been dealt with in different ways, and one solution is called the extended Kalman filter (EKF). The EKF is a widely used filter but is not guaranteed to be the best unbiased filter or even to converge. The EKF is described below but there exist numerous other approaches for nonlinear state estimation such as particle filters and points mass filters, see Gustafsson [2012].

In a general discretized form the system equations can be written as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \quad (5.2a)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{e}_k \quad (5.2b)$$

where  $\mathbf{x}$  denotes the states being estimated,  $\mathbf{u}$  are known control signals and  $\mathbf{y}$  are measurements. The index  $k$  denotes a discrete time instance,  $\mathbf{v}_k$  is process noise and  $\mathbf{e}_k$  similarly is called measurement noise. Both the process and measurement noises are modelled as zero mean Gaussian noise

$$\begin{aligned} \mathbf{v}_k &\sim \mathcal{N}(0, Q_k) \\ \mathbf{e}_k &\sim \mathcal{N}(0, R_k), \end{aligned} \quad (5.3)$$

where  $R_k$  is the noise covariance, and for uncorrelated signals this is a diagonal matrix.  $Q_k$  is the process noise variance. The process noise covariance is a tuning parameter in the Kalman filter. Setting a high process noise results in a fast changing state and vice versa. Possibilities for auto tuning the process noise also exists, see for example [Forsberg].

There are different approaches to extended Kalman filtering. One approach is to linearize the dynamical models around the last state estimate. The approach used here is a first order Taylor expansion around the last state estimate. The algorithm can be seen in Algorithm 1. In Algorithms 1 and 2 a double time index is used, where  $i|j$  means time  $i$  given measurements up to measurement  $j$ .

### Algorithm 1 Extended Kalman Filter.

Initialize with  $\hat{\mathbf{x}}_{1|0}$  and  $P_{1|0}$ .

#### Measurement update.

$$\begin{aligned} S_k &= R_k + H_{k|k-1} P_{k|k-1} H_{k|k-1}^T \\ K_k &= P_{k|k-1} H_{k|k-1}^T S_k^{-1} \\ \epsilon_k &= \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}) \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k \epsilon_k \\ P_{k|k} &= P_{k|k-1} - P_{k|k-1} H_{k|k-1}^T S_k^{-1} H_{k|k-1} P_{k|k-1} \end{aligned} \quad (5.4)$$

#### Time update.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= f(\hat{\mathbf{x}}_{k|k}) \\ P_{k+1|k} &= F_{k|k} P_{k|k} F_{k|k}^T + Q_k \end{aligned} \quad (5.5)$$

Here  $H_{k|k-1}$  and  $F_{k|k}$  are the Jacobians of  $f$  and  $h$  defined as

$$\begin{aligned} F_{k|k} &= f'(\hat{\mathbf{x}}_{k|k}) \\ H_{k|k-1} &= h'(\hat{\mathbf{x}}_{k|k-1}) \end{aligned} \quad (5.6)$$

The design parameters here are the measurement and motion models for the system and the process noise  $Q$ . The models are discussed in Chapter 4. The algorithm above is for a standard EKF. The EKF works well when the system is not too non-linear. Several improved versions of this exists, some of these are discussed below.

## 5.3 Square root implementation

The standard EKF can have numerical problems when updating the state covariance matrix  $P$ . Some problems listed in Gustafsson [2012] are:

- $P$  is unsymmetric.
- $P$  is not positive definite
- Due to an almost singular  $R$  matrix there might be problems calculating  $P$ .

Some of these problems can be handled individually but the square root implementation solves all the problems and gives more robust computations. The idea is to update the square root of the covariance matrix. The square root of a matrix is defined as

$$P = P^{1/2}(P^{1/2})^T \quad (5.7)$$

where  $P^{1/2}$  is quadratic and has the same dimension as  $P$ . Below follows an example for the time update. The time update in the EKF is

$$P_{k+1|k} = F_{k|k}P_{k|k}F_{k|k}^T + Q_k. \quad (5.8)$$

Factorizing (5.8) gives

$$\begin{pmatrix} F_k P_{k|k}^{1/2} & Q_k^{1/2} \end{pmatrix} \begin{pmatrix} F_k P_{k|k}^{1/2} & Q_k^{1/2} \end{pmatrix}^T. \quad (5.9)$$

If the QR factorisation, see for example Anton and Rorres [2014], is applied to the first factor in (5.9) the result becomes

$$\begin{pmatrix} F_k P_{k|k}^{1/2} & Q_k^{1/2} \end{pmatrix} = R^T Q^T \quad (5.10)$$

where  $R$  and  $Q$  has nothing to do with the covariance of measurements and process noise. With (5.9) and (5.10), the time update can be rewritten as

$$P_{k+1|k} = R^T R. \quad (5.11)$$

and the square root of the state covariance matrix can be identified as

$$R^T = \begin{pmatrix} P_{k+1|k}^{1/2} & 0 \end{pmatrix}. \quad (5.12)$$

This way the square root of the estimation covariance can be updated. Similar calculations can be done for the measurement update. The calculations for the Kalman filter are written in Gustafsson [2012] but the calculations for the EKF are identical. Below follows the resulting algorithm for a square root implementation of the EKF.

**Algorithm 2** Square root implementation of EKF.

**Measurement update.**

1. QR factorize the matrix as

$$\begin{pmatrix} R_k^{1/2} & H_k P_{k|k-1}^{1/2} \\ 0 & P_{k|k-1}^{1/2} \end{pmatrix} = R^T Q^T = \begin{pmatrix} S_k^{1/2} & 0 \\ K_k S_k^{1/2} & P_{k|k}^{1/2} \end{pmatrix} Q^T \quad (5.13)$$

2. Update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k S_k^{1/2} S_k^{-1/2} (y_k - h(\hat{x}_{k|k-1})). \quad (5.14)$$

**Time update.**

3. QR factorize the matrix as

$$\begin{pmatrix} F_k P_{k|k}^{1/2} & Q_k^{1/2} \end{pmatrix} = \begin{pmatrix} P_{k+1|k}^{1/2} & 0 \end{pmatrix} \quad (5.15)$$

4. Predict the next state

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) \quad (5.16)$$

## 5.4 Smoothing

Smoothers is one way of improving state estimates, and the general idea behind smoothing is to use more than one measurement for each estimation. Different types of smoothers exist, for example Rauch-Tung-Striebel (RTS) and fixed lag smoothers. The RTS-smoother calculates the forward filtering equations and saves all the estimates and covariance. The saved estimates are then used to calculate new estimates, starting from the end. This can be a good implementation for off-line applications as it requires the data to be saved. A fixed lag smoother can be implemented in a real time application. The idea behind the fixed lag smoother is to augment the state vector with delayed states. By doing this the smoother uses measurements up to time  $k + n$ , where  $n$  is a chosen delay, to estimate the states at time  $k$ . This means that the resulting estimate is delayed  $T_s n$  seconds. Below is the code for a fixed lag Kalman smoother as this was the chosen smoother for this application.

The first step is to augment the state vector with the delayed states as

$$\mathbf{x}_k^i = \mathbf{x}_{k-i}, i = 1, \dots, n + 1 \quad (5.17)$$

which increases the dimension of the state vector by a factor  $n + 2$ . The state vector becomes

$$\bar{x}_k = \begin{bmatrix} x_k^0 \\ \vdots \\ x_k^{n+1} \end{bmatrix} \quad (5.18)$$

Below is the example for using a delay of two samples in the linear case. For the generic system

$$\begin{aligned} \mathbf{x}_{k+1} &= F\mathbf{x}_k \\ y &= H\mathbf{x}_k \end{aligned} \quad (5.19)$$

where  $\mathbf{x}$  is the state vector and  $H$  and  $F$  are matrices describing the dynamics. The resulting system model becomes

$$\bar{f} = \begin{pmatrix} F & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{pmatrix} \quad (5.20)$$

$$\bar{h} = \begin{pmatrix} H & 0 & 0 & 0 \end{pmatrix} \quad (5.21)$$

where  $I$  is a unit matrix and 0 represents a zero matrix, both of the same dimensions as  $F$ . Using (5.18), (5.20) and (5.21) in the Kalman filter algorithm yields the optimal smoother for the system in (5.19).

For the linear case a smoother generally gives improved estimates. For the non-linear case the results vary between different cases. In velocity estimation a fixed lag might yield problems. For example if the slip is to be estimated and used in an Anti-lock braking system (ABS) even a small lag can be crucial. Slip is a fast changing state meaning that the lag can make the ABS compensate for slipping wheels to late.

# 6

---

## Simulations

This chapter presents the test set up for simulations and the resulting velocity estimates for simulated data. All simulations were performed in Matlab® and Simulink®, see [Mat].

### 6.1 Test scenarios

In both simulations and real world tests two representative driving cycles were chosen to evaluate the filters. These are called transportation cycle and short cycle loading and are described below. These where chosen because they represent common use cases for wheel loaders and they are also used by Volvo.

The models tested in the EKF in simulations are variations on the models described in chapter 4 and are the following:

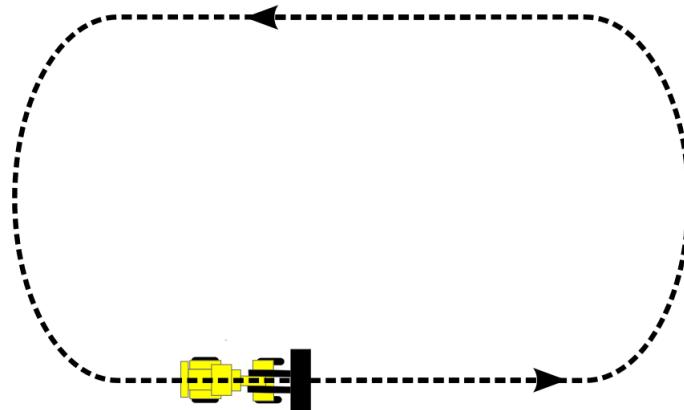
- Standard model without slip states. This is the model described in section 4.5.1, however all slip states  $\delta_{ii}$  have been removed.
- Standard model without slip states, with residuals as discussed in section 5.1 to handle slip.
- Standard model as described in section 4.5.1.
- Standard model with an aiding velocity sensor, this is done to simulate adding a GPS measurement to stabilize the filter.
- Extended model as described in section 4.6.

The filters are also extended to smoothers as discussed in section 5.4.

### 6.1.1 Transportation

During transportation the machine moves at almost constant velocity and often on fairly even ground. The weight of the machine will be constant meaning that the wheel radius is not dramatically changed. The transportation cycle used in this thesis is presented in Figure 6.1.

Normally during verification this is a first test of functionality. If the implementations fail to estimate states during transportation they have a low chance of handling the short cycle load. However in this thesis going straight forward might cause problems due to observability issues discussed in chapter 4.

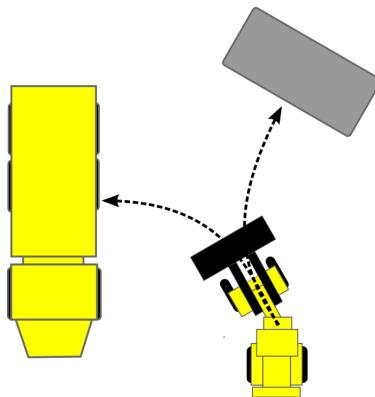


*Figure 6.1: Transportation cycle. Used as an easy estimation case.*

### 6.1.2 Short cycle loading

When the wheel loader is loading material onto a nearby load carrier the cycle is called short cycle loading (SCL). The general movement of the wheel loader can be seen in Figure 6.2.

This cycle represents difficult conditions for state estimation. The reason is that the machine changes speed and direction of travel at a fast pace and stops abruptly when hitting the pile. When loading and unloading material in the bucket the center of gravity and weight of the machine change. This will affect the tire radius, which in this thesis is assumed to be constant.

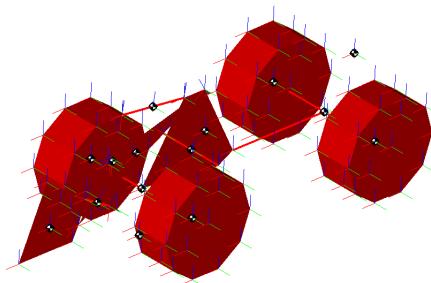


**Figure 6.2:** Schematic over a wheel loader loading gravel onto a nearby load carrier, representing short cycle loading.

## 6.2 Simulation tests

The simulations were performed using a Simulink® model developed by Volvo, see Figure 6.3. This model includes a 3D chassis model and tyre dynamics according to the Magic formula, see Pacejka [2006] for a discussion on the Magic formula, which also calculates wheel slip for all individual wheels. A slip of 0% – 5% is considered to be a low value and anything above is considered a high value.

Noise was added to the simulated measurements according to the table in Chapter 2. During this thesis work it was debated whether to add a GPS or not. In simulations this was tested by adding a velocity measurement for the mid rear axis with a sampling frequency of 1Hz.



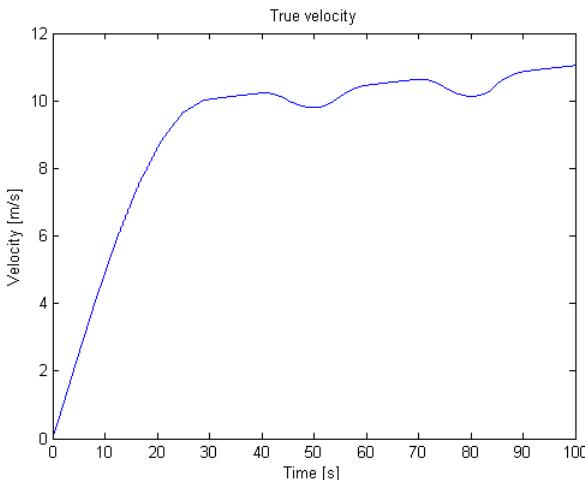
**Figure 6.3:** Picture depicting the simulation model developed by Volvo.

### 6.2.1 Transportation

The transport cycle was approximated as a very simple test scenario in the simulations. Figure 6.4 shows the true velocity for the transport cycle in the simulations i.e. the velocity in the middle of the rear axis. The vehicle accelerates slowly and then travels at constant velocity and turns at around 50s and 80s. Figure 6.5 shows the wheel slip for each individual wheel during transport. The slip is taken directly from the calculated wheel slip of the model. At approximately 80s the front right wheel has substantial slip. In reality this could for example mean a patch of ice. Apart from the slip at 80s the slip values are low.

### 6.2.2 Short cycle loading

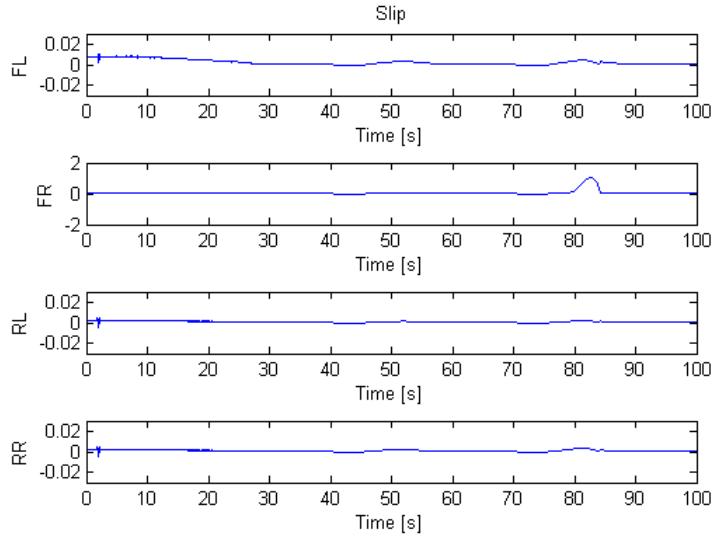
The true velocity for the SCL can be seen in Figure 6.6. The slip for all wheels in the high slip case can be seen in Figure 6.7. The slip is here taken from the calculated slip from the model and enhanced to simulate a slippery surface. The slip on all wheels are similar in shape but have different magnitude. This scenario is considered more realistic than the transportation scenario.



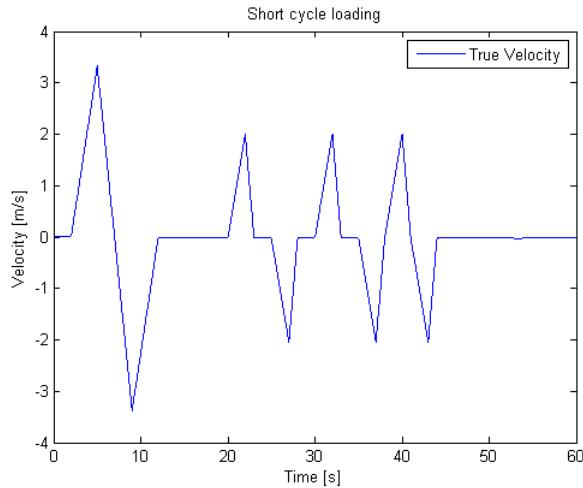
*Figure 6.4: Actual velocity for the simulated transport cycle.*

### 6.3 Standard model without slip states

In this section the results from using an EKF with the standard model discussed in Chapter 4 is presented. However, the slip states in the measurement equations

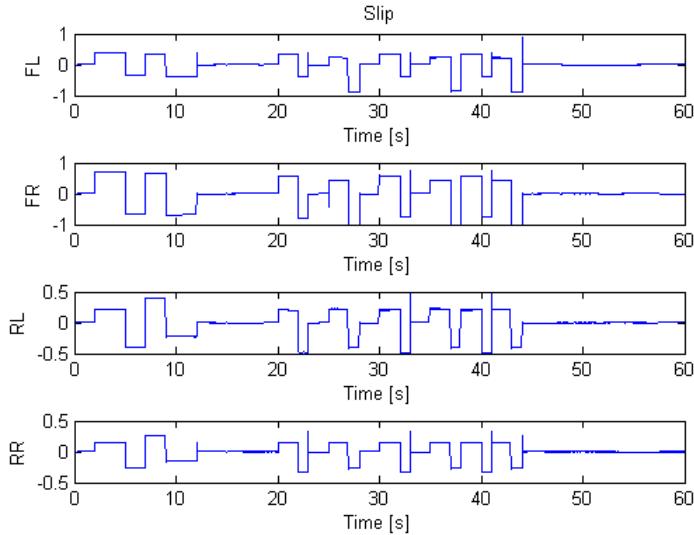


**Figure 6.5:** Slip for all four wheels during transport.



**Figure 6.6:** Actual velocity for the simulated SCL.

are omitted. The velocity estimate is then compared to the average over the four wheel speeds, and this is done since the average over the four wheel speeds is used as reference in the real world tests. The state vector for this implementation



**Figure 6.7:** Slip for all individual wheels during SCL on a slippery surface.

is

$$\mathbf{x} = (v_x^r, a_x^r, \omega_z^r, \delta_{o,acc}, \delta_{o,gyro})^T. \quad (6.1)$$

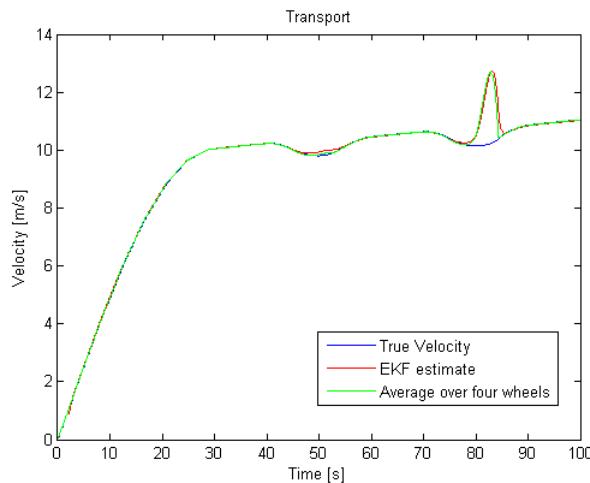
Both bias in the gyro and accelerometer are kept since they are both observable and should improve the estimates. Generally using only wheel speed sensors gives an accurate velocity estimate when the wheels are not slipping too much. The idea behind testing this filter is to evaluate if fusing an IMU with wheel speeds measurements without explicitly handling slip might decrease the velocity estimate error at slip.

The filter proved to be robust in the sense of tuning. Generally a lot of process noise is added to the last derivative of a state in the process model, therefore high covariances for  $a_x^r$  and  $\omega_z^r$  were used. The tuning of the bias states was set to a low covariance due to the slow dynamics of time varying biases as discussed in Chapter 4.

The filter does not handle slip which means that for slipping wheels the wheel speed sensors should be trusted less than the IMU. However a large measurement covariance for the wheel speed sensors proved to give worse results than setting the measurement covariance low. Since the wheel speed sensors also measure yaw rate, trusting them less will make the filter use the gyroscope as the only yaw rate measurement. Trusting the IMU more gave better results when slipping but worse results when turning.

Fusing IMU data with wheel speed sensor data proved to make little difference compared to not using the IMU, when omitting the slip. The filter velocity estimate closely resembles that of the average over the four wheels. In situations of little slip the velocity estimate follows the true velocity. Since the filter fails to capture slip it will always show a false velocity for four wheel drive. Naturally at low wheel slips it does not make the results unreliable but at high slip values the estimates will be unreliable.

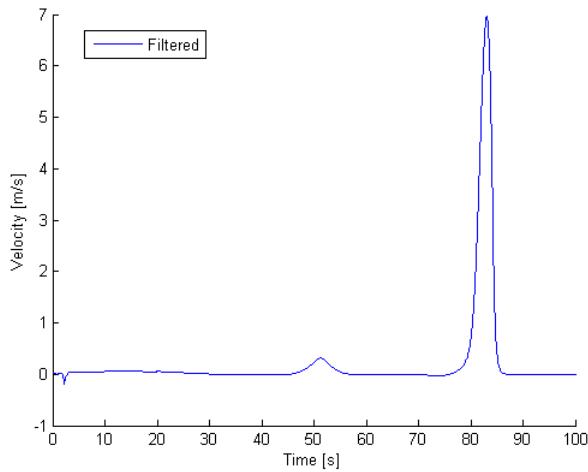
The results for the standard model without slip can be seen in Figures 6.8 and 6.10 with residuals in Figures 6.9 and 6.11.



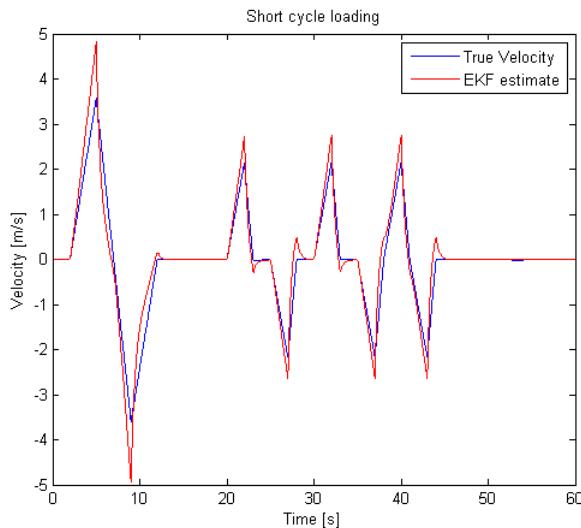
**Figure 6.8:** Estimated velocity for the standard model EKF without slip states during transport.

### 6.3.1 With slip handling

In this section the results from using the standard model and adding the residuals as discussed in Section 5.1 are presented. This solution is not as robust as the EKF without residuals. The reason for this being that if the velocity estimate starts to drift from the true value the filter will stop using the wheel speed sensors and therefore be dead reckoning using the IMU. For low slip values the filter will behave as the filter without pre-processing. At high slip values it improves the estimate. The tuning of this filter is done in a similar way as for the case without

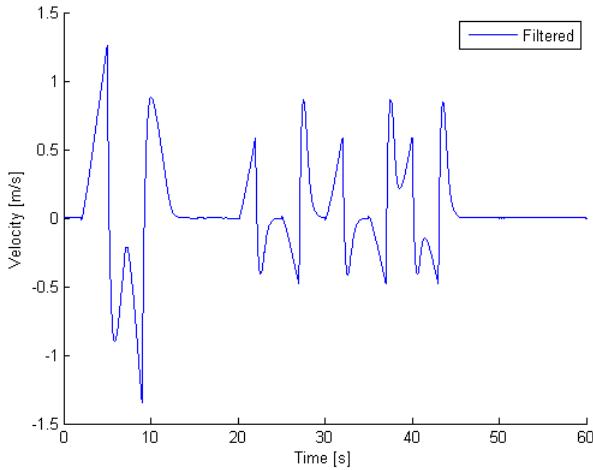


**Figure 6.9:** Residual for the standard model EKF without slip states during transport.



**Figure 6.10:** Estimated velocity for the standard model EKF without slip states during SCL.

residuals since the only difference is that slipping wheel sensors are omitted in the filter equations.



**Figure 6.11:** Residual for the standard model EKF without slip states during SCL.

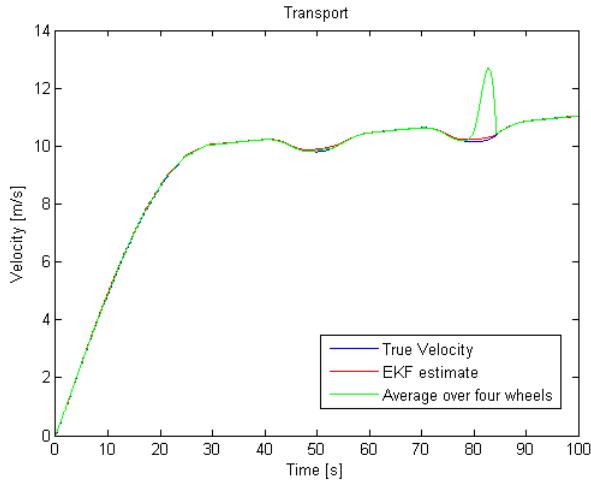
## Transport

The velocity estimate for the transport cycle can be seen in Figure 6.12 and the residual in Figure 6.13. It is obvious from the implementation that this filter will be unable to handle low slip values. Tuning the threshold down could help handling the lower values of slip. The filter estimates the velocity accurately along the whole transport cycle. During 0 – 30s in Figure 6.13 shows the errors of the constant acceleration model, although this error is small. At 50s and 80s the wheel loader is turning. During turning all wheels are subject to a small slip. Although comparing the slip in Figure 6.5 with the residual in Figure 6.13 shows that the estimation error during turning is not only due to slip. The filter suffers at turning as well, although the precision is still considered adequate. At around 80 seconds the filter still estimates the velocity with the same precision although the slip is very high which is a great improvement from the model without wheel sensor residuals.

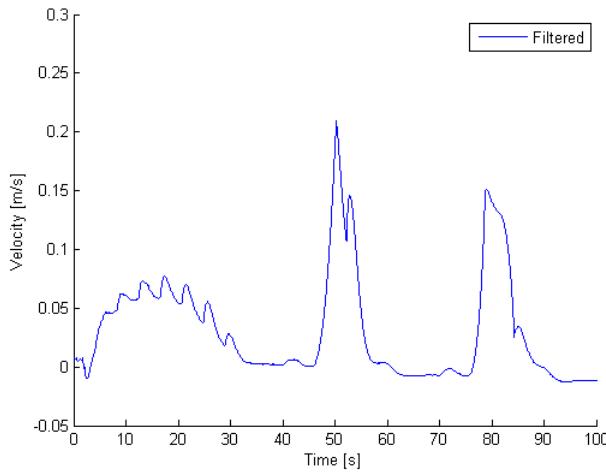
## Short cycle load

The velocity estimate for the short cycle loading can be seen in Figure 6.14 and the residual in Figure 6.15. The filter handles the SCL worse than the Transport cycle but still better than without pre-processing. From Figure 6.15 it is clear that the estimation error is larger than that for the transport cycle. One reason could be that the slip during SCL is very similar on all wheels, changing the estimate from good to bad quickly making the residuals fail to capture slipping wheels. The behaviour of the wheel loader during the simulated SCL also contains drastic changes in velocity which makes the assumption of constant acceleration worse.

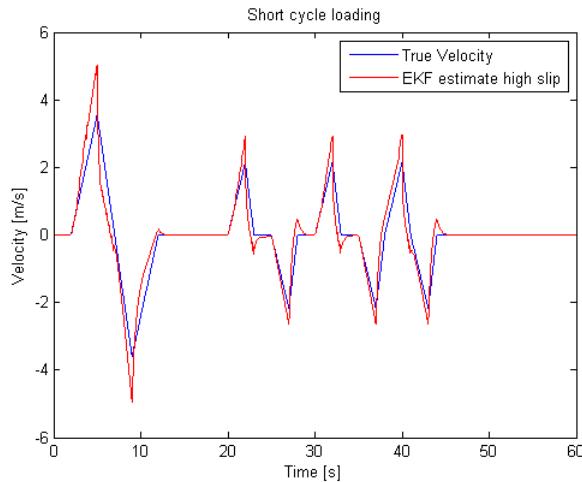
The filter showed slight errors during acceleration during transportation, in the SCL the acceleration is bigger and faster changing which makes the filter handle this situation worse.



**Figure 6.12:** Estimated velocity using a sensor validity test without slip states.



**Figure 6.13:** Residual for the no slip sensor check during transportation.

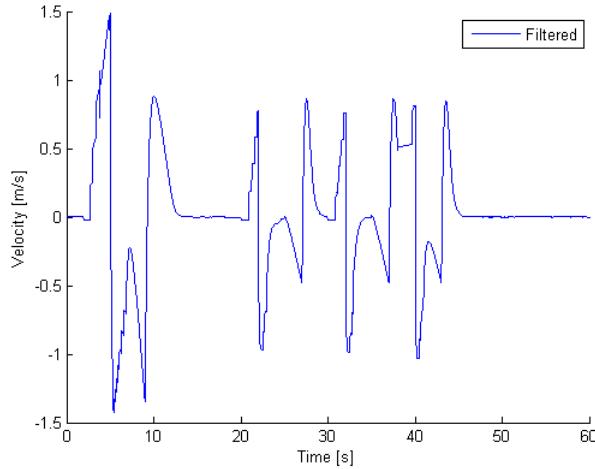


**Figure 6.14:** Estimated velocity for the EKF using no slip variables with sensor validity test.

## 6.4 Standard model

This section includes the simulation results for the standard model described in Chapter 4, with the state vector as

$$\boldsymbol{x} = (v_x^r, a_x^r, \omega_z^r, \delta_{o,acc}, \delta_{o,gyro}, \delta_{rl,slip}, \delta_{rr,slip}, \delta_{fl,slip}, \delta_{fr,slip})^T. \quad (6.2)$$



**Figure 6.15:** Residual for the no slip sensor check during SCL.

The EKF is sensitive to tuning. It is possible to tune the filter to give similar results as for the EKF without slip states. This is done by letting the dynamics in the slip states be very slow. The results here are shown with a slow changing slip, results with faster dynamics prove to be worse. An observability test showed that all the states are observable. However this means that also the articulation angle and angular rate have non-zero values. When not turning the filter seems to suffer from observability issues, the reason for this is that the filter estimates for slip does not remotely resemble the true slip.

The results regarding the standard model need more testing in order to investigate why the filter cannot handle the transport cycle but can handle the SCL.

## Transport

The velocity estimate from the transport cycle can be seen in Figure 6.16. The standard model fails at the transport cycle before any high slips occurs. Already at around 4 seconds the filter starts losing track of the velocity. Then it vaguely follows the true velocity with an offset until 80 seconds where the high slip value occurs. The reason the filter loses track of the true velocity already at the acceleration is due to the calculated acceleration in the simulation model. At around 2.5 s the accelerometer measurements contains noise of an amplitude of  $2\text{m/s}^2$ , meaning that the vehicle is vibrating a lot.

When turning the filter also loses track of the bias states in the IMU. At 80 seconds the filter completely loses track of the velocity. The reason the filter is unable to handle this case could be due to the simulated data being unrealistic, as in the

real world test other results can be seen in Chapter 7.

### Short cycle load

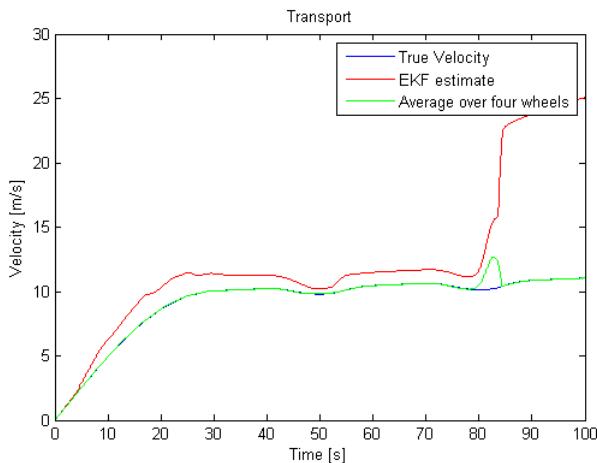
The filter handles the SCL better, see Figure 6.17 for the velocity estimate and 6.18 for the residual. At the start the estimate of the velocity is poor, however after around 25 seconds the residual values are comparable to that of the filter with sensor residuals in Figure 6.15. These results are interesting enough to further test the model on real world data.

#### 6.4.1 Adding an aiding sensor

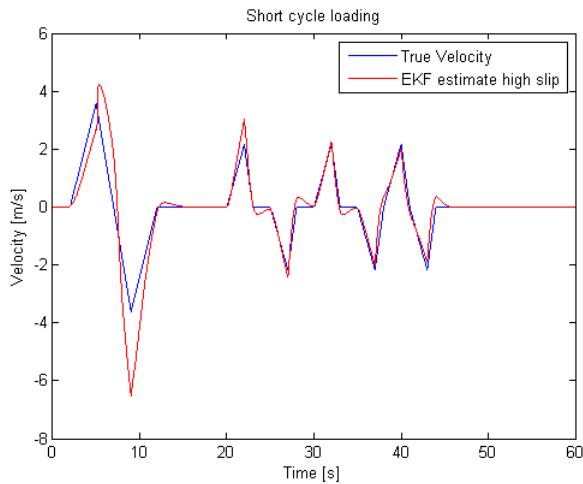
Adding an aiding sensor was thought to solve the observability issues. A 1 Hz velocity measurement that would simulate GPS measurements was added in simulations. The filter showed improved results, especially during the transport cycle. The estimate now converges quicker, but the filter is still sensitive to tuning. Adding another sensor allows the slip states to be tuned to have faster dynamics without diverging.

The aided filter handles the high slip scenario better than the filter without an extra sensor. Otherwise the extra sensor did not improve the estimates.

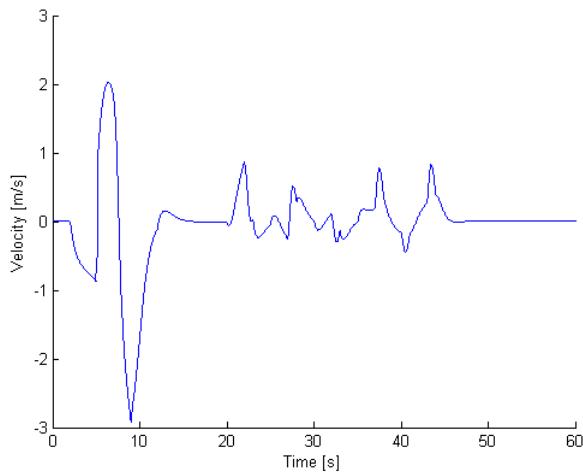
The resulting velocity estimate for the transport case and residual with an added aiding sensor can be seen in figures 6.19 and 6.20 respectively.



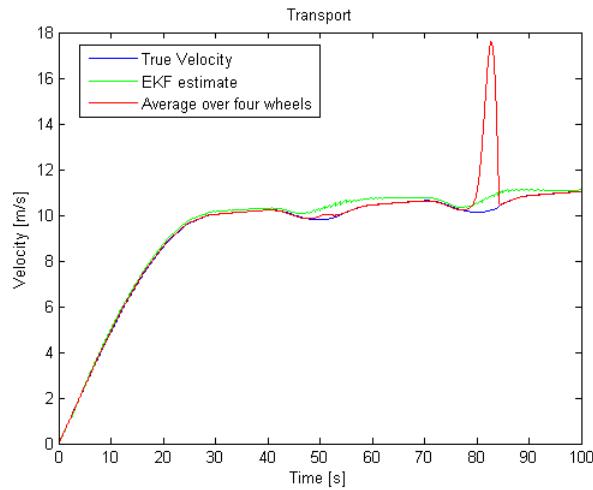
**Figure 6.16:** Actual velocity for the simulated transport cycle.



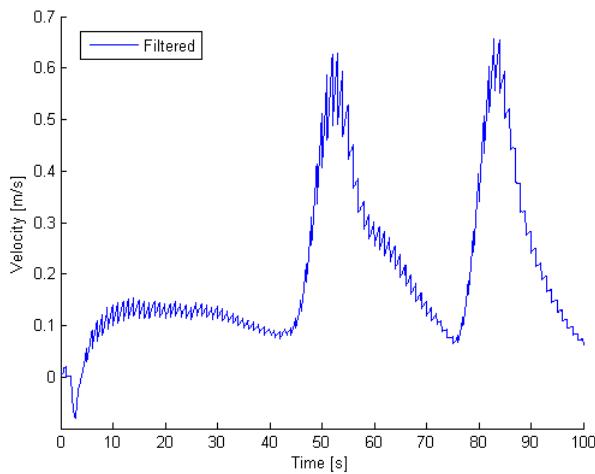
**Figure 6.17:** Estimated velocity during SCL for the standard model.



**Figure 6.18:** Residual for the simulated SCL using the standard model.



**Figure 6.19:** Estimated velocity for the standard model with a 1Hz aiding sensors added during transport.



**Figure 6.20:** Residual between estimated velocity and true velocity for the standard model with a 1Hz velocity aiding sensors added during transport.

## 6.5 Extended model

This chapter presents the velocity estimate using an EKF with the extended model, with the state vector as

$$\mathbf{x} = (\phi, \theta, \psi, \omega_x^r, \omega_y^r, \omega_z^r, v_x^r, a_x^r, a_y^r, a_z^r, \delta_{o,acc}, \delta_{o,gyro}, \delta_{rl}, \delta_{rr}, \delta_{fl}, \delta_{fr}) \quad (6.3)$$

Due to the uncertainty in the results regarding the standard model with slip states this was not tested for the extended model. The difference between the standard and extended models are that the pitch and roll angles need to be estimated, therefore the gyro and accelerometer in all directions must be included. The accelerometers support the convergence of the pitch and roll angles at standstill since no acceleration is measured, except for gravity.

### Transport

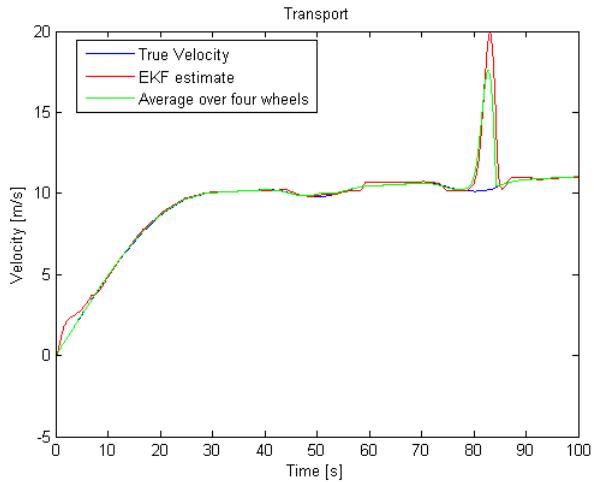
The velocity estimate for the transport cycle can be seen in Figure 6.21 and the residual in Figure 6.22. At the start the filter takes about 5 seconds before converging to an estimate close to the true velocity. Interesting to note is that the filter handles the slip part at around 40 seconds better than the standard model. At the high slip values around 80 seconds this filter shows a worse result than the average over four wheels. This could be a tuning issue as the filter is sensitive to tuning.

### Short cycle load

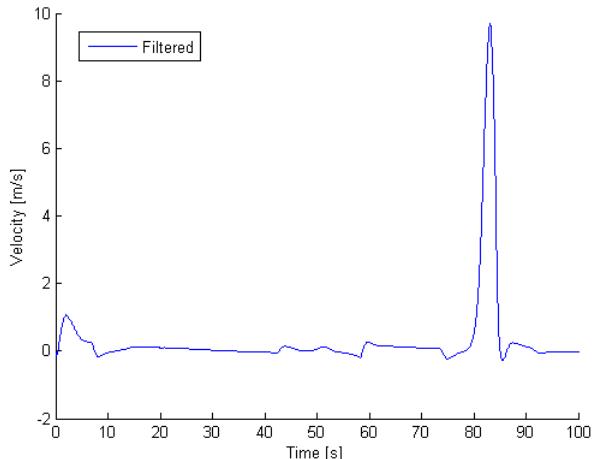
The resulting velocity estimate for SCL can be seen in Figure 6.23 and the residual in Figure 6.24. The extended model behaves a little worse than the standard model without slip. As for the no slip standard model the filter estimates the velocity well at zero to low slip but at high slip it fails. The filter gives similar results to that of the standard model no slip case. As the extended model enables applicability on non-flat ground these results are more interesting than those for the standard model. The filter gives good enough estimates to be further tested on real data.

## 6.6 Smoother

Implementing a fixed-lag smoother is easy and fast to implement and might improve the estimates. For the non-linear case the results from implementing a smoother differs for every filter. A fixed-lag smoother was implemented for all filters with a time lag of 2 and 10 samples. A sampling rate of 50Hz and a lag of 10 samples equals 0.2s delay. This might be too small to capture the needed information to make the smoother give more accurate estimates. The downside of using a fixed lag smoother is that the filter becomes computationally heavier due to the expanded state vector. The smoothers did not improve the estimates enough to make up for the increased computational load. Therefore it was concluded not to test the smoothers on real world data.

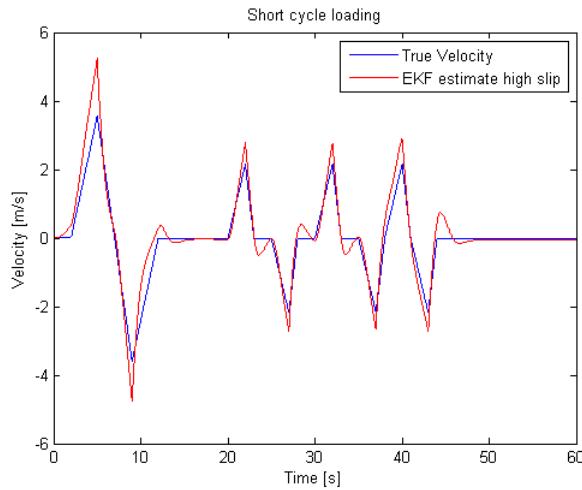


**Figure 6.21:** Estimated velocity for the extended model during transport.

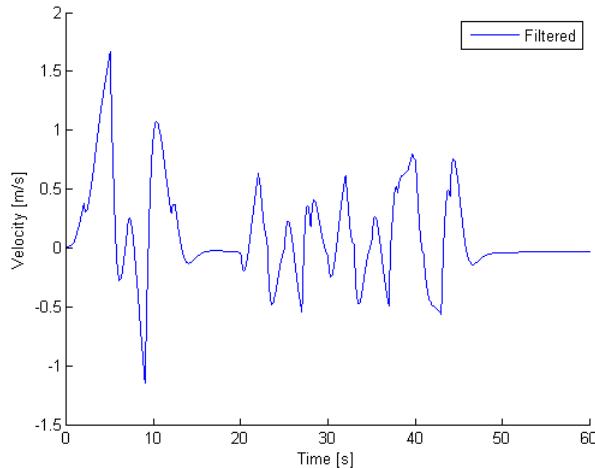


**Figure 6.22:** Residual between estimated velocity and true velocity for the extended model during transport.

Adding a smoother to the no slip model resulted in little improvement. Implementing a fixed lag smoother with a fixed lag of 10 made the filter somewhat better. The standard model smoother without slip states handled low slips slightly better than the filtered version. At high slip values the smoother gave worse velocity estimates. Otherwise the results are similar to those of the filter.



**Figure 6.23:** Estimated velocity for the extended model during SCL.



**Figure 6.24:** Residual between estimated velocity and true velocity for the extended model during SCL.

For the standard model case with slip states a smoother did not help to stabilize the estimates in simulations. Smoothing the standard model lowered the residual between estimated velocity and true velocity slightly, but it still diverged for the transport cycle. The small improvement was not considered to be enough to use

the smoother.

For the extended model similar results to those of the standard model were observed. The smoother implemented for this filter is with a time lag of two samples. Since the filter contains 18 states a lag of 10 means 198 states which was considered to be too large for any real implementation. See Appendix A for plots comparing smoothers to filters.

## 6.7 Conlusions

- The standard model without slip states showed robust results, however the difference between this filter and an average of the four wheel speeds was small.
- Adding residuals to the filter proved to make a big difference in the two simulated use cases.
- The standard model failed to estimate the velocity in the transportation test. This is thought to depend on the observability issues while going straight forward discussed in Section 4.8.1. The short cycle load showed better results than the filter without slip states.
- Adding 1hz velocity sensor to the standard model during transportation improved the estimates, however the estimates where still less accurate than the filter without slip states.
- The extended model gave similar results to that of the standard model.
- Smoothers showed more accurate estimates than the filters, however not enough to make up for the extra computational load.



# 7

---

## Experiments

This chapter presents the test scenarios and results from estimating velocity for data gathered on the wheel loader.

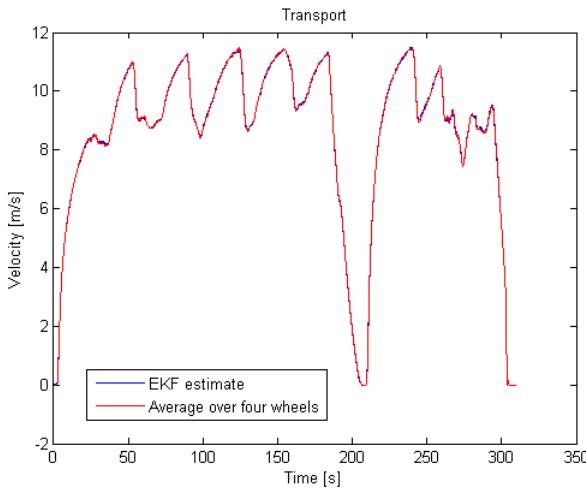
### 7.1 Test set up

The data was collected during real driving scenarios representing transport and SCL. After gathering measurements from the wheel loader these were then processed in Matlab®. During testing the CAN bus was overloaded which lead to varying sampling intervals. However this problem does not affect code running on the ECU, since the ECU constantly updates at 50Hz with new measurement data. Offline this problem was solved by interpolating between the measurements to get a stable 50Hz rate.

The transport cycle is well represented by the cycle in Figure 6.1. During testing there was no dumper available, making the short cycle loading hard to represent. SCL was performed by going forward with the wheel loader towards a gravel pile and then filling the bucket. After this emptying the bucket and going backwards. The data was collected on an approximately flat surface.

During the experiments a velocity reference was to be added to the wheel loader. However no accurate reference was mounted on the machine in time for this thesis work. Without a reference it is hard to comment on the quality of the estimates. Conclusions can still be drawn about stability and convergeability of the filters. The estimates in this section are compared to the average over the four wheels.

The models tested in the EKF in experiments are similar to those tested in simu-



**Figure 7.1:** Estimated velocity using EKF without slip states for the transport cycle.

lations:

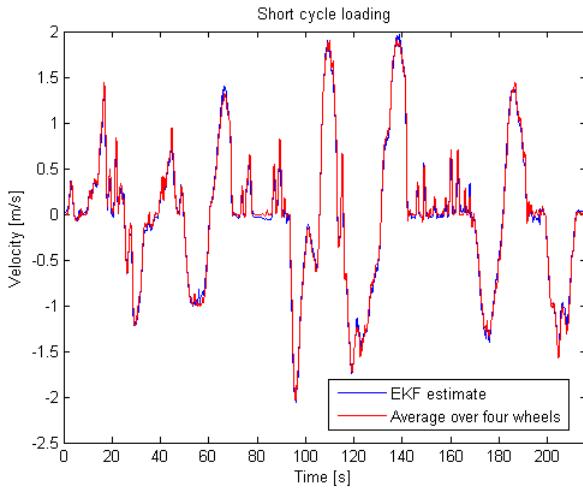
- Standard model without slip states. This is the model described in section 4.5.1, however all slip states  $\delta_{ii}$  have been removed.
- Standard model without slip states, with slip handling as discussed in section 5.1.
- Standard model as described in section 4.5.1.
- Extended model as described in section 4.6.

## 7.2 Standard model without slip states

This section presents the results from the standard model EKF described in section 4.5.1, with the states for slips removed. The velocity estimates can be seen in Figures 7.1 and 7.2. In both scenarios the velocity estimate follows the average over the four wheels very precisely which is similar to what was observed in Chapter 6.

### 7.2.1 With slip handling

This section describes the results when using the same filter, see section 7.2, however using the slip handling as described in section 5.1. The velocity estimates for



**Figure 7.2:** Estimated velocity using EKF without slip states for the SCL.

transport and short cycle loading can be seen in Figures 7.3 and 7.4 respectively. The differences between the results with and without residuals are very small. This might mean that during the collection of the measurements the slip was not very high. It might also be as in the high slip version of the SCL simulation that a large but equal slip is present. This makes the velocity estimate as much false as the slipping wheel speeds, making the EKF follow the wrong velocity estimate.

## 7.3 Standard model

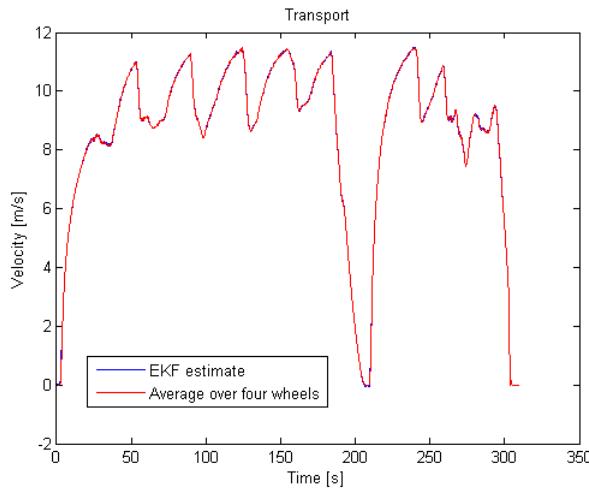
This section presents the results from the standard model EKF, as described in section 4.5.1. The filter is during real world tests able to handle both transport and SCL better than in the simulations.

### Transport

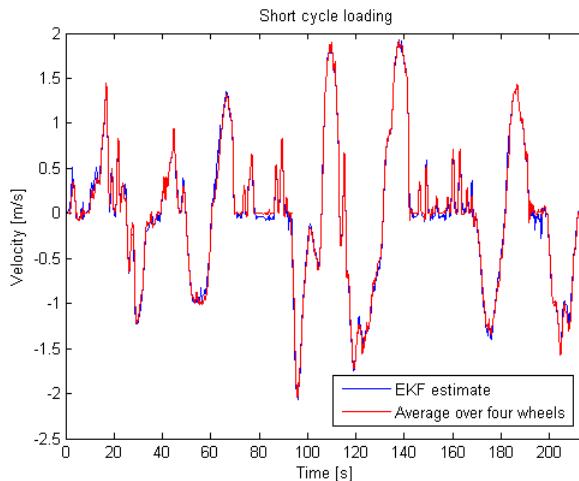
The velocity estimate for the transport cycle can be seen in Figure 7.5. The transport cycle is handled worse than the SCL. During transportation in simulations the estimates had a constant offset after accelerating whereas in Figure 7.5 the estimate more closely follows the average over the four wheels. The results from the real world tests are more interesting since in simulations the offset can be due to modelling errors.

### Short cycle load

The velocity estimate for the SCL can be seen in Figure 7.6. At around 150s in Figure 7.6 the wheel loader is entering the pile and changing velocity at a quick



**Figure 7.3:** Estimated velocity using the standard model without slip states with sensor check for the transport cycle.

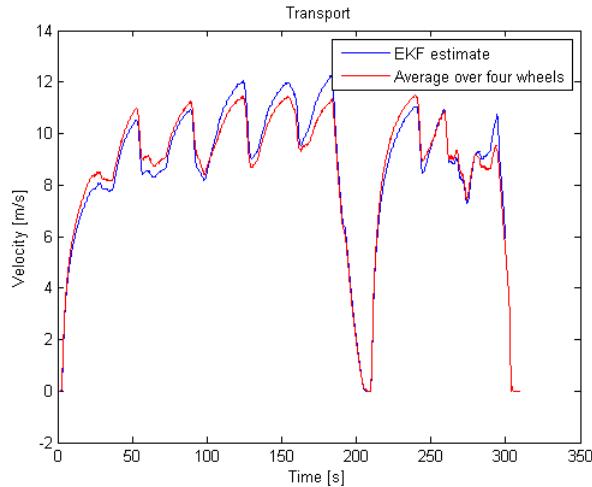


**Figure 7.4:** Estimated velocity using the standard model without slip states with sensor check for the SCL.

rate. This was thought of as a hard scenario for estimation but is handled well by the filter.

During simulations the filter converged after about 30s during SCL whereas in Figure 7.6 it can be seen that the estimate follows the average over the four wheels

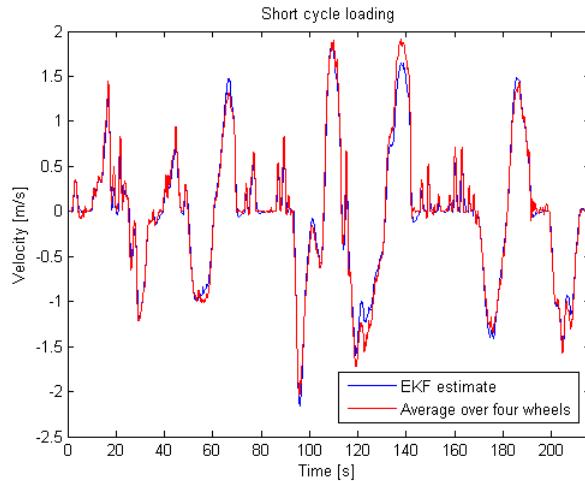
from the start. The reason is thought to be model errors in the simulations. During the test scenarios in simulations the wheel loader behaves unnaturally, with very sharp peaks at changing velocities.



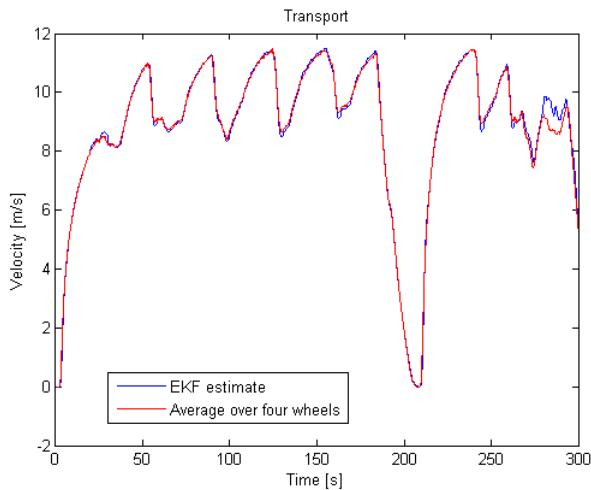
**Figure 7.5:** Estimated velocity using the standard model with slip states for the transport cycle.

## 7.4 Extended model

This section provides the results for the extended model, described in section 4.6, with omitted slip states. The velocity estimates can be seen in Figures 7.7 and 7.8. The results are similar to that of the standard model without slip states. The estimate moves slower than that of the standard model. This might be due to tuning. At around 270 seconds in Figure 7.7 the filter differs more from the standard model than during the rest of the test. This could be due to the wheel loader starting to brake at this point making the pitch and roll angles change dramatically. As the standard model does not consider pitch and roll that filter is not affected whereas the filter for the extended model is.



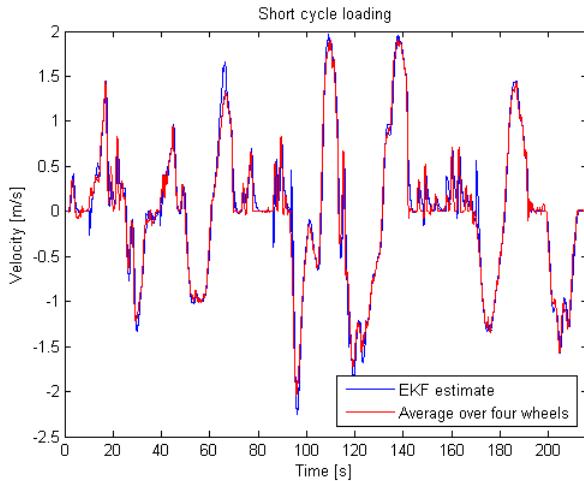
**Figure 7.6:** Estimated velocity using standard model with slip states for the SCL.



**Figure 7.7:** Estimated velocity using the extended model without slip states for the transport cycle.

## 7.5 Conclusions

Despite the lack of an accurate reference signal during the experiments, conclusions can be drawn on stability. The estimates from the experiments are more stable than the estimates from simulated data presented in Chapter 6. This could



**Figure 7.8:** Estimated velocity using the extended model without slip states for the SCL.

indicate that simulation model is inaccurate in the chosen use cases. The results for the filters described in section 7.1 are:

- The estimate from the filter using the standard model without slip states resembles the average over the four wheels.
- Adding slip handling did not improve the estimates. This could be because the real world scenarios contain low slip values, or that the slip threshold was set too high.
- The filter using the standard model with slip states handles the SCL better than the transportation which agrees with the results from simulations. The estimate did not diverge during transportation which was the case during simulations.
- The extended model showed similar results as that of the standard model.



# 8

---

## Concluding remarks

Different filters, described in section 6.1, for estimating the speed of a wheel loader and handling slip have been tested. The filters have different drawbacks. The simulation results and the results for real data are not completely agreeing. The reason for this is thought to be that the simulated model behaves unrealistically, especially due to steering and engine control.

The filter using the model without slip states proved to be robust. The filter could handle all test cases, both in simulations and during experiments. The filter is not sensitive to tuning and estimated the velocity well during low slip scenarios. The resulting estimates were close to an average of the four wheel speeds. At high slip scenarios the filter gave inaccurate speed estimates as it did not compensate for wheel slip. The solution provided to this problem is to use residuals to handle slip in the measurements by comparing wheel speed measurements with the estimated velocity of the filter. The filter with residuals showed big improvements for high slip scenarios. However the filter still cannot handle low amounts of slip. Low amounts of slip does not affect the velocity estimate too much so this was not considered a big problem. Fine-tuning the residual threshold is also thought to make the filter handle low slip values.

The filter using the standard model with slip states seems to suffer from observability issues. Interestingly the thought of an easy scenario of transport was not handled well by the filter. During the SCL the filter showed a lot better results than for that of the transport scenario. The filter showed adequate velocity estimates but failed to estimate the slip states. In real world tests this filter proved to work better than in simulations. The tests done during this thesis make it hard to draw conclusions on these results. A possible reason could be that the models in the EKF represent the real world case better. In simulations a velocity mea-

surement with a sampling rate of 1Hz was introduced and improved the results substantially. To do real world tests a GPS was to be added to the machine but this was not finished in time for this thesis.

Of the evaluated filters, the filter using the extended model for full motion is the most interesting for Volvo Construction Equipment. The results for the extended model were similar to that of the standard model with slightly bigger errors. During the time frame of the thesis the tests performed were done on flat ground. The extended model was developed to handle non flat movements better than the standard model. In the real world test scenarios the filter handled both scenarios adequately. The extended model includes dynamics for rotations in all directions meaning it is more dependent on choice of IMU than the other filters. This also means it might suffer more from vibrations with an inadequate IMU. As the vibrations of the machine are hard to model in simulations it is interesting to see if the extended model can handle these in the real world tests. The extended model with slip states was thought to have similar results as that of the standard model and was therefore not prioritized during testing.

## 8.1 Future work

The possibilities for future work are many. Wheel loaders are multi purpose machines, therefore testing for more scenarios could be interesting. Many of the filters were sensitive to tuning. Spending more time on this could improve the results.

The most important future work is to do real world data tests with a reference. Without a reference it is hard to draw conclusions about the real world situations. Longer measurement collections could be performed as well. During a longer test the IMU will drift more and a longer test will also test the robustness of the filters.

Adding more sensors to see if the slips can be estimated is a very interesting subject in theory. Although in reality cost issues can set constraints on this. Visual odometry or a good GPS could be added, a magnetometer could also be tested.

Further developing the model without slip to make it robust is interesting. Handling drifting pitch and roll angles by pre-processing or auto-tuning the sensors at start-up could be one possibility.

Identifying working situations could also be done together with a multiple model approach. As was pointed out in the conclusions the models handle different situations differently. For example during transportation with very low slip just using the four wheel speeds gave good speed estimates. However at high slip situations such as SCL a combination of sensors and more advanced filtering could be used.

# **Appendix**



# A

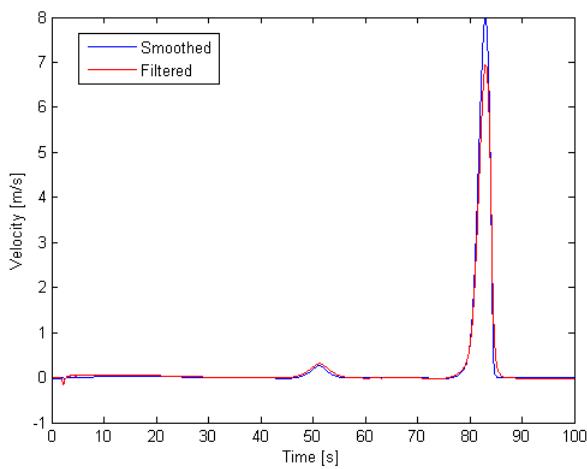
---

## Smoother

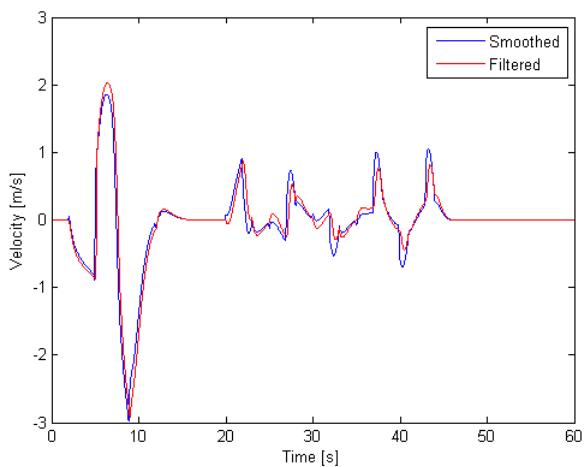
Below follows a brief comparison of smoothers and filters. The smoothers for the standard model with and without slip states are with a time lag of 10, for the extended model a time lag of 2 was used. In Figure A.1 the smoothed estimate for the standard model without slip during transport can be seen. During turning at around 50 seconds the smoothed estimate shows better results than for that of the filtered. At 80 seconds with big slip values on one wheel the smoother performs worse.

In Figure A.2 the residuals for the smoother and filter during SCL for the standard model can be seen. The smoothed estimate reacts quicker to changes in velocity however the smoother did not improve the velocity estimate in general. During transport the filter still failed to estimate the velocity.

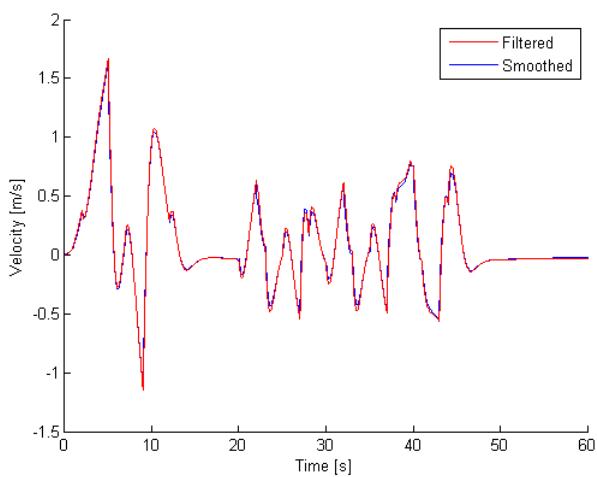
Figure A.3 shows the smoothed estimate for the extended model compared to the filtered estimate. A shorter time lag affects the smoothed estimate less than in the 10 lag scenario of Figures A.1 and A.2. The smoothed version shows minor improvements at for example around 35 seconds.



**Figure A.1:** Difference between smoothed estimate and filtered estimate for standard model without slip states during transport.



**Figure A.2:** Difference between smoothed estimate and filtered estimate for the standard model during SCL.



**Figure A.3:** Difference between smoothed estimate and filtered estimate for the extended model during SCL.



---

# Bibliography

- Accurate technologies. URL <https://www.accuratetechnologies.com/>. Cited on page 5.
- Matlab. URL <https://www.mathworks.com/>. Cited on pages 28 and 39.
- Howard Anton and Chris Rorres. *Elementary Linear Algebra with Supplemental Applications*. John Wiley and Sons, 11 edition, 2014. Cited on page 36.
- Peter I. Corke and Peter Ridle. Steering kinematics for a center-articulated mobile robot. *IEEE Transactions on robotics and automation*, 17(2):215–218, 2001. Cited on pages 2 and 23.
- Armin Daiß and Uwe Kiencke. Estimation of vehicle speed fuzzy-estimation in comparison with kalman-filtering. In *Proceedings of the 4th IEEE Conference on Control Applications*, pages 281–284, 1995. Cited on page 1.
- Hanno Essén. Motion in accelerated reference frames, 2002. URL <http://www.mech.kth.se/~hanno/RelativeMotion.pdf>. Cited on page 22.
- Torkel Glad. Extending linear theory to nonlinear systems, 2012. URL <http://www.control.isy.liu.se/student/graduate/LinSys/kompendium/gesy.pdf>. Cited on page 28.
- Fredrik Gustafsson. Rotational speed sensors: Limitations, preprocessing and automotive applications. *Instrumentation and Measurement Magazine, IEEE*, 13(2):16–23, 2010. Cited on page 2.
- Fredrik Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2nd edition, 2012. Cited on pages 16, 34, 36, and 37.
- Fredrik Gustafsson, Lennart Ljung, and Mille Millnert. *Signal Processing*. Studentlitteratur. Cited on page 17.
- Fredrik Gustafsson, Niclas Persson, Urban Forssell, and Stefan Ahlqvist. Sensor fusion for accurate computation of yaw rate and absolute velocity. In *SAE World Congress*, 2001. SAE Technical Paper 2001-01-1064. Cited on page 1.

- Jeroen Hol. *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. PhD thesis, Linköping University, 2011. Cited on page 2.
- R.E. Kalman. A new approach to linear filtering and prediction problems. *ASME-Journal of Basic Engineering Series D*, 82(2):35–45, 1960. Cited on page 34.
- Marco Di Nitale, Haibo Zeng, and Pablo Giusto. *Understanding and Using the Controller Area Network Communication Protocol*. Springer-Verlag, 2012. Cited on page 5.
- Mikael Olofsson. *Signal Theory*. Studentlitteratur, 2011. Cited on page 7.
- Hans B. Pacejka. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, 2nd edition, 2006. Cited on page 41.
- S. Scheding, G Dissanayake, E. Nebot, and H. Durrant-Whyte. Slip modelling and aided inertial navigation of an lhd. In *International Conference on Robotics and Automation*, volume 3, pages 1904–1909, 1997. Cited on page 2.
- D.H. Titterton and J.L. Weston. *Strapdown inertial navigation technology*. The institution of Electrical Engineers, 2nd edition, 2004. Cited on pages 2 and 13.
- J. Y. Wong. *Theory of ground vehicles*. John Wiley & Sons, 4th edition, 2008. Cited on page 19.
- Oliver J. Woodman. *Pedestrian localisation for indoor environments*. PhD thesis, University of Cambridge, 2010. Cited on page 27.